Data-driven Contingency Selection for Fast Security Constrained Optimal Power Flow

Constance Crozier, Kyri Baker
University of Colorado Boulder
Boulder, CO, USA
{constance.crozier, kyri.baker}@colorado.edu

Yuhan Du, Javad Mohammadi
University of Texas Austin,
Austin, TX, USA
{yuhandu, javadm}@utexas.edu

Meiyi Li
Carnegie Mellon University,
Pittsburgh, PA, USA
meiyil@andrew.cmu.edu

***Abstract*—Security constrained optimal power flow aims to minimize power system operating cost while considering a defined set of outage scenarios. Contingency selection is the process of reducing the number of scenarios to be considered in the problem, thus reducing the computational complexity. In this paper we investigate whether two data-driven methods (artificial neural networks and k-nearest neighbors) can be used to perform extremely fast contingency selection. We demonstrate that both methods are orders of magnitude faster than standard methods. We show that k-nearest-neighbors works well when the algorithm is training on only data specific to the network, while neural networks are better when trained on multiple networks' data.**

***Index Terms*—Contingency selection, Neural networks, K-nearest neighbors, Security constrained optimal power flow**

I. Introduction

Security constrained optimal power flow (SCOPF) extends the traditional optimal power flow problem to a two-stage stochastic optimization model. The problem seeks to find the lowest cost way to deliver power to consumers, while considering a set of pre-defined contingencies (such as a component outage).

The pre-contingency operation (known as the base case) and contingency cases are linked by ramping constraints, which describe the maximum change generation output can be achieved once a contingency has been realized. The optimal base case solution is therefore determined by solving a master problem, which contains a set of variables for each contingency. This is referred to as preventative SCOPF, as it seeks to find a solution that prevents issues in the case of a contingency occurring. For large networks, with many contingencies, the resulting problem contains millions of variables. Given that the power flow constraints are non-convex, it may not be possible to reach a solution to the full problem in the required timescale [1]. Although many methods for preventative SCOPF have been proposed over the past decades, solving these problems for large networks in reasonable timescales is still an active research topic – as indicated by the recent ARPA-E Grid Optimization (GO) competition [2].

If a contingency is not included in the master problem, then the network performance in that contingency is constrained by the ramping limits from the base case solution. However, some contingencies will not impact the optimal base case solution. Therefore, optimal performance can be achieved without including them in the master problem. Contingency selection is the process of reducing the set of contingencies [3].

Many methods have been proposed for contingency selection, the majority of which involving computing an AC power flow solution for each contingency to assess the severity. For example, [3] computes the constraint violations using an intermediate base case solution for all contingencies and selects those with large constraint violations. While [4] uses an iterative process repeatedly calculating pre and post contingency load flows. These methods have also been extended to include distribution grids [5] and uncertainty due to renewable generation [6]. These methods are for N-1 contingencies, i.e. with one single component failure, however [7] proposes a method for simplifying N-2 contingency selection such that only the N-1 power flows must be computed.

The AC power flow equations are non-linear and there many be many contingencies, so these methods may add a significant computational overhead before the SCOPF can begin. This might not be realistic, given that deterministic values for load and generator availability are typically required, and these can not be predicted accurately for large time horizons [8]. Other papers use linearized forms of the power flow calculations. For example, [9] uses the DC form of the equations, but these do not include voltage magnitude, so not all constraint violations can be predicted. On the other hand, [10] propose a linear estimation for voltage drop, but these will not identify line limit constraints. For line outage contingencies, methods based on graph theory have also been proposed [11], [12], but the computational burden of these methods is large for big, well-connected networks.

Some methods avoid calculating the power flow solutions, and directly estimate the value of including the contingency. For example, using simple metrics based on voltage [13], or the maximum capacity of the failed component [14], but these heuristics have limited accuracy. More computationally complex direct forecasting methods have also been proposed, such as using particle swarm optimization and tabu search [15].

In practice a transmission system operator would need to implement contingency selection very quickly, so speed is paramount. However, an operator has a wealth of experience specific to their network, which can be leveraged to assess which contingencies may have the most impact. Therefore, a data-driven approach to ranking the potential component

978-1-6654-1211-7/22/$31.00 ©2022 IEEE

outage may hold promise.

Data-driven methods have been previously used with the AC power flow calculations in contingency analysis. Artificial neural networks (ANNs) have been incorporated into contingency selection to speed up the power flow calculations [16], [17]. Alternatively, in [18] the K-nearest neighbors (KNN) algorithm is used to classify the result of contingency load flows, providing a more sophisticated analysis of the results.

In this paper, we investigate whether data-driven methods can replace the AC power flow calculations entirely in contingency analysis, and directly predict the important of including a contingency in the power flow problem. Although (as discussed above) previous methods have attempted to bypass the AC power flow calculations, to the authors' knowledge none of these have been data-driven. Part of the difficulty of this approach is that data-driven methods require training data. In this case, the data needs to describe the value of including a contingency in the master SCOPF problem for historic scenarios, and this is difficult to quantify.

We focus on the ANN and KNN algorithms, as they have shown success in similar cases. We will consider two specific cases: one where the training data is specific to a single network but with different loading and generation scenarios, and one where the training data contains many different networks. The first case is analogous to the problem faced by transmission system operators, the second case is analogous to the problem posed in the ARPA-E GO competition.

The contributions of the paper can be summarised as follows. First, that we propose a method for generating training data describing the value of including a contingency in the master problem. Second, that we test the ANN and KNN algorithms in performing contingency selection on a given network with an unknown loading. Third, that we test the algorithms ability to perform transfer learning, where data from other networks is used for training data.

II. SCOPF FORMULATION

Security constrained optimal power flow (SCOPF) is a two-stage stochastic optimization problem which seeks to find the lowest cost operation mode of a network, given a set of outage scenarios. Here we describe the standard SCOPF formulation which was used for this study.

A. Optimal power flow

The operation of the network at a given time instant is described by: the voltage magnitude and angle at each bus, v, θ, the real and reactive power output of each generator, p_g, q_g, and the real and reactive power of each load, p_j, q_j. The objective of optimal power flow is to find the generation outputs that meet the load at lowest cost, while satisfying system constraints. Typically the load values are fixed, and the voltage values are unknown. Here we consider a cost function that is quadratic in terms of real generation output, such that the objective can be described as:

$$\min_{v,\theta,p,q} \sum_g c_{g_0} + c_{g_1} p_g + c_{g_2} p_g^2 \,, \tag{1}$$

where $c_{g_0}, c_{g_1}, c_{g_2}$ represent the constant, linear, and quadratic cost coefficients of generator g. The problem constraints can be written as:

$$v_{min} \leq v_i \leq v_{max} \quad \forall i \tag{2}$$
$$0 \leq \theta_i \leq \pi \quad \forall i \tag{3}$$
$$p_{g_{min}} \leq p_g \leq p_{g_{max}} \quad \forall g \tag{4}$$
$$q_{g_{min}} \leq q_g \leq q_{g_{max}} \quad \forall g \tag{5}$$
$$-p_{e_{max}} \leq p_e \leq p_{e_{max}} \quad \forall e \tag{6}$$
$$-p_{f_{max}} \leq p_f \leq p_{f_{max}} \quad \forall f \tag{7}$$
$$\sum_{g\in G_i} p_g + \sum_{j\in J_i} p_j + \sum_{e\in E_i} p_e + \sum_{f\in F_i} p_f = 0 \quad \forall i \tag{8}$$
$$\sum_{g\in G_i} q_g + \sum_{j\in J_i} q_j + \sum_{e\in E_i} q_e + \sum_{f\in F_i} q_f = 0 \quad \forall i \,, \tag{9}$$

where i represents the bus index, g the generator index, e the line index, and f the transformer index. Constraints (2) and (3) are bounds on the voltage magnitude and angle at each bus, (4) and (5) are bounds on the outputs of each generator, (6) and (7) are bounds on the capacity of each line and transformer, and (8) and (9) ensures real and reactive power balance at each bus. The line and transformer line flows (p_e, q_e, p_f, q_f) are non-linear functions of the voltage magnitudes and angles at either end of the line. Therefore, constraints (6)-(9) are non-convex. However, there are linear approximations which can be used which reduce the problem complexity.

B. Two stage stochastic problem

As the loading and generator availability varies with time, we need to optimize the network over several scenarios. Consider three instances of network operation. The first stage is the initially committed state of the network, which is the default state of all the generators and loads. Second, we consider the base case operation, which is the planned pre-contingency operation. Finally, we consider the post-contingency state, after which one of K contingencies has occurred. Figure 1 demonstrates these states, we use the subscript k_0 to denote the base case and k to denote the contingencies.

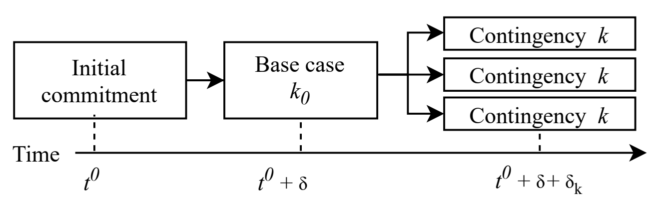

Fig. 1. The time steps considered in SCOPF.

The base case and contingency solutions are linked by ramping constraints. Given the time interval between the contingency and base cases is so short, the generator outputs can not be altered beyond reasonable bounds, so we impose:

$$p_{g,k_0} - r_g^d \leq p_{g,k} \leq p_{g,k_0} + r_g^u \quad \forall g \,, \tag{10}$$

where r_g^d is the maximum drop in generation that generator g can achieve in the time range, and r_g^u is the maximum increase.

There are two stages to the SCOPF problem. The first, called preventative SCOPF, seeks to find the optimal base case solution given all of the contingencies. This requires solving one large optimization problem which includes a set of network variables for each scenario. The second, called corrective SCOPF, calculates the optimal solution for each contingency, given a chosen base case solution. This involves solving k_T parallel optimization problems, with one set of network variables per problem (where k_T is the number of contingencies). The two stages are visualized in Fig. 2.

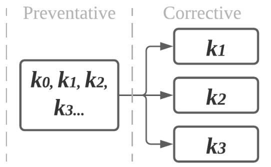

Fig. 2. The two-stage SCOPF problem.

The preventative SCOPF problem will have $(k_T + 1) * (2n_b + 2n_g)$ variables and $(k_T + 1) * (4n_b + 6n_g + 2n_e + 2n_f)$ constraints, where n_b is the number of buses, n_g the number of generations, n_e the number of lines, and n_f the number of transformers. For large networks with many contingencies, the required computation time to solve this problem may become impractical. Reducing the number of contingencies in the master problem can significantly decrease the number of variables and constraints. It is theoretically possible to achieve this reduction in complexity without loss of optimality because, in reality, many of the contingencies may not affect the optimal base case solution. So, if these contingencies can be identified, the problem complexity can be reduced without sacrificing optimality or feasibility.

This paper proposes a method to the predict the optimal a subset of contingencies to include that has the smallest effect on the problem optimality. There is no guarantee that only a small number of contingencies have an effect on the solution, therefore the problem is framed as a ranking one – the objective is to find the best n contingencies to include out of the k_T possible contingencies. The limit n is assumed to be set by the users computational and timing requirements, and a range of values are considered in this paper.

III. Data-driven Methods Background

Data-driven modelling learns a relationship between input and output variables based on observed data, rather than mathematical models. Training data refers to the set of input-output observations used to parameterize the model, which must be distinct from the input-output observations used to test the model. Here we present a brief over-view of the two specific methods used in this paper.

A. Artificial Neural Networks

Artificial neural networks (ANNs) resulted from the inspiration to model the behavior of neuron systems in the brain. An ANN is based on a number of connected nodes (or neurons) aggregated into different layers. Each neuron takes the weighted sum of signals from neurons in the previous layer and calculates the output by some non-linear function (called propagation function) then transmits it to its following layer. In this way, complicated input-output mappings can be modelled, where the weighting parameters of each branch is optimised to fit the training data.

B. K-Nearest Neighbors

K-nearest neighbors (KNN) is a non-parametric method commonly used for classification and regression. Non-parametric models are those which do not specify a specific structure for the input-output relationship. In this case, the algorithm finds the K data points that are *nearest* in feature space to the prediction point. For regression, the predicted value is the average of the output values associated with those data points. The average can be either a uniform average of the K points, or distance-weighted.

IV. Prediction Framework

The decision of whether to include a contingency is based on both the contingency value and the computational limits. It may be that there is value to including all contingencies in the master problem, but not sufficient computational resources to do so. Therefore, we frame the problem as a regression, which allows discrimination between contingencies, rather than a classification. We restrict our analysis to generator outage contingencies. Although the framework could be applied to any component outage, one of the input variables we use is specific to generator outages.

A. Generating Training Data

In order to train a data-driven algorithm to predict the value of including a contingency in the preventative optimization problem, observations of the value of including the contingencies is required. It is hard to define the “value” of including a contingency, because the chosen solution will depend on the other contingencies included in the master problem. Here we assign a value for contingency k by running a preventative SCOPF optimization with the base case and contingency k and a base-case only optimization. In each case the base case solution is for a corrective optimization for contingency k and the difference in objective is recorded (see Fig. 3). The corrective optimization first tries to solve a traditional optimal power flow, and if not feasible solution exists, solves a recovery problem to find the nearest-feasible solution.

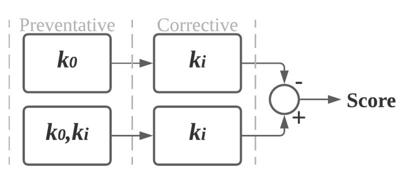

Fig. 3. The method for generating training data.

This effectively gives an upper bound to the improvement in the corrective SCOPF objective function that can be obtained by including a contingency in the master problem, given that the master problem considers only that single contingency and the base case. Contingencies which have no effect on the optimal base case solution have a value of zero and any contingency which alters the base case should get a positive score. Although this value may not be achievable once other contingencies are also included, it provides a proxy for the amount that a contingency can be improved by including it.

Note that, if a linearization of constraints (6)-(9) is used for the preventative optimization then it is possible for the score to be negative, i.e. for including the contingency to worsen its objective in the corrective optimization. The linearization means that the optimal point chosen in the preventative optimization is not AC feasible, so the chosen objective can't be reached in the corrective optimization.

B. Feature Selection

In data-driven modelling, the features of a problem and the input variables to the prediction problem. The features used for the prediction can have a large effect on the success of the neural network [19]. Therefore, the features chosen should describe the factors of the contingency thought to influence its importance. Given a limited amount of training data, choosing too many input parameters can prevent the network from learning meaningful relationships. Similarly, in KNN with too many features the euclidean distance between all points becomes large, and it is hard to find similar points. Therefore, here we consider three features which may have an effect on the value of including a generator outage contingency.

1) Real power output: The committed real power output dictates approximately the amount of power that will be lost from the network when the generator fails (given that base case power output is related by ramping constraints). Therefore, it is reasoned that outage of generators with relatively large power outputs may be more important to consider.

2) Voltage magnitude: There are constraints on the acceptable voltage magnitude at each bus. When the power injection into a bus changes, so does the voltage magnitude. Therefore, voltage magnitude may be an important consideration because generators at buses whose magnitude is closer to the voltage bounds might have a greater impact on the optimal solution.

3) Total load: The total load on the network is an example of a global variable, it will be the same for all contingencies within the same scenario. However, the total load provides context to the real output of the individual generator, as well as describing the difference between scenarios and networks.

It should be noted that all of these features are variables of the optimization, so we can not know their precise value. However, we have their values at the initially committed position of the network which, given the ramping constraints and underlying physics of the network, should provide rough estimates of their final values.

C. Algorithm Tuning

Although the methods are data-driven, both have design decisions to be made before they can be applied.

1) ANN: There are several decisions to be made when building an ANN. Firstly, the structure of the network needs to be determined. The larger the number of nodes and layers, the more complex relationships can be captured by the network. However, the number of parameters can grow rapidly with additional nodes and layers, creating a danger of over-fitting (where the parameters fit the training data very well but performs poorly on unseen data). Given that there are a limited number of loading scenarios and generator outages that can be generate for a single network, limited training data is available. Here we choose a network with two hidden layers of 4 nodes each, giving it a structure of 3-4-4-1.

2) KNN: The K-nearest neighbors algorithm has only one parameter: K, the number of nearest points to be considered. Here we take the value $K = 8$, a relatively large value. Larger values of K risk producing less accurate predictions, but make the algorithm less sensitive to bad data. There is also choice whether to use a uniform or weighted average of the closest K points. Here we select a distance-weighted average, such that the closer points have a stronger effect on the predicted value. This was chosen because the higher value of K means that some of the eight closest points might not be strongly related to the predicted contingency. The KNN algorithm is implemented using the k-d tree method in sci-kit, which structures the data during training resulting in faster predictions.

D. Evaluation Metric

There are standard metrics that evaluate how accurately algorithms predicts their numeric target. For this use case however, it is not the numerical accuracy of the predicted scores that matters, but how well the algorithm performs contingency selection. Given that the contingency selection algorithm picks the highest value n contingencies to include, it is the relative size (or ranking) of the predicted scores that is important. Therefore, we define the accuracy of the contingency filtering algorithm to be:

$$\text{Accuracy} = \frac{1}{n} \sum_{i}^{n} y_i \hat{y}_i \,, \tag{11}$$

where y_i is a binary variable which is 1 if contingency i is in highest scoring n contingencies, and $\hat{y}_i$ is a binary variable that describes whether contingency i has one of the n highest predicted scores. Therefore, the accuracy is the percentage of the highest value n contingencies which are in the highest n predictions. This metric is a function of n, the number of contingencies that can be included. For example, if all contingencies can be included then any contingency selection algorithm will have an accuracy of 100%. Therefore, it is important to evaluate the performance of the algorithm over a range of values of n. Additionally, we will consider the expected performance of a random contingency selection algorithm, which can be expected on average to have an accuracy of $\frac{n}{K_T}$ where k_T is the total number of contingencies. A contingency selection algorithm an only be regarded as successful if it consistently beats the performance of the random algorithm for various values of n.

V. Results

The proposed algorithm was tested on a 617 bus synthetic network from the ARPA-E Grid Optimization Competition [20].The network has 405 loads, 94 generators, 723 lines, 130 transformers, and 50 shunts. In the dataset 30 scenarios are provided, with the demand and generator cost information varying between scenarios. An additional 30 scenarios were generated by randomly scaling the generator and load bounds by factors between 95% and 105%. In each scenario there are 94 generator outage contingencies, one for each of the generators. Voltage bounds of 0.95 to 1.05 are used. We consider two applications of the learning algorithms: trying to learn a contingency filtering strategy specific to a single network, and trying to create a network invariant strategy.

A. Single network learning

In the single network case, we are training an algorithm using only data from other generation and loading scenarios of the 617 bus network. A total of 60 loading scenarios are used, 1 for testing and 59 for training. We include 94 data points per scenario, one for each generator on the network, such that there are 5546 training data points.

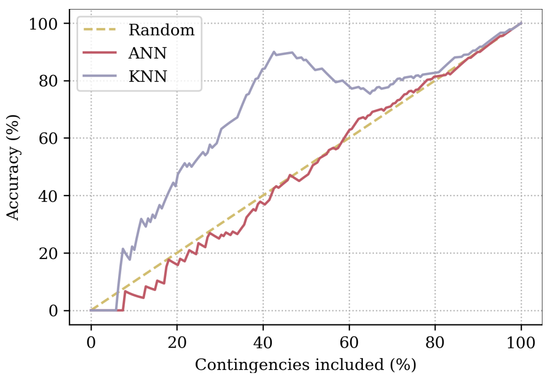

Fig. 4. The accuracy of the proposed methods compared to the expected performance of a random contingency selection for the single network case.

Figure 4 shows the accuracy of both the ANN and KNN algorithms described in Section IV-C. Also shown is the expected performance of a random selection algorithm. The KNN algorithm outperforms the random selection algorithm consistently, achieving above 60% accuracy once more than 25% of contingencies are included. At this level of accuracy, the KNN algorithm would accurately identify many of the valuable contingencies to include, although not the ideal subset. On the other hand, the ANN algorithm does not out perform the random selection algorithm.

It may be that the KNN algorithm performs better because it focuses on the most similar training data points. Given that the dataset contains only contingencies from the 617 network, those closest related are likely to give a good indication of the test contingency score. Meanwhile, ANNs attempt to develop a model that maps inputs to outputs, with the training data used to build the parameters. For the single network case, there may not be a diverse enough set of training data available to parameterize such a model.

The times taken for each algorithm to evaluate a single contingency compared to calculating a load flow are shown in Table I. We consider both the 617 bus network and a much larger network. It can be seen that both ANN and KNN are several orders of magnitude faster than calculating an AC load flow solution. It can be seen that the low flow computational time grows substantially with network size, such that it may be infeasible to evaluate all contingencies in time. The ANN evaluation complexity is the same for the larger network, although it takes slightly longer to read in the network data. KNN prediction complexity grows logarithmically with training data size, and for the larger network the number of possible outages is higher. However, even for the 31k network, the prediction time is extremely fast.

TABLE I
TIME TAKEN TO EVALUATE A SINGLE CONTINGENCY.

No. buses	ANN	KNN	AC Load Flow
617	0.00021 s	0.00067 s	0.37 s
31777	0.00027 s	0.01081 s	37.15 s

B. Cross network learning

For the cross network learning case, we aim to train a network invariant model that can perform contingency filtering for any network. We consider two distinct cases. In the first, other loading scenarios for the 617 bus network are included in the training data, such that the testing network has been "seen" before. In the second case, no training data from the 617 bus network is included, so the testing network is "unseen".

In the "seen" network case there were 13307 training data points, representing generator outage events from various loadings of seven different networks. The number of buses of these networks ranged from 500 to 768 – similar sized networks were chosen to maximize relevance. Figure 6 shows the accuracy of both training algorithms in this case, compared to the expected performance of a random selection algorithm.

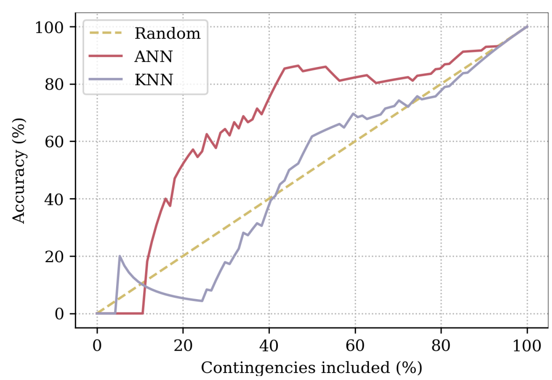

Fig. 5. The accuracy of the proposed methods compared to the expected random performance using cross network learning for a "seen" network.

In this case the ANN algorithm performs relatively well, while the KNN algorithm performs badly. This is likely because the KNN is no longer only using directly relevant data, and the linear method to define the distance between points is too simple to determine how relevant other networks' data is. Additionally, because KNN uses an average of the relevant data, the value of a single training point can drastically effect the predicted values if it becomes included in the closest K points. Whereas, the ANN's improved performance is likely due to the larger and more diverse training data set – which

TABLE II
AVERAGE ABSOLUTE ERROR FOR EACH CASE AND METHOD.

Case Study	ANN	KNN
Single network	13.69	1.13
Cross network (seen)	13.75	10.28
Cross network (unseen)	28.73	17.86

allows it to determine a more complex relationship mapping between input and output values. Lastly, Figure 6 shows the same results, but with all data from the 617 bus removed from the training dataset. Although an unlikely situation in practice, performing contingency filtering on an unseen network was required in the ARPA-E GO competition. In this case, both algorithms perform similarly, offering a modest improvement from the random selection case.

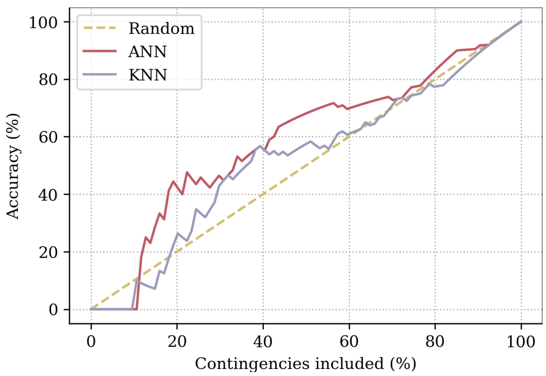

Fig. 6. The accuracy of the proposed methods compared to the expected random performance using cross network learning for an "unseen" network.

In order to understand the action of the algorithms further it is useful to also consider the accuracy which which they predicted the numeric values (rather than just the ordering). Table II shows the average absolute error of the contingency score for each case and method. In all cases KNN predicted the individual values more accurately, likely because it only uses a small but relevant subset of the training data. However, we know that the ANN algorithm performed contingency filtering more successfully in the cross network cases. Therefore, while KNN can predict the individual scores more accurately, ANN is sometimes better at discriminating between contingencies in a scenario. This makes ANN more suited for contingency filtering in these cases.

VI. CONCLUSION

This paper investigated the use of two data-driven methods: artificial neural networks (ANN), and k-nearest neighbors (KNN) for fast contingency selection in security constrained optimal power flow (SCOPF). Unlike previous methods, the algorithms were not used to speed up power flow calculations, but to estimate the value of including a contingency in the master problem. Both methods sped up contingency calculations by orders of magnitude compared to traditional methods.

The proposed algorithm was tested on a 617 bus network with 94 generator outage contingencies. Both methods performed up to 10^3 times faster than traditional contingency screening methods. We showed that KNN can achieve a reasonable ranking of the contingencies when only given network specific data, while ANN performs poorly in this case. However, when a mix of different networks' data was used, the performance was reversed. This is likely because the ANN can capture more complex relationships but requires a large and diverse set of training data to achieve a good performance.

When only data from the 617 bus network is included, the relationship to be learned is less complex and the training data set limited. Therefore, KNN is more appropriate in this case. We also demonstrated that neither algorithm performed well when data from the 617 bus network was not included in the training set. Of the three cases considered, the first is the most practical, as transmission operators only have a single network on which to operate. Therefore, KNN could be a promising method for speeding up SCOPF in practice.

REFERENCES

[1] F. Capitanescu et al., "State-of-the-art, challenges, and future trends in security constrained optimal power flow," *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, 2011.

[2] ARPA-E, "Grid optimization competition," gocompetition.energy.gov.

[3] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, "Contingency filtering techniques for preventive security-constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 22, no. 4, 2007.

[4] A. S. Xavier, F. Qiu, F. Wang, and P. R. Thimmapuram, "Transmission constraint filtering in large-scale security-constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 34, no. 3, 2019.

[5] Z. Li, J. Wang, H. Sun, and Q. Guo, "Transmission contingency screening considering impacts of distribution grids," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1659–1660, 2016.

[6] M. Gupta and A. R. Abhyankar, "Transmission contingency selection considering impact of uncertain distributed generation," in *ICPS*, 2019.

[7] Y. Zhang, C. Li, G. Yang, and H. Zhu, "An anticipated multi-contingencies selection method based on fault influence domain," in *International Conference on Power System Technology*, 2014.

[8] R. Billinton and D. Huang, "Effects of load forecast uncertainty on bulk electric system reliability evaluation," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 418–425, 2008.

[9] P. Kaplunovich and K. Turitsyn, "Fast and reliable screening of n-2 contingencies," *IEEE Trans. Power Syst*, vol. 31, no. 6, 2016.

[10] B. Otomega and T. Van Cutsem, "Fast contingency filtering based on linear voltage drop estimates," in *IEEE Power Tech*, 2005, pp. 1–8.

[11] S. Eftekharnejad, "Selection of multiple credible contingencies for real time contingency analysis," in *2015 IEEE PESGM*, 2015, pp. 1–5.

[12] Y. Zhu, R. Dai, and G. Liu, "Parallel betweenness computation in graph database for contingency selection," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[13] S. Grillo, S. Massucco, A. Pitto, and F. Silvestro, "Indices for fast contingency ranking in large electric power systems," in *IEEE Mediterranean Electrotechnical Conference*, 2010, pp. 660–666.

[14] Y. Jia, P. Wang, X. Han, J. Tian, and C. Singh, "A fast contingency screening technique for generation system reliability evaluation," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4127–4133, 2013.

[15] F. Li and A. Chegu, "High order contingency selection using particle swarm optimization and tabu search," in *International Conference on Intelligent System Applications to Power Systems*, 2009, pp. 1–4.

[16] Y. Du, F. Li, J. Li, and T. Zheng, "Achieving 100x acceleration for n-1 contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Trans. Power Syst*, vol. 34, no. 4, 2019.

[17] Bharat Vyakaranam et. al, "Novel data-driven distributed learning framework for solving ac power flow for large interconnected systems," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 281–292, 2021.

[18] G. S. Rani, M. Chakravarthy, and B. Mangu, "Power system contingency classification using knn algorithm," EasyChair, Tech. Rep., 2018.

[19] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings*, D. Sleeman and P. Edwards, Eds. San Francisco (CA): Morgan Kaufmann, 1992, pp. 249–256.

[20] ARPAE, "Go datasets," gocompetition.energy.gov/challenges/1/datasets.