Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Evaluating Machine Learning Models for the Fast Identification of Contingency Cases

Florian Schäfer*[1] | Jan-Hendrik Menke[1,2] | Martin Braun[1,2]

[1]Department of Energy Management and Power System Operation e^2n, University of Kassel, 34121 Kassel, Germany
[2]Department of Grid Planning and Grid Operation, Fraunhofer IEE, 34121 Kassel, Germany

Correspondence
*Florian Schäfer, Willhelmshoeher Allee 73, 34121 Kassel, Germany. Email: florian.schaefer@uni-kassel.de

Present Address
Willhelmshoeher Allee 73, 34121 Kassel, Germany.

Abstract

Fast approximations of power flow results are beneficial in power system planning and live operation. In planning, millions of power flow calculations are necessary if multiple years, different control strategies, or contingency policies are to be considered. In live operation, grid operators must assess if grid states comply with contingency requirements in a short time. In this paper, we compare regression and classification methods to either predict multi-variable results, e.g., bus voltage magnitudes and line loadings, or binary classifications of time steps to identify critical loading situations. We test the methods on three realistic power systems based on time series in 15 min and 5 min resolution of one year. We compare different machine learning models, such as multilayer perceptrons (MLPs), decision trees, k-nearest neighbors, gradient boosting, and evaluate the required training time and prediction times as well as the prediction errors. We additionally determine the amount of training data needed for each method and show results, including the approximation of untrained curtailment of generation. Regarding the compared methods, we identified the MLPs as most suitable for the task. The MLP-based models can predict critical situations with an accuracy of 97-98 % and a very low number of false negative predictions of 0.0 - 0.64 %.

KEYWORDS:
contingency analysis; grid planning; machine learning; power flow; time series calculation

1 | INTRODUCTION

Power flow results are the basis for power system planning and are needed in live operation to assess the system state. Quasi-static time series simulations allow evaluating asset loadings, voltage profiles and contingency situations over a long period, e.g., multiple years. This has several advantages in the planning process compared to single "worst-case" analysis including the calculation of grid losses or the integration of demand and generation flexibility[1,2]. However, the computational effort is very high. Millions of power flow calculations are necessary if multiple years, different control strategies or contingency policies ("N-1" cases) are to be taken into account. For example, the simulation of one year in 15 min resolution ($T = 35,040$ time steps) for a grid with N lines requires $(N + 1) \cdot T$ power flow calculations, if the single contingency policy (SCP) criterion is taken into account. In live operation, grid operators must assess if a loading situation is "N-1" secure in a very short time. Here, fast approximations of contingency results, including line loadings and bus voltages, are helpful to determine the system

[0]**Abbreviations:** ANN, artificial neural network; ML, machine learning; MLP, multilayer perceptron; OPF, optimal power flow; PF, power flow; RES, renewable energy sources; SMOTE, Synthetic Minority Over-sampling Technique

security state rapidly. A promising method to identify critical loading situations is to use artificial neural networks (ANNs) as a regressor[3]. In this paper, we want to extend this approach to be able to use classification methods and additionally compare other regression models. It is our goal to:

- identify the most suitable regression and classification methods (neural networks, ridge CV, decision trees, extra trees, random forest, gradient boosting, k-nearest neighbors)
- compare the required training time and prediction time
- evaluate the approximation error
- determine the amount of needed training data
- test the approximation when using generation flexibility
- show results with two different training data sampling methods

The paper is divided into six sections. Section 2 gives an overview of the state of the art methods in the field of machine learning (ML) in power systems and compares our approach with other publications. Section 3 defines the problem tackled in this work and describes how we implement the regression and classification strategy. In Section 4, results are shown for different ML methods tested on three power systems. We identify the best methods, which we then compare on untrained data. In Section 5, we show results with an alternative training method that can be used if time series data is not available. In the last section, we give a conclusion and an outlook.

2 | STATE OF THE ART

With increasing computational power, ML research has gained momentum in various fields[4]. Comparisons in the finance sector[5] show that ANNs, gradient-boosted trees or random forests have different advantages and disadvantages depending on the problem. In power systems, ML methods are used for many years to predict time series or contingency cases. The prediction of load[6] and generation[7] time series is based on historical measurements and weather data. These methods focus solely on the time series, without taking into account the power system data, e.g. the line impedance values. Contingency analysis using ML methods, e.g. the prediction of bus voltages for a small test case, based on a radial basis function (RBF) is possible[8]. Time series are not taken into account. In comparison to modern deep learning methods, RBF neurons have a maximum activation when the center or weights are equal to the inputs. Therefore, higher extrapolation errors can be excepted. Another method to predict power flow results using ANN[9], trains a multilayer perceptron (MLP) with P, Q bus injections and predicts bus voltage magnitudes and angles. The idea is similar to the proposed regression approach in this paper. However, no contingency analysis is performed, time series are not being taken into account and results are shown only for small test systems.

Blackout predictions by applying classification methods in realistic test systems[10] show that MLPs and decision tree methods are able to classify system states, characterized by load level, bus voltages, power generation and contingency cases. N-1 contingencies can be predicted by applying a "guided dropout" method, which generalizes predictions for N-2 cases[11]. Further comparisons show that different supervised learning methods are able to predict optimal power flow (OPF) costs[12]. Calculation time is reduced by using multiple regression methods, including neural networks and tree-based models. Different learning algorithms can predict real-time reliability of power systems and costs of recourse decisions[13]. A reduction of calculation times in day-ahead operational planning, including the N-1 criterion, is possible[14]. Decision trees are used to learn data-driven security rules to assess and optimize power system reliability in live operation[15]. A decision tree classifier is trained on a large number of operating points whose fault status has been determined via time-domain simulations. Power system state estimation based on ANN predicts bus voltages and line loading results for selected system states[16]. The goal is to accurately estimate a system state in live operation with few measurements available for different switching states. Line outages and time series are not taken into account during training. An overview study[17] in the context of contingency analysis with artificial intelligence in planning and operational shows that: Most publications either lack of (1) realistic test grid sizes, (2) do not use grid specific time series for training and prediction, (3) do not take contingency analysis into account or (4) analyze only one ML method. In this paper, we want to identify which approach (classification or regression) and which models are best adapted for similar problems.

3 | IMPLEMENTATION OF THE REGRESSION AND CLASSIFICATION METHODS

We use open-source ML model implementations [18,19] to identify time steps with high line loadings or voltage violations for the given time series and grid data. The objective is to significantly reduce the calculation time with a minimal loss in precision by training either a regressor or classifier with a certain percentage of time steps of the power flow results and corresponding inputs. We use the regression and classification method to predict important system variables. The regressor is trained to predict the voltage magnitudes V_m of all buses and line loading values $I_{\%}$ of all lines in the grid for a time step. By comparing with the pre-defined limits, it can then be assessed if the time step is critical. The classifier is directly trained to predict whether a time step is critical. Figure 1 shows exemplary line loading results for 10 consecutive days with exceeded line loading limits (critical) between time step 57 and 87.

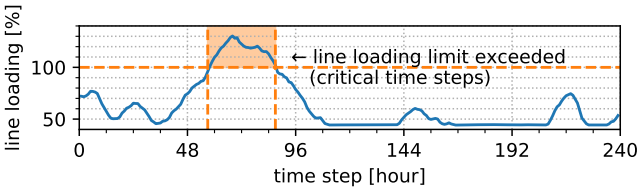

FIGURE 1 Example of critical time steps resulting due to high line loadings.

3.1 | Methods Overview

Fig. 2 shows an overview of the methods. The input data is identical for all methods and consists of the grid data with a fixed topology (switching state) and the real power P and reactive power Q time series for loads and renewable energy sources (RES) (PQ-nodes). We model large generation units as PV-nodes. For these generators, real power injections and bus voltage magnitudes are varied. In the following comparison, we assume static voltage set-points for PV- and slack-nodes. The time series can be derived from historical measurement data or by simulation. The power flow method iterates over all time steps, updates the P, Q values, and the bus voltage magnitudes V_m, voltage angles δ and branch currents $I_{\%}$. We compute line outages for each line $l \in N$, which results in N additional power flow calculations for each time step $t \in T$. In total $(N + 1) \cdot T$ power flow calculations must be calculated to obtain results for the base case and all contingency results.

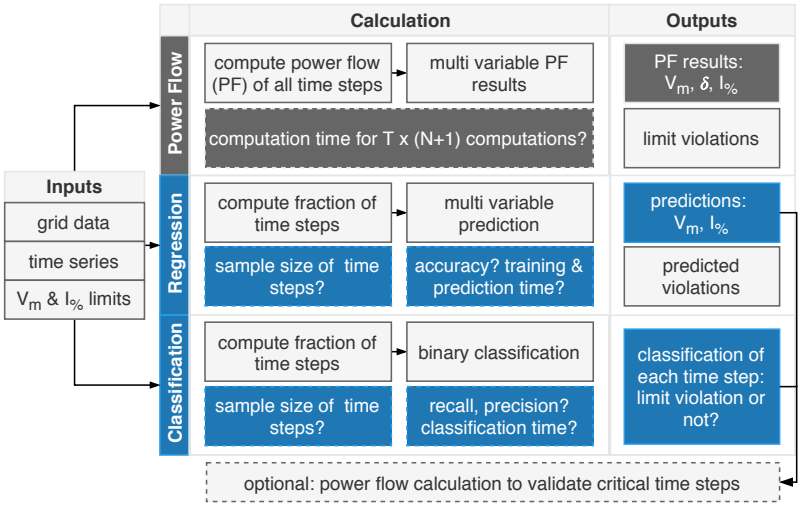

FIGURE 2 Overview of the quasi-static power flow, regression and classification method including research questions (dashed boxes).

To reduce calculation times, we simulate only a fraction of time steps to generate the training data for the regression and classification training by quasi-static power flow simulations of one year. These power flow results are the input for the regression or classification training. All regression and classification methods belong to the category of supervised learning algorithms, i.e., we provide training input data X and corresponding output data y. The training inputs contain parts of the known inputs of a power flow calculation. Outputs for the regression methods are the voltage magnitude approximations at each bus V_m and the line loading $I_{\%}$ of each line in per cent of the maximum line loading. Outputs for the classification methods are "critical" system state $s = 1$ or "uncritical system state" $s = -1$. We define a critical system by the violation of voltage magnitude limits or line loading limits. Optionally, power flow results can then be computed for the classified critical time steps or to validate regressor predictions. To test the proposed methods, we use time series data of one year with and without curtailed power injection of generators. Validation criteria for the regression method are the mean and the maximum error. We measure classification success by standard criteria, such as precision, recall and accuracy.

3.2 | Input Data

3.2.1 | Input Layer and Architecture

The input layer is identical for the regressor and classifier. Each feature, defined by (1), contains parts of the known variables of the power flow calculation for a time step t:

$$X_t = \begin{bmatrix} v_{m,r} & \delta_r & v_{m,gen} & p_{bus} & q_{bus} \end{bmatrix} \tag{1}$$

with $v_{m,r}$ and δ_r the voltage magnitude and angle of the reference buses and $v_{m,gen}$ the voltage magnitudes of the generator buses (PV-nodes). p_{bus} are the sum of the known real power values per bus including all loads and generators. q_{bus} are the known reactive power values of PQ-buses (aggregated load and RES values). We use the default hyperparameter settings [18], which show good results for common problems. A separated model is trained for each N-1 case.

3.2.2 | Training Data

We compute power flow results with pandapower [20] by iterating over all time steps. In the following comparisons, we include line contingency cases by setting each line out of service one after another and calculating the power flow results for the whole year. Depending on the total number of time steps T and the number of lines N in the grid, this process takes $T \cdot (N + 1) \cdot t_{pf}$ seconds, where t_{pf} is the average time for a single power flow calculation. We split the same time series in a training and a prediction set. An alternative method to generate training data is to use a scenario generator [16] or vine copulas [21]. The creation of training data with these methods is especially useful if no time series data is available or to obtain additional data for future planning. In Section 5, we show prediction results when using the scenario generator. The training with the scenario generator outlines that the model architecture is able to generalize from the training data and that it does not only learn to predict the remaining part of the time series.

3.3 | Regression Method

An individual regressor model is trained to predict voltage magnitudes $v_{m,\text{bus}}$ in per unit (p.u.) values of all buses as well as the line loadings $I_{\%,\text{line}}$ in percent of the rated current I_r.

$$y_1 = \begin{bmatrix} v_{m,\text{bus}} \end{bmatrix} \; y_2 = \begin{bmatrix} I_{\%,\text{line}} \end{bmatrix} \tag{2}$$

Relevant performance metrics for the regression method are the mean absolute error of the predictions as well as the maximum error. A low mean error is relevant if the regressor is used to predict the results of similar time steps. A low maximum error is needed when critical time steps/loading situations are to be identified.

3.4 | Classification Method

The classification output layer is defined by a binary state where $s = -1$ equals "uncritical" time step and $s = 1$ equals "critical" time step:

$$y_{\text{classifier}} = \begin{bmatrix} -1 & 1 \end{bmatrix} \tag{3}$$

A time step is critical when either the voltage magnitude of any bus is out of boundaries $v_m < v_{\min}$, $v_m > v_{\max}$ or the line loading $I_{\%}$ of at least one line violates its maximum ($I_{\%} > I_{\text{limit}}$). Operational restrictions for specific grids are defined by the power system operator individually. Typically, they are derived from standards such as the VDE-AR-N 4121[22]. We define a critical system state if, for any bus in the grid, the voltage magnitude violates a range between 0.9 *p.u.* - 1.1 *p.u.* of the nominal voltage magnitude V_m or the long term thermal line loadings is above their maximum loading I_{limit}.

3.4.1 | Performance Metrics

The classifier should preferably predict uncritical time steps as critical (false positives) and be less precise than fail to notice critical time steps (false negatives). Different metrics are commonly used to assess the performance of classifiers. Recall (4) measures the fraction of true positive (TP) classifications over the total amount of relevant instances. The relevant instances are the sum of TP and false negative (FN) classifications. Here, TPs are the correctly identified critical time steps and FN are critical time steps which have been mislabeled as uncritical.

$$\text{recall} = \frac{TP}{TP + FN} \tag{4}$$

In our case, the FN classifications should be minimized, since it is of high importance to identify all critical loading situations. We preferably tolerate some uncritical time steps identified as critical (false positives (FP)) than a high recall. The precision score measures the misclassification:

$$\text{precision} = \frac{TP}{TP + FP}. \tag{5}$$

When classifying critical loading situations, maximizing recall is more important than maximizing precision. The accuracy score (6) measures the ratio of correct classifications to all classifications. Here, TNs are the true negatives, which are the correctly classified uncritical time steps. The accuracy metric alone can be misleading for imbalanced datasets, where the majority of time steps are uncritical, and only a fraction is critical. In this case, the accuracy score is high by default when labeling every time step as uncritical.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

3.4.2 | Training of Imbalanced Datasets

The training data is imbalanced since the majority of time steps is "uncritical" with a few critical time steps to be identified (compare Fig. 1). We can achieve a high accuracy if all of the time steps are labeled as "uncritical". However, recall is also small in this case since the number of false negative is maximal. To overcome this issue, we predict the probabilities to which class the time step belongs to instead of predicting if the time step is critical or not. Since we have a binary classification, the classifier outputs a probability matrix of dimension $(T, 2)$. The first index refers to the probability that the time step is "uncritical", and the second refers to the probability that the time step is "critical". By reducing the probability threshold for the "critical" class, the number of positive predictions and recall increase while precision decreases. Additionally, we use Synthetic Minority Over-sampling Technique (SMOTE)[23] as an oversampling strategy to balance the dataset.

4 | RESULTS

4.1 | Case Data

We apply the regression and classification methods on three different synthetic grid models, which are derived from real power systems. All models and the corresponding time series are available in the open-data pandapower format. The characteristics of the SimBench (SB) grids[24] are typical for German meshed high-voltage grid topologies. Time series are available in 15 min resolution with 35,136 time steps in total. The Reliability Test System (RTS) test case[25] is a North American power system model with a time series resolution of 5 min, resulting in 105,408 time steps. In total, over 12 million power flow results must be calculated for the given time series of one year when assessing all N-1 cases. Fig. 3 shows the synthetic grids. Table 1 lists the relevant data of the three test cases. We compute all results on an Intel Core i7-8700K CPU at 3.70GHz speed.

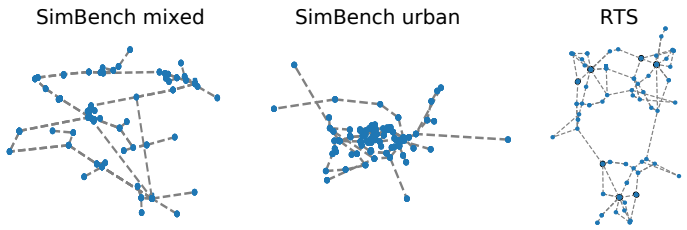

FIGURE 3 Analyzed test grids: SimBench (SB) mixed, SimBench urban, and RTS.

TABLE 1 Overview of grid data.

	SB mixed[24]	SB urban[24]	RTS[25]
voltage level [kV]	110	110	230
buses [#]	64	82	73
N-1 cases [#]	66	78	66
I_{limit} [%]	60.	60.	100.
$N_{\text{timesteps}}$	35,136	35,136	105,408
N_{PF} [10^6]	2.312	2.733	6.957

4.2 | Regression Results

We evaluate the performance of different regression methods by comparing the absolute error of line loading and voltage magnitude predictions while taking into account the training size as well as training and prediction time. We analyze regressors[18] that support multi-variable outputs: MLPRegressor (MLP), ExtraTreesRegressor (ET), DecisionTreeRegressor (DT), RandomForestRegressor (RF), RidgeCV (RCV).

First, we analyze how the prediction errors decrease with training data size. From the 31,536 (SB) or 105,048 (RTS) time steps, we randomly select training and test data by a shuffled train/test split. Based on the test data set, we evaluate the absolute prediction error. Fig. 4 shows the mean prediction error for the SB test cases (left) and the RTS test case (right) with increasing training sizes. All regressors improve with larger training sizes, except the RCV method. The prediction error decreases significantly with training sizes up to 10 %. Larger training sizes reduce the prediction error primarily for the MLP and DT regressors. We, therefore, use a train/test split of 0.1 / 0.9 for the following comparisons.

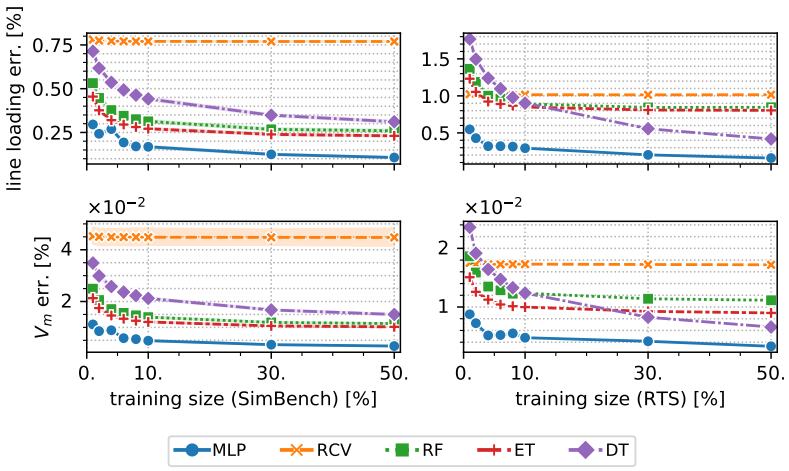

FIGURE 4 Mean prediction error with increasing training size for both SimBench test cases combined (left) and the RTS case (right).

Fig. 5 (a) shows the results of the prediction error without outliers. The MLP, ET and RF regressors yield the lowest mean error values for voltage and line loading predictions. The mean error of the regressor with the lowest error, the MLP, is only a third in comparison to the regressor with the highest error of the RCV method. The MLP has the lowest errors of all regressors when comparing the maximum error in Fig. 5 (b). The RCV, tree, and RF methods have significantly higher prediction errors. A longer training time (Fig. 5 (c)) is needed for the MLP in comparison to RCV, ET and DT. The time needed to predict the results (Fig. 5 (d)) is shortest for the DT, MLP and RCV methods.

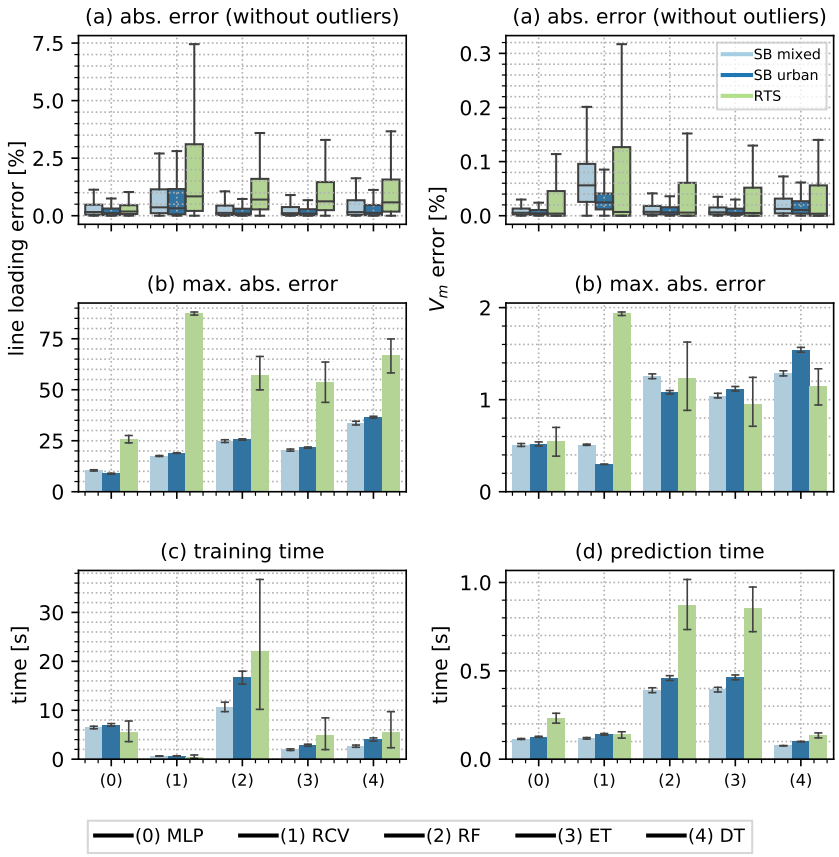

FIGURE 5 Results for the different regressors and test cases. The whiskers show the 95 % percentiles.

The MLP has the lowest overall error with decent training and low prediction time. With 10 % of all time steps being trained, the mean errors of line loading predictions are less than 0.25 % for the SB grids and less than 2 % for 99 % of the predicted values. Similarly, the voltage magnitude prediction errors are low with a mean value of 0.01 %. and 0.5 %. in the 99 % range. The mean prediction error for line loadings for the RTS test case is 0.5 % with 99 % of all values being predicted with an error less than 5 %. Voltage magnitude prediction errors are 0.05 %. (mean) and 0.5 %. (99 %). However, some outliers cannot be predicted with this accuracy.

4.3 | Classification Results

The goal of the classification is to detect time steps in the data set with high line loadings or voltage tolerance violations, which are categorized in "critical" and "uncritical". Power flow results for these time steps can be calculated separately if needed. The classifiers we analyze are: xgboost XGBClassifier (XGB)[19], RandomForestClassifier (RF), AdaBoostClassifier (AB), GaussianNaiveBayes (GNB), ExtraTreesClassifier (ET), MLPClassifier (MLP), KNeighborsClassifier (KN)[18].

Fig. 6 shows the classification accuracy and prediction timings for all classifiers. The AB and GNB classifiers have - on average - a much lower accuracy compared to the other classifiers. Their percentage of correct predictions was less than 90 % in all test

cases. The prediction by the KN classifier takes between 30 s and 1 min on average in comparison to less than 0.5 s by the other classifiers without being more accurate than the ET, RF, XGB and XLB classifier. We, therefore, conclude that the AB, GNB and KN classifiers are not as suitable for the classification of critical time steps as the other classifiers and exclude them from further comparisons.

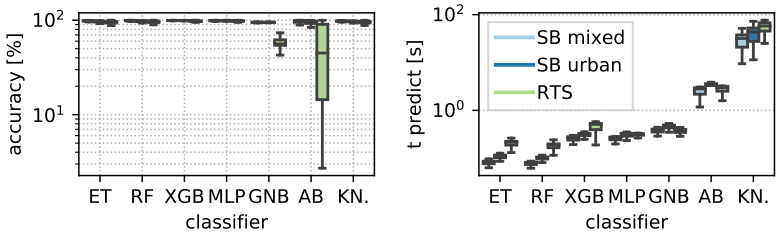

FIGURE 6 Classification accuracy and timings for all classifiers and test cases including contingency analysis.

Fig. 7 (left) shows the increasing accuracy and decreasing number of false negative predictions with increasing training size for the classifiers with an accuracy of more than 90 %. The ET and RF method can classify about 96 - 98 % of time steps correctly with approximately 1.0-1.5 % being false negatives. Both classifiers are outperformed by the MLP and the XGB methods. These methods have an accuracy starting at 98 % at a training size of 1 % of all time steps being trained. With an increasing training size both, MLP and XGB, achieve an accuracy of 99.5 % with 0.3 % of FN classifications at a training size of 50 % of time steps and N-1 cases being calculated.

Corresponding training and prediction times are shown for each grid and classifier in Fig. 7 (right). The training time for the ET and RF classifiers is on average much shorter (< 0.7 s with 50 % training data) compared to the MLP and XGB methods. Depending on the training size, the MLP training time takes ∼ 2.5 s for 1 % of the data up to more than one minute for the RTS test case. In comparison, the XBG is twice as fast in the RTS case with 0.9 s and 35 s respectively. Prediction times of the ET and RF methods are about one third compared to the times needed by the MLP and XGB methods. The time needed to predict the classification results is on average similar for the MLP and XGB methods with an exception in the RTS case. Here, the XGB needs twice the time (∼0.5 s) of the MLP (∼0.25 s). The difference in prediction time is negligible when taking into account the time needed to compute the training data (see Section 4.5).

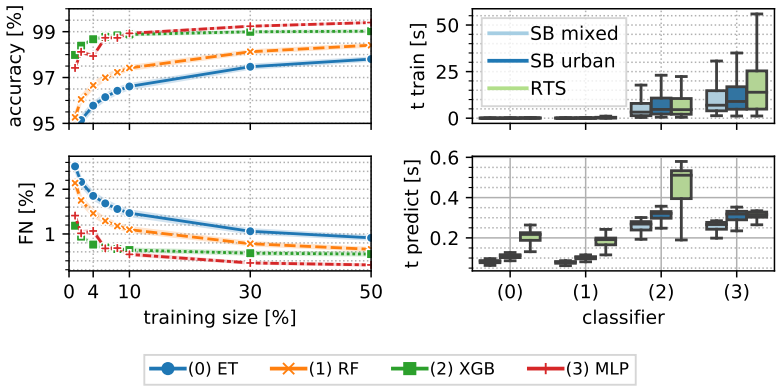

FIGURE 7 Classification accuracy and false negative rate with increasing training size for all test cases with N-1 predictions (left). Training and prediction times for each classifier and test case.

Fig. 8 shows the recall, precision and accuracy scores for the XGB and MLP classifier with a training size of 10 %. Both methods yield good results with accuracy values of at least 98 % correct classifications on average. The average recall score of

the MLP is higher compared to XGB but also has more outliers in some N-1 case predictions. The XGB classifier achieves a higher precision, on the other hand. Therefore, the classification with the MLP and XGB may help to identify critical time steps to run detailed analysis based on power flow calculations.

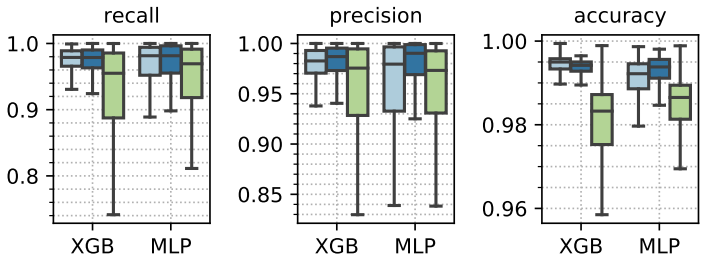

FIGURE 8 MLP and XGB recall, precision and accuracy scores including N-1 predictions (SB mixed, SB urban (blue), RTS (green)). We choose a prediction threshold of 0.2 with a train/test split of 0.1 / 0.9.

The training data is very imbalanced since only a few time steps are critical. Oversampling techniques, such as SMOTE[23], allow balancing the training data. SMOTE creates additional data for training by interpolating between existing samples, and the obtained artificial data-set is then used for training. Fig. 9 shows the difference in recall, precision and accuracy for the MLP classifier when using over-sampled data. Each box-plot contains the classification results of all N-1 cases and grids combined. Recall increases when using oversampling for all thresholds - similarly, the accuracy and precision decrease as expected. The absolute number of FN predictions decrease by 10.3 % (SB mixed), 33.01 % (SB urban), and 46.1 % (RTS) for a prediction threshold of 0.2. However, the number of FP predictions increase by 10.77 %, 23.7 %, and 27.1 % respectively. Note that for each FP prediction, an additional power flow calculation for verification is needed.

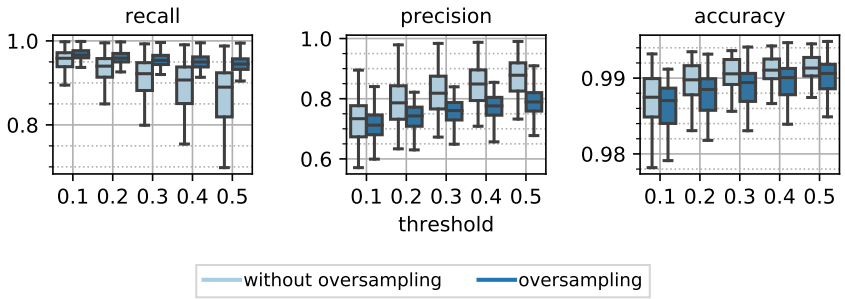

FIGURE 9 MLP training with and without oversampling.

4.4 | Comparison of MLP Regression and Classification

To compare the classification and regression method, we categorize the results of the regression method. We define a time step as critical for each (predicted) line loading value above a threshold of the max. loading limit I_{limit}. This is similar to setting a lower threshold for the classification. Fig. 10 shows the direct comparison of the MLP regression and classification methods in terms of recall, precision and accuracy. We use the oversampling method for the classification, since it showed the highest recall values. The data is obtained by a random train-test split of the time series of one year. The trained regressor has a much higher recall than the classifier, even when we use oversampling for the classifier. Recall of the regressor is close to one when setting the prediction limit to a value of $0.94 \cdot I_{\text{limit}}$ for the three test cases. At this threshold, nearly no false negative predictions are made and more than 99 % of critical time steps are identified correctly. However, precision drops to low values in that case and

accuracy decreases to mean values of less than 0.98. The precision and accuracy of the regressor significantly increase when setting a $0.98 \cdot I_{\text{limit}}$ threshold value. In this case recall drops slightly, which means that some critical time steps are not predicted correctly.

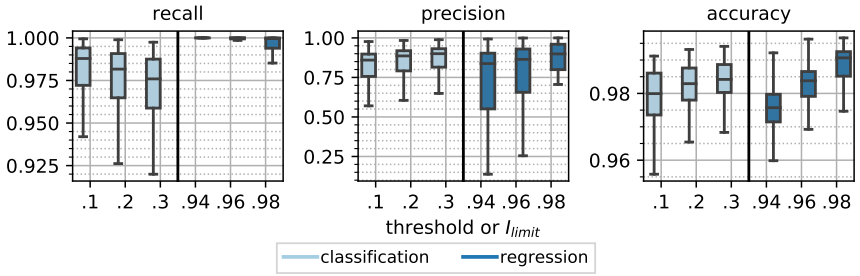

FIGURE 10 Comparison of classification and regression results for the MLP models.

We want to assess if a trained regressor/classifier can also predict unknown in-feed situations resulting when curtailing RES generation without re-training of the model. As an example, we compute the power flow results for the same year and N-1 cases but with a curtailment of 3 % of the energy generated by RES. The curtailment of renewable in-feed to reduce investments in the grid infrastructure is suggested by federal law in Germany[26]. MLP models are trained with 10 % of the power flow results *without* curtailed generation. We then use the trained MLP to predict the critical system states or line loadings based on the curtailed real power values as inputs. Fig. 11 shows a normalized sorted annual curve of these real power values. The real power inputs without curtailment are used for training, where the inputs with 3 % curtailment are used for prediction.

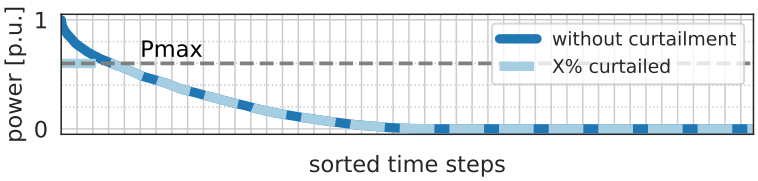

FIGURE 11 Sorted annual curve of the P inputs with and without 3 % curtailment. The P inputs without curtailment are used for training; the inputs with 3 % curtailment are used for prediction.

Fig. 12 compares the prediction results when testing with the curtailed time series. Recall increases for the classifier in comparison to previous results (see Fig. 10), since less time steps are critical in the test data set due to the curtailed generation. However, precision and accuracy decrease for the classification of time steps. The classifier anticipates the impact of the curtailed generation only to some extend. The regression method shows better results when predicting the curtailed results, since the performance metrics do not decrease as much.

Detailed comparisons are listed in Table 2. The regression method has lower FN and FP values for the exemplary thresholds of 0.2 (classifier) and $0.96 \cdot I_{\text{limit}}$ (regressor). These values result in a higher accuracy, lower false positive rates (FPR)s, and lower false negative rates (FNR)s. The FNR is equal to the share of critical time steps that could not be identified. The FPR is equal to share of mislabelled uncritical time steps and increases the computational time. An additional power flow calculation to validate the prediction is needed for each false positive prediction. Of all critical time steps and N-1 case predictions, between 0.0 - 0.48 % are not correctly identified by the regressor. This is about half the amount of the classifier. The FPRs of the regressor are between 2.01 - 3.14 % in comparison to 2.41 - 8.14 % of the classifier.

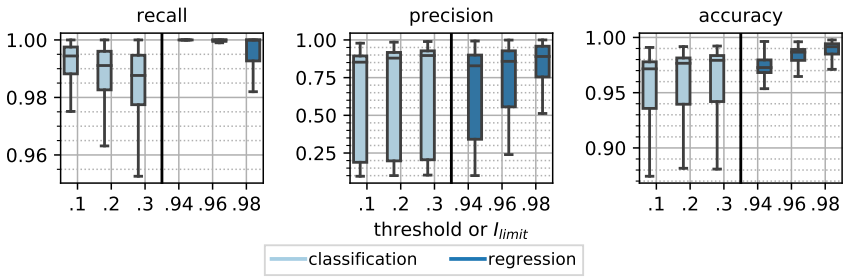

FIGURE 12 Comparison of classification and regression results for the MLP models on untrained data.

TABLE 2 Direct comparison of MLP regression and classification results on untrained data. Classification threshold = 0.2, regression threshold = $0.96 \cdot I_{\text{limit}}$. (* = lower is better, ** = higher is better)

	SB mixed	SB urban	RTS
FN classification*	3,845	5,209	460
FN regression*	1,986	2,967	0
FP classification*	45,617	70,707	552,500
FP regression*	40,854	45,446	213,010
correct classification**	2,263,178	2,657,204	6,403,968
correct regression**	2,269,800	2,684,707	6,743,918
total classification	2,312,640	2,733,120	6,956,928
total regression	2,312,640	2,733,120	6,956,928
FPR classification*	2.41 %	3.13 %	8.14 %
FPR regression*	2.15 %	2.01 %	3.14 %
FNR classification*	0.92 %	1.11 %	0.27 %
FNR regression*	0.48 %	0.63 %	0.00 %
accuracy classification**	97.86 %	97.22 %	92.05 %
accuracy regression**	98.15 %	98.23 %	96.94 %

4.5 | Comparison of Timings

Table 3 lists the time needed to compute power flow (PF) results, including the base case and N-1 cases without parallel computing. For the SimBench cases, with the 15 min resolution time series, the power flow calculation times are between 2.29 and 2.81 hours for the used hardware. For RTS, it takes nearly 8 hours to compute these results, due to the higher resolution of the time series (5 min). The training of the MLP takes 10-20 s for each N-1 case and 11-22 min in total. Prediction times are much lower with a few hundred milliseconds per N-1 case and 10-20 s in total. As already shown in Fig. 4 and 7 the regression as well as the classification method should be trained with at least 10 % of power flow results from all time steps and N-1-cases. In total, the overall time needed for the regression and classification method is dominated by the time needed to compute the training data. The overall time can be reduced by using parallel computing for every N-1 case.

5 | TRAINING DATA FROM SCENARIO GENERATOR

An alternative to using time series for training is to generate training data with a scenario generator[16]. An information-rich data-set enables the ANN to interpolate between the trained scenarios and to estimate the system variables with high accuracy. We consider three different parameters regarding the bus power injections with the scenario generator: the load, RES generator power, and the variation of fossil-fuelled power plant outputs. A scenario consists of a tuple of the scaled values for these three types with ranges between 0 % and 100 % of their maximum power. The power values are independent of each other. Thus, we

TABLE 3 Time needed to compute power flow (PF) results for training and prediction times of the MLP regressor and classifier.

	SB mixed	SB urban	RTS
$(N+1) \cdot T$ PF results [s]	8,244	10,116	28,620
10 % of PF results [s]	822	1,014	2,863
regressor training avg. [s]	660	780	660
regressor prediction avg. [s]	6.6	7.8	16.5
classifier training avg. [s]	660	936	1,320
classifier prediction avg. [s]	16.5	20.4	19.8

scale all units are individually with Gaussian noise to account for variability among the individual units of the same type. We generate the same number of training samples with the scenario generator as we have used for training with the time series data, e.g., 10 % of the time series length. Fig. 13 shows results when using the scenario generator and 10 % of the time series results for training. The maximum error of the voltage prediction significantly decreases when using the scenario training for all grids. The maximum line loading prediction error is rather constant for the SimBench grids and decreases only for the RTS case. When regarding the mean errors, the Figure shows that it increases except for the voltage predictions in the RTS case.

FIGURE 13 Training of the MLP architecture with scenario generator data and from time series data.

The reduction of the max. errors and the increase of the mean errors can be explained by the similarities of the training data set to the test data set. The majority of samples in the time series training data set is similar to the test data, which is the remaining part of the time series. This similarity results in a low mean prediction error. However, the training data set contains fewer outliers, which increases the maximum error. The scenario generator creates a more balanced training set with fewer outliers that equally distributed. This comparison shows that the ANN architecture generalises well from training data of the scenario generator and that no time series are necessarily needed for training. Such a trained model can be used in live operation to analyse contingency cases in a very short time since the prediction takes only a few milliseconds for each N-1 case.

6 | CONCLUSION

We have shown for three test cases that different machine learning algorithms can predict bus voltages and lines loading results. The prediction and training times are much shorter in comparison to the time needed to compute the power flow results. In all comparisons, the MLP architectures have shown the highest prediction accuracy. The XGB classifier has shown good results to identify critical time steps. All other tested regressors and classifiers were not as accurate, did not improve with more training data, or needed much more time to predict results. Training and prediction times for the sklearn MLP regressor and classifier were similar. Since the classification of time steps in "critical" and "uncritical" was not faster and also less accurate than the

regression method, we recommend using the MLP regressor to predict the critical contingency states. Another advantage is that bus voltage magnitudes V_m and line loadings $I_{\%}$ are outputs of the multi-variable prediction instead of binary classification. We noted that the mean and maximum prediction errors decreased with more training data, but also that the majority of time is needed to calculate these training inputs (the power flow results). We found an acceptable trade-off between calculation time and prediction error by selecting 10 % of all time steps for training. This resulted in mean errors of 1-2 % of line loading, and voltage magnitude predictions for the MLP regressor. The maximum error was in the range between 10-20 % for line loading, and around 0.5 % for voltage magnitudes. An alternative training method with scenario generator data shows that the MLP can generalize well.

The use of the prediction method is manifold. In power system planning, the method allows predicting multiple future grid states to evaluate losses or predict contingency situations when integrating RES. In live operation, N-1 security states can be assessed in seconds by using the trained ML model as a surrogate. High security margins can be considered by using lower prediction thresholds as shown. If multiple future scenarios and thus time series are to be analyzed, it might be rather acceptable to have a higher prediction error than longer calculation times since future scenarios are uncertain by definition. The final tolerable error in practice depends on a decision by the grid planner. As a general rule, we recommend considering at least a security margin in the height of the shown maximum errors in live operation as well as in planning.

The prediction accuracy of ML models strongly depends on the available training data. Further research is needed to reduce the number of false negative predictions in imbalanced data sets. Different oversampling and undersampling methods could be tested to reduce the number of these outliers in the training set. Additionally, a combination of training data from the scenario generator and time series could improve the results. For the training of the machine learning algorithms, we used a random train/test split. Since the data is imbalanced and times of high line loadings/voltage magnitudes are correlated with high in-feed, a time step selection based on the input data histogram could increase prediction accuracy.

7 | ACKNOWLEDGMENTS

The research is part of the project "SpinAI" and funded by the German Federal Ministry for Economic Affairs and Energy (funding number 0350030B). The authors are solely responsible for the content of this publication. The authors have no conflict of interest to declare.

References

1. Kays J, Rehtanz C. Planning process for distribution grids based on flexibly generated time series considering RES, DSM and storages. *IET Generation, Transmission & Distribution* 2016; 10(14): 3405–3412. doi: 10.1049/iet-gtd.2015.0825

2. Schäfer F, Menke JH, Marten F, Braun M. Time Series Based Power System Planning Including Storage Systems and Curtailment Strategies. *CIRED, 25th International Conference on Electricity Distribution, Madrid* 2019.

3. Schäfer F, Menke J, Braun M. Contingency Analysis of Power Systems with Artificial Neural Networks. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* 2018: 1-6. doi: 10.1109/SmartGridComm.2018.8587482

4. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press . 2016.

5. Krauss C, Do XA, Huck N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research* 2017; 259(2): 689–702. doi: 10.1016/j.ejor.2016.10.031

6. Fallah S, Deo R, Shojafar M, Conti M, Shamshirband S. Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions. *Energies* 2018; 11(3): 596. doi: 10.3390/en11030596

7. Yang M, Lin Y, Han X. Probabilistic Wind Generation Forecast Based on Sparse Bayesian Classification and Dempster–Shafer Theory. *IEEE Transactions on Industry Applications* 2016; 52(3): 1998–2005. doi: 10.1109/tia.2016.2518995

8. Maghrabi H, Refaee JA, Mohandes M. Contingency analysis of bulk power system using neural networks. *POWERCON '98, 1998 International Conference on Power System Technology* 1998: 1251–1254. doi: 10.1109/ICPST.1998.729286

9. Aparaschivei ED, Ivanov O, Gavrilas M. Load flow estimaton in electrical systems using artificial neural networks. *2012 International Conference and Exposition on Electrical and Power Engineering* 2012. doi: 10.1109/icepe.2012.6463917

10. Tomin NV, Kurbatsky VG, Sidorov DN, Zhukov AV. Machine Learning Techniques for Power System Security Assessment. *IFAC-PapersOnLine* 2016; 49(27): 445–450. doi: 10.1016/j.ifacol.2016.10.773

11. Donnot B, Guyon I, Schoenauer M, Marot A, Panciatici P. Fast Power system security analysis with Guided Dropout. *arXiv:1801.09870* 2018.

12. Canyasse R, Dalal G, Mannor S. Supervised learning for optimal power flow as a real-time proxy. *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)* 2017. doi: 10.1109/isgt.2017.8086083

13. Duchesne L, Karangelos E, Wehenkel L. Machine learning of real-time power systems reliability management response. *2017 IEEE Manchester PowerTech* 2017. doi: 10.1109/ptc.2017.7980927

14. Duchesne L, Karangelos E, Wehenkel L. Using Machine Learning to Enable Probabilistic Reliability Assessment in Operation Planning. *2018 Power Systems Computation Conference (PSCC)* 2018. doi: 10.23919/pscc.2018.8442566

15. Cremer JL, Konstantelos I, Strbac G. From Optimization-Based Machine Learning to Interpretable Security Rules for Operation. *IEEE Transactions on Power Systems* 2019; 34(5): 3826–3836. doi: 10.1109/tpwrs.2019.2911598

16. Menke JH, Bornhorst N, Braun M. Distribution system monitoring for smart power grids with distributed generation using artificial neural networks. *International Journal of Electrical Power & Energy Systems* 2019; 113: 472–480. doi: 10.1016/j.ijepes.2019.05.057

17. Wu L, Gao J, Venayagamoorthy GK, Harley RG. On Artificial Intelligence Approaches for Contingency Analysis in Power System Security Assessment. *2018 IEEE Power & Energy Society General Meeting (PESGM)* 2018. doi: 10.1109/pesgm.2018.8585758

18. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011; 12: 2825–2830.

19. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2016: 785–794. doi: 10.1145/2939672.2939785

20. Thurner L, Scheidler A, Schäfer F, et al. pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems* 2018; 33(6): 6510–6521. doi: 10.1109/tpwrs.2018.2829021

21. Konstantelos I, Sun M, Tindemans SH, Issad S, Panciatici P, Strbac G. Using Vine Copulas to Generate Representative System States for Machine Learning. *IEEE Transactions on Power Systems* 2019; 34(1): 225–235. doi: 10.1109/tpwrs.2018.2859367

22. Verband der Elektrotechnik . *VDE-AR-N 4121 General principles for the planning of 110 kV networks. Released 2018-04*. No. VDE-AR-N 4121VDE Verlag . 2018.

23. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 2002; 16: 321–357. doi: 10.1613/jair.953

24. Fraunhofer IEE . *SimBench - Benchmark data set for grid analysis, grid planning and grid operation management*. https://simbench.de/en . 2019. Accessed: 2020-08-17.

25. Grid Modernization Lab Consortium . *Reliability Test System*. github.com/GridMod/RTS-GMLC . 2020. Accessed: 2020-08-17.

26. German Federal Office of Justice . *Renewables Energy Act (Gesetz für den Ausbau erneuerbarer Energien) Juli 2014* . last changed on 20. Nov . 2019.

How to cite this article: F. Schäfer, JH. Menke, and M. Braun (2020), Evaluating Machine Learning Models for the Fast Identification of Contingency Cases, *Applied AI Letters.*, *2020;XXX*.