Predicting Cascading Failures in Power Grids using Machine Learning Algorithms

Rezoan A. Shuvro, Pankaz Das, Majeed M. Hayat
Department of ECE
Marquette University
Milwaukee, WI, USA
Email: {rezoan.shuvro, pankaz.das, majeed.hayat}@marquette.edu

Mitun Talukder
Software Test Engineer
Cisco Systems Inc.
San Jose, USA
Email: mithun2k5@yahoo.com

***Abstract*—Although there has been notable progress in modeling cascading failures in power grids, few works included using machine learning algorithms. In this paper, cascading failures that lead to massive blackouts in power grids are predicted and classified into no, small, and large cascades using machine learning algorithms. Cascading-failure data is generated using a cascading failure simulator framework developed earlier. The data set includes the power grid operating parameters such as loading level, level of load shedding, the capacity of the failed lines, and the topological parameters such as edge betweenness centrality and the average shortest distance for numerous combinations of two transmission line failures as features. Then several machine learning algorithms are used to classify cascading failures. Further, linear regression is used to predict the number of failed transmission lines and the amount of load shedding during a cascade based on initial feature values. This data-driven technique can be used to generate cascading failure data set for any real-world power grids and hence, power-grid engineers can use this approach for cascade data generation and hence predicting vulnerabilities and enhancing robustness of the grid.**

***Index Terms*—Cascading failures, machine learning, MATPOWER, Monte-Carlo simulation, load shedding.**

I. Introduction

Cascading failures in power grids are described as a sequence of correlated failures of individual components that successively weakens the power system [1]. Cascading failures can occur due to a wide range of events, including natural disasters, technical error, human error, and deliberate sabotage attacks [2]. Moreover, cascading-failure dynamics involve several power-grid variables and interdependency between the variables of the power grid and the communication network [3], [4]. Modeling cascading failures and examining the severity of cascading failures is a challenging task. Nonetheless, since the 2003 blackout in North America [5], notable efforts have been given by the researchers in modeling cascading-failure dynamics in the power grid and mitigating the risk of cascading failures [6]. One general approach is the use of probabilistic modeling of cascading failures in the power grid independently [3], [7]–[9] or in an interdependent setting (including the communication network) [10]–[15]. The objective of using probabilistic models is to find the average size of the blackout, blackout size probability distribution, average load loss, load loss distribution, critical transmission lines [16].

Although most of the models are based on simulating the different power grid topologies for various failure initiating events, which generate a massive volume of cascading failure data, the use of machine-learning approaches is relatively unexplored in this area. A proactive blackout prediction model for an early warning system in smart grids is proposed in [17]. In that work, a support vector machine (SVM) has been trained using the historical cascade data and is used to predict blackout events in advance. The work uses SVM to build a prediction rule, which is used to predict the scenarios of the blackout as early as possible. However, the dataset includes only fifty cases, which are used for training and testing purpose. It is challenging to learn the complex dynamics of cascading failures in power grids using fifty test cases only. Also, the authors reported that for specific parameter values 100% training and testing accuracy was achieved, which is very unlikely and unrealistic for a sophisticated event like cascading failures. Nonetheless, the paper is a novel work on proactive cascade prediction using a machine learning approach. In [18], the authors proposed a machine learning model based on Bayes networks to predict cascading failure propagation. Their model, termed ITEPV, collects power grid data from simulations, and then predicts cascading failure propagation with the highest probability. This paper is another work that focuses on data-driven cascade prediction using a machine learning approach. However, the authors do not describe clearly how they collected the data, simulated the power flow, or what simulation software they used, which makes reproducibility of the work difficult. A classification problem is formulated in [19] that classifies a cyber-attack from other classical disturbances in the power grid. The authors used various machine learning algorithms to evaluate classification performance and found the optimal algorithms under given constraints. Furthermore, the authors observed various measures (accuracy, precision, recall, and F-Measure) to show that Adaboost+JRipper [20] is the optimal algorithm for classifying various types of cyber-threats in power grids. The work is an initial benchmark for disturbance classification in the

This work was supported by the Defense Threat Reduction Agency's basic research program under grant no. HDTRA1-13-1-0020 and National Science Foundation's CRISP Type 2 award number 1541148

978-1-7281-0407-2/19/31.00©2019*IEEE*

power grid. The authors used the WEKA [21] machine learning framework for implementing various algorithms. Benchmarking of various deep learning algorithms and comparison of %RMSE (root mean square error) reduction by the different algorithms from the existing load forecasting approaches in smart grid applications were done in [22]. To the best of the authors' knowledge, no work has been found to classify cascading failures in power grids as well as predicting key attributes such as the number of failed transmission lines, the amount of load shedding for an initial disturbance conditioned on the grid operating parameters and the topological parameters. One of the main reasons is the unavailability of a large volume of the real-world cascading failure data.

In this paper, we classify and predict cascading failure based on critical power grid attributes, namely, power-flow capacity, edge betweenness centrality, demand loss, power grid loading, estimation errors, and constraints on load shedding. The contribution of this paper is three-fold. First, we describe an earlier developed cascading failure simulation (CFS) framework using MATPOWER [23], a widely used power-flow simulator. We then use this framework to generate a cascading failures dataset using the IEEE 118-bus topology under various initial disturbance conditions. Second, a comparison using different classifiers is shown to evaluate the classification performances. The objective is to do exploratory data analysis on labeled data using various supervised machine learning algorithms and identify the best algorithm based on accuracy. Third, we use a linear regression technique to calculate the number of transmission line failures, and the amount of load shed for any given initial condition.

II. Cascading failure simulation (CFS) framework

We use MATPOWER [23], a package of MATLAB m-files for solving the steady-state DC or AC power flow optimization problem. It uses the power-flow distribution framework under the given set of constraints. The standard power flow or load flow problem involves solving for the set of voltages and flows in a power grid network corresponding to a specified pattern of load and generation [3]. MATPOWER includes solvers for both AC and DC power flow problems, both of which involve solving a set of linear equations. The AC power flow captures detailed dynamics of cascading failures in the power grid, including the transient effects. However, the effect of AC power flow on the number of transmission line failures is found to be incremental compared to the DC power flow [9]. In this paper, we generate our data set using the DC power flow because of its simplicity yet effectiveness.

A. Overview of the CFS framework

We show a flowchart in Fig. 1 that illustrates the CFS framework used in this paper. We start with two initial number of transmission-line failures in the power grid initiated from an arbitrary initial event. It is important to note that the failure of at least two transmission lines is necessary to start a cascading failure event because the power grid is robust against one transmission line failure due to N-1 security considerations. Our objective is to classify the cascading failures in power grids into no, small, and large as well as to identify the initial conditions that trigger large transmission line failures. We assume that we have sufficient knowledge regarding the power grid topology and operating parameters before the initial failure event. We then remove the failed transmission lines from the grid and check whether any islands are formed in the power grid. Note that an island is a self-sufficient local network that operates independently when disconnected from the base network having a set of generators and loads [24]. Depending on whether any islands are formed or not, we then solve the DC power flow using MATPOWER on each island. If we have any overloaded lines in the grid, we either fail these lines or probabilistically fail a set of lines among them (e.g., failing the line with the highest overload). In this paper, we use a similar approach used in [3] for islanding and overloading calculations and propose two algorithms (discussed later) for calculating islanding and overloading in power grids. We repeat the same process until we end up with a power grid with no overloaded lines, which indicates the end of a cascade.

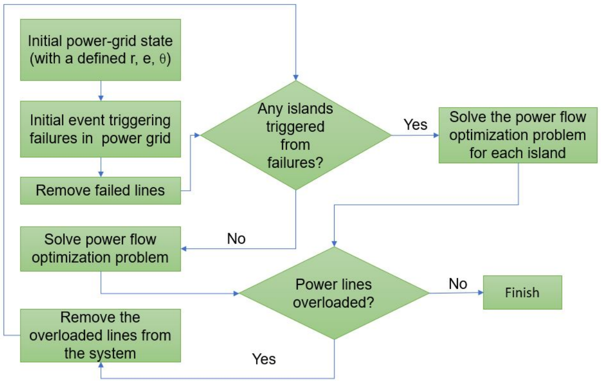

Initial power-grid state (with a defined r, e, θ)
Initial event triggering failures in power grid
Remove failed lines
Any islands triggered from failures?
Yes
Solve the power flow optimization problem for each island
No
Solve power flow optimization problem
Power lines overloaded?
No
Finish
Remove the overloaded lines from the system
Yes

Fig. 1: Flowchart of the cascading failure process in power grids

B. Power-grid operating parameters and model features

Based on power-grid simulations and prior works [3], we identify the following power-grid operating parameters and features that govern the cascading failure dynamics. In our simulation, we use the IEEE 118-bus system (which is a simple approximation of the American Electric Power system in the U.S. Midwest [25]) as the test case which contains 186 transmission lines, 118 buses (nodes) and 54 generators.

Power grid loading level, r: The power grid loading level, $r \in [0, 1]$, is the ratio of the total load demand and the generation capacity of the power grid [3]. In IEEE 118-bus system the maximum generation is 9966 MW. Here $r = 1$ indicates the demand is also 9966 MW and $r \in [0, 1]$ scales the power demand with respect to maximum possible generation. In our simulation, we define a set $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ and simulate the grid against various r from this set. We observe that for $r < 0.5$, the power grid can absorb the impact of

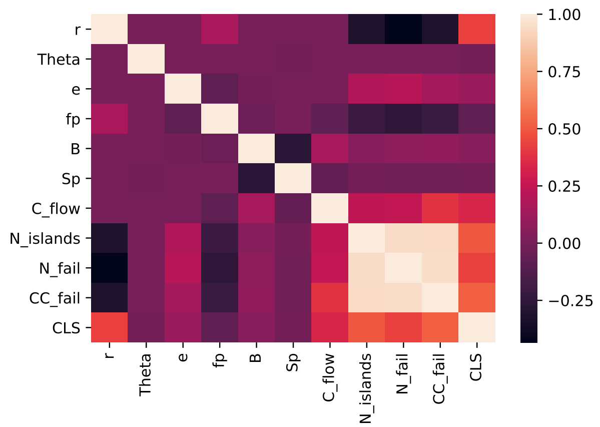


Fig. 2: Correlation among the features

two transmission line failures and redistribute the power flow without any further failures.

Load shedding constraint, θ: The load shedding constraint is the ratio of uncontrollable loads (the loads that do not participate in load shedding) and the total load in the power grid, and it is denoted by $\theta \in [0, 1]$ [3]. The parameter θ ensures the capability of implementing the control actions by the power grid operator. Namely, $\theta = 1$ indicates that all the loads are uncontrollable, and the operators can perform no load shedding. Again, $\theta = 0$ indicates that the operators can shed any load in the grid. In this paper, we consider equal load shedding constraints over all the loads in the grid for simplicity. Further, we consider a set $\{0.05, 0.1, 0.15, 0.2, 0.25\}$ and choose value of θ randomly from this set. Similar to r, a higher value of θ increases the probability of cascading failure in the power grid.

Capacity estimation error, e: Capacity estimation error, $e \in [0, 0.25]$, is the error by the control center in its estimation of the actual capacity of the lines [3]. In our CFS framework, this parameter is used to calculate overloaded lines. We used the same approach used in [3] to calculate overloaded lines. When power flow in a transmission line exceeds (1-e) capacity, we consider that line as an overloaded line. We estimate the capacity of a transmission line using power flow simulation with maximum loads, i.e. when generation equals demand (r=1). Since we use DC power flow simulation, there are no transient effects, and we can use maximum generation without any issues. We quantize the flow capacity of a transmission line into a set of five capacities $\{20, 80, 200, 500, 800\}$ MW [24] and assign this capacity of the transmission line as a constraint of the MATPOWER power flow optimization problem (discussed later). In our simulation, we consider values of e drawn from the set $\{0.05, 0.1, 0.15, 0.2, 0.25\}$

Fixed failure probability of neighboring lines, f_p: To capture the effect of hidden failures and localized failures in a power grid [6], we include a parameter f_p, the fixed failure probability of neighboring lines, in our CFS framework that fails the adjacent lines of a failed line with a small probability. We consider a set, $f_p = \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$, and choose the value of f_p randomly from that set. Since f_p is a probability, it adds uncertainty on line failures, i.e., the total number of transmission line failure after a cascading failure ends is not deterministic for a given initial condition.

Edge betweenness, B and **average shortest path**, S_p: We keep track of the average edge betweenness of the initially failed lines as a model feature which is defined as a measure of centrality based on shortest graph distance [26]. Additionally, we track the average shortest path between the two initially failed transmission lines as a model feature. The shortest path is calculated using Dijkstra's algorithm [26]. In this paper, to obtain the average shortest path between the two transmission lines, first, we calculate the distance between two starting points (from the bus) and the two ending points (to the bus) of transmission lines and then take the average distance between them. The rationale for tracking these two features is to capture the role of the physical topology of the power grid.

Flow capacity of the initially failed lines, C_{flow}: We keep track of the sum of the flow capacities of the initially failed lines. Intuitively, failing transmission lines with higher capacity yields more transmission line failures in the successive stages due to the difference between load and generation.

Cumulative installed capacity of the failed lines, CC_{fail}: We keep track of the cumulative installed capacity of the failed lines. Note that the installed capacity of a transmission line is the quantized capacity chosen from the set of five capacities.

Number of islands, $N_{islands}$: Finally, we include the number of islands formed due to cascading failures in power grids as a model feature. Since islands are formed as a result of failures of transmission lines that break the power grid into small self-sufficient microgrids, it is intuitive that the probability of a large cascade is very high if the number of islands is very high.

On top of these features of the grid during cascading failures, we track the following two output labels.

Number of failed lines, N_{fail}: We track the number of failed transmission lines after the cascade ends as an output label. We classify the number of failed lines into three distinct classes of the cascade (no, small, and large cascade).

Cumulative amount of load shed, CLS: We use the optimal power flow algorithm from MATPOWER, which includes the capability of implementing load shedding depending on the cost. Here, we set the cost of load shedding ten times higher than the cost of generation to ensure maximum generation before any load shedding. We track the cumulative amount of load shedding as a critical grid parameter.

In general, the stress over the power grid increases as we increase the operating parameters. From the simulations, we observe that depending on the topology, power grid operating parameters, and initial disturbances, the severity of the cascading failure varies from no cascading failure to complete blackout of the power grid. The correlation among the features is shown in Fig. 2. Observe that, the correlation among the topological parameters B and S_p with the number of failed lines and the amount of load shed is insignificant, which matches with the observation of [12]. The correlation plot is handy to visualize the correlation between the features of the grid.

We solve the following DC power flow equation where $\boldsymbol{F}$ is a vector of power flow in transmission lines, $\boldsymbol{A}$ is a matrix whose elements can be calculated in terms of the connectivity of transmission lines in the power grid and the impedance of the lines, and $\boldsymbol{P}$ is a vector which contains the generator and load power information [3].

$$F = AP \tag{1}$$

Similar to [3], we minimize the following cost function

$$\text{cost} = \sum_{i \in G} w_i^g g_i + \sum_{j \in L} w_j^l l_j \tag{2}$$

with the following optimization constraints

(1) Power flow equations (1).

(2) Generator power: $0 \leq g_i \leq G_i^{\max}, i \in G$

(3) Controllable loads: (1-θ_j)L_j $\leq$b$_j$ ≤ 0,
i $\in L$,l$_j$ $= b_j + \theta_j L_j$

(4) Transmission line power flow: $F_k \leq C_k^{\text{opt}}$

(5) Power balance: $\sum_{i \in G} g_i + \sum_{j \in L} l_j = 0$,

where g_i is the output of each generator, l_j is the delivered power at each load and L_i represents the demand at the load bus i . Moreover, C_k^{opt} is the capacity of a transmission line, $G_i^{\max}$ is the generator capacity, and G and L represent the set of generator and load buses. Controllable loads are defined using the θ parameter. Finally, the power balance is done by the optimizer using a reference generator, typically the generator with the largest generation capacity. The output of MATPOWER contains power flow through each transmission line satisfying the constraints. The optimal power flow utilizes dispatchable loads to implement load shedding when the cost of generation is higher than the cost of serving loads. Since the load shed cost is set ten times higher than the generation cost, load shedding is only performed when the optimizer fails to satisfy the optimization constraints.

C. Algorithms for finding the island and overloaded lines

Recall that we use DC power-flow, which is used in many earlier cascading failure analysis [3], [7], [12], [27] for its simplicity yet effectiveness. However, the introduction of the fixed failure probability of the neighboring lines transforms our model from deterministic to stochastic for the same initial conditions. We use the following algorithm for finding the maximum overloaded lines shown in algorithm 1.

Here in the algorithm, PF is the power flow through transmission lines, M is the total number of transmission lines in the power grid, P$_{\text{lf}}$ is the ratio of the absolute power flow and the adjusted installed capacity (using the capacity estimation error, e), ProbTest is a variable used to find the maximum overloaded transmission line, and FailedIndex is the index of the maximum overloaded transmission line. From the power flow data, overloaded transmission lines can be calculated, which is used in several previous works [3], [7], [13], [28] to fail transmission lines in the power grid. We consider a line failed when power flow through line exceeds the maximum allowable power flow limit through that transmission line. Once we find overflow in a transmission line, we fail that line and re-calculate optimal power flow (OPF) using the remaining transmission lines. We take one or multiple transmission line failures per time unit to understand the cascading failure dynamics effectively. One way to choose one transmission line to fail out of all the overloaded lines is that if multiple transmission lines exceed the capacity threshold, we fail the line with the maximum deviation from the overflow threshold. Since the power grid needs to balance generation and load, the overloaded failed transmission lines can initiate a cascade of failures in the successive steps.

Algorithm 1 Finding maximum overloaded lines probabilistically during cascading failures

```
Require: $e, \text{PF}, \text{Capacity}$
Ensure: FailedIndex
for $i \leftarrow 1$ to $M$ do
    $\text{P}_{\text{lf}}(i) \leftarrow \text{abs}(\text{PF}(i))/((1-e) * \text{Capacity}(i))$
ProbTest $\leftarrow 0$
for $i \leftarrow 1$ to $M$ do
    if rand $<$ LinkProb$(i)$ then
        if (ProbTest $<$ P$_{\text{lf}}(i)$) then
            ProbTest $=$ P$_{\text{lf}}(i)$);
            FailedIndex $= i$
     return FailedIndex
```

Algorithm 2 Solving power flow in each islanded grid

```
Require: mpc, s, c
Ensure: GD, PF
SG = sparse(AdjMatrix)
[s, c] = graphconncomp(G)
for $i \leftarrow 1$ to $s$ do
    struct mpc[$s$] $\leftarrow$ mpc
    [PF, GD] = rundcopf(mpc[$s$])
     return GD, PF
```

Here mpc contains the IEEE 118 bus system data available in MATPOWER, sparse and graphconncomp are two Matlab functions used to find the number of islands and their components, rundcopf is the MATPOWER optimal power flow function used to solve the power flow optimization problem. AdjMatrix is the adjacency matrix of the power grid that represents the connectivity pattern, variables s and c given the number of islands and number of components at each island respectively and GD represents generator output. Algorithm 2 shows our methodology for solving power flow in each of the islands created during the simulation. Recall that an island in a power grid is a self-sufficient grid network containing both loads and generators created when transmission line failure breaks the vast connected grid into a small localized connected grid. During each iteration, we calculate the number of islands formed in the grid and solve power flow simulation at each island using algorithm 2.

III. Results

In this section, first, we discuss the statistics of the data set and then implement machine learning algorithms to predict

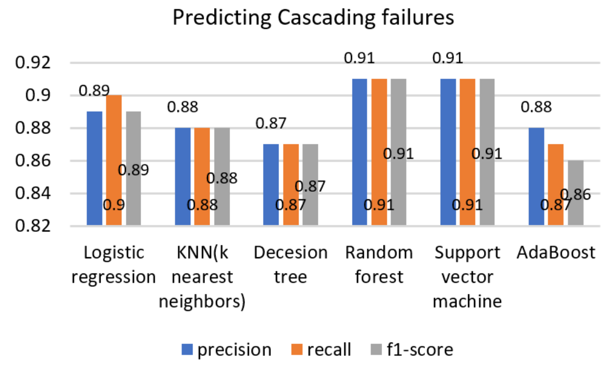


Fig. 3: Comparison of the overall accuracies for predicting cascading failure in power grids using machine learning algorithms.

cascading failures in power grids.

A. Description of the data

We have performed simulations over seventy-six thousand iterations of two random transmission line failures using the CFS framework on the IEEE 118-bus test case to collect failure data. We randomly select r, e, θ, and f_p from the sets defined earlier. We also calculate the topological parameters such as edge betweenness centrality, shortest distance as defined previously. Moreover, we store the data of the power flow through the initially failed transmission lines. The CFS framework output provides the number of transmission line failures after the cascade ends, the amount of load served, load shed, and the number of islands formed. We calculate and store the input-output parameters for all the iterations.

B. Analysis on cascading failure prediction

We have used the data set obtained from MATPOWER simulation to perform classification using the python scikit learn library [29]. The data set contains the number of transmission line failures after the cascade ends. As mentioned above, we quantize the number of failed transmission lines in three classes: no cascade (number of transmission lines failures $\leq$10), small cascade (10 $<$number of transmission lines failures $<$ 25), and large cascade (number of transmission lines failures$\geq$ 25). Note that the ranges of transmission line failures for the three classes were chosen arbitrarily. We split the data set for training (70%) and testing (30%) purposes. We have used nine features, as described above. We have used the following machine learning classification algorithms [20], [30]: logistic regression, k-nearest neighbors (KNN), decision tree, random forest, support vector machine (SVM), and AdaBoost. The precision, recall, and f1-scores are calculated using [29] for all the algorithms and shown in Fig. 3. It can be observed that all the classification algorithms have a relatively higher accuracy of classification with SVM and random forest having the best precision. Next, we show the individual cascade type classification accuracies in Fig. 4. We can observe that the classification of no cascade has higher precision compared to the classification of high cascades. This is because high cascades have low test samples compared to no cascades. For KNN, we further calculate the optimal k that yields the lowest error rate and observe that $k = 9$ gives the highest accuracy.

TABLE I: Prediction error

metric	error (number of failed lines)	error (amount of load shed)
mean absolute error	1.9	82.92
mean square error	7.07	15334.84
root mean square error	2.66	123.83

C. Linear regression to predict the number of transmission line failures and the amount of load shed

We use linear regression [20] to predict the number of cascading failures and the amount of load shed, which are shown in Figs. 5 and 6, respectively. The scatter plot of Fig. 5 shows that the relationship between the test data and the predicted values are linear, which indicates that linear regression is a reasonable model to predict the number of line failures. The error (deviation) of the predicted values from the actual value is reported in table 1. Note that the error of predicting the amount of transmission line failures is relatively small. Also, in Fig. 6, it is visible that the plot is not exactly linear, which indicates that linear regression may not be a good model for predicting the amount of load shedding.

IV. Conclusion and Future Works

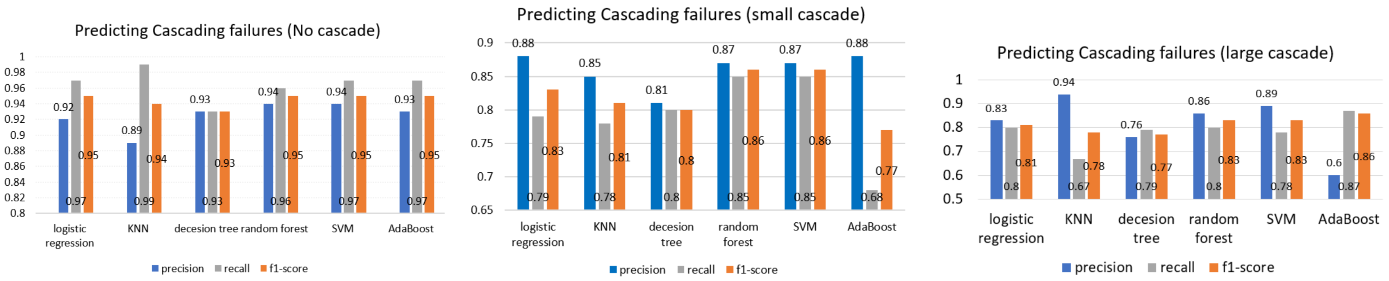
In this paper, we have used machine learning algorithms to predict cascading failures in power grids and also used linear regression to predict the number of transmission line failures, as well as the cumulative amount of load shed given an initial operating condition. Using an earlier developed CFS framework, we have effectively generated a labeled cascading failure data set by simulating IEEE 118-bus system, which is used as an input to the machine learning models. The results suggest that cascading failure prediction can be made using machine learning with high accuracy. The exploratory data analysis reported in this paper can be extended to find the distribution of line failures, distribution of load shedding, and critical line identification. In addition, the CFS framework needs to consider the generator dynamics, the disturbance of grid communication and control system, and grid operator error to analyze the dynamics of cascading failures in a more realistic setting, although that will increase the computational complexity of the CFS framework.

References

[1] R. Baldick *et al.*, "Initial review of methods for cascading failure analysis in electric power transmission systems ieee pes cams task force on understanding, prediction, mitigation and restoration of cascading failures," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, IEEE, 2008, pp. 1–8.

[2] P. Hines *et al.*, "Trends in the history of large blackouts in the united states," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, IEEE, 2008, pp. 1–8.

[3] Rahnamay-Naeini *et al.*, "Stochastic analysis of cascading-failure dynamics in power grids," *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1767–1779, 2014.



(a) no cascade

(b) small

(c) large cascade

Fig. 4: Comparison of the accuracies for predicting cascading failure in power grids using machine learning algorithms for the three classes of cascades.

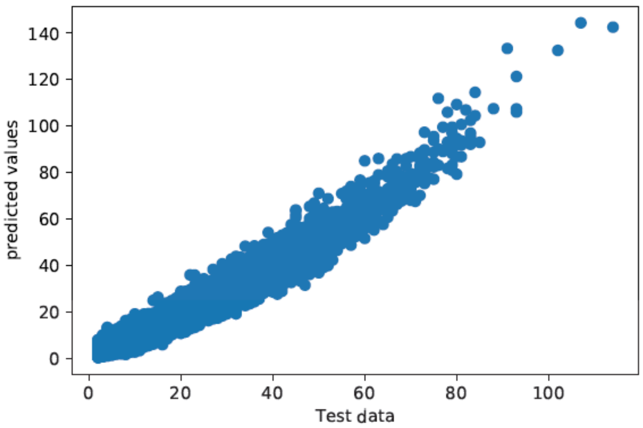


Fig. 5: Predicting the number of line failures

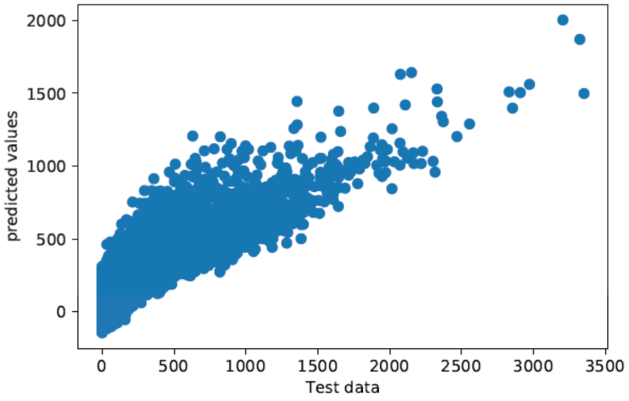


Fig. 6: Predicting the cumulative amount of load shed

[4] M. Rahnamay-Naeini and M. M. Hayat, “Cascading failures in interdependent infrastructures: An interdependent markov-chain approach,” *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1997–2006, 2016.

[5] B. Liscouski *et al.*, “Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations,” *A report to US Department of Energy*, vol. 40, no. 4, p. 86, 2004.

[6] H. Guo *et al.*, “A critical review of cascading failure analysis and modeling of power system,” *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 9–22, 2017.

[7] Y. Yang *et al.*, “Small vulnerable sets determine large network cascades in power grids,” *Science*, vol. 358, no. 6365, 2017, ISSN: 0036-8075. DOI: 10.1126/science.aan3184.

[8] I. Dobson *et al.*, “Branching process models for the exponentially increasing portions of cascading failure blackouts,” in *System Sciences, 2005. HICSS’05. Proceedings of the 38th Annual Hawaii International Conference on*, IEEE, 2005, 64a–64a.

[9] B. Schäfer *et al.*, “Dynamically induced cascading failures in power grids,” *Nature communications*, vol. 9, no. 1, p. 1975, 2018.

[10] S. V. Buldyrev *et al.*, “Catastrophic cascade of failures in interdependent networks,” *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[11] C. D. Brummitt *et al.*, “Suppressing cascades of load in interdependent networks,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 12, E680–E689, 2012.

[12] M. Korkali *et al.*, “Reducing cascading failure risk by increasing infrastructure network interdependence,” *Scientific Reports*, vol. 7, 2017.

[13] Z. Wang *et al.*, “Impacts of operators’ behavior on reliability of power grids during cascading failures,” *IEEE Transactions on Power Systems*, 2018.

[14] R. A. Shuvro *et al.*, “Modeling cascading failures in power grids including communication and human operator impacts,” in *2017 IEEE Green Energy and Smart Systems Conference (IGESSC)*,

[15] R. A. Shuvro *et al.*, “Modeling impact of communication network failures on power grid reliability,” in *North American Power Symposium(accepted)*, IEEE, 2017.

[16] P. Henneaux *et al.*, “Benchmarking quasi-steady state cascading outage analysis methodologies,” in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, IEEE, 2018, pp. 1–6.

[17] S. Gupta *et al.*, “Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2478–2486, 2015.

[18] R. Pi *et al.*, “Machine learning based on bayes networks to predict the cascading failure propagation,” *IEEE Access*, vol. 6, pp. 44 815–44 823, 2018.

[19] R. C. Hink *et al.*, “Machine learning for power system disturbance and cyber-attack discrimination,” in *2014 7th international symposium on resilient control systems (ISRCS)*, IEEE, 2014, pp. 1–8.

[20] G. James *et al.*, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[21] M. Hall *et al.*, “The weka data mining software: An update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[22] A. Almalaq and G. Edwards, “A review of deep learning methods applied on load forecasting,” in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2017, pp. 511–516.

[23] R. D. Zimmerman *et al.*, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2011.

[24] A. J. Wood *et al.*, *Power generation, operation, and control*. John Wiley & Sons, 2013.

[25] R. Christie, “118 bus power flow test case,” 1993.

[26] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[27] I. Dobson *et al.*, “Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, p. 026 103, 2007.

[28] Q. Chen and L. Mili, “Composite power system vulnerability evaluation to cascading failures using importance sampling and antithetic variates,” *IEEE transactions on power systems*, vol. 28, no. 3, pp. 2321–2330, 2013.

[29] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[30] S. B. Kotsiantis *et al.*, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.