EleGNN: Electrical-Model-Guided Graph Neural Networks for Power Distribution System State Estimation

Hui Lin and Yan (Lindsay) Sun
Electrical, Computer and Biomedical Engineering Department
University of Rhode Island
Kingston, RI, U.S.A
{huilin, yansun}@uri.edu

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000808

Abstract—**State estimation plays a critical role in power distribution systems. However, the conventional state estimation commonly used in transmission systems cannot be applied to the distribution systems because of insufficient measurements. Even though machine learning methods have begun to demonstrate their capabilities to overcome this challenge, they are not capable of explicitly incorporating distribution systems' topological information. This paper proposes EleGNN, an electrical-model-guided graph neural network (GNN), to perform power distribution system state estimation (DSSE). By explicitly considering physical topology, EleGNN enhances GNN with original node and edge feature propagation methods, allowing us to obtain accurate estimation results despite insufficient measurements and various topologies. Evaluations of six different power systems demonstrate significant improvement in state estimation accuracy than the method relying on general neural networks. Specifically, node-level mean square errors introduced by EleGNN are at least one order of magnitude smaller, even with up to 50% of missing measurements.**

Index Terms—**graph neural networks; power distribution systems; state estimation**

I. Introduction

State estimation plays a critical role in modern power systems, obtaining system states from noisy measurement data. Since its application in power transmission systems, the accurately estimated system states serve as a foundation for various control operations, from contingency analysis to economic dispatch [12]. In recent years, power distribution systems have adopted state estimation to accommodate various control operations near load units [1]. However, because power distribution systems share very different infrastructure characteristics from transmission systems, state estimation methods, which have become standardized in the transmission systems, involve original design options.

The first group of design options to perform power distribution system state estimation (DSSE) is model-based methods. If sufficient measurements and accurate topological information ensure system observability, model-based methods will result in closed-form solutions to system states. For example, the conventional weighted-least-square (WLS) algorithm and its variants can obtain a deterministic estimation of system states from the measurements that are corrupted with Gaussian noises [12]. However, power distribution systems are often unobservable due to insufficient meters. To compensate for system unobservability, many studies leverage statistical characteristics of historical sensor data to craft pseudo-measurements [10]. Unfortunately, the lack of sufficient historical data makes it challenging to ensure the accuracy of pseudo-measurements, leading to the performance downgrade of model-based DSSE.

Another group of methods, which are intrinsically not limited by system observability, is *model-free* methods. By exploiting machine and deep learning techniques, these methods estimate system states by establishing a probability relationship between system states and measurements. Learning from a large number of measurement data, the model-free methods estimate system states by performing probability inferences, regardless of whether the system is observable. Existing machine learning techniques, e.g., artificial neural networks, are initially designed to process image-structured data, e.g., the data encoded in a three-dimension tensor. Because they cannot directly process topological information from power grids, machine-learning models and parameters are trained again when they are performed for a different power system, significantly limiting the broad applications of the model-free methods [3], [4], [11], [19].

In this paper, we propose EleGNN, an electrical-model-guided graph neural network (GNN), to advance *model-free* methods for DSSE. By explicitly considering the physical topology of distribution systems, EleGNN enhances GNN with original node and edge feature propagation methods, allowing us to obtain accurate estimation results despite insufficient measurements and various topologies.

GNN is an emerging neural network specifically designed to handle relational data in graphical representations. Because of the success in node and graph classification in genomics data and social network data [8], [20], some studies have explored the potential of using GNNs to perform state estimation in power systems, demonstrating promising results when sufficient measurements are available [7], [9].

To further leverage GNNs for DSSE, we observe a critical challenge that the previous studies fail to address, e.g., *insufficient measurements in power distribution systems can result in unbalanced or missing **node** and **edge** features in a graph.* Handling the graph with missing node features is an open research problem in GNNs; existing solutions rely on probability correlations among existing node features to address the problem [6], [14], [15]. However, there is a lack of dedicated solutions regarding the missing edge features, which can prevent us from applying GNNs to DSSE. To address these challenges, we make the following contribution to EleGNN:

- By integrating electrical models of power distribution systems, e.g., the impedance of distribution lines and various power measurements injected into substations and distribution lines, EleGNN inductively estimates system states by training the model on a small set of systems.
- EleGNN introduces original node and edge feature propagation, leveraging the electrical centrality of each substation and electrical connectivity between substations, to address insufficient measurements.
- Evaluations of six power systems demonstrate significant improvement on state estimation accuracy compared to the methods relying on general neural networks, e.g., at least one order of magnitude smaller in node-level mean square errors, even with 50% of missing measurements.

The rest of the paper is organized as follows. Section II presents existing studies related to DSSE and compares EleGNN's contribution to selected related work. We include some necessary background on graph neural networks in Section III. Section IV presents the design of EleGNN; the evaluations of EleGNN's performance are detailed in Section V. We summarize the findings and present future work in Section VI.

II. Related Work

Model-based DSSE. Because power distribution systems often lack sufficient measurements, state estimation methods standardized in transmission systems cannot be directly applied. As a result, many studies created pseudo measurements before using analytical methods to solve this problem. The most common techniques to generate pseudo measurements are based on the statistical features of historical or existing measurements [10]. These methods are proprietary, and the accuracy can vary, affected by the number of existing measurements. Note that the errors of the generated pseudo measurements no longer follow the probability distributions of the errors caused by metering devices, breaking the assumptions of many model-based DSSE and downgrading their performances.

Model-free DSSE. To bypass the observability requirements in model-based methods, traditional machine learning and newly-emerged deep learning are used to estimate system states from the obtained measurements. For example, the studies in [3], [4], [11], [19] rely on various types of deep neural networks (DNN), e.g., multi-layer perceptron (MLP) or convolutional neural networks (CNN), to achieve this objective by training the model with a large number of operating conditions generated by Monte Carlo sampling. In [13], [18], the authors implicitly integrate power systems' physical properties with general DNN models. However, because DNNs do not explicitly incorporate power systems' topology information, the model structure and the parameters need to be trained again when they target a different power system.

Even though some studies have successfully applied GNNs to perform state estimation and bad data detection, their work inherits a major obstacle found in GNNs [2], [7], [9]. GNNs require that node and edge features in a graph, which encode input data, are evenly distributed. However, when we use a graph to encode measurements in a distribution system, insufficient measurements can easily result in unbalanced or missing node and edge features, making the data inapplicable to a GNN. To deal with missing node features, [6], [14], [15] estimate the missing node features by interpolating available features in their neighbors. These techniques assume that some latent random variables determine the probability distribution of the node features. Consequently, the missing node features can be deduced indirectly.

TABLE I: **Compare EleGNN's contribution to related work.**

	WLS [12]	**DNN** [3], [4], [11], [19]	**GNN** [7], [9]	**EleGNN**
Observability Limitation Handling	✗	✓	✓	✓
Electrical Model	✓	◪([13], [18])	✗	✓
Topology	✓	✗	✓	✓
Missing Feature Handling	◪([10])	N/A	✗	✓

EleGNN's Contribution. In Table I, we position EleGNN against the related work. We use "✓" and "✗" to directly specify whether an approach is or is not equipped with the corresponding capability. When ad-hoc extensions are made, we use "◪" to reflect the fact and include the corresponding references in brackets. Because EleGNN adopts the infrastructure of GNN, it is not limited by observability requirements and explicitly considers the topological knowledge of distribution systems. Consequently, the same GNN models can be trained by a few power systems and inductively applied to estimate other systems' states. Meanwhile, unlike GNNs, which cannot handle missing node and edge features, EleGNN introduces original feature propagation methods. Note that node and edge feature propagation is different from creating pseudo measurements. We leverage *electrical models* to add node and edge features to make input data compatible with GNNs (there is still no need to observe the observability requirement). Meanwhile, EleGNN's state estimation accuracy can strongly tolerate the accuracy of the generated missing features.

III. Background

A. State Estimation

Power grids started to use state estimation in their transmission systems in the 1970s. They use the following mathematical representation to correlate data collected from substations, i.e., $\mathbf{z} = h(\mathbf{x}) + \mathbf{e}$, where $\mathbf{z} = [z_1, z_2, \ldots, z_q]^T$ represents q measurements; $\mathbf{x} = [x_1, x_2, \ldots, x_p]^T$ represents p physical state; $\mathbf{e} = [e_1, e_2, \ldots, e_q]$ is the collection of q measurement errors; and h is a group of power flow equations. In a power system, sensor measurements ($\mathbf{z}$) can refer to data efficiently measured by sensors, e.g., active/reactive power injected at each substation and active/reactive power delivered by distribution lines. State variables ($\mathbf{x}$), indirectly estimated via the state estimation, refer to voltage magnitudes and angles at each substation or bus [12]. The state estimation can be regarded as a weighted-least-square (WLS) problem on non-linear power flow equations (denoted by $h(\cdot)$). We can solve this problem by minimizing a weighted measurement residual. Sufficient measurements that satisfy the observability requirements can ensure the deterministic solutions from the iterative WLS solvers.

To simplify the discussion, we consider balanced three-phase systems in this paper. However, EleGNN has no restrictions on unbalanced three-phase systems.

B. Graph Neural Networks (GNN)

GNN is a graph learning technique that leverages neural networks to process and analyze graph-structured data directly. A graph G can be represented as a set of vertices V and a set of edges E, i.e., $G = (V, E)$. Each node and edge can be characterized by a feature vector $\mathbf{f}$. A graph learning is a process, in which node features deduce a new embedding of each node (e.g., formatted as another vector $\mathbf{e}$). The similarity of node embedding in the Euclidean space reflects the similarity of nodes in their original graph. To facilitate graph learning, GNNs use deep neural networks to learn node embedding by aggregating features from neighbor nodes and incident edges. Specifically, network neighborhood defines a computation graph, which can be represented as:

$$\mathbf{e}_v^0 = \mathbf{f}_v \tag{1}$$

$$\mathbf{e}_v^{k+1} = \pi(\mathbf{e}^k_{ne[v]}, \mathbf{f}^k_{inc[v]}) \tag{2}$$

where the layer-0 embedding of node v, i.e., $\mathbf{e}_v^0$, is initialized as a node feature $\mathbf{f}_v$ and then layer-$(k+1)$ embedding of node v is iteratively obtained by applying a neural network π over the layer-k embedding of v's neighbor (specified as $ne[v]$) and the feature of edges incident to v (specified as $inc[v]$). In GNNs, k is a hyper-parameter specifying how "deep" we want to aggregate neighbor information to deduce a node's embedding. The last layer of node embedding is usually leveraged to make decisions such as node classification [8]. As shown in Equation 2, all node features and/or all edge features need to have the same dimensions. In other words, GNNs cannot directly handle a graph with missing node or edge features.

IV. EleGNN Design

A. Design Overview

We present the overall architecture of the electrical-model-guided GNN (EleGNN) for DSSE in Figure 1, which includes three components: (i) a general GNN processing graph-structured data that encode power distribution measurements and estimate system state; (ii) a Node Feature Propagation component designed based on electrical distances between k-nearest neighbors; and (iii) an Edge Feature Propagation component leveraging electrical-centrality of each substation in the target distribution system.

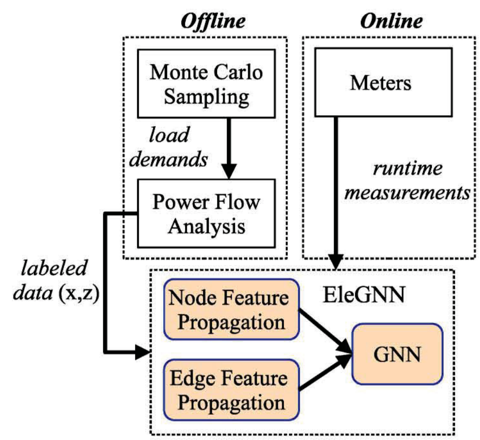

Fig. 1: **Design overview**: the components of EleGNN and the working procedure.

B. GNN Construction

Input Encoding. We encode real and reactive power injected at each bus as node features of the graph representing the corresponding distribution system. An example 5-bus distribution system is shown in Figure 2. Consequently, the feature of each node is a two-dimension vector. In addition, the feature of each edge is a four-dimension vector, including the active and reactive power delivered over a distribution line and the real and imaginary parts of the line impedance. To simplify the discussion, we only consider the power delivery at the sending end of the distribution line; adding additional features such as power delivery at the receiving end can be straightforward in EleGNN. Edge features characterizing distribution lines are often ignored by previous studies; they introduce an "inductive" capability, allowing the model trained by one system directly be applied to other systems.

GNNs use neural networks to aggregate node and edge features. In EleGNN, we adopt GATConv proposed in [16] to implement the structure of neural networks, i.e., π in Equation 2. GATConv is a powerful structure, which can aggregate node features and multi-dimensional edge features. Its computation logic can be found in [16].

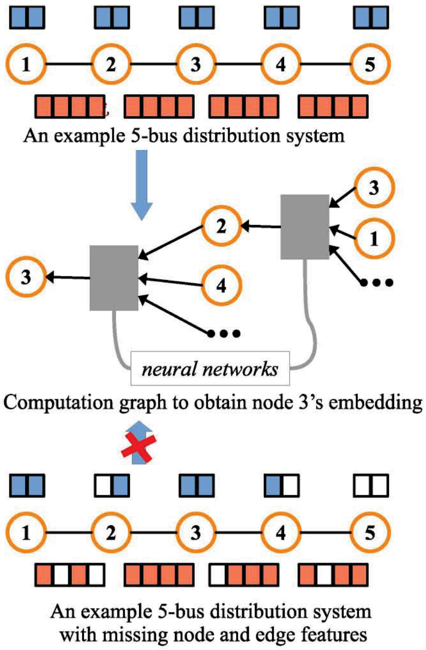

Fig. 2: **The computational graph of node 3's embedding** in an example 5-bus distribution system.

Dataset Generation. To generate sufficient amount of data, we follow a common procedure used in existing model-free state estimation (also shown in Figure 1) [11] by (i) randomly changing measurements related to load demands of distribution systems based on Monte Carlo sampling and (ii) using power flow analysis to calculate the corresponding system states $\mathbf{x}$, i.e., voltage magnitude and phasor angles at each substation, and other measurements, e.g., the power delivered by distribution line. The combination of measurements $\mathbf{z}$ and the corresponding system states $\mathbf{x}$ serve as the labeled data to train EleGNN in a supervised learning manner.

Output & Loss Function. We use node embedding as the output. Specifically, the embedding of each node is a two-dimension vector representing the estimated voltage magnitude and phasor angle at each bus. In other words, the complete node embedding directly presents the estimation of system states $\hat{\mathbf{x}}$ discussed in Section III-A. We define the loss function as the node-level mean squared errors (MSE) between the state estimated by EleGNN and the labeled state (obtained through the power flow analysis), whose definition is presented in Equation 6.

C. Node Feature Propagation

GNNs can only operate on graph-structured data, in which node features or edge features share the same dimension, e.g., two-dimension node features and four-dimension edge features considered in this work. In case a distribution system does not deploy enough meters, we can have a graph with unbalanced and missing node and edge features (an example is shown in Figure 2). In that case, input data go through the Node and Edge Feature Propagation components to complete features before they are used to train the GNN.

Unlike crafting pseudo measurements used in many model-based DSSEs, handling missing node features is based on a very different concept. For example, the studies in [5], [14] assume that node features can propagate within the graph in a similar way that energy or heat diffuses among close entities. Under this assumption, they formulate node feature propagation as a differential equation, whose discretization leads to an iterative algorithm (shown in Algorithm 1). After initializing missing node features as random values, the algorithm uses a diffusion matrix D to "propagate" known features towards missing features.

As shown in [14], this algorithm shows promising results in general graph processing problems, e.g., node classification, when the normalized adjacency matrix of a graph is used as the diffusion matrix D. However, this selection of diffusion matrix does not perform well in distribution systems, which often have a flattened topology, as shown in Figure 2. For example, if the features of nodes 2 and 3 are missing, applying this algorithm can result in the same features due to the similar structure characteristics these two nodes share in the graph.

To apply Algorithm 1 in distribution systems, we define a different diffusion matrix D, i.e., a normalized adjacency matrix built based on the electrical distances of m-nearest neighborhood in the graph. Here, the value of m can be a design parameter.

$$D(i,j) = \begin{cases} 0 & i = j \\ d_Z(i,j) & \exists \text{ a } m\text{-hop path between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $d_Z(i,j)$ is an electrical distance between nodes i and j and it is defined as:

$$d_Z(i,j) = || \sum_{(s,t)\in path(i\rightarrow j)} Z(s,t) || \tag{4}$$

where, $Z(s,t)$ refers to the impedance of the edge (s,t). Using more features from m-nearest neighbors instead of the direct neighbors can compensate for the simple graph topology found in power distributions. Meanwhile, features from different neighbors are weighted based on electrical distances, which specifies the electrical correlations between two nodes instead of their geometrical distances [17].

Algorithm 1 Node Feature Propagation in EleGAN

Input: feature vector $\mathbf{f}$, diffusion matrix D
procedure
  $\mathbf{y} \leftarrow \mathbf{f}$
  while $\mathbf{x}$ has not converged **do**
    $\mathbf{x} \leftarrow D\mathbf{f}$ ▷ node propagation
    $\mathbf{x}_{known} \leftarrow \mathbf{y}_{known}$ ▷ reset known features

D. Edge Feature Propagation

At this stage, very little work is dedicated to handling edge feature propagation. Consequently, we apply the node feature propagation shown in Algorithm 1 to an edge-to-vertex dual graph G' of G. Specifically, a node in G' corresponds to an edge in G; the connectivity of two nodes in G' is determined if the corresponding edges in G share one same vertex. Constructing a diffusion matrix D for G' requires us to define edge weights in G', which actually encodes the original distribution system's node properties. We achieve this objective by using electrical centrality for each bus of the original distribution system [17]. Similar to node centrality in a graph, electrical centrality quantifies a bus' criticality related to the states of a power distribution system. It is defined as follows, which is used as the edge weight in G'.

$$C(s) = \frac{n-1}{\sum_{t \in V \setminus s} d_z(s,t)} \tag{5}$$

where n is the number of nodes in G and $d_z(s,t)$ is defined in Equation 4.

V. Evaluation Results & Discussions

A. Implementation

We consider six different power systems in our evaluation, which are included in the MATPOWER toolbox [21]. The basic properties of each system is summarized in Table II. The first four systems are distribution systems; the latter two are large-scale transmission systems used to evaluate the scalability of EleGNN.

TABLE II: **Power systems used in the evaluation.**

Case Name	# of buses	# of branches	Case Name	# of buses	# of branches
12da	12	11	*136ma*	136	156
38si	38	37	*ACTIVSg200*	200	245
74ds	74	73	*ACTIVSg500*	500	597

To generate datasets, we randomly changed the load demands of each bus by $\pm 10\%$ and used power flow analysis in MATPOWER to obtain the labeled dataset. For each case, we have generated 5,000 labeled data. We made ten runs of training and testing; 80% of data were randomly selected as training data and the remaining 20% were used as testing data in each run. The loss function is defined as the average per-node mean-square errors:

$$MSE = \frac{1}{MN} \sum_k ||\hat{\mathbf{x}}[k] - \mathbf{x}[k]|| \tag{6}$$

where M and k is the number and the index of labeled data and N is the number of nodes. We used the mean and the 95% confidence interval of the MSE in these ten runs as the performance metric for each system.

We implemented all components of EleGNN by Pytorch and Pytorch geometric toolbox. Specifically, the GNN used by EleGNN includes two GATConv layers, each followed by a dropout and an activation layer. In the Node and Edge Feature Propagation components, 5-nearest neighbours were used. All experiments were performed in a 64-bit Ubuntu 20.04, deployed in a workstation with an AMD Threadripper 3970X 32-Core processor, 64 GB RAM, and a 24 GB Titan GPU.

B. Estimation Accuracy

In Figure 3, we present the estimation accuracy of EleGNN when sufficient measurements are available. To demonstrate the benefit of using GNNs, we compare EleGNN to two model-free state estimation methods. The first one is based on a multi-layer perceptron (MLP), consisting of two fully-connected layers. Each layer includes eight neurons, followed by a "leaky relu" activation layer. The second one is based on a convolutional neural network, including two convolutional layers, two pooling layers, and one fully-connected layer. Each convolutional layer had a kernel size of three, a stride of one, and a single zero padding. In Figure 3, the x-axis specifies the six systems considered in this work, while the y-axis specifies the mean and the 95% confidence interval of MSE on a logarithmic scale. Even though the MLP has a similar number of parameters to GATCONV used in EleGNN, EleGNN enjoys a significant improvement in estimation accuracy of system states; EleGNN's MSE is at least one order of magnitude smaller than the results obtained by the other two methods.

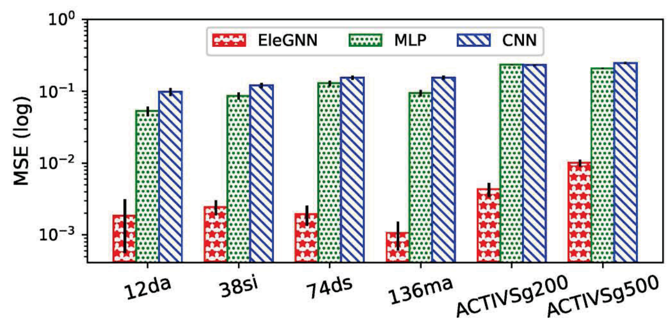

Fig. 3: **Compare EleGNN, MLP, and CNN.** The MSE obtained by EleGNN is at least one order of magnitude smaller than MLP and CNN.

Compared to MLPs and CNNs, EleGNN uses more complicated infrastructures. Despite this, EleGNN also produces stable results, especially in large-scale power systems. As we change the training and testing dataset, EleGNN's results introduce small confidence intervals. The exception happens in the *12da* system. This is probably because a GNN model trains neural networks by node and edge features. A system with a smaller size results in less effective training data, which introduces more variations.

C. Cross-system Evaluation

Compared to general neural networks, EleGNN inherits a unique feature from GNNs. By using neural networks to aggregate node and edge features, EleGNN can be trained by one system and be directly applied to another system. This inductive property can significantly reduce the training overhead of EleGNN-based state estimation compared to the state estimation based on general neural networks.

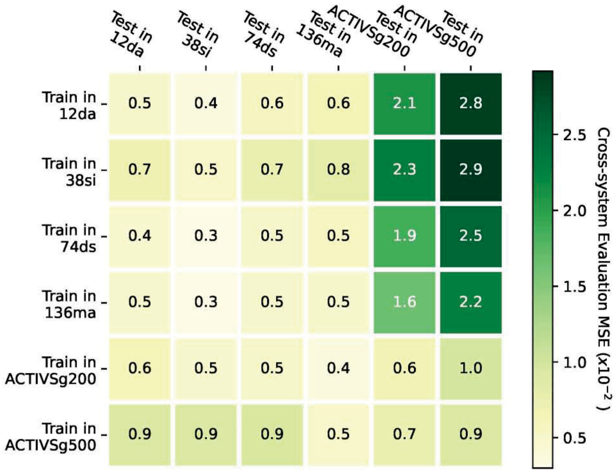

Fig. 4: **Cross-system evaluation.** State estimation trained by one system can maintain its performance in other systems.

In Figure 4, we use a heatmap to demonstrate the results of cross-system evaluation. We use the model trained by the data from one system, specified by the *y*-axis, to evaluate the testing dataset generated by different systems, specified by the *x*-axis. The corresponding MSEs are presented in the figure. An interesting observation is that if a model is trained by a distribution system, it can perform well for other distribution systems. But its accuracy downgrades slightly in the two large transmission systems. However, using the transmission systems to train the model can perform well in both distribution and transmission systems. The major reason is that transmission systems usually have more complicated typologies and have more node and edge features to train EleGNN. Consequently, cross-system evaluation can benefit from training on systems with more complicated typologies.

D. Impact of Insufficient Measurements

To evaluate the impact of insufficient measurements, we randomly remove 50% of node and edge features in the training/testing dataset and apply the proposed node and edge feature propagation algorithms.

In Figure 5, we present the impact of node/edge feature propagation on state estimation, when 50% node or 50% edge features or both features are missing (shown in different bar patterns). Because of the feature propagation algorithms, EleGNN maintains its performance. We only observe less than 1% increase in MSE in the large ACTIVSg500 power system. Also, this downgrade is mainly caused by the missing node features. Consequently, the node features or the bus

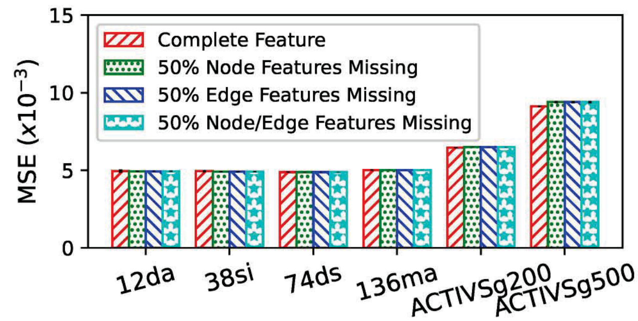

Fig. 5: **The impact of insufficient measurements.** Even with up to 50% of missing measurements, EleGNN still obtains a decent estimation accuracy.

measurements play more critical roles in EleGNN than the edge features.

E. Discussion

As discussed in [11], model-free state estimation, like EleGNN, results in a different type of error from the model-based state estimation. The model-based state estimation, relying on power flow equations and iterative algorithms like the WLS solver, attempts to minimize *modeling* errors. However, the model-free methods, like the ones based on EleGNN, MLP, and CNN, directly build probability correlations between measurements and system states, attempting to minimize *estimation* errors. Consequently, they are used as an *alternative* approach to model-based methods, not a *replacement*. In addition to the benefits shown in Table I, EleGNN is also more suitable for real time state estimation, because it takes less and stable latency than the model-based methods. In Figure 6, we can see that the latency of WLS state estimation increases with the size of the system. The latency caused by EleGNN is stable and is at least one order of magnitude smaller.

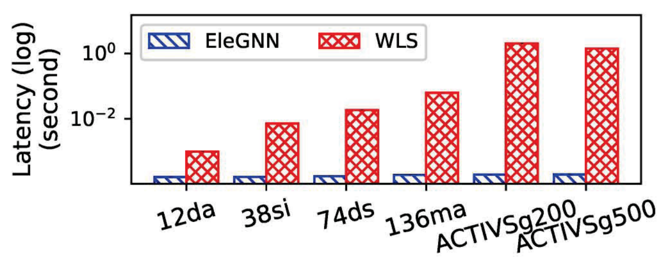

Fig. 6: **Latency of state estimation.** The latency of WLS state estimation increases dramatically with the size of power grids, at least one order of magnitude larger than EleGNN.

VI. Conclusion

In this paper, we introduce EleGNN, an electrical-model-guided GNN for DSSE. EleGNN integrates the electrical models of power distribution systems, e.g., the impedance of distribution lines and various power measurements injected into substations and distribution lines, and inductively estimates system states by training on a small set of distribution systems. When a distribution system obtains insufficient measurements, EleGNN introduces original node and edge feature propagation by leveraging electrical centrality of each substation and electrical connectivity between substations. Evaluations of six different power systems demonstrate significant improvement in state estimation accuracy, even with up to 50% of missing measurements.

Acknowledgment

This material is based upon work partially supported by the National Science Foundation under Award No. CNS-2144513. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] F. Ahmad, A. Rasool, E. Ozsoy, R. Sekar, A. Sabanovic, and M. Elitaş, "Distribution system state estimation-a step towards smart grid," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 2659–2671, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117310134

[2] O. Anderson and N. Yu, "Distribution system bad data detection using graph signal processing," in *2021 IEEE Power Energy Society General Meeting (PESGM)*, 2021, pp. 01–05.

[3] P. N. P. Barbeiro, J. Krstulovic, H. Teixeira, J. Pereira, F. J. Soares, and J. P. Iria, "State estimation in distribution smart grids using autoencoders," in *2014 IEEE 8th International Power Engineering and Optimization Conference (PEOCO2014)*, 2014, pp. 358–363.

[4] A. Bernieri, G. Betta, C. Liguori, and A. Losi, "Neural networks and pseudo-measurements for real-time monitoring of distribution systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 645–650, 1996.

[5] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, "Grand: Graph neural diffusion," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 1407–1418. [Online]. Available: https://proceedings.mlr.press/v139/chamberlain21a.html

[6] X. Chen, S. Chen, J. Yao, H. Zheng, Y. Zhang, and I. W. Tsang, "Learning on attribute-missing graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 740–757, 2022.

[7] B. Donon, B. Donnot, I. Guyon, and A. Marot, "Graph neural solver for power systems," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1609.02907

[9] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," 2022. [Online]. Available: https://arxiv.org/abs/2201.04056

[10] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac, "Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1888–1896, 2012.

[11] 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.

[12] A. Monticelli, *State estimation in electric power systems: a generalized approach*. Springer Science & Business Media, 2012.

[13] L. Pagnier and M. Chertkov, "Physics-informed graphical neural network for parameter & state estimations in power systems," 2021. [Online]. Available: https://arxiv.org/abs/2102.06349

[14] E. Rossi, H. Kenlay, M. I. Gorinova, B. P. Chamberlain, X. Dong, and M. Bronstein, "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features," 2021. [Online]. Available: https://arxiv.org/abs/2111.12128

[15] H. Taguchi, X. Liu, and T. Murata, "Graph convolutional networks for graphs containing missing features," *Future Generation Computer Systems*, vol. 117, pp. 155–168, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20330405

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017. [Online]. Available: https://arxiv.org/abs/1710.10903

[17] Z. Wang, A. Scaglione, and R. J. Thomas, "Electrical centrality measures for electric power grid vulnerability analysis," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5792–5797.

[18] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4347–4356, 2020.

[19] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 15, pp. 4069–4077, 2019.

[20] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000012

[21] R. Zimmerman, C. E. Murillo-Sanchez, and R. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.