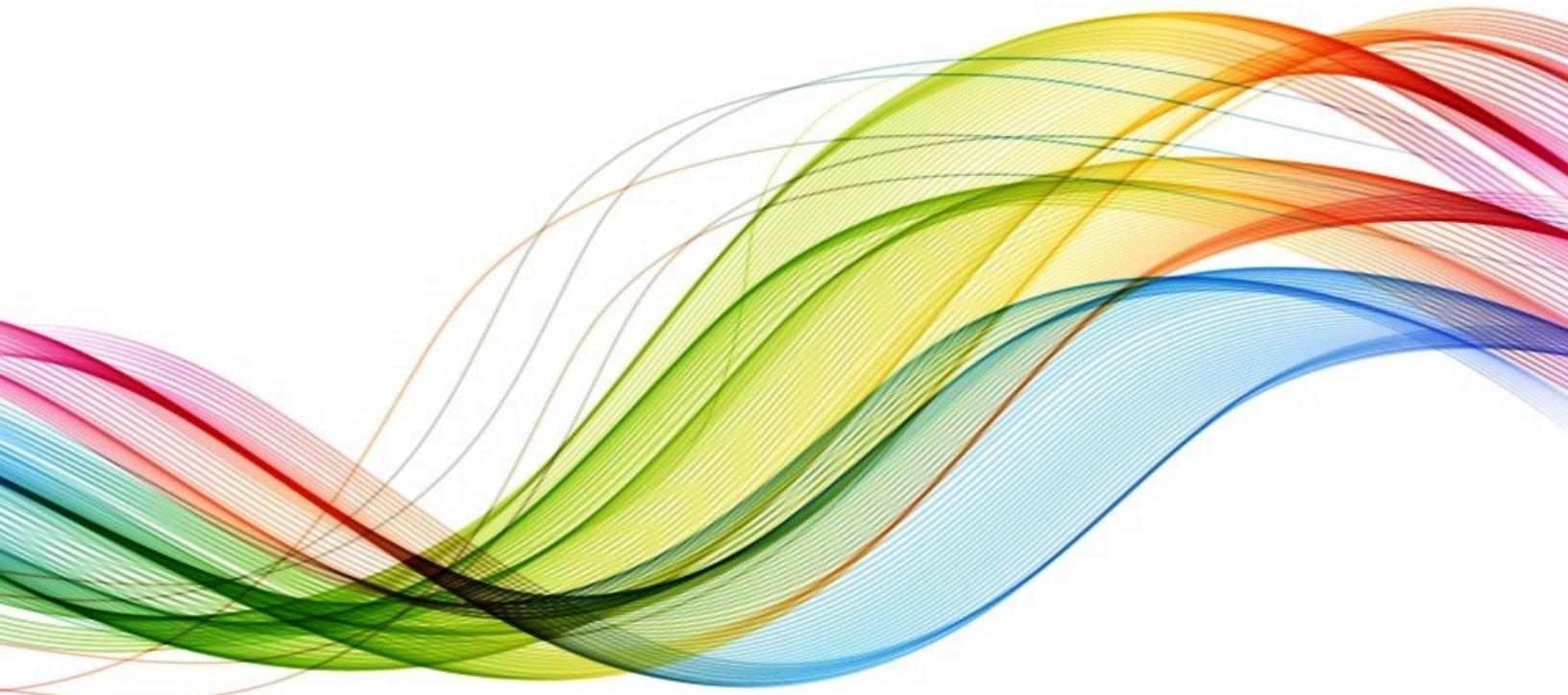


Modern Time Series Forecasting Techniques

For Predictive Analytics and
Anomaly Detection



CHRIS KUO



From Classical Foundations to Cutting-Edge Applications

INNOVATION
 PRESS

Copyright © 2024 Chris Kuo, Ph.D.

All rights reserved. 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 the prior written permission of the publisher, or in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law.

While every effort has been made to ensure the reliability of the information presented in this publication, Innovation Press assumes no responsibility for errors or omissions. However, neither the author nor Innovation Press assumes any liability for damages resulting from the use of the information contained herein.

ISBN 979-8-9907810-0-9

First edition: April 2024

Second impression, with minor extensions: May 2024

This book was typeset using L^AT_EX software. The T_EX template acknowledgement:

The *Beautybook* [<https://ctan.org/pkg/beautybook?lang=en>]

Comprehensive TEX Archive Network (CTAN)

The Innovation Press,
Ann Arbor, MI





ABOUT THE AUTHOR

Chris Kuo is a data scientist and an adjunct professor with 23+ years of experience. He led various data science solutions including customer analytics, health data science, fraud detection, and litigation analytics. He is also an inventor of a U.S. patent. He has worked at several Fortune 500 companies in the insurance and retail industries. In addition to teaching at Columbia University, he has taught courses in time series forecasting, mathematical finance, economics, and management at Boston University, University of New Hampshire, and Liberty University. Chris Kuo received his Ph.D. in Economics from SUNY at Stony Brook and his B.S. in Nuclear Engineering from National Tsing-Hua University, Taiwan. He and his wife live in New York, New York.

Chris Kuo is a writer on Medium.com, also known as 'Dr. Dataman' in the field of data science. He has published in several economic and management journals and is the author of the following data science books (by year):

- 2021 - *Modern Time Series Anomaly Detection: With Python and R examples*
- 2022 - *The eXplainable A.I.: With Python examples*
- 2022 - *Transfer Learning for Image Classification: With Python examples*
- 2023 - *The Handbook of Anomaly Detection: Build and modernize your anomaly detection models with examples*
- 2023 - *The Handbook of NLP with Gensim: Leverage topic modeling to uncover hidden patterns, themes, and valuable insights within textual data*
- 2024 - *Modern Time Series Forecasting: For Predictive Analytics and Anomaly Detection*

To my son, Nathan Kuo,
When you were 5 years old and knew some ABCs,
your mom told you there was an academic journal that has dad's paper.
You eagerly picked it up and wanted to read it as a storybook.
In that moment, I regretted why I had not written a book for you.
Now I dedicate this book to you.
Your dad.

For God so loved the world, that he gave his only begotten Son,
that whosoever believeth in him should not perish,
but have everlasting life. (John 3:16)



PREFACE

Time series data is ubiquitous in today's world, and its analysis is critical for various applications, such as finance, healthcare, and manufacturing. As the volume and complexity of data continue to grow, innovative techniques are needed to address the challenges of the evolving nature of data and effectively identify anomalies in real time. This book provides an overview of classical time series forecasting and then advances to state-of-the-art time series techniques. The contents within these pages are crafted to inform, inspire, and empower your endeavors.

Sec / What this book covers

This book organizes time series models and applications into six meticulously crafted parts. Each part equips you with the knowledge and skills to conduct time series forecasting and anomaly detection. The six parts are:

- Part 1: From Prophet to NeuralProphet
- Part 2: Getting Probabilistic Forecasts
- Part 3: Autoregressive-based Time Series Techniques
- Part 4: Tree-based Time Series Techniques
- Part 5: Deep into Deep learning-based Time Series Techniques
- Part 6: Transformer-based Time Series Techniques

Part 1 starts with Prophet and its successor NeuralProphet because they are an intuitive way to model time series data with interpretable results and user-friendly APIs. Both have active communities of users and contributors. **Part 2** teaches you the techniques for forecasting uncertainty. You will be able to build projects with forecasting uncertainty to help your users make more informed decisions and manage risks. **Part 3** and **Part 4** offer theories and applications into Autoregressive Integration Moving Average (ARIMA) modeling and modern tree-based techniques. You will gain valuable insights into algorithmic approaches and their practical implementation. **Part 5** and **Part 6** explore cutting-edge research and sophisticated techniques including deep learning architectures and Transformer-based methods. Therefore Part 1 to Part 6 range from classical models to the modern large foundational models in time series.

Sec Why read this book?

In this book, you will find a wide coverage of methodologies, algorithms, and applications. Whether you're a seasoned data scientist seeking to refine your expertise or a novice eager to embark on your analytical odyssey, this book offers a roadmap tailored to cater to diverse skill levels and objectives. You will emerge equipped with the proficiency and confidence to unravel intricate temporal patterns, harness predictive power, and unlock new horizons of insight across a myriad of domains, from finance and economics to healthcare and beyond.

The writing style of this book is another selling point. Instead of taking a technique as given, this book first describes intuitions and then dives into detail. This book also gives a landscaping view from one idea to the consequent ideas. With the real-world data cases in this book, you will gain a deeper understanding of how time series techniques are applied in diverse domains.

Sec Open-source Python libraries

Another key feature in this book is the integration of open-source Python time series libraries. They include: Prophet, NeuralProphet, Darts, GluonTS, SKtime, Pmdarima. By leveraging these libraries, you can expedite data preprocessing, explore temporal patterns, and deploy sophisticated forecasting algorithms with ease. This integration of these tools in the book let you experience the production efficiency and convenience for your proof-of-concept initiatives.

Sec Who this book is for

Whether you're seeking to enhance your skill set, solve complex problems, or simply satisfy your curiosity, this book offers a compelling journey into the captivating world of time series anomaly detection.

- **Data Scientists and Analysts:** Professionals who work with time series data and need to identify anomalies for various applications such as finance, cybersecurity, IoT, and healthcare. They will benefit from the practical methodologies and real-world examples provided in the textbook.
- **Researchers:** Academics and researchers exploring novel approaches to anomaly detection in time series data, seeking a comprehensive reference and practical insights to inform their research endeavors.
- **Students:** Undergraduate and graduate students studying data science, machine learning, statistics, or related fields, looking to gain expertise in time series analysis and anomaly detection. The book offers a structured learning path with clear explanations and hands-on exercises.
- **Industry Practitioners:** Engineers, developers, and decision-makers in industries where anomaly detection is crucial for maintaining operations, ensuring quality, and detecting fraud. They will find the book's practical focus and real-world case studies invaluable for enhancing their skills and solving real-world problems.

Sec

Recommended prerequisites

- Basic Knowledge of Statistics: Familiarity with concepts such as mean, median, standard deviation, probability distributions, and hypothesis testing will help readers grasp the statistical methods used in anomaly detection.
- Programming Skills: Proficiency in at least one programming language such as Python or R for implementing algorithms, visualizing data, and analyzing results.
- Understanding of Machine Learning: Basic understanding of machine learning concepts including supervised and unsupervised learning algorithms, feature engineering, and model evaluation.
- Prior Exposure to Time Series Data: Some experience working with time series data and understanding its unique characteristics such as trends, seasonality, autocorrelation, and stationarity will aid in comprehending the challenges and techniques discussed in the textbook.

Sec

Motivation for writing this book

Traditional time series forecasting methods, such as AutoRegressive Integrated Moving Average (ARIMA), have been widely used for decades. New techniques, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and deep learning architectures, become the new foundations for time series forecasting. Tree-based models including gradient-boosting models and LightGBM (2017) topped the time series forecasting competitions (the M competition in 2020s). Even more, after the arrival of the Transformer model in 2017, new methods including the Temporal Fusion Transformer (2019), and Lag-LLAMA (2024) came to the stage. These techniques are being used in various applications, such as finance, energy, and transportation, to forecast stock prices, electricity demand, and traffic flow.

With these unprecedented developments in time series forecasting techniques and applications, I have been longing to write a book to cover the traditional methods and bridge new techniques. This book covers new techniques developed from 2014 to the present.

The global Pandemic hit in 2019. Many office buildings around the world were closed for one and a half years. I remember the last afternoon before office closure, I rushed to the food truck to get an egg sandwich. I had one bite or two of my egg sandwich between busy meetings. Then the following months the whole world was all working remotely. During that period, we continued to witness notable large language models flourishing. During the Pandemic year, I had a dream. In my dream, I strolled through an empty Times Square. I went into the empty office building to pick up my notes. Then I found my half-finished egg sandwich was still on my desk!

Chris Kuo,
New York City, New York



CONTENTS

1 Introduction 1

1.1	Use Forecasts for resource planning and anomaly detection	1
1.1.1	Case study 1: Retail inventory management	2
1.1.2	Case study 2: Electricity demand forecasting	2
1.2	Anomaly score definitions	3
1.3	Model interpretability	3
1.4	Local models vs. global models	4
1.5	A roadmap to learn modern time series forecasting	4
1.6	Time series Python libraries	5
1.7	Model evaluation metrics	6
1.7.1	Mean Absolute Percentage Error (MAPE)	6
1.7.2	Mean Squared Error (MSE)	7
1.7.3	Mean Absolute Error (MAE)	7
1.7.4	Root Mean Squared Error (RMSE)	7
1.7.5	Symmetric Mean Absolute Percentage Error (sMAPE)	8
1.7.6	Continuous Ranked Probability Score (CRPS)	8
1.8	Cheat Sheets	9
1.9	Download the example code files	9
1.10	Data for this book	9

I PART I: From Prophet to NeuralProphet 11

2 Business Time Series Forecasting 13

2.1	The Generalized Additive Models (GAMs)	14
2.2	The Case for Prophet	15

2.3	Data preparation	15
2.3.1	In-time and out-of-time data split	16
2.4	Modeling with Prophet	16
2.5	Change Point Detection in Prophet	18
2.6	Model explainability	18
2.7	Out-of-time model performance	20
2.8	Model performance diagnostics through all periods	20
2.9	Hyper-parameter tuning	22
2.10	Conclusions	25
2.11	Appendix: Prophet prediction intervals	25

3 NeuralProphet I 28

3.1	The case for NeuralProphet	28
3.2	The components of NeuralProphet	29
3.3	Data preparation	30
3.4	Modeling with NeuralProphet	31
3.5	Understanding the trend module	32
3.5.1	The determination of the change-point locations	34
3.5.2	Trend experiment 1: No change point	35
3.5.3	In-time and out-of-time data split	37
3.5.4	Trend experiment 2: automatic change point detection	37
3.6	Understanding the seasonality module	39
3.6.1	Seasonality + Trend without change point	39
3.6.2	The Fourier terms in the seasonality module	40
3.6.3	Seasonality + Trend with change points	41
3.7	Trend + Seasonality + Holidays and Events	42
3.8	Custom events	43
3.9	Conclusions	44

4 NeuralProphet II 46

4.1	Understanding the classical AR models	46
4.2	AR-Net	47
4.3	The AR module of NeuralProphet	48
4.4	Trend + Seasonality + Holidays + AR	50

4.5 AR with hidden layers	53
4.5.1 AR experiment 1: AR-[32,32] model	53
4.5.2 AR experiment 2: AR-[8,8] model	55
4.6 Trend + seasonality + holidays + AR + lagged regressors	56
4.7 Trend + seasonality + holidays + AR + lagged regressors + future known regressors	58
4.8 Model comparison and selection	60
4.9 Forecasting future periods	60
4.9.1 Steps 1 and 2: data preparation and modeling	61
4.9.2 Step 3: Make future periods and forecast	61
4.9.3 Step 4: Append the out-of-time actual data	62
4.9.4 Step 5: Plot the actual values vs. forecasts	62
4.10 Conclusions	63

5 Change Point Detection in Time Series 64

5.1 Algorithms depends on real-time or offline data	64
5.2 Data generation	65
5.3 Offline CPD	66
5.3.1 Constant variance	67
5.3.2 Varying variance	68
5.4 Real-time CPD	68
5.4.1 Constant variance	69
5.4.2 Varying variance	71
5.5 Conclusions	72

II PART II: Getting Probabilistic Forecasts 73

6 Monte Carlo Simulation for Probabilistic Forecasting 75

6.1 Gaussian distribution	77
6.2 Student's t-distribution	78
6.3 Conclusions	83

7 Quantile Regression for Probabilistic Forecasting 84

7.1	Data preparation	85
7.2	Prediction intervals and confidence intervals are different	88
7.3	Conclusions	89

8 Conformal Prediction for Probabilistic Forecasting 90

8.1	Data preparation	92
8.2	Modeling	93
8.2.1	Training, Calibration, and Test Data.	93
8.3	Conformal Predictions	94
8.4	Conclusions	96

9 Conformalized Quantile Regression for Probabilistic Forecasting 98

9.1	The Concept of CQR	98
9.2	The Construction of CQR	100
9.3	Data preparation	101
9.4	Modeling	102
9.4.1	Training, Calibration, and Test Data.	102
9.5	Conformal Quantile Regression	103
9.6	Conclusions	105

III PART III: Autoregressive-based Time Series Techniques 106

10 Automatic ARIMA! 108

10.1	Data preparation	109
10.1.1	In-time and out-of-time data split.	110
10.2	The ARIMA models	111
10.2.1	Differencing	111
10.2.2	Use the ACF to suggest the orders of MA	113
10.2.3	Use the PACF to suggest the orders of AR	113
10.3	Automatic ARIMA	114
10.3.1	Multi-period forecasts	116
10.3.2	Update the model in each iteration	117
10.4	The SARIMA models	118

10.5 Conclusions	121
----------------------------	-----

11 Time Series Data Formats Made Easy 122

11.1 Data preparation	123
11.1.1 Get a long-form dataset	123
11.1.2 Learn to convert a long-form dataset to a wide-form	124
11.2 Darts	125
11.2.1 Darts — Loading from a long-form Pandas data frame	125
11.2.2 Darts — Loading from a wide-form Panda data frame	126
11.2.3 Darts — Plotting	127
11.2.4 Darts — Loading from a univariate Pandas series	127
11.2.5 Darts — Converting back to Pandas	128
11.2.6 Darts — Converting to Numpy array	129
11.2.7 Darts applications	129
11.3 GluonTS	130
11.3.1 GluonTS — Loading from a long-form Pandas data frame	130
11.3.2 GluonTS — Loading from a wide-form Pandas data frame	131
11.3.3 GluonTS - Any general conversion	131
11.3.4 GluonTS — converting back to Pandas	132
11.3.5 GluonTS applications	133
11.4 Sktime	134
11.5 Pmdarima	134
11.6 Prophet/NeuralProphet	135
11.7 Conclusions	137

IV PART IV: Tree-based Time Series Techniques 138

12 Linear Models for Multi-period Probabilistic Forecasting 140

12.1 Data preparation	142
12.1.1 Build a local model	143
12.1.2 In-time and out-of-time data split	144
12.1.3 Target, past and future covariates	144

12.2 Modeling: Univariate data only, no covariates	144
12.2.1 The plot for the actual vs. predicted values	145
12.2.2 Model performance	145
12.3 Modeling: Adding past and future covariates	146
12.3.1 The plot for the actual vs. predicted values	147
12.3.2 Model performance	148
12.4 Quantile forecasts	148
12.4.1 The plot for the actual vs. predicted values	149
12.5 Conclusions	149

13 Feature Engineering for Tree-based Time Series Models 151

13.1 Data preparation	151
13.2 Creating date-related features from a univariate time series	153
13.3 Create lagged and future features	154
13.4 Creating moving averages as features	156
13.5 Supervised learning framework for one-period forecast	157
13.6 Building lightGBM forecasting models	157
13.6.1 In-time and out-of-time data split	158
13.7 lightGBM modeling	158
13.8 Model performance	159
13.9 Visualizing the actuals vs. predictions	160
13.10 Model explainability	161
13.11 Conclusions	161

14 Two Primary Strategies for Multi-period Forecasting 163

14.1 Data preparation	164
14.1.1 In-time and out-of-time data split	165
14.2 Recursive forecasting strategy	166
14.3 The direct forecasting strategy	168
14.4 Conclusions	171

15 Tree-based XGB, LightGBM, and CatBoost Models for Multi-period Probabilistic Forecasting 172

15.1 Data preparation	172
15.1.1 In-time and out-of-time data split	174
15.2 XGB	175
15.2.1 Probabilistic forecasting with Darts	177
15.3 LightGBM	179
15.4 CatBoost	182
15.5 Conclusions	183

V PART V: Deep into Deep Learning-based Time Series Techniques 186

16 The Progression of Time Series Modeling Techniques 188

16.1 Moving average and exponential smooth methods	188
16.2 Classical univariate seasonal-trend decomposition model	189
16.3 Autoregressive models to capture sequential patterns	190
16.4 Kalman filter to capture latent patterns	190
16.5 State space model to capture latent sequential patterns	191
16.6 From point estimates to probabilistic predictions	192
16.7 From state space models to multilayer perceptrons (MLPs)	193
16.8 Recurrent Neural Networks (RNN)	194
16.9 From RNN to LSTM	196
16.10 From LSTM to GRU (Gated Recurrent Units)	198
16.11 Conclusions	199

17 Deep Learning-based DeepAR for Probabilistic Forecasting 201

17.1 Deep learning framework naturally extending to global models	202
17.2 The architecture of DeepAR	202
17.3 Understanding the engineering of GluonTS	204
17.4 Data preparation	205
17.4.1 In-time and out-of-time data split	206
17.5 Data Preparation in GluonTS	207
17.6 Modeling with GluonTS	207
17.7 Forecasting in GluonTS	208

17.8 Probabilistic forecasts	209
17.9 Graphic outcomes	209
17.10 Evaluation metrics	210
17.11 Conclusions	211

18 Application — Probabilistic Predictions for Stock Prices 212

18.1 Data preparation	213
18.1.1 In-time and out-of-time data split	213
18.2 Conclusions	217

VI PART VI: Transformer-based Time Series Techniques 218

19 From RNN to Transformer-based Time Series Models 220

19.1 From RNN/LSTM to Sequence-to-sequence (Seq2Seq) models	220
19.2 The attention mechanism	223
19.3 From attention to self-attention	224
19.4 The Transformer Model	225
19.5 From Transformer to Temporal Fusion Transformer (TFT)	227
19.6 Why large foundational models for time series?	228
19.7 Lag-Llama— an open-source time series foundational model	229
19.8 Conclusions	230

20 Temporal Fusion Transformer 232

20.1 Building global time series models	233
20.2 The architecture of TFT	233
20.2.1 Variable selection network	234
20.2.2 Gated residual network (GRN)	235
20.2.3 Static covariate encoders	236
20.2.4 LSTM encoders	236
20.2.5 LSTM decoders	237
20.2.6 Static enrichment with GRNs	237
20.2.7 Temporal multi-head self-attention	237
20.2.8 Position-wise feed-forward network	238

20.2.9 Quantile regression outputs for multiple periods	239
20.3 Optimization	239
20.4 Data preparation	240
20.4.1 Data preparation	240
20.4.2 In-time and out-of-time data split	241
20.4.3 Data conversion to Darts	242
20.4.4 Create time covariates	243
20.4.5 Standardization	244
20.5 Modeling	245
20.6 Forecasting	246
20.7 Model interpretability	249
20.7.1 Encoder importance	249
20.7.2 Decoder importance	250
20.7.3 Static covariate importance	250
20.7.4 Attention per horizon	251
20.8 Appendix	252
20.8.1 Dropout	252
20.8.2 Batch size	252
20.8.3 Epoch	253
20.8.4 My hardware	253
20.9 Conclusions	253

21 Lag-Llama for Time Series Forecasting 255

21.1 Inputs — Lagged covariates and calendar features	255
21.2 The architecture of Lag-Llama	256
21.3 Probabilistic forecasts	257
21.4 Zero-shot learning and few-shot learning	257
21.5 Data preparation	259
21.5.1 Plot the data	260
21.5.2 In-time and out-of-time data split	261
21.5.3 Convert to GluonTS format	261
21.6 Lag-Llama Modeling	262
21.7 Forecasting	263

21.8 The Continuous Ranked Probability Score (CRPS) for evaluating probabilistic forecasts	264
21.9 Conclusions	265

Index	266
--------------	-----

Other Books by Chris Kuo	269
---------------------------------	-----



INTRODUCTION

Part

Time series is a sequence of data at equally spaced intervals over time. It is around us in our daily lives. We derive forecasts from the time series data for various purposes. If you meet Mr. P in this story, you may find one or two persons in your life who are just like Mr. P. And you probably agree you also use time series forecasts around the clock in your life. Mr. P is a middle-aged professional living in a busy city. He usually starts his day with a morning jog. He puts on his fitness tracker for jogging. The tracker monitors his heart rate, sleep patterns, activity levels, and other physiological parameters. His doctor told him the forecasts are based on his health data over time. The forecasts tell him if he is on the way to his fitness goals. The forecasts also alert him to any anomalies or irregularities so he can share them with the doctor. Although he does not like too much regulation, he seems enjoying his disciplined life and has convinced his next-door neighbor to join him next month. Mr. P works as an inventory control manager for a fashion sports company. The company designs and produces athletic apparel, footwear, and accessories for sports enthusiasts. Part of his job is to monitor the trend for sales figures, inventory levels, and customer foot traffic. He needs to predict demand and optimize inventory levels for each line of business. His numbers help other departments to plan promotions, allocate resources, and maximize quarterly revenue. He also buys and sells some stocks on his own. He subscribes to a reputable stock forecasting service for investment guidance. He follows certain stocks for his investment decisions. He reads some investment strategies like algorithmic trading or swing trading. But he seems not too active about trading opportunities, though he has a friend who often brags about the revenues in day trading. Talking about Mr. P's passion, everyone knows it is about hiking and photography. He is a serious photographer — his blog has one million followers. He talks about camera basics like DSLR, mirrorless, point-and-shoot, and other kinds of cameras. He talks about aperture, shutter speed, ISO, and where to buy lenses, tripods, memory cards, and so on. But most of all, he enjoys traveling. He often plans vacations to really remote forests for his nice photo shots. Okay, back to time series forecasts. He checks 10-day weather forecasts to plan for his travel. Around our lives we all have one or two persons like Mr. P. Most of all, we all use time series forecasts like Mr. P.

Sec 1.1 Use Forecasts for resource planning and anomaly detection

We need forecasts to have a range of possible values for planning and monitoring purposes. When the actual values are within our predicted range, we will not be surprised because we have planned the best or worst scenarios accordingly. The terms probabilistic forecasting, prediction uncertainty, or prediction intervals are often used interchangeably in time series forecasting. Just like the white and yellow lines on a road to guide traffic, time series forecasts are used to guide business operations across a wide range of industries. In finance and banking, time series

forecasting is used for predicting possible values of an asset to help investors make informed decisions. In retail and E-Commerce, time series forecasting is used to guide the optimal inventory levels. In healthcare and pharmaceuticals, time series forecasting helps to predict hospital bed occupancy, disease outbreaks, and medical resource demand.

On the other hand, when the actual values are outside the predicted values, they are called outliers or anomalies. We also want to track the chances of them happening to mitigate risks. Either the forecasts are used for resource planning or anomaly detection or both, the probabilistic forecasts derived from the same time series modeling tool can be applied. Let's see the preceding use cases to see how forecasting helps resource planning and anomaly detection.

1.1.1 Case study 1: Retail inventory management

Story: ABC Retail is a large chain of stores specializing in consumer electronics. With thousands of products and fluctuating demand patterns, optimizing inventory levels is crucial to profitability. Anomalies such as sudden spikes or drops in demand can lead to excess inventory and lost sales opportunities.

Technology innovation: ABC Retail requires techniques that can produce a high level of prediction accuracy for the sales of its products. ABC Retail adopts various advanced forecasting techniques to improve prediction accuracy. The prediction accuracy empowers the company to adjust to market dynamics quickly.

Cost savings: By optimizing inventory levels and minimizing the need for emergency restocking or clearance sales, ABC Retail saves millions of dollars annually in inventory holding costs, markdowns, and lost sales revenue.

Actionable insights: When anomalies are detected, ABC Retail's inventory management team receives real-time alerts and actionable insights. They investigate the root causes of anomalies, such as unexpected surges in demand or supply chain disruptions. Based on their findings, they adjust inventory levels, reorder points, and procurement strategies accordingly to mitigate potential risks and optimize resource allocation.

Additional benefits: By ensuring product availability and minimizing out-of-stock instances, ABC Retail obtains improved Customer Satisfaction, increased loyalty, and repeat purchases.

1.1.2 Case study 2: Electricity demand forecasting

Story: XYZ Power is a utility company responsible for generating and distributing electricity to residential, commercial, and industrial customers. Maintaining a reliable and stable power supply is critical to meet customer demand, avoid blackouts, and optimize operational costs.

Technology innovation: XYZ Power implements advanced time series forecasting techniques combined with anomaly detection algorithms to improve its electricity demand forecasting capabilities. The technology helps the company optimize generation schedules, manage grid stability, and avoid unnecessary expenses associated with overproduction or emergency measures.

Cost savings: By avoiding overproduction and emergency measures, XYZ Power saves millions of dollars annually in fuel costs, maintenance expenses, and penalties associated with capacity shortages or grid imbalances.

Actionable insights: The forecasted demand values are used to establish prediction intervals. Any observed demand data points falling outside these prediction intervals are flagged as anomalies. When anomalies are detected, XYZ Power's grid management team receives real-time alerts. They can investigate the root causes of anomalies, such as unexpected spikes or drops in demand, equipment failures, or network congestion. Based on their findings, they can adjust generation schedules or grid configurations to mitigate potential risks.

Since we have mentioned the use cases of anomaly detection, we shall also address the definitions for anomaly scores.

Sec 1.2 Anomaly score definitions

An anomaly refers to a significant deviation from the expected patterns within the data. These deviations can take various forms, such as sudden spikes, dips, or irregular behavior. An anomaly score quantifies how unusual or deviant a data point is compared to the expected behavior within the time series. The definition of anomaly score can vary based on specific business contexts or requirements. Here we offer the common definitions.

Prediction-interval-based Anomaly Score: A prediction interval estimates a range of values within upper and lower bounds where a certain percentage of data is likely to fall. For example, a 0.99 prediction interval means that 99 percent of the data is expected to lie within the interval. If a data point lies beyond the bounds of the prediction interval, it is considered anomalous. An anomaly score can be defined as 0.0 if the actual value is within the prediction interval, and otherwise 1.0 if it falls outside the interval.

Z-score-based Anomaly Score: This method calculates the anomaly score based on z-scores, which represent the number of standard deviations a data point is away from the mean of the time series. Data points with larger z-scores are considered more anomalous.

$$\text{Anomaly Score at time } t = \frac{(\text{Observed value at } t - \text{Mean of time series})}{\text{Standard deviation of time series}}.$$

Sec 1.3 Model interpretability

This book emphasizes the interpretability of a modeling technique. A model should be transparent and its prediction should be interpretable. A black-box model is less than ideal, as Kuo mentioned in *Explainable A.I.: with Python Examples*, “no one wants to swallow a black pill without knowing what’s in it.” An important factor is the **trust** in the model. Users will be willing to accept the model prediction as their guidance for actions. Transparency means the model parameters are present, and the model prediction can be decomposed. Users can even do an elementary-school type of addition like $A + B + C =$ the predicted value. Model interpretability often comes with visualization. Users can visualize the key factors and assess their relative significance. For example, a visual exhibit may show that a factor of a model is positively or negatively correlated to the prediction. Users of the model may approve or disapprove of such a relationship through the visual exhibits and gain their confidence in the model.

Sometimes model interpretability can refer to causality because people may want to use the discovered relationships in a model to infer causality, or to approve their hypotheses about the natural world. For example, if there is a spike in the sales volume due to a special promotional event, we may interpret that the high demand is due to the event and may predict demand spikes for similar promotional events in the future.

Sec 1.4 Local models vs. global models

In time series forecasting, a local model deals with a single time series or **univariate time series**. A global model is developed on multiple time series and the parameters are shared by all the time series. The techniques in this book include both local and global modeling techniques. Local models are useful when the patterns and trends in the data are specific to a particular time series, and there is little or no relationship between multiple time series. In this case, a local model can provide accurate forecasts without being influenced by the noise or variability in other time series.

On the other hand, global models are useful when there are common patterns across multiple time series, and the relationships between the time series are important for accurate forecasting. A global model can leverage the shared information across the time series to provide robust forecasts. For example, the sales volume over time of products within the same product category are multiple time series. A global model can be built for all products even when a new product is newly added to the same product category and has no history.

Sec 1.5 A roadmap to learn modern time series forecasting

An important goal of this book is to guide you through the rich landscape of time series forecasting techniques. First, knowing the development journey from classical to advanced techniques is as important as the proficiency of a particular technique. Second, we focus on probabilistic forecasting and multi-period forecasting. This is because many applications require both. Third, we highlight model interpretability for the advantages mentioned in the previous section. Fourth, we emphasize software competency.

We group the modern techniques into six parts to explore their evolution, intricacies, and transformative potential. **Part 1** starts with the Prophet technique because of its intuitive design and ease of use. Then it continues to NeuralProphet to assist you in building more complex models. In **Part 2**, we present four techniques to quantify uncertainty and deliver probabilistic forecasts. They include the Monte Carlo simulation, Quantile regression, Conformal prediction, and Conformalized quantile regression. These techniques are used in later techniques as well. **Part 3** uncovers the power of autoregressive-based techniques, shedding light on their ability to capture temporal dependencies and model dynamic relationships within sequential data. **Part 4** shifts focus to tree-based methodologies, showcasing their effectiveness in handling nonlinear patterns and successes in recent time series competitions. Venturing into the realm of deep learning, **Part 5** explores the depths of neural network architectures, revealing their capacity to learn complex patterns and extract latent features from high-dimensional data. Finally, in **Part 6**, we journey into the realm of transformer-based techniques. You will understand the architectures of the

Transformer-based models and their successes. Together, these six parts offer you a roadmap to learn modern forecasting techniques and apply them to your projects.

Chapter	Chapter content	Prophet/ Neuralprophet	Statmodels	pmdarima	Sktime	lightGBM	Darts	GluonTS
1	Introduction							
2	Prophet	o						
3	NeuralProphet	o						
4	NeuralProphet	o						
5	Change point detection							
6	Monte Carlo							
7	Quantile regression	o						
8	Conformal prediction	o						
9	Conformalized quantile prediction	o						
10	Automatic ARIMA		o	o			o	
11	Data formats	o		o	o		o	o
12	Linear regression						o	o
13	Tree-based Feature engineering					o		o
14	Primary strategies			o	o	o		
15	Tree-based						o	
16	Progression of techniques							
17	Deep learning							o
18	Deep learning							o
19	RNN to Transformer							
20	Temporal fusion transformer						o	
21	Lag-Llama							o

Figure 1.1: Time Series Python Libraries in this book

Sec 1.6 Time series Python libraries

The criteria for adopting the Python libraries in this book are **popularity** and **documentation support**. The time series Python libraries in this book are `statsmodels`, `pmdarima`, `sktime`, `Darts`, `GluonTS`, and `Autogluon`. All these libraries have active communities and documentation support. Figure 1.1 shows the Python libraries used in this book. Chapters 1, 16, and 19 are survey-type chapters and do not have code examples.

The `statsmodels` library contains all the classical tools and is a popular library for the statistics community. It has fundamental statistical tests, such as t-tests, F-tests, or chi-square tests. On model techniques, it has the most commonly used ordinary least squares (OLS) and generalized linear models (GLM). It includes the classical time series models such as autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA), vector autoregression (VAR), and state space models.

The `pmdarima` library extends the ARIMA modeling capabilities of the `statsmodels` library. It enables users to automatically select the optimal ARIMA and Seasonal ARIMA models. It provides the ability to perform multi-step forecasts. It also includes time-series functions to diagnose the fitted ARIMA models, such as residual analysis, ACF/PACF plots, and Ljung-Box tests.

The `sktime` library is another open-source library that combines many forecasting tools under a unified API. It wraps up a wide range of tools and algorithms for time series forecasting, classification and anomaly detection. Let's see its underlying modules. From `statsmodels`, it includes Holt-Winter's Exponential Smoothing, Theta Forecaster, and ETS. The ETS model is the decomposition model for trend (T), seasonal (S), and an error term (E). From `pmdarima`, it includes ARIMA and AutoARIMA. It also uses the `tbats` library. TBATS stands for Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, and Seasonal components. Finally, it also wraps the Prophet forecasters from the `Prophet` library.

The `Darts` Python library is a one-stop shop for time series analysis and forecasting. It provides a comprehensive set of tools and models for building, training, and evaluating time series forecasting models. The `Darts` library includes implementations of various time series forecasting models and is well-engineered for multi-period probabilistic forecasting. It includes popular forecasting techniques such as linear regression, ARIMA, XGB, GBM, CatBoost, RNN/LSTM/GRU, and Temporal Fusion Transformer (TFT).

The `GluonTS` Python library stands as a pioneering achievement from the Amazon Web Services (AWS) team. By leveraging MXNet and PyTorch, `GluonTS` empowers users to build complex models such as DeepAR, Transformer-based models, and TFTa. DeepAR is a deep-learning architecture for probabilistic forecasting. Chapter 17: *Deep Learning-based DeepAR for Time Series Probabilistic Forecasting*. `GluonTS` offers a wide array of techniques from traditional statistical methods to machine-learning algorithms. The “AutoGluon” library is a more general-purpose automatic machine learning (AutoML) framework. In other words, while `GluonTS` specializes in time series forecasting and probabilistic modeling, `AutoGluon` is a more general AutoML framework that automates the process of building machine learning models across various tasks, including time series analysis.

The above libraries are covered in Chapter 11: *Time Series Data Formats Made Easy*. All chapters except 1, 16, and 19 require Pandas and Matplotlib. These libraries continue to receive active developments. The above highlights are limited to the time I wrote this book.

Sec 1.7 Model evaluation metrics

This book uses several standard evaluation metrics. The choice of metric depends on the specific requirements of the forecasting task and the characteristics of the time series data. They include MAPE, MAE, MSE, sMAPE, RMSE, and Continuous Ranked Probability Score (CRPS). Since they will be used often throughout the book, we give the definitions here.

1.7.1 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) provides a measure of the accuracy of a forecasting model in terms of the percentage error relative to the actual values.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left(\frac{|F_i - A_i|}{A_i} \right) \times 100\%$$

Where:

- F_t is the forecasted value at time t ,
- A_t is the actual value at time t ,
- n is the number of observations.

1.7.2 Mean Squared Error (MSE)

Mean Squared Error (MSE) measures the average squared difference between the forecasted values and the actual values. It's a measure of the average squared deviation of the forecasts from the actual values.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (F_i - A_i)^2$$

MSE is particularly useful because it penalizes larger errors more heavily than smaller errors due to the squaring operation. This property makes it sensitive to outliers and large errors, which can be important in many applications. However, since MSE involves squaring the errors, it's not in the same unit as the original data, which can sometimes make interpretation less intuitive.

1.7.3 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average absolute difference between the forecasted values and the actual values. It provides a more intuitive measure of error compared to other metrics like MSE because it is in the same unit as the original data.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |F_i - A_i|$$

MAE is robust to outliers since it does not involve squaring the errors.

1.7.4 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) measures the square root of the average squared difference between the forecasted values and the actual values. It provides a measure of the typical magnitude of error, and since it's in the same unit as the original data, it's more interpretable than MSE.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

RMSE penalizes larger errors more heavily than smaller errors due to the squaring operation in MSE. Like MSE, RMSE is sensitive to outliers and large errors, making it a useful metric for assessing the overall accuracy of a predictive model.

1.7.5 Symmetric Mean Absolute Percentage Error (sMAPE)

Symmetric Mean Absolute Percentage Error (sMAPE) is a widely used metric for evaluating forecast accuracy. While it is similar to MAPE, it addresses some of the shortcomings of the traditional Mean Absolute Percentage Error (MAPE) by symmetrically handling both overestimations and underestimations. sMAPE is calculated using the following formula:

$$\text{sMAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \times 100\%$$

sMAPE is a symmetric measure, meaning it treats under- and over-forecasting equally. In contrast, MAPE can be influenced by the direction of the error, which can make it difficult to compare the performance of models that have different error distributions. sMAPE is less sensitive to extreme values than MAPE. This is because sMAPE scales the errors by the average of the actual and forecasted values, rather than just the actual value as in MAPE.

1.7.6 Continuous Ranked Probability Score (CRPS)

Continuous Ranked Probability Score (CRPS) is an evaluation metric when the forecasts are probabilistic forecasts rather than a point estimate. CRPS ranges from 0 to positive infinity. When the forecasted cumulative distribution function (CDF) perfectly matches the observed outcome, CRPS is zero. We want the CRPS to be as low as possible. The formula for the Continuous Ranked Probability Score (CRPS) is:

$$CRPS(F, y) = \int_{-\infty}^{\infty} [F(x) - h(x \geq y)]^2 dx \quad (1.1)$$

Let's break down the components:

- $F(x)$: This term represents the predicted cumulative distribution function (CDF) at each point x . The CDF gives the probability that the observed value will be less than or equal to x .
- $x \geq y$: This is an indicator function that evaluates to 1 if x is greater than or equal to the observed value y , and 0 otherwise. It essentially creates a step function that helps in comparing the predicted CDF with the observed value.
- $h()$: This is the Heaviside step function. It is 1.0 if $x \geq y$, otherwise 0. It defines whether each forecasted probability exceeds the observed outcome.
- $[F(x) - h(x \geq y)]$: This term calculates the difference between the predicted CDF and the indicator function. It essentially measures the distance between the predicted cumulative probabilities and the step function representing the observed value.
- $[F(x) - h(x \geq y)]^2$: Squaring the difference penalizes larger discrepancies more than smaller ones, ensuring that under- and over-predictions are treated equally.
- $\int_{-\infty}^{\infty}$: This represents the integral over the entire real line, summing up the squared differences across all possible values of x .

CRPS evaluates the alignment between predicted and observed outcomes by comparing their cumulative distribution functions (CDFs). This function measures the probability that the actual value falls below or equals a specific value. Here are the key aspects of CRPS:

- CRPS assesses the consistency of probabilistic forecasts, represented as CDFs, with observed outcomes. It computes the average deviation between predicted cumulative probabilities and actual results.
- It penalizes both under-dispersion (when predicted uncertainty is lower than observed) and over-dispersion (when predicted uncertainty exceeds observation). This dual penalty system enhances CRPS's robustness in evaluating forecast accuracy.
- Unlike metrics focusing solely on point estimates, CRPS evaluates the entirety of the forecast distribution. This characteristic renders it apt for assessing probabilistic forecasts, as it accounts for the entire probability distribution.

We will apply CRPS in Chapter 21: *Lag-Llama for time series forecasting*.

Sec 1.8 Cheat Sheets

This book has developed cheat sheets for quick reference. The cheat sheets summarize essential syntax examples from the chapters in the book. You will benefit from having specialized information at your fingertips. Whether you're a seasoned data scientist or just starting, having a well-curated cheat sheet can significantly enhance your productivity.

Sec 1.9 Download the example code files

The Python notebooks are available for download at

1 | <https://github.com/dataman-git/modern-time-series>

If there's an update to the code, it will be updated in the GitHub repository.

Sec 1.10 Data for this book

The datasets for this book cover energy, electricity, retail sales, and selected economic time series. The frequencies include hourly, daily, weekly, and monthly. All the datasets are available on Kaggle.com.

The **Website visitors** dataset [2] has the number of visitors to an academic teaching notes website from Kaggle.com. This dataset has daily counts of page loads, unique visitors, first-time visitors, and returning visitors. The dataset covers the date range from September 14, 2014, to August 19, 2020.

The **Bike-rental** dataset [3] has daily bike rental count records for 2011–2012. It includes other covariates like weather and seasonal information.

The **Hourly Electricity consumption** dataset [4] contains hourly electricity consumption and other covariates like temperature, dew point, heat index, and wind gust.

The **AEP hourly energy** dataset [5] is from the American Electric Power Company, Inc. It has hourly energy consumption data and other covariates.

The **Retail clothing** dataset [6] has monthly retail sales for clothing and clothing accessories.

The **Avocado** dataset [7] has monthly avocado sale volumes and price levels.

The **Store sales** dataset [8] has the sales for the thousands of product families sold at Favorita stores located in Ecuador. The columns include dates, store and product information, promotions, and other related fields.

References

1. Kuo, C. (2022). The eXplainable A.I.: With Python examples, <https://a.co/d/9I578JF>
2. Website vistors: <https://www.kaggle.com/datasets/bobnau/daily-website-visitors>
3. Bike rental: <https://www.kaggle.com/datasets/archit9406/bike-sharing>
4. Hourly electricity consumption: <https://www.kaggle.com/code/unajtheb/time-series-hourly-electricity-consumption-lstm/input>
5. AEP hourly energy: <https://www.kaggle.com/code/msripooja/hourly-energy-consumption-time-series-rnn-lstm/input>
6. Retail clothing: <https://www.kaggle.com/code/mrizwanse/time-series-forecasting-with-lstm-rnn/input>
7. Avocado: <https://www.kaggle.com/datasets/navleenbhatia/avocado-dataset>
8. Store sales: <https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data>



INDEX

- ACF, 113
Add and Norm, 239
Akaike Information Criterion (AIC), 115
anomaly, 3
AR-Net, 28, 46, 47
ARIMA, 108, 111, 163, 171, 188, 199, 204, 220
ARIMA(p,d,q), 111
attention mechanism, 220
Attention per horizon, 251
attention visualization, 251
attention weights, 251
Auto-correlation Function (ACF), 108
auto-regressive (AR), 111
Auto-Regressive Integration Moving-Average (ARIMA), 28
auto-regressive model, 154
Autogluon, 205
autoregressive (AR), 46, 68, 108, 190
AutoRegressive Integrated Moving Average, 111
AutoRegressive Integrated Moving Average (ARIMA), VII, 108

batch size, 252
bell-shaped curve, 79
breadth-first tree growth, 179
Business time series data, 13

calendar events, 153
CatBoost, 172
cell state, 197
Change point detection (CPD), 64
change points, 34
CKPT file, 262
classification, 175
confidence interval, 89
Conformal prediction, 4
Conformal Prediction (CP), 90, 98

Conformalized quantile regression, 4
Conformalized Quantile Regression (CQR), 98
Continuous Ranked Probability Score (CRPS), 8, 264, 265
cost function, 67
covariates, 56, 174
coverage of the upper bound and lower bound, 22
CPD, 64
cumulative distribution function (CDF), 8, 264
cyclical patterns, 152

daily, 29
Darts, 6, 125, 142, 149, 172, 174, 175, 181, 183, 242, 244
Decoder importance, 250
DeepAR, 6, 201–203, 209, 211
degrees of freedom, 79
depth-first tree growth, 179
differencing, 109, 111–113
direct forecasting strategy, 145, 163, 168
dropout, 252

e-commerce businesses, 201
electricity consumption, 152, 164
Encoder importance, 249
epoch, 253
estimates of uncertainty, 228
exponential moving average (EMA), 189
exponential smooth, 188
extreme events, 78
Extreme Gradient Boosting (XGB), 151

feed-forward neural network (FFNN), 250
few-shot learning (FSL), 255
forget gates, 197
Fourier series, 41

- Fourier terms, 40
 future covariates, 174, 176
 future features, 151
 future regressors, 29
 future values, 153
- GAM**, 29
 Gamma distribution, 209, 215
 Gated Recurrent Units (GRUs), 198
 Gated Residual Networks (GRNs), 235
 Gaussian distribution, 75, 192, 203, 204, 209
GBM, 171
 Generalized Additive Model (GAMs), 13
 global model, 4, 143, 202, 233, 241
GluonTS, 6, 130–133, 201, 204, 207, 209, 211, 215, 261
 Google Colab, 151, 164, 205
 Gradient Boosting Machine (GBM), 151
 Gradient boosting models, 157, 158
 gradient vanishing problem, 196
 gradient-boosting, 175, 179
 gradient-boosting models, VII, 151
 growth rate, 32
GRU, 198
- Hidden Markov Models (HMMs)**, 220
 hidden state, 222
 historical volatility, 77
 holidays and events, 28
Hugging Face, 259
 Hyper-parameters, 14, 15, 22, 116, 119, 158, 166, 245, 252
- in-time, 37, 131, 213, 261
 input chunk length, 146, 176, 245
 input embeddings, 224
 input gates, 197
 International Institute of Forecasters, 151
 International Journal of Forecasting, 151
 interpretability, 3, 249
 intra-attention, 224
- Kalman filter, 188, 190, 191, 199, 220
 key vectors, 224
- L1 and L2 regularization, 175, 239
Lag-Llama, VII, 220, 229, 230, 255, 257, 258
 Lag-Llama model, 229
 lagged features, 151
- lagged regressors, 29, 153
 large foundational models, 220, 228
 large language models (LLMs), 220, 255
 latent state, 192
 leaf-wise, 157
 leaf-wise tree growth, 179
 level-wise, 157
 level-wise tree growth, 179
LightGBM, VII, 151, 157, 164, 166–172
lightGBM, 181
 linear regression, 88, 89, 140, 171, 188
LLaMA, 229, 255, 256
 local model, 4, 143, 202, 233
 Long Short-Term Memory (LSTM), 47, 222, 236
 long-form, 130
 long-form , 125
 lower bound, 78
LSTM, VII, 188, 197, 198, 223, 236
- M competition, 151
MAPE, 159
 Markov Chain Monte Carlo (MCMC), 25
 Mean Absolute Error (MAE), 7, 20, 22
 mean absolute error (MAE), 145, 264
 Mean Absolute Percent Error (MAPE), 22
 Mean Absolute Percentage Error (MAPE), 6, 20
 mean absolute percentage error (MAPE), 145, 168, 264
 Mean Absolute Percentage Errors (MAPE), 159
- Mean Squared Error (MSE), 7, 20, 22
 mean squared error (MSE), 264
 memory, 194
MLP, 193
 model explainability, 151
 Model interpretability, 249
 model-agnostic, 90
 Monte Carlo simulation, 4, 75, 84, 192, 193, 203, 209
- monthly pattern, 19
 Moving average, 188
 moving average (MA), 108, 111
 moving averages, 156
 Multi-head, 223
 multi-head, 238
 multi-head attention, 251

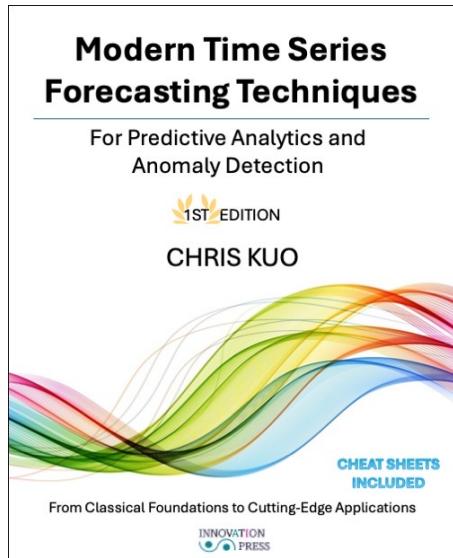
- multi-horizon forecasts, 228
- multi-period, 4, 145, 149, 163, 183, 201, 211, 212
 - multi-period forecasts, 163, 166, 168
 - multi-period?, 140
 - multi-series, 201, 211, 212
- Multilayer Perceptrons (MLPs), 193
- multilayer perceptrons (MLPs), 188
- multiple time series, 125
- multivariate time series, 125
- mvng average, 188
- Negative Binomial distribution, 209, 215
- neural network, 202
- neural-network, 135, 221, 222, 252
- NeuralProphet, 4, 28, 87, 135
- observed, 56
- offline data, 64
- one-step-ahead, 157
- ordinary least squares (OLS), 88, 89
- out-of-time, 37, 117, 131, 213, 261
- output chunk length, 146, 176, 245
- output gates, 197
- PACF, 113
- Pandas, 122, 123, 127, 130, 134–137
- Pandas-compatible, 165
- Partial Auto-correlation Function (PACF), 108
- past covariates, 144, 174
- past values, 153
- piecewise trend, 33
- Plotly, 32
- plotly-style, 135
- pmdarima, 5, 109, 134
- point estimate, 90
- Position-wise Feed-forward Network, 238
- Positional Encodings, 226
- prediction intervals, 3, 73
- prediction uncertainty, 202
- probabilistic distribution, 203
- probabilistic forecasting, 4
- Probabilistic forecasts, 73, 90
- Probabilistic prediction, 192
- Probabilistic predictions, 192
- probabilistic predictions, 192
- probability distributions, 209
- Prophet, 28, 135
- PyTorch, 29, 262
- quantile predictions, 141
- Quantile regression, 4, 84, 87, 89
- Quantile Regression (QR), 98
- quantile statistics, 78
- query vectors, 224, 225
- Random Forest (RF), 151
- real-time data stream, 64
- Real-world applications, 201
- Recurrent Neural Networks (RNNs), 47, 188
- recurrent neural networks (RNNs), VII, 194
- recursive forecasting strategy, 163
- regularization, 67
- reset gate, 198
- RMSNorm normalizing function, 256
- RNN, 188, 202, 221, 222, 236
- RNN/LSTM, 188, 202, 211, 220, 222, 230, 255
- RNN/LSTM/GRU, 220
- Root Mean Squared Error (RMSE), 22
- Root Mean Squared Error (RMSE)), 7
- Rotary Embeddings, 256
- SARIMA, 118
- seasonal ARIMA, 119
- seasonal-trend decomposition, 189, 199, 220
- seasonality, 13, 28, 32
- self-attention mechanism, 220, 224, 228, 237, 238, 250
- self-attention mechanisms, 249
- Seq2Seq, 223
- sequence-to-one, 194
- sequence-to-sequence, 194, 202, 220, 227
- sequential dependencies, 190
- Sequentially Discounting AutoRegressive (SDAR), 68
- SHAP, 249
- sigmoid function, 197
- simple moving average (SMA), 189, 220
- Sktime, 6, 134
- spread, 79
- State space model, 188, 191, 193, 199, 220
- static covariate encoders, 236
- Static covariate importance, 250
- Static enrichment, 237

- Stationarity, 111
stationary, 111
Statistical Machine Translation (SMT), 220
statsmodels, 5
stochastic gradient descent (SGD), 239
stock daily returns, 76
Student's t-distribution, 75, 209, 215, 257
summary statistics, 156
supervised learning, 108, 151, 157, 161, 167, 175
SwiGLU activation function, 256
Symmetric Mean Absolute Percentage Error (sMAPE), 8, 22, 168
temporal dependencies, 228
Temporal Fusion Transformer (TFT), VII, 6, 220, 227, 230, 232, 233
TFT, 228, 239, 249
time index, 153
time-related features, 153
trading strategies, 212
Transformer, VII, 220, 223
Transformer-based, 5, 220
tree-based, VII, 151
trend, 18, 28, 32
Uncertainty, 98
univariate time series, 4, 29, 152, 167, 188, 202
update gate, 198, 199
upper bound, 78
value vectors, 224, 225
vanishing gradient problem, 198
variability, 79
Variable Selection Networks (VSNs), 235
weekly, 29
wide-form, 126
XGB, 171
XGBoost, 172
yearly, 29
z-scores, 3
Zero-shot learning (ZSL), 255, 257



BOOKS BY CHRIS KUO

Modern Time Series Forecasting: For Predictive Analytics and Anomaly Detection 2024

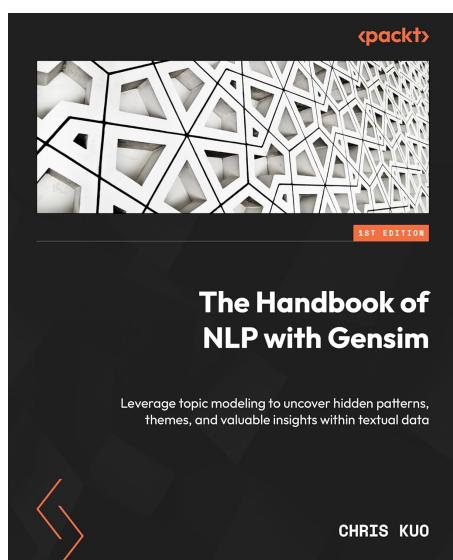


Key features

This book offers a comprehensive exploration of methodologies, algorithms, and practical applications, catering to both seasoned data scientists looking to refine their skills and newcomers eager to begin their analytical journey. Equipped with this resource, you'll confidently decipher intricate temporal patterns and harness predictive capabilities across various domains.

- Part 1: From Prophet to NeuralProphet
- Part 2: Getting probabilistic forecasts
- Part 3: Autoregressive-based Time Series Techniques
- Part 4: Tree-based Time Series Techniques
- Part 5: Deep into Deep learning-based Time Series Techniques
- Part 6: Transformer-based Time Series Techniques

The Handbook of NLP with Gensim: Leverage topic modeling to uncover hidden patterns. <https://a.co/d/ifO2LxR> 2023



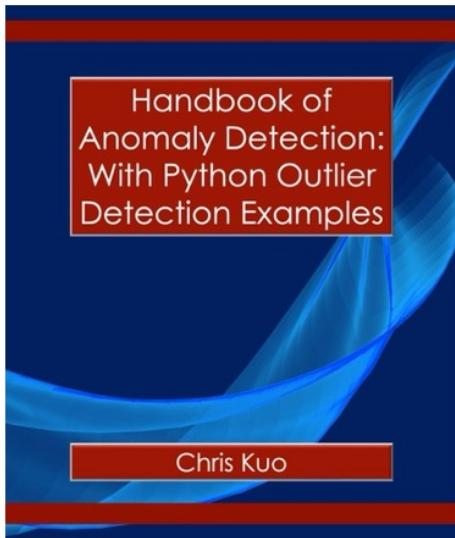
Key features

This book demystifies NLP and equips you with hands-on strategies spanning healthcare, e-commerce, finance, and more to enable you to leverage Gensim in real-world scenarios. You'll begin by exploring motives and techniques for extracting text information like bag-of-words, TF-IDF, and word embeddings. This book guides you on topic modeling using methods such as Latent Semantic Analysis (LSA) for dimensionality reduction and discovering latent semantic relationships in text data, Latent Dirichlet Allocation (LDA) for probabilistic topic modeling, and Ensemble LDA to enhance topic modeling stability and accuracy. By the end of this book, you'll have mastered the techniques essential to create applications with Gensim and integrate NLP into your business processes.



BOOKS BY CHRIS KUO

Handbook of Anomaly Detection: With Python Outlier Detection: Build and modernize your anomaly detection with examples 2023



<https://a.co/d/hHuHJEV>

Key features

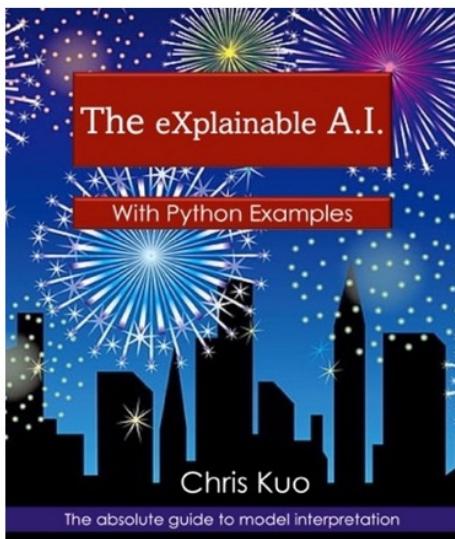
Learn a wide range of proven techniques that cover proximity-based, distribution-based, and ensemble-based algorithms. Enjoy systematic learning of modern techniques through appealing visualization of complex concepts. Become an experienced data scientist in outlier detection by hands-on code examples

Developer review

"Anomaly detection is a critical technique to identify rare items in risk modeling, security, and healthcare. Dr. Kuo's book provides hands-on guidance on how to use existing tools, such as PyOD, to leverage this technique in your daily data science work. More importantly, this book reviews more than 10 leading detection algorithms, with both algorithm descriptions and detailed code examples." ~ Yue Zhao, Principal Developer of PyOD

2022

The eXplainable A.I.: With Python examples <https://a.co/d/i8htOES>



Key features

- Learn algorithms with easy-to-understand examples.
- Learn what explainability A.I. is
- Learn SHAP to explain your machine learning models
- Learn SHAP with H2O Models. Learn SHAP Kernel Explainer
- Learn Microsoft's interpretML
- Learn how to use LIME to explain your models.

Who this book is for

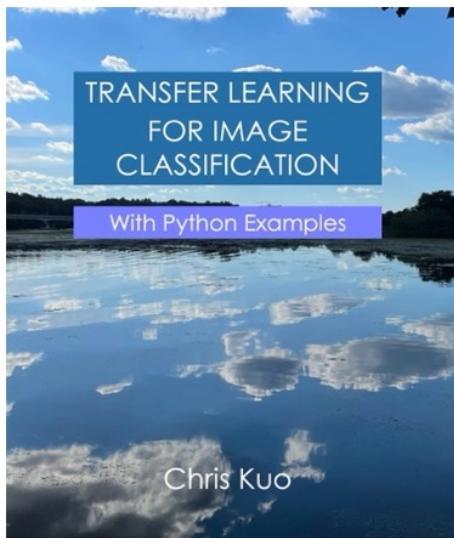
This book is for anyone who wants to understand model explainability and get your complex machine-learning models adopted. Although the content is extensive, data science professionals who do not have much machine learning background will find this book accessible



BOOKS BY CHRIS KUO

Transfer Learning for Image Classification: With Python examples. <https://a.co/d/e3MuX0e>

2022



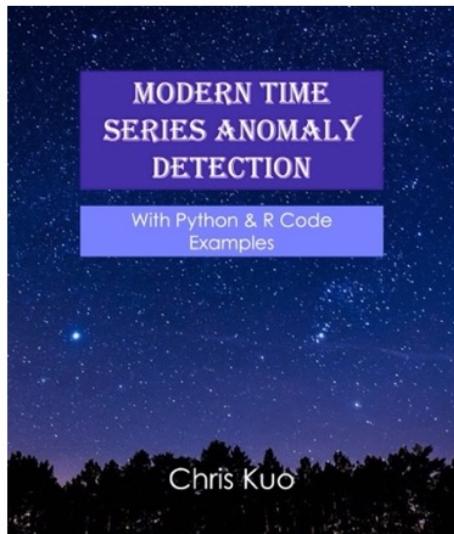
What you will learn

- Learn how deep learning models treat image data.
- Learn what a convolutional neural network (CNN) is.
- Learn each layer of a CNN by visualizing what it sees in an image layer-by-layer
- Learn the development of pre-trained image models
- Learn how to annotate images programmatically and pre-process images for modeling
- Follow Step 1,2,3 in the book to build deep learning models with Keras
- Build a repeatable pipeline to apply transfer learning for any image projects

Book Description

The best way of learning is to understand the motives of a technique and then practice with code examples. Thus, this book explains the reasons, describes a method and then shows the code examples. This book presents large-scale pre-trained image models and how to apply transfer learning techniques to build your models.

2021



Modern Time Series Anomaly Detection: With Python and R examples
(Out of stock)



Modern Time Series Forecasting Techniques For Predictive Analytics and Anomaly Detection

This book offers a comprehensive exploration of methodologies, algorithms, and practical applications, catering to both seasoned data scientists looking to refine their skills and newcomers eager to begin their analytical journey. Equipped with this resource, you'll confidently decipher intricate temporal patterns and harness predictive capabilities across various domains.

This book offers unique surveys from classical time series forecasting to state-of-the-art techniques:

- ❖ Part 1: From Prophet to NeuralProphet
- ❖ Part 2: Getting probabilistic forecasts
- ❖ Part 3: Autoregressive-based Time Series Techniques
- ❖ Part 4: Tree-based Time Series Techniques
- ❖ Part 5: Deep into Deep learning-based Time Series Techniques
- ❖ Part 6: Transformer-based Time Series Techniques

WHAT YOU GET IN THE BOOK

- Learning time series techniques comprehensively in a short period
- A roadmap from the classical techniques to modern time series forecasting
- Applying forecasting for resource planning and anomaly detection
- Mastering time series Python libraries
- Hands-on example code
- Cheat Sheets