IET Smart Grid

Special Issue: Machine Learning in Power Systems

IET Journals
The Institution of Engineering and Technology

Real-time stability assessment in smart cyber-physical grids: a deep learning approach

eISSN 2515-2947
Received on 8th July 2019
Revised 12th December 2019
Accepted on 16th January 2020
E-First on 27th May 2020
doi: 10.1049/iet-stg.2019.0191
www.ietdl.org

Farzad Darbandi[1], Amirreza Jafari[1], Hadis Karimipour[2] ✉, Ali Dehghantanha[3], Farnaz Derakhshan[1], Kim-Kwang Raymond Choo[4]

[1]*Electrical and Computer Engineering Department, University of Tabriz, Tabriz, Iran*
[2]*School of Engineering, University of Guelph, Guelph, Canada*
[3]*School of Computer Science, University of Guelph, Guelph, Canada*
[4]*Department of Information Systems and Cyber Security, the University of Texas at San Antonio, Texas, USA*
✉ *E-mail: hkarimi@uoguelph.ca*

Abstract: The increasing coupling between the physical and communication layers in the cyber-physical system (CPS) brings up new challenges in system monitoring and control. Smart power grids with the integration of information and communication technologies are one of the most important types of CPS. Proper monitoring and control of the smart grid are highly dependent on the transient stability assessment (TSA). Effective TSA can provide system operators with insightful information on stability statuses and causes under various contingencies and cyber-attacks. In this study, a real-time stability condition predictor based on a feedforward neural network is proposed. The conjugate gradient backpropagation algorithm and Fletcher–Reeves updates are used for training, and the Kohonen learning algorithm is utilised to improve the learning process. By real-time assessment of the network features based on the minimum redundancy maximum relevancy algorithm, the proposed method can successfully predict transient stability and out of step conditions for the network and generators, respectively. Simulation results on the IEEE 39-bus test system indicate the superiority of the proposed method in terms of accuracy, precision, false positive rate, and true positive rate.

1 Introduction

With the integration of communication technologies and advanced metering infrastructures in the traditional power system, the smart power grid has been recognised as an important form of cyber-physical system (CPS). Smart grids play an important role in reliable and efficient energy production and management [1]. However, its interoperability and dependency on communications networks increase the system vulnerability to cyber-attacks and intelligent fault and disturbances [2]. Since failures in the critical CPSs, such as smart grids, may lead to catastrophic events, it is highly important to investigate the effect of faults and disturbances on them. Therefore, real-time monitoring and control of smart power grids are highly important to mitigate the effect of the intentional faults and cyber disturbances to enhance the security margin of the system.

Lack of system awareness is the main reason for several major North American blackouts [3, 4]. Under the faults or random disturbances, the synchronous generators in a power system lose their synchronism, which results in transient instability or out of step (OOS) condition. OOS condition results in power swing among the generators, which may lead to the unwanted islanding, outages, and extensive blackouts [5]. Therefore, the early prediction and mitigation of the OOS condition through transient stability assessment (TSA) are paramount for proper system control and monitoring.

TSA techniques are categorised into two main groups: (i) instability detection after OOS condition and (ii) instability prediction before faults occurrence. The detection techniques are not functional for dynamic networks as they detect instability after fault clearance. Common methods in this area include the use of impedance relays [6], current assessment [7], and wavelet transform [8]. The main drawbacks of these methods are the possibility of misidentification between fault and OOS condition, high latency in the detection of OOS, and high probability of instability and unwanted outages.

The prediction methods try to anticipate future stability conditions according to the current behaviour of the system. These techniques are classified into four categories:

(i) Time-domain techniques [9], which formulate the TSA problem as non-linear differential equations. These methods usually result in poor performance for dynamic systems due to high dependency on the system's parameters, and heavy computational burden.
(ii) Trajectory characteristics approach, which assesses variations of the voltage and current phasors. These methods usually implement Lyapunov exponents and phase-plane assessment theories [10, 11].
(iii) Direct methods that solve the TSA problem based on the precise assessment of energy functions [12]. These methods have various disadvantages, such as constrained scalability, the complexity of models, and low-prediction speed.
(iv) Intelligent techniques utilise the correlations between different features in the system to predict instability condition. There are limited works in the literature, including decision tree (DT) [13], support vector machine [14], extreme learning machine [15], and fuzzy knowledge systems [16, 17]. Other intelligent techniques are developed to predict stability conditions based on the velocity-acceleration criterion [18], state estimation [19], and Bayesian technique [20]. Artificial neural networks (ANNs) are among the fastest and most accurate intelligent techniques for large data analysis. There is limited work on the application of data analytics and ANN-based techniques for TSA. In [21, 22], a new method based on the cascaded convolutional neural network (CNN) is developed to predict the stability condition of the power system. Chen *et al*. [23] proposed a transient stability (TS) prediction method based on the extreme gradient boosting approach.

Although proposed techniques have their own advantages, most of them use random scenarios generation by considering random fault duration, load level, and fault location (based on discrete values), which decreases the robustness of the algorithm. In addition, the

majority of the existing works only consider post fault data to predict stability conditions, which results in low accuracy and lack of robustness for high-dimensional and complex applications such as TSA in the power system. In addition, most of these methods randomly extract inputs from features, which will significantly affect the prediction and detection accuracy. Most importantly, all the existing techniques in the literature predict the stability condition after fault clearance, which will waste lots of critical time, which can be used for proper protection actions. Moreover, the majority of the proposed techniques are only tested on a single fault type.

This study proposes an online TS prediction technique using a feedforward neural network (FNN). Despite the existing techniques in the literature, the proposed method predicts the TS condition before the fault clearance, which significantly saves time for control actions by the operator. A modified version of FNN using the conjugate gradient algorithm, Fletcher–Reeves method, and Kohonen learning algorithm is developed. The minimum redundancy maximum relevancy (MRMR) feature selection algorithm is utilised to extract the optimal set of features. To evaluate the performance of the proposed method, 2000 different scenarios are generated for training and testing. Like previous works [24], true positive, false positive (FP), true negative (TN), and false negative (FN) metrics were used with used ten-fold cross-validation to evaluate the proposed system. Precision, recall, F-measure, accuracy, and Matthews's correlation coefficient (MCC) were calculated as per (1)–(5)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F} - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{5}$$

We calculated MCC to show that our predictions are far from random guessing. In random guessing, precision, recall, and F-measure are >0.5, while MCC is around 0 [25].

The results show that the proposed method outperforms the existing techniques in the literature in terms of prediction accuracy and speed. The proposed method achieves an average F-measure of 0.996 with a false positive rate (FPR) of 0.002 for different case studies and functions. The findings also show that the stability condition and time remained to instability can be predicted with the accuracy and precision of 98.9% and 98.6%, respectively.

In summary, the main contributions are summarised as follows:

- Proposing a modified FNN for real-time assessment of TS with prediction capability before fault clearance.
- Comprehensiveness assessment through numerous fault scenario simulations, including various fault duration, fault location, and load profile intervals.
- Localising the effect of the fault by predicting the critical generators.
- Increasing the efficiency of the protection step by providing extra time to the operator through estimating the remaining time to instability.
- Real-time assessment capability and fast performance by predicting stability condition before fault clearance.

The rest of this paper is organised as follows. In Section 2, the proposed methodologies, including the modified neural network, feature selection, and extraction techniques, and training model, are explained. Simulation results and case studies are discussed in Section 3, followed by the conclusion in Section 4.

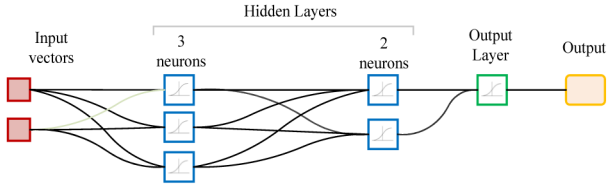

Fig. 1 *General scheme of the FNN*

2 Proposed FNN-based transient stability predictor

Generally, pattern recognition techniques attempt to predict the current or future condition of a system by evaluating its symptoms and based on historical data. One of the most common pattern recognition techniques is the FNN. FNN has three main layers, including input, hidden, and output layers. Each layer includes a number of units named 'neuron'. Neurons take inputs from previous layers and compute activation value. FNN approximates target value based on input vectors and using different learning algorithms. The general scheme of FNN is shown in Fig. 1. In this work, a modified FNN-based TS predictor is proposed, which utilises one training algorithm for the input layer and another one for hidden and output layers.

2.1 Proposed modified FNN algorithm

In the proposed method, the conjugate gradient algorithm combined with the Fletcher–Reeves update method is used for the training of hidden and output layers. It is an iterative process where the learning rate changes in each iteration. Fletcher–Reeves method is used for searching in the conjugate gradient directions that have faster convergence and lower memory requirements compared to the gradient descent methods.The algorithm starts with the searching process in the steepest descent direction (negative of the gradient) as follows:

$$P_0 = -g_0 \tag{6}$$

In the next iterations, the searching direction is determined in combination with the new steepest descent direction and the previous iteration direction. This procedure can be described as follows:

$$\boldsymbol{P}_k = -\boldsymbol{g}_k + \beta_k^f \boldsymbol{P}_{k-1} \tag{7}$$

$$\boldsymbol{g}_k = \nabla_x f(x) \tag{8}$$

where $\boldsymbol{P}_k$ and $\boldsymbol{g}_k$ are related to searching direction vector and gradient vector of the kth iteration, respectively. β_k^f is the searching parameter related to the Fletcher–Reeves method that is calculated according to

$$\beta_k^f = \frac{\boldsymbol{g}_k^t \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^t \boldsymbol{g}_{k-1}} \tag{9}$$

where $\boldsymbol{g}_k^t$ is the transposed gradient vector of the intended function at kth iteration. β_k^f is determined as the ratio of the norm-squared value of the gradient vector in the current iteration to the norm-squared value of the gradient vector in the previous iteration.

In the next stage, the optimal displacement distance of the algorithm in the determined searching direction is calculated as follows:

$$x_{k+1} = x_k + \alpha_k \boldsymbol{P}_k \tag{10}$$

where α_k is the step length of the kth iteration that is determined by a line searching process in the specified direction in such a way that the objective function ($f(x)$) has the minimum value, according to

$$\alpha_k = \arg\min_{\alpha} f(x_k + \alpha \boldsymbol{P}_k) \tag{11}$$

In addition to the conjugate gradient method, to improve the performance of the proposed FNN, the Kohonen learning algorithm is used to train the inputs layer. The training input layer and weights by the Kohonen algorithm enhance the conjugate gradient performance on hidden and output layers. This algorithm can be formulated as follows:

$$dw = lr \times (p' - w), \qquad \text{if } a \sim = 0 \quad \text{otherwise } dw = 0 \tag{12}$$

where p and a are the input and output of the network and w is the neuron's weight. dw refers to the variation of w and lr is the learning rate of the algorithm. Other parameters, including initial learning rate, drop period, the number of hidden layers, and the maximum number of epochs, are set by trial and error. Also, to improve the network performance, input vectors are normalised for a better correlation between input and output vectors.

The sigmoid and tansig functions are also utilised for output and hidden layers, respectively. The mathematical expressions of these functions are as follows:

$$\log\text{sig}(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

$$\text{tansig}(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{14}$$

The tansig transfer function has high-gradient strength, and due to average output, around zero is suitable for data clustering and facilitates training for hidden layers. Also, because of the intrinsic characteristic of the sigmoid function, its output values are obtained near 0 or 1. Therefore, it is suitable for the output layer of classification problems with binary targets.

For training purposes, an error function should be calculated between the actual and predicted output of the network. In this work, cross-entropy with high-penalty weight for inaccurate outputs is considered. The cross-entropy (ce) function for a pair of actual (t) and predicted (y) outputs can be calculated as follows:

$$\text{ce} = -t \times \log(y) - (1 - t) \times \log(1 - y) \tag{15}$$

2.2 Feature selection and extraction

In large complex power systems, the huge number of measured features from different parts of the network suffers from the curse of dimensionality. Therefore, it is necessary to select proper features from the original data measured by PMUs such as branch active/reactive power flows, rotor angles, speeds, voltage magnitudes, and angles etc. In this work, the MRMR approach is used to select the optimal combination of the features [26]. MRMR is an efficient and fast feature, which aims to find a set of relevant and complementary features based on mutual information (MI) [27]. The basis of this technique is that if two features have an intimate connection with each other, they play a similar role in classification or prediction. Therefore, it is unnecessary to include both in selected feature sets. In this work, using the MRMR algorithm, a set of features that show the maximum relevance to the stability boundary and have minimum redundancy are selected.

The power system dataset I consist of n features $N = \{n_1, n_2, n_3 \ldots n_n\}$ derived from PMUs. For optimal features selection, the relevancy and redundancy indices are calculated among the database and target class.

2.2.1 Relevancy calculation: The relevance of the data set I to the target class T, denoted by $R(I, T)$, is measured by the mean value of total MI for all features with respect to the target class as follows:

$$R(I, T) = \frac{1}{N} \sum_{i=1}^{N} \text{MI}(\boldsymbol{n}_i, T) \tag{16}$$

where N is the total number of features, and T is the target class of the proposed FNN. The correlation between features and the target class T is defined as follows:

$$\text{MI}(\boldsymbol{n}_i, T) = \sum_{x=1}^{S} p(\boldsymbol{n}_{i,x}, T_x) \log\left(\frac{p(\boldsymbol{n}_{i,x}, T_x)}{p(\boldsymbol{n}_{i,x}) p(T_x)}\right) \tag{17}$$

where S is the number of samples that are extracted from PMUs. The parameter $\boldsymbol{n}_{i,x}$ represents the xth element of the feature vector $\boldsymbol{n}_i$, T_x is the xth element of the target class T, $p(T_x)$ and $p(\boldsymbol{n}_{i,x})$ are the marginal probability density function of T_x and $\boldsymbol{n}_{i,x}$, respectively. The joint probability distribution of the T_x and $\boldsymbol{n}_{i,x}$ is represented by $p(\boldsymbol{n}_{i,x}, T_x)$.

2.2.2 Redundancy calculation: To avoid redundancy between features, $D(n)$ illustrates discrepancies between the set of features, which can be calculated as follows:

$$D(N) = \frac{1}{N^2} \sum_{i,j=1}^{N} \text{MI}(\boldsymbol{n}_i, \boldsymbol{n}_j) \tag{18}$$

where $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ are ith and jth features and their MI can be calculated as follows:

$$\text{MI}(\boldsymbol{n}_i, \boldsymbol{n}_j) = \sum_{x=1}^{S} p(\boldsymbol{n}_{i,x}, \boldsymbol{n}_{i,y}) \log\left(\frac{p(\boldsymbol{n}_{i,x}, \boldsymbol{n}_{i,y})}{p(\boldsymbol{n}_{i,x}) p(\boldsymbol{n}_{i,y})}\right) \tag{19}$$

where $p(\boldsymbol{n}_{i,x}, \boldsymbol{n}_{i,y})$ calculates the joint probability distribution of $\boldsymbol{n}_{i,x}, \boldsymbol{n}_{i,y}$.

2.2.3 MRMR implementation: To implement the MRMR algorithm, it is essential to perform an optimisation process to achieve the optimal set of features. In our case study, two features for each generator are selected. Therefore, an incremental search procedure can be implemented instead of the optimisation process. In the first step, the relevancy term between each feature and the feed forward neural network (FFNN) output class is calculated, and the feature with the highest relevancy term value is selected as the best feature. After that, the redundancy term between the first selected feature and other features, is calculated. The features with the minimum redundancy term and maximum relevancy term are selected as the second optimal feature of the final feature set.

2.3 Offline training

For offline training, four functions are defined, which can be used to predict fault occurrence and its type using instability condition, remaining time to instability or OOS condition, and the pair of critical generators. These functions are explained in the following.

- *Fault type identifier (FTI)*: The proposed method has a hierarchical architecture, which tries to break down the problem into four sub-problems to be solved using proposed functions. At first, a FTI is proposed to detect the fault type, and then according to the detected type, the corresponding transient instability predictor (TIP) is activated and assesses the future stability condition of the smart grid. Since different faults will come with different levels of severity, this preprocessing will provide more time to the operator to take action ahead of the time.
- *TIP*: To predict the stability condition, for each fault type, a separate TIP function is trained. For each TIP function, the training scenarios are related to one specific fault.

- *Remaining time to instability estimator (RTIE)*: Once the instability condition is predicted; it is important to know how much time remained until the OOS condition happens. RTIE function is a regression function that estimates the remaining time based on the type of fault and historical data related to the effect of that fault in the behaviour of the selected features. Similar to the TIP functions, for each fault type, a separate RTIE function is trained.
- *Critical generators identifier (CGI)*: Once transient instability is predicted, the critical generator should be identified to mitigate the possibility and effect of the instability. CGI functions use the difference between the rotor angles in every pair of generators to identify those who will reach 180° before other generators. Similar to the previous functions, for each fault type, a unique CGI function is trained.

To generate input vectors, the time window method extracts samples from selected features. The length of the time interval and sampling frequency is considered as constant values, so the length of input vectors will be the same for all scenarios. The static time window which includes the entire fault duration and pre-fault times is considered for the training. With the variation of fault duration and the constant length of the static time window, the duration of pre-fault data in different scenarios will be different. This property along with the changes in the behaviour of the features in pre-fault and during fault times creates an effective index for training the network and distinction between stable and unstable modes.

2.4 Implementation and online assessment

For online implementation, the moving time window is considered where the old samples are replaced with the new one over time. As opposed to traditional methods, which wait for post fault and fault clearance data, in the proposed method, using a moving window, the samples of features under the fault condition are added to the input vectors as soon as fault starts, which can be used to predict stability condition. The length of the time window and sampling frequency is identical for both static and moving modes. Fig. 2 shows the flowchart of the online implementation of the proposed method. As seen in Fig. 2, once the phasor measurement unit (PMU) samples are received from the measurement devices in the field, the input vectors are updated by moving the time window. Refresh rates of the PMUs are 2 ms. In the first stage, fault occurrence and its type are detected by FTI function, and in the second stage based on detected fault type, and related TIP function is activated to predict the stability condition of the system. If a stable condition is predicted, the algorithm moves to the next time step. On the other hand, if an unstable condition is predicted, relative RTIE and critical generators identifier (CGI) functions will be activated to estimate remaining time to instability and group of critical generators, respectively. The execution time for the whole process is <1 ms. The elapsed time between data generation by PMUs until providing the intended decision includes both execution time and communication time. In this work, the assumption is that the data will be communicated without delay, and TSA can be performed in <2 ms (PMU refresh rate). It should be noted that even if the communication delay takes >1 ms, we can perform offline TSA prediction based on the historical data until the next sample arrives.

3 Result and discussion

To evaluate the performance of the proposed method, simulations are performed on the IEEE 39 bus test system modelled in MATLAB software. The smart grid is modelled as a multi-agent, CPS, where each of these agents includes a generator, a measurement device, a distributed control agent, and an energy storage system that can inject or absorb real power in the system. The overall sequence of the proposed technique is illustrated in Fig. 3. The number of hidden layers in the neural network is 18, which is selected based on cross-entropy error analysis. In addition, the initial value of learning rate and learning rate drop period are selected as 0.01 and 1 step, respectively. Hidden layers of the network are categorised into four groups with a specific number of neurons. The number of neurons for each group is selected based on trial and error. For hidden layers, 1st to 4th, 5th to 9th, 10th to 14th, and 15th to 18th, the number of neurons has been selected as 54, 36, 24, and 15, respectively.

The test system consists of 39 buses and ten synchronous generators with ten PMUs installed in the generator buses. Features

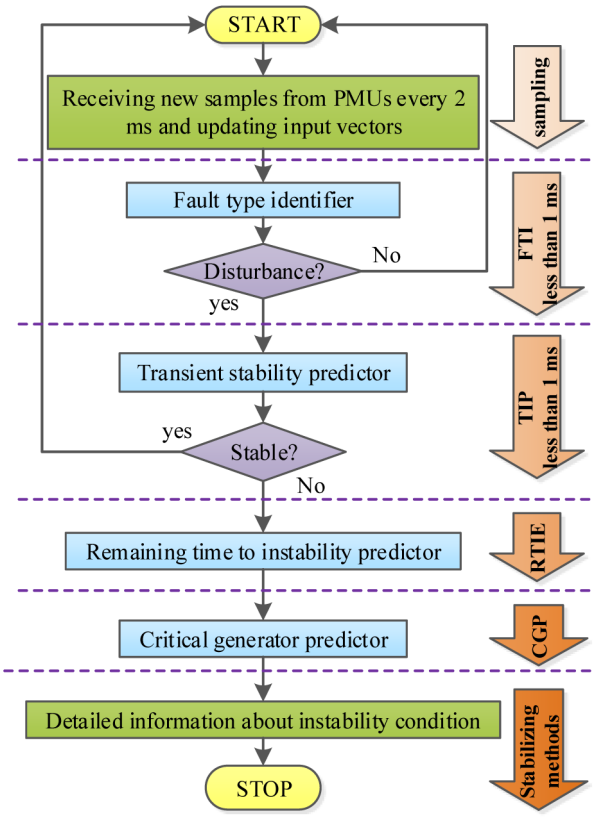

Fig. 2 *Flowchart of the proposed method*

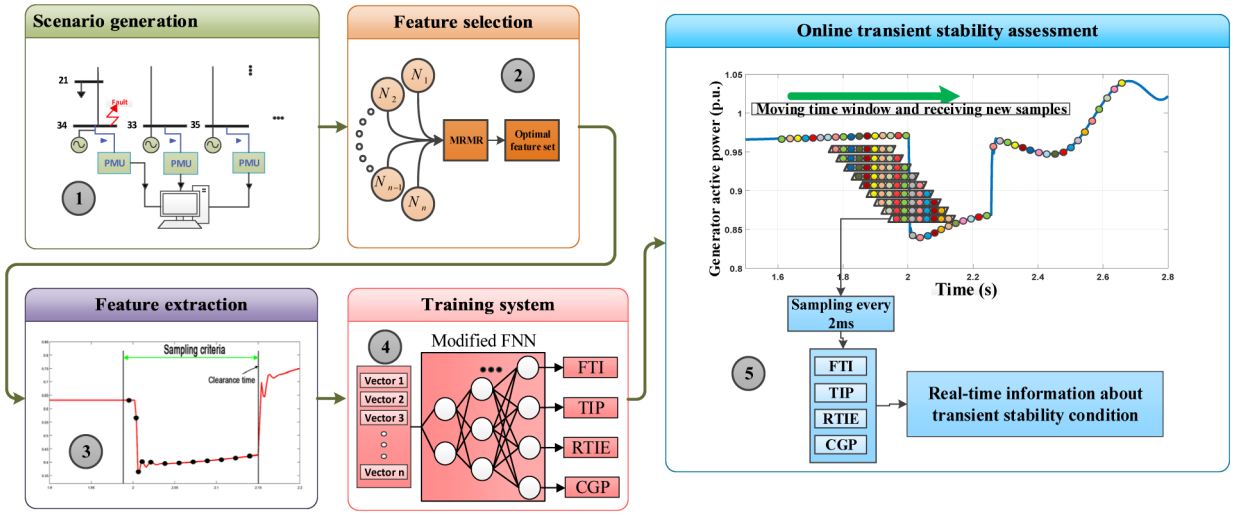

Fig. 3 *General sequence of the proposed methodology for online TSA*

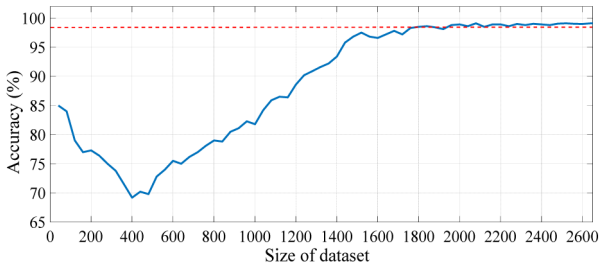

Fig. 4 *Effect of the size of the dataset on the average accuracy*

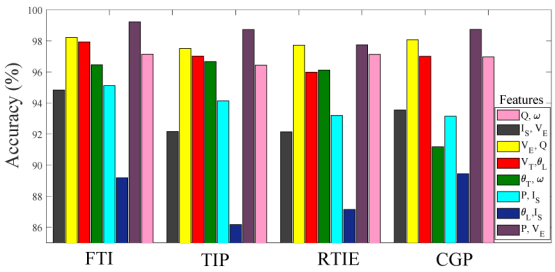

Fig. 5 *Accuracy of the proposed method for different pairs of features*

are extracted from the active power and magnitude of the excitation voltage in all generators. The length of the time window and the period of sampling are considered as 500 and 2 ms, respectively. Therefore, the number of samples in input vectors will be equal to 250. Since the maximum fault duration is 0.4 s, the length of the window is set to 0.5 s to include some samples from the initial state (pre-fault period) in the input vectors. It has been observed that larger window size will not affect the performance of the proposed algorithm since the system is under the steady-state condition before the fault, and therefore, extra samples from the pre-fault period will not have more useful information for TSA.

3.1 Scenario generation

To simulate different scenarios for the training phase, stochastic modelling of short circuit faults with a random duration between [0.06, 0.4] s is performed. Fault locations are changed along all lines in different scenarios. To model the effect of load variations, the network loads are randomly selected in the interval [0.7, 1.4] of the basic load level. Totally, 2000 scenarios are generated that include 500 scenarios for each fault type. In addition, 800 new scenarios have been generated for testing with 200 scenarios for each fault type. To validate the comprehensiveness of the dataset, the learning curve theory is used. The proposed network is trained by a different number of scenarios (40–2600 scenarios) to investigate the effect of dataset size on the training performance. The learning curve is shown in Fig. 4. As seen in the figure, by increasing the size of the dataset, at first, the training accuracy decreases and then starts to increase up to about 98% in the 2000 scenarios. It is observed that after 2000 scenarios, there are no significant changes in training accuracy. Therefore, 2000 scenarios are considered for the training process.

3.2 Performance evaluation

To evaluate the performance of the proposed method, different types of fault are simulated. In addition, fault duration, location, and load levels are stochastically generated to increases the robustness of the results. Ten-fold cross validation is performed to evaluate the proposed system. The average results are reported here. In the first step, the fault type is identified, and the relative TIP function is activated to predict the stability condition of the system. Once instability is predicted, the relative RTIE will be activated. To predict the pair of critical generators, which will result in OOS condition before other generators CGI, is used. This function localises the effect of fault by isolating critical generator and saves a lot of time for fault mitigation through proper protection schemes.

3.2.1 Feature selection: Fig. 5 reports the accuracy of the proposed method for different groups of features. Stator current (V_S), excitation voltage (V_E), terminal voltage magnitude (V_T), terminal voltage angle (θ_T), load angle (θ_L), rotor speed (ω), active power (P), and reactive power (Q) are used. By MRMR, the groups of two features (V_E and P), which result in the highest accuracy are selected. For the sake of space, the cases with the highest accuracy are reported here.

Groups of three features and higher were also tested, but the accuracy will not change significantly. Therefore, to reduce the size of data, groups of two features for each generator are selected, which results in a total of 20 features. It was observed that more features would increase the accuracy by 1%; however, it will compromise the speed of the algorithm. Increasing the number of features will significantly increase the processing time to >2 ms (refresh rate of the PMU). Therefore, to maintain the accuracy and speed at the same time, we employed the MRMR method to select the optimum feature set with minimum redundancy and maximum relevancy.

3.2.2 Accuracy and precision analysis: Figs. 6*a* and *b* depict the performance of the proposed method in terms of average accuracy and precision with varying numbers of hidden layers. The results are plotted for ten PMUs. As seen from both figures, the proposed methods identify the fault type with the maximum accuracy and precision of 99.1 and 98.5%.

It predicts the stability condition and time remained to instability with the maximum accuracy and precision of 98.9 and 98.6%. The pair of critical generators is also identified correctly,

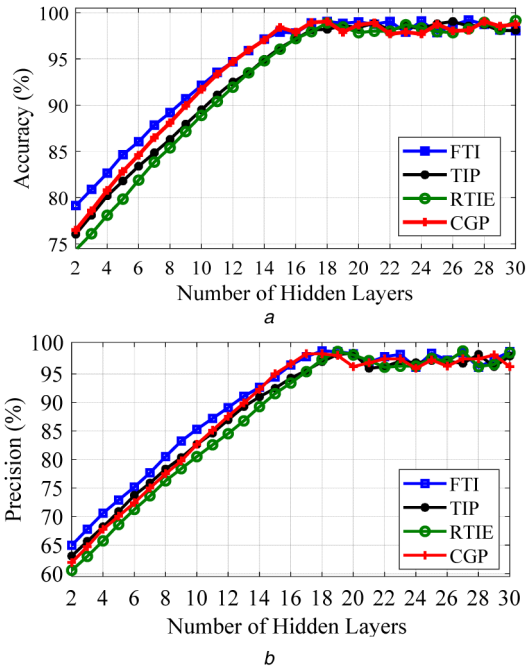

Fig. 6 *Performance evaluation for different number of hidden layers*
(a) Accuracy, ***(b)*** Precision of the proposed method for different number of hidden layers

with the maximum accuracy and precision of 98.8 and 98.4%. The results show that the performance of the proposed method enhances as the number of hidden layers increases. However, both accuracy and precision are almost constant for hidden layers of 18 and higher. This clarifies the number of hidden layers was selected properly in the first place. It should be mentioned that the average detection time is equal to 35.2% of the fault duration (equal to 65.6 ms after fault occurrence in our case study), with a maximum and minimum of 44 and 21.2%.

To investigate the effects of the number of PMUs on accuracy and precision, Figs. 7 and 8 are plotted with a varying number of PMUs. The results show that the optimum numbers of hidden layers are not sensitive to the number of PMUs. Moreover, as seen from the figure, for the increasing number of PMUs, both accuracy and precision increase. However, the amount of performance enhancement is not significant for more than ten PMUs. Considering the costs of PMU installation in the system, we must compromise between the accuracy and cost with an optimum number of PMUs. Besides that, more PMUs mean more measurements, which increase the size of the data and results in computational burden. In this work, ten PMUs are considered as the optimum number of PMUs, and all are installed in the generator buses, which are the most sensitive buses in the system.

Fig. 9 shows the average prediction accuracy with respect to the time remaining to instability/OOS condition. The stability condition is predicted with 90 and 98% accuracy in >65 and 55% of the remaining time to OOS condition, respectively. For example, for a fault with a duration of 1 s, the proposed method, on average, takes 0.55 s to predict stability condition, while for other techniques, it takes up to 1.1 s after fault occurrence. As seen from

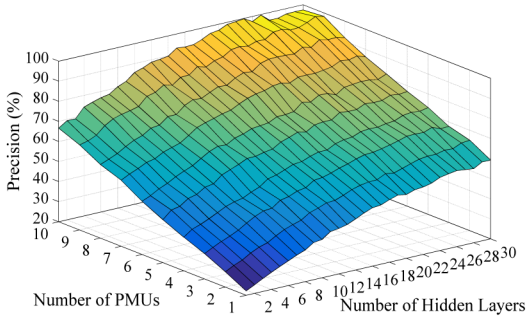

Fig. 7 *Accuracy of the TIP function for different number of hidden layers and PMUs*

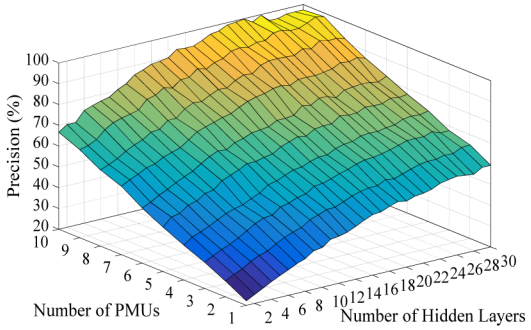

Fig. 8 *Precision of the TIP function for different number of hidden layers and PMUs*

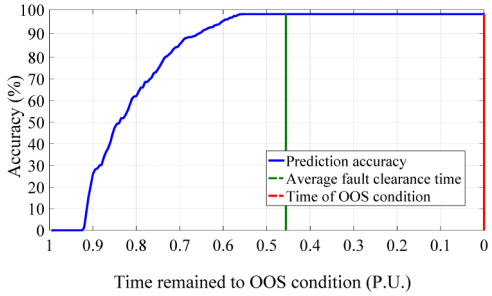

Fig. 9 *Percentage of accurate predictions of TIP per fault duration*

Table 1 Estimating accuracy and the average remaining time to OOS condition for different fault types

Fault type	Three-phase	Two-phase	Two-phase to ground	Single phase
testing accuracy, %	97.6	97.2	96.9	97.8
average estimated remaining time to OOS, s	1.01	1.16	1.19	1.28

Table 2 Comparison of prediction accuracy using different machine learning techniques

Prediction method	Accuracy, %	Prediction time	Fault type
XGBoost [23]	97.8	3 ms after F_d	three-phase
Bayesian rate (BR) [20]	91.6	100 ms after F_d	all types
DT [28]	90.3	50 ms after F_d	three-phase
CNN [22]	89.22	90 ms after F_d	three-phase
long short term memory (LSTM) [29]	98.4	30 ms after F_d	three-phase
proposed method	98.7	0.554 F_d	three-phase
	99.2	0.582 F_d	single-phase
	98.8	0.542 F_d	two-phase to ground
	98.9	0.560 F_d	two-phase
	98.9	0.559 F_d	all types

Table 3 Performance evaluation of the proposed method

Number of PMUs	FTI				TIP				RTIE				CGP			
	MCC	FM	TPR	FPR	MCC	FM	TPR	FPR	MCC	FM	TPR	FPR	MCC	FM	TPR	FPR
6	0.803	0.901	0.892	0.089	0.814	0.907	0.909	0.095	0.791	0.888	0.873	0.102	0.800	0.895	0.892	0.092
8	0.895	0.947	0.954	0.059	0.884	0.941	0.945	0.062	0.879	0.931	0.942	0.068	0.894	0.938	0.952	0.060
10	0.996	0.998	0.998	0.001	0.994	0.996	0.997	0.002	0.991	0.995	0.995	0.004	0.993	0.996	0.997	0.003
12	0.997	0.998	0.998	0.001	0.996	0.997	0.998	0.001	0.993	0.995	0.996	0.003	0.995	0.997	0.998	0.001

the figure, the presented method can predict the transient stability of all scenarios with an accuracy of 98%, before fault clearance.

The group of RTIE functions is trained with the dataset and tested on related scenarios. The testing accuracy of the proposed model and the average value of remaining time to OOS conditions have been reported in Table 1. As seen in Table 1, the remaining time to OOS has been estimated with high accuracy for all fault types by RTIE functions.

3.2.3 Comparison analysis: For further analysis, the performance of the proposed method is also compared with several machine learning techniques that are already used for TS prediction. Table 2 shows a summary of the results.

As the results show, the proposed method outperforms all techniques in terms of prediction accuracy and time. It should be noted that other techniques are mainly tested only on one type of fault, which results in higher accuracy compared to multiple types of faults, which are simulated in our case studies. It should be also noted that the average prediction time for the proposed method is 55.95% of the fault duration (F_d) as opposed to other methods that predict the OOS condition after fault clearance. This unique property significantly reduces the chance of instability by saving a lot of time for the operator to take control actions ahead of the time. In addition, in real power system operation, when the fault starts, even before fault clearance, the operator starts preventive action, e.g. opens relays and switches to be prepared in case faults results in instability. Stability prediction before fault clearance will save a huge amount of cost by avoiding preventive actions that the operator should take when he/she is not aware of the possibility of the instability due to a fault.

Mathew correlation (MCC) and F-measurer (FM) values, along with true positive rate (TPR) and FPR for three different numbers of PMUs, are reported in Table 3. All values are rounded to three decimal points. As the results show, the performance of the proposed method increases significantly as the number of PMUs increases. However, for more than ten PMUs, the changes are not significant. Again, this is because ten PMUs are equal to the number of generator buses, which are the sensitive buses in the system and their behaviour changes the performance of the system significantly. The results also indicate that for all functions MCC and F-measures are very close to 1, which proves that the proposed method is scalable, and its performance will not be affected by the size of data.

4 Conclusion

This study proposes an online TSA technique based on FNN that predicts the post-contingency stability condition of the power system based on the real-time sampling of the network parameters. The conjugate gradient backpropagation algorithm and the Fletcher–Reeves updates method are utilised as training and searching algorithms. In addition, the Kohonen learning algorithm is used to improve the learning of inputs and layers weights. The MRMR algorithm is used to select the optimal combination of training features. The proposed method is tested on the IEEE 39 bus system and compared with the existing techniques in the literature. The results verify the efficiency and superiority of the proposed method in terms of prediction accuracy and response speed. The proposed methods identify the fault type with 99.1% accuracy and 98.5% precision. It predicts the stability and time remained to OOS with the accuracy and precision of 98.9 and 98.6%, respectively. The pair of critical generators is also identified with the accuracy and precision of 98.8 and 98.4%, respectively.

5 References

[1] Ayar, M., Obuz, S., Trevizan, R.D., *et al.*: 'A distributed control approach for enhancing smart grid transient stability and resilience', *IEEE Trans Smart Grid*, 2017, **8**, (6), pp. 3035–3044

[2] Yang, Q., Li, D., Lin, J.: 'Toward data integrity attacks against optimal power flow in smart grid', *IEEE Internet Things J*, 2017, **4**, (5), pp. 1726–1738

[3] Karimipour, H., Dehghantanha, A., Parizi, R.M., *et al.*: 'A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids', *IEEE Access*, 2019, **7**, pp. 80778–80788

[4] Yang, X., Zhao, P., Zhang, X., *et al.*: 'Toward a Gaussian-mixture model-based detection scheme against data integrity attacks in the smart grid', *IEEE Internet Things J.*, 2018, **4**, (1), pp. 147–161

[5] Karimipour, H., Dinavahi, V.: 'Robust massively parallel dynamic state estimation of power systems against cyber-attack', *IEEE Access*, 2018, **6**, pp. 2984–2995

[6] Patel, U.J., Chothani, N.G., Bhatt, P.J.: 'Distance relaying with power swing detection based on voltage and reactive power sensitivity', *Int. J. Emerg. Electr. Power Syst.*, 2016, **17**, (1), pp. 27–38
[7] Hashemi, S.M., Sanaye-Pasand, M.: 'Current-based out-of-step detection method to enhance line differential protection', *IEEE Trans. Power Deliv.*, 2019, **34**, (2), pp. 448–456
[8] Salimian, M.R., Haghjoo, F., Aghamohammadi, M.R.: 'Out of step detection and protection using online WT', *Int. Electr. Eng. J.*, 2013, **4**, (4), pp. 1133–1139
[9] Zadkhast, S., Jatskevich, J., Vaahedi, E.: 'A multi-decomposition approach for accelerated time-domain simulation of transient stability problems', *IEEE Trans. Power Syst.*, 2014, **30**, (5), pp. 2301–2311
[10] Wei, S., Yang, M., Qi, J.*, et al.*: 'Model-free MLE estimation for online rotor angle stability assessment with PMU data', *IEEE Trans. Power Syst.*, 2017, **33**, (3), pp. 2463–2476
[11] Su, F., Zhang, B., Yang, S.*, et al.*: 'Power system first-swing transient stability detection based on trajectory performance of phase-plane'. 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conf. (APPEEC), Xi'an, China, 2016, pp. 2448–2451
[12] Chiang, H.-D.: '*Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications*' (John Wiley & Sons, USA, 2011)
[13] Siryani, J., Tanju, B., Eveleigh, T.J.: 'A machine learning decision-support system improves the internet of things' smart meter operations', *IEEE Internet Things J*, 2017, **4**, (4), pp. 1056–1066
[14] Moulin, L.S., Da Silva, A.P.A., El-Sharkawi, M.A.*, et al.*: 'Support vector machines for transient stability analysis of large-scale power systems', *IEEE Trans. Power Syst.*, 2004, **19**, (2), pp. 818–825
[15] Zhang, R., Xu, Y., Dong, Z.Y.*, et al.*: 'Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system', *IET Gener. Transm. Distrib.*, 2015, **9**, (3), pp. 296–305
[16] Kamwa, I., Samantaray, S.R., Joós, G.: 'On the accuracy versus transparency trade-off of data-mining models for fast-response PMU-based catastrophe predictors', *IEEE Trans Smart Grid*, 2011, **3**, (1), pp. 152–161
[17] Karimipour, H., Leung, H.: 'Relaxation-based anomaly detection in cyber-physical systems using ensemble Kalman filter', *IET Cyber-Phys. Syst., Theory Appl.*, 2019, **3**, pp. 29–38
[18] Salimian, M.R., Aghamohammadi, M.R.: 'Intelligent out of step predictor for inter area oscillations using speed-acceleration criterion as a time matching for controlled islanding', *IEEE Trans Smart Grid*, 2018, **9**, (4), pp. 2488–2497
[19] Karimipour, H., Dinavahi, V.: 'Extended Kalman filter based massively parallel dynamic state estimation', *IEEE Trans Smart Grid*, 2015, **6**, (3), pp. 1539–1549
[20] Zare, H., Alinejad-Beromi, Y., Yaghobi, H.: 'Intelligent prediction of out-of-step condition on synchronous generators because of transient instability crisis', *Int. Trans. Electr. Energy Syst.*, 2019, **29**, (1), p. e2686
[21] Yan, R., Geng, G., Jiang, Q.*, et al.*: 'Fast transient stability batch assessment using cascaded convolutional neural networks', *IEEE Trans. Power Syst.*, 2019, **34**, pp. 2802–2813
[22] Gupta, A., Gurrala, G., Sastry, P.S.: 'An online power system stability monitoring system using convolutional neural networks', *IEEE Trans. Power Syst.*, 2019, **34**, (2), pp. 864–872
[23] Chen, M., Liu, Q., Chen, S.*, et al.*: 'XGBoost-based algorithm interpretation and application on post-fault transient stability Status prediction of power system', *IEEE Access.*, 2019, **7**, pp. 13149–13158
[24] HaddadPajouh, H., Dehghantanha, A., Khayami, R.*, et al.*: 'A deep recurrent neural network based approach for internet of things malware threat hunting', *Future Gener. Comput. Syst.*, 2018, **85**, pp. 88–96
[25] Mohammadi, S., Mirvaziri, H., Ahsaee, M.G.*, et al.*: 'Cyber intrusion detection by combined feature selection algorithm', *J. Inf. Secur. Appl.*, 2019, **44**, pp. 80–88
[26] Unler, A., Murat, A., Chinnam, R.B.: 'mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification', *Inf. Sci.*, 2011, **181**, (20), pp. 4625–4641
[27] Homayoun, S., Dehghantanha, A., Ahmadzadeh, M.*, et al.*: 'DRTHIS: deep ransomware threat hunting and intelligence system at the fog layer', *Future Gener. Comp. Syst.*, 2019, **90**, pp. 94–104
[28] Aghamohammadi, M.R., Abedi, M.: 'DT based intelligent predictor for out of step condition of generator by using PMU data', *Int. J. Electr. Power Energy Syst.*, 2018, **99**, pp. 95–106
[29] Yu, J.J. Q., Hill, D.J., Lam, A.Y. S.*, et al.*: 'Intelligent time-adaptive transient stability assessment system', *IEEE Trans. Power Syst.*, 2017, **33**, (1), pp. 1049–1058