Graph Embeddings for Outage Prediction

Rashid Baembitov, Mladen Kezunovic
Department of Electrical and Computer Engineering
Texas A&M University
College Station, Texas, USA
orcid.org/0000-0002-6515-8007, kezunov@ece.tamu.edu

Zoran Obradovic
Computer and Information Sciences Department
Temple University
Philadelphia, PA, USA
zoran.obradovic@temple.edu

***Abstract*— This paper discusses how the risk of electricity grid outages is predicted using machine learning on historical data enhanced by graph embeddings of the distribution network. The process of graph creation using different embedding approaches is described. Several graph constructing strategies are used to create a graph, which is then transformed into the form acceptable for ML algorithm training. The impact of incorporating different graph embeddings on outage risk prediction is evaluated. The method used for graph embeddings is Node2Vec. The grid search is performed to find optimal hyperparameters of Node2Vec. The resulting accuracy metrics for a set of different hyperparameters are presented. The resulting metrics are compared against base scenario, where no graph embeddings were used.**

Keywords—graph embeddings, machine learning, risk prediction

I. INTRODUCTION

Short circuits in power systems caused by faults lead to outages and are a significant safety hazard and economic detriment to the society. The problem of frequent outages in the distribution network has been one of the main concerns of the utility companies. There is a growing need to improve quality of electrical supply to customers, as well as urgent necessity to enhance the resilience of the grid. Weather related outages constitute a major cause of outages in the distribution grids and are increasing due to climate change [1-3].

Recent advances in Machine Learning (ML) Algorithms offer ways of predicting the risk of outages in the grid and consequently mitigating it. That leads to improved resiliency, reduced economical losses, and higher customer satisfaction. The combination of Big Data, ML, and GIS software allows for analysis of the historical outages and creation of an ML model capable of predicting the risk levels for different time horizons for separate parts of the distribution system [4-7]. A timely prediction of high-risk levels allows utility companies to take preventive measures to reduce the risk. Mitigation actions may include improved dynamic tree trimming schedules, customer notifications, back-up generator start-up, targeted restoration actions for different parts of the system, etc. [8].

Authors of [9-11] use ML model to predict number of outages during severe storms in electric distribution networks. Distribution network resiliency is assessed in [12], using predicted risk levels [13]. [14] uses Logistic Regression method to predict probability of distribution transformers (DT) failure by analyzing a correlation between weather parameters and historical DT failures. These approaches can benefit from using spatial relationship among data of the network. Adding graph embeddings could benefit the accuracy of the predictions [15].

To improve the accuracy and performance of such models, different features (dimensions) may be used as input variables to the ML model. One of the main sources of data are weather indicators, correlated to the outage time and location [16]. The distribution grid may span across hundreds of miles; thus, the underlying geographical conditions may be different for each part of the system. That leads to necessity of incorporating spatial information about each part of the system into the features for the ML algorithm. Graph representation of the system with graph embeddings offers a solution to this problem [17].

Various graph embedding methods are developed by encoding each vertex with its own vector representation, or by representing the entire graph as a single vector. One of the oldest methods is Spectral Clustering [18], which first constructs a similarity matrix from a k-NN similarity graph and then calculates eigenvectors of Laplacian of the similarity matrix for embedding. Perozzi et al. generalized recent advancements in NLP and unsupervised feature learning (or deep learning) and proposed DeepWalk method [19], which became a foundation for many other methods. DeepWalk method allows learning latent representations of vertices in a network by generating truncated random walks in the corresponding graph for each node. It treats each walk as a sentence in a text. Authors demonstrated a substantial 10% increase in F1 score in multi-label network classification tasks. A very efficient method, referred to as Large-scale Information Network Embedding (LINE), was proposed in [20]. The method optimizes an objective function that preserves both the global and local network structures. It is applicable to arbitrary types of information in graphs. Authors of [21] introduced the Node2Vec method. The advantage of this method is its ability to change its underlying search strategy. It can be tuned to focus on the immediate neighbors of each node, or it can be set to explore deep structure of the underlying graph.

We analyze the impact of graph incorporation into the ML prediction model. Our contribution is in deploying different ways of constructing several types of graphs for the distribution network and then conducting a sensitivity analysis to find optimal hyper parameters for graph embedding method. We demonstrate that the graph embeddings increase the model performance measured by variety of metric ML indicators.

The remainder of the paper is structured as follows. Section II discusses various types of graphs suitable for our application.

A graph embedding process is described in Section III, followed by Section IV, which gives details about ML model training and testing. The conclusions are summarized in Section V.

II. Graph Construction

There are number of ways, in which the graph can be constructed. The obvious choice for power system applications is to use the electric grid connectivity graph. The other way would be to use an artificial surface grid placed on top of the geographic representation of the system.

In this work we have chosen a method where we use geographical centroids of different feeders as nodes. The edges are created based on the distance between the centroids of the feeders. This choice is also driven by the prediction entity or prediction object, which in our case is the grid outage risk map.

A. Nodes

The geographical representation of an actual power system we use for our study is shown in Fig. 1. The centroids are shown as dots. A centroid is determined as a geometrical equivalent of center of mass of an object. It is a point located at the weighted average of x and y coordinates of the midpoints of all line segments that form a feeder, where the weight of a particular midpoint is the length of the correspondent line segment [22]. ArcGis Pro allows two methods for calculating a centroid of an object: the “true” one and one that is contained within the object itself. We used the latter option, since we want to find a point in each feeder that represents it as a whole and is actually located somewhere on the feeder.

B. Edges

One can construct edges of a graph from nodes in a different manner. For example, there are number of methods for randomized graph construction [23-26]. In our application we connect nodes with edges based on geographical proximity. The underlying rational is that if feeders are in close geographical proximity, they should possess similar characteristics and should have similar measures of risk. That is in line with the Tobler's first law of geography [27], which states: "everything is related to everything else, but near things are more related than distant things."

We created 4 graphs with 1, 2, 4 and 7 closest nodes being connected by edges (NE). The resulting graphs are presented in Fig. 2-Fig. 5. As can be seen, 1 and 2 closest nodes actually form several subgraphs, which are not connected with each other. And then, starting with 4 nodes, the whole network is interconnected. The number of closest nodes represent the level of complexity of information contained in the graph. One of the aims of this paper is to assess the optimal number of closest nodes with respect to preserving structural properties of the power network. The process of connecting N closest nodes together is performed in ArcGIS Pro by using several tools in specific order. The diagram of the process is presented Fig. 6. First, one needs to determine the set of N closest nodes for each node in the network. The optimal tool for that is “Generate Near Table” [28], where N is specified and also the fields “FROM” and “TO” are generated, to keep track of location of each pair. Second, the Near Table is converted to lines, using “XY to Line” tool [29]. The result is yet not usable since the edges are

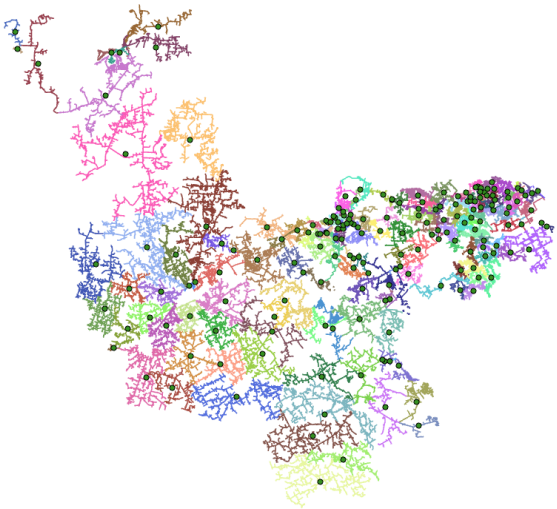

Fig. 1. Geographical Representation of the Network with 191 nodes

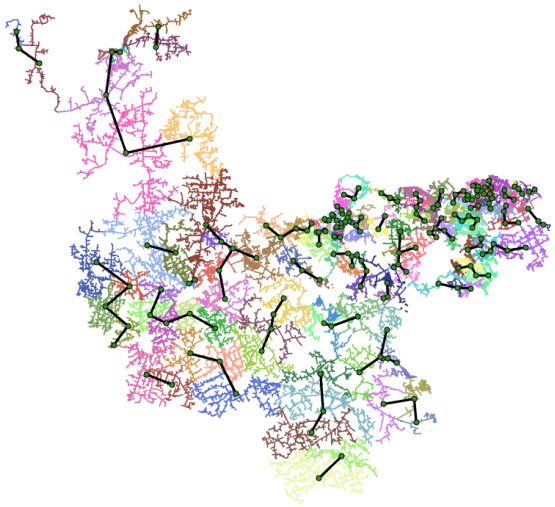

Fig. 2. Graph for NE = 1. 191 nodes, 141 edges.

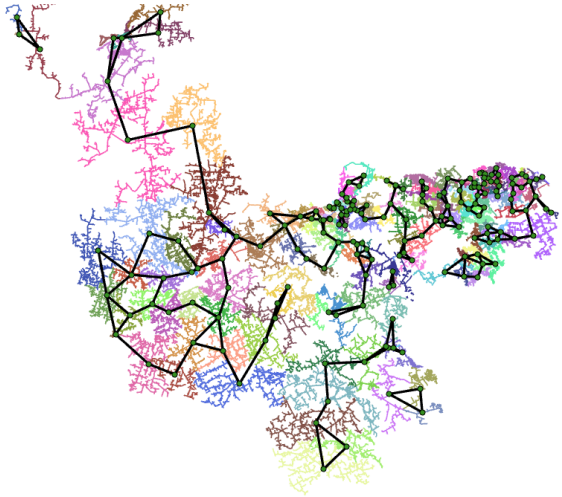

Fig. 3. Graph for NE = 2. 191 nodes, 257 edges.

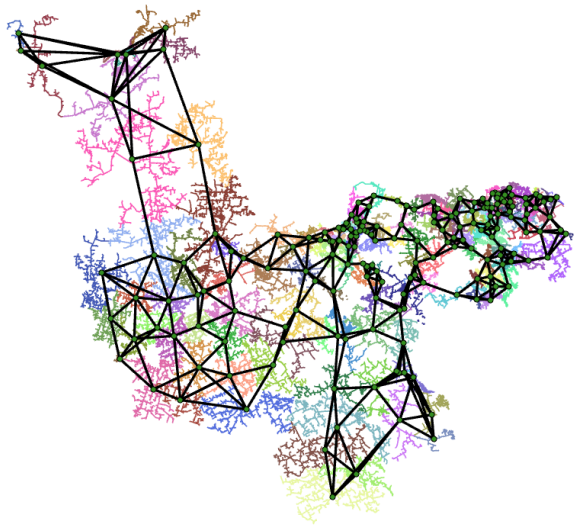

Fig. 4. Graph for NE = 4. 191 edges, 487 edges.

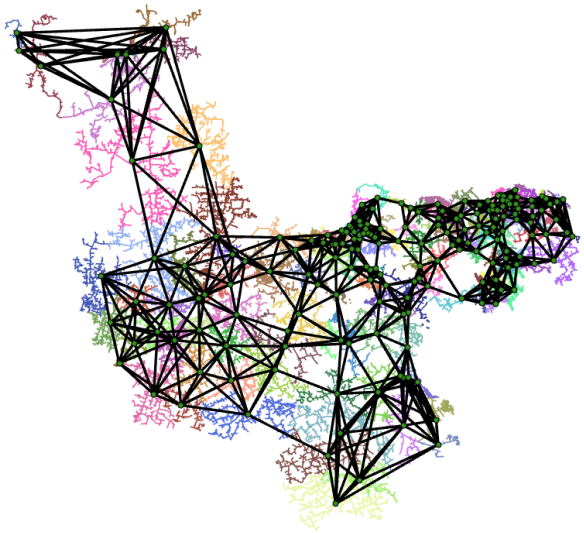

Fig. 5. Graph for NE = 7. 191 nodes, 829 edges.

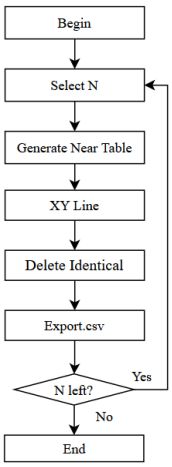

Fig. 6. Graph Generation Diagram

formed twice between some nodes (ex. node K is closest to node M, and M is closest to node K, then 2 rows are generated in Near Table). To discard double edges, the "Delete Identical" tool [30] was used. We should note that if one is to obtain directed graph, the method should be used with caution, since the "Delete Identical" removes first instance of the repeating feature. The resulting feature classes are exported as .csv tables for further processing outside of ArcGis Pro in Python.

III. Graph Embedding

In recent years there was an advent of graph embedding methods (GEM) [31]. In general, the embedding process is mapping of a graph G, which is characterized by its nodes (N), edges (E) and edges' weights (W), to R^N dimensional vector space, which preserves some part of information about the original graph (1).

$$\mathrm{GEM}(G(N, E, W)) \rightarrow R^N \quad (1)$$

Due to its flexibility and ability to differentiate between different search strategies, we have chosen Node2Vec [21] for our application. A description of the method is given below.

The input for the method is a graph G (N, E, W). For each node of the graph the random walks of length WL are generated NW times. The random walk is affected by two hyper parameters: return parameter p and in-out parameter q (Fig. 7). These are used to calculate the probability of going from node v to the preceding node t (go back) or going to any other node (explore). The parameter p correlates with the probability of going back to the previous node, and q with probability going to some other node. In this fashion, one can regulate how much the method tends to gain distance from original node and expand further into the network; or the opposite – how much the method tends to pick nodes that are immediate neighbors of the original node. That defines the sampling strategy of the algorithm. The original paper refers to them as Breadth-first Sampling (BFS) for going back and Depth-first Sampling (DFS) for exploring [21]. The scheme used in the original publication is shown in Fig. 8.

After the walks are generated for all the nodes, the neural network (NN) with one input, one hidden and one output layer, that have dimensions of L, N and L accordingly, is trained to predict the probability of each node having the rest of the nodes as its neighbors, based on the "corpus" of random walks. The L

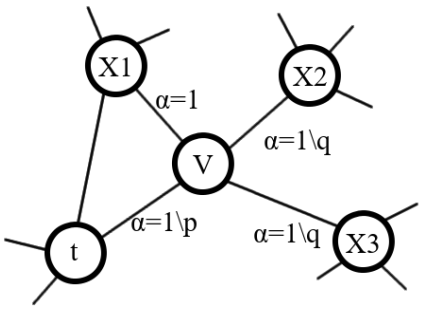

Fig. 7. Node2Vec Graph [21]

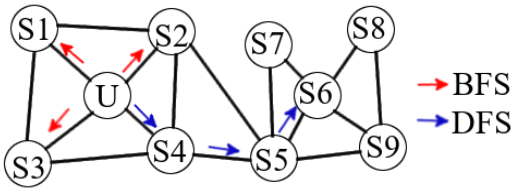

Fig. 8. BFS and DFS search strategies from node U [21]

is the number of unique objects (vocabulary) in "corpus", which in our case is 191 feeders in the network. N is a hyperparameter of the method and it defines the dimensionality of a vector space representation of the embedding method. We have tested several values of N for our application. This process is known as skip-gram [32] and is used in many embedding techniques, for example a well-known Word2Vec. However, after training the NN, the NN itself is not used. Instead, the weight matrix of the inner layer of dimension L by N is used as a final embedding matrix. Based on that matrix, the similarity between nodes can be measured as cosine distance, for instance. We use the matrix elements as additional features to describe the baseline training dataset for the ML model.

IV. ML Algorithm Training and Sensitivity Analysis

A. Base Model

To have comparable results, we establish a baseline solution against which we would check the results of the ML model with graph embedding. The baseline training dataset includes only weather parameters for each node (feeder), namely:

- Air Temperature in Fahrenheit, typically at the height of 2 of meters
- Dew Point Temperature in Fahrenheit, typically at the height of 2 meters
- Relative Humidity in %
- Wind Direction in degrees from "true" north
- Wind Speed in knots
- One hour precipitation in inches for the period from the observation time to the time of the previous hourly precipitation reset.
- Wind Gust in knots.
- Present Weather Codes.

The weather parameters are obtained from ASOS network [33] by means of API provided by Iowa Environmental Mesonet [34]. The target feature is the occurrence of an outage within an hour of analyzed timestamp. The historical outage dataset is acquired from the utility company that provided data from their network. The outages for the duration of three years from January 2015 to December 2017 are analyzed. There is a total of 5615 outages, each associated with single feeder. The model is trained to predict risk of outage for whole feeder. To balance the dataset with non-outage instances, the same amount of 5615 non-outage timestamps were randomly selected, that are at least 3 hours apart from the outage occurrence.

The original dataset is cleaned, preprocessed, and wrangled to fit the input of ML algorithm. The ML algorithm used is Catboost [35], which has proved its efficiency in several ML problems [36],[37]. It is the Gradient boosting algorithm type. The Catboost has an advantage of automatically recognizing the categorical features, which makes the process of training faster and more efficient.

We use 5-fold cross validation for obtaining performance metrics. The metrics in use are Area Under the Receiver operating characteristic (ROC AUC), F1 Score, Area Under the Precision-Recall Curve (PRC AUC) [38]. We also introduce the final compound metric (FM), which is the weighted average of the three previous metrics (ROC AUC has a bigger weight, because it was used as loss function for model training)

$$FM = 0.4 \cdot ROC\_AUC + 0.3 \cdot F1 + 0.3 \cdot PRC\_AUC \quad (2)$$

We use the final metric for a better representation of the model performance with different embeddings. The results of the model applied on the baseline dataset are presented in Table I.

B. Model with Graph Embeddings

After training the baseline model we move on to adding graph embeddings to the base dataset. We join the embedding to the cleaned and preprocessed baseline training dataset. In this way, the dataset is identical, except for the additional dimensions of graph embeddings. It is important to have same rows in baseline and modified training datasets, otherwise the results may have noise and become non-comparable. In general, the additional features (dimensions) to training dataset may need cleaning and preprocessing as well. That may lead to some of the rows of the baseline dataset being discarded after the additional features are added. A precaution should be taken to track such changes and adjust final datasets accordingly.

In each algorithm run we change one of the hyperparameters of the method. A unique combination of hyperparameters is coded with its own number (UID). The combinations for sensitivity analyses are created from following sets of hyperparameters:

- Number of closest points, connected with edges (NE): (1, 2, 4, 7)
- Embedding dimensions (N): (15, 40, 85)
- Length of single walk (WL): (4, 10, 16)
- Window size for skim-gram (WS): (3,6)

There are total of 72 unique runs in the experiment. Some of the hyperparameters for the algorithm are fixed. These are Number of walks per node: (NW=100), Minimum length of one walk: (min count=1), Return parameter: (p=1), In-out parameter: (q=3). Also, we fixed the random seed parameter for reproducible results. The choice of return and in-out parameter inclines the model to BFS strategy, since we want to capture the immediate neighbors of the feeders.

The results for model with graph embeddings, organized as bar-chart for different hyper parameters, are presented in Fig. 9-Fig. 12. Numbers above bars indicate UID of each case. The scale of FM axis is kept the same throughout all figures to keep

Table I. Baseline Model Metrics

	ROC AUC	F1 Score	PRC AUC	FM
Baseline Model	0.939	0.856	0.944	91.57

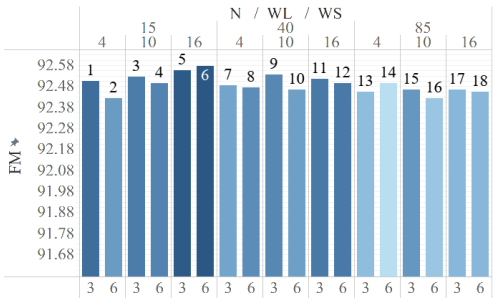

Fig. 9. Results for 1-Edge embedding

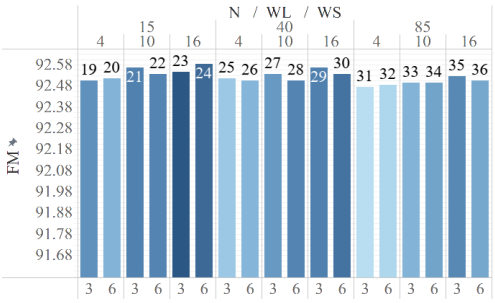

Fig. 10. Results for 2-Edge embedding

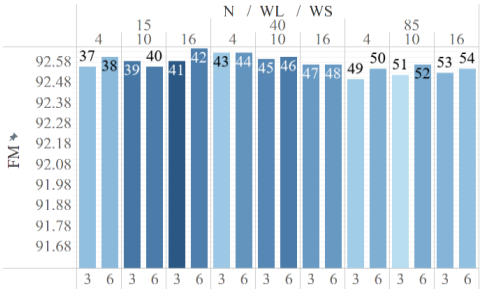

Fig. 11. Results for 4-Edge embedding

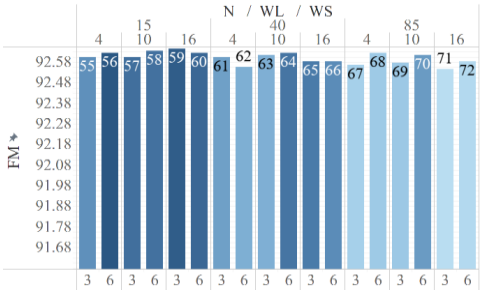

Fig. 12. Results for 7-Edge embedding

representation of results easy to read. We also used a darker color for bars with higher ROC AUC score, since it was used as objective function for training the algorithm.

C. Results discussion

We assume that the FM represents the desired balance in performance metrics. One can tweak the FM's weight coefficients to rise importance of some metric or one can even make judgement on a single metric. The decision should be based on the cost of false positives and false negatives in the model performance.

All graph embedding-based experiments resulted in significant improvements versus baseline, as evident at Figs 9-12, where bars correspond to improvement versus baseline result. For the 1-edge embedding the best case is UID = 6, with 15 dimensions, window size of 6 and walk length of 16. This shows that by even adding a trivial graph we improve model performance up by 1%. Adding more edges yields better results. For 2-edge embedding, the best outcome has a FM of 92.58% (N = 15, WL = 16, WS = 6). When increasing closest nodes in graph to 4, the best outcome is 42 with the same hyperparameters. The increase in FM is 1.07%. Moving to 7 edges, the best outcome is number 59 with 15 dimensions, walk length of 16 and window size of 3. In latter case we do not see any increase in performance as compared to 4-edge embedding. In fact, we experienced that increasing graph complexity above 4 closest nodes does not yield better results.

One can see that the strongest cases for each type of graph have similar hyperparameters: N = 15, WL = 16, WS = 6. We may conclude that those are optimal for the application in hand. Hence, 15 dimensions are enough to represent spatial difference between feeders in the network, and increasing the amount only degrades the performance. WL and WS of higher levels allow the model to capture more neighbors, than it otherwise does with lower values.

V. Conclusion

In this paper, we analyzed the effect of spatial information embedding into features by constructing a graph and then passing it through graph embedding method. The resulting features were used to train the ML model. Several important outcomes are achieved:

- Graph embedding improves prediction accuracy of risk outages in the network.
- 4-edge graph is optimal for application at hand.
- Sensitivity analysis reveals optimal hyper parameters for the prediction model.

Acknowledgment

The utility data used in this project is provided by United Cooperative Services and funding is made available through the Smart Grid Center Membership Agreement. Special thanks go to United staff Mr. Jared Wennemark for the management guidance and Mr. Cory Menzel for coordination of the technical support.

References

[1] "Economic Benefits of Increasing Electric Grid Resilience to Weather Outages," Executive Office of President Report, Aug. 2013.

[2] A. Murzintsev, A. Korolev, K. Zhgun, and R. Baembitov, "Short-circuit Current Reduction in Auxiliary Network of Traction Substations," *Transportation Research Procedia,* vol. 54, pp. 346-354, 2021, doi: 10.1016/j.trpro.2021.02.082.

[3] A. Vlachokostas *et al.*, "Ship-to-grid integration: Environmental mitigation and critical infrastructure resilience," in *2019 IEEE Electric Ship Technologies Symposium (ESTS)*: IEEE, 2019, pp. 542-547.

[4] M. Kezunovic, Z. Obradovic, T. Djokic, and S. Roychoudhury, "Systematic Framework for Integration of Weather Data into Prediction Models for the Electric Grid Outage and Asset

Management Applications," in *The Hawaii International Conf. on System Sciences - HICSS*, Waikoloa Village, Hawaii, USA, January 2018, pp. 2737-2746, doi: 10.24251/HICSS.2018.346.

[5] M. Kezunovic, P. Pinson, Z. Obradovic, S. Grijalva, T. Hong, and R. Bessa, "Big data analytics for future electricity grids," *Electric Power Systems Research,* vol. 189, p. 106788, 2020, doi: 10.1016/j.epsr.2020.106788.

[6] M. Kezunovic and T. Dokic, "Big Data Framework for Predictive Risk Assessment of Weather Impacts on Electric Power Systems," in *Grid of the Future, CIGRE US Nat. Committee*, Atlanta, Nov., 2019.

[7] A. Ghasemi, A. Shojaeighadikolaei, K. Jones, M. Hashemi, A. G. Bardas, and R. Ahmadi, "A Multi-Agent Deep Reinforcement Learning Approach for a Distributed Energy Marketplace in Smart Grids," in *2020 IEEE International Conf. on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1-6, doi: 10.1109/SmartGridComm47815.2020.9302981.

[8] T. Dokic and M. Kezunovic, "Predictive Risk Management for Dynamic Tree Trimming Scheduling for Distribution Networks," *IEEE Transactions on Smart Grid,* vol. 10, no. 5, pp. 4776-4785, 2018, doi: 10.1109/TSG.2018.2868457.

[9] F. Yang, D. Cerrai, and E. N. Anagnostou, "The Effect of Lead-Time Weather Forecast Uncertainty on Outage Prediction Modeling," *Forecasting,* vol. 3, no. 3, pp. 501-516, 2021, doi: 10.3390/forecast3030031.

[10] D. Cerrai *et al.*, "Predicting Storm Outages Through New Representations of Weather and Vegetation," *IEEE Access,* vol. 7, pp. 29639-29654, 2019, doi: 10.1109/ACCESS.2019.2902558.

[11] D. Cerrai, M. Koukoula, P. Watson, and E. N. Anagnostou, "Outage prediction models for snow and ice storms," *Sustainable Energy, Grids and Networks,* vol. 21, p. 100294, 2020, doi: 10.1016/j.segan.2019.100294.

[12] J. B. Leite, J. R. S. Mantovani, T. Dokic, Q. Yan, P.-C. Chen, and M. Kezunovic, "Resiliency Assessment in Distribution Networks Using GIS-Based Predictive Risk Analytics," *IEEE Transactions on Power Systems,* vol. 34, no. 6, pp. 4249-4257, 2019, doi: 10.1109/TPWRS.2019.2913090.

[13] J. B. Leite and M. Kezunovic, "Risk Mitigation Approaches for Improved Resilience in Distribution Networks," in *2020 IEEE PES Transmission & Distribution Conf. and Exhibition - Latin America (T&D LA)*, 2020: IEEE, pp. 1-6, doi: 10.1109/TDLA47668.2020.9326210.

[14] E. H. Ko, T. Dokic, and M. Kezunovic, "Prediction Model for the Distribution Transformer Failure Using Correlation of Weather Data," in *5th International Colloquium on Transformer Research and Asset Management*, Trkulja B., Štih Ž., and J. Ž. Eds. Singapore: Springer, 2020.

[15] T. Vujicic, J. Glass, F. Zhou, and Z. Obradovic, "Gaussian conditional random fields extended for directed graphs," *Machine Learning,* vol. 106, no. 9, pp. 1271-1288, 2017/10/01 2017, doi: 10.1007/s10994-016-5611-7.

[16] P. Dehghanian, B. Zhang, T. Dokic, and M. Kezunovic, "Predictive Risk Analytics for Weather-Resilient Operation of Electric Power Systems," *IEEE Transactions on Sustainable Energy,* vol. 10, no. 1, pp. 3-15, 2018, doi: 10.1109/TSTE.2018.2825780.

[17] C. Han, S. Zhang, M. F. Ghalwash, S. Vucetic, and Z. Obradovic, "Joint Learning of Representation and Structure for Sparse Regression on Graphs," in *Proc. of the 2016 SIAM Int. Conf. on Data Mining (SDM)*, Miami, FL, USA, 2016, pp. 846-854, doi: 10.1137/1.9781611974348.95.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *NIPS'01: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, USA: MIT Press, 2001, pp. 849-856.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: online learning of social representations," in *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: Association for Computing Machinery, 2014, pp. 701-710.

[20] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale Information Network Embedding," in *WWW '15: Proceedings of the 24th International Conference on World Wide Web*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015, pp. 1067-1077.

[21] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," *KDD,* vol. 2016, pp. 855-864, 2016, doi: 10.1145/2939672.2939754.

[22] ESRI. "ArcGIS Pro Documentation. Feature To Point (Data Management)." https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/feature-to-point.htm (accessed Jul. 2021).

[23] J. Wang and Y. Xia, "Fast Graph Construction Using Auction Algorithm," *ArXiv e-prints,* 2012. [Online]. Available: https://arxiv.org/abs/1210.4917v1.

[24] E. Plaku and L. E. Kavraki, "Distributed computation of the knn graph for large high-dimensional point sets," *Journal of Parallel and Distributed Computing,* vol. 67, no. 3, pp. 346-359, 2007, doi: 10.1016/j.jpdc.2006.10.004.

[25] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery, 2009, pp. 441-448.

[26] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *WWW '11: Proceedings of the 20th international conference on World wide web*. New York, NY, USA: Association for Computing Machinery, 2011, pp. 577-586.

[27] W. R. Tobler, "A Computer Movie Simulating Urban Growth in the Detroit Region," *Economic Geography,* vol. 46, Proceedings. International Geographical Union. Commission on Quantitative Methods, pp. 234-240, 1970.

[28] ESRI. "Generate Near Table (Analysis)—ArcGIS Pro | Documentation." https://pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/generate-near-table.htm (accessed Jul. 2021).

[29] ESRI. "XY To Line (Data Management)—ArcGIS Pro | Documentation." https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/xy-to-line.htm (accessed Jul. 2021).

[30] ESRI. "Delete Identical (Data Management)—ArcGIS Pro | Documentation." https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/delete-identical.htm (accessed Jul. 2021).

[31] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems,* vol. 151, pp. 78-94, 2018, doi: 10.1016/j.knosys.2018.03.022.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ArXiv e-prints,* 2013. [Online]. Available: https://arxiv.org/abs/1301.3781v3.

[33] "Automated Surface Observing Systems." NOAA's National Weather Service. https://www.weather.gov/asos/asostech (accessed 2021).

[34] Iowa Environmental Mesonet: ASOS One Minute Data Download [Online] Available: https://mesonet.agron.iastate.edu/request/asos/1min.phtml

[35] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *ArXiv e-prints,* 2018. [Online]. Available: https://arxiv.org/abs/1810.11363v1.

[36] R. Baembitov, T. Dokic, M. Kezunovic, Y. Hu, and Z. Obradovic, "Fast Extraction and Characterization of Fundamental Frequency Events from a Large PMU Dataset using Big Data Analytics," in *54th Hawaii International Conference on System Sciences*, 2021, pp. 3195-3204, doi: 10.24251/HICSS.2021.389.

[37] P. S. Kumar, K. Anisha Kumari, S. Mohapatra, B. Naik, J. Nayak, and M. Mishra, "CatBoost Ensemble Approach for Diabetes Risk Prediction at Early Stages," in *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON)*: IEEE, 2021, pp. 1-6.

[38] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Mach. Learn. Technol,* vol. 2, no. 1, 2008.