

Final Technical Report (FTR)

COVER PAGE

a. Federal Agency	Department of Energy	
b. Award Number	DE-EE0009356	
c. Project Title	Graph-Learning-Assisted State and Event Tracking for Solar-Penetrated Power Grids with Heterogeneous Data Sources	
d. Recipient Organization	Northeastern University	
e. Project Period ¹	Start: June 1, 2021	End: May 31, 2024
f. Budget Period	Start: June 1, 2023	End: Dec. 31, 2024
g. Reporting Period	Start: June 01, 2024	End: August 31, 2024
h. Report Term or Frequency	Quarterly	
i. Principal Investigator (PI)	<p>Name: Ali Abur Title: Professor Email address: abur@ece.neu.edu Phone Number: 617-373-3051</p>	
j. Business Contact (BC)	<p>Name: Dan Dapkas Title: Grant and Agreement Officer Email address: d.dapkas@northeastern.edu Phone number 617-373-5154</p>	
k. Certifying Official (if different from the PI or BC)	N/A	



Signature of Certifying Official

February 20, 2025

Date

By signing this report, I certify to the best of my knowledge and belief that the report is true, complete, and accurate. I am aware that any false, fictitious, or fraudulent information, misrepresentations, half-truths, or the omission of any material fact, may subject me to criminal, civil or administrative penalties for fraud, false statements, false claims or otherwise. (U.S. Code Title 18, Section 1001, Section 287 and Title 31, Sections 3729-3730). I further understand and agree that the information contained in this report are material to Federal agency's funding decisions and I have any ongoing responsibility to promptly update the report within the time frames stated in the terms and conditions of the above referenced Award, to ensure that my responses remain accurate and complete.¹

¹If you have received No Cost Time Extensions (NCTE), please add a note below the table indicating the length of each one and which budget periods were affected.

ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) Solar Energy Technologies Office (SETO) under the DE-FOA 0002243 Award Number: DE-EE 0009356.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

EXECUTIVE SUMMARY

Major Goals and Objectives

Provide a one-page summary of the motivation, importance, goals, and major accomplishments of the project. The summary should include: (1) how the research or project adds to the understanding of the area investigated; (2) the technical effectiveness and economic feasibility of the methods or techniques investigated or demonstrated; and (3) how the project is beneficial to the public.

The main objective of this project is to make the distribution systems fully observable, such that the hosting capacity for solar generation can be accurately estimated, and unnecessary solar curtailments can be avoided. The expected outcome of the project is a prototype software with two main parts: 1) a grid-model-informed machine learning (ML) tool which integrates heterogeneous data streams and creates synchronous measurement snapshots for the state estimator (SE); and a hybrid robust SE which provides not only accurate state estimates but also real-time feedback for ML model refinement. The project also develops an event/topology tracker that detects and locates switching events and faults, providing up-to-date grid models to the ML and SE algorithms.

Technical Effectiveness and Feasibility

Impact

State estimation is a crucial function in monitoring the system state to efficiently dispatch generation, avoid unnecessary solar curtailment, optimize storage management, and control voltages. Currently, most distribution systems do not use SEs due to the lack of high-resolution and time synchronized measurements. This project addresses this barrier via the following innovations over the state of the art: 1) seamless integration and synchronization of heterogeneous measurement assets with a variety of scan rates and accuracy classes; 2) ML techniques supervised by feedback from the SE and topology tracker; 3) distributed implementation enabling scalability to very large utility systems; 4) agile event identification across the system based on sparse real-time measurements.

This project addresses the current deficiency of visibility in distribution systems, and thus facilitates improved efficiency, reliability, implementation of innovative energy trading functions incorporating prosumers, flexible microgrid operation, and management of new types of loads such as electric vehicles. Without requiring deployment of new sensors, it will exploit the potential in already existing heterogeneous measurement assets. It has potential to be commercially implemented and used by future distribution system operators to improve efficiency and reliability of power grid operation.

TABLE OF CONTENTS

Contents

1	BACKGROUND	6
2	PROJECT OBJECTIVE	8
3	PROJECT RESULTS AND DISCUSSION	9
4	SIGNIFICANT ACCOMPLISHMENTS AND CONCLUSION	119
5	PATH FORWARD	123
6	PRODUCTS	124
7	PROJECT TEAM AND ROLES	125

I BACKGROUND

Provide up-to-date references to peer-reviewed journal publications, conference proceedings, and other technical publications on the latest developments in similar technology fields as the work undertaken in the project. Include a technical discussion that shows how the published work is: 1) similar to this project; 2) different from this project; and 3) being fully leveraged to accelerate the timeline and improve the quality of the current project. This should place the project in living context alongside the current state of the art in the literature and ongoing R&D.

With the proliferation of distributed energy resources (DERs), the operating conditions of distribution systems are becoming more uncertain and volatile, and the dispatch, control, and protection applications require timely and accurate monitoring. This is accomplished via state estimation (SE) which utilizes measurements acquired from various points in the system and determines the best estimate of the system state. In the absence of sufficient measurements, state estimation cannot be carried out due to the lack of network observability. A major challenge in power distribution system monitoring is the diversity of the reporting rates of measurements from phasor measurement units (PMUs), supervisory control and data acquisition (SCADA) systems, and advanced metering infrastructure (AMI). In particular, sensors with slow reporting rates, such as smart meters, cannot keep up with the execution rate of state estimation, leaving gaps of observability between the arrival of two measurement samples. A comprehensive review of the recent work recognizing and attempting to address this challenge can be found in [1]. A data-driven technique is employed to predict the consumption patterns of customers without smart meters to enhance the observability of distribution systems in [2]. The authors of [3] propose using historical low-voltage side smart meters to forecast load and DERs injections via the support vector machine with optimally tuned parameters. In [4], an optimal measurement placement is proposed to obtain the accurate pseudo measurements even with limited knowledge of the profile of the injected power. A popular category of approaches is the use of machine learning (ML) techniques to predict measurements with slow reporting rates to enhance their time resolution and meet state estimation requirements [5]. There are two main limitations of the existing work. First, most of these methods completely disregard the power network models or simply learn the mapping with the underlying assumption of fixed network topology. Hence, they fail to produce fully satisfactory results under network model changes which are frequent in feeder operation. Second, most of them do not receive proper supervision from power system domain knowledge. As a result, the predicted measurements by the ML models are not compliant with the power flow model, reducing their accuracy and reliability.

Another major challenge in distribution system monitoring is the timely detection and identification of events such as faults, line switching, and substantial DER/load switching. In recent years, several interesting alternative solutions have been proposed, mainly focusing on detecting those events [6, 7, 8, 9]. These methods are well-established, with detailed examples illustrating their application. Many rely on graph search or brute force to identify power grid events, inherently complicating their implementation or limiting their scalability. Therefore, in this project a sparse estimation approach is based on the Least Absolute Shrinkage and Selection Operator (LASSO) is employed. Event detection (ED) and SE are two closely related problems in power system monitoring. For example, references [10, 11] establish a generalized SE framework based on the minimum-information-loss (MIL) theory, unifying analog and digital quantities in the estimation of system states and switch status and proving that the weighted-least-square (WLS) and weighted-

least absolute-value (WLAV) estimators are special cases under the MIL principle. In distribution systems, a major challenge in event detection is the sparsity of time-synchronized measurements. To this end, a sparse estimation-based approach is developed in this project to detect such events using a few synchronized measurements.

This project introduces a comprehensive paradigm for distribution systems SE and ED using sparse and heterogeneous measurements. The main objective is to make the states and events in distribution systems fully observable. This is accomplished by developing a grid-model-informed machine learning (ML) tool that uses several different types of measurement data to create a consistent and reliable set of synchronous measurement snapshots for a robust SE. The differentiating features of the proposed framework include the use of a graph neural network to capture topology information, and the feedback of the robust SE to enhance the ML model's prediction accuracy and compliance with the physical power flow model. The project also features a highly efficient implementation of the SE in large systems, as well as the detection and location of events based on sparse PMU measurements. The proposed approach is validated using data and measurements from a utility Microgrid.

II PROJECT OBJECTIVE

Describe the impact this project will have on achieving national goals of clean energy progression and economic benefit to the U.S. Describe the project goals, expected outcomes, and significance, innovation, and fundamental advances of the proposed body of work and how they support the claimed impacts. Please include a summary of the tasks within the Statement of Project Objectives (SOPO) or Technical Work Plan (TWP) for the entire project, including the milestones and go/no-go decision points.

Integration of renewable energy sources into nation's power grid is a timely and critical challenge. Given that a majority of these sources are connected to the grid in the distribution system along feeders, it is crucial to establish monitoring capability for the distribution systems in order to control and operate these sources in an optimal fashion. Since most distribution systems are historically not well monitored, their measurement systems do not render these systems fully observable. The project's main goal is to address this lack of observability by using machine learning methods to incorporate measurements from AMI meters which have slow scan rates not compatible with state estimators and develop an efficient and reliable approach to generate pseudo-measurements that can render the entire distribution system fully observable. These measurements are further improved by using a robust state estimator which provides feedback to the machine learning measurement predictor so that its training can be further improved. The developed approach also includes a topology change detector which also provides input to the measurement predictor further improving its reliability and accuracy. Overall, the project's results are expected to be quite significant by enabling monitoring of distribution systems which were otherwise not observable by conventional existing measurements. The impact will be also seen on the overall power grid operation since it will allow control and incorporation of renewable energy sources to increase their contribution to overall generation dispatch, avoid unnecessary curtailment of clean energy, management of congestion, etc. Detailed description of the completed tasks and associated milestones of the entire project are given in the next section.

III PROJECT RESULTS AND DISCUSSION

Provide a high-level, quantitative comparison of anticipated project outcomes against realized results with clearly stated quality metrics to assess the confidence of the results. A clear sense of progress against award milestones, both throughout and at award end, should be conveyed. State the project tasks and go/no-go milestones/deliverables and metrics and compare them to what was actually achieved. This section's structure should be based on the SOPO, making comparisons at the task level with the subtasks providing support for claimed progress. Milestone rows should be copied verbatim to appropriate points in the technical discussion. Enough detail and/or references to supporting documentation must be provided to make it clear that milestones were successfully accomplished. If milestones were not met, discuss any extenuating circumstances and difficulties encountered. The methodologies (e.g., modeling approaches, experimental methods) utilized to obtain the results should also be included. Relevant figures and data tables should be included and discussed in enough detail to demonstrate the technical progress made on the award.

Subtask 3.1: Robust WLAV Estimator for Measurements with Different Scan Rates

Subtask Summary: In this subtask, a robust WLAV estimator is developed. The predicted measurements form a virtual synchronous measurement snapshot together with real-time measurements. With the help of such information redundancy, the data interpolation property of WLAV estimator enables automatic selection of a minimum number of measurements which will meet the observability requirements while being most probable to be accurate. The proof of concept will be performed by using synthetic data in PPM. Based on the load profiles provided by the project partner, power flow analysis is performed, based on which true values of measurement can be generated. Subsequently, noise and gross errors of various scales will be introduced into the synthetic measurements to mimic different qualities of different types of measurements. The accuracy and bad data rejection capabilities of the WLAV estimator can then be tested.

The subtask is conducted as planned. A robust WLAV estimator is developed, and tested in IEEE 13, 14 bus systems as well as in the network provided by the project partner. The structure of the PPM is slightly modified such that a few virtual lines are added to merge two islands, and to avoid strictly radial structure.

The simulation results of estimator validates that LAV approach enables automatic selection of the minimum number of measurements which will meet the observability requirements while being most probable to be accurate. As defined in Use-case 1.3.1, estimator is tested under diverse measurement set configurations. The true measurement values are obtained by three-phase power flow analysis for both IEEE system and PPM network. The accuracy class of different measurement devices are modeled by introducing noise signals with diverse variance values. Moreover, the rejection capability of proposed estimator against bad data is tested using various measurement sets. Single or multiple bad data are intentionally injected into the measurement set. The location and number of bad data are defined in Use-case 1.3.2.

The results of the simulations reveal that the conditions stated in Milestone 1.3.1 and Milestone 1.3.2 are met successfully. Please see the details given further down in this section.

Mathematical model of LAV-based SE: The system states for the estimation are three-phase bus voltage magnitudes and bus voltage phase angles. The measurements are three-phase bus voltage magnitudes, real/reactive power flow measurements, real/reactive power injection measurements, phase angle measurements, real/imaginary current flow measurements. The objective function of the LAV based estimation is given in (1).

$$\min \sum_{i=1}^m |r_i| \quad (1)$$

$$\text{subject to } z_i = h_i(x) + r_i \quad (2)$$

where, r_i is the residual of i^{th} measurement corresponding to the difference between the real value of the measurement and the estimated value of the measurement, i.e., $r_i = z_i - h(\hat{x}_i)$ and m is the number of measurements. Using the first-order approximation of $h_i(x^0)$ around of x^0 , the problem can be rewritten as linear programming (LP) problem as shown in (3).

$$\min c^T \cdot Y \quad (3)$$

$$A \cdot Y = b \quad (4)$$

$$Y \geq 0$$

where;

- $A = [H, -H, I, -I],$
- $c^T = [0_n, 0_n, 1_m, 1_m],$
- $b = \Delta z,$
- $0_n :$ a zero vector of order $n,$
- $1_m :$ a vector of order $m,$ where all the entries are 1,
- $H :$ the Jacobian matrix.

The structure of the measurement vector, $z,$ is given in (5) while full measurement redundancy of AMI, SCADA and PMU measurements are available. Note that the number of measurements are three times of the number buses and branches since three-phase model is utilized.

$$z^T = [\underbrace{V_{mag}, P_{flow}, Q_{flow}, P_{inj}, Q_{inj}}_{(SCADA/AMI)}, \underbrace{V_{mag}, \theta, I_{flow,real}, I_{flow,imag}}_{(PMU)}] \quad (5)$$

- $V_{mag} :$ magnitude of bus voltages in p.u., $(3n \times 1),$
- $P_{flow} :$ real power flow measurement in p.u., $(3m \times 1),$
- $Q_{flow} :$ reactive power flow measurement in p.u., $(3m \times 1),$
- $P_{inj} :$ real power injection measurement in p.u., $(3n \times 1),$
- $Q_{inj} :$ reactive power injection measurement in p.u., $(3n \times 1),$
- $\theta :$ phase angle of bus voltages in radian, $(3n \times 1),$
- $I_{flow,real} :$ real part of the current flow measurement in p.u., $(3m \times 1),$
- $I_{flow,imag} :$ imaginary part of the current flow measurement in p.u., $(3m \times 1),$
- $n :$ number of buses,
- $m :$ number of branches,

The accuracy and bad data rejection capability performance of the SE is further improved using a strategic weighting of the measurements. The weights of the measurements are adjusted using the projection statistics method for power systems proposed in [12].

Test Results for PPM network: The implemented LAV SE is tested in the modified microgrid network provided by the project partner. The load data is also provided by project partner. Using the topology and load data, three-phase power flow analysis is conducted to obtain true bus voltages and true measurement values. For the PPM network tests, SCADA, AMI and PMU measurements are assumed to be available. The accuracy of the different measurement devices are given in Table 1. To model different accuracy classes of measurements, random Gaussian noise signal is introduced to measurement set. A sample noise signal is given in Fig. 1 where mea-

surements 1 – 381 correspond to SCADA/AMI V_{mag} measurement type which has 0.01 variance value, measurements 382 – 784 correspond to P_{flow} measurement type which has 0.02 variance, etc. All measurement types, and their corresponding measurement numbers and variance values are given in Table 1 and Table 2. There are nine different measurement types employed in the synthesized measurement set, and they are ordered in the horizontal scale of Fig. 1 as follows; V_{mag} , P_{flow} , Q_{flow} , P_{inj} , Q_{inj} , V_{mag} , θ , $I_{flow,real}$, $I_{flow,imag}$, respectively.

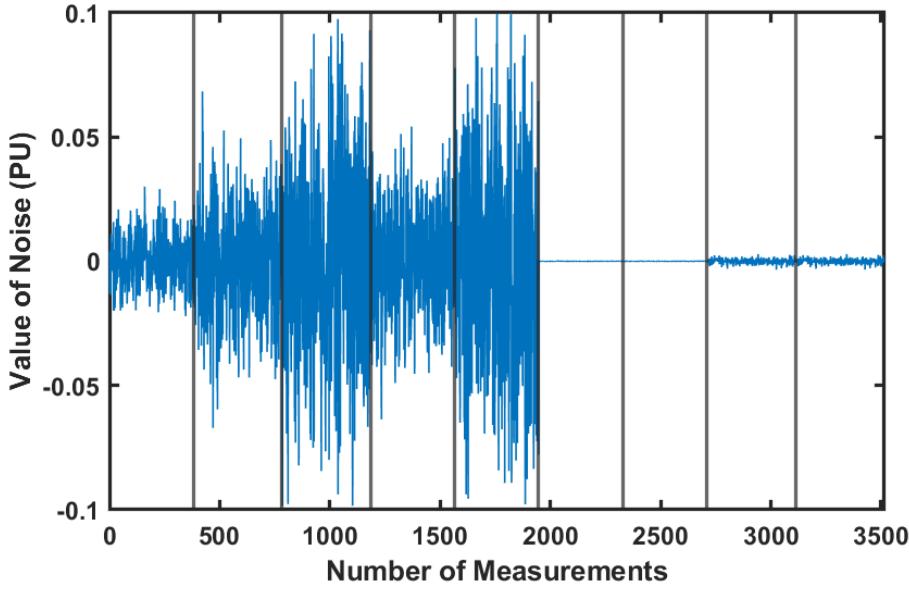


Figure 1: Randomly generated noise signal to be introduced to measurements.

Table 1: Standard deviation for different measurements.

Measurement Type		Standard Deviation, σ (PU)
SCADA/AMI	Voltage Magnitude	0.01
	Real Power Flow	0.02
	Reactive Power Flow	0.04
	Real Power Injection	0.02
	Reactive Power Injection	0.04
PMU	Voltage Magnitude	0.0001
	Voltage Phase Angle	0.006
	Real Current Flow	0.001
	Reactive Current Flow	0.001

The states of estimator are conventional states, i.e., bus voltage magnitudes and bus voltage phase angles. Thus, the relation between measured quantities and states is non-linear which yields an iterative solution process. In the PPM, 127 buses and 134 branches yields total of 762 system

Table 2: Number of measurements utilized in the three-phase PPM network.

Measurement Type		Number of Measurements	
SCADA/AMI	Voltage Magnitude	381	SCADA/AMI Total : 1947
	Real Power Flow	402	
	Reactive Power Flow	402	
	Real Power Injection	381	
	Reactive Power Injection	381	
PMU	Voltage Magnitude	381	PMU Total : 1566
	Voltage Phase Angle	381	
	Real Current Flow	402	
	Reactive Current Flow	402	

states in three-phase model. Table 3 shows the accuracy of the LAV and WLS based SE. The accuracy of the WLS SE is slightly better than LAV SE using measurement set with noise considering the objective function of WLS is to minimize mean squared error. However, once single or multiple bad data is injected, LAV based estimation is significantly superior. The automatic rejection of bad data in LAV approach makes the estimation robust against bad data while WLS based estimator is biased drastically.

Table 3: Accuracy of LAV and WLS estimators in PPM network under various measurement set configurations.

	LAV		WLS	
	RMSE	MAPE	RMSE	MAPE
True Measurement Set	2.68×10^{-13}	4.07×10^{-13}	2.84×10^{-16}	7.43×10^{-16}
Measurement Set with Noise	4.09×10^{-5}	2.03×10^{-4}	4.53×10^{-6}	2.2×10^{-5}
Meas. Set with Single Bad Data	4.09×10^{-5}	2.03×10^{-4}	0.0096	0.1058
Meas. Set with Multiple Bad Data	3.94×10^{-5}	2.04×10^{-4}	0.0177	0.1362

Table 4: Accuracy of LAV SE in PPM network using only PMU and only SCADA measurements .

	LAV	
	RMSE	MAPE
Only PMU Measurements	5.72×10^{-6}	4.11×10^{-4}
Only SCADA Measurements	7.24×10^{-4}	0.0014

Instead of full measurement set configuration, only PMU measurements or only SCADA/AMI measurements are utilized in the estimator to test the accuracy of SE under different class of accuracy measurements. As shown in Table 4, the accuracy of SE by only PMU measurements is

improved compared to SE with only SCADA/AMI. Furthermore, Fig. 2 shows the improvement in the RMSE as the number of PMU devices utilized in the estimation increases.

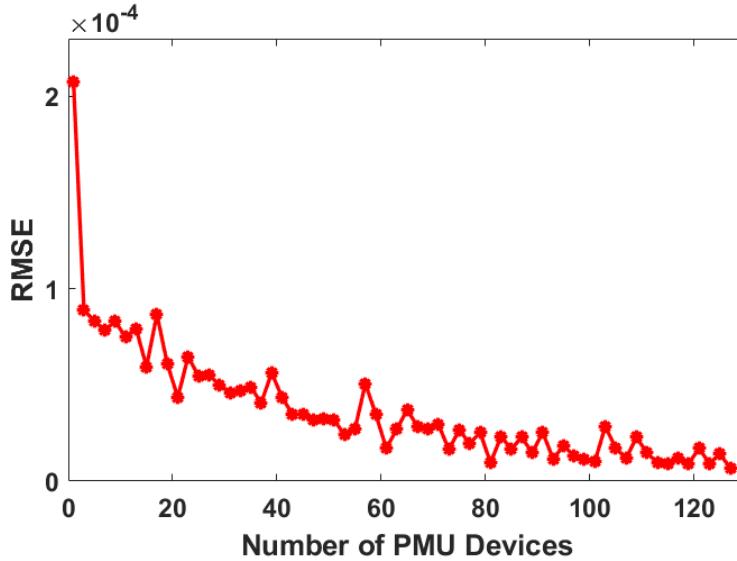


Figure 2: The RMSE of voltage estimates with varying number of utilized PMU devices.

Milestone 1.3.1 requires RMSE of the voltage estimates to be less than the median percentage error, α , in the synthesized measurements. The α value is median of the measurement standard deviations since the error in the synthesized measurements is introduced according to the standard deviation of each measurement type. The α value is obtained as 1×10^{-2} with the current measurement configuration, whereas the RMSE value of the estimator using full measurement set without bad data is 4.09×10^{-5} . Thus, milestone is achieved where, RMSE is less than α value.

Bad data rejection capability of the LAV based SE is tested in the cases defined in Use-case 1.3.2. In the Fig. 3, the accuracy of LAV and WLS based estimations are visualized by plotting the biased measurements and measurement residuals. Randomly generated 50 bad data are introduced to random measurement locations. While yellow lines show the magnitude of bad data, i.e., the difference between the actual and true measurements, the residuals of LAV estimation is not biased by the erroneous measurements as shown in Fig. 3a. On the other hand, WLS based SE reveals highly biased residuals as shown in Fig. 3b. The overall results regarding the bad data rejection capabilities of the estimators are given in Table 5. As shown in table, while LAV is robust against up to 200 random bad data inserted to PMU or SCADA/AMI or whole measurement set, WLS is vulnerable in the most of the cases. The values highlighted in red in the table show the RMSE values greater than the pre-defined α value which makes the estimation unsuccessful.

The RMSE values shown in Table 5 are calculated using the average of 100 Monte Carlo simulations. To calculate bad data rejection percentage, again 100 Monte Carlo simulations are conducted. The rejection percentage is $(n_{\text{successful}} / n_{\text{total}}) \times 100$ where $n_{\text{successful}}$ is the number of successful simulations, i.e., RMSE in those simulations are less than α value, and n_{total} is the number of total simulations. As seen in the Table 6, LAV based SE rejects up to 200 bad data with 100% success. On the other hand, WLS SE poorly rejects bad data.

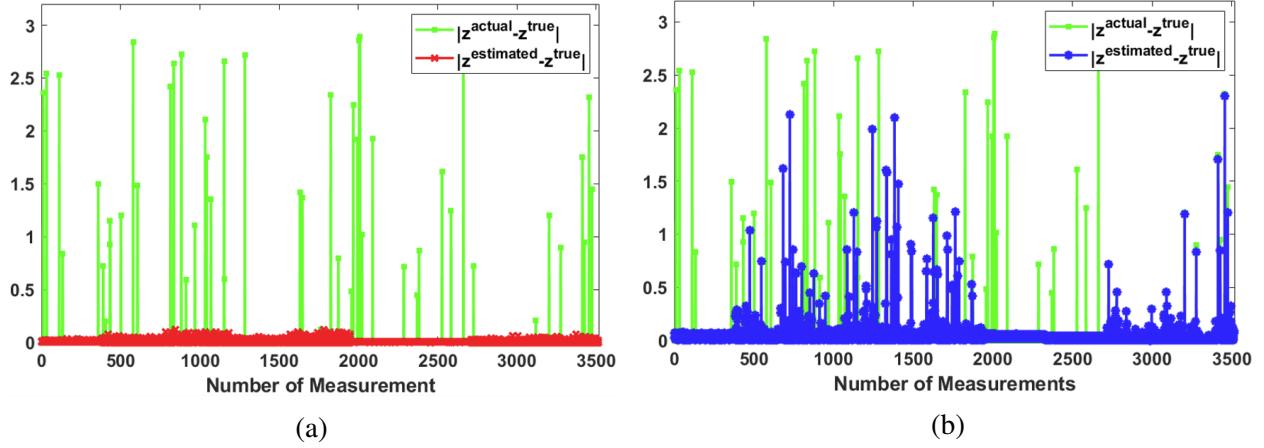


Figure 3: Error in the measurement residuals when 50 randomly generated and located bad data are introduced to (a) LAV SE, (b) WLS SE.

Table 5: Accuracy of LAV and WLS SE in PPM network under diverse bad data injection scenarios.

Location of Bad Data	Number of Bad Data	RMSE by LAV ($\times 10^{-5}$)	RMSE by WLS
SCADA/AMI	1	4.06	7.18×10^{-5}
	50	4.09	4.82×10^{-4}
	200	4.78	0.0016
PMU	1	4.07	0.0039
	50	4.00	0.0679
	200	4.49	0.0720
Whole System	1	4.05	8.44×10^{-4}
	50	4.32	0.0225
	200	4.92	0.0343

Table 6: Percentage of bad data rejection for LAV and WLS based SE.

Number of bad data	LAV	WLS
1	100%	76%
50	100%	11%
200	100%	0%

Milestone 1.3.2 requires the percentage of bad data rejection to be greater than 90%. As shown in Table 6, LAV based SE rejects 1, 50, 200 randomly generated and located bad data with 100% success. Hence, milestone 1.3.2 has been successfully achieved.

Subtask 8.1: Massively Parallel Distributed Implementation of Robust WLAV Estimator

Subtask Summary: In this subtask, we extend the work on the WLAV estimator developed in Subtask 3.1 in order to address the scalability issue, i.e. facilitating the utilization of the estimator for very large scale power grids. To this end, we propose an innovative multiple-area multiple-copy solution to fully resolving the scalability challenge by leveraging distributed computing technologies. This algorithm creates several copies of the system model where each copy is assigned a different and strategically designed area partitioning. The approach requires creation of multiple copies of the power grid followed by strategic partitioning of each copy in such a way that every bus appears as an “internal” bus in at least one copy. This will ensure robustness of the solution against bad data due to the automatic bad data rejection feature of the WLAV estimator for internal measurements.

This subtask is conducted as planned. A Massively Parallel Distributed (MPD) SE algorithm is implemented, validated, and tested in the IEEE 14 bus system, PPM network as well as a very large-scale network [13]. The proof of concept study of the proposed method is presented in the quarterly report 1 of the budget year 2. The formulation of the proposed MPD method is verified with a comprehensive bad data analysis using IEEE 14 bus system. Moreover, a strategic automated partitioning algorithm is presented in the quarterly report 2 of the budget year 2. The partitioning algorithm is developed to optimize MPD SE performance considering multiple constraints. To maximize the performance of SE, two different partitioning algorithms for radial and meshed structured systems are implemented.

The concept of the MPD SE is explained using flow chart, and followed by the formulation of the MPD framework. Afterwards, novel partitioning algorithms for both meshed and radial systems will be explained. Lastly, simulation results for bad data performance in IEEE 14 bus system, and simulation results for scalability performance in the PPM network and very large sized (\sim 12 thousand buses) network will be presented.

Tests are conducted to experimentally validate that **Milestone 2.8.1** is successfully achieved using the proposed MPD SE using the developed partitioning algorithm. In the results section, it will be shown that the computational time of the MPD SE remains insensitive to the system size assuming that there are enough processor cores. To reduce the computation time for the entire system, one can increase the number of zones by proper system partitioning.

Massively Parallel Distributed State Estimation (MPD SE) : The primary goal of the parallel distributed implementation of the robust LAV estimator is to make the computation time of the SE almost independent of the system size. This can be accomplished by first partitioning the network into multiple zones where the solution of each zone will be executed separately. However, the partitioning process inadvertently leaves out certain measurements incident to boundary buses leading to reduced redundancy at the zone boundaries which may affect the robustness of the estimator. While a boundary bus may be connected to one or more buses of other zones, an internal bus is connected to only those buses in the same zone. Unlike the measurements incident to boundary buses, measurements incident to internal buses will not be affected by system partitioning. Thus, multiple copies of the system will be generated where it is ensured that each bus is an internal bus in at least one copy of the system.

The flowchart of the overall MPD SE algorithm is given in Fig. 4. As shown in the flowchart,

different partitioning algorithms are employed depending on the system structure. Once the original copy of the system is partitioned into n_0 zones using either Algorithm I or II, additional system copies are generated using Algorithm III. The reasoning for using two different partitioning algorithms and details of the algorithms are given in the following sections.

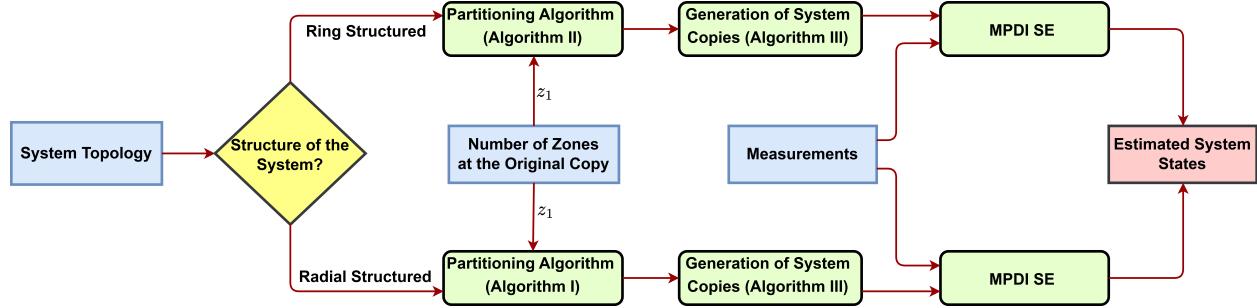


Figure 4: Flowchart of the MPD SE procedure

It is critical to develop an optimal zone generation algorithm so that a minimum number of system copies will be used without any loss of information. Moreover, since the computation time for the largest zone will determine the overall computation time, evenly sized zones will have to be generated in each system copy. In the original (first) copy of the system, the system is partitioned into multiple zones; however, only one strategic zone is created in each of the remaining copies to minimize the computational burden. In other words, the buses which are located outside of the strategic zone in the additional copies are disregarded to avoid unnecessary computations. The details of Algorithms I, II, and III are given in the following sections. Total number of independent SE runs will be the total number of zones in all copies. The total number of zones, n_{zones} , is determined by the summation of the number of zones in the original copy and the number of additional copies of the system as given below:

$$n_{zones} = (1 \times n_0) + ((n_{copies} - 1) \times 1) \quad (6)$$

where n_{copies} is the total number of copies, and n_0 is the number of zones in the original (first) copy.

Partitioning Algorithm for Fully Radial Structured Systems (Algorithm I):

The pseudo-code of the developed partitioning algorithm is given in Algorithm 1. The algorithm finds the branch whose removal minimizes the size difference of two islands.

Algorithm 1 Fully Radial Partitioning(N, n_0)

Require: N : network topology, n_0 : number of zones at the original copy of the system,
Ensure: N_k^1 and N_k^2 are the two islanded sub-networks of N_k where k is the number of sub-networks in N_k .

```

1:  $n_z = 1$                                      ▷ Set number of zones to 1
2: while  $n_z \leq n_0$  do
3:   for  $j \in k$  do
4:      $branch - m = \min_{m \notin N_j} (size(N_j^1) - size(N_j^2))$     ▷ Find branch-m whose removal minimizes the size
       difference of two sub-networks
5:     remove  $branch - m$ 
6:   end for
7:    $n_z = n_z \times 2$ 
8: end while
9: return  $N_{n_z}$ 
```

Partitioning Algorithm for Non-radial (Meshed) Systems (Algorithm II): To use the graph-based partitioning algorithm, a graph is constructed using MATLAB's built-in functions and system topology. The coordinates of the buses are obtained using the constructed graph in MATLAB environment as shown in Algorithm 2.

Algorithm 2 Ring Partitioning (N, n_0)

Require: N : network topology, n_0 : number of zones at the original copy of the system,

Ensure: N_k is the network with k -many sub-networks

```

1: Construct graph using  $N$ 
2: Get coordinates of buses from graph
3: Execute graph-based-partitioning for  $n_0$  zones
4: return  $N_{n_0}$ 
```

Generation Algorithm for Additional System Copies (Algorithm III): Once the original copy of the system is partitioned into zones, loss of certain measurements incident to boundary buses leads to reduced redundancy at the zone boundaries which may affect the robustness of the estimator. The generation of additional system copies aims to ensure that each bus is an internal bus in at least one copy of the system. Instead of partitioning the additional copies into multiple zones, it is aimed to create only one strategic zone in each additional system copy. The algorithm for the generation of zones is given in Algorithm 3.

Algorithm 3 Generation Algorithm of Zones in the Additional System Copies (N, N_b, max_{zone})

Require: N : network topology, N_b : list of boundary buses, max_{zone} : the size of the largest zone in copy 1.

Ensure: N_c is the zone generated in copy c

```

1: c = 1
2: while  $N_b \neq \emptyset$  do
3:   select bus =  $N_b(1)$ 
4:    $s_z = 1$                                      ▷ Set the size of the zone to 1
5:    $N_c = neighbor(bus)$                       ▷ Find all buses incident to the selected bus
6:   while  $s_z \leq max_{zone}$  do
7:      $N_c = N_c \cup neighbor(N_c)$ 
8:     calculate  $s_z = size(N_c)$ 
9:   end while
10:  find internal bus list,  $N_{internal,c}$ , in the  $N_c$ ,
11:  remove  $N_{internal,c} \cap N_b$  from  $N_b$ 
12:  c = c + 1
13: end while
14: Delete redundant copies if any
15: return  $N_{n_z}$ 
```

Bad Data Performance of MPD SE: While the developed SE algorithm in the first budget period of the project provides unbiased robust estimates using the LAV based SE method, the robustness against bad data should be maintained in the modified parallel structure. The bad data performance of the proposed MPD SE is presented on IEEE 14 bus system using detailed numeric results.

Fig. 6 shows the IEEE 14-bus system partitioned into two zones. The two types of buses, namely internal and boundary buses for this system are identified. Buses 6 and 12 belong to different zones, and are both designated as boundary buses. On the other hand, Buses 1,2,3 and 5 are internal buses of zone 1. The partitioning given in Fig. 6 reflects the configuration of the original copy of the system. Two other zones are generated in the additional copies. Table 7 shows the designation of all buses either as an internal (I) or a boundary (B) bus for the three copies of the system.

In processing Copy 1, zones 1 and 2 will be solved individually in parallel. Note that, the current phasor measurements between boundary buses will have to be discarded by individual zone state estimators. This loss of redundancy at the boundary may lead to reduced bad data detection capability. This issue will be addressed by introducing multiple copies of the system with different partitioning where each bus is an internal bus in at least one copy. As seen in Table 7, each bus is an internal bus in one of the copies. The strategic zone generation in the additional system copies is shown in Fig. 7a and Fig. 7b.

Table 7: The boundary and internal buses in each copy of the system.

[width=10em]Copy	Bus 1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	I	I	I	B	I	B	B	I	B	I	B	B	B	I
2	NA	NA	NA	NA	B	I	NA	NA	NA	B	I	I	I	B
3	NA	B	I	I	B	NA	I	I	I	B	NA	NA	NA	B

For the considered 14-bus example, a fully measured system is assumed. The LAV based SE problem is solved for two zones in the original copy and zones in the additional copies as

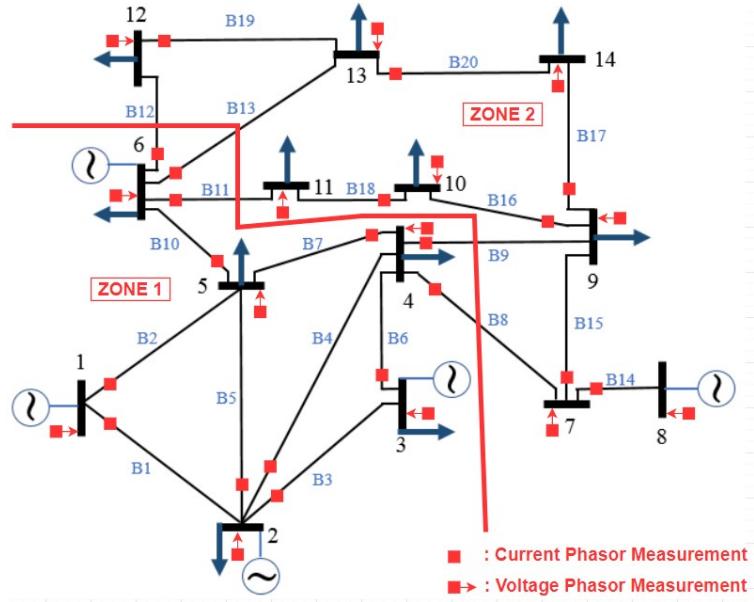


Figure 5: IEEE 14 bus system partitioned into 2 zones

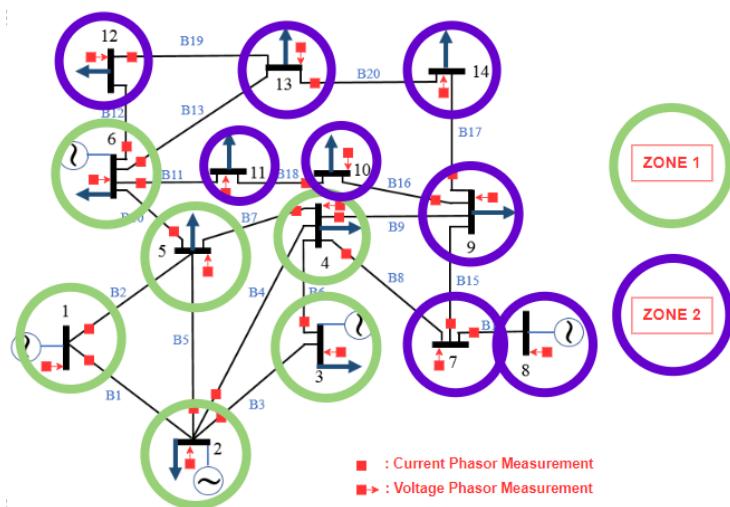


Figure 6: IEEE 14 bus system partitioned into 2 zones

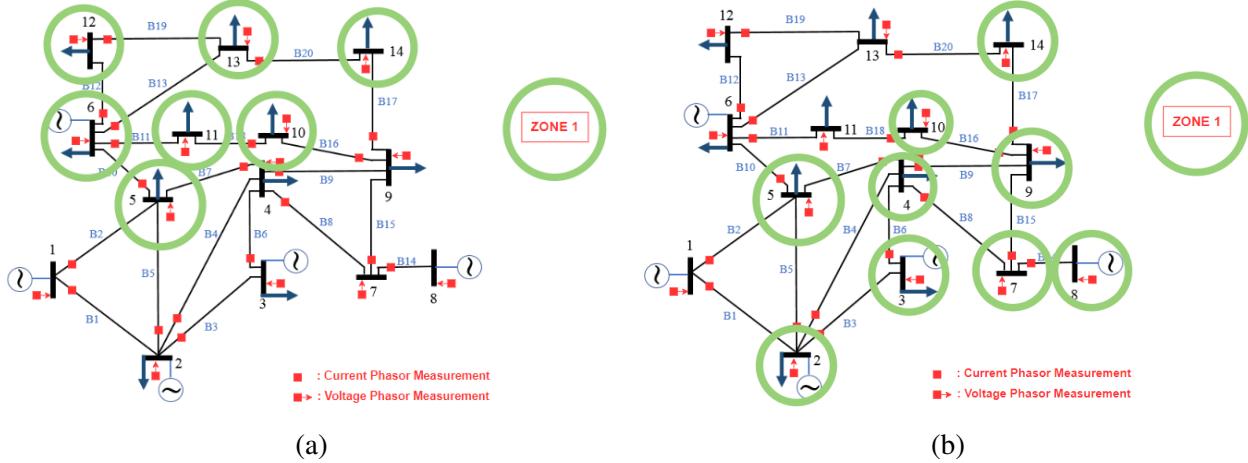


Figure 7: Additional zones generated in IEEE 14 bus system (a) in Copy 2, (b) in Copy 3.

independent sub-problems. Since there are multiple estimates of the states in different copies, the resultant estimate is obtained by taking the mean value of the estimates of internal buses in each copy. This yields a robust estimate of the system state, free of gross errors.

The implemented distributed SE algorithm is tested on IEEE 14 bus system. The results of the developed algorithm are compared with the results of centralized conventional SE. The accuracy and robustness of the distributed SE is tested under various bad data injections. The metric to evaluate the accuracy of the estimator is Mean Squared Error (MSE) which is calculated using equation 7.

$$MSE = \sqrt{\frac{1}{n} \cdot (X_{estimated} - X_{true})^2} \quad (7)$$

where n is the number of system states.

As a base case, the accuracy of the centralized state estimation (CSE) and MPD SE is tested under perfect measurement set with full redundancy. It is assumed that each bus has voltage phasor measurement, and each branch has current phasor measurement as shown in Fig. 6. Each copy of the system is solved with local measurements. Moreover, Gaussian noise with 0.001 variance is introduced to voltage phasor measurements, and noise with 0.01 variance is introduced to current phasor measurements to model realistic erroneous measurements. The MSE values shown in Table 8 are obtained by 100 Monte Carlo simulations. The accuracy of the developed distributed method is approximately same with the centralized SE accuracy under perfect and noise measurement sets with full redundancy as expected.

Table 8: Accuracy of CSE and MPD SE under different measurement configurations.

	Centralized SE	Distributed SE
MSE under perfect measurement set	2.522×10^{-13}	1.74×10^{-13}
MSE under noisy measurement set	4.81×10^{-4}	4.326×10^{-4}

To show the bad data rejection capability of the proposed algorithm, the actual values of random

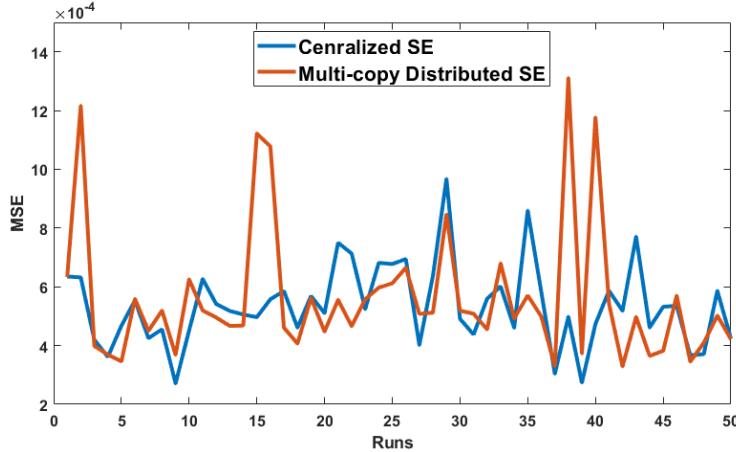


Figure 8: MSE of CSE and MPD SE under random single bad data injection

Table 9: The true and estimated states of Bus 6 in IEEE 14 bus system

	V_A	V_B	V_C	θ_A	θ_B	θ_C
True States of Bus 6	1.1138	1.1138	1.1138	-13.9501	-133.9502	106.0500
Estimated States of Bus 6 in Copy 1	1.2947	1.1138	1.1138	0.1944	-133.9504	106.0503
Estimated States of Bus 6 in Copy 2	1.1138	1.1138	1.1138	-13.9500	-133.9501	106.0499

measurement are manipulated by -%200 in 50 runs. As shown in Fig. 8, MSE of the estimators are comparable under single bad data cases.

The MSE for each SE run in 3 copies are given separately in Table 10. The resultant system estimates will be given by the distributed SE.

Table 10: Accuracy of CSE and MPD SE under a specific bad data injection

	Centralized SE	Copy 1	Copy 2	Copy 3	Distributed SE
MSE	2.37×10^{-11}	0.0011	4.44×10^{-11}	5.48×10^{-11}	1.98×10^{-11}

Scalability Performance of MPD SE: The computational performance tests are conducted in the PPM network with 84 buses as well as a very large-sized network (VLSN) with 12589 buses. The change of the computational time of the MPD SE with the varying number of zones as well as the performance of the conventional centralized SE are given to reveal the improvement in the computational time. All tests are conducted in MATLAB environment using Intel(R) Core i7 2.30 GHz computer with 16 cores.

Although MPD SE slightly reduces the computational time in the PPM network with 84 buses, the improvement in the computational time is significantly evident in the VLSN. The PPM network is a fully radial network whereas the VLSN is a meshed network. Moreover, VLSN is converted into a fully radial structure by removing some branches to test the scalability of the proposed method in large radial networks.

PPM network is 84 bus fully radial structure and the graph of the system is constructed in the

MATLAB environment to visualize the network as given in Fig. 9. To demonstrate the MPD SE in the PPM network, the original copy of the PPM network is partitioned into 4 zones. Using Algorithm I which is developed to partition the fully radial structures, the system is firstly divided into two zones by the removal of the branch between Buses 23 and 25. The branch between buses 23 and 25 minimizes the difference between the sizes of the two sub-networks. In the second iteration, each of the sub-networks is partitioned into 2 sub-sub-networks which results in a total of 4 zones.

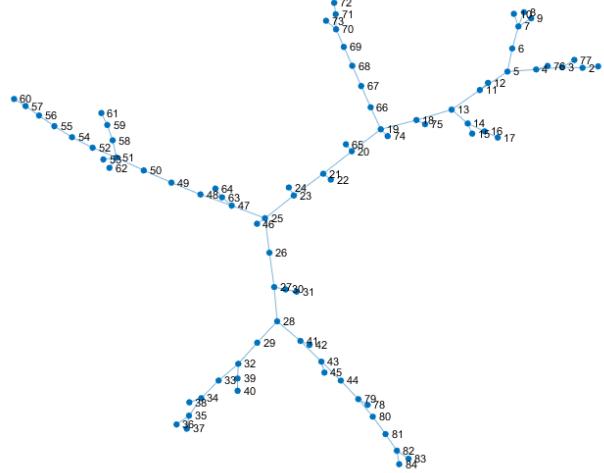


Figure 9: PPM Network Visualized in MATLAB environment.

Once the original copy is partitioned into pre-defined n_0 many zones, the boundary buses are marked so that Algorithm III can be executed to create additional system copies. In this case, three additional system copies are generated to ensure that each bus is an internal bus in at least one copy. The sizes of each zone in every copy are given in Table 11 where the zones created in the additional system copies are strictly smaller than the largest zone in the original copy.

Table 11: The size of each zone in every system copy.

	Zone 1 of Copy 1	Zone 2 of Copy 1	Zone 3 of Copy 1	Zone 4 of Copy 1	Zone of Copy 2	Zone of Copy 3	Zone of Copy 4
Size of the Zone	24	23	19	18	22	21	21
Total Size of the System	84						

MPD SE is executed with the proposed zone partitioning algorithm, and computational performance is given in Table 12. The resultant computational time of the MPD SE with 4 zones in the original copy is determined by the largest zone (Zone 1 of Copy 1). Comparing 0.0031 seconds of computational performance with the proposed method to the 0.0111 seconds of centralized solution performance, it can be concluded that computational time is improved by more than 3 times by partitioning the system into 4 zones. In the following parts of the simulation results, VLSN will be employed to show the superiority of the MPD SE in terms of computational performance in large networks.

Table 12: The computational performance of MPD and centralized SE solution.

	Zone 1 of Copy 1	Zone 2 of Copy 1	Zone 3 of Copy 1	Zone 4 of Copy 1	Zone of Copy 2	Zone of Copy 3	Zone of Copy 4
Local SE Run Time (s)	0.0031	0.0028	0.0023	0.0021	0.0026	0.0026	0.0025
Centralized SE Run Time (s)	0.0111						

A fully radial network with 12589 buses is utilized to test the scalability of the proposed method in the large radial networks. The radial structured VLSN is given in Fig. 10. To demonstrate the MPD SE in the radial VLSN, the original copy of the network is partitioned into 4 and 8 zones. The system is partitioned using Algorithm I. When the original copy is partitioned into 4 zones, only one additional copy of the system is generated using Algorithm III. When the original copy is partitioned into 8 zones, three additional copy of the system is generated using Algorithm III.

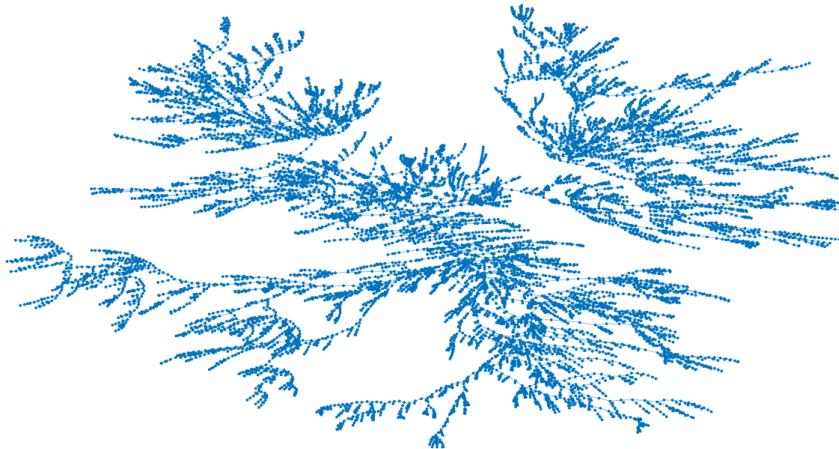


Figure 10: Radial Structured VLSN Visualized in MATLAB environment.

As shown in Table 13, while the centralized SE solution employs 1 core and takes approximately 520 seconds in the radial structured VLSN, the computational time decreases and the number of required processor cores increases as the system is partitioned. While the original copy is partitioned into 8 zones, a total of 11 cores is required where the total computational time is almost 1/120 of the centralized solution. With the 16 zones in the original copy, computational time decreases to nearly a second. Note that by using Algorithm I instead of graph search-based Algorithm II, similar-sized zones are obtained which reduces the computational time significantly. The number of cores required for parallel processing is calculated by the number of independent local SE runs in each case. When the number of zones in the original copy is 8, on top of the 8 SE runs in the 8 zones of the original copy, there are 3 strategic zones created in the 3 additional system copies which make a total of 11 independent SE runs.

Lastly, A meshed network with 12589 buses is utilized to test the scalability of the proposed method in the large meshed networks. The ring-structured VLSN is given in Fig. 11. To demonstrate the MPD SE in the meshed VLSN, the original copy of the network is partitioned into 5 zones. The system is partitioned using Algorithm II which employs graph search-based partition-

Table 13: The computational performance of MPD and centralized SE solution in radial VLSN.

	Centralized Solution	MPD SE (4 zones)	MPD SE (8 zones)	MPD SE (16 zones)
Computational Time	524.21 seconds	13.33 seconds	4.33 seconds	1.47 seconds
Required # of Cores	1	5	11	21

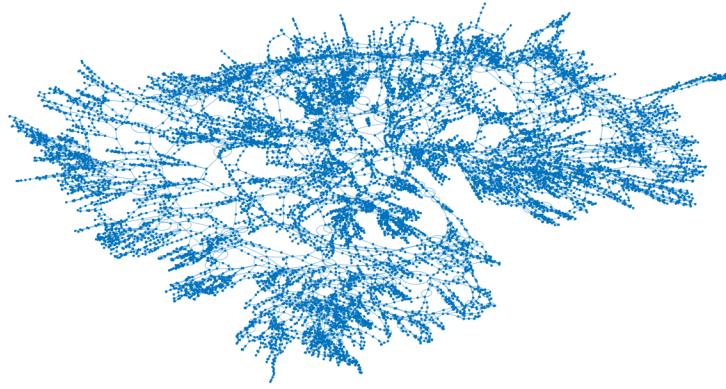


Figure 11: Meshed Structured VLSN Visualized in MATLAB environment.

ing. When the original copy is partitioned into 5 zones, 5 additional copy of the system is generated using Algorithm III.

As shown in Table 14, while the centralized SE solution employs 1 core and takes approximately 580 seconds in the mesh structured VLSN, the computational time decreases and the number of required processor cores increases as the system is partitioned. While the original copy is partitioned into 5 zones, a total of 10 cores is required where the total computational time is almost 1/60 of the centralized solution. With the 20 zones in the original copy, computational time decreases to nearly a second.

Milestone 2.8.1: Metric that will evaluate the scalability of the state estimator. CPU time of state estimation should be commensurate with desirable frequency of voltage control.

This milestone is concerned about scalability of LAV estimator for large scale systems containing more than 1000 nodes. It considers the slow moving voltage dynamics as its main target and tests the proposed massively parallel state estimation method's cpu performance relative to frequency of voltage control. It should be noted that considering the computational power of the test bench, a limited number of cores, and the not fully optimized code used in our tests, the computational time can still be further reduced. However, by assuming as many processors as system copies and areas, as evident from the results shown in Tables 13 and 14 milestone 2.8.1 is successfully achieved for a test system with over 12,000 buses under both meshed and radial network configurations.

Table 14: The computational performance of MPD and centralized SE solution in meshed VLSN.

	Centralized Solution	MPD Solution with 5 zones	MPD Solution with 20 zones
Computational Time	582.6 seconds	10.77 seconds	1.16 seconds
Required # of Cores	1	10	30

Subtask 8.2: *Closed-loop Formulation between Robust WLAV Estimator and NG-LSTM Synchronizer and Testing on Standard Test Systems*

Subtask Summary: *In this subtask, we developed a closed-loop operation framework between the state estimator developed in Subtask 8.1 and the measurement predictor developed in Subtask 7.1. The key idea is to use the measurement residual produced by the WLAV state estimator as a feedback of the measurement prediction such that the machine learning model can be improved and adapted online.*

Validation of Closed Loop Operation of Measurement Predictor and State Estimator. We use clean data, corrupted data generated from subtask 6.1, and residual-corrected data generated from the WLAV state estimator developed in Task 8.1 for evaluation of the proposed closed-loop operation framework. We down-sample the node active and reactive power injection data to 30-minute intervals as the SA measurement input of the model from AMI, and the 1-minute interval data is only used for evaluation. The line active and reactive power injection data and node voltage data remain 1-minute intervals as they are assumed to be FS measurements from SCADA.

For model training, we use data from the last 20 days of January and February. For the model test, we use data from the last 20 days of March. The time step of the model is set to 30, and we evaluate the predicted results by mean absolute error (MAE).

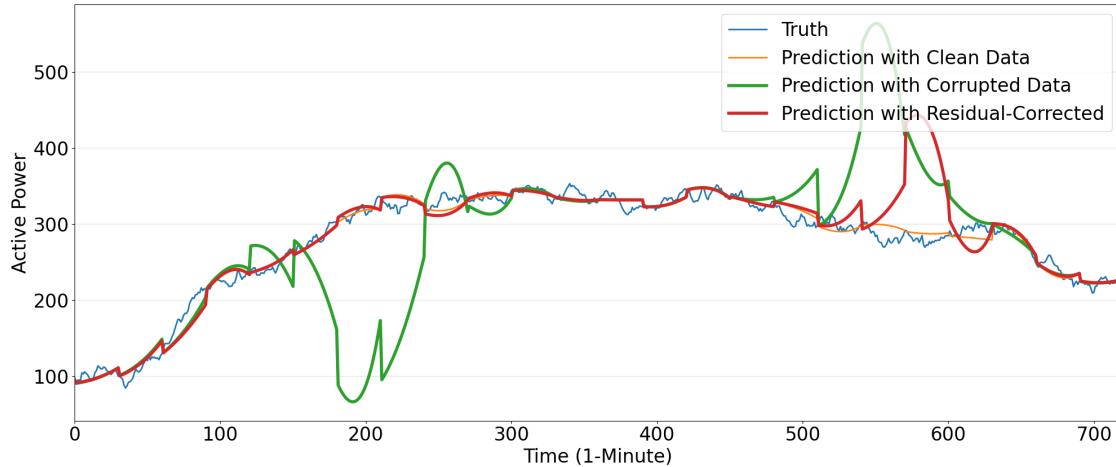
Table 15: FS measurement prediction results with different data over the first group of bus voltage control and distribution system topology information in terms of MAE.

Model	Data	Active Power	Reactive Power
NG-Transformer	Clean	11.9233	5.7904
NG-Transformer	Corrupted	29.1641	9.7516
NG-Transformer	Residual-Corrected	13.9248	6.2632

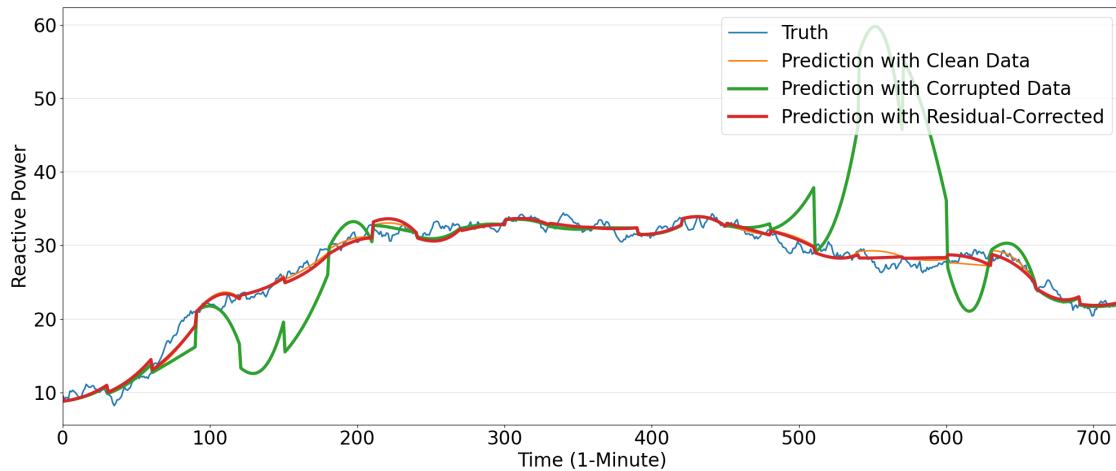
We evaluate the first group of bus voltage control and distribution system topology information. Table 30 shows the prediction results with clean, corrupted, and residual-corrected data. For model training, we use the same NG-Transformer model with different SA and FS measurement data but the same topology information. The topology information varies 10 times throughout the whole period. For the test, we use different kinds of data as input, but use ground truth from clean data for all of the trained models.

Milestone 2.8.2: Metric that will assess relative reduction of RMSE of SA measurement prediction. The improvement should be greater than 20% to meet the success criteria.

From the results of Table 30, we can see that corrupted data has a significant influence on model training, and the models' performance drops when data is corrupted compared to clean data. When the data was corrected by residual provided by the WLAV state estimator, the performance improved a lot compared to corrupted data. For active power prediction, the improvement is $\frac{29.1641 - 11.9233}{(29.1641 - 11.9233) - (13.9248 - 11.9233)} = 88.39\% > 20\%$, and for reactive power prediction, the improvement is $\frac{9.7516 - 5.7904}{(9.7516 - 5.7904) - (6.2632 - 5.7904)} = 88.06\% > 20\%$, satisfying Milestone 2.8.2. Figure 12 and Figure 13 are examples of prediction results with different kinds of data, from where we can see the residual-corrected data brings a significant improvement to the prediction performance compared to the corrupted data. This validates the concept of the project that the feedback of the state



(a) Active power prediction for Node 93



(b) Reactive power prediction for Site 93

Figure 12: Prediction results using different kinds of data for Node 93.

estimator is beneficial for the ML-based predication of SA measurements.

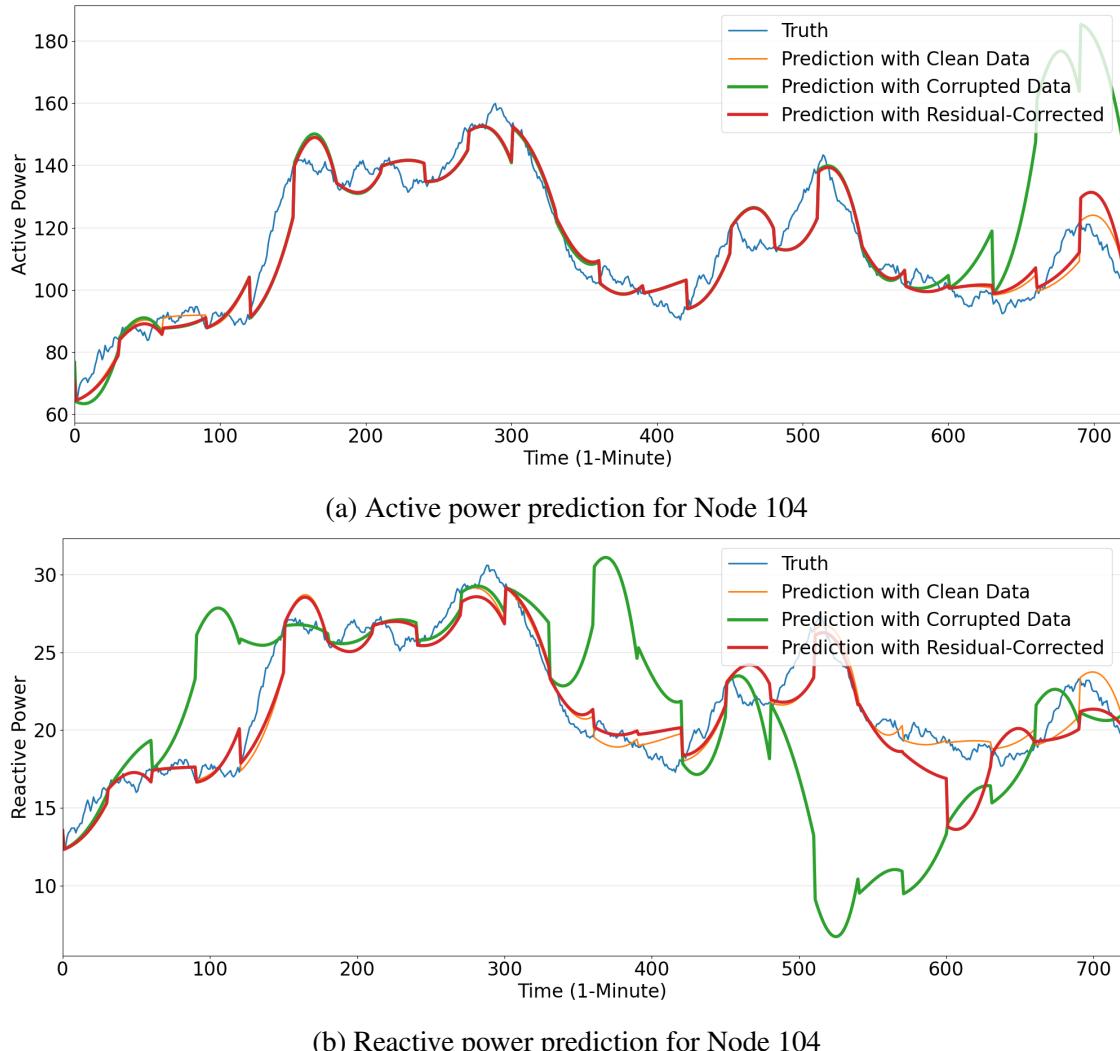


Figure 13: Prediction results using different kinds of data for Node 104.

Subtask 4.1: Detecting Line Switching and Generation/Load Switching

Subtask Summary: We will use sparse estimation methods to obtain the locations and values of these nonzero entries. By solving this estimation problem, we only need a very small number of PMU measurements to accurately locate the switched lines in a large-scale distribution system, which remarkably improve the dynamic event awareness of sparsely measured systems. Note that, locations of the small number of PMUs will be determined based on our previous work (citation: G. Feng and A. Abur, "Fault Location Using Wide-Area Measurements and Sparse Estimation," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2938-2945, July 2016. doi: 10.1109/TPWRS.2015.2469606)

To make a model for the line outage detection, DC power flow model is used since it provides a reasonably accurate linear approximation between real power injections and voltage phase angles [14, 15]. The following linear formulation can be used to relate the changes in bus phase angles ($\Delta\theta$) to the real power injections (ΔP) [16].

$$B * \Delta\theta = \Delta P \quad (8)$$

where,

- $B = \begin{cases} if k \neq j \rightarrow B_{kj} = -\frac{1}{x_{kj}} \\ if k = j \rightarrow B_{kk} = \sum_{j=1, j \neq k}^N \frac{1}{x_{kj}} \\ 0 \quad otherwise \end{cases}$
- x_{kj} is the branch reactance value between bus k and bus j,
- $\Delta\theta$ is the angle difference of buses (post outage theta – pre outage theta),
- ΔP is the injections at corresponding buses connected to disconnected line.

To detect outages in a power system with limited number of PMU measurements, the system can be partitioned into two parts, containing those buses with PMUs and those without:

$$\begin{bmatrix} \Delta P_e \\ \Delta P_i \end{bmatrix} = \begin{bmatrix} B_{ee} & B_{ei} \\ B_{ie} & B_{ii} \end{bmatrix} * \begin{bmatrix} \Delta\theta_e \\ \Delta\theta_i \end{bmatrix} + \begin{bmatrix} \Delta e_e \\ \Delta e_i \end{bmatrix} \quad (9)$$

where,

- subscript i refers to buses with PMUs and,
- subscript e refers to buses without PMUs.

When one of the lines goes out-of-service, B matrix and the bus angles ($\Delta\theta$) will change accordingly. However, instead of modifying the matrix B, line outage can be represented by virtual power injections (ΔP_s and ΔP_t) at the terminal buses of the disconnected branch. The representation of the line outage by using virtual power injections can be illustrated using Figures 14 and 15.

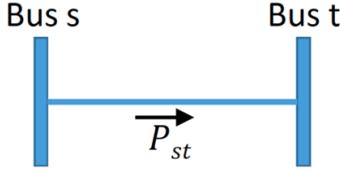


Figure 14: Pre-contingency flow on branch s-t.

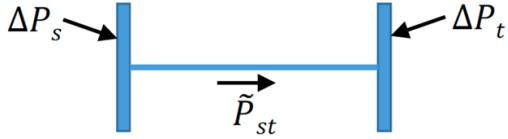


Figure 15: Post-contingency flow on branch s-t

The virtual power injections (ΔP_s and ΔP_t) in Figure 15 have the same magnitude but opposite signs, and their absolute value is equal to the real power flow of the line, making the net flow effectively zero while matching the terminal bus phase angles to the post-outage solution. This approach facilitates modeling of outages without physically removing the lines, but modifying the terminal power injections as:

$$\Delta P_s = -\Delta P_t = \tilde{P}_{st} \quad (10)$$

In order to solve 9, a well-known sparse solution method, namely LASSO is used. Using the partitioned equation, the phase angles can be expressed as [17]:

$$\begin{bmatrix} \Delta\theta_e \\ \Delta\theta_i \end{bmatrix} = \begin{bmatrix} B_{ee} & B_{ei} \\ B_{ie} & B_{ii} \end{bmatrix}^{-1} * \begin{bmatrix} \Delta P_e \\ \Delta P_i \end{bmatrix} + \begin{bmatrix} \Delta e_e \\ \Delta e_i \end{bmatrix} \quad (11)$$

Matrix B^{-1} is typically badly conditioned, i.e. it is nearly singular, and a LASSO-based sparse estimation approach may fail or misidentify actual outages due to the ill-conditioned B^{-1} having nearly collinear columns [18, 19]. Hence, instead of taking its inverse, it is transformed by QR decomposition. However, there are only a limited number of PMU measurements among the buses. Therefore, using QR decomposition without partitioning B matrix as in Equation 2 is impractical, since all phase angles must be known to obtain the product of R by $\Delta\theta$. Based on these considerations, the following ordered formulation can be used instead:

$$Q_{ordered} * R_{ordered} * \Delta\theta_{ordered} = \Delta P_{ordered} \quad (12)$$

$$\begin{bmatrix} Q_{ee} & Q_{ei} \\ Q_{ie} & Q_{ii} \end{bmatrix} * \begin{bmatrix} R_{ee} & R_{ei} \\ 0 & R_{ii} \end{bmatrix} * \begin{bmatrix} \Delta\theta_{ee} \\ \Delta\theta_{ii} \end{bmatrix} = \begin{bmatrix} \Delta P_{ee} \\ \Delta P_{ii} \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} R_{ee} & R_{ei} \\ 0 & R_{ii} \end{bmatrix} * \begin{bmatrix} \Delta\theta_{ee} \\ \Delta\theta_{ii} \end{bmatrix} = \begin{bmatrix} Q_{ee}^T & Q_{ei}^T \\ Q_{ie}^T & Q_{ii}^T \end{bmatrix} * \begin{bmatrix} \Delta P_{ee} \\ \Delta P_{ii} \end{bmatrix} \quad (14)$$

$$R_{ii} * \Delta\theta_{ii} = [Q_{ie}^T Q_{ii}^T] * \begin{bmatrix} \Delta P_{ee} \\ \Delta P_{ii} \end{bmatrix} \quad (15)$$

By solving Equation 31 using LASSO, the virtual power injections at the terminal buses of disconnected lines can be obtained. In addition to that, the number of non-zeros found by LASSO solution can be reduced further by utilizing the connectivity matrix A . This will yield the following:

$$R_{ii} * \Delta\theta_{ii} = \begin{bmatrix} Q_{ie}^T & Q_{ii}^T \end{bmatrix} * A^T * \begin{bmatrix} \Delta \tilde{P}_{ee} \\ \Delta \tilde{P}_{ii} \end{bmatrix} \quad (16)$$

where,

- A is the connectivity matrix $b \times n$.
- n is the number of buses and,
- b is the number of branches in the system.

Entries of the connectivity matrix used in Equation 16 are defined as:

$$A(k, m) = \begin{cases} 1 & \text{if } k \text{ is the sending bus at branch } m \\ -1 & \text{if } k \text{ is the receiving bus at branch } m \\ 0 & \text{otherwise} \end{cases} \quad \text{For all } m = 1, \dots, b \quad (17)$$

After incorporating the connectivity matrix as in Equation 16, the optimization problem to be solved by LASSO will take the following form:

$$\Delta \tilde{P} := \min_{\Delta \tilde{P}} \|\Delta \theta' - M \Delta \tilde{P}\|_2^2 + \lambda \|\Delta \tilde{P}\|_1 \quad (18)$$

where,

- $\Delta P = A^T * \Delta \tilde{P}$,
- $M = \begin{bmatrix} Q_{ie}^T & Q_{ii}^T \end{bmatrix} * A^T$,
- $\Delta \theta' = R_{ii} * \Delta \theta_{ii}$.

While the above LASSO formulation using partitioned QR decomposition significantly improves the solution, there will still be cases where the virtual power injections identified by $\Delta \tilde{P}$ vector may not correspond to the true outage, especially when there are too few PMUs installed in the system [20]. To identify and eliminate such false line outage detections during the sparse

estimation process, the post-outage power flow given in $\Delta\tilde{P}$ is compared with the predicted \tilde{P}_{st} value as shown in [21]:

$$\tilde{P}_{st} = \frac{P_{st}}{1 - PTDF_{(s,t),s,t}} \quad (19)$$

where,

- P_{st} refers to pre-outage flow along the line connecting bus s and bus t,
- \tilde{P}_{st} refers to virtual power flow which models the outage without physically removing the line.

$$PTDF_{(s,t),s,t} = \frac{1}{x_{st}}[(X_{ss} - X_{st}) - (X_{st} - X_{tt})] \quad (20)$$

where,

- s and t refers to sending and receiving terminals of the disconnected line,
- X_{ss} , X_{tt} , X_{st} refer to B^{-1} matrix entries corresponding to the line s-t,
- x_{st} refers to reactance of the branch between bus s and bus t.

If the mismatch between the result given by LASSO algorithm and \tilde{P}_{st} evaluated by Equation 19, does not remain below a specified margin, then the corresponding row of the detected line outage in A matrix is set to zero. This process is repeated until the mismatch satisfies the margin which is selected as %20, since there may be a value of λ that will make the LASSO result more accurate.

Subtask 4.2: Test and Validation on Small Systems

Subtask Summary: *Proof of concept for the developed sparse event identifier will be performed on IEEE 13-node and 123-node systems using synthetic measurements. A variety of scenarios, including line switching, generation and load switching. The impact of missing such events will be illustrated to highlight the significance of these tools.*

The network model of PPM will be used to synthesize dynamic PMU signals when line switching occurs. The total number of simulation cases will be determined based on the number of the significant branches in the BCM network. Significance of a branch will be determined by using the outage sensitivity of the state estimation output to the outage of that branch. Those branches with less than 3% sensitivity will be disregarded from consideration. The 3% threshold is chosen using 3 times the standard deviation of 0.01 p.u. as an acceptable number for a line flow.

To determine the significant branch outages in modified Bronzeville system, each branch was removed from the system iteratively and the results were obtained for those by utilizing the power-flow solution. Once all Voltage and Theta values were obtained for branch outages, the results have been compared with Voltage and Theta values of non-outage case. Then, in order to determine the significant branch outages, the abs norm value of these differences has been sought compared with 3% sensitivity. Based on the sensitivity analysis 33 line outages found as significant line outages in PPM network.

Then proposed line outage algorithm was tested for those 33 line outages. The results are given in Table 16.

For the load and generation outages, sensitivity analysis was performed, revealing that 26 loads were classified as significant load outages among the 46 total loads/generations. This is due to the very lightly loaded condition of the PPM network. Subsequently, using the network model and measurement configuration, pre-outage and post-outage voltage phasor measurements were generated using a three-phase power flow program for load and generation outage scenarios. These measurements were then used as inputs to evaluate the performance of the developed load and generation outage detection algorithm.

The results of load/generation outage detection algorithm for those load/generation outages are given in Table 17.

In addition to network model of PPM system, line outages for following systems are also investigated;

- 14 Bus System
- 118 Bus System
- 1354 Bus System

Simulation Results for Line Outage Detection in 14 Bus System:

This was tested by creating line outage scenarios for the three-phase 14-bus system, where three-phase voltage phasor measurements were assumed to be available at buses "2, 3, and 10". Sensitivity analysis was then performed to identify significant line outages within the 14-bus system. It was found that all line outages in the 14-bus system were classified as significant. Subsequently, the proposed line outage detection algorithm was applied, and the results are presented in Table 16.

Simulation Results for Line Outage Detection in 118 Bus System:

This time, the 118-bus system was tested by creating line outage scenarios for the three-phase 118-bus system, where 14 three-phase voltage phasor measurement units were assumed to be strategically placed within the network. Sensitivity analysis was then performed to identify significant line outages within the 118-bus system, revealing that 126 line outages were classified as significant. Following this, the proposed line outage detection algorithm was applied, and the results are presented in Table 16.

Simulation Results for Line Outage Detection in 1354 Bus System: To further evaluate the performance of the proposed algorithm, a larger system (1354-bus) was tested by creating line outage scenarios. Sensitivity analysis was conducted to identify significant line outages within the 1354-bus system, revealing that 677 line outages were classified as significant. Subsequently, the proposed line outage detection algorithm was applied, and the results are presented in Table 16.

Milestone 1.4.1 : Line/Load/Gen Outage Detection Capability

Milestone Summary : Percentage of detected line/load/gen switching cases > 95%

This milestone refers to the successful detection of line and load/gen outage and accuracy of the outage location. Below results validate this milestone whose detection error threshold metric was set at 95%.

Table 16: The results of proposed line outage detection algorithm.

System	PMU Coverage	Significant Line Outages	Success Rate (%)
PPM	22	33	90.91
14 Bus	21	14	100
118 Bus	21	126	98.41
1354 Bus	17	677	97.34

Table 17: The results of proposed load/gen outage detection algorithm.

System	PMU Coverage	Significant Load/Gen Outages	Success Rate (%)
PPM	22	26	96.15

In Table 16, the success rate of the line outage detection algorithm in the PPM network falls below the targeted percentage. This is due to the presence of multiple zero-load buses in the PPM network, which create multiple possible solutions for the LASSO algorithm. Among the unsuccessful cases, the proposed line outage detection algorithm consistently identifies one of the terminal buses of the disconnected line. However, due to the zero-injection buses, it misidentifies the other terminal bus. Despite this, since the affected buses are zero-injection buses, the detected outage line and the actual disconnected line are functionally equivalent. Therefore, the project can be declared to have successfully achieved this milestone, particularly when taking into account the proposed algorithm's performance in other systems.

Milestone 3.12.3 : Line/Load/Gen Outage Detection Capability

Milestone Summary : Percentage of detected line/load/gen switching cases > 95%

This milestone refers to the successful detection of line and load/generation outages, as well as the accuracy of outage location identification using the HIL environment. For this purpose, ComEd has provided example line outage scenarios data using the HIL environment for the Bronzeville system, then the proposed event detection algorithm has been performed and results are given in Table 18.

Table 18: The results of proposed line outage detection algorithm for HIL Data.

Disconnected Line	Detected Line
349 - 348.1	349 - 348.1
216 - 215.1	216 - 351
216 - 217	216 - 217
212 - 213	213.1 - 414
188 - 188.1	188 - 188.1
228 - 236	228 - 236
228 - 226.1	228 - 226.1
185.1 - 185	185.1 - 185
193 - 194	~

During the test, it was observed that two of the outage scenarios resulted in very small angle differences, which hindered the algorithm's ability to accurately determine the outage location. Therefore, these cases have been omitted from the showcase.

Another case, which involves a line outage between Bus 212 and Bus 213, was identified by the proposed algorithm as an outage between Bus 213.1 and Bus 414. The reason for this is that the line between Bus 212 and Bus 213 is connected through Bus 213-213.1 and then Bus 213.1-414. However, all these buses are zero-injection buses, meaning that the impact of disconnecting Line 212-213 is practically the same as disconnecting Line 213.1-414. Therefore, this case is considered a successful detection.

The proposed algorithm successfully identified 7 out of 9 scenarios, resulting in a success rate of $\frac{7}{9} = 78\%$. While this is below the predetermined threshold, it is important to note that the PMU locations provided by ComEd were not optimized for event detection purposes. Therefore, the success rate of the proposed algorithm can be further improved by strategically placing PMUs in a more optimal manner, considering their effectiveness in detecting events [22]. This is proved in previous report, where, optimizing the locations of the PMUs significantly boost the performance of the event detection algorithms.

The final key observation is that the Bronzeville system is a very lightly loaded network with multiple zero-injection buses. This characteristic also impacts the performance of the event detection algorithm based on the Least Absolute Shrinkage and Selection Operator (LASSO), as it introduces challenges in accurately identifying outage locations due to minimal voltage and angle deviations.

Subtask 9.1: Detecting and Locating Faults

Subtask Summary: We will use the limited set of available PMU measurements for detecting and locating unexpected short circuit faults occurring along distribution lines. This follows a similar formulation to the one above and uses measured changes in node voltage phasors due to the fault. The approach has two stages: the first stage involves a sparse estimation problem where sudden changes in “virtual” injections at the two terminals of the faulted branch are detected and estimated using the virtual fault model that replaces faults by equivalent terminal bus injections; in the second stage, these estimated virtual injections are used to compute the exact location of the fault along the faulted branch. The success of the approach closely depends on the accurate measurements obtained from a few PMUs located in the power grid. Selection of the number and location of PMUs will follow the same approach used in Subtask 4.1 in BP1.

When a fault occurs along a transmission line, its impact will be observed at all system buses as a voltage transient. In the hypothetical (not realistic) scenario where none of the protective relays operate, the impact of the fault will diminish gradually, and bus voltages will settle down at a new post fault steady state operating point. Therefore, all bus voltages will change with respect to their pre fault steady state conditions [23]. Faults can be visualized as shown in Figure 16.

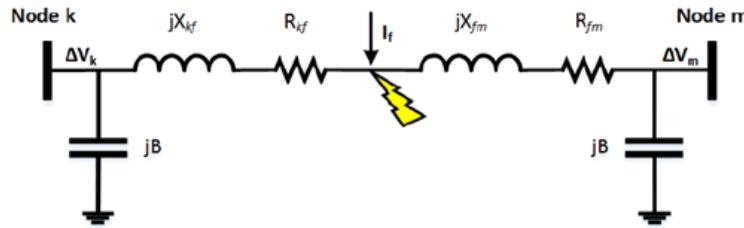


Figure 16: Fault on branch k-m.

It can be shown that the fault current can be replaced by equivalent virtual current injections at the faulted line terminal buses as shown in Figure 17.

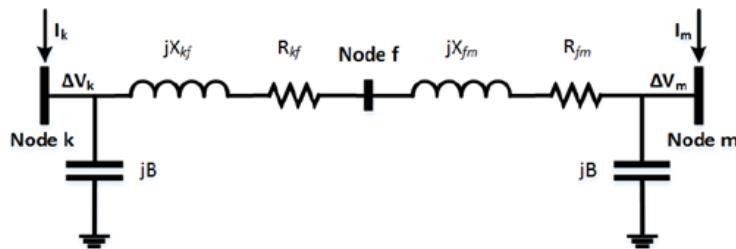


Figure 17: Virtual Current Injections on adjacent buses for faulted branch k-m.

Note that, if all buses are measured by PMUs, it is trivial to directly solve for the virtual fault current injections using the following equation.

$$Y * \Delta V = \Delta I \quad (21)$$

where,

- Y is the bus admittance matrix,

- ΔV is the voltage phasor difference between pre and post fault conditions,
- ΔI is the sparse virtual fault current injection vector.

Given that in most power systems only a few buses will be measured by PMUs. Therefore, a sparse estimation method needs to be used to find the virtual fault currents based on the measurements made by the few PMUs for the pre and post fault voltages [24].

Initially, three-phase pre and post fault measurements are converted to the sequence domain as follows using the well documented symmetrical component transformation below [25]:

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix} = T^{-1} * \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} I_0 \\ I_1 \\ I_2 \end{bmatrix} = T^{-1} * \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \quad (23)$$

$$Y^{012} = T^{-1} * Y_{abc} * T \quad (24)$$

where,

- 0, 1, 2 represent the “zero, positive, and negative” sequence networks respectively,
- a, b, c represent the “Phase A, Phase B, and Phase C” respectively,
- T is the transformation matrix given by:

$$T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \quad (25)$$

where, $a = 1\angle 120^\circ$.

Then positive sequence network is utilized to determine the faulted branch location.

Consider a power system where each bus is assigned to either partition “i” which contains the buses with PMUs or to partition “e” containing buses without PMUs. Writing (21) using this partition and adding measurement errors e_e and e_i to represent the measurement noise associated with PMU measurements:

$$\begin{bmatrix} \Delta I_e \\ \Delta I_i \end{bmatrix} = \begin{bmatrix} Y_{ee} & Y_{ei} \\ Y_{ie} & Y_{ii} \end{bmatrix} * \begin{bmatrix} \Delta V_e \\ \Delta V_i \end{bmatrix} + \begin{bmatrix} \Delta e_e \\ \Delta e_i \end{bmatrix} \quad (26)$$

where,

- subscript i refers to buses with PMUs and,

- subscript e refers to buses without PMUs.

Rewriting (26) to express the voltage changes in terms of the virtual currents:

$$\begin{bmatrix} \Delta V_e \\ \Delta V_i \end{bmatrix} = \begin{bmatrix} Y_{ee} & Y_{ei} \\ Y_{ie} & Y_{ii} \end{bmatrix}^{-1} * \left(\begin{bmatrix} \Delta I_e \\ \Delta I_i \end{bmatrix} - \begin{bmatrix} \Delta e_e \\ \Delta e_i \end{bmatrix} \right) \quad (27)$$

Matrix Y^{-1} is typically badly conditioned, hence, QR decomposition is used to improve its condition number. Keeping the same partitioning, QR decomposition is applied to (26) yielding:

$$Q_{ordered} * R_{ordered} * \Delta \theta_{ordered} = \Delta P_{ordered} \quad (28)$$

$$\begin{bmatrix} Q_{ee} & Q_{ei} \\ Q_{ie} & Q_{ii} \end{bmatrix} * \begin{bmatrix} R_{ee} & R_{ei} \\ 0 & R_{ii} \end{bmatrix} * \begin{bmatrix} \Delta V_{ee} \\ \Delta V_{ii} \end{bmatrix} = \begin{bmatrix} \Delta I_{ee} \\ \Delta I_{ii} \end{bmatrix} \quad (29)$$

$$\begin{bmatrix} R_{ee} & R_{ei} \\ 0 & R_{ii} \end{bmatrix} * \begin{bmatrix} \Delta V_{ee} \\ \Delta V_{ii} \end{bmatrix} = \begin{bmatrix} Q_{ee}^T & Q_{ie}^T \\ Q_{ei}^T & Q_{ii}^T \end{bmatrix} * \begin{bmatrix} \Delta I_{ee} \\ \Delta I_{ii} \end{bmatrix} \quad (30)$$

$$R_{ii} * \Delta V_{ii} = [Q_{ei}^T Q_{ii}^T] * \begin{bmatrix} \Delta I_{ee} \\ \Delta I_{ii} \end{bmatrix} \quad (31)$$

Since only those buses in partition “i” will have voltage measurements, (31) will have to be used to solve for the virtual currents. This is an underdetermined set of equations, yet the unknown vector is known to be sparse. Hence, sparse estimation can be used to solve it. Using the well documented method of LASSO, the following optimization problem can be formulated and solved by LASSO [26]:

$$\Delta I := \min_{\Delta I} \|\Delta V' - A * \Delta I\|_2^2 + \lambda \|\Delta I\|_1 \quad (32)$$

where,

- $\Delta V' = R_{ii} * \Delta V_{ii}$,
- $A = [Q_{ei}^T \ Q_{ii}^T]$.

Given the fact that (32) contains complex numbers, their real and imaginary parts are used to transform the optimization problem into a strictly real one as below:

$$\Delta I := \min_{\Delta I} \left\| \begin{bmatrix} \Delta V'_{real} \\ \Delta V'_{imag} \end{bmatrix} - \begin{bmatrix} A_{real} & -A_{imag} \\ A_{imag} & A_{real} \end{bmatrix} * \Delta I \right\|_2^2 + \lambda \|\Delta I\|_2 \quad (33)$$

Solving (33) with LASSO will yield the location and values of virtual current injections. While the locations are highly accurate, the values may be close approximations. Thus, a third step is executed to determine all three sequence currents.

Once the location of the fault is identified, the virtual sequence currents can be estimated with higher accuracy by using the OLS method as follows [27]:

$$[Q_{bus1}^{fault} \ Q_{bus2}^{fault}] * \begin{bmatrix} \Delta I_{bus1}^{fault} \\ \Delta I_{bus2}^{fault} \end{bmatrix} = \Delta V' \quad (34)$$

where,

- Q_{bus1}^{fault} and Q_{bus2}^{fault} denotes the reduced $[Q_{ei}^T \ Q_{ii}^T]$ matrix, with only column elements of the associated fault buses and all line elements of the measured buses,
- ΔI_{bus1}^{fault} and ΔI_{bus2}^{fault} , denotes the reduced current vector, with only line elements of the associated fault buses,
- $\Delta V' = R_{ii} * \Delta V_{ii}$,

The solution to (34) can be found directly using the Ordinary Least Squares (OLS) algorithm:

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (35)$$

where,

- $\hat{\beta}$ is a $p \times 1$ of unknown parameters,
- y is $n \times 1$ vectors of the response variables,
- X is an $n \times p$ of regressors whose row i is x_i^T and contains the i_{th} observations on all the explanatory variables.

Then, those calculated virtual sequence currents are transformed back to phase domain using (25) and the distance of the fault is calculated using the (36) and (37).

Finally, the fault distance from each terminal of the faulted branch k-m can then be calculated by the following expressions:

$$FaultLocation(m) = \frac{|I_{fk}|}{|I_{fk}| + |I_{fm}|} * 100\% \quad (36)$$

$$FaultLocation(k) = \frac{|I_{fm}|}{|I_{fk}| + |I_{fm}|} * 100\% \quad (37)$$

where,

- I_{fm} is the virtual fault current at bus m,

- I_{fk} is the virtual fault current at bus k.

Subtask 9.2: Test and Validation on Small Systems

Subtask Summary: *Proof of concept for the developed sparse event identifier will be performed on PPM using synthetic measurements. This will involve simulation of faults along all “significant” branches in the power grid where “significant” is defined as those branches whose switching will have an impact on the bus voltages by more than 3%. Special cases where the grid has radial protruding sections will be identified to illustrate the limitations introduced by observability considerations. A variety of scenarios, including different types (single line-ground, three-phase, double-line-ground, line-line) of short-circuit faults will be used in testing / validating the developed approach. The impact of missing such events will be illustrated to highlight the significance of the developed tool.*

In Subtask 9.1, the methodology of the fault detection algorithm is given.

In Subtask 9.2, to demonstrate the performance of the fault detection algorithm using the exact PMU locations provided by the ComEd, the node configuration of PPM network is kept still as 196 node by simply replacing those circuit breakers with low impedance lines.

Then, 11 different branches are selected to create case samples. For each branch, different cases are created to mimic various type of faults occurred at different locations along those branches. After all, total of 105 fault cases are created as in Table 19.

In Table 19, the "Branch" column indicates the faulted branch, the "Fault Type" column indicates the type of the fault and finally the "Fault Location" column indicates how far the fault is relative to the sending end bus and receiving end bus.

After all cases are created, the fault detection method detailed in Subtask 9.1 is repeatedly tested on each of these cases. During the test, pre and post voltage values are assumed to be measured for those PMU locations provided by ComEd. The fault types and locations are calculated using the fault detection algorithm for cases in Table 19 and the results are given in Table 20.

As evident from the simulation results given in Table 20, the fault detection algorithm successfully detects "99" out of "105" fault cases with at most "3%" error margin for a detected fault location. The maximum error margins calculated for results in Table 20 are given in Table 21.

Milestone 2.9.1 : Fault Detection Capability and Fault Locating Accuracy

Milestone Summary : Detection error < 10%, Fault location accuracy < 3%

This milestone refers to the successful detection of faults and accuracy of the fault location. Above results validate this milestone whose detection error threshold metric was set at 10% which was actually found to be much less around 6% ($\frac{6}{105} \cdot 100$). Also, location accuracy for the detected faults were all found to be less than 3% which was the target indicated in the original SOPO. Therefore, the project is declared to achieve this milestone successfully.

Milestone 3.12.4 : Fault Detection Capability and Fault Locating Accuracy

Milestone Summary : Detection error < 10%, Fault location accuracy < 3%

This milestone refers to the successful detection of faults and accuracy of the fault location using the HIL environment. For this purpose, ComEd has provided example fault scenarios data

using the HIL for the Bronzeville system, then the proposed event detection algorithm has been performed and results are given in Table 22.

This milestone refers to the successful detection of faults and the accuracy of fault location identification. The results obtained from the HIL scenario validate that the proposed algorithm effectively detects the faulted line and accurately determines the fault location within the pre-determined thresholds.

Table 19: 196 Bus System Fault Cases

Bus From-To	Fault Type	Fault Location		
		Middle	Sending Side	Receiving Side
111-112	ABC_G	50% - 50%	21.05% - 78.94%	75.00% - 25.00%
111-112	AB_G	50% - 50%	21.05% - 78.94%	75.00% - 25.00%
111-112	A_G	50% - 50%	33.33% - 66.66%	75.00% - 25.00%
138-139	ABC_G	50% - 50%	32.43% - 67.56%	62.50% - 37.50%
138-139	A_G	50% - 50%	32.43% - 67.56%	62.50% - 37.50%
138-139	BC_G	50% - 50%	32.43% - 67.56%	62.50% - 37.50%
150-189	ABC_G	50% - 50%	23.07% - 76.92%	71.42% - 28.57%
150-189	A_G	50% - 50%	23.07% - 76.92%	71.42% - 28.57%
150-189	BC_G	50% - 50%	23.07% - 76.92%	71.42% - 28.57%
38-39	ABC_G	50% - 50%	16.66% - 83.33%	83.33% - 16.66%
38-39	BC_G	50% - 50%	16.66% - 83.33%	83.33% - 16.66%
38-39	C_G	50% - 50%	16.66% - 83.33%	83.33% - 16.66%
49-66	ABC_G	50% - 50%	83.33% - 16.66%	62.50% - 37.50%
49-66	AB_G	50% - 50%	16.66% - 83.33%	62.50% - 37.50%
49-66	A_G	50% - 50%	16.66% - 83.33%	62.50% - 37.50%
49-66	BC_G	50% - 50%	16.66% - 83.33%	62.50% - 37.50%
49-66	C_G	50% - 50%	16.66% - 83.33%	62.50% - 37.50%
53-54	ABC_G	50% - 50%	25.00% - 75.00%	71.42% - 28.57%
53-54	A_G	50% - 50%	25.00% - 75.00%	71.42% - 28.57%
53-54	BC_G	50% - 50%	25.00% - 75.00%	71.42% - 28.57%
71-72	ABC_G	50% - 50%	23.07% - 76.92%	76.92% - 23.07%
71-72	A_G	50% - 50%	23.07% - 76.92%	76.92% - 23.07%
71-72	BC_G	50% - 50%	23.07% - 76.92%	76.92% - 23.07%
81-82	ABC_G	50% - 50%	9.09% - 90.90%	80.00% - 20.00%
81-82	A_G	50% - 50%	7.40% - 92.59%	83.33% - 16.66%
81-82	BC_G	50% - 50%	9.09% - 90.90%	80.00% - 20.00%
87-97	ABC_G	50% - 50%	37.50% - 62.50%	62.50% - 37.50%
87-97	A_G	50% - 50%	37.50% - 62.50%	62.50% - 37.50%
87-97	BC_G	50% - 50%	37.50% - 62.50%	62.50% - 37.50%
91-92	ABC_G	50% - 50%	42.85% - 57.14%	66.66% - 33.33%
91-92	A_G	50% - 50%	42.85% - 57.14%	66.66% - 33.33%
91-92	BC_G	50% - 50%	42.85% - 57.14%	66.66% - 33.33%
97-98	ABC_G	50% - 50%	25.00% - 75.00%	78.94% - 21.05%
97-98	A_G	50% - 50%	25.00% - 75.00%	78.94% - 21.05%
97-98	BC_G	50% - 50%	25.00% - 75.00%	78.94% - 21.05%

Table 20: Calculated Fault Types and Locations for Cases in Table 19

Faulted Branch	Fault At Middle			Fault Close To Sending Bus			Fault Close To Receiving Bus		
	Detected Branch	Detected Fault	Detected Fault Location	Detected Branch	Detected Fault	Detected Fault Location	Detected Branch	Detected Fault	Detected Fault Location
111-112	111-112	ABC_G	49.91% - 50.09%	111-112	ABC_G	20.99% - 79.01%	111-112	ABC_G	74.90% - 25.10%
111-112	111-112	AB_G	49.83% - 50.17%	111-112	AB_G	20.92% - 79.08%	111-112	AB_G	74.80% - 25.20%
111-112	111-112	A_G	49.92% - 50.08%	111-112	A_G	33.28% - 66.72%	111-112	A_G	74.89% - 25.11%
138-139	139-138	ABC_G	50.03% - 49.97%	139-138	ABC_G	32.46% - 67.54%	139-138	ABC_G	62.52% - 37.48%
138-139	139-138	A_G	50.06% - 49.94%	139-138	A_G	32.50% - 67.50%	139-138	A_G	62.55% - 37.45%
138-139	139-138	BC_G	50.06% - 49.94%	139-138	BC_G	32.50% - 67.50%	139-138	BC_G	62.55% - 37.45%
150-189	189-150	ABC_G	50.00% - 50.00%	189-150	ABC_G	23.08% - 76.92%	189-150	ABC_G	71.43% - 28.57%
150-189	189-150	A_G	50.00% - 50.00%	189-150	A_G	23.08% - 76.92%	189-150	A_G	71.43% - 28.57%
150-189	189-150	BC_G	50.00% - 50.00%	189-150	BC_G	23.08% - 76.92%	189-150	BC_G	71.43% - 28.57%
38-39	38-39	ABC_G	49.99% - 50.01%	38-39	ABC_G	16.65% - 83.35%	38-39	ABC_G	83.32% - 16.68%
38-39	38-39	BC_G	49.76% - 50.24%	38-39	BC_G	16.45% - 83.55%	38-39	BC_G	83.08% - 16.92%
38-39	38-39	C_G	49.99% - 50.01%	38-39	C_G	16.66% - 83.34%	38-39	C_G	83.32% - 16.68%
49-66	49-66	ABC_G	50.02% - 49.98%	49-66	ABC_G	83.34% - 16.66%	49-66	ABC_G	62.52% - 37.48%
49-66	49-66	AB_G	50.05% - 49.95%	49-66	AB_G	16.75% - 83.25%	49-66	AB_G	62.54% - 37.46%
49-66	49-66	A_G	50.06% - 49.94%	49-66	A_G	16.75% - 83.25%	49-66	A_G	62.54% - 37.46%
49-66	49-66	BC_G	50.05% - 49.95%	49-66	BC_G	16.75% - 83.25%	49-66	BC_G	62.54% - 37.46%
49-66	49-66	C_G	50.06% - 49.94%	49-66	C_G	16.75% - 83.25%	49-66	C_G	62.54% - 37.46%
53-54	53-54	ABC_G	49.97% - 50.03%	53-54	ABC_G	24.98% - 75.02%	63-54	ABC_G	~% - ~%
53-54	53-54	A_G	49.93% - 50.07%	53-54	A_G	24.96% - 75.04%	63-54	A_G	~% - ~%
53-54	53-54	BC_G	49.94% - 50.06%	53-54	BC_G	24.96% - 75.04%	63-54	BC_G	~% - ~%
71-72	71-72	ABC_G	50.00% - 50.00%	71-72	ABC_G	23.08% - 76.92%	71-72	ABC_G	76.92% - 23.08%
71-72	71-72	A_G	50.00% - 50.00%	71-72	A_G	23.08% - 76.92%	71-72	A_G	76.92% - 23.08%
71-72	71-72	BC_G	50.00% - 50.00%	71-72	BC_G	23.08% - 76.92%	71-72	BC_G	76.92% - 23.08%
81-82	82-81	ABC_G	49.94% - 50.06%	82-81	ABC_G	9.08% - 90.92%	82-81	ABC_G	79.90% - 20.10%
81-82	82-81	A_G	49.84% - 50.16%	82-81	A_G	7.39% - 92.61%	82-81	A_G	83.10% - 16.90%
81-82	82-81	BC_G	49.89% - 50.11%	82-81	BC_G	9.07% - 90.93%	82-81	BC_G	79.84% - 20.16%
87-97	97-87	ABC_G	49.93% - 50.07%	101-87	ABC_G	~% - ~%	97-87	ABC_G	37.44% - 62.56%
87-97	97-87	A_G	49.83% - 50.17%	101-87	A_G	~% - ~%	97-87	A_G	37.37% - 62.63%
87-97	97-87	BC_G	49.99% - 50.01%	101-87	BC_G	~% - ~%	97-87	BC_G	37.49% - 62.51%
91-92	92-91	ABC_G	50.02% - 49.98%	92-91	ABC_G	42.88% - 57.12%	92-91	ABC_G	66.68% - 33.32%
91-92	92-91	A_G	50.05% - 49.95%	92-91	A_G	42.92% - 57.08%	92-91	A_G	66.69% - 33.31%
91-92	92-91	BC_G	50.02% - 49.98%	92-91	BC_G	42.89% - 57.11%	92-91	BC_G	66.66% - 33.34%
97-98	97-98	ABC_G	49.94% - 50.06%	97-98	ABC_G	24.98% - 75.02%	97-98	ABC_G	78.86% - 21.14%
97-98	97-98	A_G	49.86% - 50.14%	97-98	A_G	24.94% - 75.06%	97-98	A_G	78.76% - 21.24%
97-98	97-98	BC_G	50.02% - 49.98%	97-98	BC_G	25.01% - 74.99%	97-98	BC_G	78.97% - 21.03%

Table 21: 196 Bus System Detected Location Errors of Fault Cases

Fault Location	Max Location Error
Middle	<1%
Sending End	<1%
Receiving End	<1%

Table 22: The result of the proposed event detection algorithm for HIL Fault Scnerio

Fault Location	Fault Type	Fault Distance	Detected Fault Location	Detected Fault Type	Detected Fault Distance
344-345	ABC-G	50%-50%	344-345	ABC-G	50.34%-49.66%

Subtask 1.1: Data and Data Management Plan

Subtask Summary: Network data associated with the project partner's Microgrid (PPM) will be used. One-line diagram of the PPM will be acquired and saved both as a flat file containing coordinates of the nodes and connectivity information among them, as well as a graphics file showing the figure. Network data will be saved as an ASCII flat file with an accompanying information document describing individual data columns.

In order to verify the performance of the proposed graph learning technologies for capturing grid topological information, we created power flow cases under different topologies. The team has developed a modified network configuration of PPM by adding several loop structures. The different topologies are created by changing switch statuses and cutting off connections in some of the loops. The criterion is to open switches with relatively large power flows, such that before and after the topology change, the power flow profiles will be significantly different. The modified PPM configuration with the 20 switches with relatively high power flows are shown in Figs. 18. Different combinations of the statuses of these 20 switches could form a very large number of topologies for verification of the proposed graph learning methods.

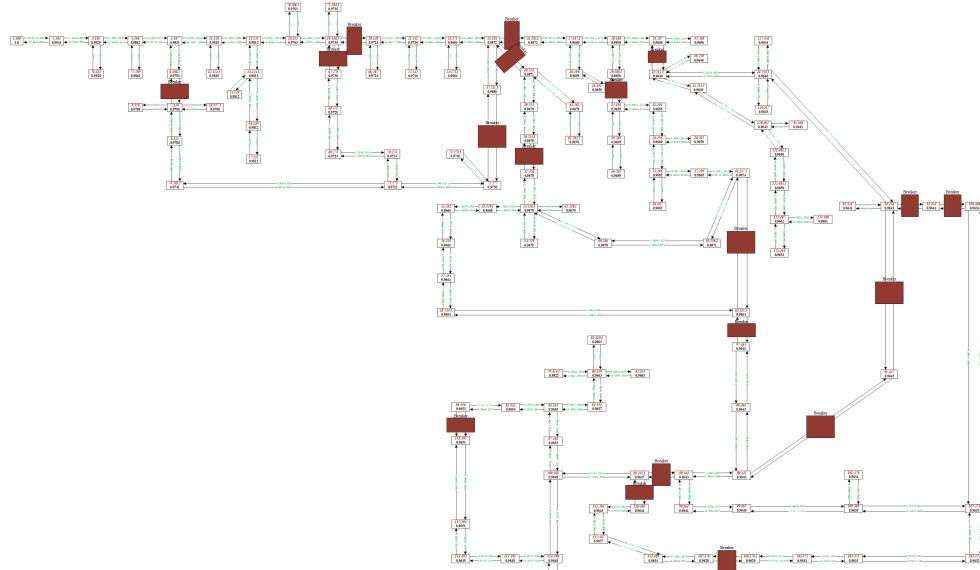


Figure 18: The positions of switches for topology changes.

On the condition that the whole system remains connected and no island is formed, we randomly disconnect several switches and generate 10 topologies. For each topology, we run power flow analysis for all the half-hourly measurements scans of transformer loads in 2018 using the automated power flow analysis program developed. Figs. 19 - 28 show the topology, node voltages, and active line flows under the 10 different topologies.

In order to further illustrate the impact of topology changes on power flow profiles, typical daily voltage curves of node 94 for the topology where lines 90-91 and 93-94 are disconnected (Topology 1) and the topology where lines 25-26, 90-91, 6-7, 19-20, 29-32, 61-73, and 88-89 are disconnected (Topology 10) are comparatively displayed in Fig. 29. Similarly, typical weekly voltage curves in the two topologies are shown in Figs. 30; typical monthly voltage curves in the

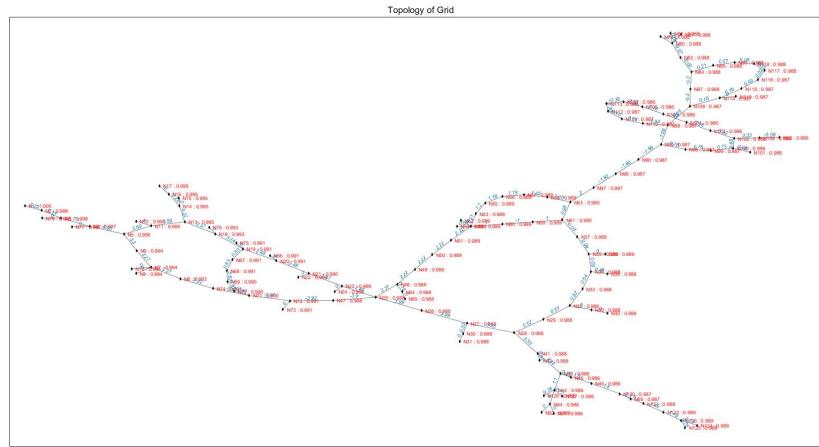


Figure 19: Power flow profile under the topology where lines 90-91 and 93-94 are disconnected.

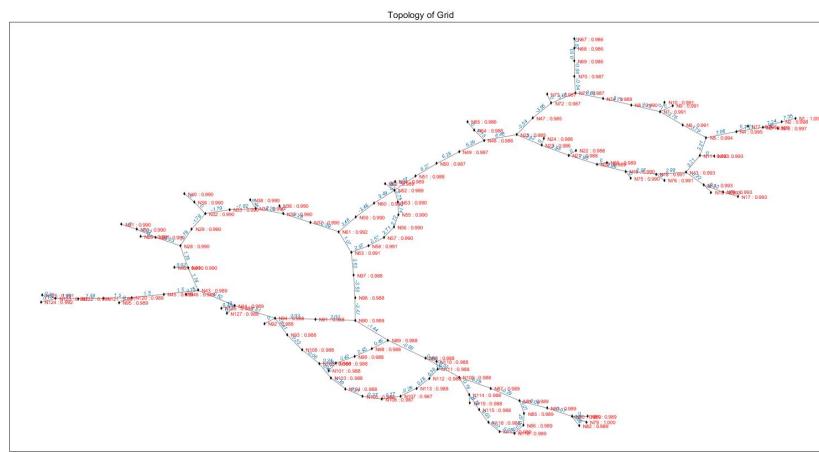


Figure 20: Power flow profile under the topology where lines 19-67 and 25-26 are disconnected.

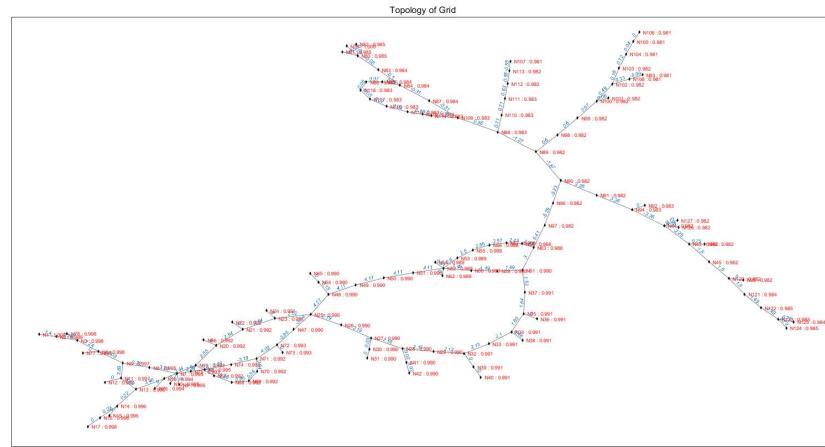


Figure 21: Power flow profile under the topology where lines 41-43, 93-94, and 106-107 are disconnected.

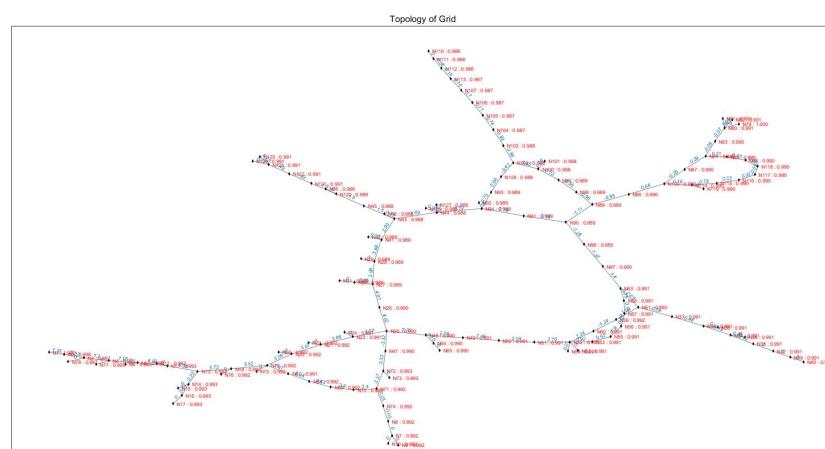


Figure 22: Power flow profile under the topology where lines 6-7, 29-32, and 88-110 are disconnected.

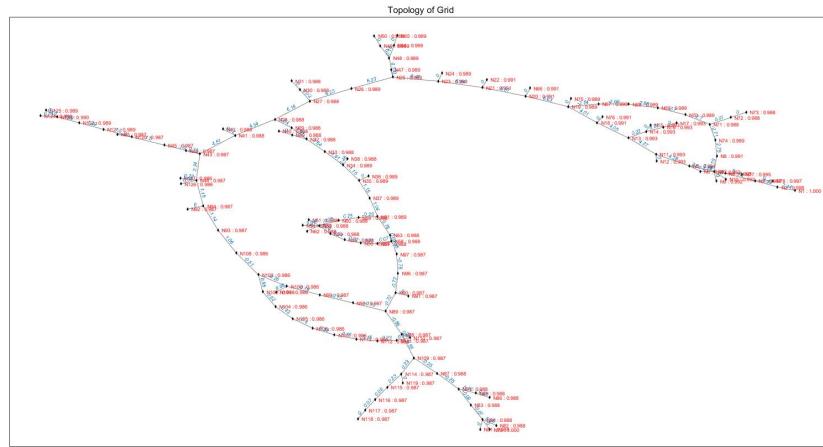


Figure 23: Power flow profile under the topology where lines 47-72, 50-51, 86-118, and 91-94 are disconnected.

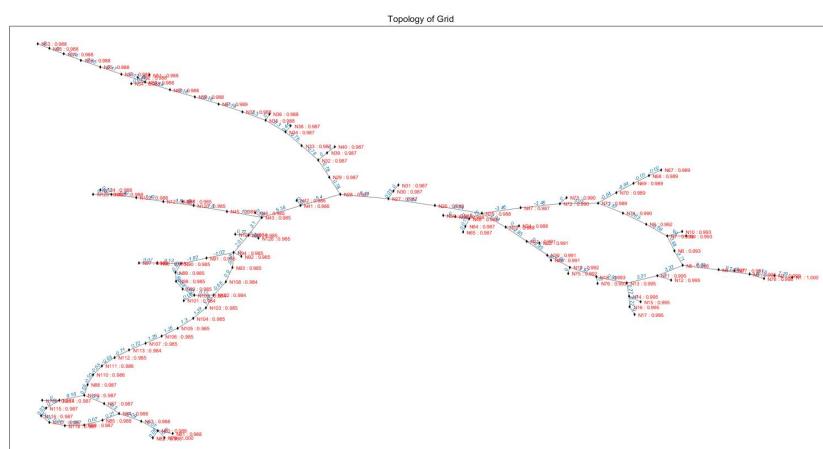


Figure 24: Power flow profile under the topology where lines 61-63, 50-51, 63-97, 19-67, and 88-89 are disconnected.

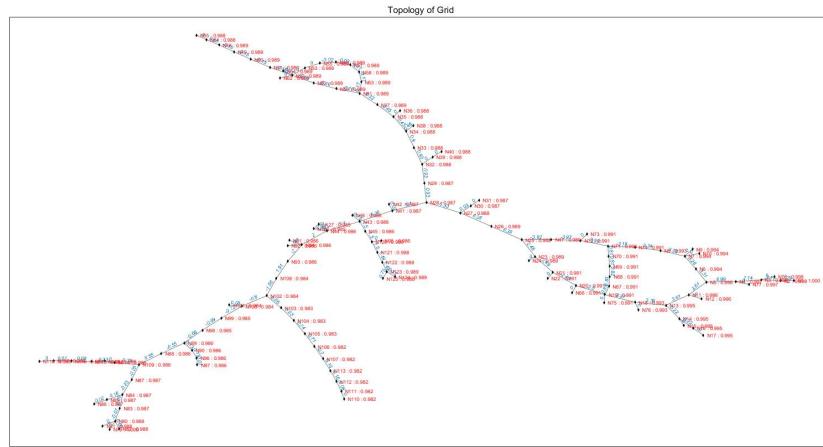


Figure 25: Power flow profile under the topology where lines 86-118, 25-48, 88-110, 90-91, and 63-97 are disconnected.

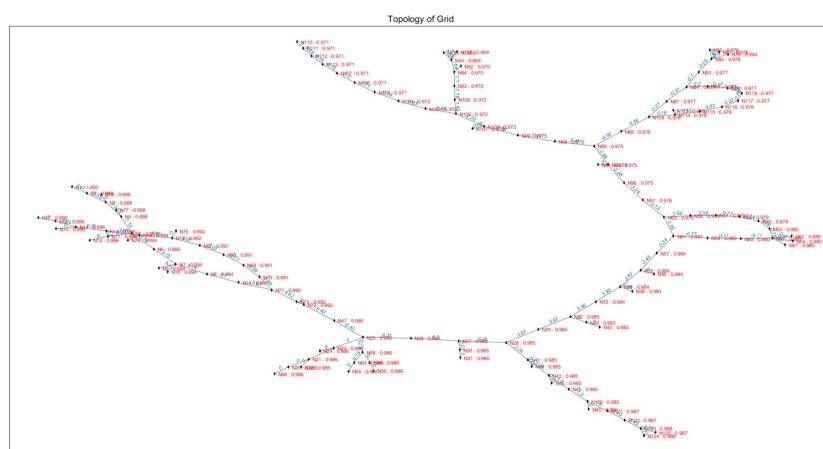


Figure 26: Power flow profile under the topology where lines 88-110, 50-51, 91-94, 19-20, and 43-44 are disconnected.

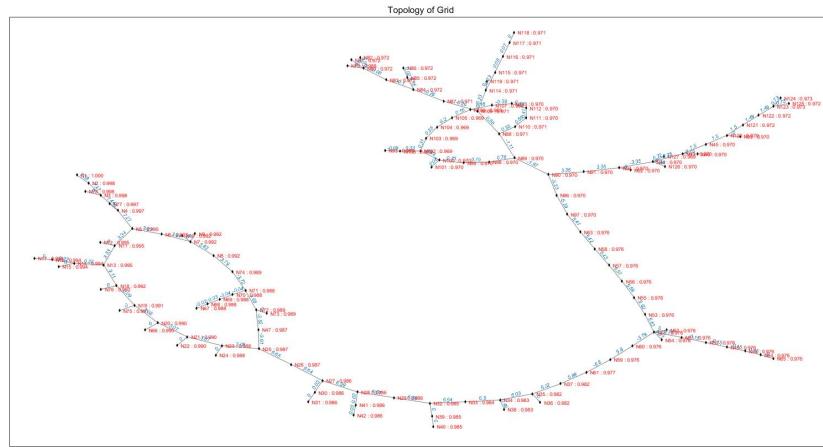


Figure 27: Power flow profile under the topology where lines 41-43, 61-63, 25-48, 86-118, 19-67, and 93-94 are disconnected.

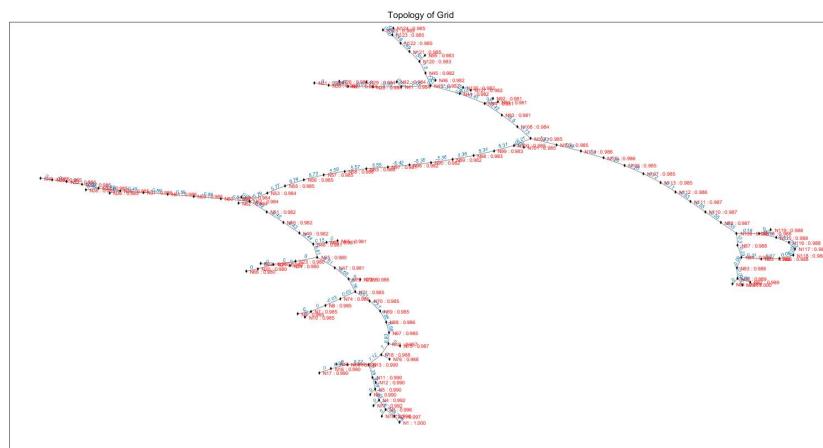


Figure 28: Power flow profile under the topology where lines 25-26, 90-91, 6-7, 19-20, 29-32, 61-73, and 88-89 are disconnected.

two topologies are shown in Fig. 31. There is a significant difference in the voltage patterns of node 94 under the two topologies. The voltage of other nodes, especially those that are electrically close to the switches that change their statuses, have similar phenomena. This indicates that the time prediction model must combine topology information to perform well.

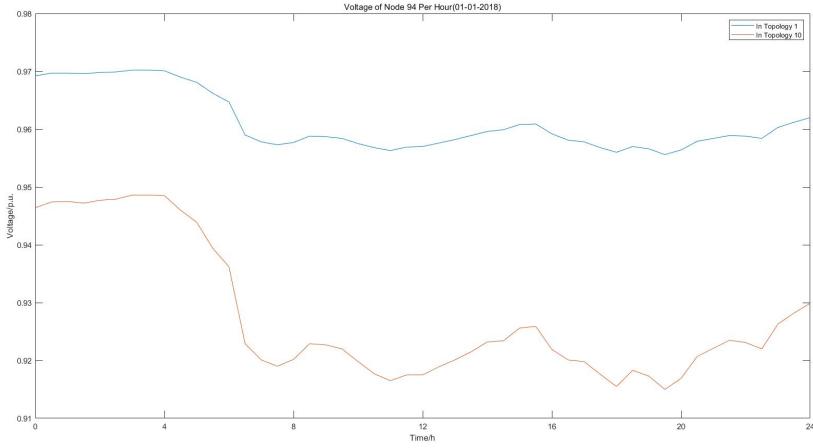


Figure 29: Typical daily voltages of Node 94 in different topologies.

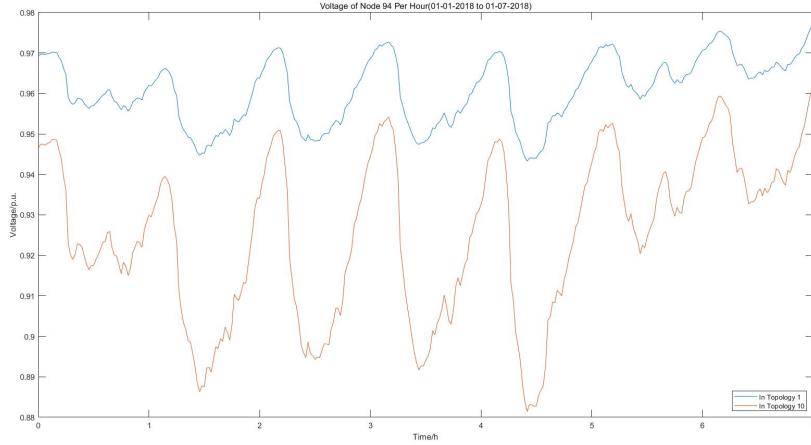


Figure 30: Typical weekly voltages of Node 94 in different topologies.

The generated power flow data under various topologies will be used to test our slow measurement prediction framework combining Graph Convolutional Network (GCN) and Long-Short-Term-Memory (LSTM) models.

Subtask 6.1: Data and Data Management Plan

Subtask Summary: Network data associated with the project partner's Microgrid (PPM) will be used. One-line diagram of the PPM will be acquired and saved both as a flat file containing coordinates of the nodes and connectivity information among them, as well as a graphics file showing the figure. Network data will be saved as an ASCII flat file with an accompanying information document describing individual data columns.

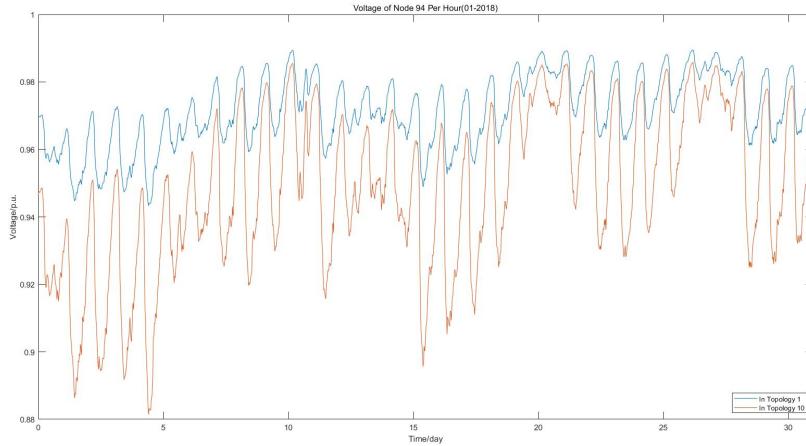


Figure 31: Typical monthly voltages of Node 94 in different topologies.

In order to further verify the performance of the proposed graph learning technologies and sparse-estimation based event detection technologies for capturing grid topological information, we created power flow cases under different topologies. The team has developed a modified network configuration of PPM by adding several loop structures. Based on this base case, we generate 5 series of topologies. Each series starts from the base case, and new topologies are generated by disconnecting lines one at a time throughout the year of 2018. A total of 3-5 lines are disconnected in each series, resulting in 3-5 different topologies in each series. Note that in the topology generation process, the whole system remains connected and no island is formed. For each topology, we run power flow analysis for all the half-hourly measurements scans of transformer loads during the time period with the topology using the automated power flow analysis program developed.

Fig. 32 shows topology series #1. The lines with red circles are the lines to be opened in this topology series. The numbers in the red circles indicate the sequence of the switching. As there are 4 line disconnections, a series of 4 different topologies are generated, and any two consecutive topologies are different by only one line. Fig. 33 shows the voltage profile of all nodes throughout year 2018. The times of line disconnections are also shown.

Fig. 34 shows topology series #2. The lines with red circles are the lines to be opened in this topology series. The numbers in the red circles indicate the sequence of the switching. As there are 4 line disconnections, a series of 4 different topologies are generated, and any two consecutive topologies are different by only one line. Fig. 35 shows the voltage profile of all nodes throughout year 2018. The times of line disconnections are also shown.

Fig. 36 shows topology series #3. The lines with red circles are the lines to be opened in this topology series. The numbers in the red circles indicate the sequence of the switching. As there are 3 line disconnections, a series of 3 different topologies are generated, and any two consecutive topologies are different by only one line. Fig. 37 shows the voltage profile of all nodes throughout year 2018. The times of line disconnections are also shown.

Fig. 38 shows topology series #4. The lines with red circles are the lines to be opened in this topology series. The numbers in the red circles indicate the sequence of the switching. As there are 4 line disconnections, a series of 5 different topologies are generated, and any two consecutive

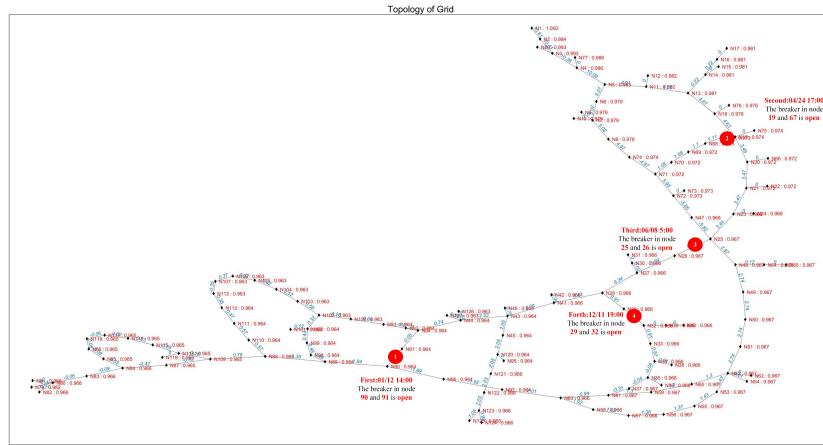


Figure 32: Topology series #1.

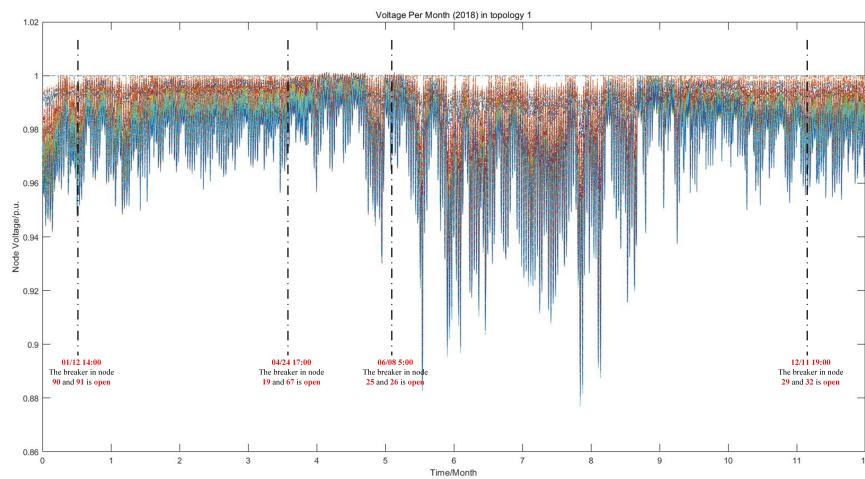


Figure 33: Node voltages of topology series #1.

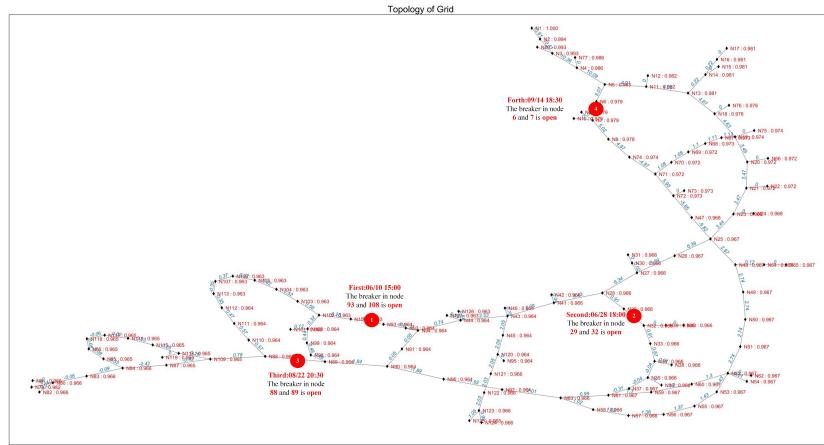


Figure 34: Topology series #2.

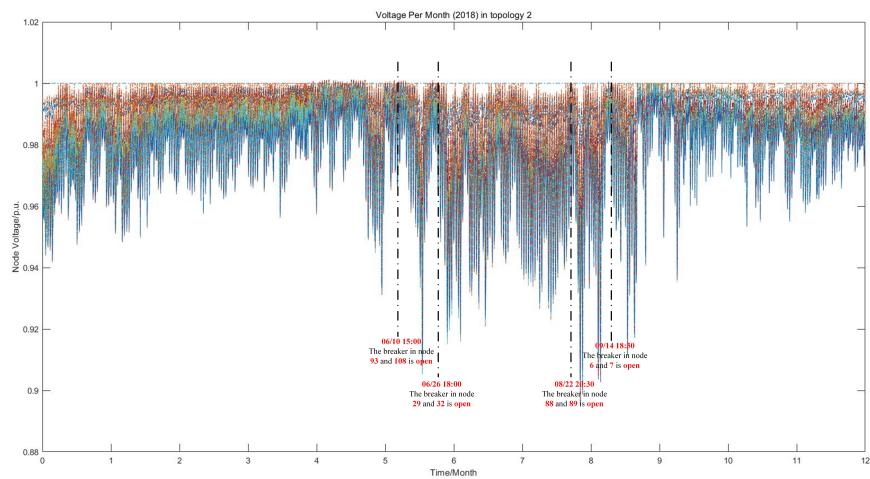


Figure 35: Node voltages of topology series #2.

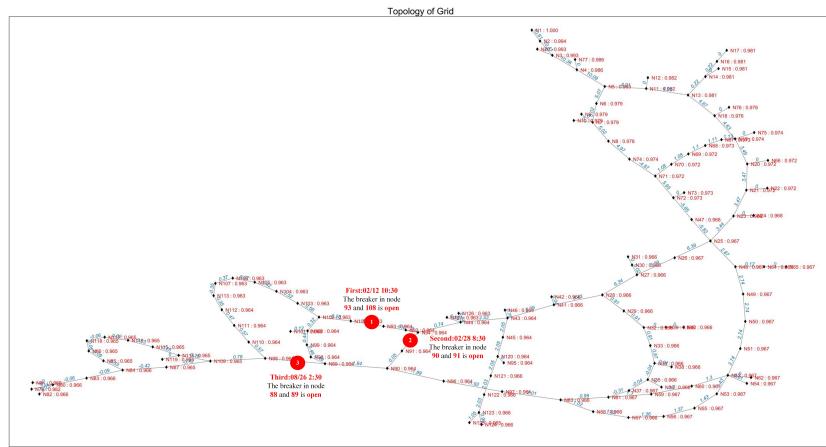


Figure 36: Topology series #3.

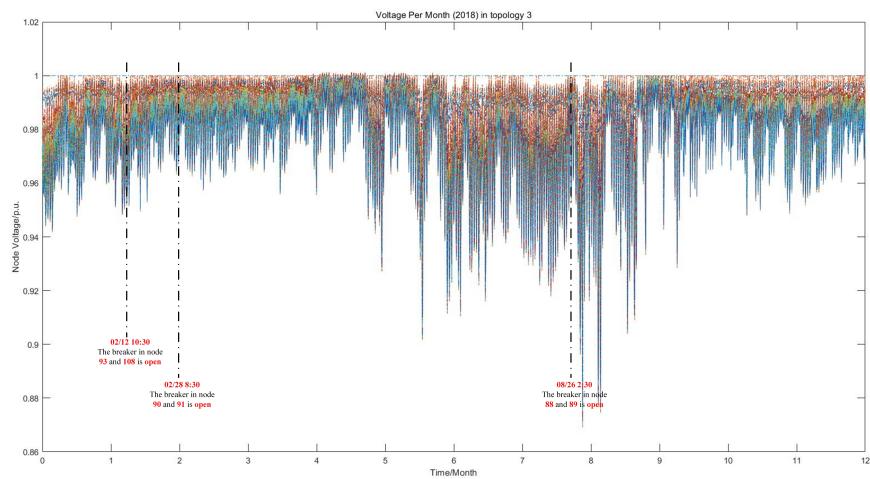


Figure 37: Node voltages of topology series #3.

topologies are different by only one line. Fig. 39 shows the voltage profile of all nodes throughout year 2018. The times of line disconnections are also shown.

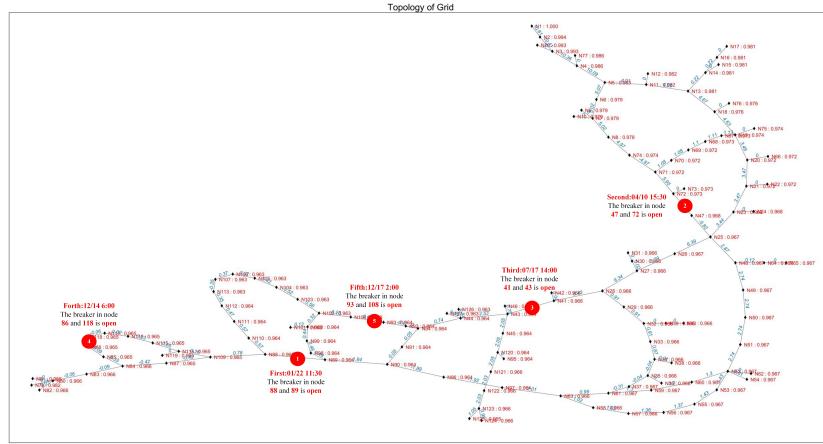


Figure 38: Topology series #4.

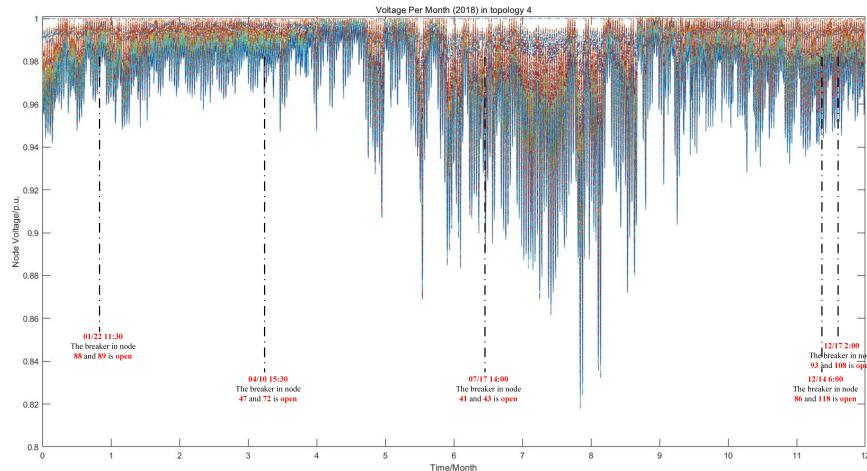


Figure 39: Node voltages of topology series #4.

Fig. 40 shows topology series #5. The lines with red circles are the lines to be opened in this topology series. The numbers in the red circles indicate the sequence of the switching. As there are 4 line disconnections, a series of 5 different topologies are generated, and any two consecutive topologies are different by only one line. Fig. 52 shows the voltage profile of all nodes throughout year 1. The times of line disconnections are also shown.

The generated power flow data under various topologies will be used to test our slow measurement prediction framework combining Graph Convolutional Network (GCN) and Long-Short-Term-Memory (LSTM) models. It will also be used to test the sparse-estimation-based line switching detection method. The team will continue working on generating more power flow data under

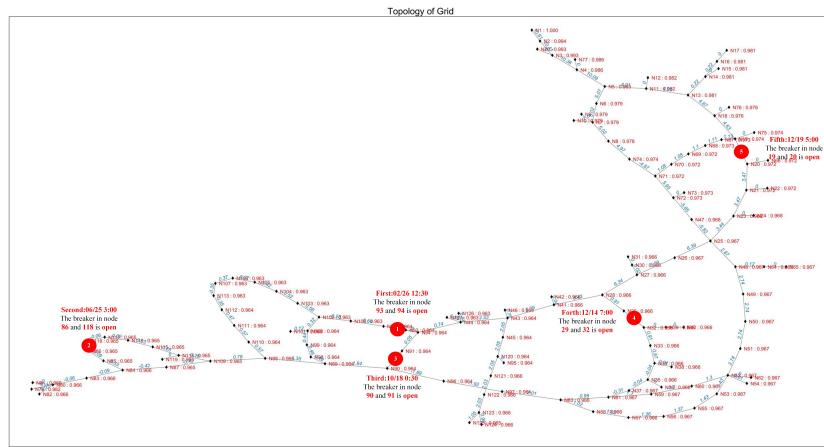


Figure 40: Topology series #5.

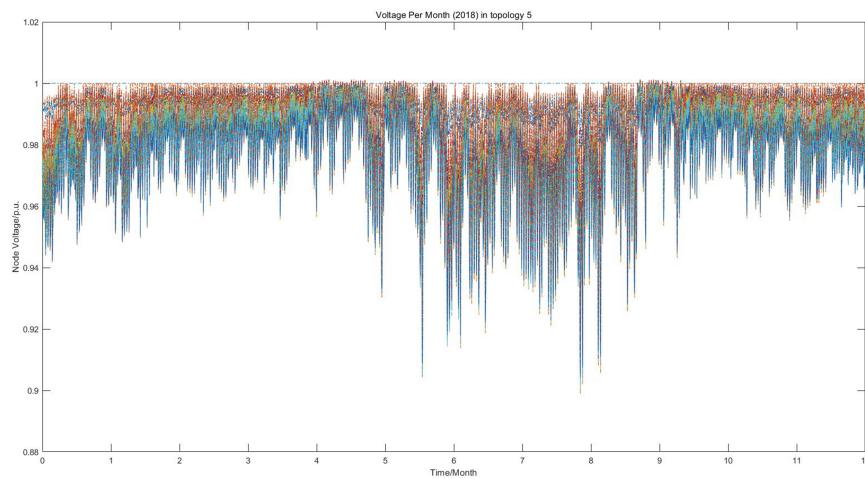


Figure 41: Node voltages of topology series #5.

different topologies to enrich the dataset and scenarios for training and testing the proposed algorithms.

Our goal is to compute the power flow every minute with various topologies and generate power flow data for Graph Convolutional Network (GCN) and Long-Short-Term-Memory (LSTM) models to test slow measurements prediction framework. In real-world power distribution systems, the statuses of network switches frequently change. In order to verify the method we proposed in the cases of various topologies, we select some switches with changed statuses to simulate the network. The steps are as follows:

1. Interpolate load data. The raw load data file from the PPM records the electrical energy consumed by 50 customers every half hour in the whole year of 2018. We convert the energy data to power data. To make measurements more refined we use the method of cubic spline interpolation to generate the power at every minute in the first three months. Thus, there are 129,601 time steps in total. To simulate the real-time power variation, we add colored noise into the interpolated power data. Figure 1 shows samples of simulated colored noise. Figure 2 shows the raw power load, power load with cubic spline interpolation, and power load with cubic spline interpolation and colored noise of the first node load in one day.

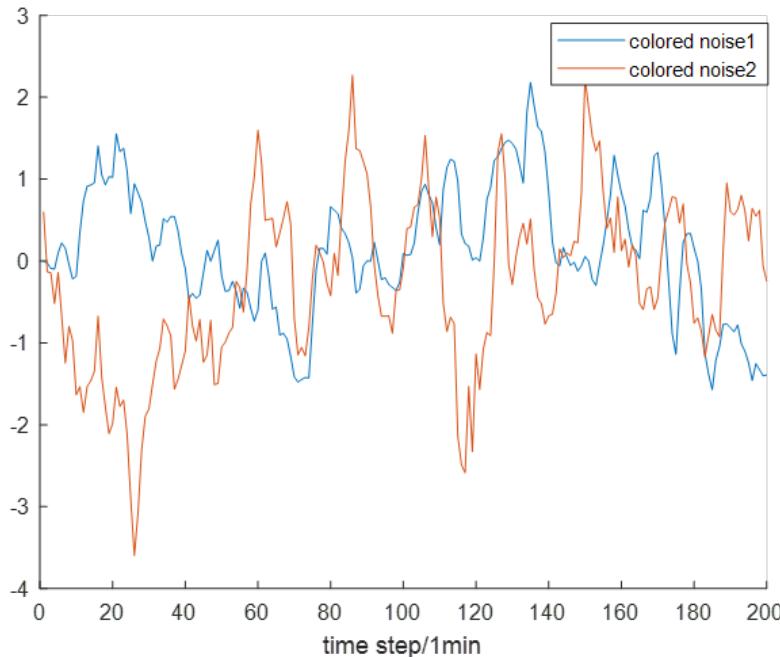


Figure 42: Generated colored noise.

2. Set different network topologies. We change the statuses of switches in the network to make different topologies. In the PPM, we previously identified 20 switches in different branches with significant impact on power flow. In the initial time step, we assume that 17 switches randomly chosen are closed and others are open. At the next randomly chosen time step, we randomly choose one switch from 20 switches and change its open or closed status, for example, if a switch chosen is open, we make it close; vice versa. We choose 10 time steps in a total of 129,601 time steps in three months for switch status change. In order to generate a larger amount of data, we repeated this

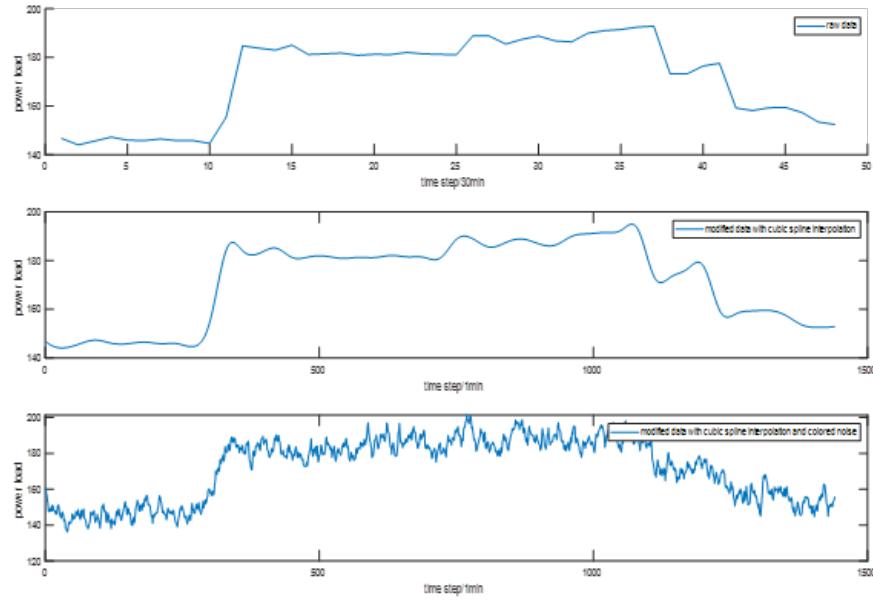


Figure 43: Raw power load, power load with cubic spline interpolation, and power load with cubic spline interpolation and colored noise on the first day.

random process 50 times and obtain power flow solutions under 500 different topologies. Figure 3 is the schematic of topology changes over time.

3. Power flow solution. After setting the network topologies at different times, we calculate the power flow at each time step based on the corresponding topology. We respectively record the voltage angle, voltage amplitude, line active power, and line reactive power in separate files. The files storing the voltage angle, voltage amplitude, line active power, and line reactive power data are shown in Figure 4, Figure 5, Figure 6, and Figure 7, respectively. The fluctuations of voltage angle, voltage amplitude, line active power, and line reactive power over all time steps in a repetition are shown in Figure 8, Figure 9, Figure 10, and Figure 11, respectively.

Finally, with the power flow solutions taken as the ground truth of system operating conditions, realistic measurement data are generated by corruption with bad data. This is to mimic data quality issues occurring under many conditions in reality, including temporary malfunctions of sensors, failure/delay of communication, or false data injection attacks. The generated corrupted data will be used by the graph-learning-based slow measurement predictor as well as the robust state estimator, while the ground truth will only be used for the evaluation of the prediction/estimation results.

In order to simulate three cases, we design four types of bad data as follows:

Case 1: Spike-shape. Randomly choose one data point and add/subtract 10%-25% of original data into/from original data.

Case 2: Square-shape. Randomly choose a time interval and make the data of all the time steps chosen to increase or decrease by either 0% or 10%-25%. Increasing or decreasing by 0% means keeping the data unchanged in this time interval.

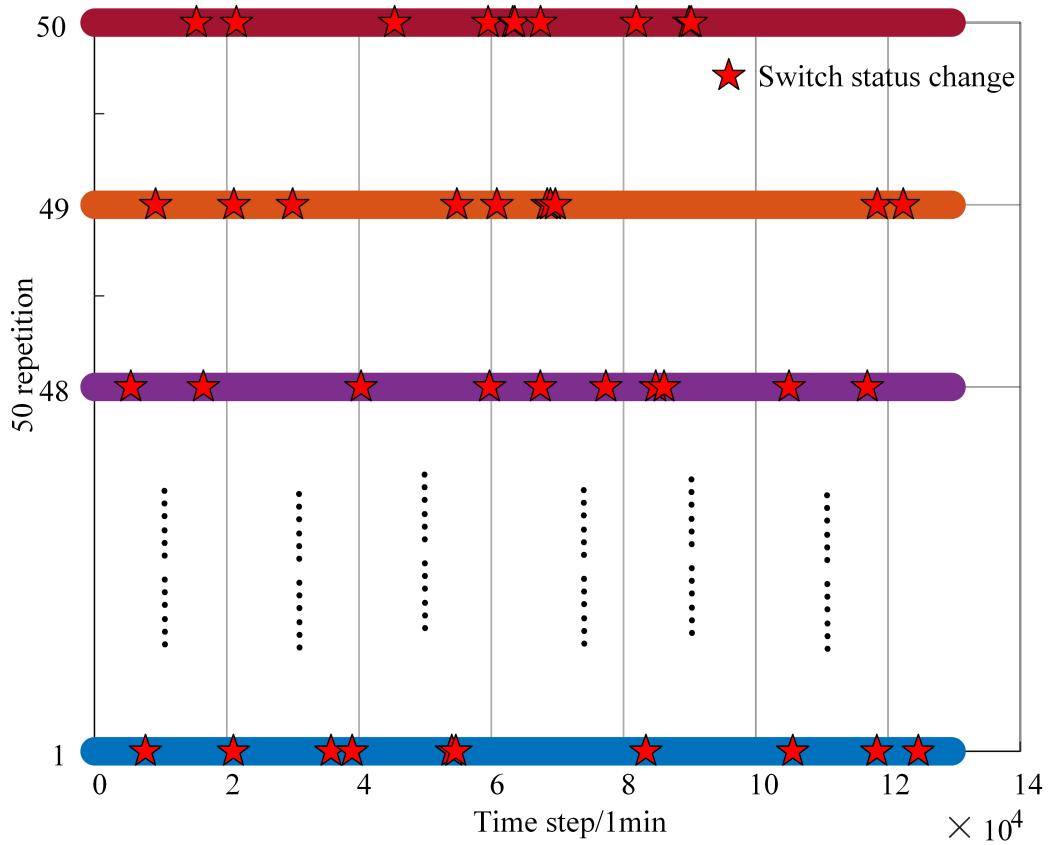


Figure 44: Switch status change schematic diagram.

	A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	15	16	17
1	Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14						
2	01/01/2018 0:00	0	-1.28	-1.61	-3.45	-4.11	-4.44	-4.47	-4.62	-4.47	-4.47	-4.29	-4.29	-4.35	-4.35	-4.35	-4.35	-4.35	-4.35	-4.35	
3	01/01/2018 0:01	0	-1.28	-1.6	-3.44	-4.09	-4.43	-4.45	-4.61	-4.45	-4.45	-4.27	-4.27	-4.33	-4.34	-4.34	-4.34	-4.34	-4.34	-4.34	
4	01/01/2018 0:02	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
5	01/01/2018 0:03	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
6	01/01/2018 0:04	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.25	-4.25	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
7	01/01/2018 0:05	0	-1.27	-1.59	-3.42	-4.07	-4.41	-4.43	-4.59	-4.43	-4.43	-4.25	-4.25	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	
8	01/01/2018 0:06	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.25	-4.25	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
9	01/01/2018 0:07	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.25	-4.25	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
10	01/01/2018 0:08	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.43	-4.59	-4.43	-4.43	-4.25	-4.25	-4.31	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
11	01/01/2018 0:09	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.25	-4.25	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
12	01/01/2018 0:10	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
13	01/01/2018 0:11	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
14	01/01/2018 0:12	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
15	01/01/2018 0:13	0	-1.28	-1.6	-3.43	-4.09	-4.43	-4.45	-4.6	-4.45	-4.45	-4.27	-4.27	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
16	01/01/2018 0:14	0	-1.28	-1.6	-3.43	-4.09	-4.42	-4.45	-4.6	-4.45	-4.45	-4.26	-4.26	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
17	01/01/2018 0:15	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.59	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
18	01/01/2018 0:16	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
19	01/01/2018 0:17	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
20	01/01/2018 0:18	0	-1.27	-1.59	-3.42	-4.07	-4.41	-4.43	-4.58	-4.43	-4.43	-4.25	-4.25	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	
21	01/01/2018 0:19	0	-1.27	-1.59	-3.41	-4.07	-4.4	-4.43	-4.58	-4.43	-4.43	-4.24	-4.24	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	
22	01/01/2018 0:20	0	-1.27	-1.59	-3.41	-4.07	-4.4	-4.43	-4.58	-4.43	-4.43	-4.24	-4.24	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	-4.31	
23	01/01/2018 0:21	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
24	01/01/2018 0:22	0	-1.27	-1.6	-3.42	-4.08	-4.41	-4.44	-4.59	-4.44	-4.44	-4.26	-4.26	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	-4.32	
25	01/01/2018 0:23	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
26	01/01/2018 0:24	0	-1.28	-1.6	-3.43	-4.08	-4.42	-4.44	-4.6	-4.44	-4.44	-4.26	-4.26	-4.32	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
27	01/01/2018 0:25	0	-1.28	-1.6	-3.43	-4.09	-4.42	-4.45	-4.6	-4.45	-4.45	-4.26	-4.26	-4.32	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
28	01/01/2018 0:26	0	-1.28	-1.6	-3.43	-4.09	-4.43	-4.45	-4.6	-4.45	-4.45	-4.27	-4.27	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	
29	01/01/2018 0:27	0	-1.28	-1.6	-3.43	-4.09	-4.42	-4.45	-4.6	-4.45	-4.45	-4.27	-4.27	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	-4.33	

Figure 45: Voltage angle data.

Figure 46: Voltage amplitude data.

	Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	01/01/2018 00:00	1	0.9951	0.9935	0.9881	0.9853	0.9819	0.9817	0.9808	0.9817	0.9817	0.9843	0.9843	0.9836	0.9836	0.9836	0.9836	0.9836
3	01/01/2018 00:01	1	0.9952	0.9937	0.9884	0.9857	0.9823	0.9821	0.9812	0.9821	0.9821	0.9847	0.9847	0.984	0.984	0.984	0.984	0.984
4	01/01/2018 00:02	1	0.9953	0.9937	0.9884	0.9859	0.9826	0.9823	0.9814	0.9823	0.9823	0.9849	0.9849	0.9843	0.9842	0.9842	0.9842	0.9842
5	01/01/2018 00:03	1	0.9953	0.9937	0.9886	0.9859	0.9825	0.9823	0.9814	0.9823	0.9823	0.9849	0.9849	0.9842	0.9842	0.9842	0.9842	0.9842
6	01/01/2018 00:04	1	0.9954	0.9938	0.9887	0.9861	0.9828	0.9825	0.9817	0.9825	0.9825	0.9851	0.9845	0.9844	0.9844	0.9844	0.9844	0.9844
7	01/01/2018 00:05	1	0.9954	0.9938	0.9888	0.9861	0.9828	0.9826	0.9817	0.9826	0.9826	0.9851	0.9851	0.9845	0.9845	0.9844	0.9844	0.9844
8	01/01/2018 00:06	1	0.9953	0.9938	0.9887	0.9861	0.9827	0.9825	0.9816	0.9825	0.9825	0.985	0.985	0.9844	0.9844	0.9844	0.9844	0.9844
9	01/01/2018 00:07	1	0.9953	0.9938	0.9887	0.9861	0.9827	0.9825	0.9816	0.9825	0.9825	0.985	0.985	0.9844	0.9844	0.9844	0.9844	0.9844
10	01/01/2018 00:08	1	0.9954	0.9938	0.9887	0.9861	0.9828	0.9825	0.9817	0.9825	0.9825	0.9851	0.9851	0.9844	0.9844	0.9844	0.9844	0.9844
11	01/01/2018 00:09	1	0.9954	0.9938	0.9887	0.9861	0.9828	0.9825	0.9817	0.9825	0.9825	0.9851	0.9851	0.9844	0.9844	0.9844	0.9844	0.9844
12	01/01/2018 00:10	1	0.9953	0.9938	0.9886	0.986	0.9826	0.9824	0.9815	0.9824	0.9824	0.9849	0.9849	0.9843	0.9843	0.9843	0.9843	0.9843
13	01/01/2018 00:11	1	0.9953	0.9938	0.9886	0.986	0.9826	0.9824	0.9815	0.9824	0.9824	0.9849	0.9849	0.9843	0.9843	0.9842	0.9842	0.9842
14	01/01/2018 00:12	1	0.9953	0.9937	0.9886	0.9859	0.9826	0.9823	0.9814	0.9823	0.9823	0.9849	0.9849	0.9842	0.9842	0.9842	0.9842	0.9842
15	01/01/2018 00:13	1	0.9952	0.9937	0.9884	0.9857	0.9824	0.9821	0.9812	0.9821	0.9821	0.9847	0.9847	0.984	0.984	0.984	0.984	0.984
16	01/01/2018 00:14	1	0.9953	0.9937	0.9885	0.9858	0.9825	0.9822	0.9813	0.9822	0.9822	0.9848	0.9848	0.9841	0.9841	0.9841	0.9841	0.9841
17	01/01/2018 00:15	1	0.9953	0.9938	0.9886	0.986	0.9826	0.9824	0.9815	0.9824	0.9824	0.9849	0.9849	0.9843	0.9843	0.9842	0.9842	0.9842
18	01/01/2018 00:16	1	0.9953	0.9938	0.9886	0.986	0.9826	0.9824	0.9815	0.9825	0.9825	0.9851	0.9851	0.9844	0.9844	0.9844	0.9844	0.9844
19	01/01/2018 00:17	1	0.9953	0.9937	0.9886	0.9859	0.9826	0.9823	0.9814	0.9823	0.9823	0.9849	0.9849	0.9842	0.9842	0.9842	0.9842	0.9842
20	01/01/2018 00:18	1	0.9954	0.9938	0.9886	0.9862	0.9829	0.9826	0.9817	0.9826	0.9826	0.9852	0.9852	0.9844	0.9845	0.9845	0.9845	0.9845
21	01/01/2018 00:19	1	0.9954	0.9939	0.9887	0.9863	0.983	0.9827	0.9819	0.9827	0.9827	0.9852	0.9852	0.9846	0.9846	0.9846	0.9846	0.9846
22	01/01/2018 00:20	1	0.9954	0.9939	0.9886	0.9863	0.983	0.9827	0.9819	0.9827	0.9827	0.9853	0.9853	0.9847	0.9846	0.9846	0.9846	0.9846
23	01/01/2018 00:21	1	0.9953	0.9938	0.9886	0.986	0.9826	0.9824	0.9815	0.9824	0.9824	0.9849	0.9849	0.9843	0.9843	0.9842	0.9842	0.9842
24	01/01/2018 00:22	1	0.9953	0.9938	0.9887	0.986	0.9827	0.9824	0.9816	0.9824	0.9824	0.985	0.985	0.9843	0.9843	0.9843	0.9843	0.9843
25	01/01/2018 00:23	1	0.9953	0.9937	0.9885	0.9859	0.9825	0.9822	0.9814	0.9822	0.9822	0.9848	0.9848	0.9842	0.9841	0.9841	0.9841	0.9841
26	01/01/2018 00:24	1	0.9953	0.9937	0.9885	0.9859	0.9825	0.9823	0.9814	0.9823	0.9823	0.9848	0.9848	0.9842	0.9842	0.9842	0.9842	0.9842
27	01/01/2018 00:25	1	0.9953	0.9937	0.9885	0.9858	0.9824	0.9822	0.9813	0.9822	0.9822	0.9847	0.9847	0.9841	0.9841	0.9841	0.9841	0.9841
28	01/01/2018 00:26	1	0.9952	0.9936	0.9884	0.9857	0.9823	0.9821	0.9812	0.9821	0.9821	0.9846	0.9846	0.984	0.984	0.984	0.984	0.984
29	01/01/2018 00:27	1	0.9952	0.9936	0.9884	0.9857	0.9823	0.9821	0.9812	0.9821	0.9821	0.9846	0.9846	0.984	0.984	0.984	0.984	0.984

Figure 46: Voltage amplitude data.

Figure 47: Line active power data.

	Branch number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	01/01/2018 00:00	9.498	9.338	-9.338	9.294	0	-9.294	9.072	0	-9.072	4.55	4.434	-8.993	4.509	-4.509	4.505	0	0
3	01/01/2018 00:01	9.433	9.275	-9.275	9.231	0	-9.231	9.012	0	-9.011	4.513	4.411	-8.924	4.472	-4.472	4.468	0	0
4	01/01/2018 00:02	9.395	9.238	-9.238	9.194	0	-9.194	8.976	0	-8.975	4.491	4.398	-8.888	4.45	-4.45	4.446	0	0
5	01/01/2018 00:03	9.397	9.24	-9.24	9.196	0	-9.196	8.978	0	-8.978	4.492	4.398	-8.89	4.451	-4.451	4.448	0	0
6	01/01/2018 00:04	9.361	9.205	-9.205	9.161	0	-9.161	8.944	0	-8.944	4.471	4.366	-8.857	4.43	-4.43	4.427	0	0
7	01/01/2018 00:05	9.353	9.197	-9.197	9.154	0	-9.154	8.936	0	-8.936	4.467	4.363	-8.85	4.426	-4.426	4.423	0	0
8	01/01/2018 00:06	9.367	9.21	-9.21	9.166	0	-9.166	8.949	0	-8.949	4.475	4.388	-8.862	4.434	-4.434	4.43	0	0
9	01/01/2018 00:07	9.365	9.208	-9.208	9.165	0	-9.165	8.947	0	-8.947	4.474	4.387	-8.86	4.433	-4.433	4.43	0	0
10	01/01/2018 00:08	9.36	9.203	-9.203	9.16	0	-9.16	8.942	0	-8.942	4.471	4.385	-8.855	4.43	-4.43	4.426	0	0
11	01/01/2018 00:09	9.363	9.206	-9.206	9.163	0	-9.163	8.945	0	-8.945	4.472	4.386	-8.858	4.431	-4.431	4.428	0	0
12	01/01/2018 00:10	9.381	9.224	-9.224	9.181	0	-9.181	8.963	0	-8.963	4.483	4.393	-8.876	4.443	-4.443	4.439	0	0
13	01/01/2018 00:11	9.389	9.232	-9.232	9.188	0	-9.188	8.97	0	-8.97	4.487	4.396	-8.882	4.446	-4.446	4.443	0	0
14	01/01/2018 00:12	9.395	9.237	-9.237	9.194	0	-9.194	8.975	0	-8.975	4.491	4.397	-8.888	4.445	-4.445	4.447	0	0
15	01/01/2018 00:13	9.426	9.268	-9.268	9.224	0	-9.224	9.005	0	-9.005	4.509	4.408	-8.917	4.466	-4.466	4.465	0	0
16	01/01/2018 00:14	9.41	9.253	-9.253	9.209	0	-9.209	8.99	0	-8.99	4.463	4.38	-8.843	4.422	-4.422	4.419	0	0
17	01/01/2018 00:15	9.387	9.23	-9.23	9.186	0	-9.186	8.968	0	-8.968	4.486	4.395	-8.88	4.445	-4.445	4.442	0	0
18	01/01/2018 00:16	9.385	9.228	-9.228	9.184	0	-9.184	8.966	0	-8.966	4.485	4.394	-8.879	4.444	-4.444	4.441	0	0
19	01/01/2018 00:17	9.393	9.235	-9.235	9.192	0	-9.192	8.973	0	-8.973	4.49	4.397	-8.886	4.449	-4.449	4.445	0	0
20	01/01/2018 00:18	9.347	9.19	-9.19	9.147	0	-9.147	8.93	0	-8.93	4.463	4.38	-8.843	4.422	-4.422	4.419	0	0
21	01/01/2018 00:19	9.33	9.174	-9.174	9.131	0	-9.131	8.915	0	-8.915	4.454	4.375	-8.828	4.413	-4.413	4.41	0	0
22	01/01/2018 00:20	9.324	9.169	-9.169	9.125	0	-9.125	8.909	0	-8.909	4.45	4.373	-8.823	4.41	-4.41	4.406	0	0
23	01/01/2018 00:21	9.382	9.225	-9.225	9.182	0	-9.182	8.964	0	-8.964	4.484	4.393	-8.877	4.443	-4.443	4.44	0	0
24	01/01/2018 00:22	9.375	9.218	-9.218	9.174	0	-9.174	8.957	0	-8.957	4.48	4.39	-8.869	4.439	-4.439	4.435	0	0
25	01/01/2018 00:23	9.401	9.244	-9.244	9.2	0	-9.2	8.982	0	-8.982	4.							

Brunch number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
01/01/2018 00:00	-4.969	-5.155	5.155	-5.199	0	5.199	-5.465	0	5.465	-4.422	-1.132	5.554	-4.433	4.433	-4.434	0	0
01/01/2018 00:01	-4.989	-5.174	5.174	-5.217	0	5.217	-5.481	0	5.481	-4.423	-1.146	5.569	-4.434	4.434	-4.435	0	0
01/01/2018 00:02	-5.001	-5.184	5.184	-5.228	0	5.228	-5.49	0	5.49	-4.424	-1.154	5.578	-4.435	4.435	-4.436	0	0
01/01/2018 00:03	-4.997	-5.18	5.18	-5.224	0	5.224	-5.486	0	5.486	-4.422	-1.152	5.574	-4.433	4.433	-4.434	0	0
01/01/2018 00:04	-5.01	-5.193	5.193	-5.237	0	5.237	-5.498	0	5.498	-4.424	-1.16	5.585	-4.435	4.435	-4.436	0	0
01/01/2018 00:05	-5.011	-5.193	5.193	-5.237	0	5.237	-5.497	0	5.497	-4.423	-1.161	5.584	-4.434	4.434	-4.435	0	0
01/01/2018 00:06	-5.006	-5.189	5.189	-5.233	0	5.233	-5.494	0	5.494	-4.423	-1.158	5.581	-4.434	4.434	-4.435	0	0
01/01/2018 00:07	-5.005	-5.188	5.188	-5.232	0	5.232	-5.493	0	5.493	-4.422	-1.158	5.58	-4.433	4.433	-4.434	0	0
01/01/2018 00:08	-5.008	-5.19	5.19	-5.234	0	5.234	-5.495	0	5.495	-4.423	-1.159	5.582	-4.434	4.434	-4.434	0	0
01/01/2018 00:09	-5.009	-5.192	5.192	-5.235	0	5.235	-5.496	0	5.496	-4.424	-1.159	5.583	-4.433	4.435	-4.436	0	0
01/01/2018 00:10	-5	-5.183	5.183	-5.227	0	5.227	-5.489	0	5.489	-4.421	-1.154	5.576	-4.432	4.432	-4.433	0	0
01/01/2018 00:11	-5.002	-5.185	5.185	-5.229	0	5.229	-5.491	0	5.491	-4.424	-1.154	5.578	-4.435	4.435	-4.435	0	0
01/01/2018 00:12	-4.997	-5.181	5.181	-5.225	0	5.225	-5.487	0	5.487	-4.422	-1.152	5.574	-4.433	4.433	-4.434	0	0
01/01/2018 00:13	-4.988	-5.173	5.173	-5.217	0	5.217	-5.48	0	5.48	-4.422	-1.146	5.568	-4.433	4.433	-4.434	0	0
01/01/2018 00:14	-4.992	-5.176	5.176	-5.22	0	5.22	-5.483	0	5.483	-4.421	-1.149	5.57	-4.433	4.432	-4.433	0	0
01/01/2018 00:15	-5	-5.183	5.183	-5.227	0	5.227	-5.489	0	5.489	-4.422	-1.154	5.576	-4.433	4.433	-4.434	0	0
01/01/2018 00:16	-4.999	-5.182	5.182	-5.225	0	5.226	-5.488	0	5.488	-4.421	-1.154	5.575	-4.432	4.432	-4.433	0	0
01/01/2018 00:17	-4.998	-5.181	5.181	-5.225	0	5.225	-5.487	0	5.487	-4.422	-1.152	5.574	-4.433	4.433	-4.434	0	0
01/01/2018 00:18	-5.012	-5.194	5.194	-5.237	0	5.237	-5.498	0	5.498	-4.423	-1.162	5.585	-4.434	4.434	-4.435	0	0
01/01/2018 00:19	-5.017	-5.198	5.198	-5.242	0	5.242	-5.501	0	5.501	-4.423	-1.165	5.588	-4.434	4.434	-4.435	0	0
01/01/2018 00:20	-5.017	-5.199	5.199	-5.242	0	5.242	-5.502	0	5.502	-4.422	-1.166	5.588	-4.433	4.433	-4.434	0	0
01/01/2018 00:21	-4.998	-5.181	5.181	-5.225	0	5.225	-5.486	0	5.486	-4.42	-1.153	5.574	-4.431	4.431	-4.432	0	0
01/01/2018 00:22	-5.002	-5.185	5.185	-5.228	0	5.228	-5.49	0	5.49	-4.421	-1.156	5.577	-4.432	4.432	-4.433	0	0
01/01/2018 00:23	-4.993	-5.176	5.176	-5.22	0	5.22	-5.482	0	5.482	-4.42	-1.15	5.57	-4.431	4.431	-4.432	0	0
01/01/2018 00:24	-4.994	-5.178	5.178	-5.221	0	5.221	-5.484	0	5.484	-4.421	-1.15	5.571	-4.432	4.432	-4.432	0	0
01/01/2018 00:25	-4.987	-5.17	5.17	-5.214	0	5.214	-5.477	0	5.477	-4.418	-1.146	5.564	-4.429	4.429	-4.43	0	0
01/01/2018 00:26	-4.981	-5.165	5.165	-5.209	0	5.209	-5.473	0	5.473	-4.418	-1.143	5.556	-4.429	4.429	-4.44	0	0
01/01/2018 00:27	-4.981	-5.165	5.165	-5.209	0	5.209	-5.472	0	5.472	-4.417	-1.143	5.556	-4.428	4.428	-4.429	0	0

Figure 48: Line reactive power data.

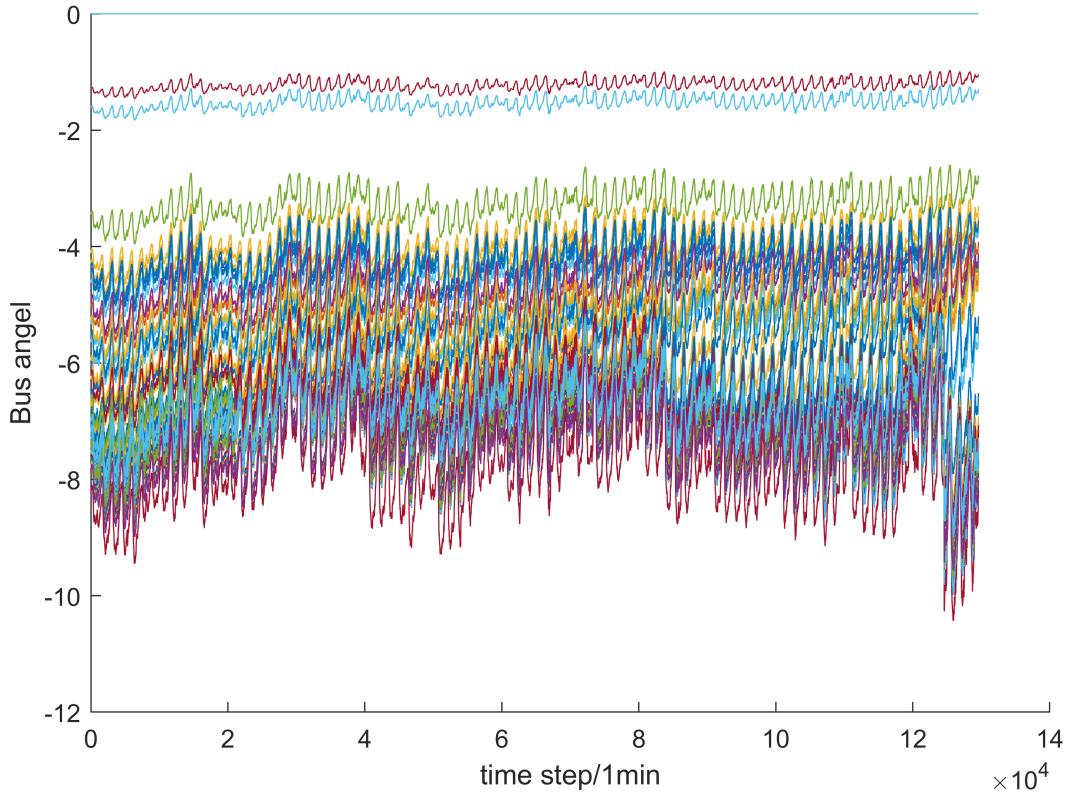
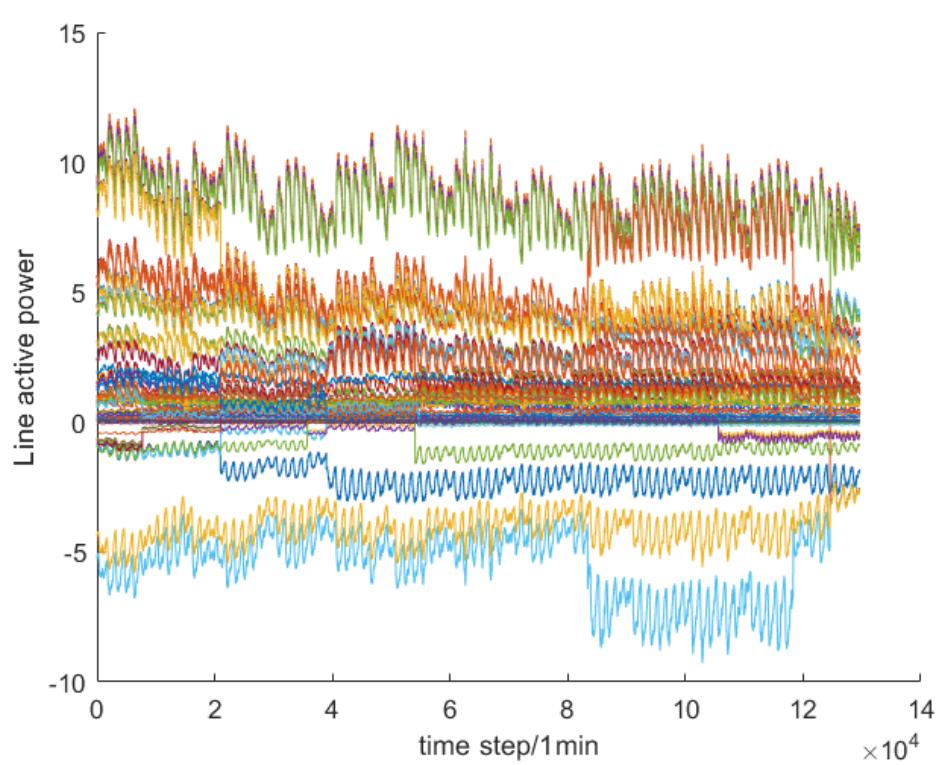
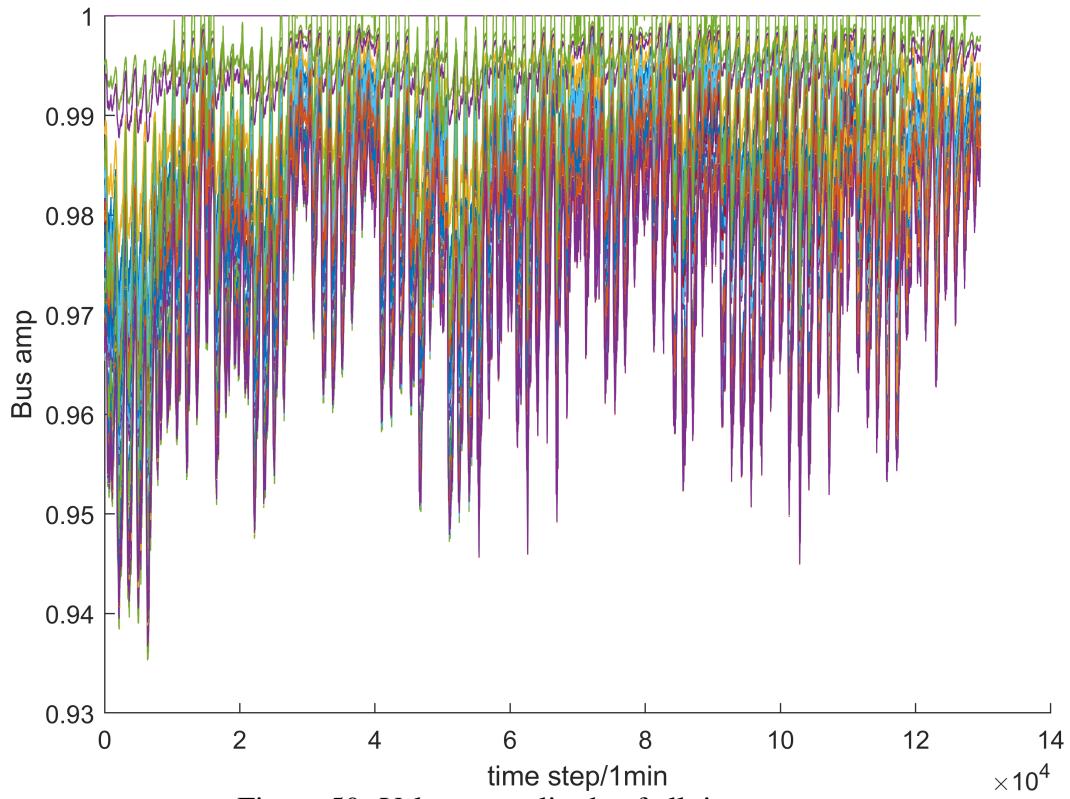


Figure 49: Voltage angle of all time steps.



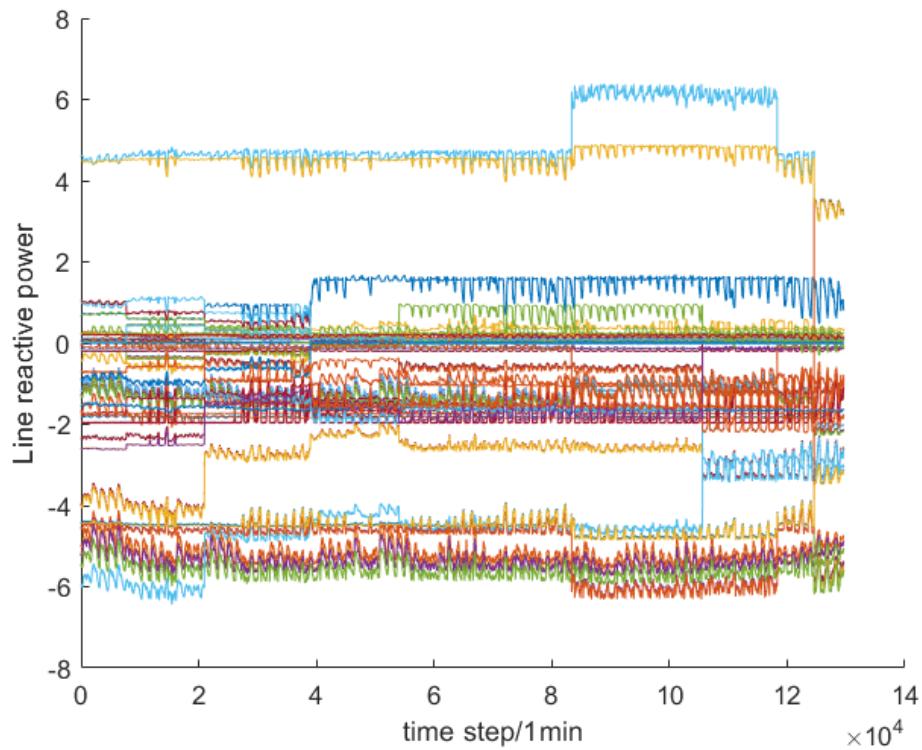


Figure 52: Line reactive power of all time steps.

Case 3: Triangular or inverted triangle shape. Choose a time interval randomly and make the data curve climb and descend as a ramp. The length of the time interval is determined by chi-square distribution.

Case 4: Trapezoidal or inverted trapezoidal shape. Choose a time interval randomly and make the data curve ramp up. Then, keep the data curve horizontal for small time intervals. Finally, make the data curve descend like a ramp. The length of the time interval is determined by chi-square distribution.

Case 1, case 2, case 3, and case 4 account for 50%, 30%, 10%, and 10% of all the bad data respectively. All the data of the node or branch have been added into bad data. Figs. 62 ,63,64,65 and 66 show examples of node active power injections, node reactive power injections, node voltage magnitudes, line active power lows, and line reactive power lows with bad data, respectively.

Subtask 11.1: Data and Data Management Plan

Subtask Summary: In this task, the team will identify all available data sets from the PPM, including both the ones from available historical data base, as well as real time data. The PPM has a range of measuring devices and technologies, including SCADA, AMI data, PMU measurements, as well as power quality measurements. It also has a range of Distribution Energy Resources (DER) with corresponding measurements. In addition, the PPM is well modeled and various scenarios can be simulated utilizing project participant's Grid Integration and Technology lab, Hardware in the loop (HIL) setup.

We undertook two efforts to create the datasets and test scenarios needed to test the proposed

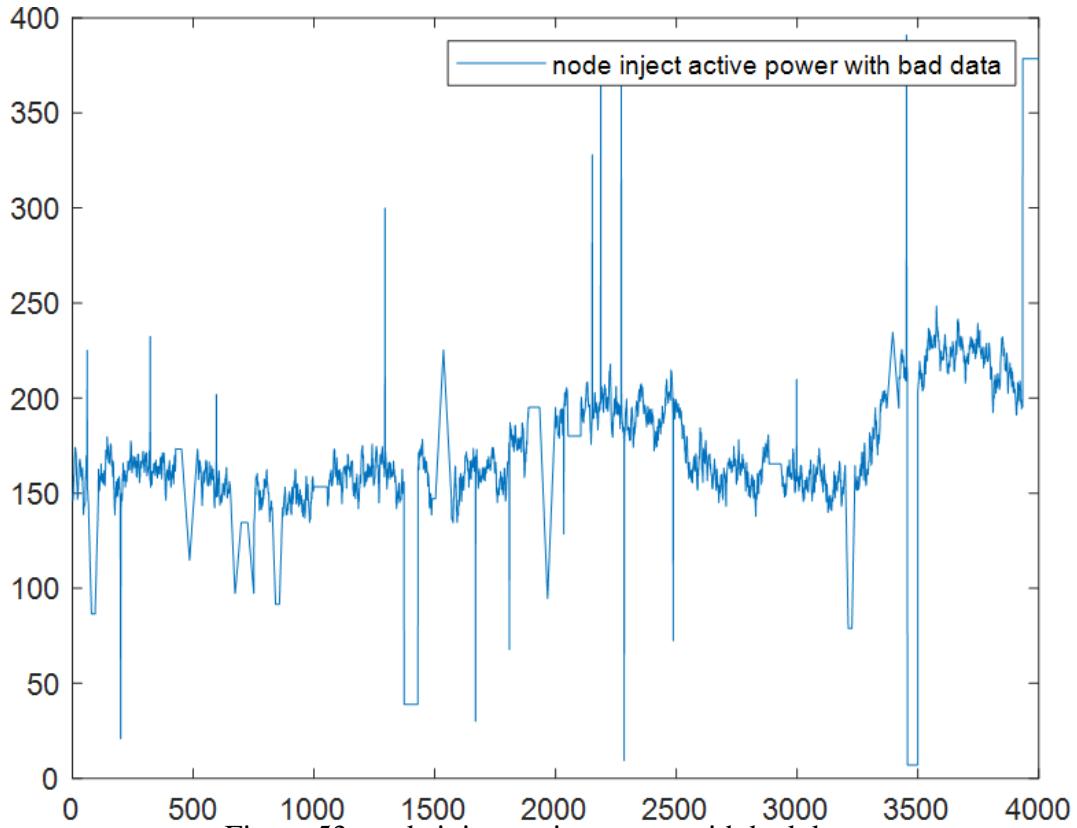


Figure 53: node inject active power with bad data

ML models under more realistic conditions.

1. Extraction of data associated with topology changes. In the previous testing, we used a dataset with temporally sparse topology changes, i.e., 10 topology changes in 3 months. The lack of topology changes leads to the inability to train and test the ML models with a sufficient number of samples. We extract the time-series data surrounding topology changes and form a dataset with a large number of topology changes within a limited volume of data. Specifically, we extract three hours' time-series data of node voltages, node injected active and reactive powers, and line active and reactive powers surrounding each topology change (line switching), and reassembled them into one dataset with 500 topology changes in 1500 hour's time-series data. These selected data will be used to train our model to show that our proposed method can accurately predict the node injected active and reactive power when the network topology changes.

First, we localized all the time steps when the topology changes. Then the data in the hour that encompasses each time step with a topology change, as well as in the hour before and the hour after, are extracted, forming a total of three hours of data segment. If the time step of a topology change is in the first or last hour of a simulation (3 months' data), then only two hours of data will be selected. Each simulation (3 months) has a total of 10 times of topology changes, so there are close to $60*3*10=1800$ time steps of data. We have conducted a total of 50 such simulations, so the data used for training has a total of nearly 90,000 time steps.

2. Measurement selection and observability analysis. In the previous studies, we randomly selected a subset of lines as the ones observed by FS measurements, and the remaining ones are

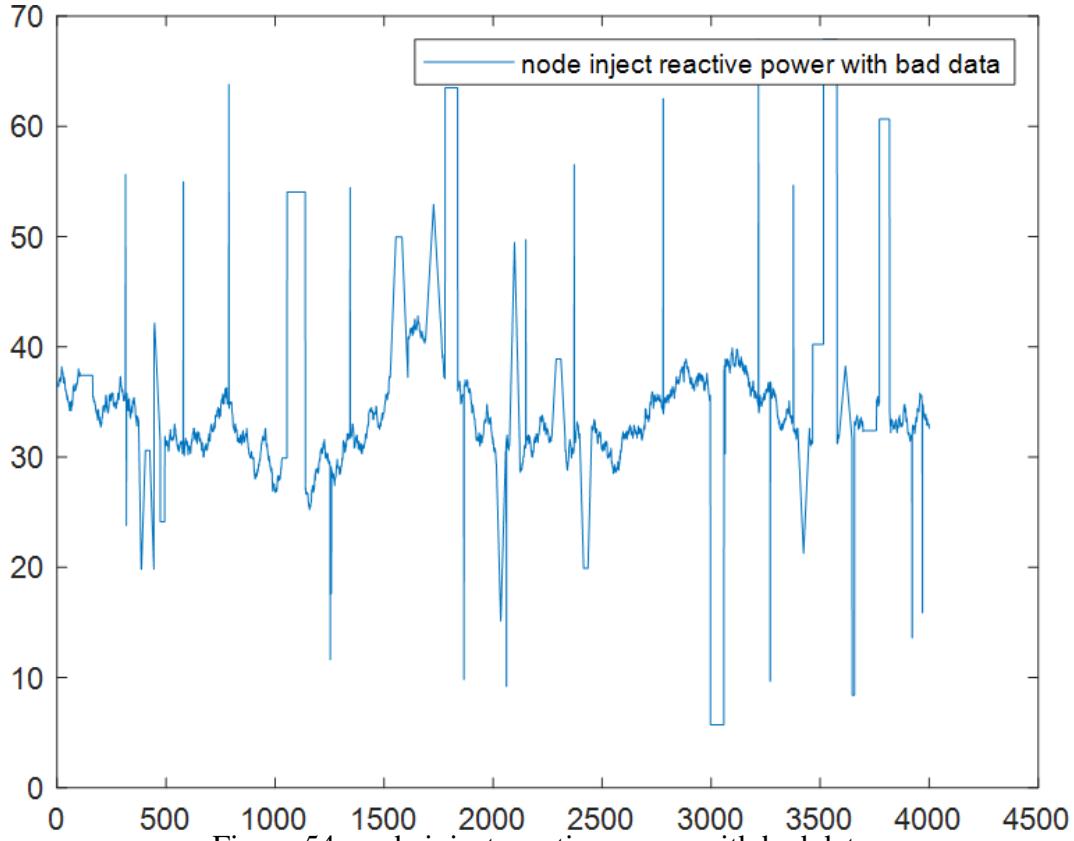


Figure 54: node inject reactive power with bad data

considered to be unobservable and their power flows need to be estimated by the proposed ML model. Such a random selection does not accurately reflect the observability of the PPM, as the power flows of some lines without direct measurements can be inferred based on zero injection measurements. To better reflect the practical operational conditions, we created various scenarios of FS measurements (line flow measurements) and identified the observability statuses of line flows according to power system observability analysis and the locations of zero-injection buses.

First, we selected some lines to place power flow measurements and combined them with the known locations of zero-injection buses to perform an observability analysis. It can infer other observable lines that are not directly measured by flow measurements. Consequently, the lines that are truly unobservable can be identified, whose power flows are set as variables to be estimated by the ML model. To test the ML under different measurement levels, we created measurement configurations by incremental addition of flow measurements. Table 23 shows eight measurement configurations with different measurement levels. Figure 58, Figure 82, Figure 60 and Figure 83 show the case of 10%, 30%, 50% and 70% unobservable lines respectively.

We created datasets with corrupted data based on the extraction of data associated with topology changes. Realistic measurement data are generated by corruption with bad data. This is to mimic data quality issues occurring under many conditions in reality, including temporary malfunctions of sensors, failure/delay of communication, or false data injection attacks. The generated corrupted data will be used by the graph-learning-based slow measurement predictor as well as the robust state estimator, while the ground truth will only be used for the evaluation of the

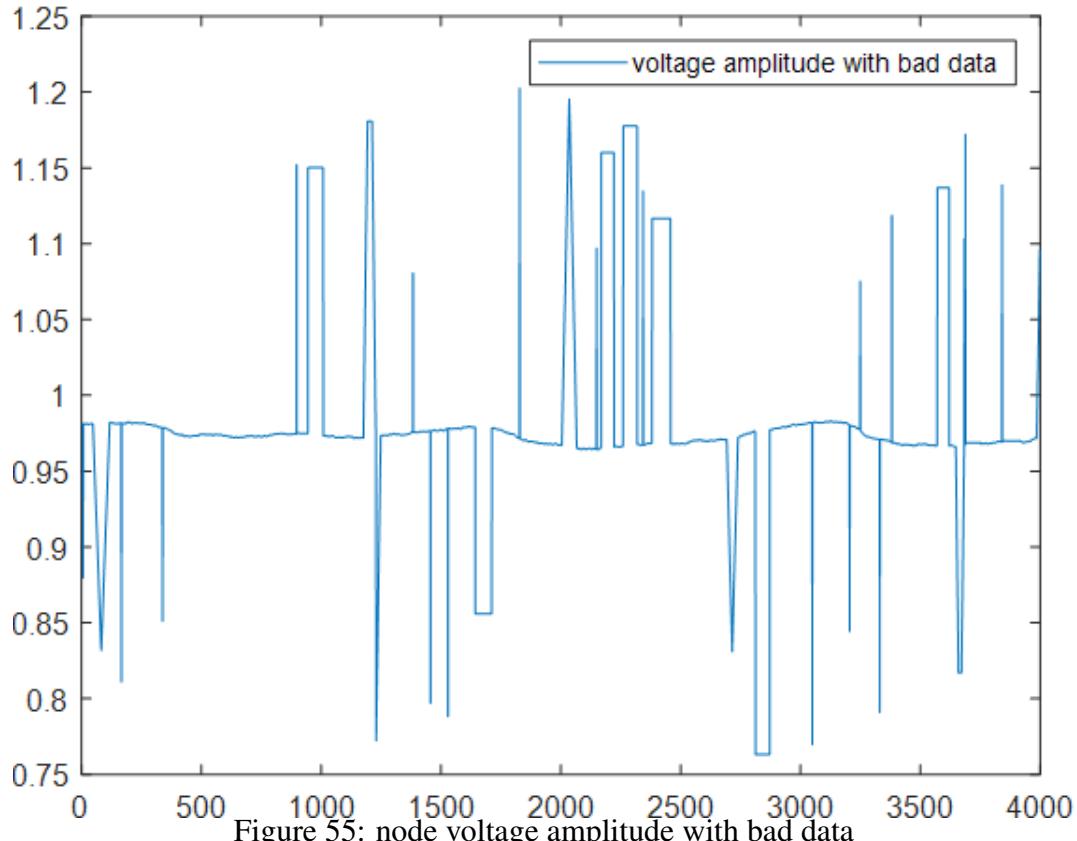


Figure 55: node voltage amplitude with bad data

prediction/estimation results.

In order to simulate three cases, we design four types of bad data as follows:

Case 1: Spike-shape. We randomly choose one data point and add/subtract 10%-25% of original data into/from original data.

Case 2: Square-shape. We randomly choose a time interval and make the data of all the time steps chosen to increase or decrease by either 0% or 10%-25%. Increasing or decreasing by 0% means keeping the data unchanged in this time interval.

Case 3: Triangular or inverted triangle shape. We randomly choose a time interval and make the data curve climb and descend as a ramp. The length of the time interval is determined by chi-square distribution.

Case 4: Trapezoidal or inverted trapezoidal shape. We randomly choose a time interval and make the data curve ramp up. Then, keep the data curve horizontal for small time intervals. Finally, make the data curve descend like a ramp. The length of the time interval is determined by chi-square distribution.

Case 1, case 2, case 3, and case 4 account for 50%, 30%, 10%, and 10% of all the bad data respectively. The overall bad data account for about 5% of the total data. All the data of the node or branch have been added to bad data. Figs.62,63,64,65 and 66 show examples of node active power injections, node reactive power injections, node voltage magnitudes, line active power lows, and line reactive power lows with bad data in the first day, respectively.

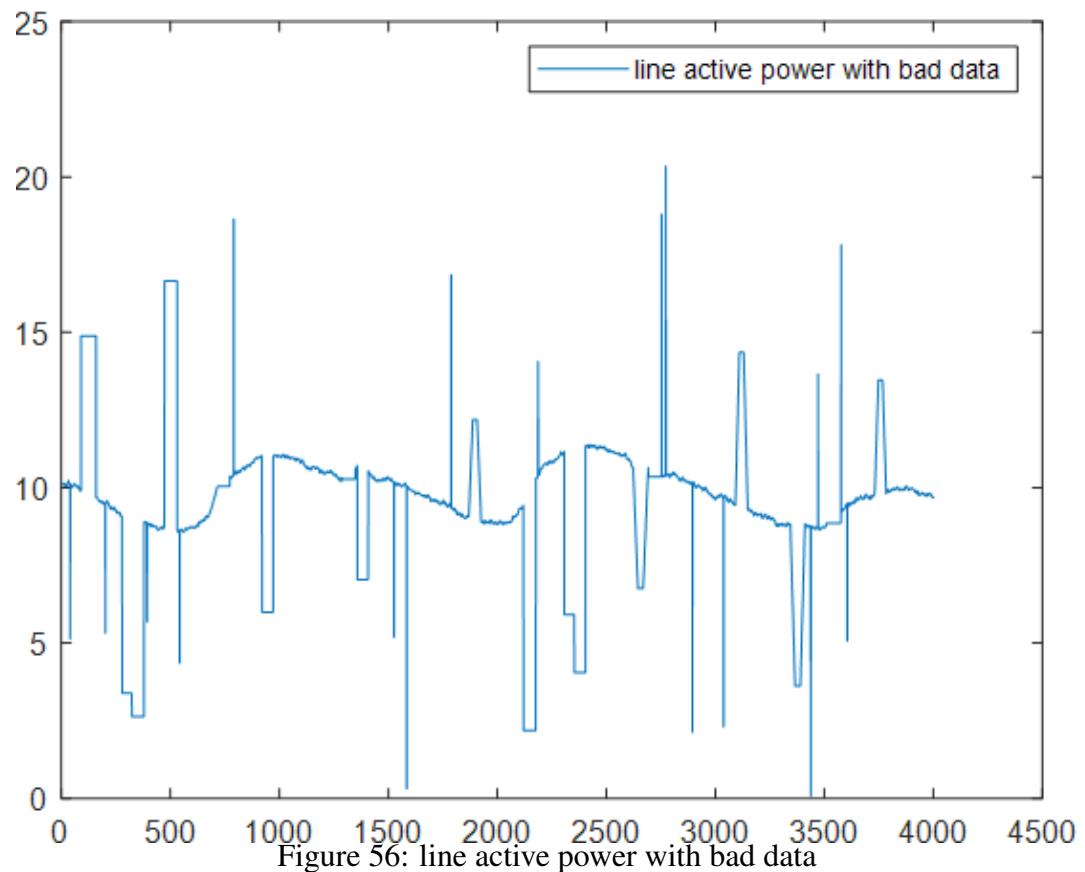


Figure 56: line active power with bad data

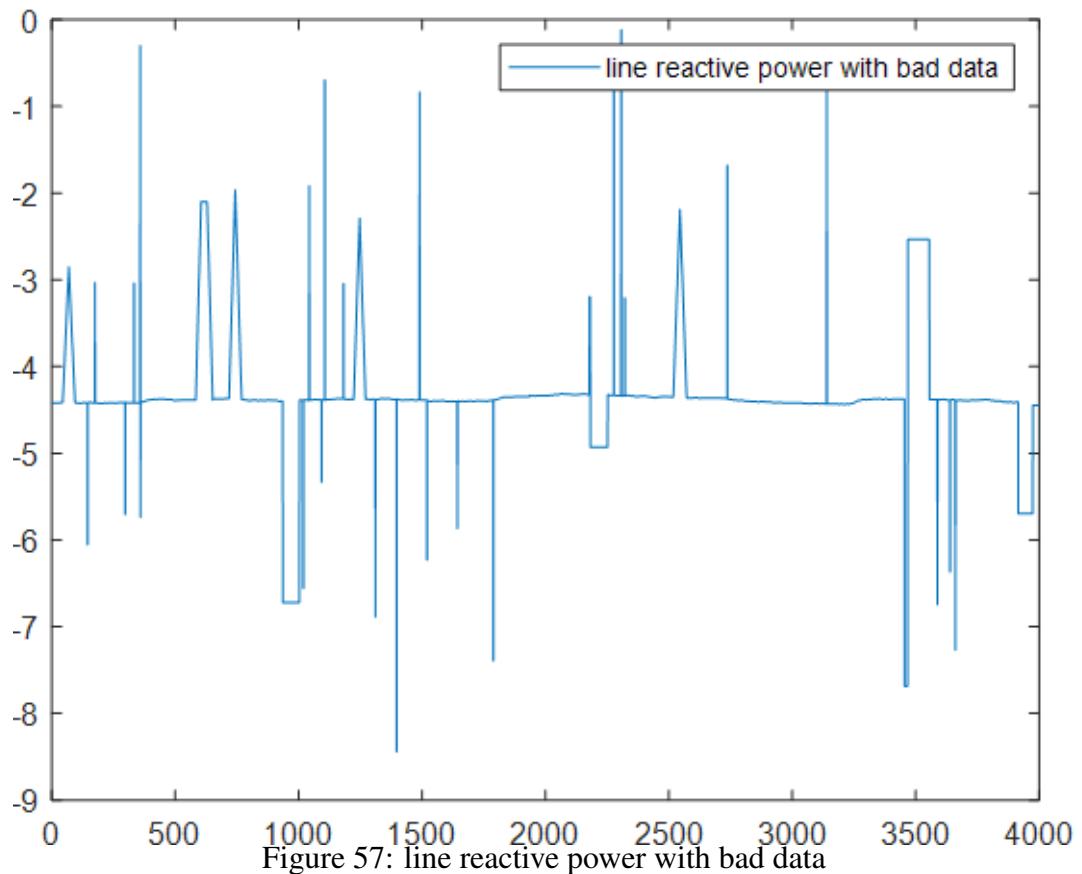


Table 23: Measurement configurations with different measurement levels

Number of lines with flow measurements	Percentage of lines with flow measurements	Number of observable lines	Percentage of observable lines	Number of unobservable lines	Percentage of unobservable lines
76	56.7%	121	90%	13	10%
61	45.5%	107	80%	27	20%
52	38.8%	94	70%	40	30%
41	30.6%	80	60%	54	40%
33	24.6%	67	50%	67	50%
27	20.2%	53	40%	81	60%
19	14.2%	41	30%	93	70%
2	1.5%	27	20%	107	80%

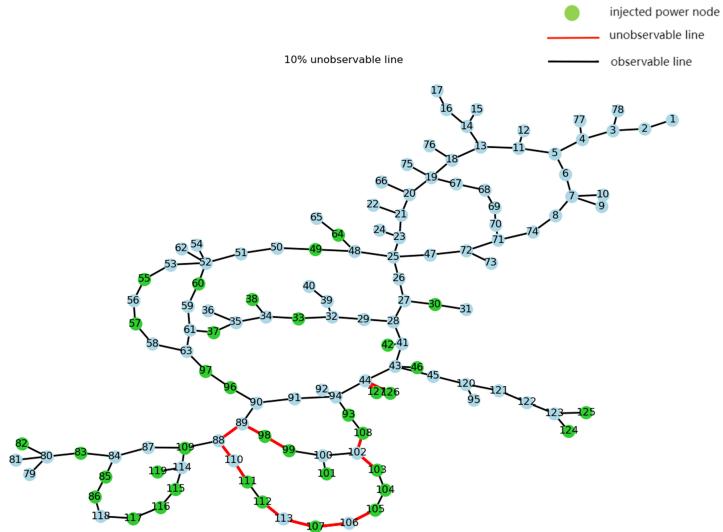


Figure 58: The system diagram of 10% unobservable lines.

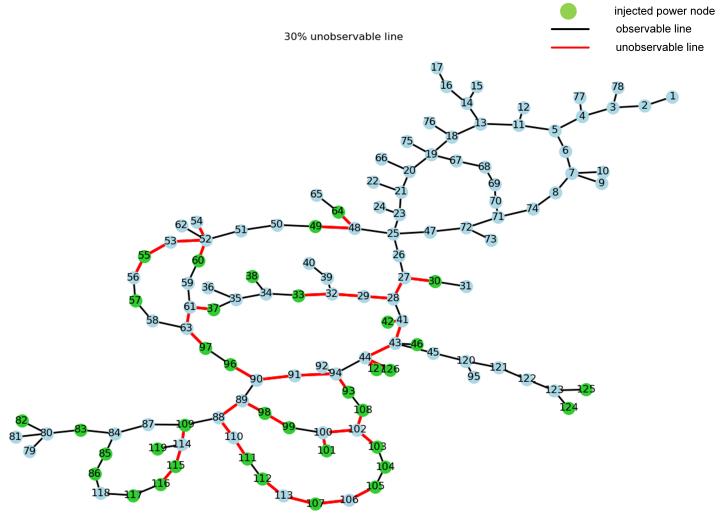


Figure 59: The system diagram of 30% unobservable lines.

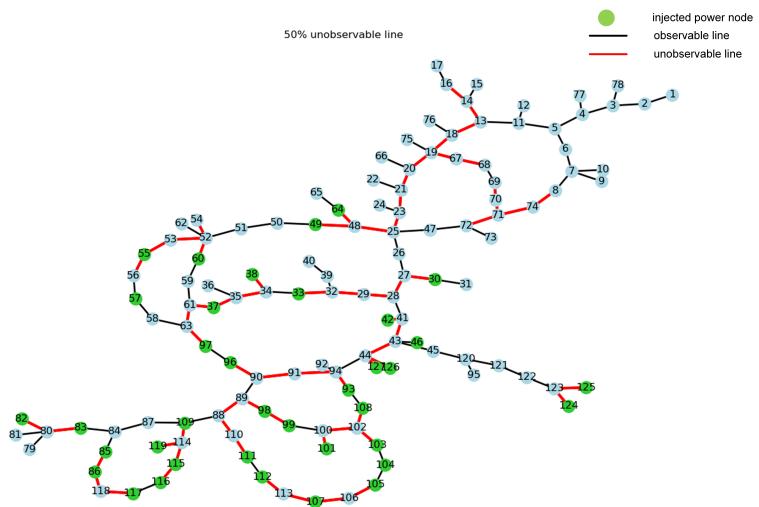


Figure 60: The system diagram of 50% unobservable lines.

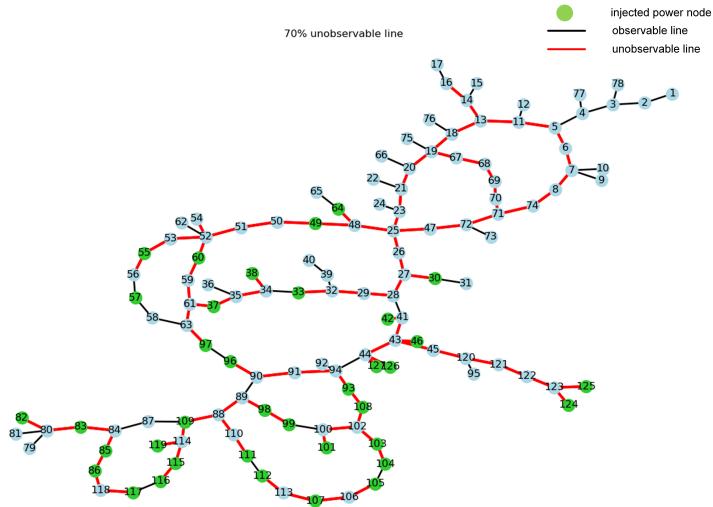


Figure 61: The system diagram of 70% unobservable lines.

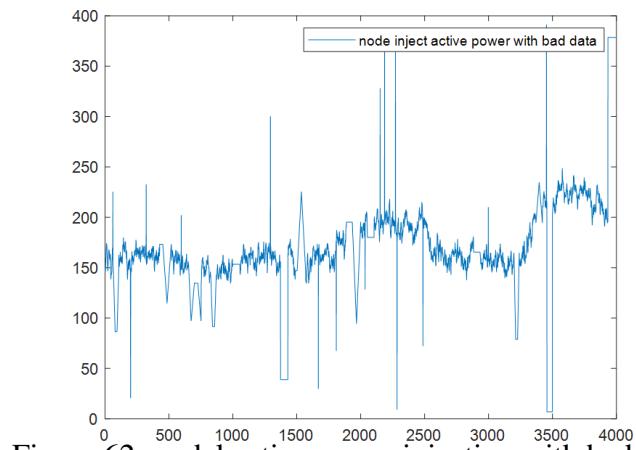


Figure 62: nodal active power injection with bad data

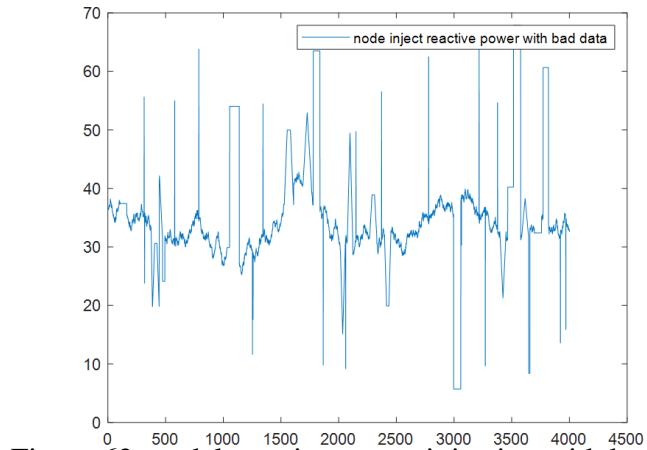


Figure 63: nodal reactive power injection with bad data

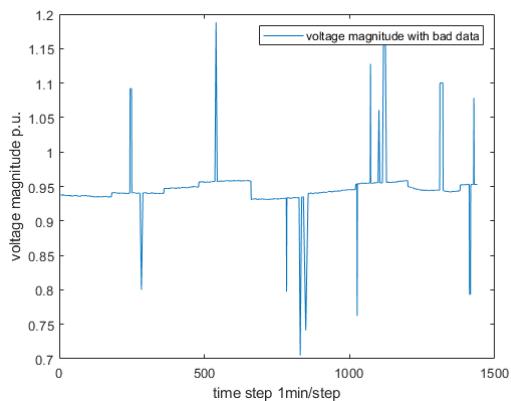


Figure 64: nodal voltage amplitude with bad data

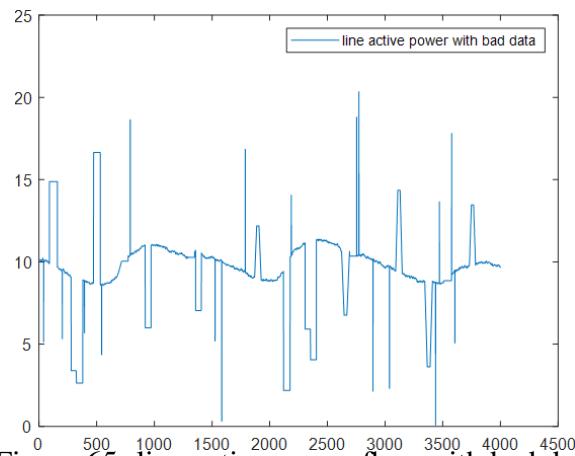


Figure 65: line active power flow with bad data

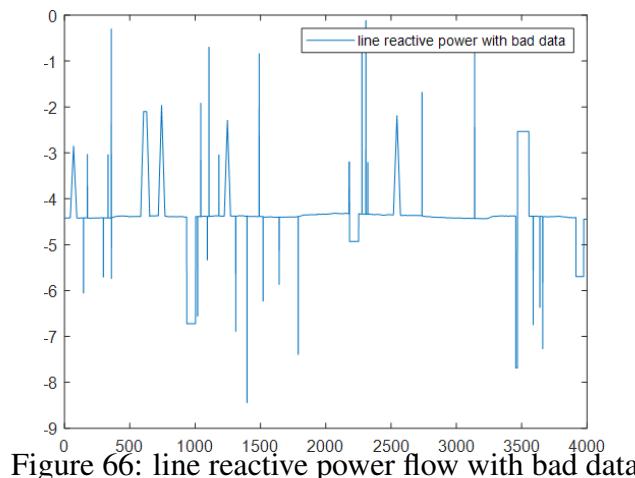


Figure 66: line reactive power flow with bad data

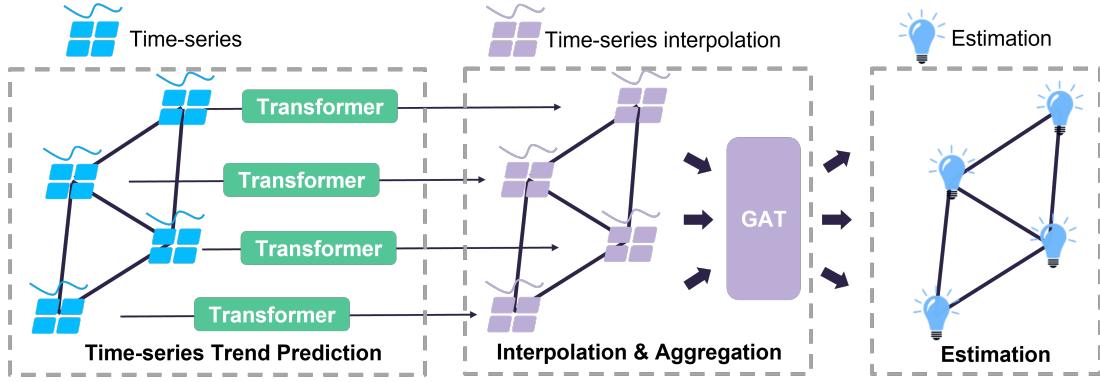


Figure 67: Overview of NG-Transformer.

Subtask 7.1: Overall NG-Transformer Framework for Measurement Synchronization

Subtask Summary: In this subtask, the Transformer module will be combined with the Graph Attention Network (GAT) module to form a complete NG-Transformer framework. It will be capable of predicting SA measurement time-series incorporating distribution system topology information. Beyond capturing the spatial and temporal trends of the SA measurements, we intend to employ a dynamic matrix with a fixed-length sliding window as inputs for our NG-Transformer framework. For validation, three steps will be carried out. First, high-resolution synchronized measurement streams will be synthesized based on power flow solutions using typical load and solar profiles in PPM. Then, most of the data points will be removed in order to emulate the slow scan rates and asynchronous fashion of SA measurements. These synthetic data, along with the distribution system topologies, will be used to train the NG-Transformer model. The predicted values given by the NG-Transformer model will be compared with the ground truth values that are removed before training to evaluate the prediction accuracy.

Figure 67 shows the framework of the NG-Transformer model, which consists of Transformer for 30-minute level trend prediction, interpolation for 1-minute level estimation, and Graph Attention Network (GAT) for 1-minute level correction. Firstly, the Transformer part takes time series of SA measurements including active and reactive power with 30-minute intervals as input, and forecasts active and reactive power 30 minutes ahead. Then Cubic Spline Interpolation is adopted to get the 1-minute level SA measurement estimation from the 30-minute level SA measurement prediction. Finally, we use Graph Attention Network (GAT) with FS measurements including bus voltages and line active and reactive power flows as input to correct the 1-minute level estimation and get the final prediction result of SA measurements. Details of each part are as follows.

Transformer module is to handle time-series data, which is denoted by $X_i^{(t)} = \{\mathbf{x}_i^{(t-T+1)}, \mathbf{x}_i^{(2)}, \dots, \mathbf{x}_i^{(t)}\}$, where $\mathbf{x}_i^{(t)}$ denotes the features of node i at the t -th timestamp, and T denotes the time steps of Transformer. The Transformer consists of Positional Encoding and several Transformer Encoders.

Positional Encoding is designed for the model to make use of the order of the sequence without involving recurrence and convolution. To inject some information about the relative or absolute position of the time-series data, we adopt a fixed positional encoding method, which is sine and cosine functions of different frequencies as follows:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d}), \quad (38)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d}), \quad (39)$$

where pos is the position, and i is the dimension. That is, each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 \cdot 2\pi$. Since for any fixed offset k , PE_{pos+k} can be represented as a linear function of PE_{pos} , it would allow the model to easily learn to attend by relative positions. Then the positional encoding is added to the input data as follows:

$$H = X + PE, \text{ and } \hat{H} = \hat{X} + PE, \quad (40)$$

where H and \hat{H} are embeddings for X and \hat{X} , respectively. While this positional encoding method is fixed, there are no learnable parameters in this part. We do not add an additional learned embedding layer to convert input data of the first and second period to a same dimension because the dimensions of the two pieces of data not only have the same length but also have the same meaning.

Each Transformer Encoder has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise, fully connected feed-forward network. There is a residual connection around each of the two sub-layers, followed by layer normalization. It first defines a scaled dot-product attention function, which is shown as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^\top}{\sqrt{d_k}}\right)V, \quad (41)$$

where Q , K , and V are embeddings generated by H and θ_A , denoting queries, keys, and values, respectively. Here d_k is the dimension of K . Based on Eq. (41), multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions, which can be formulated as:

$$E = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)\theta_O, \quad (42)$$

$$\text{head}_i = \text{Attention}(H \cdot \theta_{Q_i}, H \cdot \theta_{K_i}, H \cdot \theta_{V_i}), i \in [1, h], \quad (43)$$

where h denotes the number of heads, and θ_O , $\theta_Q = \{\theta_{Q_1}, \dots, \theta_{Q_h}\}$, $\theta_K = \{\theta_{K_1}, \dots, \theta_{K_h}\}$, and $\theta_V = \{\theta_{V_1}, \dots, \theta_{V_h}\}$ are learnable parameters. With a layer normalization function $\text{LayerNorm}(\cdot)$, the output of the first sub-layer in Transformer Encoder can be written by:

$$\tilde{E} = \text{LayerNorm}(H + E). \quad (44)$$

Then for the second fully connected layer together with another normalization layer, the Transformer Encoder generates a hidden representation Z by the following equation:

$$Z = \text{LayerNorm}(\tilde{E} + \tilde{E} \cdot \theta_F), \quad (45)$$

where θ_F denotes the learnable parameters in the fully connected layer. Lastly, a fully connected layer for prediction is built as follows:

$$P = Z \cdot \theta_P, \quad (46)$$

where θ_P denotes the learnable parameters in the prediction layer.

We adopt mean squared error (MSE) in our objective function for Transformer, which is as follows:

$$\mathcal{L}_T = \sqrt{\frac{1}{n} \sum_i^n (P_i - Y_i)^2}, \quad (47)$$

where Y_i denotes the ground truth values of node i .

Cubic Spline Interpolation is a form of interpolation where the interpolant is a special type of piecewise cubic polynomial called a cubic spline. That is, instead of fitting a single, high-degree polynomial to all of the values at once, cubic spline interpolation fits low-degree cubic polynomials to small subsets of the values. Specifically, we assume that the points (x_i, y_i) and (x_{i+1}, y_{i+1}) are joined by a cubic polynomial $S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$ that is valid for $x_i < x < x_{i+1}$ for given i . To find the interpolating function, we must first determine the coefficients a_i, b_i, c_i, d_i for each of the cubic functions. For n points, there are $n - 1$ cubic functions to find, and each cubic function requires four coefficients. In our setting, we use $n = 4$ points to get the 1-minute level SA measurement estimation from the 30-minute level SA measurement prediction, where the first 3 points are from historical data and the last point is from the prediction of Transformer.

Graph Attention Network (GAT) is an attention-based architecture for graph-structured data, which computes the hidden representations of each node in a graph by attending over its neighbors, following a self-attention strategy. In the attention mechanism, normalized coefficients are calculated by the following equation:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_k])), \quad (48)$$

where $\alpha_{i,j}$ denotes the normalized coefficient of edge (i, j) for node i , $\mathcal{N}(i)$ is the set of all neighbor nodes of node i , \mathbf{h}_i denotes features of node i , Θ is a learnable matrix applied to every node, and \mathbf{a} is a shared single-layer feedforward neural network as the attentional mechanism. Here $\|$ is the concatenation operation. In order to handle edge features as well, we follow the idea of Edge-Featured Graph Attention Network (EGAT), which considers edge features in the attention mechanism as follows:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_j \| \Theta_e \mathbf{e}_{i,j}]))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_k \| \Theta_e \mathbf{e}_{i,k}])), \quad (49)$$

where Θ_e is a learnable matrix applied to every edge, and $\mathbf{e}_{i,j}$ denotes features of edge (i, j) . With the attention mechanism, the hidden representation of an aggregation can be written as:

$$\mathbf{h}_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \Theta \mathbf{h}_j \right), \quad (50)$$

where \mathbf{h}_i is initialized as input features of node i .

We adopt mean squared error (MSE) in our objective function for GAT, which is as follows:

$$\mathcal{L}_G = \sqrt{\frac{1}{n} \sum_i^n (h_i - g_i)^2}, \quad (51)$$

where g_i denotes the gap between the ground truth and the interpolation value of node i .

Experiments with Different Topology Information. We use clean data and assume all 1-minute interval line active and reactive power injection data are given. Under this assumption, we still keep the training/test split described in the above-mentioned experiments and evaluate model performance by mean absolute error (MAE).

Table 24: SA measurement prediction results with different topology information in terms of MAE.

Model	Setting	Active Power	Reactive Power
Transformer	no topology	24.6426	5.9463
NG-Transformer	varying topology	1.0875	1.4501
Function	varying topology	0.9734	1.2542

We evaluate the bus voltages of the 1st topology series generated in Task 6.1 in previous quarters. Table 24 shows the prediction results of models with different topology information. Here Transformer did not have topology information as input, and NG-Transformer involved the topology information. We also include a model “Function,” which calculates the power and reactive power by power flow solver functions, for comparison. The errors of “Function” mainly come from the precision of the recorded line active and reactive power injection data.

From the results of Table 24, we can see that topology information has a significant effect on model prediction. For active power, the improvement is $\frac{(24.6426-0.9734)-(1.0875-0.9734)}{24.6426-0.9734} = 99.52\% > 80\%$, and for reactive power, the improvement is $\frac{(5.9463-1.2542)-(1.4501-1.2542)}{5.9463-1.2542} = 95.82\% > 80\%$, satisfying **Milestone 2.7.1**. Figure 68 and Figure 69 are examples of prediction results with different topology information, which show the effectiveness of topology information in active and reactive power prediction.

In addition, the team tested the computational performance of the developed measurement predictor. For the prediction of a measurement scan, the time consumption is **2.3954 ms**. Considering the fact that distribution system state estimation is usually executed every 1 minute to 15 minutes, the developed prediction model commensurate very well with the state estimation and is well suited for real-time application. Therefore, **Milestone 2.7.2** is satisfied.

Subtask 12.1: Testing and Validation Based on Actual Measurements.

The team has created algorithms and simulations to generate enough amount of data to support the measurement prediction and state estimation tasks. They include fast SCADA measurements and slow AMI measurements, as well as the ground truth of the variables. Corrupted data with gross errors were also synthesized. Topology changes were simulated in a random fashion. Please refer to previous reports for more information.

Subtask Summary: *The developed software will be tested using actual measurements. The full coverage of PMUs in PPM allows us to verify the state estimation results from the developed software conveniently. Two measurement sets will be consolidated: test set and reference set. The*

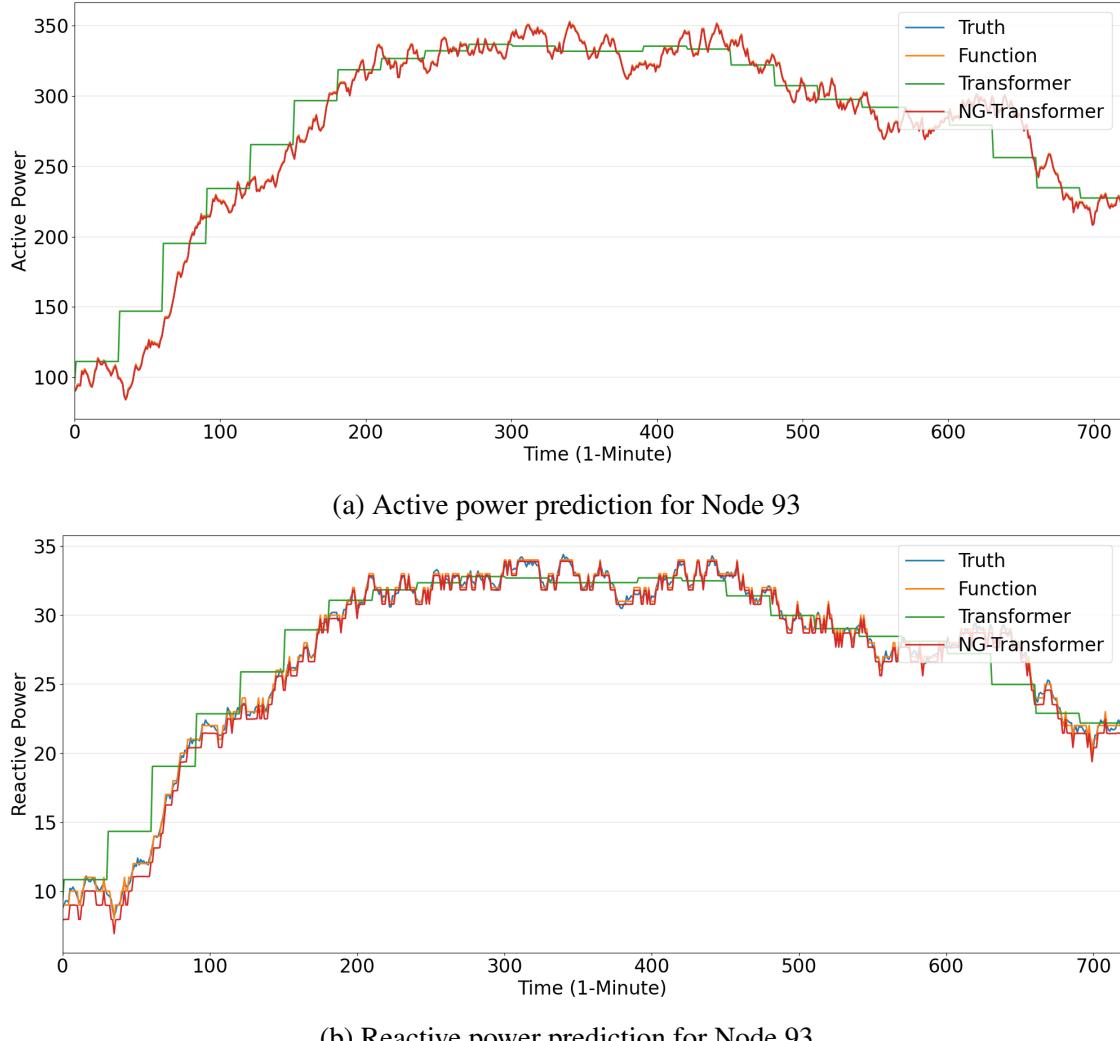


Figure 68: Prediction results using different topology information for Site 93.

test set will include all AMI measurements, all SCADA measurements, and a sparsely selected subset of PMU measurements in PPM. The purpose of the selection is to mimic the measurement configuration of a common distribution system where PMU coverage is limited. The test set will be fed into the developed software to produce estimated values of state variables. The reference set will include all PMU measurements in PPM. They are regarded as the “ground truth” of state variables. The performance of the developed software can then be evaluated by comparing the estimated values from the test set with the “true” values from the reference set. Offline test cases covering practical scenarios and different topologies will be used for testing and validation of the developed software.

In this subtask, we provide a detailed explanation of our approach to tackling the challenges inherent in the problem. We begin by discussing the motivation behind our model design, outlining the key principles and considerations that influenced our decision-making. This includes identifying the specific issues we aimed to address, recognizing the limitations of existing methods, and

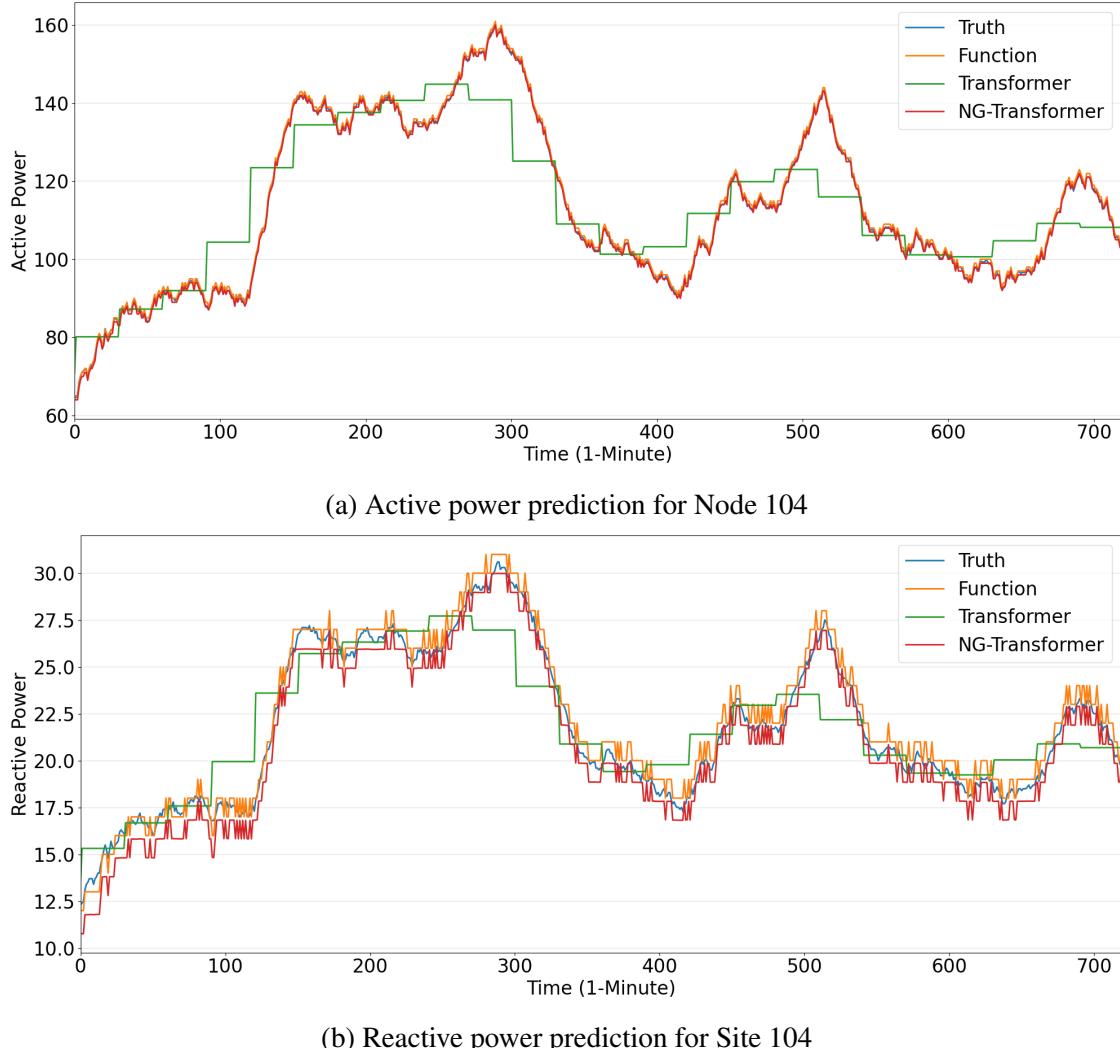


Figure 69: Prediction results using different topology information for Node 104.

highlighting the innovative aspects of our proposed solutions.

We first examine a scenario where the dataset is clean but only partial information about line power—both in forward and reverse directions—is available within a distribution system. To address this, we introduce our first model, LineMLP, and provide a comprehensive breakdown of its architecture, key components, and design rationale. We demonstrate how LineMLP effectively leverages the available data for accurate predictions and analyze its performance across various test cases, emphasizing its advantages over traditional methods.

Next, we shift our focus to a more dynamic setting where the distribution system topology frequently changes, introducing greater uncertainty and complexity. To handle this challenge, we build upon LineMLP and propose EGC-GAT (Edge Graph Convolutional Graph Attention Network), a model specifically designed to adapt to these variations. We delve into the architecture of EGC-GAT, explaining its ability to process and analyze directional power data using advanced graph-based techniques. Additionally, we present extensive testing and validation results, demon-

strating EGC-GAT's robustness, accuracy, and superior performance compared to both LineMLP and existing state-of-the-art models.

We then consider an even more constrained scenario where only partial line power data in a single direction is available. To manage this limitation, we propose a simple estimation strategy to infer missing power data before applying the EGC-GAT model. Our experimental results show that EGC-GAT remains effective even under these restrictive conditions, maintaining strong predictive performance.

To further enhance model robustness, we introduce noise into the clean dataset to generate corrupted data and establish a closed-loop operational framework integrating EGC-GAT with the WLAV robust state estimator. In this framework, EGC-GAT enhances system observability and measurement redundancy, while the WLAV estimator enforces physical grid constraints and facilitates retraining of EGC-GAT. This synergy improves the model's reliability in handling corrupted data scenarios.

In summary, this section traces the evolution of our model design from LineMLP to EGC-GAT, offering a comprehensive discussion of both theoretical and practical advancements. By systematically addressing different levels of data availability and system variability, our models make significant contributions toward improving distribution network analysis and prediction accuracy.

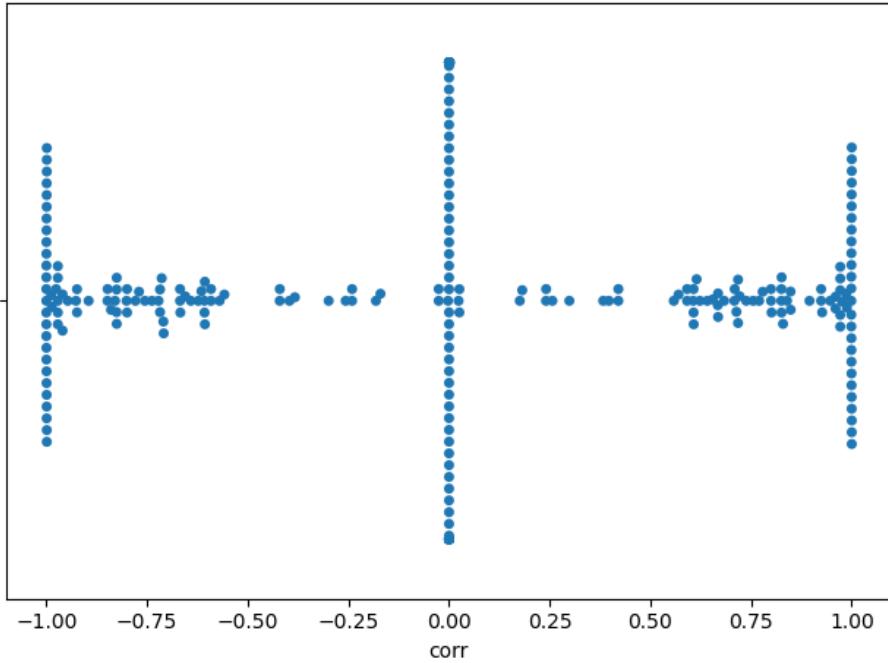


Figure 70: Example distribution of correlations based on 20% historical active power data.

Motivation. The relationship between active and reactive powers in electrical lines is a fundamental aspect of power system analysis, as these two quantities are inherently interconnected. To illustrate this relationship, we present a detailed analysis using Figure 70 and Figure 71, which depict the distributions of correlations between one specific line and all other lines within a distribution network. These correlations are calculated based on historical active power data, with Figure 70 representing a scenario where only 20% of the historical data is available, and Figure 71

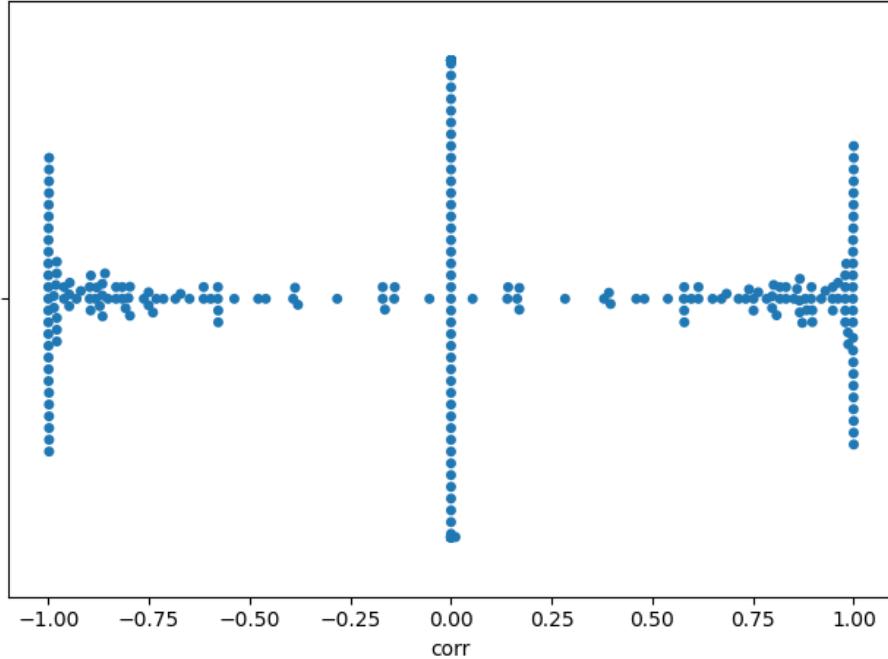


Figure 71: Example distribution of correlations based on 80% historical active power data.

corresponding to a case where 80% of the historical data is utilized. From these figures, a striking observation emerges: regardless of how the topology of the network changes—whether due to reconfigurations, faults, or other dynamic factors—there consistently exist numerous lines that exhibit strong correlations with one another in terms of active power. These correlations are often close to either 1 or -1, indicating highly positive or negative relationships, respectively. This persistent pattern underscores the intrinsic dependencies among lines in a power distribution system, even in the face of topological variability.

Scenario: Clean Data + Bidirectional Edge Information + Unchanged Topology. Motivated by the above-mentioned observation, we develop a two-step methodology to address the challenge of estimating power flows in scenarios where measurements are incomplete or unavailable. In the first step, we focus on learning the active and reactive power flows of lines that lack direct measurements. This is achieved by leveraging the data from lines that are equipped with fast and sparse (FS) measurements, which provide comprehensive information about their power flows. By analyzing the correlations and dependencies among lines, as demonstrated in the figures, we are able to predict the active and reactive power flows for unmeasured lines with a high degree of accuracy. This step is crucial because it allows us to fill in the gaps in the data, creating a more complete picture of the power flows across the entire network.

In the second step, we utilize the predicted active and reactive power flows of the lines to estimate the real and reactive powers at the nodes, which correspond to the slow and abundant (SA) measurements. This involves applying power flow equations and network topology information to derive the nodal powers from the line powers. By doing so, we are able to reconstruct the state of the entire distribution system, even in cases where direct measurements at the nodes are unavailable or sparse. This approach not only enhances the accuracy of state estimation but also provides a robust framework for handling scenarios where the network topology is subject to

frequent changes.

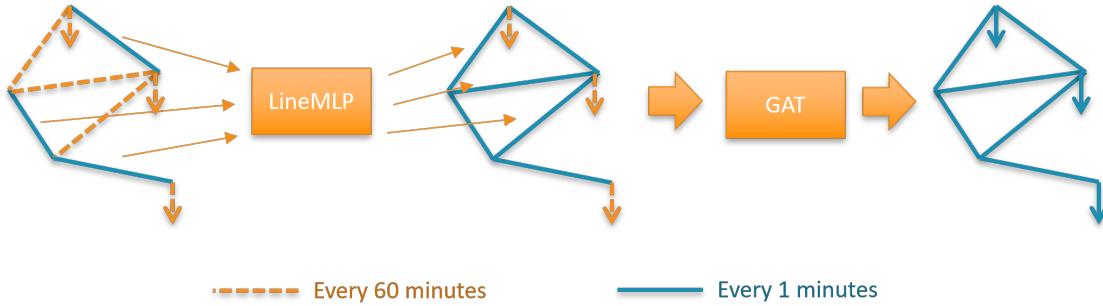


Figure 72: Overall framework for measurement synchronization.

Figure 72 presents the overall structure of the proposed model, which is designed to perform 1-minute level predictions for both lines and nodes within a power system. The framework integrates two primary components: a Line Multilayer Perceptron (MLP) and a Graph Attention Network (GAT) [28, 29], each specializing in a different aspect of the prediction task. The first component, the LineMLP, processes fast and sparse (FS) measurements from known lines, referred to as FS lines, which include both active and reactive power data recorded at 1-minute intervals. These inputs allow the model to forecast the active and reactive power of lines that do not have direct measurements available, labeled as SA (Supervisory) lines. This prediction of power for SA lines forms the foundation for the next phase of the model, where the Graph Attention Network comes into play. The second component, the Graph Attention Network (GAT), takes as input a combination of FS measurements, including bus voltages, alongside the predicted line-level active and reactive power values. The GAT is tasked with estimating the SA measurements at the node level, leveraging its ability to model the relationships between different nodes in the power system. However, since SA measurements are only available at 30-minute intervals, the model adapts by refining the 1-minute level predictions every 30 minutes using the available SA measurements. This correction process ensures that the final predictions are accurate and aligned with the 30-minute interval data. Through this approach, the model effectively blends high-frequency line-level forecasts with lower-frequency node-level measurements to produce a comprehensive, high-accuracy prediction system. The following provides an in-depth description of each individual part of the framework, explaining the intricacies of the LineMLP, the GAT, and how these components work together to achieve the final predictive results.

The Line Multilayer Perceptron (MLP) component in the proposed model is responsible for predicting the data of SA (Supervisory) lines, utilizing fully connected layers for accurate forecasting. In particular, we utilize a two-layer MLP architecture, which takes in FS (Full-Spectrum) line data as input and generates predictions for the corresponding SA lines. The layers of this MLP network are connected via trainable weight matrices, $\Theta_L^{(l)}$, where l represents the layer number, and $e^{(l)}$ denotes the embeddings of the nodes at the l -th layer. This layer-wise processing follows the equation:

$$e_{\Theta_L}^{(l+1)} = \Phi(e_{\Theta_L}^{(l)} \cdot \Theta_L^{(l)}), \quad (52)$$

where $\Phi(\cdot)$ represents the activation function, which in this case is the Rectified Linear Unit (ReLU) function. This activation helps introduce non-linearity into the network, enabling it to

model complex relationships between the input and output variables. To train this model, we aim to minimize the error between the predicted and ground truth values of the SA lines, incorporating a regularization term to encourage sparsity in the weight matrix, Θ_L . Specifically, the objective function for the LineMLP is formulated as a combination of Mean Squared Error (MSE) and a soft constraint for sparsity:

$$\mathcal{L}_L = \sqrt{\frac{1}{m} \sum_i^m (e_i - l_i)^2 + \alpha |\Theta_L|_2}, \quad (53)$$

where l_i represents the ground truth values for line i , m is the total number of lines in the dataset, and α controls the strength of the regularization term, helping to avoid overfitting by encouraging sparsity in the learnable weight parameters.

The Graph Attention Network (GAT) [28], on the other hand, is a crucial component used to process graph-structured data, which, in this case, involves modeling the relationships between various nodes in a power grid. The GAT employs an attention-based mechanism, which assigns weights to different neighbors of a node based on their importance, allowing for a flexible and adaptive method of aggregating information from neighboring nodes. In the attention mechanism, the normalized coefficients, $\alpha_{i,j}$, which denote the importance of the edge between nodes i and j , are computed by:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_j]))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_k])), \quad (54)$$

where \mathbf{h}_i and \mathbf{h}_j represent the feature vectors of nodes i and j , respectively, Θ is a learnable matrix that is applied to node features, and \mathbf{a} is a learnable vector used in the attention mechanism. The operator $\|$ denotes concatenation, which combines the features of the nodes being attended to. This attention coefficient reflects how much influence node j has on the feature aggregation for node i . To extend this model to handle edge features, we adopt the Edge-Featured Graph Attention Network (EGAT) [29] approach, which incorporates edge features, $\mathbf{e}_{i,j}$, into the attention mechanism as follows:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_j \| \Theta_e \mathbf{e}_{i,j}]))}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\text{LeakyReLU}(\mathbf{a}^\top [\Theta \mathbf{h}_i \| \Theta \mathbf{h}_k \| \Theta_e \mathbf{e}_{i,k}])), \quad (55)$$

where Θ_e is a learnable matrix applied to edge features, and $\mathbf{e}_{i,j}$ is the feature vector associated with the edge connecting nodes i and j . Once the attention coefficients are calculated, the hidden representation of each node is updated by aggregating information from its neighbors, weighted by the attention coefficients, as follows:

$$\mathbf{h}_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \Theta \mathbf{h}_j \right), \quad (56)$$

where $\sigma(\cdot)$ denotes a non-linear activation function, and \mathbf{h}_i is the updated feature vector for node i . This mechanism allows the model to selectively focus on important neighbors and edges, leading to more precise node-level predictions.

To train the GAT, we utilize the Mean Squared Error (MSE) loss function, similar to the one

used in the LineMLP component. The objective function for the GAT is formulated as:

$$\mathcal{L}_G = \sqrt{\frac{1}{n} \sum_i^n (h_i - g_i)^2}, \quad (57)$$

where h_i represents the predicted feature of node i , and g_i denotes the ground truth value for node i . The model is trained to minimize the difference between the predicted and actual values of the nodes, refining the hidden representations through the attention mechanism to achieve accurate predictions.

Finally, to jointly optimize both the LineMLP and GAT components, we combine the individual loss functions for each component into a unified objective function. This overall objective function is given by:

$$\min_{\Theta} \mathcal{L}_L + \beta \mathcal{L}_G, \quad (58)$$

where \mathcal{L}_L corresponds to the loss for the LineMLP, aimed at improving predictions for the lines, and \mathcal{L}_G corresponds to the loss for the GAT, focused on enhancing the accuracy of node predictions. The parameter β controls the relative importance of the two components, allowing the model to balance the optimization process and make accurate predictions for both the line and node data simultaneously. This unified optimization ensures that the model effectively integrates information from both the line-level and node-level perspectives, leading to more robust and accurate power system predictions.

Table 25: FS measurement prediction results with proportions of unknown lines over the first group of bus voltage control and distribution system topology information in terms of MAE.

Proportions of Unknown Lines	Line Prediction		Node Prediction		Node Error Percentage	
	Active Power	Reactive Power	Active Power	Reactive Power	Active Power	Reactive Power
10%	0.0013	0.0004	1.1889	1.2699	0.64%	3.03%
15%	0.0040	0.0007	4.3735	1.4033	2.35%	3.35%
20%	0.0020	0.0010	1.3191	1.2963	0.71%	3.10%
25%	0.0022	0.0009	2.1070	1.4722	1.13%	3.52%
30%	0.0023	0.0024	1.6970	2.2638	0.91%	5.41%

In this series of experiments, we evaluate the effectiveness of the proposed framework by simulating conditions where a certain proportion of the line data is unavailable. To assess the model's robustness and predictive capability under these conditions, we use a clean dataset, ensuring that all the data used in the evaluation is free of noise or anomalies. The evaluation process involves down-sampling the node active and reactive power injection data to 30-minute intervals, as this serves as the SA (Supervisory) measurement input to the model, which is obtained from Advanced Metering Infrastructure (AMI). Notably, the 1-minute interval data is reserved for evaluation purposes only, allowing us to validate the model's performance at a finer resolution. For consistency, the node voltage data is maintained at 1-minute intervals, as it is assumed that these values come from SCADA (Supervisory Control and Data Acquisition) systems, which are expected to provide continuous, high-resolution measurements.

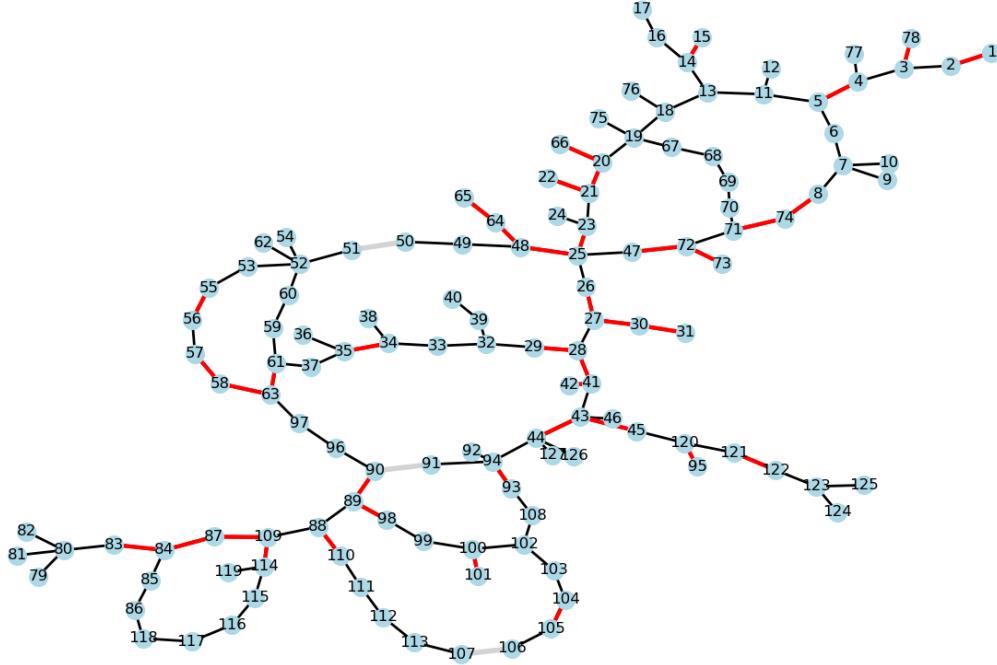


Figure 73: Topology with 30% lines unknown.

Regarding the line data, we consider the active and reactive power injection measurements. A certain proportion of the lines are designated as SA lines, meaning they lack direct measurements. Instead, the values for these lines are derived from the available nodal SA measurements (from the AMI data, recorded every 30 minutes). The remaining lines, which still have direct measurements available, are considered as FS (Full-Scale) lines and are treated as having high-resolution 1-minute interval data. This setup mimics real-world scenarios where some lines may be missing measurement data, and predictions need to be made based on indirect information.

For the model training process, we utilize data from the last 20 days of January and February. This period serves as a training set to allow the model to learn patterns and relationships within the system. In contrast, for testing the model's performance, we use data from the last 20 days of March as both the test set and the reference set. This provides a robust evaluation by using unseen data that simulates real-time operational conditions. To quantify the model's prediction accuracy, we use the Mean Absolute Error (MAE) as the evaluation metric. MAE provides a clear measure of the model's prediction accuracy, showing the average difference between the predicted and actual values without being overly sensitive to outliers.

In these experiments, we specifically focus on the first group of bus voltage control and distribution system topology information. To simulate varying levels of missing data, we experiment with different proportions of unknown lines, ranging from 10% to 30%. These proportions represent the percentage of lines for which measurement data is unavailable. Table 25 presents the prediction results, clearly demonstrating that even with a significant portion of line data missing (up to 30%), the model is able to make reasonably accurate predictions with MAE values that remain relatively low, well under 10% in all scenarios.

Additionally, Figure 73 illustrates a specific topology where 30% of the lines are unknown,

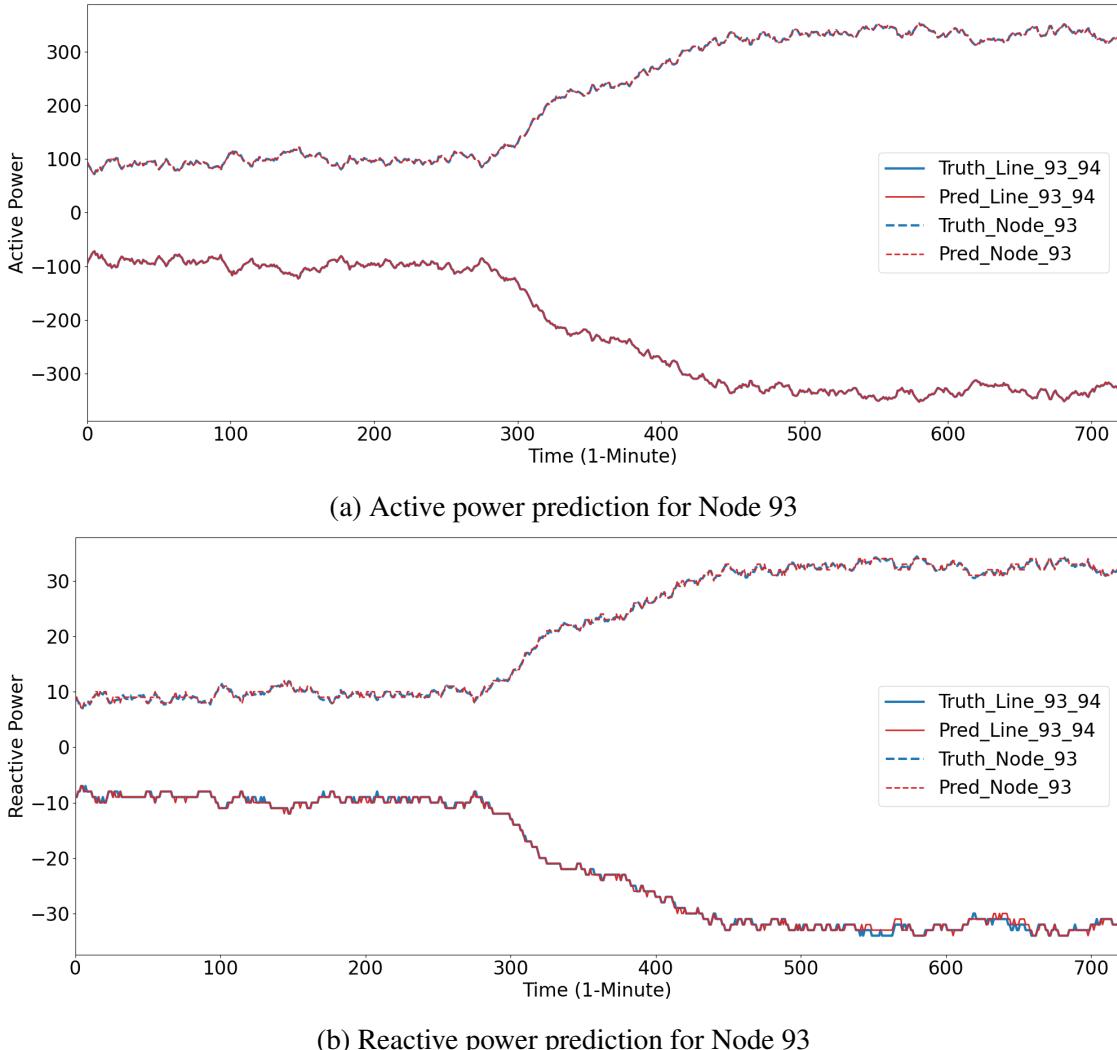


Figure 74: Prediction results for Node 93 with 30% lines unknown.

helping to visualize how the model deals with missing data. In this figure, the black lines represent FS lines, where data is available at a high resolution. The red lines are the unknown SA lines, and the gray lines indicate disconnected lines that do not factor into the calculations at the beginning of the data series. This visualization highlights how the missing data is distributed across the network, affecting both the line and node predictions.

To further evaluate the model's performance, we present several examples of predictions for nodes that are connected to unknown lines. Figures 74, 75, and 76 show the predicted active and reactive power for selected nodes under the 30% unknown line scenario. In these examples, the blue lines represent the ground truth values, which are the actual measurements, while the red lines depict the predictions generated by the proposed model. The solid lines in the figures represent the active/reactive power of the lines, scaled by a factor of 1000 to align with the node power values. The dashed lines, on the other hand, represent the power at the nodes themselves.

From these visual examples, it is evident that in most cases, the model's predictions are very

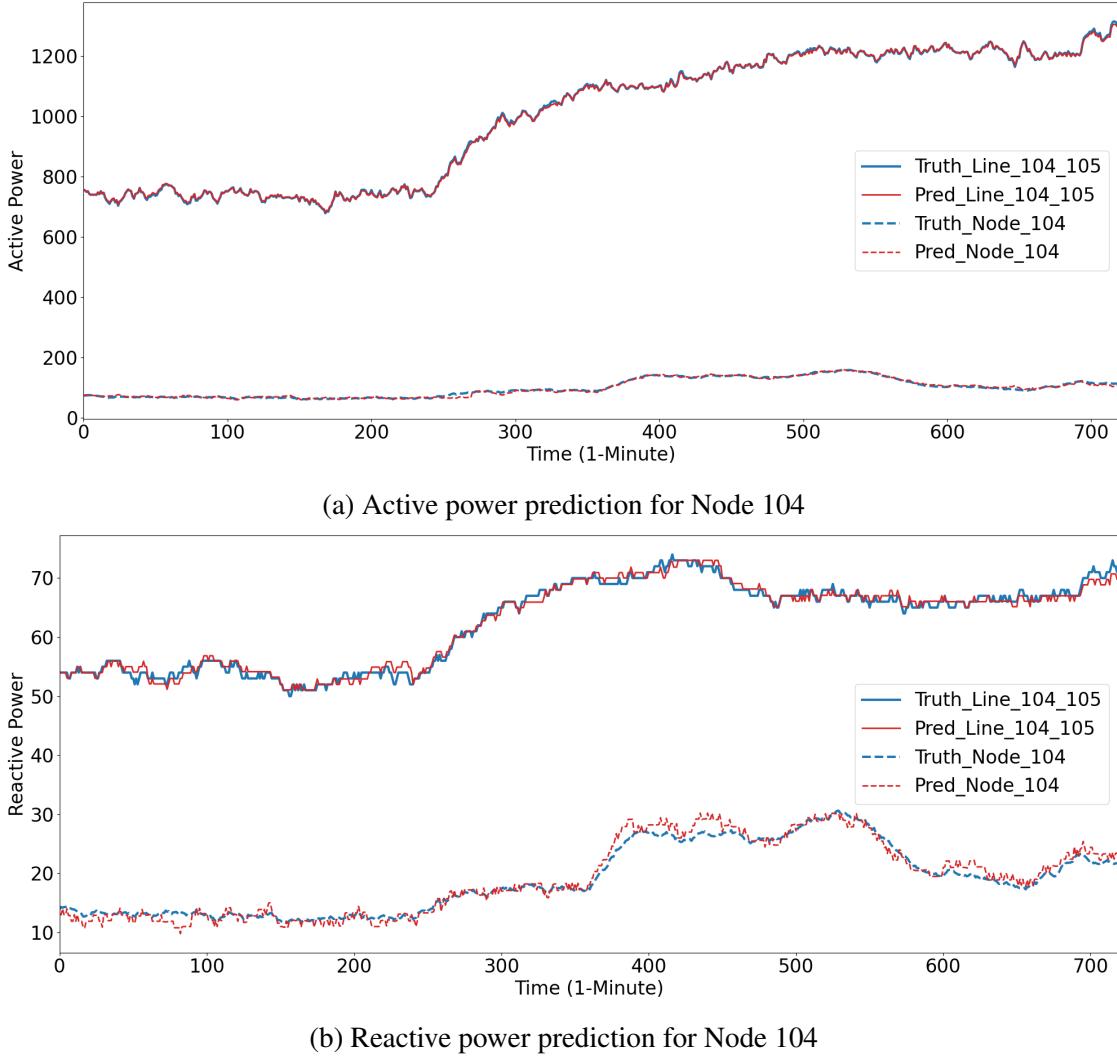
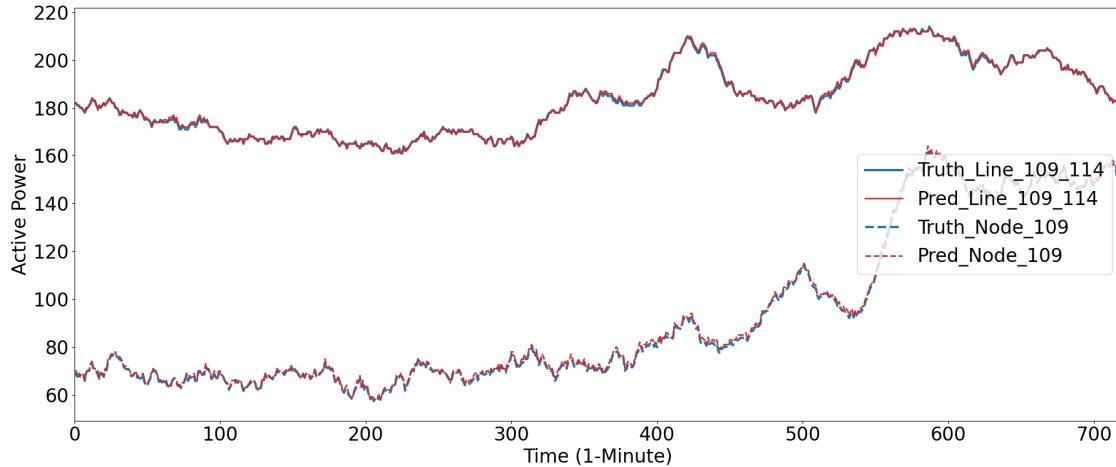


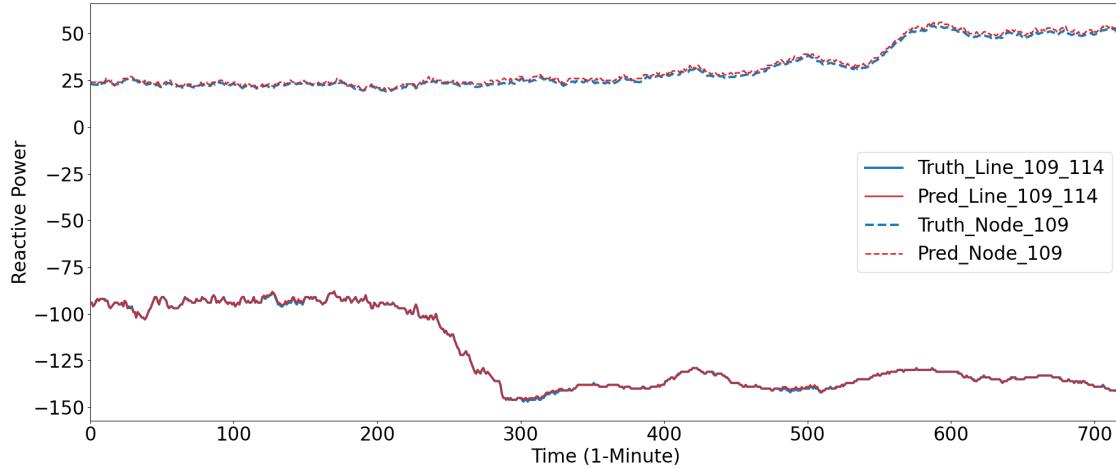
Figure 75: Prediction results for Node 104 with 30% lines unknown.

close to the ground truth. This indicates that the proposed framework is capable of producing highly accurate predictions even in the presence of missing data, which is a critical feature for real-world applications where not all lines may have available measurement data at any given time. These results validate that the model can effectively handle scenarios where part of the system's line data is missing, demonstrating its robustness and applicability in environments where real-time data availability is an ongoing challenge. Furthermore, the ability to achieve such high accuracy with partial input data opens up opportunities for using this model in real-world power systems, where missing or incomplete data is a common occurrence. Thus, the proposed framework is shown to be a powerful tool for making accurate predictions even in the face of uncertainty in system measurements.

Scenario: Clean Data + Bidirectional Edge Information + Changing Topology. We take a deeper dive into analyzing PPM's AMI (Advanced Metering Infrastructure) data to improve the performance of our proposed deep learning model, specifically for SA measurement prediction in the context of changing topologies.



(a) Active power prediction for Node 109



(b) Reactive power prediction for Node 109

Figure 76: Prediction results for Node 109 with 30% lines unknown.

As discussed in the motivation section, there is a clear interdependence between the active and reactive power flows of the lines in the power network. Recognizing this relationship, we designed the LineMLP model. This model aims to predict the active and reactive power flows for lines that lack measurements, using data from lines that have FS measurements. Once the missing line measurements are predicted, the model proceeds to estimate the real and reactive power at the nodes, effectively providing SA measurement predictions. The LineMLP model performs adequately when the topology of the system remains static, as it relies on the assumption that the relationships between lines and nodes are constant.

However, a significant limitation of the LineMLP model arises when the system's topology changes. The model fails to account for shifts in the network structure because it does not incorporate any information about the network topology. This lack of adaptability to topology changes means that the LineMLP model cannot accurately handle scenarios where the power system undergoes reconfiguration, which is a common occurrence in real-world power grids due to maintenance, faults, or re-routing.

To overcome this limitation and extend the capabilities of the model to accommodate dynamic changes in the system's topology, we introduce a key improvement. Instead of using the static LineMLP model, we design an Edge Graph Convolution (EGC) layer. The EGC layer is a more advanced component that leverages the inherent graph structure of the power network, effectively taking into account the topology of the system. By incorporating graph-based convolution operations, the EGC layer can dynamically adjust to changes in the network's structure, ensuring that the model can continue to make accurate predictions even when the power grid topology shifts.

This new architecture enables the model to adapt to changes in the power grid more effectively, as it learns not only from the relationships between power flows but also from the evolving structure of the network itself. The EGC layer thus represents a significant enhancement over the traditional LineMLP model, offering the ability to handle topology changes while maintaining high prediction accuracy for SA measurements. This enhancement brings the model closer to real-world applicability, where power networks are rarely static and constantly change in response to operational conditions.

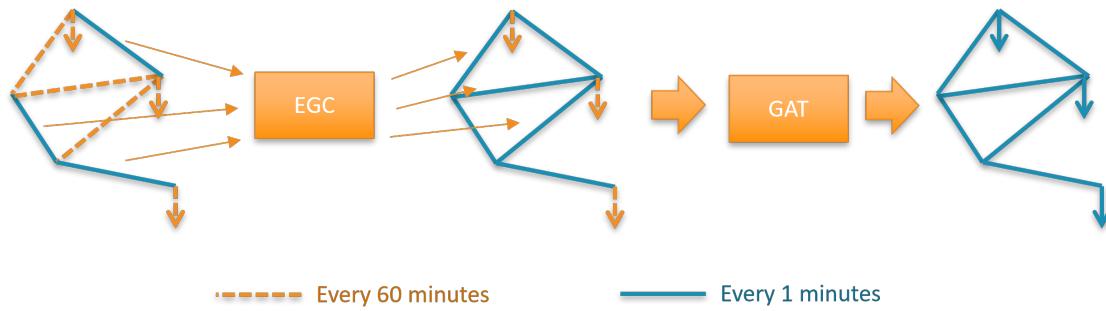


Figure 77: Overall framework for measurement synchronization.

Figure 77 illustrates the structure of the newly proposed model, which is designed to address the challenge of predicting power system parameters at a high temporal resolution, specifically at the 1-minute level. The model is composed of two primary components: Edge Graph Convolution (EGC) for predicting line-level parameters and Graph Attention Network (GAT) [28, 29] for node-level predictions. The EGC component focuses on providing accurate predictions for active and reactive power in transmission lines that do not have direct measurement data. To achieve this, it takes as input the network's topological information—such as the connectivity and relationships between different lines—and the FS measurements from known lines, which include both active and reactive power values at 1-minute intervals. These inputs are then processed to forecast the active and reactive powers for the lines that are not directly measured, which we refer to as SA lines. This approach allows the model to leverage the observed power flow in the system to infer the missing measurements for other lines.

The second key component of the model, the Graph Attention Network (GAT), is as discussed in the LineMLP part. It uses the FS measurements, including bus voltages, along with the predicted active and reactive powers of the lines generated by the EGC component. These inputs enable the GAT to estimate the missing node-level measurements for the SA nodes in the system. The use of the GAT is particularly advantageous because it allows the model to selectively focus on the most relevant parts of the graph structure by applying attention mechanisms, which help improve the accuracy of the predictions for the node-level measurements.

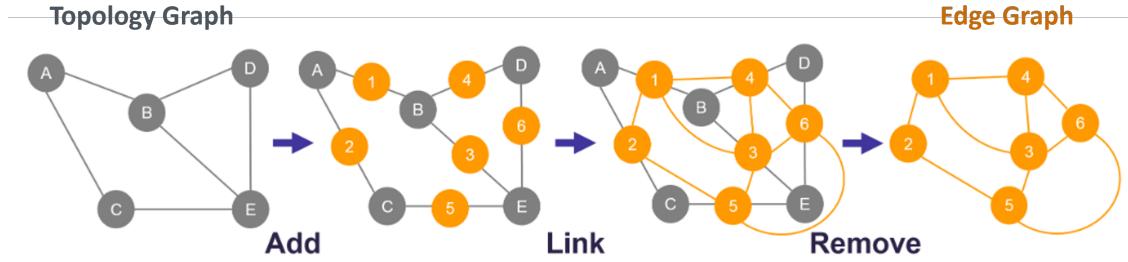


Figure 78: The construction of edge graph.

Moreover, because the available SA measurements are provided at 60-minute intervals, the model incorporates a correction mechanism to refine the 1-minute interval predictions. This is achieved by comparing the 1-minute interval predictions against the actual SA measurements obtained at the 60-minute mark. The model then adjusts the 1-minute level estimations accordingly to account for any discrepancies, ensuring that the final prediction is as accurate as possible. This continuous correction process helps maintain the accuracy of the model’s predictions over time, even as new data becomes available. In the following, a detailed breakdown of each component and the underlying methodologies used in this framework will be provided, offering further insights into how each part contributes to the overall predictive process.

The concept of the Edge Graph [30] is a novel approach derived from the original topology graph by transforming the edges of the topology graph into oriented nodes in the edge graph. In this new edge graph structure, each node corresponds to an edge in the original topology graph, and the edges in the edge graph represent the connections between two corresponding edges that share a common node in the original topology graph. The process of constructing this edge graph from the original graph involves several steps, as depicted in Fig. 78. These steps include three distinct operations: the Add step, the Link step, and the Remove step. During the Add step, a new node is created for each edge in the original topology graph, and this new node is uniquely represented by the two vertices that define the edge. In the Link step, the new nodes that correspond to edges sharing a common vertex in the original topology are connected to one another. Finally, in the Remove step, all of the original topology graph’s nodes and edges are deleted, leaving behind the newly formed nodes and edges that now make up the edge graph.

Given a graph G_0 , its edge graph $G_E = (A_E, X_E)$ is a graph such that: 1) each node of G_E represents an edge of G_0 and 2) two nodes of G_E are adjacent if their corresponding edges share a common endpoint in G_0 . Here A_E is the adjacency matrix and X_E is the feature matrix of G_E . Formally, two nodes in G_E are linked as expressed by the following equation:

$$A_{E(u,v)(u',v')} = \begin{cases} 1, & \text{if } v = u' \text{ and } u \neq v' \\ 0, & \text{otherwise} \end{cases}$$

where $u, u', v, v' \in V_0$ are nodes in G_0 , and $(u, v), (u', v') \in E_0$ denotes the edges in G_0 , i.e., the corresponding nodes in G_E . For an undirected graph G_0 , (u, v) and (v, u) denote the same edge. Thus the mapping from E_0 to V_E is one-to-one.

Edge Graph Convolution (EGC) leverages graph convolutional networks (GCN) [31] to process

the edge graph and predict the values of unobservable lines based on observable lines in the system. Initially, the model obtains hidden features for all lines using a Line Multilayer Perceptron (MLP), which is essentially a fully connected neural network layer used for predicting the data of SA lines (lines without measurements). Specifically, a two-layer MLP is employed to generate predictions for the SA lines based on the FS lines, as defined by the following equation:

$$\mathbf{e}_{\Theta_L}^{(l+1)} = \Phi(\mathbf{e}_{\Theta_L}^{(l)} \cdot \Theta_L^{(l)}), \quad (59)$$

where $\Theta_L^{(l)}$ is a layer-specific trainable weight matrix, $\mathbf{e}^{(l)}$ denotes the embeddings in the l -th layer. We use $\Phi(\cdot) = \text{ReLU}(\cdot)$ as the activation function for layers.

Next, the Laplacian matrix for the edge graph is computed using the following equation:

$$\hat{A}_0 = D^{-\frac{1}{2}} \overline{A'_0} D^{-\frac{1}{2}},$$

where D is the degree matrix of $\overline{A'_0}$. An identity matrix is not necessary if it is already considered in $\overline{A'_0}$.

For the convolution process itself, we consider a two-layer GCN, which performs graph convolution as defined by the following equation:

$$e = \sigma_1(\hat{A}_0 \sigma_0(\hat{A}_0 e_{\Theta_L} \theta_0 + b_0) \theta_1 + b_1),$$

where $\theta_0, \theta_1, b_0, b_1$ are learnable parameters, and $\sigma_0(\cdot), \sigma_1(\cdot)$ are activation function. Here we set $\sigma_0(\cdot)$ to be $\text{ReLU}(\cdot)$ and $\sigma_1(\cdot)$ to be $\text{Softmax}(\cdot)$.

The model employs Mean Squared Error (MSE) as the loss function to evaluate the accuracy of the line predictions, which is formulated as:

$$\mathcal{L}_L = \sqrt{\frac{1}{m} \sum_i^m (e_i - l_i)^2}, \quad (60)$$

where l_i denotes the ground truth of line i , and m is the number of lines. This loss function allows the model to minimize the difference between the predicted and actual values, leading to better performance in predicting the active and reactive power for the lines in the system. value of node i .

Combining Eq. (60) and Eq. (57), the whole framework is jointly optimized by the following function:

$$\min_{\Theta} \mathcal{L}_L + \beta \mathcal{L}_G, \quad (61)$$

where \mathcal{L}_L is used to improve predictions for lines, and \mathcal{L}_G is used to improve predictions for nodes, and β controls their relative weights.

To assess the performance of the proposed framework, we focus on analyzing data that corresponds to specific time periods when the topology of the power grid undergoes changes. In our evaluation, we down-sample the node active and reactive power injection data to 60-minute intervals, which are used as the SA measurement input for the model, sourced from the AMI (Advanced Metering Infrastructure). The original data collected at 1-minute intervals is retained only

Table 26: FS measurement prediction results with proportions of unobservable lines in terms of MAE.

Proportions of Lines		Naive		LineMLP		EGC-GAT	
With Measurements	Unobservable	Active Power	Reactive Power	Active Power	Reactive Power	Active Power	Reactive Power
56.7%	10%	7.7187	1.6588	2.2536	0.7562	1.9615	0.7297
38.8%	30%	7.7187	1.6588	6.5689	1.6098	3.1554	0.8194
24.6%	50%	7.7187	1.6588	6.7173	1.6466	4.0309	0.8936
14.2%	70%	7.7187	1.6588	6.9515	1.7934	5.6134	1.3105

for evaluation purposes. Meanwhile, the node voltage data is kept at 1-minute intervals, as these are considered FS measurements obtained from SCADA (Supervisory Control and Data Acquisition) systems. Regarding the line active and reactive power injection data, we select a portion of the lines to be designated as SA lines, for which the measurements are unobservable. These values are inferred based on nodal SA measurements, which are recorded at 60-minute intervals. The remaining lines continue to use FS measurements, recorded at 1-minute intervals.

For the training phase of the model, we utilize the first 70% of the data, which serves as the training set. The subsequent 10% is reserved for model validation, and the final 20% of the data is used as the test set, which is also employed as the reference set. To assess the prediction performance of the model, we compute the mean absolute error (MAE) for the predicted results.

In this study, we compare our model with two baseline methods: the Naive method and the LineMLP method. The Naive method simply uses the 60-minute-interval SA measurements as input and outputs an unchanged 1-minute-interval prediction over the course of the 60-minute intervals. The LineMLP method, which was developed in our previous report, leverages a LineMLP (Multi-Layer Perceptron) layer for line prediction, in combination with the GAT (Graph Attention Network) [28, 29] layer for node prediction. This method was shown to perform well in static topology scenarios, but it does not handle changes in topology effectively.

For our experiments, we evaluate the performance of the models on data that corresponds to topology-changing scenarios. The proportion of unobservable lines is varied, with settings of 10%, 30%, 50%, and 70% of lines being unobservable. The results of the predictions are summarized in Table 26. Despite the presence of unobservable lines, our proposed model consistently outperforms both the Naive method and the LineMLP method, achieving a relatively low error of less than 5%. It is important to note, however, that the performance of the LineMLP model in this report is lower than in the previous report. This decline is particularly evident in the reactive power prediction when the proportion of unobservable lines reaches 70%. The reduced performance can be attributed to the fact that the data used in this study is specifically designed to reflect topology-changing scenarios, which include a greater number of line-switching events. Moreover, it specifically selects a subset of lines that are unobservable, meaning their measurements cannot be inferred from nodal injection measurements due to zero power injection at certain nodes.

In contrast, the Naive method outperforms the LineMLP method in reactive power prediction when 70% of the lines are unobservable. However, this does not provide meaningful insight, as the Naive method does not actually offer a predictive mechanism. It simply outputs the same prediction for all 1-minute intervals within a 60-minute window, which makes it a less useful model for real-time prediction tasks. On the other hand, our current EGC (Edge Graph Convolution)

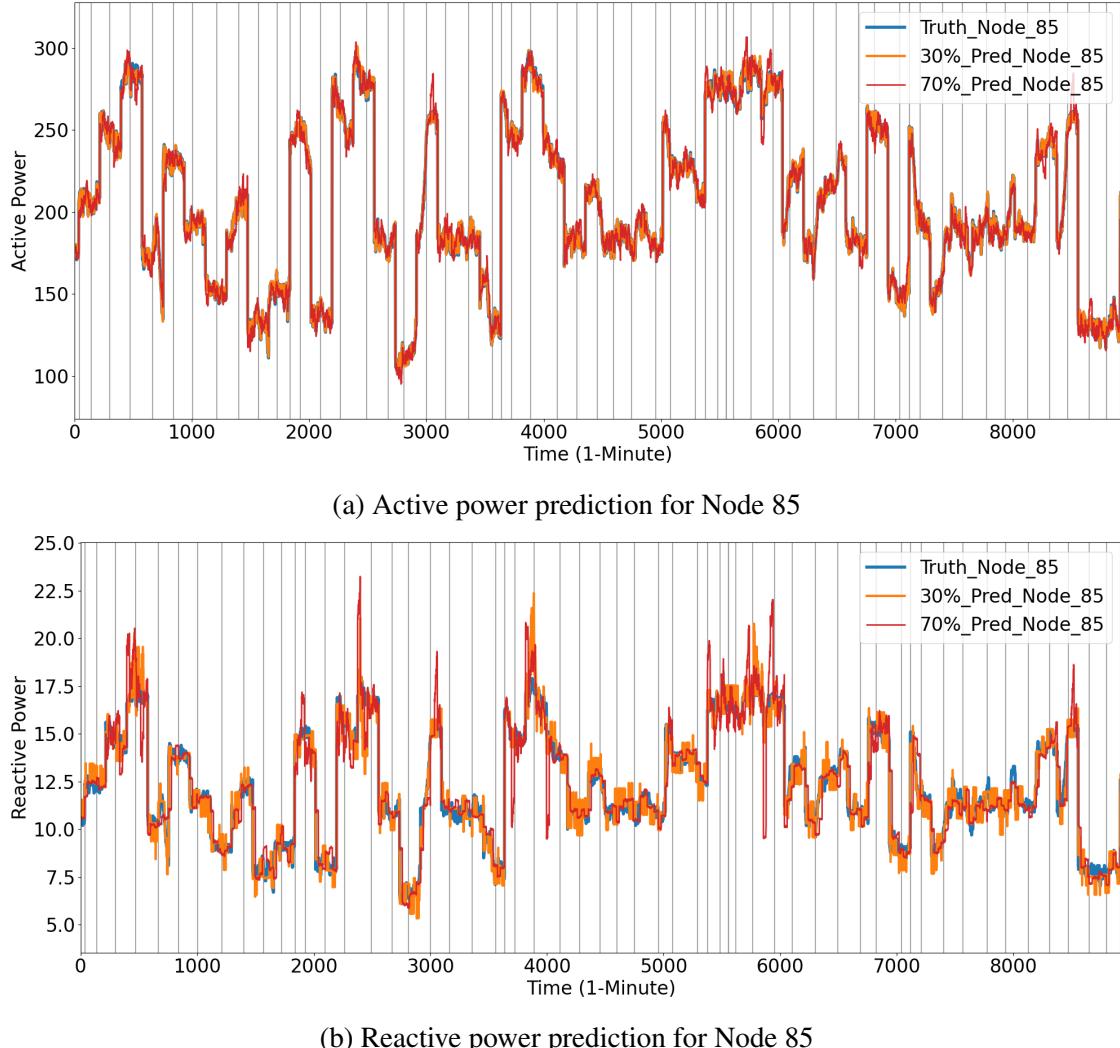


Figure 79: Prediction results for Node 85 with 30% and 70% unobservable lines.

method shows superior performance compared to both the Naive method and the LineMLP model. The EGC model incorporates topology information for both line prediction and node prediction, allowing it to adapt to topology changes and provide accurate predictions accordingly.

Figures 82 and 83 illustrate the topologies with 30% and 70% of the lines unobservable, respectively. In these figures, the black lines represent the FS lines, while the red lines denote the unobservable SA lines. Under these conditions, Figures 79, 80, and 81 show examples of prediction results for nodes that are connected to unobservable lines. The blue lines represent the ground truth, while the orange and red lines indicate the predictions made by our proposed model under the 30% and 70% unobservable line settings, respectively. The vertical gray lines in the figures mark the times when topology changes occurred.

From the displayed results, we observe that the predictions made when 30% of the lines are unobservable exhibit fewer anomalies compared to those made when 70% of the lines are unobservable. This difference can be attributed to the larger amount of observable data available in

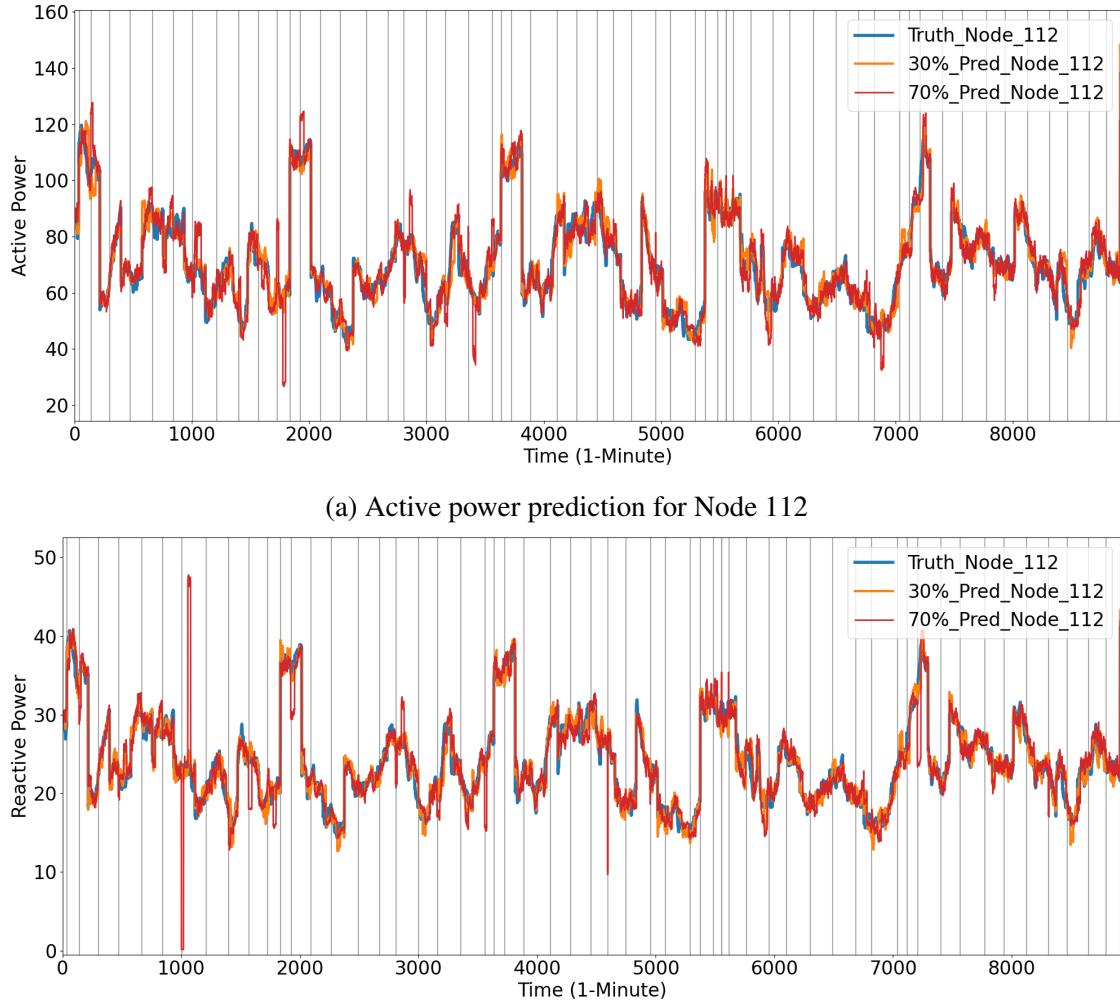
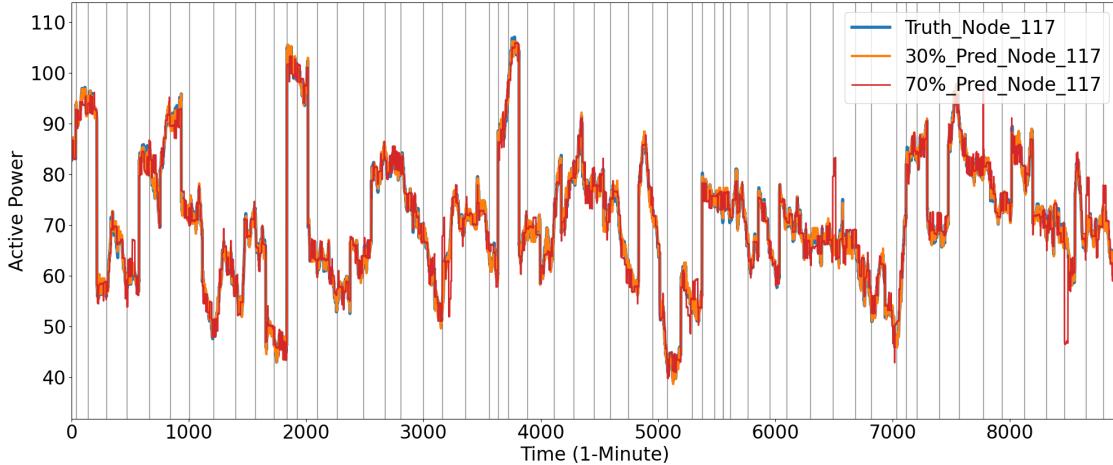


Figure 80: Prediction results for Node 112 with 30% and 70% unobservable lines.

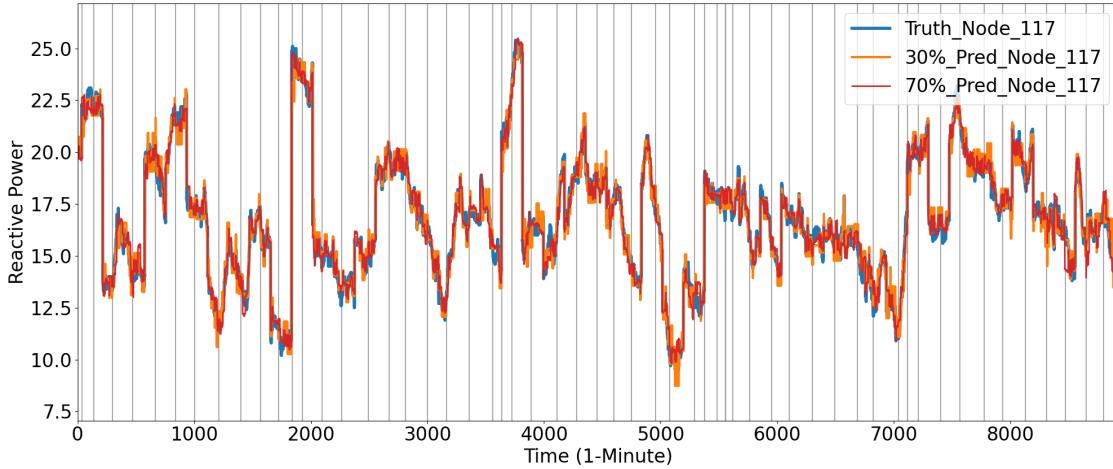
the 30% unobservable setting. Nevertheless, in both scenarios, the predictions remain close to the ground truth in most cases, demonstrating that the model can still deliver accurate predictions even with a substantial portion of the line data unobservable. This finding highlights the robustness and adaptability of the model, confirming that it remains effective even when faced with incomplete observational data and dynamic changes in the system’s topology.

Scenario: Clean Data + One-way Edge Information + Changing Topology. We further extend our analysis of PPM’s AMI data by enhancing the deep-learning model to address the challenge of predicting SA measurements with one-way line flow data when the system topology changes.

To evaluate the effectiveness of our proposed framework, we extract data corresponding to time periods when topology changes occur. For the evaluation, we down-sample the node active and reactive power injection data to 60-minute intervals, which are used as the input for the model’s SA measurements derived from the AMI. The original 1-minute interval data is retained only



(a) Active power prediction for Node 117



(b) Reactive power prediction for Node 117

Figure 81: Prediction results for Node 117 with 30% and 70% unobservable lines.

for evaluation purposes. The node voltage data are kept at 1-minute intervals, assuming these are FS measurements obtained from SCADA systems. Regarding the line active and reactive power injection data, we randomly select a subset of the lines to be classified as SA lines, for which there are no direct measurements. These unmeasured values can be inferred using nodal SA measurements (from AMI) at 60-minute intervals. The remaining lines continue to be represented with FS measurements at 1-minute intervals.

For model training, we use the first 70% of the data, reserving the next 10% for model validation. The final 20% of the data serves as the test set and reference set. The performance of the model is evaluated based on the mean absolute error (MAE).

To provide a baseline for comparison, we include a Naive method, which takes the 60-minute-interval SA measurements as input and outputs a prediction that remains unchanged throughout the 1-minute intervals during each 60-minute window. This method, while simple, serves as a useful comparison for evaluating the predictive capability of more sophisticated models.

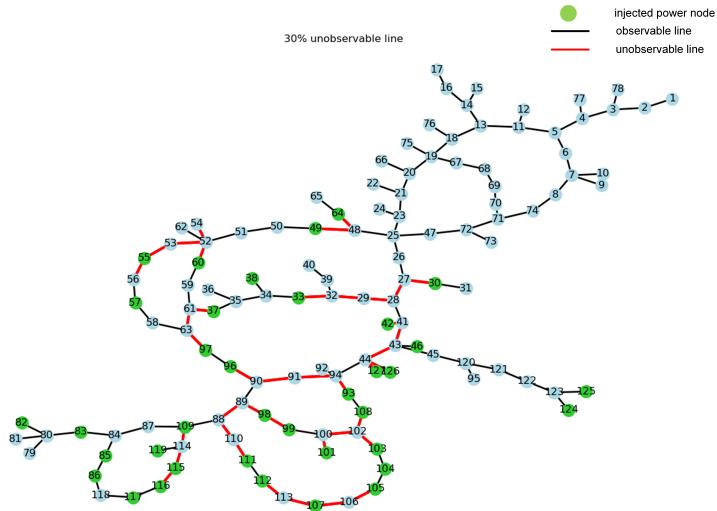


Figure 82: The system diagram of 30% unobservable lines.

Table 27: One-end line FS measurement prediction results in terms of MAE.

One-way Active Line Power				One-way Reactive Line Power			
SAME	ADD	MULTIPLY	RESET	SAME	ADD	MULTIPLY	RESET
6.2346	2.8341	2.6259	0.4339	6.3311	2.9360	3.2369	0.4476

For this evaluation, we focus on scenarios involving one-way line flow measurements, specifically assessing the model's ability to handle changing topologies with varying proportions of unobservable lines. The proportions of unobservable lines are set to 10%, 30%, 50%, and 70%.

Given that we only have one-way line flow data, for each measured line (n_a, n_b) , we need to infer the line power for the reverse direction, (n_b, n_a) , to improve the accuracy of our model's predictions. To accomplish this, we designed four distinct methods, as summarized in Table 27, to estimate the missing information for the reverse direction. These methods are as follows:

SAME: This method simply assumes that $(n_b, n_a) = (n_a, n_b)$, treating the line flows as symmetric.
ADD: This method adjusts (n_b, n_a) by adding the average difference between (n_b, n_a) and (n_a, n_b) in the training set: $(n_b, n_a) = (n_a, n_b) + \text{AVG}((n_b, n_a) \text{ on Training Set}) - \text{AVG}((n_a, n_b) \text{ on Training Set})$.
MULTIPLY: This method scales (n_b, n_a) by a ratio of the averages of (n_b, n_a) and (n_a, n_b) in the training set: $(n_b, n_a) = (n_a, n_b) * \text{AVG}((n_b, n_a) \text{ on Training Set}) / \text{AVG}((n_a, n_b) \text{ on Training Set})$.
RESET: This method adjusts the line flow every 60 minutes, correcting the 1-minute level estimates based on the available SA measurements. Among these methods, we found that the RESET method achieved the lowest error, with a prediction error of less than 1% for the unobservable one-way line data.

Using the predicted one-way line flow information, Table 28 shows the results for the prediction of all unmeasured SA measurements using our proposed machine learning model. The results

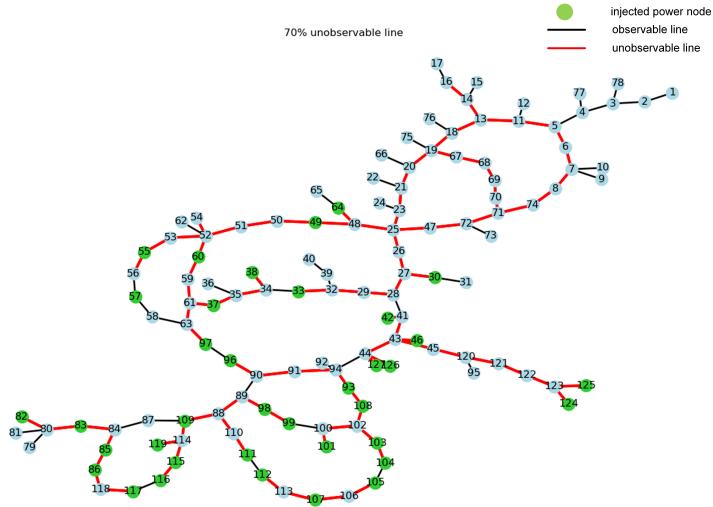


Figure 83: The system diagram of 70% unobservable lines.

Table 28: FS measurement prediction results with proportions of unobservable lines in terms of MAE.

Proportions of Lines		Naive		With All Data (Previous Version)		With One-way Data (Current Version)	
With Measurements	Unobservable	Active Power	Reactive Power	Active Power	Reactive Power	Active Power	Reactive Power
56.7%	10%	7.7187	1.6588	1.9615	0.7297	2.1080	0.7817
38.8%	30%	7.7187	1.6588	3.1554	0.8194	3.1242	0.7607
24.6%	50%	7.7187	1.6588	4.0309	0.8936	4.0757	0.9134
14.2%	70%	7.7187	1.6588	5.6134	1.3105	5.9366	1.4358

demonstrate that even when a portion of the line data is unobservable, our model outperforms the Naive method and achieves a relatively low error (less than 5%). Furthermore, when comparing the case with one-way line flow data to the case with bi-directional line flow information, the model's performance only decreases slightly—by less than 0.5%. This shows that our model can effectively handle one-way line flow data while maintaining high predictive accuracy, due to its ability to incorporate topology information for both line prediction and node prediction. The model is aware of topology changes and can adjust predictions accordingly.

Figures 83 displays the topology with 70% of the lines unobservable. In these figures, the black lines represent the FS lines, and the red lines denote the unobservable SA lines. Additionally, Figures 84, 85, and 86 illustrate examples of prediction results for nodes that are connected to unobservable lines. In these examples, the blue lines represent the ground truth, while the orange and red lines correspond to predictions made by the proposed model under the 70% unobservable line settings. The vertical gray lines indicate times when the topology changed.

From these figures, it is evident that the predictions remain close to the ground truth in most cases. The model demonstrates better accuracy with the 30% unobservable lines compared to the 70% unobservable lines, which can be attributed to the larger amount of observable data available

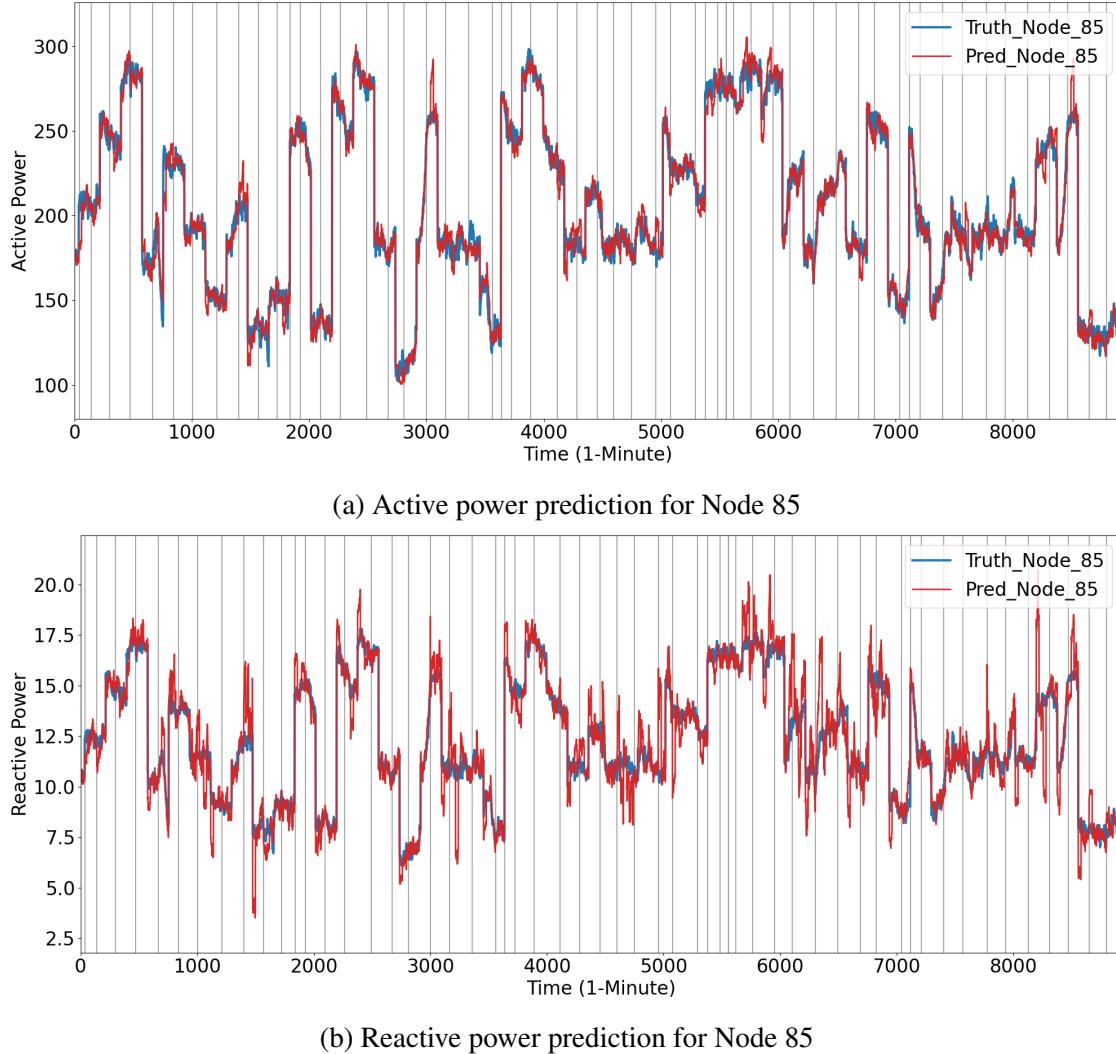
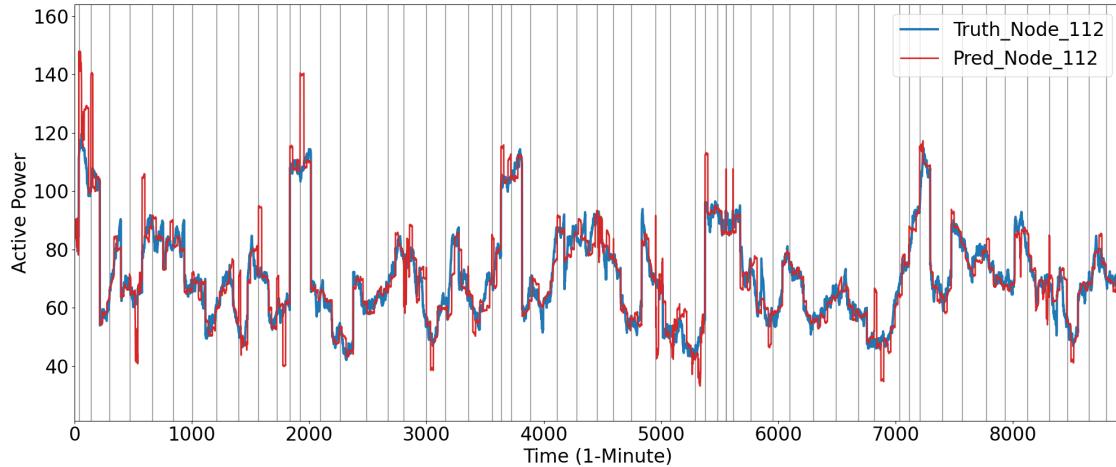


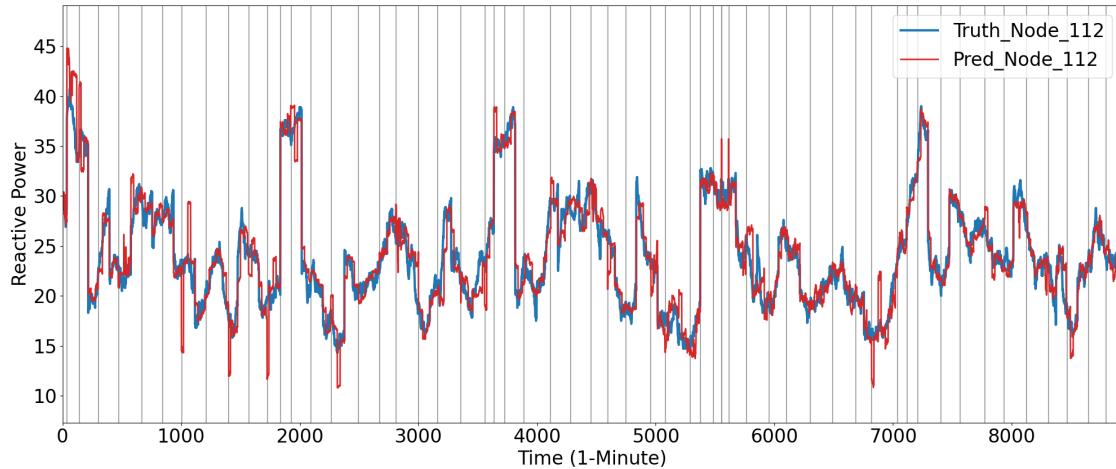
Figure 84: Prediction results for Node 85 with 70% unobservable lines.

in the 30% setting. Nevertheless, the model performs well under both conditions, validating that our proposed framework can effectively handle scenarios where part of the line data is unobservable and where one-way line flow information is available. The robustness of the model under such circumstances reinforces its practical applicability in real-world power grid management scenarios, where topology changes and incomplete measurements are common challenges.

From the results of Table 29, we can see that the Root Mean Square Errors (RMSEs) of SA measurement prediction are all smaller than the RMSEs of the power measurement in the different tests of unobservable lines. The RMSE of power measurement is the Root Mean Square Error (RMSE) between the true SA measurement with 1-minute interval and the measured SA measurement with 60-minutes interval. The RMSE of the SA measurement prediction is the Root Mean Square Error (RMSE) between the true SA measurement with 1-minute interval and the predicted SA measurement with 1-minute interval. The percentage for power measurement is calculated by $\frac{\text{RMSE of the SA measurement prediction}}{\text{RMSE of power measurement}} \times 100$. For example, as shown in the Table 29, for active power, the



(a) Active power prediction for Node 112



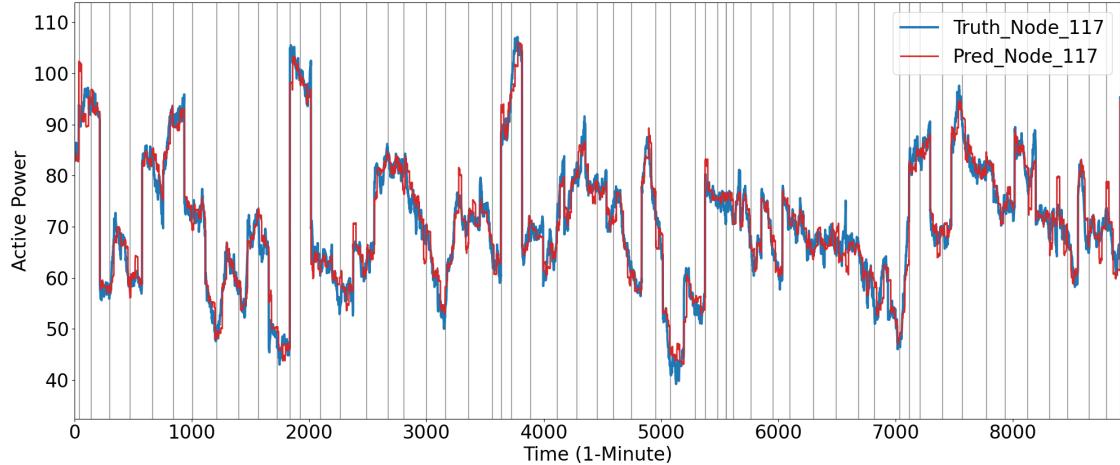
(b) Reactive power prediction for Node 112

Figure 85: Prediction results for Node 112 with 70% unobservable lines.

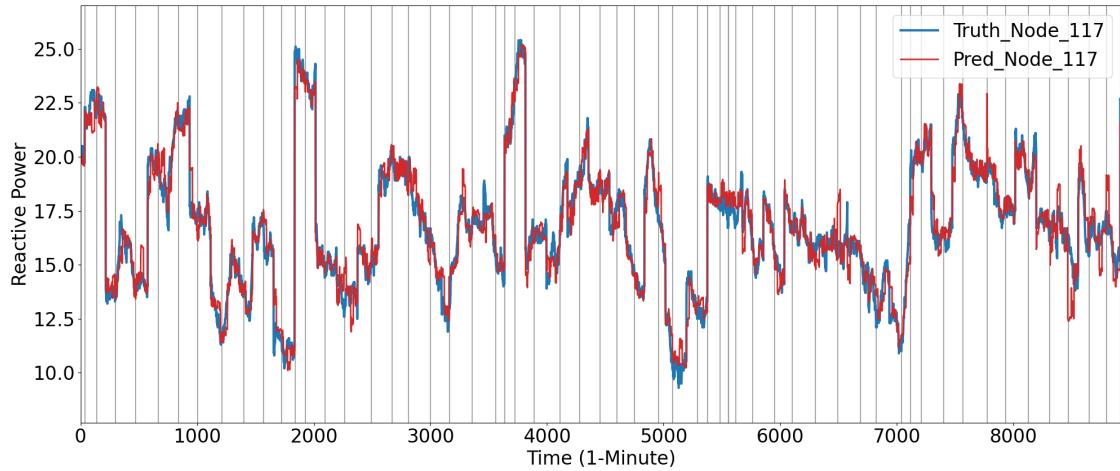
percentage for power measurement is $\frac{0.0021}{0.0185} = 11.35\% < 35\%$ in the test of 10% unobservable lines and or active power, the percentage for power measurement is $\frac{0.0034}{0.0185} = 17.00\% < 35\%$ in the test of 10% unobservable lines, satisfying **Milestone 3.12.1**. However, in the test of the 70% unobservable lines, the percentages of the active and reactive power measurements exceed 35% of the power measurements, with values of 54.05% and 60.42% respectively. Figure 87 are examples of RMSE results of SA power measurements with 10% unobservable lines, which show most of the SA power measurement prediction is closer to the true SA power measurement compared with the SA power measurements.

Validation of Closed Loop Operation of Measurement Predictor and State Estimator.

Before we carry on the validation of closed loop operation, we introduce the well-known robust WLAV estimator. In the balanced three-phase power system, the steady state of the balanced three-phase distribution system is described by the AC power flow equation. The equations are derived from Kirchhoff's Current Law (KCL), Kirchhoff's Voltage Law (KVL) and Ohm's Law in AC



(a) Active power prediction for Node 117



(b) Reactive power prediction for Node 117

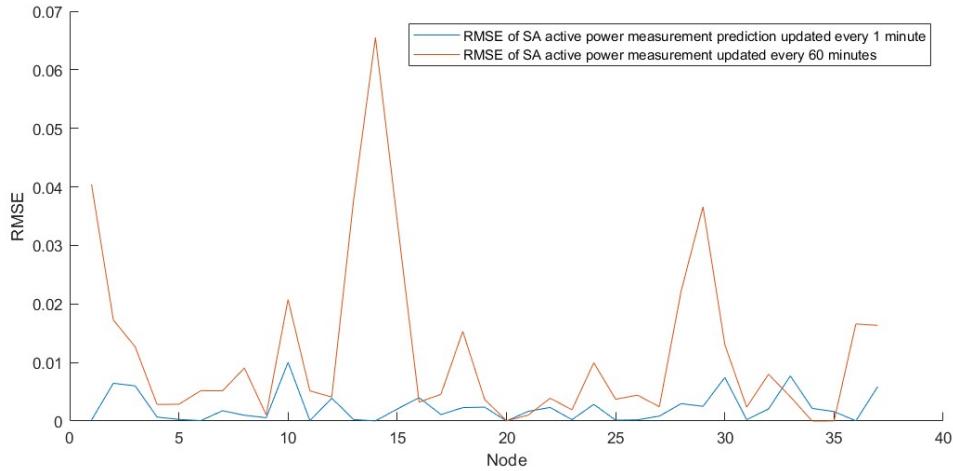
Figure 86: Prediction results for Node 117 with 70% unobservable lines.

circuits, and can be summarized as follows:

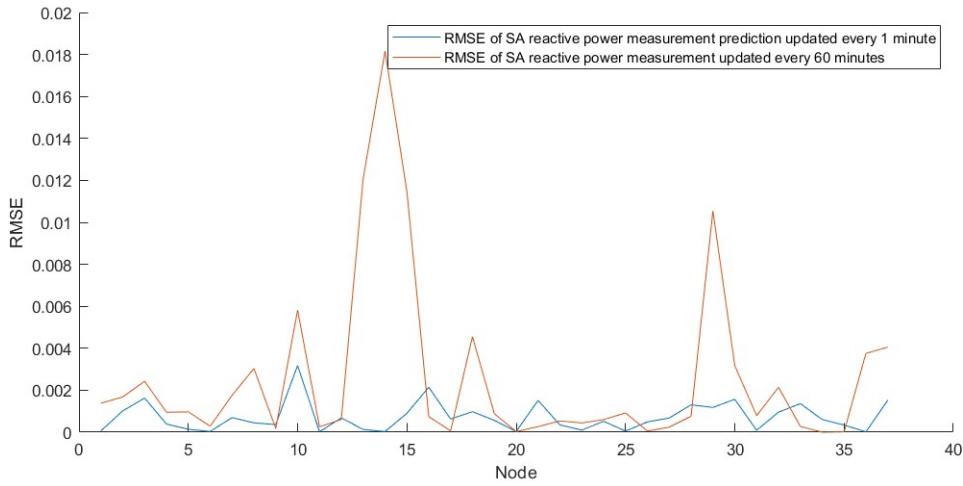
$$P_i(t) = V_i(t) \sum_{j \in S_i^B} V_j(t) (G_{ij} \cos(\theta_i(t) - \theta_j(t)) + B_{ij} \sin(\theta_i(t) - \theta_j(t))), \quad (62)$$

$$Q_i(t) = V_i(t) \sum_{j \in S_i^B} V_j(t) (G_{ij} \sin(\theta_i(t) - \theta_j(t)) + B_{ij} \cos(\theta_i(t) - \theta_j(t))), \quad (63)$$

$$P_{ij}(t) = V_i^2(t) (g_{si} + g_{ij}) - V_i(t) V_j(t) (g_{ij} \cos(\theta_i(t) - \theta_j(t)) + b_{ij} \sin(\theta_i(t) - \theta_j(t))), \quad (64)$$



(a) RMSE of SA active power measurement in the test of 10% unobservable lines



(b) RMSE of SA reactive power measurement in the test of 10% unobservable lines

Figure 87: RMSE of SA power measurement with 10% unobservable lines.

$$Q_{ij}(t) = -V_i^2(t)(b_{si} + b_{ij}) - V_i(t)V_j(t)(g_{ij}\sin(\theta_i(t) - \theta_j(t)) + b_{ij}\cos(\theta_i(t) - \theta_j(t))), \quad (65)$$

where $P_i(t)$ and $Q_i(t)$ are the active and reactive powers injected into bus i at time t ; $P_{ij}(t)$ and $Q_{ij}(t)$ are the active and reactive power flows from the bus i to the bus j at time t ; $V_i(t)$ and $\theta_i(t)$ are the voltage magnitude and phase angle of bus i at time t ; $G_{ij} + jB_{ij}$ is the ij th entry of the complex bus admittance matrix; $g_{ij} + jb_{ij}$ is the series admittance between buses i and j ; $g_{si} + jb_{si}$ is the shunt admittance connected to bus i ; S_i^B is the set of bus numbers that are directly connected to bus i .

The measurement equations of the SE model of networks, i.e., the nonlinear equations between the state variables and the measured variables in SE can be constructed using Eqs. (62) - (65):

Table 29: Root Mean Square Error (RMSE) of SA measurement prediction

Proportions of Lines		RMSE of Power Measurements (Naive)		RMSE of SA Measurement Prediction		Percentage for Power Measurements	
With Measurements	Unobservable	Active Power (p.u.)	Reactive Power (p.u.)	Active Power (p.u.)	Reactive Power (p.u.)	Active Power	Reactive Power
56.7%	10%	0.0185	0.0048	0.0021	8.16x10⁻⁴	11.35%	17.00%
38.8%	30%	0.0185	0.0048	0.0034	9.98x10⁻⁴	18.38%	20.78%
24.6%	50%	0.0185	0.0048	0.0041	0.001	22.16%	20.83%
14.2%	70%	0.0185	0.0048	0.01	0.0029	54.05%	60.42%

$$z = h_{phy}(x) + e, \quad (66)$$

where z denotes the vector of the selected measurement variables which include bus voltage magnitudes, active and reactive power injections, and active and reactive power flows, where the total number of measurements is N ; x represents the vector of state variables to be estimated, which is defined as the magnitudes and angles of voltages at all nodes; vector function $h_{phy}(x)$ relates the measurements to the state variables, and e represents the vector of measurement errors, which is traditionally assumed as a normally distributed random vector with zero means and known variances.

The WLAV algorithm is known to be robust against gross errors in measurements. It is the absolute values of the residuals in the objective function, and can be expressed as the solution to the l_1 optimization problem below:

$$\{\hat{x}(t)\} = \arg \min_{\hat{x}(t)} \left\| (R)^{-1/2}(t) [z(t) - h_{phy}(x(t))] \right\|_1. \quad (67)$$

To solve Eq. (67), we need to linearize this problem as follows [29]:

$$\begin{cases} \arg \min_x \left\| (R)^{-1/2}(t) r \right\|_1, \\ s.t. r = z(t) - h_{phy}(x(t)), \\ \Delta z = H \Delta x + r, \end{cases} \quad (68)$$

where H is the Jacobian matrix of the measurement function. Eq. (68) can be transformed and solved using well-developed linear programming (LP) algorithms such as the simplex algorithm [30, 31]:

$$\begin{cases} \arg \min_x c^T (r^+ + r^-), \\ s.t. \Delta z = H \Delta x^+ - H \Delta x^- + r^+ - r^-, \\ \Delta x^+, \Delta x^-, r^+, r^- \geq 0, \end{cases} \quad (69)$$

where c is the diagonal vector of $(R)^{-1/2}$; $r = r^+ - r^-$; and $\Delta x = \Delta x^+ - \Delta x^-$; $r^+, r^-, \Delta x^+$ and Δx^- are non-negative auxiliary variables used to convert the WLAV problem to the LP problem.

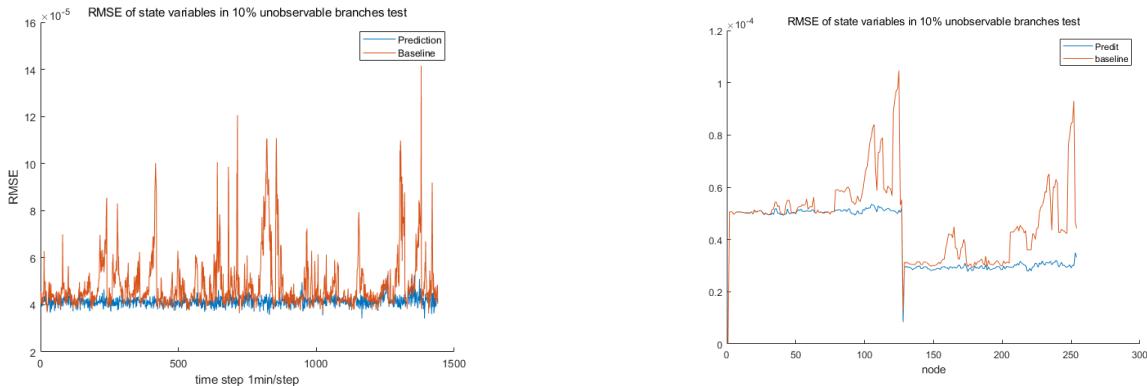
We use the WLAV state estimator for the evaluation of the proposed closed-loop operation framework. The data used is clean data. We down-sample the node active and reactive power injection data to 60-minute intervals as the SA measurement input of the model from AMI, and the 1-minute interval data predicted by the deep learning method is only used for comparison. The

line active and reactive power data and node voltage data remain at 1-minute intervals as they are assumed to be FS measurements from SCADA.

The tests for state estimation carry on four times according to the 10%, 30%, 50%, and 70% unobservable branches in measurement configuration. The RMSEs of state variables are used to evaluate the performance of SE. The Figs.96, 97, 98, and 99, respectively show the RMSE of state variables during the first several days in the measurement configuration of the 10%, 30%, 50%, and 70% unobservable branches. In these figures, the RMSE of voltage angle are in the first 127 nodes of x axis and the RMSE of voltage magnitude are in the last 127 nodes of x axis. Table 30 shows the overall performance and improvement of the prediction and baseline. From the table 30, we can conclude that the more branches are unobservable, the more improvement. This is due to the fact that when the SCADA measurements are fewer, the state estimator relies more on the AMI measurements, and the higher accuracy of the predicted AMI measurements than the baseline takes more effect. However, if the unobservable branches are over many (like 70% unobservable), the prediction accuracy can be impacted. Thus, the degree of improvement may not continue to grow.

Table 30: SE result of RMSE of state variables in different measurement configuration

Test	RMSE of Baseline	RMSE of Prediction	improvement
10%	4.88×10^{-5}	4.14×10^{-5}	15.2%
30%	5.00×10^{-5}	4.32×10^{-5}	13.6%
50%	2.96×10^{-4}	7.02×10^{-5}	76.3%
70%	5.16×10^{-4}	1.66×10^{-4}	67.8%

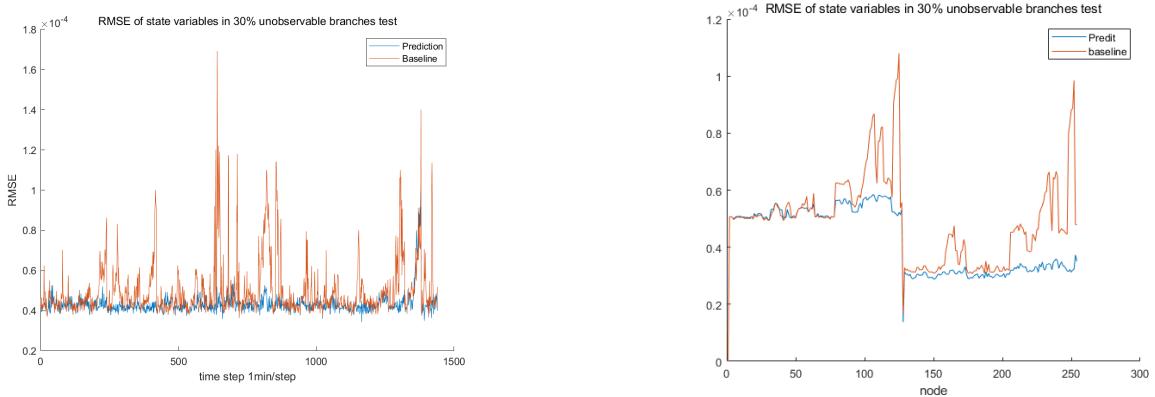


(a) RMSEs of state estimates over time steps in 10% unobservable branches.

(b) RMSEs of state estimates over node numbers in 10% unobservable branches.

Figure 88: State estimation result under 10% unobservable branches.

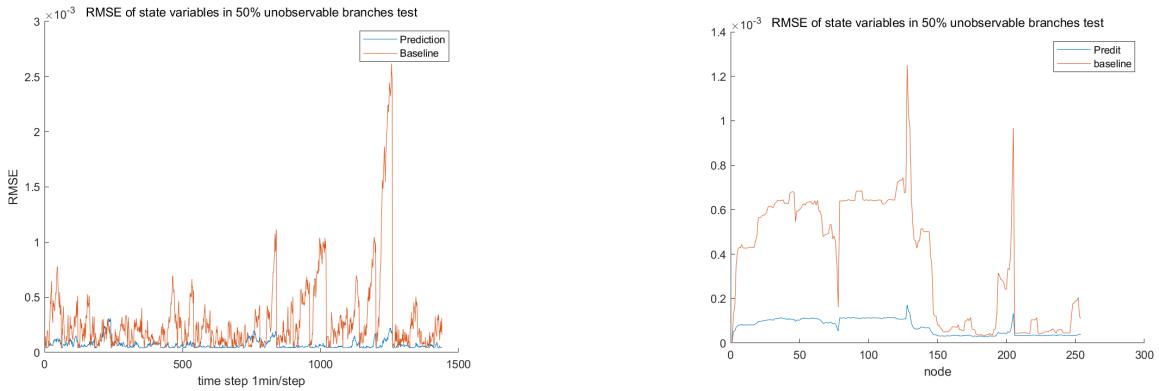
From the results of Table 31, we can see that the Root Mean Square Errors (RMSEs) of voltage estimations are all smaller than the RMSEs of measured voltage in the different tests of unobservable lines. RMSEs of voltage estimation are all smaller than alpha% which is defined as the standard deviation of the Gaussian noise of voltage measurement with the value of 4×10^{-4} , satisfying **Milestone 3.12.2**. Figure 92, 93, 94 and 95 are examples of RMSE results of voltage



(a) RMSEs of state estimates over time steps in 30% unobservable branches.

(b) RMSEs of state estimates over node numbers in 30% unobservable branches.

Figure 89: State estimation result under 30% unobservable branches.



(a) RMSEs of state estimates over time steps in 50% unobservable branches.

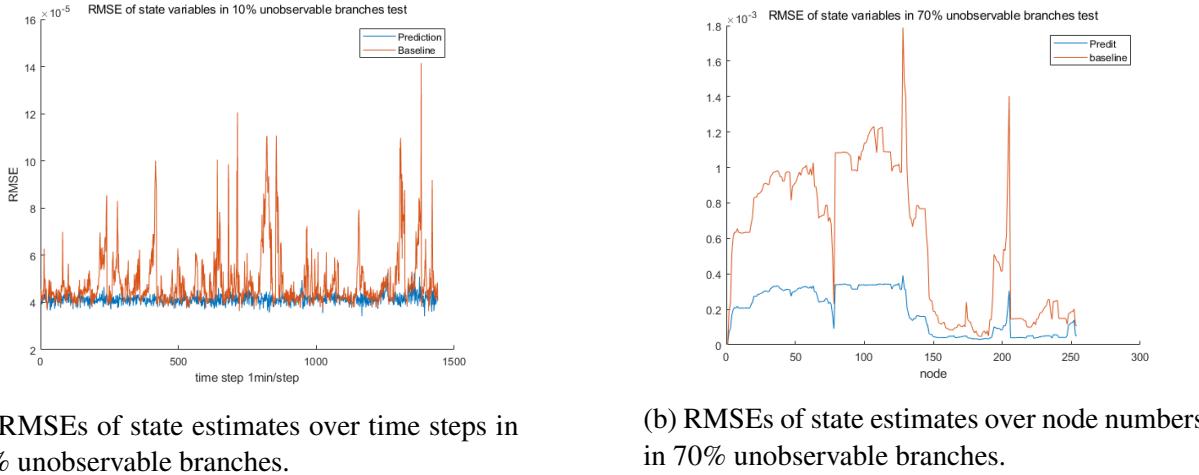
(b) RMSEs of state estimates over node numbers in 50% unobservable branches.

Figure 90: State estimation result under 50% unobservable branches.

estimaion and voltage measurements with 10%, 30%, 50% and 70% unobservable lines, which show most of the voltage estimation is closer to the true voltage value compared with the voltage measurements.

Scenario: Corrupted Data + One-way Edge Information + Changing Topology. we further evaluate the performance of our EGC-GAT model using various datasets, including clean data, corrupted data, and residual-corrected data generated from the WLAV state estimator. This is done to assess the proposed closed-loop operation framework under conditions involving measurement noise.

As in previous evaluations, we extract data around times when the topology of the system changes. We down-sample the node active and reactive power injection data to 60-minute intervals to serve as the SA measurement input from AMI to the model. The 1-minute interval data is only used for evaluation purposes. The node voltage data are kept at 1-minute intervals, as they are



(a) RMSEs of state estimates over time steps in 70% unobservable branches.

(b) RMSEs of state estimates over node numbers in 70% unobservable branches.

Figure 91: State estimation result under 70% unobservable branches.

Table 31: RMSE of Voltage Estimations

Proportions of Lines		RMSE of Estimated Voltage (p.u.)	RMSE of Measured Voltage (p.u.)
56.7%	10%	9.04x10⁻⁵	4x10 ⁻⁴
38.8%	30%	1.28x10⁻⁴	4x10 ⁻⁴
24.6%	50%	1.15x10⁻⁴	4x10 ⁻⁴
14.2%	70%	1.02x10⁻⁴	4x10 ⁻⁴

assumed to be FS measurements from SCADA. For the line active and reactive power data, we randomly select a subset of lines as SA lines, which do not have direct measurements, but whose values can be inferred from nodal SA measurements (AMI) at 60-minute intervals. The remaining lines continue to have FS measurements at 1-minute intervals.

For model training, we use the first 70% of the data. The next 10% is used for model validation, and the last 20% serves as the test and reference set. The evaluation metric used for model performance is the mean absolute error (MAE). A Naive method is also included for comparison, which uses the 60-minute-interval SA measurements as input and outputs unchanged predictions for the 1-minute intervals during each 60-minute window. We evaluate the models over one-end line flow measurements, with proportions of unobservable lines set to 10%, 30%, 50%, and 70%, and we present the results in Table 32.

We introduce the WLAV state estimator to evaluate the proposed closed-loop operation framework, incorporating measurement noise.

Validation of Closed Loop Operation of Measurement Predictor and State Estimator. We use the WLAV state estimator for the evaluation of the proposed closed-loop operation framework under condition of the measurement noise while in the first quarter, we do SE under the ideal condition. The data used is corrupted data. We down-sample the node active and reactive power injection data to 60-minute intervals as the SA measurement input of the model from AMI, and the 1-minute interval data predicted by the deep learning method is only used for comparison. The line active and reactive power data and node voltage data remain at 1-minute intervals as they are assumed to be FS measurements from SCADA. The standard deviation of measurements are 1% of

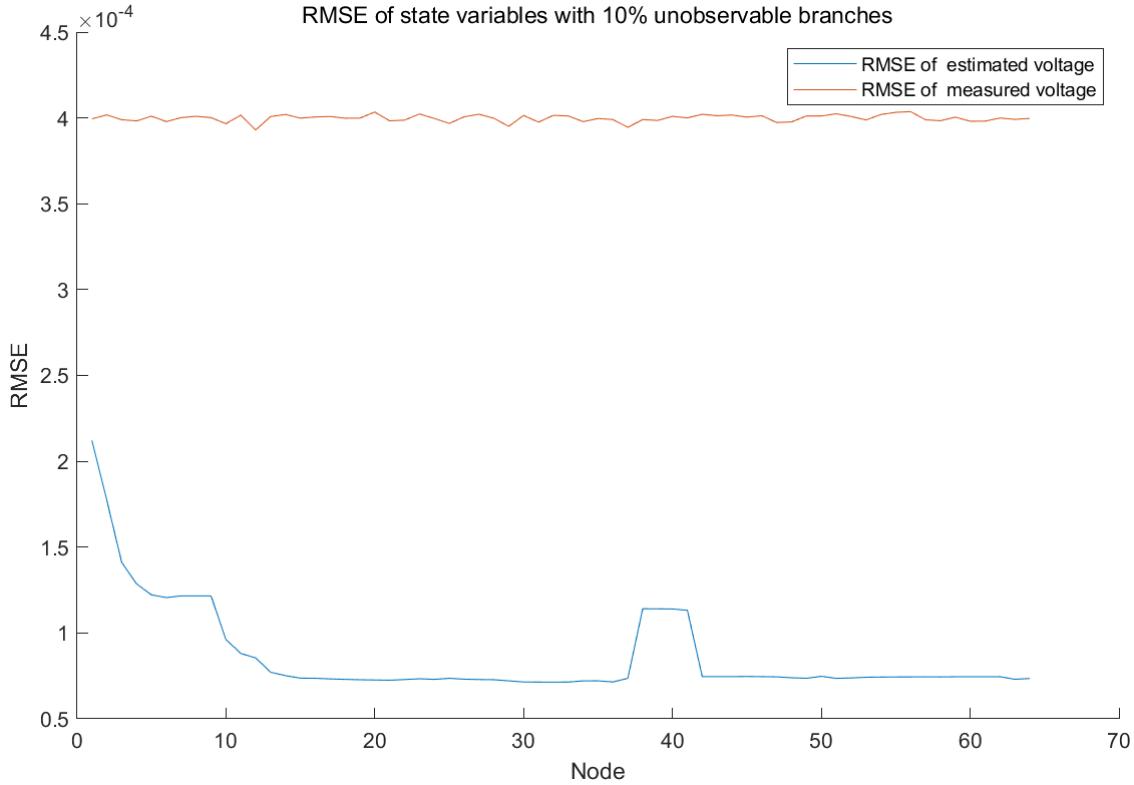


Figure 92: RMSE of Voltage Estimations in the test of 10% unobservable line.

value of voltage magnitude plus 8×10^{-5} p.u. for voltage magnitude and 2% of value of line power plus 8×10^{-5} p.u. for line power respectively.

The tests for state estimation carry on four times according to the 10%, 30%, 50%, and 70% unobservable branches in measurement configuration. The number of total time steps are 89280. The RMSEs of state variables are used to evaluate the performance of SE. The Figs.96, 97, 98, and 99, respectively show the RMSE of state variables during the first several days in the measurement configuration of the 10%, 30%, 50%, and 70% unobservable branches. In these figures, the RMSE of voltage angle are in the first 127 nodes of x axis and the RMSE of voltage magnitude are in the last 127 nodes of x axis. Table 33 shows the overall performance and improvement of the prediction and baseline. From the table 33, we can conclude that the improvements of SE in different test are all above 80%. Due to the measurement noise, the more branches are unobservable, the less improvement. That is because more measurements with noise can help the WLAV estimator be biased to the better measurements to improve the performance of SE.

Subtask 12.2: Use of Project Partner's Testing Lab – Technology Validation in HIL Environment

In this subtask, our purpose is to validate the effectiveness of observability improvement in state estimation with the SA measurement prediction. We test our proposed model in HIL Environment by use of project partner's testing Lab. The test is performed in the 90-node test system with 90 branches. At the first time, the project partner achieved the 30-minutes-level PMU measurements including the voltage magnitude, voltage angle, active and reactive power flow in the HIL environment. After receiving the data from the project partner, we use the linear state estimation model to

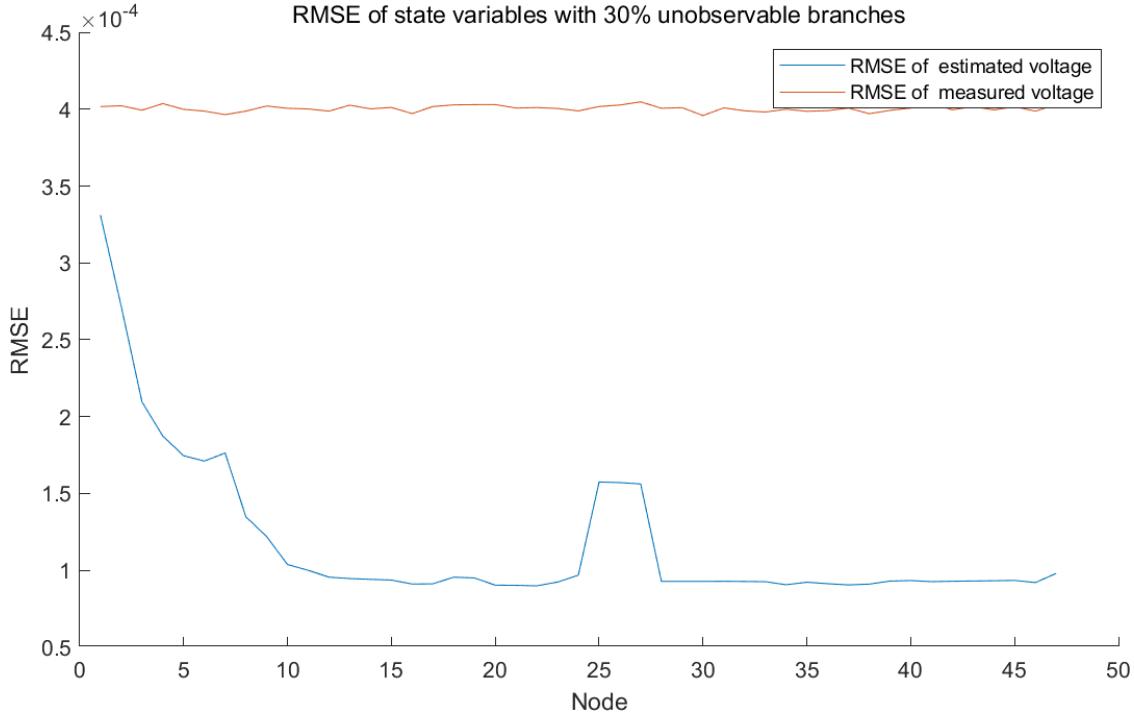


Figure 93: RMSE of Voltage Estimations in the test of 30% unobservable line.

Table 32: FS measurement prediction results with different data and proportions of unobservable lines in terms of MAE.

Proportions of Lines		With Clean Data		With Corrupted Data		With Residual-corrected Data	
With Measurements	Unobservable	Active Power	Reactive Power	Active Power	Reactive Power	Active Power	Reactive Power
56.7%	10%	4.6042	1.1026	15.5226	3.7646	13.0072	3.6437
38.8%	30%	5.3965	1.3293	15.1179	3.6609	13.3372	3.8287
24.6%	50%	5.8813	1.4225	15.2224	3.4919	15.0830	3.4458
14.2%	70%	8.0902	1.9377	15.6761	3.8969	16.9295	4.1571

estimate the node active and reactive power injection. Then, we perform the power flow analysis to generate the data of true voltage magnitude, voltage angle, active and reactive power flow with the interval of 30 minutes. Finally, we preprocess all the data as mentioned before and use the interpolation method to achieve the 5-minute-level data as true data.

Scenario: Clean Data + One-way Edge Information. We perform our analysis by enhancing the deep-learning model to address the challenge of predicting SA measurements with one-way line flow data.

For the evaluation, we down-sample the node active and reactive power injection data to 60-minute intervals, which are used as the input for the model's SA measurements derived from the AMI. The original 5-minute interval data is retained only for evaluation purposes. The node voltage data are kept at 5-minute intervals, assuming these are FS measurements obtained from SCADA systems. Regarding the line active and reactive power injection data, we randomly select a subset of the lines to be classified as SA lines, for which there are no direct measurements. These un-

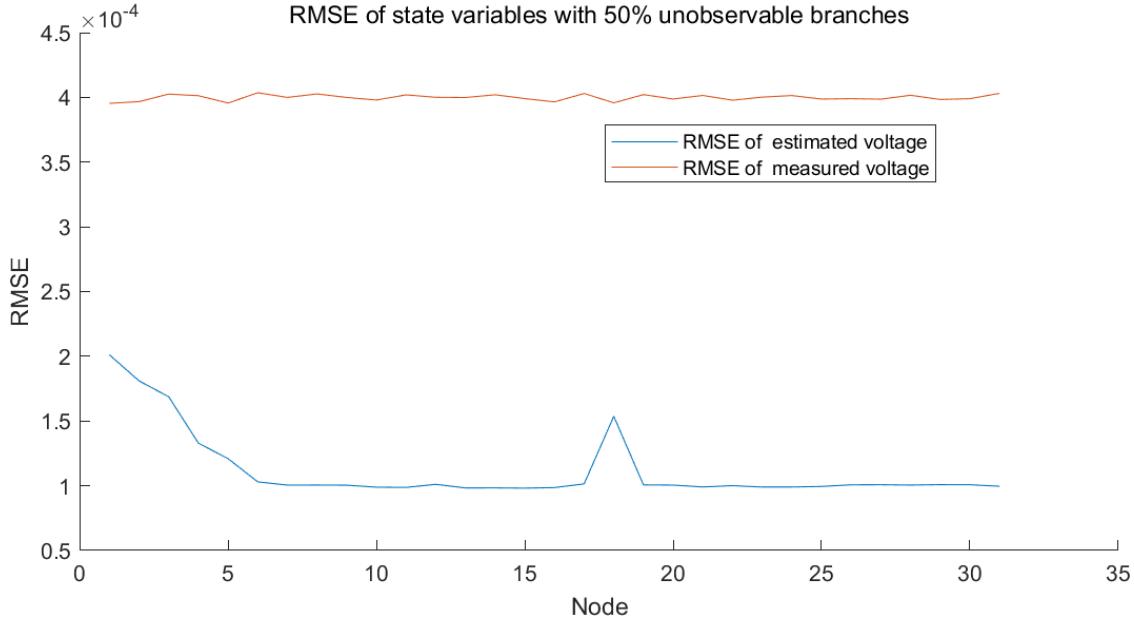


Figure 94: RMSE of Voltage Estimations in the test of 50% unobservable line.

Table 33: SE result of RMSE of state variables in different measurement configuration

Test	RMSE of Prediction	RMSE of Baseline	improvement
10%	7.07×10^{-5}	6.52×10^{-4}	89.2%
30%	1.06×10^{-4}	6.73×10^{-4}	84.3%
50%	1.11×10^{-4}	6.36×10^{-4}	82.5%
70%	1.15×10^{-4}	6.53×10^{-4}	82.4%

measured values can be inferred using nodal SA measurements (from AMI) at 60-minute intervals. The remaining lines continue to be represented with FS measurements at 5-minute intervals.

For model training, we use the first 70% of the data, reserving the next 10% for model validation. The final 20% of the data serves as the test set and reference set. The performance of the model is evaluated based on the mean absolute error (MAE).

To provide a baseline for comparison, we include a Naive method, which takes the 60-minute-interval SA measurements as input and outputs a prediction that remains unchanged throughout the 5-minute intervals during each 60-minute window. This method, while simple, serves as a useful comparison for evaluating the predictive capability of more sophisticated models.

For this evaluation, we focus on scenarios involving one-way line flow measurements, specifically assessing the model's ability to handle varying proportions of unobservable lines. The proportion of unobservable lines is set to 30%.

Using the predicted one-way line flow information, Table 34 shows the results for the prediction of 30% unmeasured SA measurements using our proposed machine learning model. The results demonstrate that even when a portion of the line data is unobservable, our model outperforms the Naive method and achieves a relatively low error (less than 5%). Furthermore, when comparing the case with one-way line flow data to the case with bi-directional line flow information, the

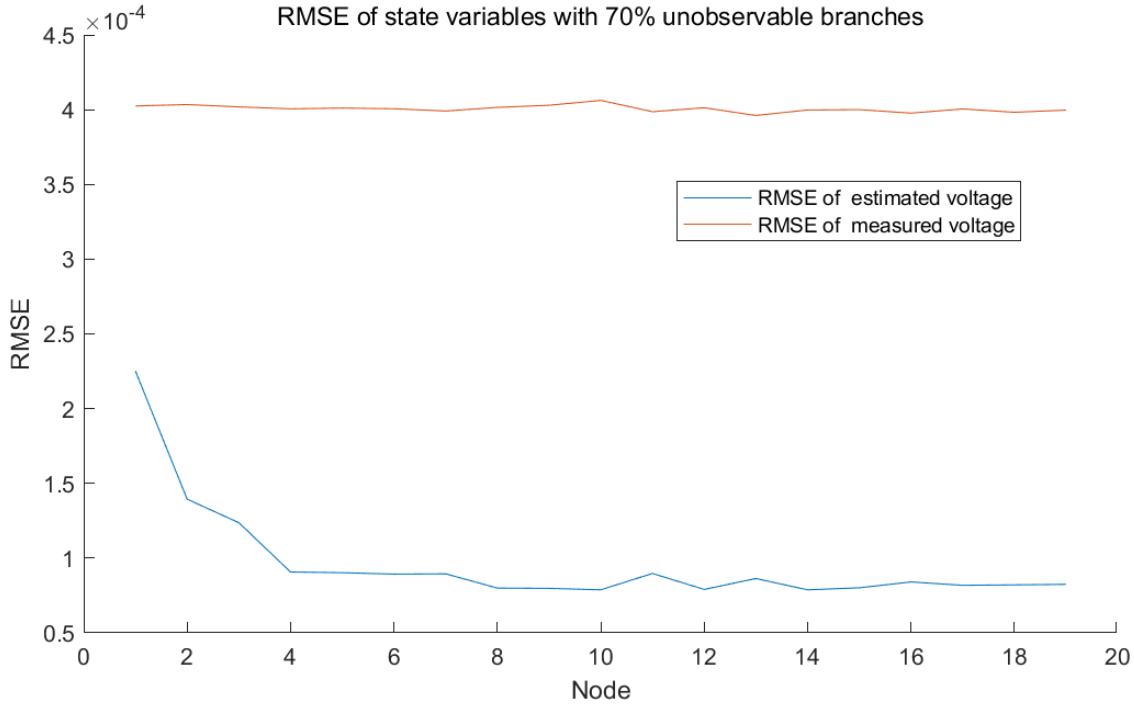
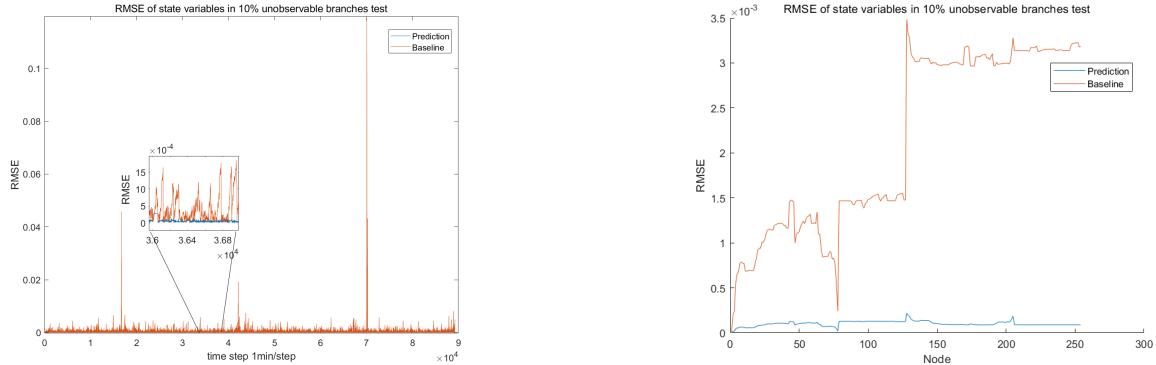


Figure 95: RMSE of Voltage Estimations in the test of 70% unobservable line.



(a) RMSEs of state estimates over time steps in 10% unobservable branches.

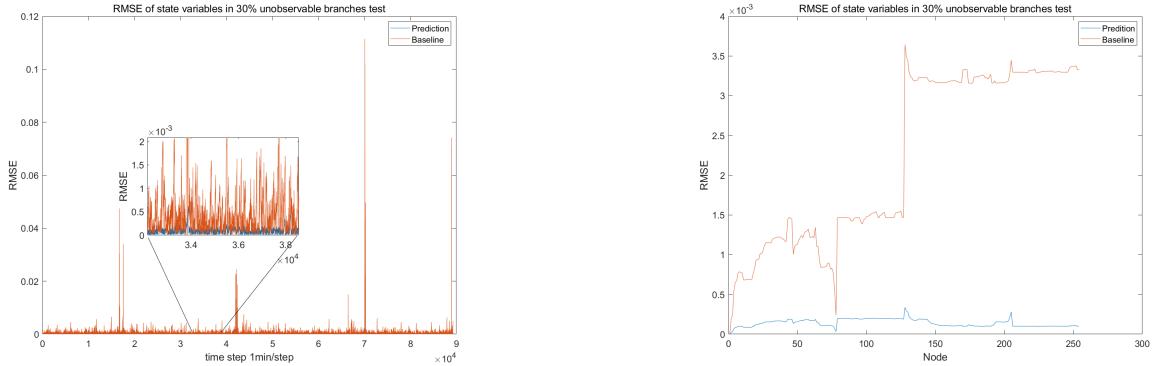
(b) RMSEs of state estimates over node numbers in 10% unobservable branches.

Figure 96: State estimation result under 10% unobservable branches.

model's performance only decreases slightly—by less than 0.5%. This shows that our model can effectively handle one-way line flow data while maintaining high predictive accuracy, due to its ability to incorporate topology information for both line prediction and node prediction.

The test of HIL environment: 4e-4 for voltage ,8e-5 for power flow, 0 for power injection

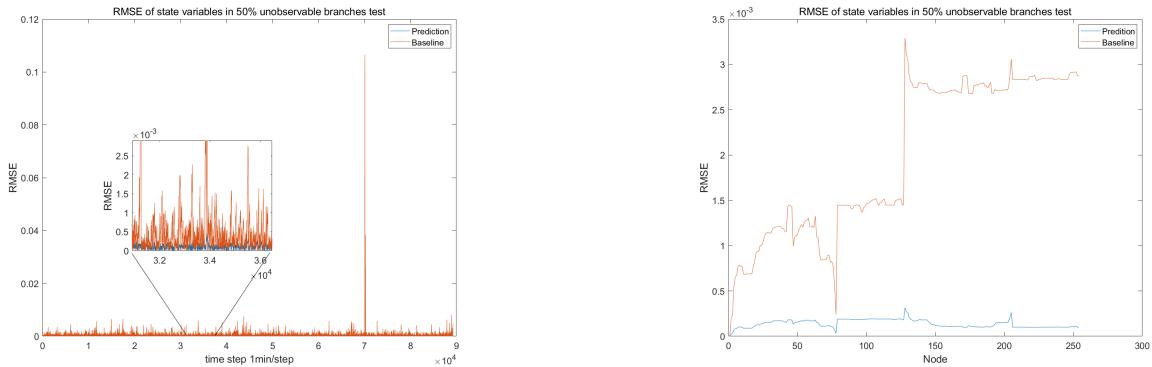
Additionally, figures 100, 101, and 102 illustrate examples of prediction results for nodes that are connected to unobservable lines. In these examples, the blue lines represent the ground truth, while the red lines correspond to predictions made by the proposed model under the 30% unobservable line settings. The vertical gray lines indicate times when the topology changed.



(a) RMSEs of state estimates over time steps in 30% unobservable branches.

(b) RMSEs of state estimates over node numbers in 30% unobservable branches.

Figure 97: State estimation result under 30% unobservable branches.



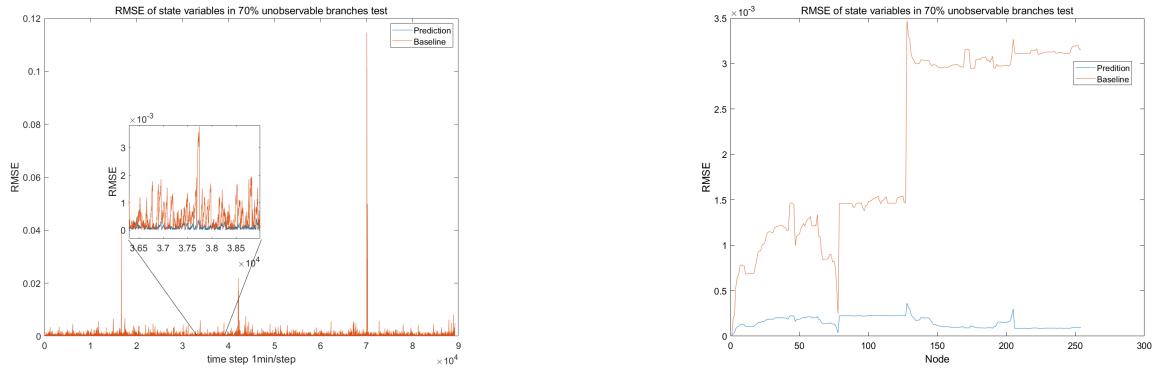
(a) RMSEs of state estimates over time steps in 50% unobservable branches.

(b) RMSEs of state estimates over node numbers in 50% unobservable branches.

Figure 98: State estimation result under 50% unobservable branches.

From these figures, it is evident that the predictions remain close to the ground truth in most cases. Nevertheless, the model performs well under both conditions, validating that our proposed framework can effectively handle scenarios where part of the line data is unobservable and where one-way line flow information is available. The robustness of the model under such circumstances reinforces its practical applicability in real-world power grid management scenarios, where topology changes and incomplete measurements are common challenges.

Validation of Closed Loop Operation of Measurement Predictor and State Estimator. We use the WLAV state estimator for the evaluation of the proposed closed-loop operation framework. The data used is clean data. We down-sample the node active and reactive power injection data to 60-minute intervals as the SA measurement input of the model from AMI, and the 5-minute interval data predicted by the deep learning method is only used for comparison. The line active and reactive power data and node voltage data remain at 5-minute intervals as they are assumed to be FS measurements from SCADA. The standard deviation of Gaussian noise of voltage magnitude, active and reactive power flow are respective 4×10^{-4} , 8×10^{-5} and 8×10^{-5} .



(a) RMSEs of state estimates over time steps in 70% unobservable branches.

(b) RMSEs of state estimates over node numbers in 70% unobservable branches.

Figure 99: State estimation result under 70% unobservable branches.

Table 34: FS measurement prediction results with proportions of unobservable lines in terms of MAE.

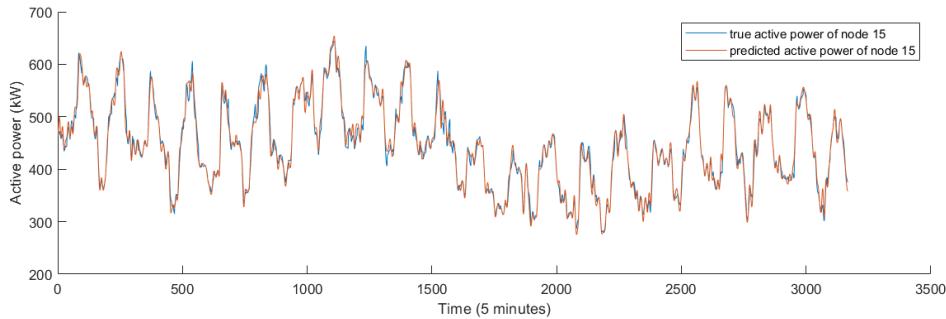
Proportions of Lines		Naive		With Clean Data	
With Measurements	Unobservable	Active Power	Reactive Power	Active Power	Reactive Power
38.8%	30%	9.1298	2.6441	2.7785	1.1155

The tests for state estimation carry on according to the 30% unobservable branches in measurement configuration. The RMSEs of state variables are used to evaluate the performance of SE. From the results of Table 35, we can see that the Root Mean Square Errors (RMSEs) of voltage estimations are all smaller than the RMSEs of measured voltage in the different tests of unobservable lines. RMSEs of voltage estimation 7.89×10^{-5} are all smaller than alpha% which is defined as the standard deviation of the Gaussian noise of voltage measurement with the value of 4×10^{-4} , satisfying **Milestone 3.12.5**. Figure 103 is the example of RMSE results of voltage estimation and voltage measurements with 30% unobservable lines in terms of time and node number, which show most of the voltage estimation is closer to the true voltage value compared with the voltage measurements.

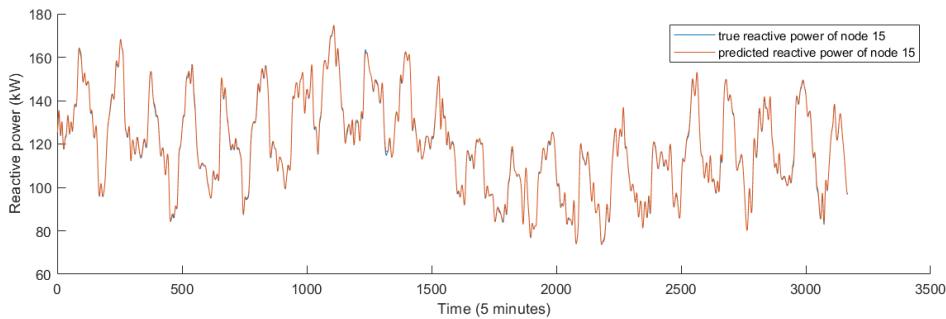
Compared with the baseline, our proposed model shows the great improvement. The Figs. ?? shows the RMSE of state variables during the first several days in the measurement configuration of the 30% unobservable branches. In these figures, the RMSE of voltage angle are in the first 90 nodes of x axis and the RMSE of voltage magnitude are in the last 90 nodes of x axis. Table 36 shows the overall performance and improvement of the prediction and baseline.

Subtask 13.2: Workshop Organization.

A full-day workshop entitled "**Unleashing the Power of Data and AI In Distribution Grid Operation and DER Management**" was held at Project Partner, New York University on Monday, December 2nd, 2024. It was synchronously broadcast over Zoom for remote attendees. It attracted over 30 in-person attendees and over 25 remote attendees. Prof. Ali Abur delivered an opening remark for the workshop. 12 speakers from academia, national labs, and industry made technical



(a) Active power prediction for Node 15



(b) Reactive power prediction for Node 15

Figure 100: Prediction results for Node 15 with 30% unobservable lines.

Table 35: RMSE of Voltage Estimations

Proportions of Lines	RMSE of Estimated Voltage (p.u.)	RMSE of Measured Voltage (p.u.)
38.8%	30%	7.89x10⁻⁵

presentations and discussed with the audience, listed as follows.

Yuzhang Lin, New York University, “*Physics-Informed Graph Learning for Robust Distribution System State Estimation with Heterogenous Measurements*”

Panagiotis Moutis, The City College of New York, “*The Many Challenges of Optimizing Operation & Planning of Our Failing Distribution Grids*”

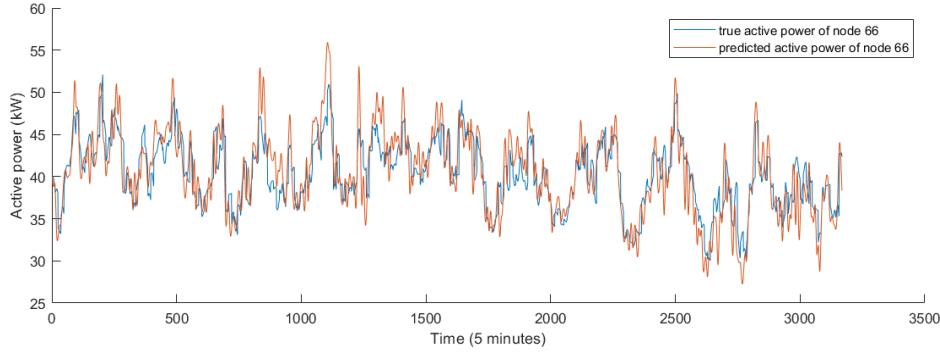
Le Xie, Harvard University, “*Exploring the Capabilities and Limitations of Large Language Models in the Electric Energy Sector*”

Fei Ding, National Renewable Energy Laboratory, “*Data Orchestration and Solar-Assisted Restoration for Resilience Enhancement*”

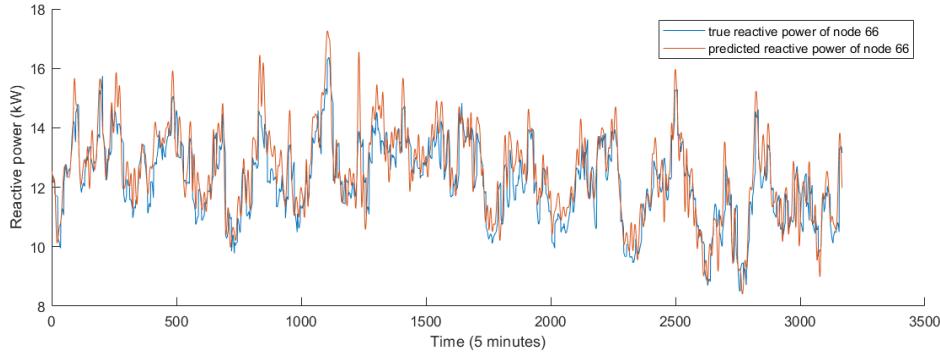
Simon Odie, Consolidated Edison, “*New Grid Topology via Advanced Inverters in an AI/ML Application Era - A Utility R&D Perspective on Grid Modernization, Journeying from Research to Scalability*”

Hanchao Liu, GE Vernova, “*Community Resilience through Rapid Restoration Leveraging DERs and Low-Cost Sensors*”

Bolun Xu, Columbia University, “*Efficient Management of Grid-scale Energy Storage Using Physics-informed Machine Learning*”



(a) Active power prediction for Node 66



(b) Reactive power prediction for Node 66

Figure 101: Prediction results for Node 66 with 30% unobservable lines.

Table 36: SE result of RMSE of state variables in different measurement configuration

Test	RMSE of Baseline	RMSE of Prediction	improvement
30%	9.69×10^{-5}	7.89×10^{-5}	18.49%

Yue Zhao, Stony Brook University, “*Identifying High-Significance Latent Physical Anomalies in Solar Energy Systems*”

Carson Zerpa, Itron, “*Distributed Intelligence Platform for AMI Edge Computing and Data Analytics*”

Anna Scaglione, Cornell University, “*Voltage Security in Three-Phase Unbalanced Distribution Systems with High Inverter Penetration: Challenges and Mitigation Strategies*”

Ugur Yilmaz, Northeastern University, “*Highly Efficient Robust State Estimation for Very Large Power Distribution Networks*”

Tuna Yildiz, Northeastern University, “*Event Detection and Location with Sparse Sensor Measurements in Power Distribution Systems*”

There was an open discussion among all the attendees at the end of the workshop. Key takeaways are summarized as follows.

1. In the field of distribution system operation and DER management, there are many advanced technologies being deployed and applications being implemented. However, there is a lack

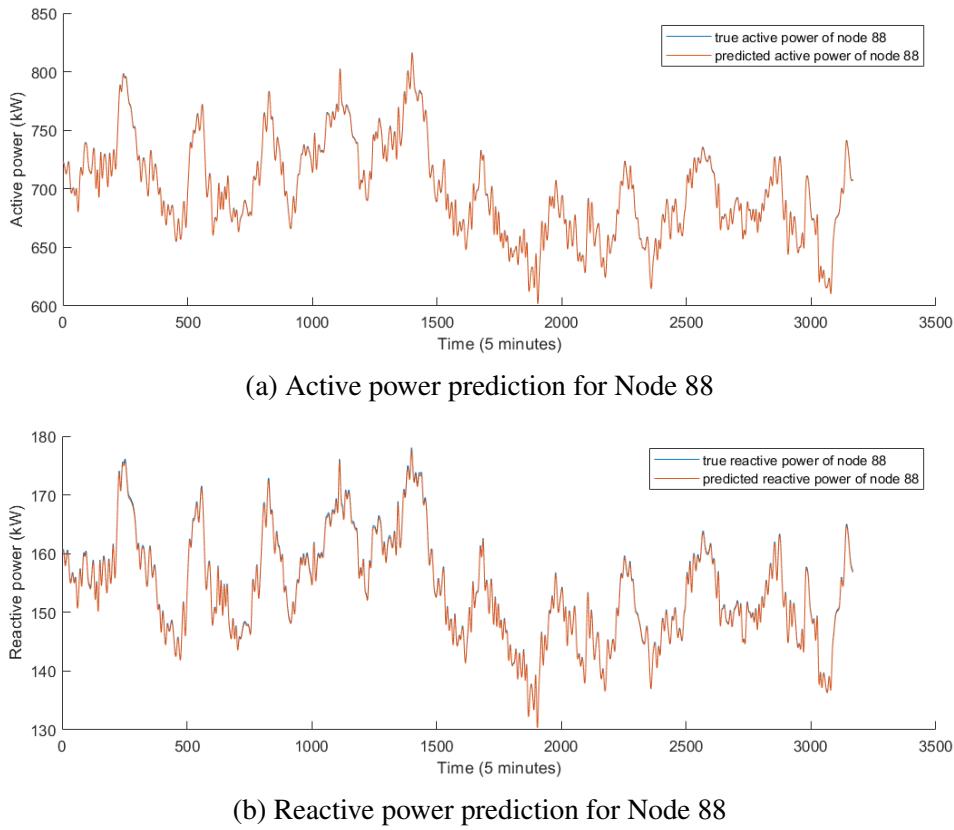
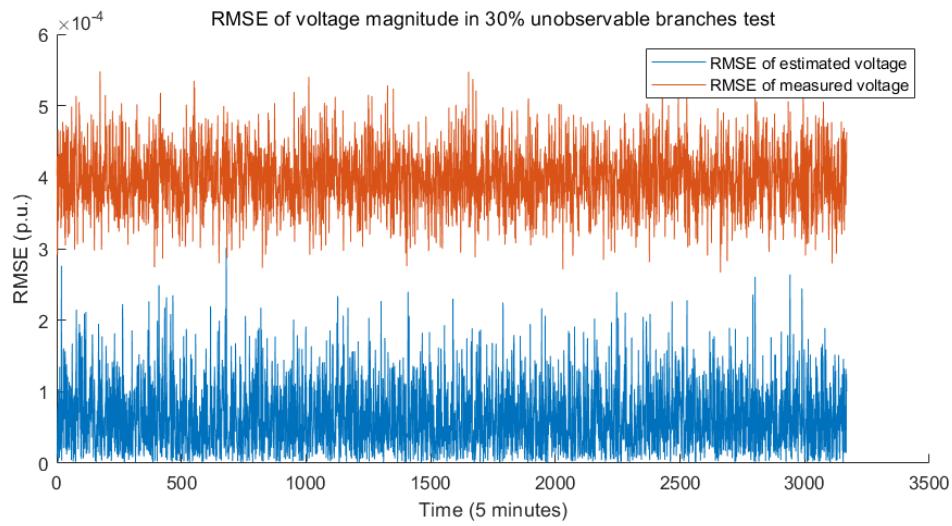


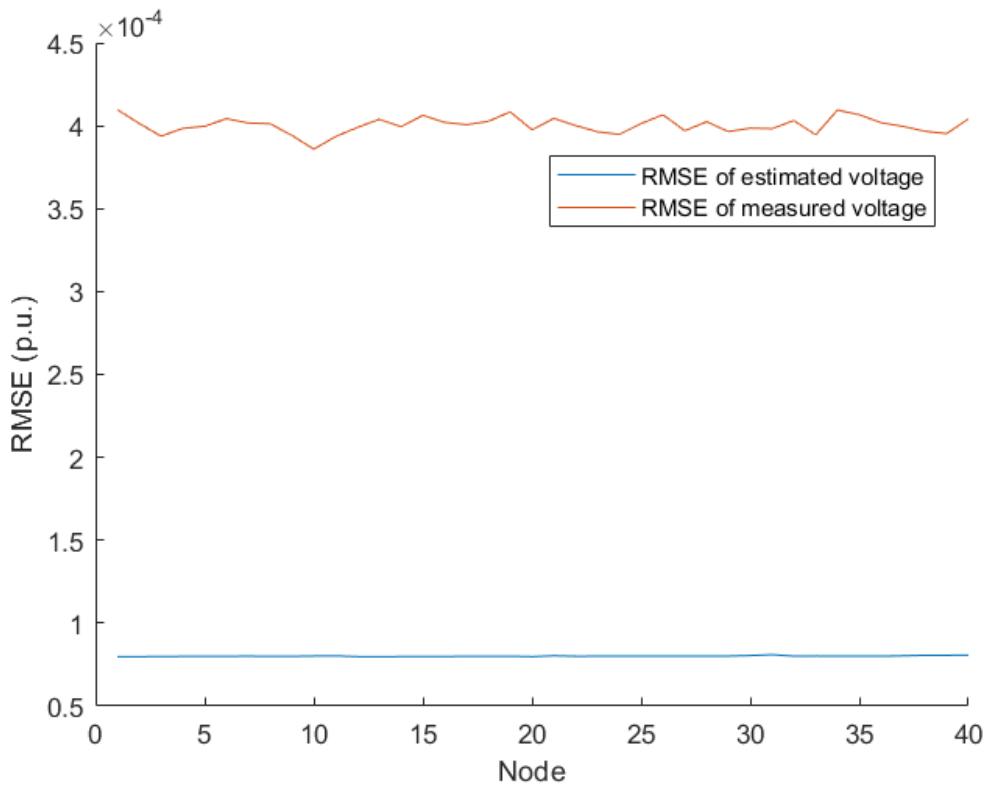
Figure 102: Prediction results for Node 88 with 30% unobservable lines.

of convergence among different technologies and applications. Different utilities are developing different practices and cost-benefit analysis models, and the success of one case cannot be easily transplanted elsewhere.

2. One of the reasons of the divergence is the lack of benchmarking effort. There is a lack of consensus on definitions and procedures. There is also a lack of standard procedures, criteria, datasets, and metrics to evaluate and compare the performances, costs, and benefits of different technologies and applications. DOE-hosted competitions are a very good start, but much more needs to be done.
3. As smart grid spans across multiple technical domains, there also needs to be convergence between different technical communities, including but not limited to power systems, power electronics, communication, cybersecurity, data science and artificial intelligence. Two examples: 1) power-electronics-based oscillations often occur, because utility teams in power systems do not have enough expertise in power electronics, the expertise of vendors; 2) within utilities, OT teams have power system expertise and IT teams have communications and cybersecurity expertise, but they do not understand each other's systems leading to system design vulnerabilities or economic inefficiency.
4. More technically informed people at policy making positions may help with the convergence.



(a) RMSEs of state estimates over time steps in 30% unobservable branches.



(b) RMSEs of state estimates over node numbers in 30% unobservable branches.

Figure 103: State estimation result under 30% unobservable branches.

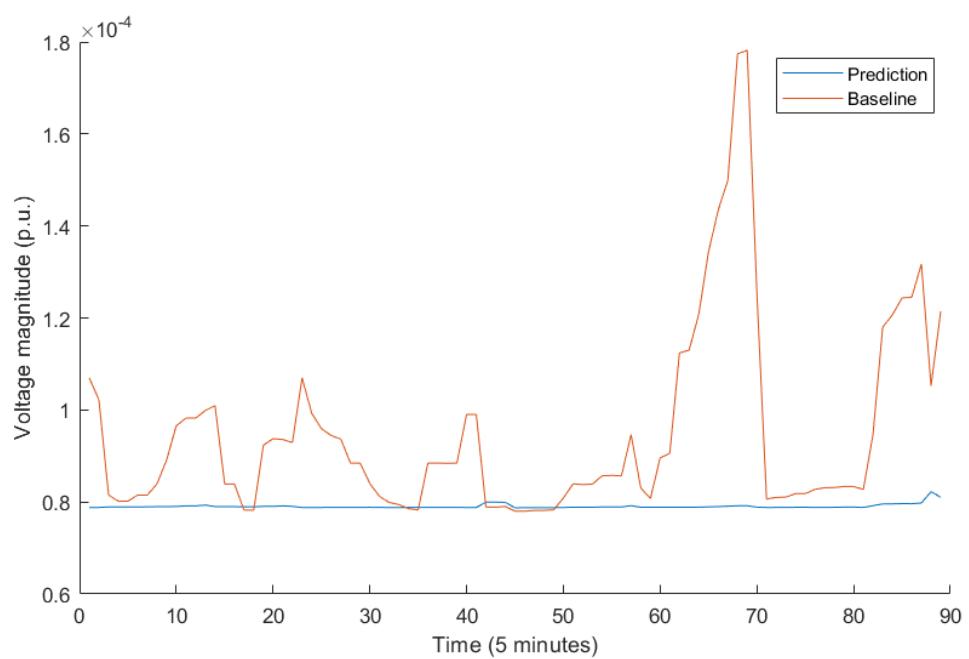


Figure 104: RMSE of Voltage Estimations in the test of 50% unobservable line.

	Budget Period 1				Budget Period 2				Budget Period 3			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Task 1.1												
Task 1.2												
Milestones	1.1.1											
Task 2.1												
Task 2.2												
Milestones		1.2.1			1.2.2							
Task 3.1												
Milestones					1.3.1 1.3.2							
Task 4.1												
Task 4.2												
Milestones					1.4.1							
Task 5.1												
Go/No-Go DP#1					DP#1							
Task 6.1												
Task 6.2												
Task 7.1												
Milestones								2.7.1 2.7.2				
Task 8.1												
Task 8.2												
Milestones						2.8.1		2.8.2				
Task 9.1												
Task 9.2												
Milestones								2.9.1				
Task 10.1												
Task 10.2												
Go/No-Go DP#2								DP#2				
Task 11.1												
Task 11.2												
Task 12.1												
Task 12.2												
Task 12.3												
Milestones										3.12.1	3.12.2 3.12.3	3.12.4 3.12.5 3.14.1
Task 13.1												
Task 13.2												
Task 14.1												
Task 14.2												
EOP Goals												EOP

IV SIGNIFICANT ACCOMPLISHMENTS AND CONCLUSION

Highlight any significant accomplishments and explain why they are significant. Both positive and negative outcomes should be included. Describe any challenges encountered under this award and the lessons learned. If applicable, describe why pivots or reduction in scope occurred.

The following highlights include significant accomplishments of the project related to **the state estimation** :

1. A three-phase state estimation program is developed and implemented. The program is using the weighted least absolute value (WLAV) estimation algorithm which is capable of automatically rejecting gross errors in the measurements. It can also differentiate between accurate and less accurate measurements via the assigned measurement weights. The program can handle balanced as well as unbalanced operating conditions.
2. The performance of the estimator is further improved. A specific angular reference issue in the context of observability analysis of three-phase state estimation is highlighted through straightforward examples [32]. A novel alternative solution developed to overcome the issue is proposed afterwards, and its effectiveness is validated using test scenarios [33].
3. The three-phase multi-copy multi-area state estimation program is developed, implemented, and tested. It is improved by incorporating an efficient network partitioning algorithm both for radial as well as meshed systems. The program is used to validate the massively parallel solution scheme on a 12,500-bus network.
4. Massively Parallel Distributed (MPD) WLAV Estimator is tested on physical multi-core processors using NEU's high performance computing center. The simulation results for radial/meshed networks with 12589 buses are presented in the report.
5. In addition to state estimation mechanism, a novel parameter tracking algorithm is developed. .

The following highlights are related to the significant accomplishments for **event detection purposes**:

1. A Line/Load/Gen Outage algorithm has been developed using the LASSO-based approach, achieving a satisfactory success rate.
2. A Fault Detection and Identification algorithm has been developed using the LASSO-based approach, also achieving a satisfactory success rate.
3. The co-linearity issue in the LASSO-based event detection algorithm has been resolved by implementing the QR decomposition formula within the proposed algorithm.
4. Ordinary Least Squares (OLS) estimation has been incorporated for virtual current injections at the terminal buses of line outages and faults, improving the success rate for phase outage and fault type detection.

5. A PMU placement algorithm has been developed to identify optimal PMU locations, effectively utilizing the limited number of available measurements to enhance the performance of the event detection algorithm.
6. Artificial Neural Network (ANN) models, including Convolutional Neural Networks (CNNs), have been developed as a complement to the analytical LASSO-based approach for event detection purposes.

1. A Line/Load/Gen Outage algorithm. Reliable operation of a power system depends on accurate monitoring and control of the operating state of the system. Monitoring is commonly accomplished via a state estimator which not only provides the best estimate of the system state but also detects and removes bad measurements. In doing all this, the state estimator implicitly assumes that the network model is perfectly known. This assumption may not always hold true. A line may be disconnected but this information may somehow not be reported to the control center. In such cases, results of the state estimator will be biased and these results also impact other application functions such as the contingency analysis, optimal power flow as well as market functions. Thus, it is crucial to detect changes in the network topology in a timely manner and to prevent such errors to create risky operating conditions leading to blackouts.

Thus, during this project, a successful outage detection algorithm was developed to identify unreported topology changes, ensuring continuous and reliable system operation.

Additionally, it was observed that for lightly loaded systems, including those with multiple zero-injection buses and poorly conditioned system matrices, the success rate of the outage detection algorithm decreases. However, this issue can be mitigated by employing various approaches, such as implementing QR decomposition within the event detection algorithm and strategically placing PMUs in optimal locations to enhance event detection capabilities.

2. A fault detection and identification algorithm. Besides outage detection, fault detection plays a crucial role in ensuring robust and reliable system operations. Undetected faults can result in significant financial losses and disrupt system operations. Therefore, timely and accurate fault detection is essential to prevent such adverse outcomes. Moreover, pinpointing the location of faults within power systems significantly reduces fault recovery time, thereby minimizing financial expenses and improving system resilience.

Therefore, in this project, an accurate algorithm for fault detection and pinpointing the exact location of faults was developed using only a limited number of available PMUs. This approach enables system operators to maximize the utility of their existing PMU measurements to ensure continuous system operation without the need for additional measurement installations.

3. Implementing QR decomposition formula within the proposed algorithm.

For event detection purposes, the Least Absolute Shrinkage and Selection Operator (LASSO) was utilized. However, due to the limited number of available PMU measurements, which renders the system unobservable, a co-linearity issue arises in the design matrix X . This co-linearity problem creates challenges for the LASSO solver, as it tends to select one of the collinear predictors while eliminating the others. This behavior may lead to the misidentification of events. To address this issue, the proposed algorithm incorporates the QR decomposition technique, effectively reducing the impact of co-linearity and enhancing the accuracy and reliability of the event detection algorithm.

4. Implementing OLS method within the proposed algorithm.

It was observed that event detection algorithms can be further enhanced by incorporating Ordinary Least Squares (OLS) estimation as a final step after identifying the faulted or disconnected line. In this process, OLS is applied to determine the values of virtual current injections at the terminal buses of the events, instead of using LASSO. This approach significantly improves the success rate in detecting the fault type, fault location, and outage type while utilizing the same number of PMUs. This enhancement is crucial, as the OLS method enables users to fine-tune the performance of the event detection algorithm without the need for additional measurement installations.

5. Optimal PMU placement algorithm for event detection purposes.

With the increasing installation of PMUs in power grids, their measurements were extensively utilized to detect and identify events in power systems. However, due to the limited number of PMUs, not all line terminals will be directly measured. Consequently, the placement of these limited number of PMUs will play a crucial role in accurate detection and identification of these events. Optimal PMU placement has been widely studied in the literature for different network applications. While these approaches are based on the assumption of abundant availability of PMUs, which are mainly utilized to attain system observability. On the contrary, in the context of event detection and localization, it is essential to shift the focus from simply ensuring network observability to strategically positioning PMUs to enable precise, system-wide event detection. Thus, during this project, a strategical PMU placement algorithm was designed with the goal of maximizing the performance of event detection methods without significantly increasing the required number of PMUs [22].

6. An ANN aided event detection algorithm.

For event detection purposes, another algorithm was developed utilizing Artificial Neural Networks (ANN) to improve the detection rate of events. Specifically, a Convolutional Neural Network (CNN) was selected and implemented. The results, presented in previous quarterly reports, were promising in terms of the event detection success rate. It was demonstrated that model-based approaches can be significantly enhanced by incorporating ANN techniques. This advancement also indicates that the number of PMUs required for the event detection algorithm can be further reduced, making the approach more efficient and cost-effective.

The following highlights are related to the significant accomplishments for **machine learning's role in state estimation purposes:**

1. Machine learning has been devised and proven effective for synchronizing power distribution measurement data with different reporting rates. With machine learning, measurements with slower reporting rates can be synchronized in the rate of state estimation execution, with significantly reduced errors compared with the original time skew errors.
2. A graph neural network architecture has been established to address topology changes in the measurement synchronization problem. It has been shown that graph neural networks can capture and generalize the mapping between fast and slow measurements, and perform reliable inference under new, unseen topologies. This makes the measurement synchronization idea applicable to distribution systems with frequent topology changes.

3. A closed-loop operation framework has been developed, in which the machine learning module for measurement synchronization has been shown to be synergistic and mutually-assistive with the robust state estimation module. Bad predictions by the machine learning module can be identified by the robust state estimation module, and the feedback information can help retrain the machine learning module for performance improvement.
4. Similar philosophy has been shown effective in other monitoring tasks of power distribution systems, for example, ultra-short-term forecasting of distributed solar PV generation with incomplete/multi-rate measurements.

CONCLUSION

The project was concerned about monitoring distribution systems which do not have sufficient measurements to make them fully observable. In order to address this challenge, the project relied on slow but abundant AMI meter measurements which are available every 15-30 minutes. Such resolution is too slow to enable real-time monitoring of systems. Hence, the project developed a machine-learning (ML) based approach which used these slow AMI measurements to predict higher resolution (every few minutes) Supervisory Control and DAta Acquisition (SCADA) type measurements. Using these predicted measurements the system is rendered fully observable and a robust state estimator is executed. Robust estimator provides feedback on the utilized measurements indicating whether or not they carry significant prediction errors and this information is used by the ML-based predictor to correct the training set and improve the prediction errors. In addition, the project developed a fault and line outage detection approach based on a limited set of Phasor Measurement Unit (PMU) measurements in order to detect unreported changes in the network model. This tool allowed further improvements in the predicted measurements and also helped the system operators by enabling situational awareness. All of the developed software tools are implemented and tested in a closed loop fashion to demonstrate the benefits of using robust state estimator as the enabler of training set correction. Finally the project successfully demonstrated the feasibility of the proposed ML-based approach to render distribution systems observable despite the lack of fully observable real-time measurement set.

V PATH FORWARD

Provide a detailed description of any plans for future research and development, as well as any opportunities for technology transfer or commercialization. If no immediate plans, please discuss the appropriate next R&D steps that might be taken to advance the technology. Please note any remaining technical risks and what barriers might prevent immediate commercialization.

Various modules which were developed as parts of the overall project could be effectively transformed into commercial tools. In particular, three-phase robust state estimator, machine learning based measurement predictor, fault and line outage detector/identifier are prime candidates for such effort. Technology transfer to market requires careful market analysis and risk assessment, strong engagement of industry partners, etc. In order to address these challenges as well as the technical challenges of transforming research grade software into commercial grade, we will need to carry out further work which will require support. Northeastern University supports such efforts through a program called Spark Fund. We intend to apply for this program with a detailed proposal of commercialization in late Spring 2025. This effort will not only involve technical work but also market research to identify the opportunities as well as barriers of market entry for the considered applications.

VI PRODUCTS

List all publications/papers, scientific/technical software/data, websites, inventions/patents, or other products developed or submitted under this award (see RPPR-2 “Products” Tab for a complete list and required formats). Also list all public releases of results including any significant media reports/articles, awards received, or networks/collaborations fostered/formed as a result of this award. Recipients are also required to upload Accepted Manuscripts of Journal Articles resulting in whole or in part from an EERE-funded project to OSTI and provide the OSTI IDs (See the FARC for detailed instructions and links).

- T. Yildiz and A. Abur, "Improved Line Outage Detection in Transmission Systems with Few PMUs," 2022 North Power Symposium (NAPS), Salt Lake City, UT, USA, 2022, pp. 1-6, doi: 10.1109/NAPS56150.2022.10012140.
- U. C. Yilmaz and A. Abur, "A General State Estimation Formulation for Three-Phase Unbalanced Power Systems," 2022 North American Power Symposium (NAPS), Salt Lake City, UT, USA, 2022, pp. 1-6, doi: 10.1109/NAPS56150.2022.10012212.
- T. Yildiz and A. Abur, "Computationally Robust Line Outage Detection and Identification in Three-Phase Networks," 2023 IEEE PES GTD International Conference and Exposition (GTD), Istanbul, Turkiye, 2023, pp. 109-113, doi: 10.1109/GTD49768.2023.00047.
- U. C. Yilmaz and A. Abur, "A Robust Parallel Distributed State Estimation for Large Scale Distribution Systems," in IEEE Transactions on Power Systems, vol. 39, no. 2, pp. 4437-4445, March 2024, doi: 10.1109/TPWRS.2023.3292552.
- T. Yildiz and A. Abur, "Sparse PMU Placement Algorithm for Enhanced Detection and Identification of Power Grid Events," in IEEE Transactions on Power Systems, doi: 10.1109/TPWRS.2024.3446237.
- U. C. Yilmaz and A. Abur, "Specifying angular reference for robust three-phase state estimation," International Journal of Electrical Power & Energy Systems, Volume 158, 2024, 109945, ISSN 0142-0615, <https://doi.org/10.1016/j.ijepes.2024.109945>.
- Han Yue, Wentao Zhang, Ugur Can Yilmaz, Tuna Yildiz, Heqing Huang, Hongfu Liu, Yuzhang Lin, Ali Abur, Graph-learning-assisted state estimation using sparse heterogeneous measurements, Electric Power Systems Research, Volume 235, 2024, 110644, ISSN 0378-7796, <https://doi.org/10.1016/j.epsr.2024.110644>.
- U. C. Yilmaz and A. Abur, "Ensuring Solution Uniqueness in Three-Phase Power System State Estimation," 2024 56th North American Power Symposium (NAPS), El Paso, TX, USA, 2024, pp. 1-6, doi: 10.1109/NAPS61145.2024.10741772.
- Tuna Yildiz, Ali Abur, Convolutional Neural Network-assisted fault detection and location using few PMUs, Electric Power Systems Research, Volume 235, 2024, 110705, ISSN 0378-7796, <https://doi.org/10.1016/j.epsr.2024.110705>.

VII PROJECT TEAM AND ROLES

List all participants along with their individual roles and/or intellectual contribution (e.g., DOE personnel, students, collaborating organizations).

The following participants played key roles during the project period:

Professor Yuzhang Lin and Professor Hongfu Liu:

Conceptualization, development of machine learning (ML) based measurement prediction, selection of the appropriate ML approaches for the development, supervision of graduate students who carried out the modeling and coding tasks, documentation of the results in form of reports and technical papers.

Han Yue and Wentao Zhang:

These graduate research associates carried out the modeling, data manipulation, coding, testing and validation work for the ML-based measurement predictor as well as training data generation, debugging, correcting and documenting the results.

Professor Ali Abur:

Overall coordination of the project, monitoring and tracking status of the budget and personnel, organizing quarterly conference calls, report writing, slide presentations, supervising graduate students who are working on the state estimation and event detection tasks, coordinating documentation of the results in reports and technical publications, working with university business office to coordinate subawards, tracking external invoices, communicating with DOE technical and business managers of the project to resolve issues, coordinating and finalizing reports, ensuring G/NG points and milestones are properly addressed.

Tuna Yildiz and Ugur Can Yilmaz;

These graduate research associates were responsible for the state estimation and event detection related tasks of the project. They carried out research, developed models, computer codes, tested the developed codes and validated results, documented results in reports and technical publications, helped with slide presentations for the quarterly reports as well as workshops.

Bo Chen and Ruoxi Zhu:

Colleagues from Commonwealth Edison Company (ComEd) helped with creation of testing data from Bronzville microgrid, developed the network model, created testing scenarios to provide system and measurement data for final testing of developed programs.

References

- [1] G. Cheng, Y. Lin, A. Abur, A. Gómez-Expósito, and W. Wu, “A survey of power system state estimation using multiple data sources: Pmus, scada, ami, and beyond,” *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.
- [2] Y. Yuan, K. Dehghanpour, F. Bu, and Z. Wang, “A multi-timescale data-driven approach to enhance distribution system observability,” *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3168–3177, 2019.
- [3] J. Zhao, C. Huang, L. Mili, Y. Zhang, and L. Min, “Robust medium-voltage distribution system state estimation using multi-source data,” in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2020, pp. 1–5.
- [4] J. Liu, F. Ponci, A. Monti, C. Muscas, P. A. Pegoraro, and S. Sulis, “Optimal meter placement for robust measurement systems in active distribution grids,” *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1096–1105, 2014.
- [5] K. R. Mestav, J. Luengo-Rozas, and L. Tong, “Bayesian state estimation for unobservable distribution systems via deep learning,” *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4910–4920, 2019.
- [6] T. Banerjee, Y. C. Chen, A. D. Dominguez-Garcia, and V. V. Veeravalli, “Power system line outage detection and identification a quickest change detection approach,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3450–3454.
- [7] S. Azizi, M. R. Jegarluei, A. M. Kettner, and A. S. Dobakhshari, “Wide-area line outage monitoring by sparse phasor measurements,” *IEEE Transactions on Power Systems*, pp. 1–11, 2022 (Early Access).
- [8] Y. Liao, Y. Weng, C.-W. Tan, and R. Rajagopal, “Quick line outage identification in urban distribution grids via smart meters,” *CSEE Journal of Power and Energy Systems*, vol. 8, no. 4, pp. 1074–1086, 2022.
- [9] J. E. Tate and T. J. Overbye, “Line outage detection using phasor angle measurements,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1644–1652, 2008.
- [10] H. Sun, F. Gao, K. Strunz, and B. Zhang, “Analog-digital power system state estimation based on information theory—part i: Theory,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1640–1646, 2013.
- [11] H. Sun, F. Gao, K. Strunz, B. Zhang, and Q. Li, “Analog-digital power system state estimation based on information theory—part ii: Implementation and application,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1647–1655, 2013.
- [12] L. Mili, M. Cheniae, N. Vichare, and P. Rousseeuw, “Robust state estimation based on projection statistics,” *Power Systems, IEEE Transactions on*, vol. 11, pp. 1118 – 1127, 05 1996.
- [13] U. C. Yilmaz and A. Abur, “A robust parallel distributed state estimation for large scale distribution systems,” *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 4437–4445, 2024.
- [14] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

- [15] K. Purchala, L. Meeus, D. Van Dommelen, and R. Belmans, “Usefulness of dc power flow for active power flow analysis,” in *IEEE Power Engineering Society General Meeting, 2005*, 2005, pp. 454–459 Vol. 1.
- [16] R. Emami and A. Abur, “External system line outage identification using phasor measurement units,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1035–1040, 2013.
- [17] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation, and control*. John Wiley & Sons, 2013.
- [18] N. Herawati, “Regularized multiple regression methods to deal with severe multicollinearity,” vol. 8, pp. 167–172, 05 2018.
- [19] R. J. Tibshirani, “The LASSO problem and uniqueness,” *Electronic Journal of statistics*, vol. 7, pp. 1456–1490, 2013.
- [20] T. Yildiz and A. Abur, “Improved line outage detection in transmission systems with few PMUs,” in *2022 North American Power Symposium (NAPS)*, 2022, pp. 1–5.
- [21] B. Donmez and A. Abur, “Sparse estimation based external system line outage detection,” in *2016 Power Systems Computation Conference (PSCC)*, 2016, pp. 1–6.
- [22] T. Yildiz and A. Abur, “Sparse pmu placement algorithm for enhanced detection and identification of power grid events,” *IEEE Transactions on Power Systems*, pp. 1–11, 2024.
- [23] A. Mouco and A. Abur, “Fault location using sparse l1 estimator and phasor measurement units,” 06 2019.
- [24] G. Feng and A. Abur, “Fault location using wide-area measurements and sparse estimation,” *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2938–2945, 2016.
- [25] A. R. Bergen and V. Vittal, *Power System Analysis*. Prentice-Hall, 2000.
- [26] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
- [27] T. Yildiz and A. Abur, “Computationally robust line outage detection and identification in three-phase networks,” in *2023 IEEE PES GTD International Conference and Exposition (GTD)*, 2023, pp. 109–113.
- [28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, “Graph attention networks,” *Stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [29] Z. Wang, J. Chen, and H. Chen, “Egat: Edge-featured graph attention network,” in *International Conference on Artificial Neural Networks, 2021*. Springer, 2021, pp. 253–264.
- [30] L. W. Beineke and J. S. Bagga, *Line graphs and line digraphs*. Springer, 2021.
- [31] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [32] U. Yilmaz and A. Abur, “A general state estimation formulation for three-phase unbalanced power systems,” *2022 North American Power Symposium (NAPS)*, pp. 1–6, 2022.

- [33] U. C. Yilmaz and A. Abur, "Specifying angular reference for robust three-phase state estimation," *International Journal of Electrical Power Energy Systems*, vol. 158, p. 109945, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0142061524001662>