Machine Learning-Driven Prediction of Load Shedding During Cascading Outages

George Paphitis[1], Balaji V. Venkatasubramanian[2], Mathaios Panteli[1]
[1]Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus
[2]School of Technology, Woxsen University, Telangana, India
[pafitis.g.georgios, panteli.mathaios]@ucy.ac.cy;
balaji.venkateswaran@woxsen.edu.in

***Abstract*—The complexity of modern power systems makes predicting and mitigating cascading outages challenging. Quantifying their impact requires extensive simulations, which are computationally expensive and impractical for real-time applications. To address this issue, we introduce a novel methodology that integrates different machine learning regressors to create a stacking ensemble regressor along with a cascading failure model enabling rapid quantification of cascading-driven nodal load shedding through prediction. The effectiveness of the proposed model is evaluated using the R^2 metric, with studies conducted on the IEEE 39-bus system. Additionally, the model is tested on the IEEE 24-bus Reliability Test System to demonstrate the effect of network reliability on cascading failures and its impact on model fitness. The outcomes of this study indicate that the proposed model fits better on the IEEE 39-bus system, achieving an R^2 of 0.88. In contrast, on the IEEE 24-bus Reliability Test System, the R^2 is 0.42; the reasons behind this discrepancy are examined and discussed.**

***Index Terms*—Cascading Failures, Load Shedding, Machine Learning, Reliability, Resilience**

I. Introduction

In an interconnected power system, the outage of critical components, either due to common failures or high-impact low-probability (HILP) events, such as natural hazards and extreme weather events, can potentially trigger protection mechanisms. These mechanisms, in turn, could lead to a series of cascading failures throughout the network causing significant load shedding (LS) and compromising the reliability of the system. Dynamic simulation models are applied for an in-depth analysis of the dynamic stability of cascading failures that are propagated across the network caused by an initiating event, such as common failures and cyber attacks [1], [2]. In addition to modeling protection mechanisms within a dynamic model, time is explicitly considered. The model is solved across a large range of time steps, which increases the computational complexity. Therefore, researchers have proposed different quasi-dynamic cascading failure models with significantly lesser time steps compared to dynamic model, still reflecting the behavior of protection mechanism for an initiating event [3]. However, most of them are restricted to specific protection mechanisms like line overloading.

This work was supported by the Horizon Europe program through the project “Reliability, Resilience and Defense Technology for the Grid” (R2D2) (Grant agreement ID: 101075714). 979-8-3503-9042-1/24/$31.00 ©2024 IEEE

Meanwhile, [4] proposed a quasi steady-state cascading failure model that reflects most common protection mechanisms such as under/over frequency, excitation limits, and line overloading. Further, it highlights that this model can be applicable to analyze cascading-driven resilience analysis. In the context of analyzing the impact of a possible future event, stochastic approaches are being employed that generates a large set of initiating event scenarios [5]. For this purpose, quantifying the cascading failure impact and its propagation via quasi-dynamic or quasi-steady-state models (covering most protection schemes e.g., [4]) may still be computationally intense, if the size of the network and number of scenarios increases. In other words, considering computational time and applying quasi models for decision making in stochastic planning against possible initiating events pose challenges.

Several researchers have explored machine learning algorithms for different applications, such as security-constrained optimal power flow [6], sparking interest in LS prediction for various scenarios, including cascading failures. For instance, in [7], a linear regression model predicted LS due to cascading failures, albeit limited to cascading outages triggered by line overloading. Similarly, [8] employed a graph convolution neural network for LS prediction, but the dataset only considered $N-1$ contingency failures, potentially insufficient for analyzing cascading failures triggered by stochastic events (e.g., weather events) where many assets may fail in a short period of time. Further, [9] utilized neural networks for adaptive LS focused on under-frequency protection but lacked consideration on cascading failure propagation. These studies reflect a narrow focus on LS prediction with limited consideration of cascading failures and a wide range of protection mechanisms.

Bridging these gaps in existing works, this paper proposes a stacking ensemble regression model by integrating different machine learning regressors (similar to the meta-classifier in [10]) with a cascading failure model for predicting cascading-driven nodal LS. The dataset for training the machine learning model includes a wide range of contingencies (i.e., $N-k$ with varied values for k) that address both less impactful and HILP events. In other words, for less impactful events, we create contingencies for $N-1$ and $N-2$ deterministically. For high-impact low-probability (HILP) events, we create contingencies for $N-3$ in the same deterministic way. Additionally, we use a weather event simulator [11] to generate a large set of stochastic contingencies. This approach represents more realistic scenarios with higher k values. Further, the AC cascading failure model (AC-CFM) from [4] is applied to quantify the nodal LS for all the generated contingencies, considering a wide range of protection mechanisms like under frequency load shedding (UFLS), over frequency generator shedding (OFGS), voltage collapse load shedding (VCLS), over (OXL) and under excitation limiters (UXL). In essence, the generated dataset for training the ML models contains all the contingencies and their corresponding cascading-driven nodal LS.

To demonstrate the effectiveness of the stacking ensemble regression model, a comparative analysis based on the R^2 metric is conducted with individual regression models on the IEEE 39-bus system. Additionally, the proposed model is tested on the IEEE 24-bus Reliability Test System (RTS) to investigate the effect of network reliability on cascading failures and the fitness of both the proposed and individual regression models.

This paper is structured as follows: Section II describes the proposed framework including dataset generation and the stacked ensemble regression model. Section III presents the case study, discusses the results, and analyzes their implications. Finally, Section IV concludes the paper with key remarks.

II. Machine Learning-Driven Prediction of Cascading-Induced Load Shedding

The proposed methodology for machine learning-based cascading-driven nodal LS prediction is depicted in Fig. 1. Initially, the dataset for training the proposed model is developed, which includes a wide range of contingencies representing both common and HILP events, along with their corresponding nodal LS due to cascading failure propagation, evaluated using AC-CFM. This dataset is then used to train the stacking ensemble regressor model, as shown in Fig. 1. One of the main novelties of this paper is that the models are using solely the line status as input to predict the resulted LS without any dependency on measurements. For comparative analysis, the same dataset is used to train individual regression models, and the comparison is conducted using the R^2 metric. The details of dataset generation and the stacked ensemble regressor model are elaborated in the subsequent subsections.

A. Dataset Generation

1) Generation of Contingency Scenarios

The generated contingencies consists of two sets that are designed to represent less impactful and HILP event scenarios. These scenarios are generated as follows:

- For representing less impactful events: $N-1$ and $N-2$ contingency scenarios are generated deterministically.
- For representing HILP events: (1) $N-3$ contingency scenarios are generated deterministically and (2) a large set of stochastic contingency scenarios are generated using a wind event simulator [11] to account for realistic contingencies with higher values of k contingencies.

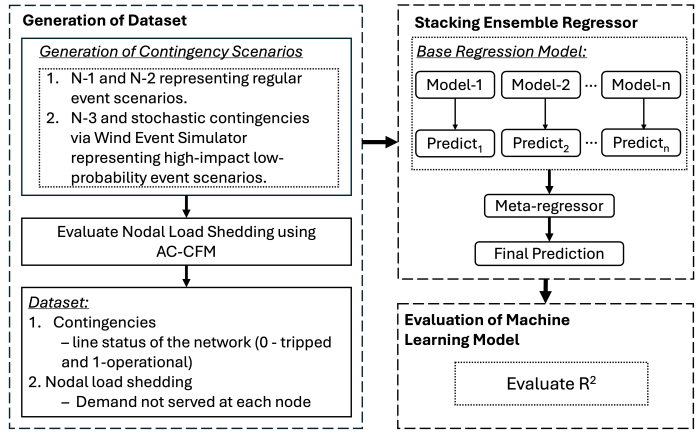

Fig. 1: Proposed Methodology

The reason for using a stochastic event simulator is to mitigate the dramatic increase in the number of contingency scenarios that would result from deterministic generation (specifically beyond $N-3$) and to simulate realistic scenarios tailored to the geographical region of the network. As mentioned earlier, this paper implements a wind event simulator to generate stochastic contingencies within the proposed framework. It employs a fragility-based approach to assess the impact of windstorms on transmission lines. Fragility curves represent the probability of failure based on hazard intensity. Using these curves, the simulator models the failure probabilities of transmission lines. The simulator generates realistic wind events in a spatio-temporal manner using historical data parameters such as wind gust speed, radius, and direction.

Finally, these two sets of contingencies representing less impactful and HILP event scenarios are augmented to form the overall contingency scenarios, representing most realistic scenarios covering both less impactful and HILP or extreme events. Further, all these scenarios are applied as input to the AC-CFM to evaluate the cascading-driven nodal LS. To mention, all these contingency scenarios i.e., operational status of the network lines (either tripped or online) represent the initiating events for cascading failure analysis.

2) Evaluating Nodal Load Shedding using AC-CFM

Cascading failure models simulating cascading events are crucial for enhancing the resilience of power networks as they quantify cascading failure propagation. The AC-CFM proposed in [4] is applied in this study. This quasi-steady-state cascading analysis model incorporates the most common protection mechanisms, i.e., UFLS, OFGS, UVLS, OXL and UXL. The reason for selecting this model is its computational performance compared to fully dynamic models, particularly when applied to stochastic analysis. Furthermore, it has been validated following the approaches outlined by the IEEE PES Working Group on Cascading Failures [12].

In general, a cascading failure is triggered by the sequential activation of protection mechanisms, potentially leading to the network splitting into islands. These failures propagate within the islands, perpetuating the cascade. The AC-CFM uses the generated initiating contingency scenarios from the previous step of the methodology. Importantly and differently from other cascading analysis models, it employs a recursive approach to manage cascading within each island until the cascade halts. For a given initiating contingency scenario, AC-CFM identifies the number of active and passive islands, computes power flow within each island, and applies protection mechanisms. If the protection mechanism alters anything within the island, recursion is applied again. Otherwise, the cascade is concluded within that island, and the model moves on to the next island. This recursive procedure continues until all identified islands in the network are analyzed, and corresponding LS values are evaluated. As mentioned earlier, each of the contingencies generated in the previous subsection is passed to AC-CFM for calculating the nodal LS. Hence the final dataset utilized in the proposed machine learning model will contain the contingencies and their corresponding cascading impact in terms of nodal LS.

B. Stacked Ensemble Regressor Model

This paper leverages a stacked ensemble methodology to enhance the predictive performance of the machine learning model [10]. In the proposed methodology, the selection of machine learning models, specifically regression models, are chosen as base models used to train a portion of the dataset generated (detailed in the previous subsection). To mention the regression models are generally applied for prediction of continuous values in machine learning applications. Each base regression model is trained on the same dataset, and a distinct set of data is used to obtain predictions from each base model. Subsequently, the predictions generated by these models are employed as inputs to train a meta-regressor. Finally, a subset of data that has not been previously applied within the dataset is utilized to produce the final prediction.

In this study, after investigating the fitness of different regression models within the meta-model, Random Forest is chosen as meta-regressor. This meta-regressor is integrated with the predictions generated by the base regression models, including Random Forest [13], XGBoost [14], and Linear Regression. The selection of these regression models to frame stacking ensemble model is based on their diverse approaches to regression modeling: XGBoost and Random Forest are ensemble methods known for their strong predictive performance and ability to capture complex relationships in the data, while Linear Regression offers simplicity and interpretability [15]. By stacking these models, the study aims to leverage the complementary strengths of each approach to enhance overall prediction accuracy and robustness.

Furthermore, through a nested cross-validation procedure (i.e., k-fold), the base models are trained on the training data, and their predictions on validation folds are employed as features for training the meta-model. In this technique, the original dataset is first split into training and testing sets to avoid data leakage. The training set consists of 80% of the dataset and is partitioned into k folds. The base regression models are trained on $k-1$ folds and evaluated on the k^{th} fold. This procedure is repeated k times, with each partition serving as the validation set once. The model's performance is averaged across all validation folds. Notably, the predictions reported later are based on a separate testing set, untouched during training. This nested cross-validation approach ensures proper evaluation and validation, yielding robust performance estimates and preventing overfitting to the training data.

III. Case Study

In this study, the proposed methodology is initially applied to the standard IEEE 39-bus system which is not $N-1$ secure to assess its effectiveness. Furthermore, to analyze the model's dependency on degree of system reliability (e.g., $N-1/N-2$ reliable), the proposed network is tested on a reliability test system, namely the IEEE 24-bus RTS system. In other words, the effectiveness of the proposed method is evaluated on $N-1$ reliable system (i.e., the IEEE 24-bus RTS system), where there is no significant cascading impact for a single contingency. This facilitates a comparison of nodal R^2 values obtained in the IEEE 39-bus and IEEE 24-bus RTS systems.

The IEEE 39-bus system considered for this study comprises 39 nodes and 46 lines with 10 generators spread across the network. For this network, a total of 17,261 contingency scenarios are generated, including $N-1$ contingency with 46 scenarios, $N-2$ contingency with 1035 scenarios, $N-3$ contingency with 15180 scenarios, and stochastic contingency generated from the wind event simulator with 1000 scenarios. Since the event simulator models the storm trajectory to derive the number of lines that fail, this can result in the failure of one, two, three, or more lines depending on the storm's trajectory. This variability is illustrated in Fig. 2, where the x-axis represents the degree of contingencies (i.e., the value of k indicating the number of lines failed in each scenario) and the y-axis represents the corresponding count over all the stochastic scenarios generated by the event simulator. However, it's important to note that some stochastic events are already accounted for in the deterministic approach of $N-1$ to $N-3$ contingencies, leading to some duplicate scenarios. Consequently, these duplicate scenarios generated from the stochastic wind event simulator are removed, leading to 244 stochastic scenarios with more than three line contingencies. Hence, the total number of augmented contingency scenarios utilized for machine learning models is 16,505.

Similarly, for the IEEE 24-bus RTS consisting of 24 nodes, 38 lines, and 33 generators, initially 10,177 contingency scenarios were generated. These scenarios are resulted from $N-1$ with 38 scenarios, $N-2$ with 703 scenarios, $N-3$ with 8,436 scenarios, and stochastic contingencies from the wind event simulator with 1,000 scenarios. The stochastic scenarios generated by the event simulator will include the $N-1$, $N-2$, and $N-3$ contingency scenarios as depicted in Fig. 2 (similar to IEEE 39-bus system). Therefore, duplicate scenarios in the stochastic scenario set were removed, resulting in 250 stochastic scenarios. Therefore, the total number of augmented scenarios obtained for utilization in machine learning models is 9,427. It is important to note that the number of augmented contingency scenarios for both networks cannot be the same,

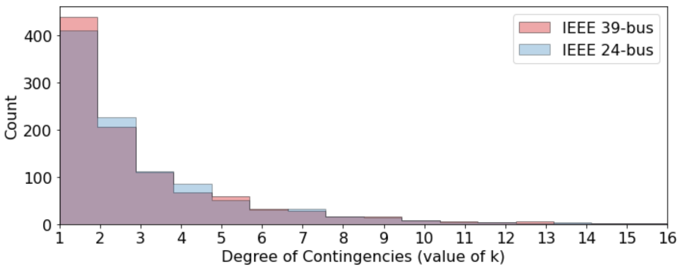

Fig. 2: Stochastic Contingencies generated using Wind Event Simulator.

as the deterministic contingencies depend on the number of lines in the network, which is significantly different between these two networks. Since a larger portion of the dataset is generated deterministically, Principal Component Analysis (PCA) was conducted to check for any potential biases. The PCA revealed two principal components that contribute almost equally to the variance, suggesting no dominant structure in the data. The independence of the features was further confirmed by the low correlation values observed in the correlation matrix, indicating that the data is not influenced by dominant features or multicollinearity.

All simulations were conducted on a workstation with an AMD Ryzen 5800X processor and 32 GB RAM. The results first focus on the IEEE 39-bus system, comparing individual regression models with the stacked ensemble regressor using R^2. This is followed by the results for the IEEE 24-bus RTS. A comparison of both networks concludes the analysis, highlighting the impact of network reliability on nodal LS prediction during cascading failures.

A. Simulation Results on the IEEE 39-bus system

Figure 3 presents the R^2 values are obtained from both the stacked ensemble model and the individual regression models. Here, the x-axis of the plot represents the load buses, while the y-axis represents their corresponding R^2 values. Observing this plot reveals that the Random Forest and Linear Regression models exhibit relatively lower performance compared to XG-Boost, with overall R^2 values of 0.44 and 0.40, respectively while XGBoost achieves an average R^2 of 0.70. However, it is clearly evident that the stacked ensemble regression model surpasses all the individual machine learning models for all nodes, boasting an average R^2 value of 0.88. This clearly shows that the stacked ensemble regression model predicts the cascading-driven nodal LS with higher accuracy compared to individual regression models.

B. Simulation Results on the IEEE 24-bus RTS

In this section, initially, the results obtained on the IEEE 24-bus RTS from both the stacked ensemble model and individual machine learning regression models are presented. Subsequently, a comparison of the overall R^2 is performed between the IEEE 39-bus system and IEEE 24-bus RTS. The dataset, including contingencies corresponding to the IEEE 24-bus RTS and its corresponding cascading-driven nodal LS, is used to train both the stacked ensemble regression model

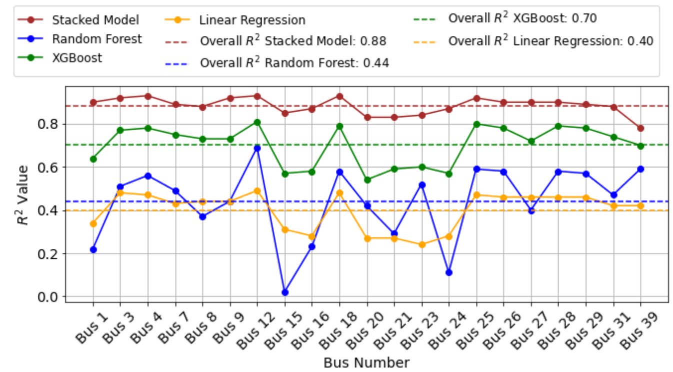

Fig. 3: Nodal R^2 of IEEE 39-bus system: Stacked Model vs. Individual Machine Learning Models.

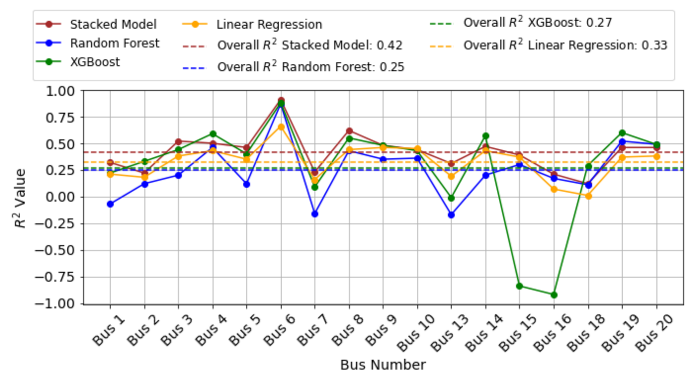

Fig. 4: Nodal R^2 of IEEE 24-bus RTS: Stacked Model vs. Individual Machine Learning Models.

and individual regression models. Similar to the IEEE 39-bus system, R^2 values corresponding to each load bus and the overall value are obtained and presented in Fig. 4. From this plot, it is clear that among the individual regression models, the Linear regression model performs relatively better compared to all other individual models, with an average R^2 value of 0.33. Furthermore, it can be seen that some specific nodal R^2 values obtained from the XGBoost and Random Forest are negative. For instance, at "Bus15", the R^2 from the XGBoost model is -0.84, and at "Bus13", the R^2 from the Random Forest regression model is -0.17. This shows that these regression models are unable to capture the trend of the data, and they perform worse in prediction than simply use the mean of the data. However, the stacked model performs better compared to all individual models with an average R^2 value of 0.42 and all positive R^2 nodal values.

C. Discussion

Comparing the R^2 values mentioned earlier from stacked ensemble and individual models for both the IEEE 39-bus and IEEE 24-bus RTS, it is evident that the stacked model outperforms individual machine learning regression models in both the IEEE 39-bus system and IEEE 24-bus RTS. However, the stacked ensemble regressor is more accurate on IEEE 39-bus system compared to IEEE 24-bus RTS. In other words, based on the R^2 values, most of the regression

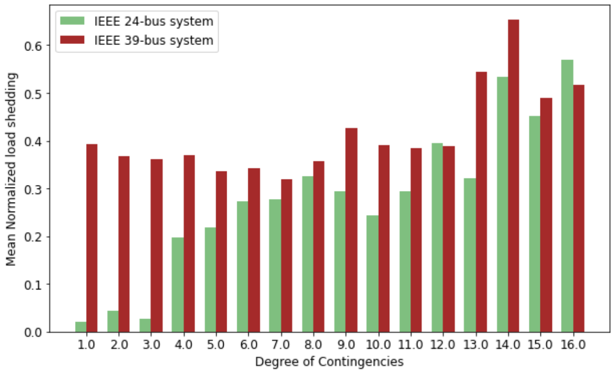

Fig. 5: Comparison of Mean Normalized LS between IEEE 39-bus and IEEE 24-bus RTS.

models analyzed are not efficient in predicting nodal LS for the IEEE 24-bus RTS. To understand the reason for this condition, further investigation is conducted based on the number of contingencies within each scenario (i.e., the total number of lines tripped for a given scenario) and its corresponding mean LS value. Figure 5 shows the comparison of mean normalized LS values corresponding to each contingency (up to $N-16$) between the IEEE 39-bus and IEEE 24-bus RTS. Note that for the calculation of mean LS, zero LS scenarios are excluded as this could bias the results. This figure clearly indicates that the IEEE 24-bus RTS is highly reliable and can withstand up to $N-3$ with a significantly lower value of mean normalized LS. For instance, only 15% of $N-1$, 23% of $N-2$, and 27% of $N-3$ contingency scenarios have LS greater than zero. In other words, out of 9,177 deterministic scenarios, 6,669 scenarios have zero LS, leading to a biased dataset, causing the machine learning models to be unable to effectively learn the underlying patterns and make accurate predictions. From this, it is evident that the synthetic dataset generation process applied for not $N-1$ reliable networks may not be suitable for highly reliable networks. The time needed per prediction on the IEEE 39-bus is 23.30 μs while for the IEEE 24-bus RTS is 19.15 μs.

IV. Conclusions

In this study, the effectiveness of a stacked ensemble regression model train solely on line status in predicting cascading-driven nodal LS was investigated, focusing on both deterministic N-1, N-2 and N-3 contingency scenarios, and stochastic HILP event scenarios with a large number of initiating contingencies. The cascading-driven nodal LS from these initiating set of contingencies was evaluated using AC-CFM. The performance of the stacked ensemble model and the individual models was assessed on both the IEEE 39-bus system and the IEEE 24-bus RTS. Analysis of the R^2 values revealed that while the stacked ensemble model outperformed individual regression models for both networks, indicating its efficacy in predicting LS, both the stacked ensemble model and individual models exhibited lower accuracy for the IEEE 24-bus RTS. The accuracy discrepancy in the IEEE 24-bus RTS likely stems from fewer high LS instances in the dataset, impacting the stacked model's predictive accuracy. This study indicates that a datasets with low-degree (i.e., lower k value) contingencies may not suit all networks. Reliable networks may need more higher-degree contingencies to avoid lack of information. Future research will focus on predicting both LS and the lines tripped due to cascading failures. It is worth noting that while generator tripping was not directly included as a feature, the AC-CFM model used in this study accounts for it indirectly through OFGT and UFGT when calculating load shedding. Dynamic studies and the evaluation of the model under different loading levels will be part of future work.

References

[1] V. S. Rajkumar, A. Ştefanov, A. Presekal, P. Palensky, and J. L. R. Torres, "Cyber attacks on power grids: Causes and propagation of cascading failures," *IEEE Access*, vol. 11, pp. 103 154–103 176, 2023.

[2] H. Guo, C. Zheng, H. H.-C. Iu, and T. Fernando, "A critical review of cascading failure analysis and modeling of power system," *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 9–22, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032117308432

[3] M. Zakariya and J. Teh, "A systematic review on cascading failures models in renewable power systems with dynamics perspective and protections modeling," *Electric Power Systems Research*, vol. 214, p. 108928, 2023.

[4] M. Noebels, R. Preece, and M. Panteli, "Ac cascading failure model for resilience analysis in power networks," *IEEE Systems Journal*, vol. 16, no. 1, pp. 374–385, 2022.

[5] A. Umunnakwe, H. Huang, K. Oikonomou, and K. Davis, "Quantitative analysis of power systems resilience: Standardization, categorizations, and challenges," *Renewable and Sustainable Energy Reviews*, vol. 149, p. 111252, 2021.

[6] J. Rahman, C. Feng, and J. Zhang, "Machine learning-aided security constrained optimal power flow," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[7] R. A. Shuvro, P. Das, M. M. Hayat, and M. Talukder, "Predicting cascading failures in power grids using machine learning algorithms," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[8] C. Kim, K. Kim, P. Balaprakash, and M. Anitescu, "Graph convolutional neural networks for optimal load shedding under line contingency," in *2019 IEEE Power Energy Society General Meeting (PESGM)*, 2019, pp. 1–5.

[9] R. Hooshmand and M. Moazzami, "Optimal design of adaptive under frequency load shedding using artificial neural networks in isolated power system," *International Journal of Electrical Power Energy Systems*, vol. 42, no. 1, pp. 220–228, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142061512001226

[10] W. Jiang, Z. Chen, Y. Xiang, D. Shao, L. Ma, and J. Zhang, "Ssem: A novel self-adaptive stacking ensemble model for classification," *IEEE Access*, vol. 7, pp. 120 337–120 349, 2019.

[11] B. V. Venkatasubramanian, M. Lotfi, P. Mancarella, A. Águas, M. Javadi, L. Carvalho, C. Gouveia, and M. Panteli, "Machine learning based identification and mitigation of vulnerabilities in distribution systems against natural hazards," in *27th International Conference on Electricity Distribution (CIRED 2023)*, vol. 2023, 2023, pp. 2908–2912.

[12] J. Bialek, E. Ciapessoni, D. Cirio, E. Cotilla-Sanchez, C. Dent, I. Dobson, P. Henneaux, P. Hines, J. Jardim, S. Miller, M. Panteli, M. Papic, A. Pitto, J. Quiros-Tortos, and D. Wu, "Benchmarking and validation of cascading failure analysis tools," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4887–4900, 2016.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[14] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[15] Y. Song, P. N. Suganthan, W. Pedrycz, J. Ou, Y. He, Y. Chen, and Y. Wu, "Ensemble reinforcement learning: A survey," *Applied Soft Computing*, p. 110975, 2023.