

# On Zero-shot Learning in Neural State Estimation of Power Distribution Systems

✉ Aleksandr Berezin\*, ✉ Stephan Balduin\*, ✉ Eric MSP Veith\*,  
 ✉ Thomas Oberließen† and ✉ Sebastian Peter†

\*OFFIS – Institute for Information Technology, Oldenburg, Germany  
 e-mail: {name.surname}@offis.de

†Institute of Energy Systems, Energy Efficiency and Energy Economics  
 Technical University of Dortmund, Germany

**Abstract**—This paper addresses the challenge of neural state estimation in power distribution systems. We identified a research gap in the current state of the art, which lies in the inability of models to adapt to changes in the power grid, such as loss of sensors and branch switching, in a zero-shot fashion. Based on the literature, we identified graph neural networks as the most promising class of models for this use case. Our experiments confirm their robustness to some grid changes and also show that a deeper network does not always perform better. We propose data augmentations to improve performance and conduct a comprehensive grid search of different model configurations for common zero-shot learning scenarios.

**Keywords**—neural state estimation; zero-shot learning; transfer learning; graph neural networks.

## I. INTRODUCTION

Power System State Estimation (PSSE) is the task of inferring the “state” of an electrical power grid from real-time data collected by various sensors distributed across the system. The “state” in this context generally refers to the voltage magnitudes and phase angles at each bus in the grid.

For many years, PSSE was mainly performed for the transmission grids using simplifying assumptions such as near-DC power flow and computational methods with poor scalability [1]. This is enabled by balanced operation with a relatively simple, predominantly linear topology of transmission grids, given their scale and structure.

On the contrary, distribution grids, which transport electricity from substations to end consumers, present distinct challenges. Their unbalanced nature, radial or weakly meshed topology, high R/X ratios, and cost inefficiency to achieve sufficient sensor coverage complicate the state estimation process. Initially designed with transmission systems in mind, conventional methods often struggle to provide accurate state estimation in these more complex, dynamic, and less predictable distribution systems [1].

However, with the proliferation of Distributed Energy Resources (DERs) and other complex consumers, grid operators are facing the necessity of performing PSSE for distribution grids. Additionally, §14a of the German Energy Industry Act effectively requires operators to develop transparency in distribution grids in order to align consumption with production from renewable energy sources, which requires PSSE.

In this paper, we begin by reviewing relevant prior work in Section II, followed by a formal statement of our research question in Section III. Section IV details the methodology,

including model selection, data preprocessing, and experimental setup. We present and analyze our results in Section V and discuss their implications. Finally, Section VII summarizes our findings and suggests directions for future work.

## II. RELATED WORK

The traditional and most widely-used approach for PSSE is the Weighted Least Squares (WLS) method [2]. This algorithm minimizes the sum of the squared differences between the observed and estimated measurements, with each term being weighed inversely proportionally to the square of the measurement error standard deviation.

However, the WLS algorithm is computationally intensive. Its time complexity is generally considered to be  $\mathcal{O}(N^3)$  in the number of buses  $N$ , assuming a dense system matrix [2]. This is due to the need for matrix inversions and solving linear equations. This complexity can become a limitation for large-scale power systems with thousands of buses, leading to significant computational burden and time constraints, especially when real-time or near-real-time estimations are required. Additionally, WLS assumes that all error distributions are Gaussian, a condition that may not always hold true in practice.

To overcome these limitations, an increasing number of publications instead use Artificial Neural Networks (ANNs) for PSSE, a combination that is called Neural State Estimation (NSE). ANNs may be able to perform the calculation faster than iterative solutions and achieve a higher solution quality simultaneously [3][4]. However, like all Machine Learning (ML) methods, the performance of ANNs is contingent on the quality and quantity of the available training data. Therefore, NSE approaches are usually valid only for the grid they have been trained on. Once the topology or characteristics of nodes change, the ANN needs to be retrained. This is known as the problem of Transfer Learning (TL).

The most logical way to overcome this limitation is to use models that incorporate information about the graph topology into their calculations. Such models are known under an umbrella term Graph Neural Networks (GNNs). Expectedly, recent years have seen a high volume of publications that propose utilizing GNNs for NSE in various ways. To name a few examples:

- Park *et al.* [5] lays important groundwork in comparing different matrix representations of graphs within the Graph Convolutional Network (GCN) model;

- Kundacina *et al.* [6] utilizes Graph Attention Networks (GATs) with a different graph representation of the power grid;
- Hossain and Rahnamay-Naeini [7] explore the possibility of utilizing temporal correlations in the datasets using recurrent GCNs.

However, to our knowledge, none of these research projects specifically considered the problem of Zero-Shot Learning (ZSL) in PSSE. The contribution of this work is in setting up multiple evaluation scenarios for ZSL and testing different configurations of GNNs in them.

### III. RESEARCH QUESTION

When discussing the ability of a model to generalize to different grid topologies, it is important to differentiate between *homogeneous* and *heterogeneous* modes of TL. In general, homogeneous TL mode means that the source and target data are in the same feature space, while in heterogeneous TL mode, they are represented in different feature spaces.

In the context of power grids, this is the difference between two use cases. In the homogeneous case, the power grid remains the same, but some connections between its nodes appear or disappear due to changes in switch states or elements going in and out of service. In the heterogeneous case, the model trained on one grid is used to make predictions about a completely different grid [8].

This distinction becomes very important in production environments. Integrating a model into the control system of a real grid naturally takes time, and training the model on that specific grid could be incorporated into this process without noticeably slowing it down. On the other hand, changes in grid topology due to switching can happen suddenly and unpredictably, and the model must adapt to them in real-time.

There is also another way in which the data distribution can shift in the context of PSSE: the observable subset of buses can change, which changes the amount of input data points available to the model. This can also be considered a form of homogeneous TL.

A subset of TL is Zero-Shot Learning (ZSL). This scenario excludes the possibility of fine-tuning the model on the new distribution and evaluates its performance directly after the transfer. In this project, we specifically focus on ZSL because it is more representative of real-life situations where a model must make predictions immediately after a topology change without access to any training data for fine-tuning. In other words, the model should be *robust* to distributional shifts.

Of course, in practice, a model can be fine-tuned to provide the best performance for the new topology. Still, until this process is complete, the previous version of the model has to substitute for it and provide good enough estimations, even if they are of lower quality.

The research question for this paper is which existing models in application to the PSSE problem are robust to changes in the data distribution, specifically:

A To the reduction of the subset of observable buses;

B To grid topology changes resulting from changing switch states;

C To transfer to a completely different power grid.

## IV. METHODOLOGY

### A. Model selection

The general question of model selection for NSE was addressed by us previously in [9]. The main conclusion from that paper was the selection of GNNs as the most promising direction for further research. Now, we will perform a similar comparison study within the GNN family. We are comparing four models using the implementations provided by PyTorch Geometric framework [10]:

- 1) Graph Convolutional Network (GCN) as proposed in [11]
- 2) Graph Attention Network (GAT) as proposed in [12]
- 3) Graph Isomorphism Network (GIN) as proposed in [13]
- 4) Graph Sample and Aggregate (GraphSAGE) as proposed in [14]

### B. Graph representation of power systems

A successful application of GNN models naturally depends on how well the underlying data can be represented in the graph format. The first step is to represent buses in the grid as nodes of the graph and lines as its edges. In this project, we also represented transformers as edges without any additional parameters. For this to work, the voltage levels across the transformer must be normalized to avoid large voltage gradients.

It is also theoretically advantageous to use a weighted graph with line admittances as weights. Admittances are chosen because the graph Laplacian operator assumes higher edge weights to mean a higher correlation between nodes. This operator is, in turn, used in both the GNN models and the feature propagation algorithm discussed in the next subsection. It should be noted that the models in question support neither complex-valued weights nor multidimensional weights, so we have to use the magnitude of the true complex impedance.

However, using admittance instead of impedance as edge weights becomes a problem for representing closed switches, which have zero impedance and, therefore, infinite admittance. This problem is solved by fusing buses connected by closed bus-to-bus switches into one bus. This is complicated because multiple closed switches are often connected to the same bus, so a naive approach of fusing adjacent buses in random order does not work. Instead, we use an iterative algorithm. Firstly, we build an auxiliary graph of just the closed bus-to-bus switches with buses as nodes and switches as edges. In this graph, nodes with a degree of one can be safely removed (fused with their adjacent buses). This will, in turn, lower the degree of the adjacent node. Eventually, every node will reach a degree of one and can be fused until every connected component of the auxiliary graph is fused into a single node.

### C. Data preprocessing

GNN models are geometrically isodimensional, meaning that each output node must have a corresponding input node.

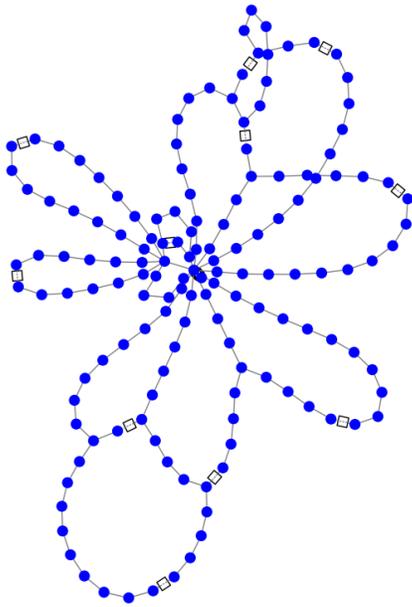


Figure 1: A visualization of the SimBench 1-MV-urban-1-sw grid

This presents a problem for the PSSE use case, where we lack input features for many, if not most, input nodes. The question, therefore, is: How do we initialize the missing features in the input data?

The solution we chose is the feature propagation algorithm from [15], which interpolates missing node-level features by solving a heat equation with known features as boundary conditions. This results in a smooth interpolation of features between known nodes and forms a starting point for the subsequent application of GNNs.

#### D. Datasets

The main dataset used in this project is the SimBench *1-MV-urban-1-sw*, a 147-node, 10 kV medium voltage grid [16] depicted in Figure 1. It is composed of a grid model and a per-bus complex (active and reactive) power yearly time series. To calculate the resulting grid state, we performed a power flow calculation using SIMONA energy system simulation [17]. The resulting dataset comprises the base data and a year of complex voltage time series with a 15-minute temporal resolution. This dataset is hereafter called PQ.

Most grid branches in this model are of the open loop type, which means an open switch (depicted as a square) connects two separate branches. To simulate a realistic topology change, we made a line in one of the open loop branches inoperable, resembling a line fault, and closed the loop switch to resupply all nodes. Performing this operation on different branches resulted in multiple variations of the base grid topology. Afterward, we reran the simulation for each variation to obtain a topology change dataset, which is referenced hereafter as TC.

Unfortunately, the base dataset did not contain information about measurement devices. Therefore, we had to choose observable nodes randomly based on an observability level of 50%, which we assume is realistic for distribution grids.

This means that the state estimator has access to true voltage values for half of the grid buses.

An auxiliary dataset used in the heterogeneous ZSL experiments is based on the CIGRE medium voltage distribution network from Pandapower [18]. It is a much smaller grid with only 15 nodes, which allows us to study how the complexity of the grids affects the performance of ZSL. The voltage data for it is generated using the Midas simulation framework [19]. The shorthand name for this dataset is MV.

#### E. Use cases

Our experiments will be composed of three benchmarks that we call use cases. They correspond to the three subquestions of the main Research question.

In the first use case corresponding to subquestion A, we train the model on the grid with a baseline level of observability and then linearly reduce it from the baseline level to zero at testing time. Of course, the model performance decreases along with this reduction. The shorthand name of this use case is observability degradation (OD).

The second use case corresponds to subquestion B and tests ZSL for homogeneous topology changes. In it, we split the TC dataset in a 50:50 ratio, train the model on the first part, and evaluate on the second. We also evaluate another model trained on the PQ dataset on the TC testing subset to see if the model needs to observe the topology changes happening in order to be able to adapt to them at testing time, but our null hypothesis is that this is not the case. This scenario has the shorthand name TC1, and the former, where the model is trained on the TC dataset, is called TC2.

The third use case corresponds to subquestion C and covers the heterogeneous ZSL scenario. Here, we transfer the model between the PQ and MV datasets in both directions, that is, training on one and then testing on another. The scenario in which the model is trained on PQ and tested on MV has the shorthand name PQ2MV, and the other has MV2PQ.

#### F. Experiment setup

Let us now establish the full hyperparameter space for the models in question. It consists of the following dimensions:

- Model, as listed in the Model selection subsection. Categorical parameter with four values.
- Number of layers in the model. Integer parameter that we limit to 10.
- Use of feature propagation (as opposed to initializing the missing features with zeros). Boolean parameter.
- Use of admittances as edge weights (as opposed to not using any edge weights). Boolean parameter.

Unfortunately, preliminary experiments have demonstrated that the hyperparameter space is not separable, meaning that a full grid search of the space is required. We performed this search for all model configurations and use cases and collected the Mean Squared Error (MSE) metric for each one. The results of this experiment are organized into an evaluation table where rows correspond to model configurations and columns are the following:

TABLE I: BEST CONFIGURATIONS FOR OBSERVABILITY DEGRADATION (OD)

model	layers	fp	adm	mse
GraphSAGE	3	True	False	0.86
GraphSAGE	2	True	True	0.87
GraphSAGE	3	True	True	0.87
GraphSAGE	2	True	False	0.88
GCN	3	True	False	0.90

TABLE II: BEST CONFIGURATIONS FOR SWITCHING CHANGES (TC1)

model	layers	fp	adm	mse
GraphSAGE	3	True	False	0.31
GraphSAGE	1	True	True	0.33
GAT	1	True	False	0.33
GraphSAGE	2	True	False	0.34
GraphSAGE	3	True	True	0.34

- 1) “model” is the name of the model;
- 2) “layers” is the number of layers in the model;
- 3) “fp” is a binary parameter indicating whether feature propagation is used;
- 4) “adm” is a binary parameter indicating whether admittance weights are used;
- 5) “mse” is the value of MSE for the configuration defined by the above parameters.

The full table is available in our repository in the “results.csv” file, and in the next section, we will use subsets of it as illustrations of results.

## V. EVALUATION

In this section, we will analyze the results of the full grid search, attempting to answer the following questions:

- 1) Which model configurations perform best for each use case?
- 2) How does model complexity affect performance?
- 3) How do the data augmentations proposed in the Methodology section affect performance?
- 4) How is performance on different tasks correlated?

The answers to these questions will then be used to answer the main research questions from Section III.

### A. Ranking model configurations

To interpret the numerical results listed in this section, it is useful to keep in mind the baseline value of MSE obtained by evaluating the models trained on the first half of the PQ dataset on the second half of the same dataset and taking the best result. This value is 0.32. We can then broadly say that ZSL is possible in scenarios where the value of MSE after the topology change does not significantly exceed it.

The winning model for the first use case is Graph Sample and Aggregate (GraphSAGE) utilizing feature propagation. It also appears from Table I that there is a sweet spot in model complexity of 2-3 layers.

In both scenarios of the second use case (Tables II and III), the model rating is similar, which suggests that the

TABLE III: BEST CONFIGURATIONS FOR SWITCHING CHANGES (TC2)

model	layers	fp	adm	mse
GraphSAGE	1	True	False	0.32
GraphSAGE	1	True	True	0.32
GAT	3	True	False	0.32
GAT	1	True	True	0.32
GAT	1	True	False	0.32

TABLE IV: BEST CONFIGURATIONS FOR HETEROGENEOUS TRANSFER (PQ2MV)

model	layers	fp	adm	mse
GCN	1	False	True	0.30
GCN	2	False	True	0.87
GCN	4	False	True	0.97
GAT	2	False	False	1.06
GraphSAGE	2	False	True	1.26

tasks themselves are similar as well. The winning models are GraphSAGE and GAT, also with the help of feature propagation and, curiously, in their shallowest versions, with single-layer models showing some of the best results. The main observation, however, is that the MSE values are identical to the baseline, meaning that homogeneous ZSL works very well.

In the third use case, we see a significant difference between the two scenarios. In the first scenario (Table IV), where the model is transferred from a larger to a smaller grid, the best-performing by a large margin is a single-layer GCN, which is the simplest of all the compared models. A possible explanation is that there are only a few correlations that are reusable between grids, which the simple model can capture. Any more complex model picks up too many correlations that are specific to the grid it was trained on and then misapplies them. In the second scenario (Table V), the results are mixed between complex and simple models, and we cannot come up with a sound theoretical interpretation of this result.

### B. Impact of model complexity

In Figure 2, we plot the performance of models against their trainable parameter counts. We also plot the average for all models of a given complexity: the “Mean” line on the graph.

Here, we can see a break point at about 84 parameters or 8 layers, starting from which the performance of models becomes much more consistent between use cases and configurations. The explanation for this effect is the over-smoothing phenomenon described in [20]. In short, GNN layers of all

TABLE V: BEST CONFIGURATIONS FOR HETEROGENEOUS TRANSFER (MV2PQ)

model	layers	fp	adm	mse
GAT	6	True	False	0.62
GAT	2	True	False	0.64
GCN	2	True	True	0.67
GIN	1	True	True	0.69
GAT	2	False	True	0.78

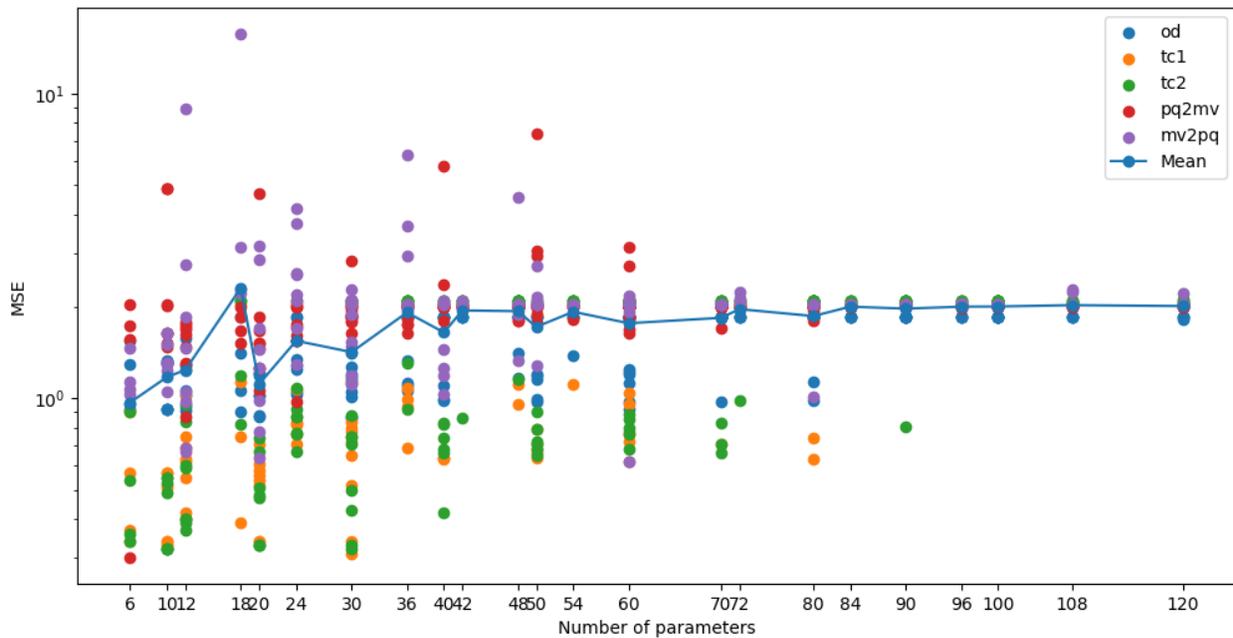


Figure 2: Performance as a function of model complexity

TABLE VI: IMPACT OF DATA AUGMENTATIONS

	fp	adm	od	tc1	tc2	pq2mv	mv2pq
True	True	1.33	1.17	1.13	2.19	1.80	
True	False	1.25	1.03	1.06	1.99	2.55	
False	True	1.51	1.24	1.25	1.93	2.00	
False	False	1.49	1.23	1.10	1.98	2.09	

architectures tend to act as low-pass filters, which effectively averages the output values over multiple iterations. Eventually, the model converges to an output where the values at all nodes of the graph are identical.

Since we are not interested in over-smoothed results, we can drop the model configurations that output them from further analysis. Therefore, the following sections will use the results table truncated to a maximum of 7 layers.

C. Impact of data augmentations

In Table VI, we average the performance values across all model configurations, leaving only the data augmentations as parameters. The results are unsurprising and follow the observations we already made previously. Homogeneous scenarios benefit from feature propagation but are held back by admittance weights. In the heterogeneous scenarios, we once again see a split where MV2PQ benefits from admittance weights and PQ2MV does not. On average, the impact of the augmentations is not very significant.

D. Correlation analysis

To explore the correlations between hyperparameters and use cases, we compute the Pearson correlation matrix between their associated performance values in Figure 3. Here, we also use the number of trainable parameters (“#params”) instead of layers to compare model complexity more fairly. Note

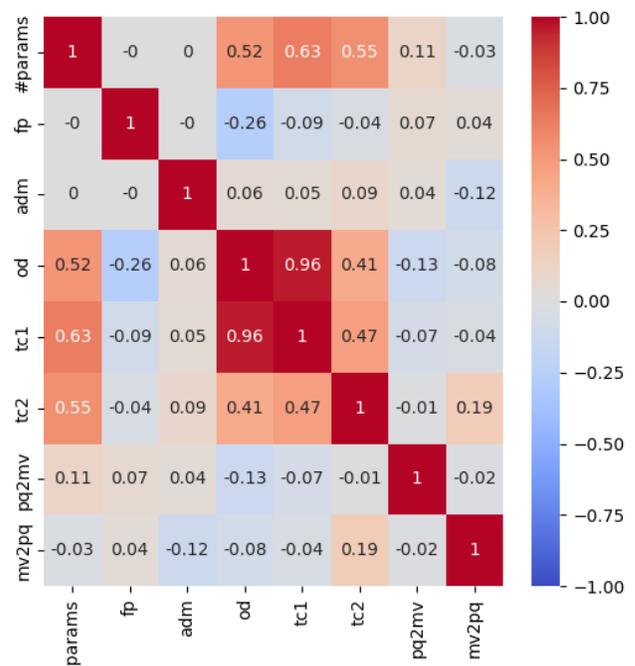


Figure 3: Performance correlation between use cases

that since lower MSE means better performance, a positive correlation between a hyperparameter and performance is shown as negative and vice versa.

The analysis confirms the conclusions that we made previously: model complexity is detrimental to performance for all use cases except MV2PQ, and our proposed augmentations only marginally affect performance, with feature propagation being most useful in homogeneous scenarios.

## VI. LIMITATIONS AND FUTURE WORK

During the evaluation stage of this research, it became evident that MSE alone does not convey enough information to confidently make conclusions about ZSL performance of the models. However, we could not find a better alternative in the literature.

The main problem is that MSE only shows us the instantaneous performance and does not account for the transfer process. An ideal metric  $M$  for ZSL and TL experiments would be a differential one that takes into account the magnitude of the change in the underlying grid  $\Delta G$  and the performance change  $\Delta P$ , for example,

$$M = \frac{\Delta P}{\Delta G}$$

However, an algorithm to compute  $\Delta G$  is not trivial to develop. We hope to tackle this problem in our future work.

## VII. CONCLUSION

The findings of this paper can be summarized as follows:

- 1) GNNs are very robust to homogeneous topology changes in the underlying power grid.
- 2) Some GNNs can perform well in a zero-shot transfer from a larger grid to a smaller one, but not in the other direction.
- 3) Despite the conventional wisdom in the ML community being “Scale is all you need,” scaling GNNs up is not the best way to improve NSE performance. This is explained by the oversmoothing phenomenon [20].
- 4) Measuring the performance of NSE by MSE is not always helpful, especially in the context of ZSL. However, we could not find another commonly accepted metric. The development of such a metric appears to be a research gap at the moment.

*Acknowledgments:* This research is a part of project TRANSENSE, funded by the German Federal Ministry for Economic Affairs and Climate Action (FKZ 03EI6044A).

*Availability of data and source code:* The source code for this project, along with the datasets, is openly available in the following repository:

<https://gitlab.com/transense/nse-tl-paper/tree/IARIA>

## REFERENCES

- [1] F. F. Wu, “Power system state estimation: A survey,” *International Journal of Electrical Power & Energy Systems*, vol. 12, no. 2, pp. 80–87, 1990, ISSN: 0142-0615. DOI: 10.1016/0142-0615(90)90003-T.
- [2] A. Abur and A. G. Expósito, *Power System State Estimation*. CRC Press, Mar. 2004, ISBN: 9780203913673. DOI: 10.1201/9780203913673.
- [3] K. R. Mestav, J. Luengo-Rozas, and L. Tong, “State estimation for unobservable distribution systems via deep neural networks,” in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 2018, pp. 1–5. DOI: 10.1109/PESGM.2018.8586649.
- [4] S. Balduin, T. Westermann, and E. Puiutta, *Evaluating different machine learning techniques as surrogate for low voltage grids*, Oct. 2020. DOI: 10.1186/s42162-020-00127-3.
- [5] S. Park, F. Gama, J. Lavaei, and S. Sojoudi, “Distributed power system state estimation using graph convolutional neural networks,” in *Hawaii International Conference on System Sciences*, 2023.
- [6] O. Kundacina, M. Cosovic, D. Miskovic, and D. Vukobratovic, “Distributed nonlinear state estimation in electric power systems using graph neural networks,” pp. 8–13, 2022. DOI: 10.1109/SmartGridComm52983.2022.9960967.
- [7] M. J. Hossain and M. Rahnamay-Naeini, “State estimation in smart grids using temporal graph convolution networks,” in *2021 North American Power Symposium (NAPS)*, IEEE, Nov. 2021, pp. 01–05. DOI: 10.1109/naps52732.2021.9654642.
- [8] S.-G. Yang, B. J. Kim, S.-W. Son, and H. Kim, “Power-grid stability predictions using transferable machine learning,” *Chaos*, vol. 31 12, p. 123 127, 2021. DOI: 10.1063/5.0058001.
- [9] A. Berezin, S. Balduin, T. Oberließen, E. Veith, S. Peter, and S. Lehnhoff, “Application of recurrent graph convolutional networks to the neural state estimation problem,” *International Journal of Electrical and Electronic Engineering & Telecommunications*, pp. 209–215, 2023, ISSN: 2319-2518. DOI: 10.18178/ijeetc.12.3.209-215.
- [10] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [11] T. Siameh, *Semi-supervised classification with graph convolutional networks*, Dec. 2023. DOI: 10.13140/RG.2.2.22993.71526.
- [12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018.
- [13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, *How powerful are graph neural networks?* 2019.
- [14] W. L. Hamilton, R. Ying, and J. Leskovec, *Inductive representation learning on large graphs*, Long Beach, California, USA, 2017.
- [15] E. Rossi, H. Kenlay, M. I. Gorinova, B. P. Chamberlain, X. Dong, and M. M. Bronstein, “On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features,” *Proceedings of Machine Learning Research*, vol. 198, B. Rieck and R. Pascanu, Eds., 11:1–11:16, Dec. 2022.
- [16] S. Meinecke, D. Sarajlić, S. R. Drauz, A. Klettke, L.-P. Laufen, C. Rehtanz, *et al.*, “Simbench—a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis,” *Energies*, vol. 13, no. 12, p. 3290, Jun. 2020, ISSN: 1996-1073. DOI: 10.3390/en13123290.
- [17] J. Hiry, “Agent-based discrete-event simulation environment for electric power distribution system analysis,” en, Ph.D. dissertation, 2021. DOI: 10.17877/DE290R-22549.
- [18] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, *et al.*, “Pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, 2018. DOI: 10.1109/TPWRS.2018.2829021.
- [19] S. Balduin, E. Veith, and S. Lehnhoff, “Midas: An open-source framework for simulation-based analysis of energy systems,” in *Simulation and Modeling Methodologies, Technologies and Applications*. Springer International Publishing, 2023, vol. 780, pp. 177–194, ISBN: 978-3-031-43823-3. DOI: 10.1007/978-3-031-43824-0\_10.
- [20] K. Oono and T. Suzuki, “Graph neural networks exponentially lose expressive power for node classification,” in *International Conference on Learning Representations*, 2020.