Identifying Critical Components in Power Networks through Machine Learning

George Paphitis[1], Balaji V. Venkatasubramanian[2], and Mathaios Panteli[1]

[1] *Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus*

[2] *School of Technology, Woxsen University, Telangana, India*
**Email: pafitis.g.georgios@ucy.ac.cy*

Abstract

As modern power systems become more complex, identifying critical components for their reliable operation is becoming increasingly important, especially in dealing with widespread cascading outages from large-scale disturbances. In this context, this paper introduces a novel method for identifying critical components in terms of their contribution to cascading-induced demand not served (DNS), utilizing feature selection techniques integrated with a machine learning classifier model. While existing methods predominantly identify critical assets using DNS, they often overlook the effects of cascading failures. Furthermore, those that do consider cascading failures typically focus solely on line overloading. This work, therefore, introduces a machine learning-based methodology coupled with a cascading failure model, considering protection mechanisms like excitation limits, under/over frequency, and line overloading. Tested on the IEEE 24-bus system, the validation method exhibits strong sensitivity (80.35%-90.03%) and specificity (80.35%-94.10%). Additionally, SHapley Additive exPlanations (SHAP) is used for validation by comparing its results with the proposed method.

1 Introduction

Natural hazards such as hurricanes, floods, earthquakes, and wildfires can cause serious damage to power systems causing prolonged and widespread outages [1]. Timely identification of critical assets can help system operators in developing appropriate mitigation strategies. Identifying these critical assets typically involves deterministic security analysis, such as $N-1$ and $N-2$, due to their computational efficiency. However, with growing concerns for power system resilience, the scope of contingencies may extend to $N-k$, where k is significantly higher. The primary challenge lies in the exponential increase in outage scenarios as k increases. For example, in the IEEE 24-bus system, which is relatively small network in terms of the number of nodes, there are 714 contingencies for $N-2$, 9,880 for $N-3$, and 101,270 for $N-4$, and so on. Consequently, accounting for all possible contingency combinations becomes exceedingly challenging.

Most vulnerability identification methods follow contingency analysis that quantifies the impact by modelling the power system using power flow models, i.e., power flow equations that can be solved by the optimal power flow (OPF) algorithm to determine the demand/energy not served (DNS or ENS) [2]. Following the calculation of the impact for each contingency scenario, various approaches, including the application of machine learning (ML) techniques, have been proposed to identify critical assets [3–6]. In [3, 4], the authors quantified DNS for a set of random contingency scenarios and used the k-means algorithm to identify critical assets in the distribution network. Similarly, in [5], the authors applied k-means to identify critical assets in the bulk electric system, while [6] employed game theory for this purpose. Meanwhile, some authors have considered the impact of cascading failures in identifying critical assets. For instance, in [7], a synchronization matrix is used to identify clusters of critical assets due to cascading failures by employing a dynamic model. However, this framework generates random contingency scenarios up to $N-5$. Similarly, in [8], the authors proposed a framework for identifying critical transmission lines by ranking the DNS and frequency deviation through the introduction of random cascading failures. Although this framework models random contingencies up to $N-7$, 99.9% of the contingencies are covered up to $N-3$. This approach can overlook the impact of high-impact, low-probability events.

Furthermore, in [9], interaction graphs were employed to identify critical assets, considering cascading failures due to line overloading. Similarly, [10] proposed a fault-chain model to identify critical lines in the network resulting from cascading failures, limited to line overloading. In [11], the authors introduced a flow-based estimator to predict critical lines in the network affected by cascading failure propagation, primarily focusing on line overloading. Additionally, in [12], the authors developed a fast identification algorithm to determine critical lines in the network. This approach also relies on heat accumulation, i.e., line overloading, for cascading failure propagation.

Although several methodologies in the literature quantify cascading failures, only a few have explored the application of ML algorithms to assess cascading impacts comprehensively. Moreover, those studies that do address this issue predominantly focus on line overloading. Therefore, this paper proposes a framework that utilizes cascading failure propagation data generated by the AC cascading failure model (AC-CFM) from [13] to accurately identify critical components. This

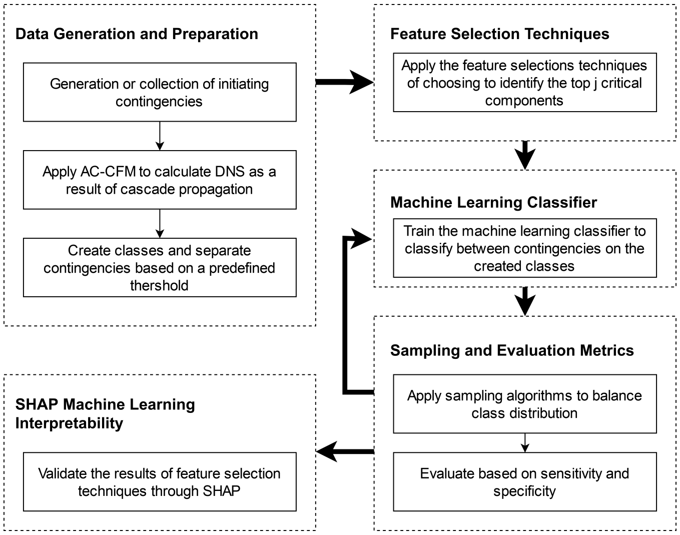
Data Generation and Preparation
Generation or collection of initiating contingencies
Apply AC-CFM to calculate DNS as a result of cascade propagation
Create classes and separate contingencies based on a predefined thershold
Feature Selection Techniques
Apply the feature selections techniques of choosing to identify the top j critical components
Machine Learning Classifier
Train the machine learning classifier to classify between contingencies on the created classes
Sampling and Evaluation Metrics
Apply sampling algorithms to balance class distribution
Evaluate based on sensitivity and specificity
SHAP Machine Learning Interpretability
Validate the results of feature selection techniques through SHAP

Figure 1: Flowchart of Critical Component Identification in Cascading Power Outages.

framework includes common protection mechanisms such as under/over frequency, excitation limits, and line overloading. Subsequently, ML algorithms are employed to identify critical assets, with a specific focus on network lines in this study. Furthermore, a set of commonly used ML classifiers is employed to demonstrate its effectiveness in identification of critical assets using SHapley Additive exPlanations (SHAP).

The paper's structure is as follows: Section II outlines the proposed methodology, including the feature selection strategies, introduction of SHAP, and the specific machine learning model used. Further, it also discusses on the techniques that are used to address the imbalanced data distribution within the dataset. Section III presents and analyzes the results, while Section IV concludes the paper.

2 Identifying Critical Components in Cascading Power Outages

The proposed methodology is illustrated in Fig. 1. Initially, data generation and preparation involve creating a large set of random contingencies and evaluating cascading-driven DNS using AC-CFM. Feature selection techniques are then applied to identify the most critical components, focusing on critical lines. Subsequently, a machine learning classifier is used to validate the results, with class imbalance addressed through sampling algorithms. Finally, the proposed methodology is validated using SHAP, which demonstrates the contribution of features to the prediction and aligns with established feature selection techniques.

2.1 Dataset Generation and Preparation

The generation of the dataset is one of the crucial steps in the proposed methodology. The network selected for analysis is exposed to a large set of n random contingencies, with the number of failed components ranging from k_{min} to k_{max}. The DNS resulting from cascading failure propagation for all these contingencies is evaluated using AC-CFM. Furthermore, the contingencies are separated based on a threshold value of cascading-induced DNS into small and big disturbances. This threshold can be defined by the end-user of the model. In this study, two datasets are generated to cover a wide range of contingencies. The first dataset contains a large set of n random contingencies, with offline components ranging from k_{min} to k_{max}. In other words the random contingencies are generated with $N-k$ scenarios with k varying from k_{min} to k_{max}. During this process, there is a possibility of certain network components being under-represented, leading to a potential bias in the dataset. For instance, if a network component A appears 10 times while component B appears 100 times, this disparity could introduce significant bias into the dataset. Therefore, a second dataset is constructed to provide a more comprehensive set of contingencies. In this dataset, 1000 contingencies are generated for each of the m components in the network, along with random contingencies with offline components ranging from $k_{min} - 1$ to $k_{max} - 1$. This results in a total number of offline components ranging from k_{min} to k_{max}. This approach ensures that each feature is represented equally on the dataset and it will result on a dataset with $1000m$ contingencies.

Once the contingencies are generated, the two datasets are prepared for further steps in the methodology. Initially the dataset is cleaned by removing the duplicates and later it is classified into two different classes based on the cascading impact, i.e., resulting cascading-driven DNS and a predefined threshold. For this, these contingency datasets are initially evaluated using an AC-CFM model to quantify the cascading-driven DNS. These classes will exhibit an approximate $1 : h$ ratio where $h > 1$. This ratio implies that for every sample in the minority class (big disturbance), there are h samples in the majority class (small disturbance). It is possible that the minority class could be the small disturbances; in this scenario, the opposite would apply. This poses an imbalanced classification problem and should be treated. More details on handling this type of skewed dataset is given in Section II.D.

After preparing the dataset, the next step is to apply the feature selection techniques, detailed in the following subsection.

2.2 Feature Selection Techniques

There are many feature selection techniques available in the literature. In this study, five of them are chosen to identify the j most critical features out of m features (network lines/branches) in total. These are Lasso Regression (LAR), Logistic Regression (LR), Logistic Regression with low p-values (LRP), Random Forest Importance (RF), and eXtreme Gradient Boosting Importance (XGBoost / XGB). In this, three are statistic-based feature selection techniques (LAR, LR, LRP) and two are tree-based techniques (RF, XGB), selected for comparison purposes.

These five techniques operates on distinct principles. For example, LAR introduces L1 regularization, which can shrink certain feature coefficients to zero, effectively omitting them from the model. LR, a binary classifier, can be adapted for feature selection by examining the coefficients' weights; larger absolute values indicate higher significance. Enhancing this method, using LRP with low p-values emphasizes selecting features whose coefficients are statistically significant, in our case the features with the lowest p-value are selected. RF Importance offers a different lens by utilizing an ensemble of decision trees; features are ranked based on how frequently they reduce impurity across trees. Lastly, XGBoost / XGB importance operates similarly but in the context of the gradient-boosted framework, ranking features by their contribution to the model.

It's important to note that the critically identified features from the feature selection techniques are not necessarily critical lines in means of cascade propagation but in means of DNS due to cascading failures. In other words, the goal is not to identify lines that are causing more components to fail, but rather to find the ones that are causing the biggest DNS. Following the identification of the j critical features, the ML classifier is trained, as detailed in the subsequent subsection.

2.3 Machine Learning Classifier

After evaluating various machine learning algorithms based on their sensitivity performance, XGBoost is selected to be used due to its superior results. XGBoost operates as an ensemble technique, integrating numerous decision trees (referred to as weak learners) by leveraging the boosting approach to produce the final predictive outcome [14]. This approach allows XGBoost to sequentially learn from the previous trees' mistakes, thereby improving accuracy with each iteration. It's a dynamic and robust approach that effectively handles various types of predictive modeling problems. In this study, XGBoost is trained on 80% of the data and evaluated on 20% of the data using a stratify split to maintain the class distribution of the original dataset. Crucial point to any ML model is the evaluation phase. In this case, the chosen evaluation metrics are sensitivity and specificity. To enhance the model's performance, it is essential to employ techniques that address the issue of imbalanced class distribution. Following the training of the machine learning model on the original dataset, which serves as the baseline model, the model is evaluated and the problem of class imbalance is addressed, as detailed in the subsequent subsection.

2.4 Addressing Class Imbalance through Sampling and Evaluation metrics

As aforementioned, an imbalanced classification problem with a ratio of 1:h is being dealt with. Ideally, a balanced class distribution in the dataset is preferred when a machine learning techniques is applied. To address the class imbalance, oversampling and undersampling techniques are used and the model performance is evaluated.

Sampling methods are applied only during the model's training phase to address class imbalance, not to the evaluation dataset. To prevent data leakage (information leakage from the training set to the test set), the dataset should first be split into training and testing sets. Sampling is then performed only on the training data, ensuring the model evaluates on real and representative examples without exposure to test set information, thus maintaining evaluation integrity.

The algorithm used for oversampling in this study is BorderlineSMOTE [15]. BorderlineSMOTE, an extension of Synthetic Minority Over-sampling Technique (SMOTE), generates synthetic examples near the minority class's decision boundary to enhance classification. Tomek Links [16] is used for undersampling, removing overlapping instances near the boundary to improve class separation for the classifier.

First, the machine learning model is trained and evaluated on the original dataset. Subsequently, sampling techniques were applied and new models were trained. Specifically, four models were developed for each dataset: 1) trained on the original dataset without sampling, 2) with oversampling applied, 3) with both oversampling and undersampling applied, and 4) with tuned hyperparameters and an increased classification threshold, also incorporating both oversampling and undersampling. Increasing the classification threshold of a model involves raising the decision boundary score required for an instance to be classified as a positive.

The most common metric used in evaluating ML classifiers is accuracy. However, when dealing with imbalanced datasets, evaluation of a ML classifier is not straightforward, and accuracy might not be an appropriate performance indicator. Accuracy, calculated as $\frac{TP+TN}{TP+TN+FP+FN}$ (where TP is the true positives, TN the true negatives, FP the false positives and FN the false negatives) can be misleading in the context of imbalanced datasets, which may lead to overoptimistic results; this phenomenon is known as the accuracy paradox. Sensitivity or true positive rate is defined as the number of true positives divided by the total number of positive cases. It is calculated as $\frac{TP}{TP+FN}$. Sensitivity is crucial when the consequences of missing a positive instance are high. Specificity or true negative rate is defined as the number of true negatives divided by the total number of negative cases. It is calculated as $\frac{TN}{TN+FP}$. Specificity is important when false positive classifications are significant. While sensitivity and specificity are informative, there is a trade-off between the two. Subsequently, to evaluate the ability of feature selection techniques in addressing this matter, SHAP is used.

2.5 SHAP for Machine Learning Interpretability

When it comes to critical infrastructures, ML interpretability is of utmost importance to ensure that the predictions of the model are based on logical assumptions and to observe any possible biases. Operators who depend on ML models to make decisions should be able to trust and be confident in the predictions of such models. Risks associated with black box models in high-stakes scenarios, as well as real-world examples, are provided in [17, 18].

To validate the results from the feature selection techniques, and to gain insights into the proposed model, SHAP is used. Specifically, *TreeExplainer* is employed, which is optimized for decision tree models, to compute Shapley values. The primary objective of SHAP is to explain how each feature of an instance x contributes to its prediction. In order to do so, SHAP employs Shapley values, a concept derived from coalitional game theory [19]. These SHAP values offer a unified measure of feature importance by fairly distributing the contribution of each feature to every individual prediction [20]. More specifically, SHAP values are expressing the influence wielded by each individual feature in the model over its predictions.

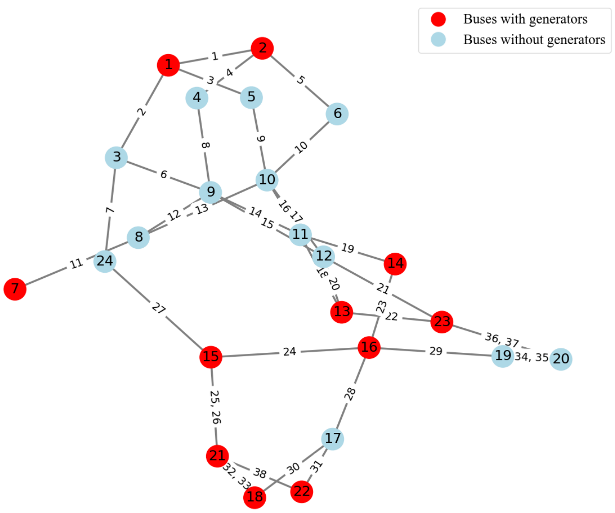

Figure 2: Graph illustration of IEEE RTS 24 Bus System

3 Results

3.1 Case study application

The selected network is the IEEE 24-bus reliability test system which consists of 38 transmission network assets (33 transmission lines and 5 transformers). The peak load of this network is 2850 MW. The first dataset (referred to hereafter as Case 1) consists of 10660 contingencies. This number occurred after randomly generating 24000 (1000 per bus) and removing duplicates. The second case dataset (referred to hereafter as Case 2) consists of 38000 contingencies which resulted due to having 38 features modeled as lines and 1000 unique contingencies per feature. The minimum number of failed components k_{min} and the maximum number of failed components k_{max} are selected to be 2 and 7 respectively in each contingency set. In this work, the cascading driven DNS threshold chosen for ML classification is considered to be 20% of the peak demand. The reason for selecting this value is that above this threshold, there are observed contingencies of 5 or more in both datasets, which is on the higher side. Consequently, the 20% threshold results with a ratio of 1:2 between minority and majority classes for both the cases. In Fig. 2 a visual representation of the network as a graph is shown, where nodes and edges represent the busses and lines respectively.

The model was trained on a workstation with Ryzen 7 5800x CPU, Nvidia RTX 3080 GPU and 32GB of DDR4 RAM.

3.2 Simulation Results

In this section, the results of the feature selection techniques and the machine learning classifier performance are demonstrated. In Fig. 3 and Fig. 5, the x-axis represents the features or network components and the y-axis represents how many algorithms selects a particular feature, i.e., how many feature selection techniques out of the five considered choose a particular network component or feature as critical. Figure 3 shows that most of the five feature selection techniques selected for this work choose the same features as the critical ones. Further, all methods identify the same 13 out of 16 critical features. Specifically, LR, LRP and LAR identify the same 15 features as the most important ones. XGB and RF agree on 14 features with the previously mentioned methods, but individually RF and XGB identifies 15 features with RF leaving Line 18 and XGB leaving Line 19 among the 16 critical features.

To investigate the differences between the feature selection methods, the network's topology is examined. Line 18 as shown in Fig. 2, connects buses 11 and 13, where on bus 13 there are 3 generators connected with a total capacity of 285.3 MW, equal to approximately 10% of the peak load. On the other hand, Line 19 connects buses 11 and 14, where on bus 14 there is a synchronous condenser. Line 30 connects buses 17 and 18, where on bus 18 a generator of 400 MW is connected. It is observed that the models tend to choose lines that carry more load and therefore their failure may result in bigger DNS events. Therefore, it is evident that all the methods had valid reasons to choose these 3 lines even if they do not agree.

The ML classifier performs slightly better when the feature set identified by LR, LRP and LAR is used for Case 1. In Table 1, the performance of the XGBoost classifier is shown for the two cases, considering all features and the features selected by specific algorithms.

In Table 1, OV denotes oversampling, UN signifies undersampling, and TT refers to both an increased classification threshold and tuned hyperparameters. The highest sensitivity is achieved by XGB OV UN TT at 90.95%, utilizing a combination of oversampling, undersampling, an increased classification threshold, and hyperparameter tuning. Conversely, XGB achieves the highest specificity of 94.10%, operating under default model settings without sampling. Notably, oversampling alone (XGB OV) results in a sensitivity increment of 2.88%, albeit with a specificity reduction to 92.35%. When both oversampling and undersampling are applied (XGB OV UN), there is a marginal increase in sensitivity of 0.89%, with a less pronounced decrease in specificity to 92.77%. The model also shows efficiency, requiring only 1.17 μs per prediction.

From the results, the trade-off between sensitivity and specificity and the drop in performance when a subset of features is used can be observed. Specifically, sensitivity is reduced by 1% on average while specificity is reduced by 3% on average across all cases. The performance degradation is caused by loss

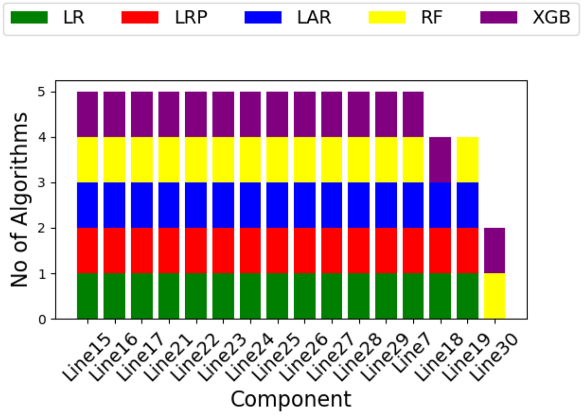

Figure 3: Frequency of selected features for Case 1.

Table 1: XGBoost performance

Metric	XGB	XGB OV	XGB OV UN	XGB OV UN TT
Using all features for Case 1				
Sensitivity	83.00%	85.88%	83.89%	90.95%
Specificity	93.54%	92.35%	92.77%	91.41%
Using the features identified by LR, LRP, and LAR for Case 1				
Sensitivity	80.35%	86.10%	85.65%	87.64%
Specificity	92.94%	86.82%	90.90%	87.41%
Using all features for Case 2				
Sensitivity	86.75%	91.48%	92.55%	90.37%
Specificity	94.10%	92.44%	89.89%	92.49%
Using the features identified by XGBoost and RF for Case 2				
Sensitivity	80.66%	90.03%	83.86%	86.46%
Specificity	92.29%	80.35%	90.18%	86.35%

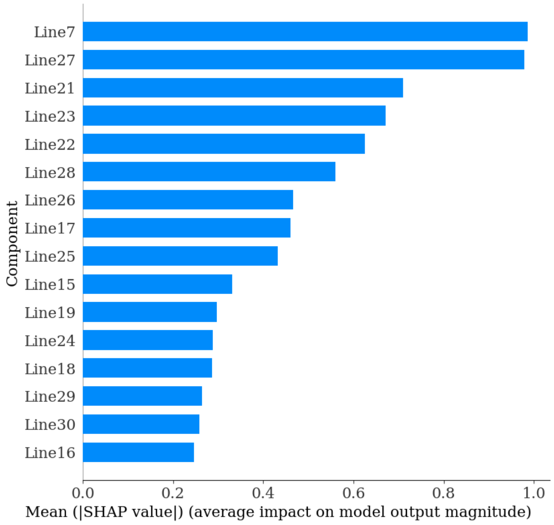

Figure 4: SHAP summary plot for Case 1.

of information; that is, 23 features were excluded. However, the performance of the model is still considered to be satisfactory based on the sensitivity and specificity scores. The top 15 features provide sufficient information for the entire dataset. The aim is to use the classifier as a validation tool rather than to create a perfect classifier. Choosing from the four options presented in the table—XGB, XGB OV, XGB OV UN, and XGB OV UN TT— ultimately depends on the end user's specific requirements. It is important to note that the goal is not to create the best-performing classifier but to find the most relevant features related to the cascades. This validation procedure aims to evaluate the consistency in model performance when using all available features compared to a selected subset. The objective is to ascertain if the identified critical features adequately represent the entire dataset.

The mean SHAP values of the features are shown in Fig. 4. While each feature selection technique selects the 15 most important features, as explained before, not all the methods agree for every feature, therefore, the total number of selected features displayed in Fig. 3 is 16. To map the results from the feature selection techniques to the features with the most contribution to the model according to SHAP, only the 16 features with the highest mean SHAP value, were chosen for visualization. Comparing the results from Fig. 3 and Fig. 4 shows that feature selection techniques and SHAP agree to all features for Case 1.

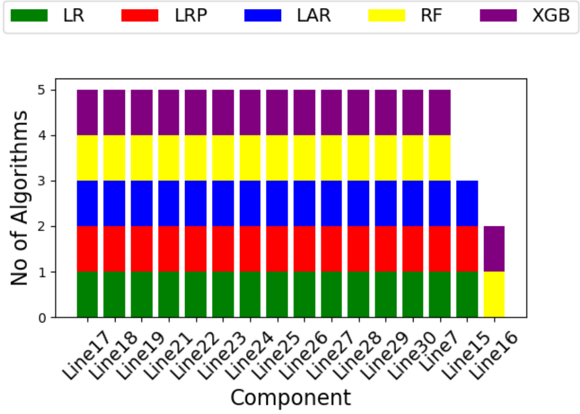

Figure 5: Frequency of selected features for Case 2.

Figure 5 reveals that as the data volume increases (Case 2), the feature selection methods converge in agreement. In Case 1, all methods agreed on 13 common features while in Case 2 all the methods chose the same 14 features as critical. For Case 2 LR, LRP and LAR methods chose the same 15 features, and the same applies for XGB and RF. The main difference is that the first three methods chose Line 15 as critical while the latter the Line 16. Line 15 connects buses 9 and 12 while Line 16 connects buses 10 and 11. Both lines connect central nodes of the network, and they are close to each other. In this Case, the ML classifier performs better when using the feature set identified by the XGB and RF. Further, in Case 2, a bigger drop in performance for both sensitivity and specificity is observed. Specifically, sensitivity is reduced by 5.04% and specificity is reduced by 4.94% on average across all cases when the model is trained on the subset of features.

The only difference between the two cases from SHAP is that SHAP ranks line 14 as more important in Case 2 than line 16 in Case 1. SHAP and feature selection techniques are in agreement in the remaining 15 features.

For further validation, the average DNS value was calculated for scenarios where a critical component fails versus when all critical components are online. The results show that when any critical component fails, the average DNS value is 14.5%, compared to 2% when all critical components are operational.

4 Conclusions

The aim of this paper was to introduce a novel approach for identifying critical components of the power network that mitigates existing limitations. The proposed methodology solely relies on feature (or network line) status and DNS. Moreover, the ML model is computationally efficient as it requires only 1.17 μs for predictions on average. The feature selection techniques work fast too, finding the critical components in time of 0.5s to 1.2s depending on the feature selection technique algorithm. As shown in the results, different datasets (Case 1 and Case 2) produce almost identical results which shows the very low variance of the model. The results demonstrates that the performance of the model is similar for both the cases. This indicates that the method can produce good results even with 3.5 times less data. A slight degradation of performance regarding the evaluation metrics was observed when the subset of features is used. However, the goal of this work is to find the most critical features, not to develop the ideal classifier. Even with the absence of 23 features the model performs adequately, meaning that the deviation from the classification case of using the full features is low. Highlighting that the smaller dataset (Case 1) produces comparable training results to the larger one (Case 2) is noteworthy. This realization can decrease the time needed for the whole methodology.

5 Acknowledgments

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). M. Panteli was partially supported by “Climate-aware Resilience for Sustainable Critical and interdependent Infrastructure Systems enhanced by emerging Digital Technologies” (ReCharged) (Grant agreement ID: 101086413).

References

[1] D. Abi Ghanem, S. Mander, and C. Gough, ““i think we need to get a better generator”: Household resilience to disruption to power supply during storm events,” *Energy Policy*, vol. 92, pp. 171–180, 2016.

[2] I. B. Sperstad, G. H. Kjølle, and O. Gjerde, “A comprehensive framework for vulnerability analysis of extraordinary events in power systems,” *Reliability Engineering & System Safety*, vol. 196, p. 106788, 2020.

[3] B. Venkatasubramanian, D. K. Saini, and M. Sharma, “Techno-economic hardening strategies to enhance distribution system resilience against earthquake,” *Reliability Engineering & System Safety*, vol. 213, p. 107682, 2021.

[4] B. 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. IET, 2023, pp. 2908–2912.

[5] C. Qin, K. P. Guddanti, B. Vyakaranam, T. Nguyen, K. Mahapatra, Q. Nguyen, Z. Hou, P. Etingov, and N. Samaan, “Critical zone identification framework for bulk electric system security assessment,” *International Journal of Electrical Power & Energy Systems*, vol. 155, p. 109542, 2024.

[6] B. Zhu, L. Zhang, and G. Li, “Identification of vulnerable transmission lines in power system based on game theory,” *IEEE Access*, 2024.

[7] B. Carreras, D. Newman, and I. Dobson, “Determining the vulnerabilities of the power transmission system,” 01 2012, pp. 2044–2053.

[8] B. Gjorgiev and G. Sansavini, “Identifying and assessing power system vulnerabilities to transmission asset outages via cascading failure analysis,” *Reliability Engineering & System Safety*, vol. 217, p. 108085, 2022.

[9] U. Nakarmi, M. Rahnamay-Naeini, and H. Khamfroush, “Critical component analysis in cascading failures for power grids using community structures in interaction graphs,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1079–1093, 2020.

[10] Y. Liu, T. Wang, and J. Guo, “Identification of vulnerable branches considering spatiotemporal characteristics of cascading failure propagation,” *Energy Reports*, vol. 8, pp. 7908–7916, 2022.

[11] B. Schäfer, D. Witthaut, M. Timme, and V. Latora, “Dynamically induced cascading failures in power grids,” *Nature communications*, vol. 9, no. 1, p. 1975, 2018.

[12] S. He, Y. Zhou, Y. Zhou, J. Wu, M. Zheng, and T. Liu, “Fast identification of vulnerable set for cascading failure analysis in power grid,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5645–5655, 2023.

[13] 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.

[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] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-smote: A new over-sampling method in imbalanced data sets learning,” vol. 3644, 09 2005, pp. 878–887.

[16] “Two modifications of cnn,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976.

[17] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. (2019) Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[18] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nat Mach Intell*, vol. 1, pp. 206–215, 2019. [Online]. Available: https://doi.org/10.1038/s42256-019-0048-x

[19] C. Molnar, *A Guide for Making Black Box Models Explainable*, 2018. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[20] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30.* Curran Associates, Inc., 2017, pp. 4765–4774.