A novel data-driven approach for transient stability prediction of power systems considering the operational variability

Yanzhen Zhou\textsuperscript{a}, Qinglai Guo\textsuperscript{a}, Hongbin Sun\textsuperscript{b,⁎}, Zhihong Yu\textsuperscript{b}, Junyong Wu\textsuperscript{c}, Liangliang Hao\textsuperscript{c}

\textsuperscript{a} Department of Electrical Engineering, State Key Laboratory of Electricity Systems, Tsinghua University, 100084 Beijing, China
\textsuperscript{b} China Electric Power Research Institute, 100192 Beijing, China
\textsuperscript{c} School of Electrical Engineering, Beijing Jiaotong University, 100044 Beijing, China

\textbf{A R T I C L E   I N F O}

Keywords:
Data driven
Transient stability prediction
Convolutional neural network
Active learning
Fine-tuning

\textbf{A B S T R A C T}

Data driven methods are playing an increasingly important role in transient stability assessment, primarily because of the availability of large annotated datasets. Nevertheless, training data cannot cover all the possible operating conditions of a modern power system with variable power generations and loads. The classifier should adjust to the near-future operation condition in limited time, and this adjustment may be hindered by the computational time of the simulations and classifier training. To dramatically reduce the computational cost, this paper presents a systematic approach for building and updating an accurate transient stability classifier. First, the time-series trajectories of generators after disturbance are used as the inputs, and then a convolutional neural network (CNN) ensemble method is proposed to generate the transient stability predictor using these multi-dimensional data. To reduce the misclassification of instability, different cost weights are considered for the stable and unstable instances in the loss function. When the operating condition changes substantially and makes the pre-trained classifier unavailable, the active learning and fine-tuning techniques are integrated to update the classifier with good performance using fewer labelled instances and short computational time. The simulation results of two power systems illustrate the effectiveness of the proposed approach.

1. Introduction

During the past several years, there has been a substantial promotion of renewable energy and flexible loads to modern power systems, which brings challenges for the security and stability of the power system [1]. Transient stability, or large-disturbance rotor angle stability, is one of the major concerns for planning, operation and control of power systems [2]. Many dynamic units in power systems act on one another, which results in the complexity of transient stability analysis.

The most accurate method for transient stability analysis is time-domain simulation; however, this approach is time consuming, especially for large-scale power systems. In [3], the piecewise constant-current load equivalent is proposed to reduce the computing time for integrating the differential/algebraic equation model of post-fault power system dynamics, but the computational speed seems insufficient for real-time application. By contrast, the direct methods, such as transient energy function method [4] and extended equal-area criterion [5], can compute quickly and provide the stability margin, whereas the model limitations are inevitable. In [6], a hybrid method based on the new concept of equal area criterion and corrected kinetic energy function is proposed for real-time transient stability assessment. It can be concluded that the methods with detailed models can provide accurate prediction, whereas the computation speed seems insufficient for real-time application. To achieve the fast prediction, it is unavoidable to make some assumptions to simplify the problem at the expense of accuracy.

Thanks to the wide availability of massive energy data in the power systems, it is expected that data analytics will play an increasingly important role in modern power systems [7,8]. Data-driven method, which has powerful and flexible modelling capability [9], as well as the fast prediction after the inputs are obtained, has been considered as alternatives for transient stability prediction [10–17].

When using the data-driven methods for transient stability prediction, a set of appropriate features should be selected and extracted as the inputs; then an appropriate classification method is utilized to predict the transient stability status. The input features of data-driven based methods used for transient stability prediction include the generator rotor angles and speeds [10–13], generator terminal voltages [14,15], apparent impedance [16] and the combination of different variables [17], etc. In [18], the terms of transient energy functions are
used as the input features. In [19], several wide-area severity indices are defined in the time and frequency domains. These indices are used as the inputs to build the classifiers for catastrophe predictor. In [20], the empirical orthogonal function is used for feature extraction and selection before training a classifier for transient stability analysis.

Various algorithms have been used to train the classifier for transient stability prediction, such as artificial neural network [10], decision tree [11–14,16,21], support vector machine [15,17,18,20,22], and random forest [19]. Recently, the rapid development of deep learning has promoted the widespread exploration of deep learning in industry applications [23–28]. The deep learning model can provide more accurate results than the shallow models. In [29], the long short-term memory (LSTM) is proposed for transient stability prediction. The results show that the LSTM-based method can achieve a superior assessment accuracy within a short period of time after disturbance.

In general, there are two types of application scenarios for the data-driven based methods. One scenario is to analyse the transient stability considering anticipated but non-occurring contingencies [18,30]. In this case, the steady-state measurements are usually used as the inputs [30]. When the power system is predicted as insecure, prevention control actions should be conducted. In [18], the post-disturbance data are used for online dynamic security assessment considering anticipated contingencies. These dynamic data can be obtained from short-term simulations and the generated classifier has good performance. Therefore, the dynamic data can also be used for pre-contingency analysis. The other scenario is to provide real-time prediction of the transient stability status after contingencies [10–17,19,20]. In this case, the dynamic responses of different variables that can be obtained by wide area measurement systems are always used as the inputs.

This paper focuses on the transient stability prediction using dynamic data. The dynamic data can be obtained from post-contingency real-time measurements or pre-contingency short-term simulations. Therefore, the proposed method can be applied to predict the transient stability status under both occurred and anticipated contingencies. When using the dynamic trajectories of different variables and different generators for transient stability prediction, the input is a massive amount of spatio-temporal data. Spatio-temporal data can be stored as multi-dimensional arrays and have similar data structure to that of the images. This characteristic motivated us to utilize the convolutional neural network (CNN) [31], a popular deep learning network especially in computer vision applications, to perform the transient stability prediction. During the operation of power systems, the operating conditions, such as the load levels and network topologies, vary with generation and load changes, generation dispatch, scheduled maintenance, etc. Thus, the number of possible operating conditions is infinite, and there are unavoidable differences between the offline training and the practical conditions. The classifier obtained by offline simulation and historical data may become unsuitable and need to be updated periodically [18,19,21]. To obtain a new classifier, a comprehensive dataset should be generated considering the uncertainties associated with load levels, system topology, fault types, locations, clearance time, etc. And the classifier is retrained using the new dataset. The data generation and classifier retraining may be time consuming and become an obstacle for updating.

To dramatically reduce the computational cost of classifier updating, this paper presents a CNN-based method integrating active learning and fine-tuning techniques. Active learning aims to train an accurate model with low cost by only labelling the important instances [32,33]. Fine-tuning involves adapting an existing pre-trained model to another domain by executing a few training iterations on a new dataset [23]. By integrating these two techniques, the generated classifier can be updated using less training instances in shorter time, making the updating process more suitable for practical application.

The main contributions of this paper can be summarized as follows: (1) All the time-series features are stored as multi-dimensional arrays, and then three different CNNs with different window sizes of kernels are built to construct an ensemble classifier with high accuracy and generalization ability. (2) Before the operating conditions and topologies change greatly (as predicted by generation and load forecasting, dispatching, scheduled maintenance, etc.), the classifiers can be updated rapidly by fine-tuning the pre-trained classifier using the most informative and representative instances, and the computational costs of both the data generation and classifier training process are greatly reduced, making the proposed approach more suitable for practical application.

The remainder of this paper is organized as follows. The physical description of transient stability, the mathematical description and analysis of data-driven based method for transient stability analysis are included in Section 2. Section 3 introduces the proposed CNN ensemble classifier considering the high-dimensional characteristic of the inputs. The total data-driven approach for transient stability prediction considering the operational variability is described in Section 4. Comprehensive case studies and discussions are presented in Section 5. The conclusions are given in Section 6.

2. Problem description and analysis

The basic structure of power system for transient stability analysis is shown in Fig. 1. There is one synchronous generator combined with the loads and other dynamic units through the power grid. The imbalance between the generator mechanic power $P_m$ and electromagnetic power $P_e$ is associated with the variation of rotor angles and speeds. The rotor speed signal is then put into the prime mover and its speed control system to correct the output torque of the prime mover $T_m$, and hence influence the $P_m$. On the other hand, the generator voltage $U_t$ is echoed back to the excitation system to control the field voltage $E_f$. For practical power systems, there are many generators, loads and other dynamic units. These units are combined together through the power grid and act on one another, which results in the complexity of transient stability analysis.

Data driven methods have been adopted as alternative approaches for transient stability analysis, which is mainly because the rapid computation. When using data driven methods to predict the transient stability, a set of variables $X = (X_{1,1}, X_{2,2}, \ldots, X_{4,4})$ should be selected to describe the operating conditions of power systems. Next, a set of

![Fig. 1. Basic structure of power system model for transient stability analysis [34].](image-url)

Y. Zhou et al.

Table 1

<table>
<thead>
<tr>
<th>Input features</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative rotor angle [5] of generator (i) at (t = t_k)</td>
<td>(\delta_i(t_k) = \sum_{j=1}^{n} M_i \delta^j(t_k) / \sum_{j=1}^{n} M_j)</td>
</tr>
<tr>
<td>Relative rotor speed of generator (i) at (t = t_k)</td>
<td>(\omega_i(t_k) = \left[\sum_{j=1}^{n} M_i \delta^j(t_k) \right] / \Delta t)</td>
</tr>
<tr>
<td>Relative rotor acceleration of generator (i) at (t = t_k)</td>
<td>(\ddot{\omega}<em>i(t_k) = \left[\sum</em>{j=1}^{n} M_i \delta^j(t_k) \right] / \Delta t^2)</td>
</tr>
<tr>
<td>Kinetic energy of generator (i) at (t = t_k)</td>
<td>(KE_i(t_k) = \frac{1}{2}M_i (\dot{\delta}_i(t_k))^2)</td>
</tr>
<tr>
<td>Relative electromagnetic power of generator (i) at (t = t_k)</td>
<td>(\bar{P}_e_i(t_k) = \bar{P}_r_i(t_k) / \bar{P}_r(0))</td>
</tr>
</tbody>
</table>

where \(n\) is the total number of generators, \(M_i\) is the inertia coefficient of generator \(i\), \(\delta_i(t_k)\) is the rotor angle of generator \(i\) at \(t = t_k\), \(\Delta t = t_k - t_{k-1} = t_{k-1} - t_{k-2}\), \(P_{e_i}(t_k)\) is the electromagnetic power of generator \(i\) at \(t = t_k\).

The transient stability directly depends on the dynamics of generator rotor angles. Thus, the rotor angle-related variables are always used as the inputs \([10–13,17]\). Moreover, the rotor angle is affected by the electromagnetic power \([17]\). When the fault occurs, the electromagnetic power will suddenly change. After the fault clearance, relative electromagnetic power of all the generators are used as the inputs. Detail information of the input features are shown in Table 1. More case studies and discussions about the input features will be provided in Section 5.3 and 5.6.

The aforementioned input data can be stored as multi-dimensional arrays, as shown in Fig. 2.

The input data are high dimensional when using these time-series features, especially for a large-scale power system with hundreds of generators. As a result, the computational space and time increase greatly with increasing system size. Actually, the data structure of these time-series variables is similar to that of the images. The variable axis of these time-series data depicts different views of all the generators (e.g., the rotor angle, speed, kinetic energies), analogous to the colour channels of an image. Moreover, there are two axes (time and generator) featuring different variables that can be viewed as being analogous to the width and height axes of the image. This characteristic motivated us to utilize the CNN, a well-known deep learning method for image classification, to perform the transient stability prediction using these time-series features.

3. CNN ensemble classifier for transient stability prediction

3.1. Brief introduction of CNN

For the transient stability prediction with high-dimensional input, it is not practical to apply the fully connected neural network to generate a classifier since a high number of weights for each neuron is necessary. And the required number of training instances will increase as the parameter number increases. The CNN provides a solution to reduce the parameter number. The convolutional neuron shares the same weights for each spatial location \((i, j)\); therefore, the parameter number is greatly reduced. The output \(o_{ij}\) of a convolutional layer with location \((i, j)\) is as follows:

\[
o_{ij} = \sigma \left( (W \ast X)_{ij} + b \right)
\]

where \(W\) is a kernel with learned weight, \(b\) is the bias, \(\ast\) is the convolution operation that is always substituted by the correlation operation, and \(\sigma()\) is the activation function. After the convolutional layer, it is common to add a pooling layer to reduce the data dimension. The most commonly used pooling layer, max-pooling, is utilized here.

After the convolutional and pooling process, an activation layer is required to make the learned features more dividable. In this research, the activation function of the convolutional layers and the pooling layers is the rectified linear unit (ReLU), which is widely used as an activation unit to accelerate the convergence of the CNN \([31]\):

\[
ReLU(x) = \max(0, x)
\]

Fig. 2. Data structure of the time-series variables for transient stability prediction.
where \( x \) is the input to the activation layer.

A fully connected layer is required to generate the classification stage after the convolutional layers and pooling layers. The mathematical representation of fully-connected layer is:

\[
o_{ij} = \sigma_i(W_i \ast x_r + b_r)
\]

where \( x_r \) is the input data of the fully-connected layer, \( W_i \) is the weight matrix, \( b_r \) is the bias. The activation function of the fully connected layer is Softmax:

\[
P(C_k \mid X) = \frac{e^{\eta_k(X)}}{\sum_{k=1}^{V} e^{\eta_k(X)}}
\]

where \( \eta_k(X) \) is the \( k \)-th input data of the Softmax layer with instance \( X \), \( P(C_1 \mid X) \) and \( P(C_2 \mid X) \) are the probabilities of an unknown instance \( X \) identified as class-1 and class-2 (binary classification), respectively. The final output of CNN is \( \hat{y}_i = (\hat{y}_i^{(1)}, \hat{y}_i^{(2)}) = (P(C_1 \mid X), P(C_2 \mid X)) \). If \( P(C_1 \mid X) > P(C_2 \mid X) \), then the unknown instance will be identified as class 1 and vice versa.

Fig. 3 shows a CNN structure with one convolution and pooling layer.

The weight matrix and the corresponding biases in CNN are achieved through the minimization of the loss between the true label vectors \( y_i = (y_i^{(1)}, y_i^{(2)}) \) and the predicted label vectors \( \hat{y}_i = (\hat{y}_i^{(1)}, \hat{y}_i^{(2)}) \), for \( N \) training instances \( i = 1, \ldots, N \). If \( y_i^{(1)} = 1 \), then the unknown instance will be identified as stable and vice versa. In this research, the common cross-entropy loss is used as the loss function, shown as Eq. (5):

\[
L = -\frac{1}{N} \sum_{i=1}^{N} H(y_i, \hat{y}_i) = -\frac{1}{N} \sum_{i=1}^{N} [y_i^{(1)} \log(\hat{y}_i^{(1)}) + y_i^{(2)} \log(\hat{y}_i^{(2)})]
\]

where \( N \) is the total number of training instances. The cross-entropy loss has fast convergence rates and is numerically stable when coupled with softmax normalization [35].

The Adam stochastic optimization algorithm is applied to minimize the loss function. Adam is a straightforward, memory-saving and computationally effective method and is quite suitable for models with large inputs of data. Details of this optimization algorithm can be found in [36].

3.2. Proposed CNN-based ensemble classifier

For the transient stability prediction, the original training data are always unbalanced with fewer unstable instances. This condition will lead to under-fitting for unstable instances. If a stable case is false alarmed, then there is little impact on the power system security because the emergency measures always make the power system strong. In contrast, an unstable case that is misidentified will result in a disaster when no measure has been taken to prevent the collapse. Therefore, the cost of misdetecting instability is greater than stability. One solution to reduce the misdetection of instability is to set different costs to impose greater weight on the unstable instances, and the loss function in Eq. (5) can be rewritten as

\[
L = -\frac{1}{N} \sum_{i=1}^{N} \left[ W_u y_i^{(1)} \log(\hat{y}_i^{(1)}) + W_s y_i^{(2)} \log(\hat{y}_i^{(2)}) \right]
\]

where \( W_u \) and \( W_s \) are the cost weight of unstable and stable instances, respectively. When \( W_u = W_s = 1 \), the loss function put the same importance to the stability and instability. When \( W_u > W_s \), that means the misdetection cost of instability is greater than stability.

On the other hand, the performance of a CNN is greatly influenced by the parameter construction. In the literatures, the window sizes of the first convolutional layer of different kinds of CNNs are different, e.g., \( 3 \times 3 [37], 5 \times 5 [31] \) and \( 7 \times 7 [38] \). The size of the first convolutional layer can be larger than that of the subsequent layers, but a larger window size corresponds to more parameters. The most commonly used size of the other convolutional layers is \( 3 \times 3 \), and the size of max-pooling is usually selected as \( 2 \times 2 [31,37,38] \). A better classifier with appropriate construction can only be determined via many time-consuming trials. Moreover, it is difficult to ensure that the selected classifier can always perform the best of any test scenarios. For this reason, different CNNs with different parameter configurations are trained to construct an ensemble classifier. All the sub-classifiers can be trained by multiple graphic processing units (GPUs), making the proposed approach more suitable for practical application. It should be noted that the determination of CNN number should depend on the comprehensive case studies and hardware condition. The combination of different classifiers is a successful approach to reduce test errors [39]. Whereas, using too many CNNs may require more graphic processing units. In this research, three different CNNs are trained to generate the ensemble classifier to verify the effectiveness. For different applications and hardware conditions, the number of CNNs may be different.

Two types of power systems are used for case studies: the New England 10-machine 39-bus system and the Northeast Power Coordinating Council (NPCC) 48-machine 140-bus system. It is assumed that all the dynamic variables are sampled simultaneously every cycle (0.0167 s for 60 Hz system), therefore the input features are time series of which time interval is 0.0167 s. The response time of dynamic trajectories is \( N_{cycle} / 60 \) s in the case studies; so, the number of cycles is \( N_{cycle} \). The input sizes are \( N_{cycle} \times 10 \times 5 \) and \( N_{cycle} \times 48 \times 5 \) for the 10-machine and 48-machine test systems, respectively. Two convolutional and max pooling layers are enough for these two kinds of data. The data pre-processing and network structure of proposed CNN ensemble classifier are shown in Fig. 4.

In Fig. 4, all the input data are pre-processed by maximum and minimum normalization first. Then, the normalized data are divided into three axes according to the generator number, variables and response time. Therefore, the input size are \( N_{cycle} \times 10 \times 5 \) and \( N_{cycle} \times 48 \times 5 \) for the 10-machine system and the 48-machine system,
### The data pre-processing and network structure of the proposed CNN-based ensemble classifier.

![Diagram](image_url)

**Fig. 4.** The data pre-processing and network structure of the proposed CNN-based ensemble classifier.

### Details of the CNN models.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel Size/Stride</th>
<th>Kernel Number</th>
<th>Padding</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convolution</td>
<td>10-machine</td>
<td>32</td>
<td>Same</td>
<td>10 × 32</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 10 × 32</td>
<td>Same</td>
<td>24 × 32</td>
</tr>
<tr>
<td>2. Pooling</td>
<td>10-machine</td>
<td>32</td>
<td>Same</td>
<td>10 × 32</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 10 × 32</td>
<td>Same</td>
<td>24 × 32</td>
</tr>
<tr>
<td>3. Convolution</td>
<td>10-machine</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 10 × 64</td>
<td>Same</td>
<td>24 × 64</td>
</tr>
<tr>
<td>4. Pooling</td>
<td>10-machine</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 10 × 64</td>
<td>Same</td>
<td>24 × 64</td>
</tr>
<tr>
<td>5. Fully connected</td>
<td>10</td>
<td>120</td>
<td>–</td>
<td>120 × 1</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 120</td>
<td>–</td>
<td>24 × 1</td>
</tr>
<tr>
<td>6. Fully connected</td>
<td>10</td>
<td>30</td>
<td>–</td>
<td>30 × 1</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 30</td>
<td>–</td>
<td>24 × 1</td>
</tr>
<tr>
<td>7. Softmax</td>
<td>10-machine</td>
<td>2</td>
<td>–</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>48-machine</td>
<td>Ncycle × 2</td>
<td>–</td>
<td>24 × 2</td>
</tr>
</tbody>
</table>

### Details of the CNN models.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Kernel Size/Stride</th>
<th>Kernel Number</th>
<th>Padding</th>
<th>Output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convolution</td>
<td>(1) 3 × 3/1 × 1</td>
<td>32</td>
<td>Same</td>
<td>10 × 32</td>
</tr>
<tr>
<td></td>
<td>(2) 5 × 5/1 × 1</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td></td>
<td>(3) 7 × 7/1 × 1</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td>2. Pooling</td>
<td>2 × 2/2 × 2</td>
<td>32</td>
<td>Same</td>
<td>10 × 32</td>
</tr>
<tr>
<td>3. Convolution</td>
<td>3 × 3/1 × 1</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td>4. Pooling</td>
<td>2 × 2/2 × 2</td>
<td>64</td>
<td>Same</td>
<td>10 × 64</td>
</tr>
<tr>
<td>5. Fully connected</td>
<td>120</td>
<td>1</td>
<td>–</td>
<td>120 × 1</td>
</tr>
<tr>
<td>6. Fully connected</td>
<td>30</td>
<td>1</td>
<td>–</td>
<td>30 × 1</td>
</tr>
<tr>
<td>7. Softmax</td>
<td>2</td>
<td>1</td>
<td>–</td>
<td>2</td>
</tr>
</tbody>
</table>

### The data-driven approach for transient stability prediction considering the operational variability

The data-driven method for transient stability prediction consists of three important processes: (1) Offline classifier training, (2) Period updating and (3) Practical application. The temporal relationships of these three processes are shown in Fig. 5.

The offline training and period updating process should be finished before application. Detail of the data-driven method for transient stability prediction is shown in Fig. 6.

#### 4.1. Offline classifier training

During the offline training process, a comprehensive dataset, \((X_i, y_i)\), should be generated for the next application period (e.g. \(t_2-t_5\) in Fig. 5), where \(X_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,k})\) is a set of input variables, \(y_i\) is the transient stability status of the \(i\)-th instances based on long-term time-domain simulation, \(i = N_{oc} \times N_i\) represents that \(N_{oc}\) operation conditions (OCs) and \(N_i\) contingencies are included in the dataset, the determination of \(N_{oc}\) and \(N_i\) is related to the scale of power systems and the time period for application. The uncertainties associated with load levels, system topology, fault types, locations, clearance time should be considered [11]. By training from massive data, the data-driven method can find a mapping between the input features \(X_i\) and the transient stability analysis result \(y_i = \psi(X_i)\). Before application, the classifier should be verified by several test datasets or cross-validation to avoid the over-fitting.

#### 4.2. Period updating integrating active learning and Fine-tuning techniques

The updating process should be conducted infrequently on daily, hourly or shorter. The prospective OCs in the next period can be
predicted using a short-term generation & load forecast and the associated unit commitment rules [18,21,42]. If the prospective OCs and anticipated contingencies are close to any of the OCs and contingencies already considered in the offline process, the classifier can be viewed as effective. When new OCs or anticipated contingencies appear or are predicted, new instances \( D_{\text{new}} \) should be generated based on the new OCs and contingencies [21]. If the prediction accuracy of the classifier for \( D_{\text{new}} \) reaches to a set value \( A_{\text{set}} \), the classifier remains frozen until the next updating period. Otherwise, the classifier should be updated by the proposed updating scheme.

During the updating process, generally, the required number of instances considering \( N_{oc} \) new operation conditions and \( N_{c} \) contingencies for the next period should be obtained via a short-term simulation, that is \( X_{\text{new}}^{s} \) where \( s = 1 \ldots N_{c} \times N_{oc} \), \( N_{oc} \) and \( N_{c} \) are related to the scale of power systems and the time period for application. These unlabeled instances will be placed into the pre-trained or old classifier to determine which \( N_{\text{new}} \) instances are important for the updating process. Only \( N_{\text{new}} \) instances will be annotated with the target labels using long-term simulation at each iteration (The value of \( N_{\text{new}} \) is manually set). These new labeled instances \( (X_{\text{new}}^{s}, y_{\text{new}}^{s}) \) will be put into the new training database, then the classifier is updated by fine-tuning the pre-trained classifier using the new training database. The aforementioned process will continue until the performance of the new classifier for \( D_{\text{new}} \) is satisfied. Then the new classifier will be used as the final classifier for the next period.

Given an unlabelled instance set \( D_{l} \), a labelled training set \( D_{l} \), a new labelled testing set \( D_{\text{new}} \), an uncertainty function \( f_{u} \), and a pre-trained classifier \( M_{0} \), the proposed updating algorithm is shown as follows.

**Algorithm 1 Proposed Updating Scheme Integrating Active Learning and Fine-tuning Techniques**

Given the number of new labelled instances at each iteration, \( N_{\text{new}} \), and the iteration time \( i = 0 \).

**While** (the terminating condition is not satisfied) **do**:

1. \( i = i + 1 \)
2. Compute the uncertainty index, \( e_{u,c} \), of all the unlabelled instances.
3. Query the label of the top \( N_{\text{new}} \) instances with higher uncertainty by long-term simulation, \( D_{l} \)
4. \( D_{l} \leftarrow D_{l} \cup D_{\text{new}} \)
5. Obtain the new classifier \( M_{i} \) by fine-tuning \( M_{i-1} \) using \( D_{l} \)
6. Go to 1 if the terminating condition is not satisfied

**end while**

One key component of this algorithm is to construct the uncertainty measure for the unlabelled instances. For the binary classification for transient stability prediction, the uncertainty measure is sufficient to determine which instances should be annotated labels. A basic uncertainty sampling strategy is to query the instance whose predicted output has the lowest confidence [32]:

\[
e_{u,c} = \arg \min_{\hat{y}} P_{c}(\hat{y}|X)
\]

where \( \hat{y} \) is the most likely label with the highest posterior probability in the hypothesis. This index prefers the instances with the least confidence (or highest uncertainty) in deciding their most likely class labels. For example, the output of an unknown instance \( X_{1} \) is \( P_{d}(C_{1}|X_{1}) = 0.45 \), \( P_{d}(C_{2}|X_{1}) = 0.55 \). Therefore the most likely label of \( X_{1} \) is class-2, the confidence of \( X_{1} \) is 0.55, and the uncertainty of \( X_{1} \) is 0.45. In contrast, the output of an unknown instance \( X_{2} \) is \( P_{d}(C_{1}|X_{2}) = 0.05 \), \( P_{d}(C_{2}|X_{2}) = 0.95 \). Although the most likely label of both \( X_{1} \) and \( X_{2} \) are class-2, the prediction confidence of \( X_{2} \) is 0.95 whereas \( X_{1} \) is 0.55. Therefore, the instance \( X_{1} \) with less confidence is preferred to be annotated a label in this strategy.

In the Algorithm 1, the number of epochs is selected as eight during the fine-tuning process at each iteration. The terminating condition is satisfied only if one of the following conditions is true: (1) The accuracy for \( D_{\text{new}} \) reaches to a set value \( A_{\text{set}} \). (2) The accuracy for \( D_{\text{new}} \) cannot improve during \( T_{\text{set}} \) iterations. In the case studies, \( A_{\text{set}} = 98\% \), \( T_{\text{set}} = 10 \).
Based on the proposed updating process, only \( i \times N_{\text{new}} \) instances are annotated with the target labels using long-term simulation and added into the training database, where \( i \) is the iteration time, \( N_{\text{new}} \) is manually set. Therefore, the computational time of both the time-domain simulation and classifier training can be greatly reduced, making the proposed approach more suitable for practical application.

4.3. Application

The inputs of the CNN ensemble classifier include dynamic data. As mentioned before, these dynamic responses can be obtained from real-time measurements as well as short-term simulations [18], therefore, the CNN ensemble classifier can be applied to both the post-contingency and pre-contingency application.

For the pre-contingency application, the short-time simulation is used to prepare dynamic inputs. The transient stability of power systems under anticipated contingencies can be estimated using the generated classifier. The system security state is determined through the comprehensive results of the stability under all anticipated contingencies.

For the post-contingency application, the rotor position measure method is adopted to obtain the measurements of generator rotor angles [12], and the trajectories of electromagnetic power are calculated by the voltage magnitude and current of different generators. Therefore, these dynamic responses can be obtained from measurements and sent into the CNN ensemble classifier to rapidly predict the transient stability of power systems. Generally, the performance of classifiers with longer response time is better than the classifiers with shorter response time. The sooner the prediction is completed, the longer the time available to take countermeasures to prevent a possible collapse will be. It is hard to draw a definite conclusion about how long the best response time for any power system should be. In [43], a hierarchical scheme is proposed for transient stability prediction, the classifiers with different response times constitute the different layers to balance the rapidity and accuracy. And the hierarchical scheme can also be used for the proposed CNN ensemble classifier in this research.

On the other hand, it should be noted that the data-driven method is NOT a replacement but addition for other methods. In practical application, the data-driven method should be used combining with the time-domain simulation and other methods based on a multiple level structure in Fig. 7. On the first level, the data-driven method is responsible to rapidly predict the transient stability status of power systems under anticipated or occurred contingencies. For some cases with low prediction confidence, it is reasonable to view these cases as uncertain temporary. And these uncertain cases can be further identified by time-domain simulation or other methods.

5. Case study

5.1. Data generation

The generations and loads are randomly changed around 80–120% of standard operating point to generate a statistically sound dataset. The generation of real power, relative power, real power generation and the terminal voltage of generation at i-th operating point are based on the following equations [18]:

\[
P_{G}(i) = P_{G}(i) + \Delta P_{G}(i) \quad (10) \\
Q_{L}(i) = Q_{L}(i) + \Delta Q_{L}(i) \quad (11) \\
P_{L}(i) = P_{L}(i) + \Delta P_{L}(i) \quad (12) \\
V_{C}(i) = V_{C}(i) + \Delta V_{C}(i) \quad (13)
\]

where \( \Delta P_{G}(i), \Delta Q_{L}(i) \) and \( \Delta P_{L}(i) \) stand for the variation range of load and generation power, \( \Delta V_{C}(i) = 0.02 \) means the variation range of the generator terminal voltage is \( \pm 2\% \), \( \epsilon_{P_{L}}(i), \epsilon_{Q_{L}}(i), \epsilon_{P_{G}}(i) \) are the random numbers in \([0,1]\).

Three-phase short circuit faults at three locations (10%, 50% and 90% of the head of line) of all the transmission lines are considered. Four types of fault clearing times ranging from 6 to 12 cycles with an increment of 2 cycles (1 cycle is 0.0167 s for 60 Hz system) are assumed for all the contingencies.

To annotate the label of different operation conditions and contingencies, the time domain simulation should be conducted. The transient stability index, \( \eta \), is calculated from the generator rotor angles [44]:

\[
\eta = \frac{360^\circ - |\Delta \delta_{\text{max}}|}{360^\circ + |\Delta \delta_{\text{max}}|} \quad (14)
\]

where \( |\Delta \delta_{\text{max}}| \) is the absolute value of the maximum angle of separation between any two generators after disturbance. When \( \eta \geq 0 \), the system is considered as stable and vice versa. Thus, the power system is considered as transiently unstable when \( |\Delta \delta| > 360^\circ \). It should be noted that the stability criterion is dependent on the power system characteristics [10], and different researches have used different criteria. The criterion in [18] is whether the difference between the rotor angles of any two generators exceeds 360° in the 3 s after clearing time. In [30,43,45], the criterion is 360° in the 4 s after clearing time. In [11], the criterion is 360° during 6 s simulation. In this research, the criterion for transient stability is whether the difference between the rotor angles of any two generators exceeds 360° within 4 s after the fault clearance. All the transient stability simulations are conducted using the Power System Toolbox 3.0 [46].

Two test power systems—the 10-machine 39-bus power system and the NPCC 48-machine 140-bus system—are used for the case study. The excitation systems and detailed models of synchronous generators are considered in these two power systems. The dynamic parameters of 10-machine power system can be found in [47], and the dynamic parameters of 48-machine power system are taken from the power system toolbox [46].

5.2. Evaluation indices

A useful tool for evaluating the performance of classifiers is the confusion matrix, shown in Table 3.

### Table 3

<table>
<thead>
<tr>
<th>Confusion matrix</th>
<th>Stable (Predicted)</th>
<th>Unstable (Predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable (Actual)</td>
<td>( T_{s} )</td>
<td>( F_{us} )</td>
</tr>
<tr>
<td>Unstable (Actual)</td>
<td>( F_{s} )</td>
<td>( T_{us} )</td>
</tr>
</tbody>
</table>
Setting \( N_s \) and \( N_u \) as the total number of stable and unstable instances, respectively, there are \( N_s = T_s + F_u \) and \( N_u = T_u + F_s \). Therefore, the related indices are defined as follows [48,49]:

\[
\text{Accuracy: } \text{Acc} = \frac{(T_s + T_u)}{(N_s + N_u)} \tag{15}
\]

\[
\text{Reliability: } A_{us} = \frac{T_u}{(N_u + F_u)} \tag{16}
\]

\[
\text{Security: } A_s = \frac{T_s}{(N_s + F_s)} \tag{17}
\]

\[
G - \text{mean: } G = \sqrt{A_{us} \times A_s} \tag{18}
\]

Accuracy is a widely used index without considering class imbalance. Reliability and Security denote the ratio of correctly identified unstable cases and stable cases, respectively. G-mean is a geometric mean of the Reliability and Security, evaluating the overall performance of the classifier for imbalanced data, which is the main concern in this research.

5.3. Performance of the CNN ensemble classifier

(1) Ensemble CNN Classifier vs Single CNN Classifier

To verify the effectiveness of the proposed CNN ensemble classifier, two test systems are used to generate the training data and testing data. The ratio of the test data to the training data is approximately equal to 1/4, as shown in Table 4.

Different CNN classifiers are generated using the training data with 9 cycles data (150 ms for this 60 Hz system) to predict the transient stability. Setting \( W_{us} = W_s = 1 \) in the loss function. Tables 5 and 6 show the performances of different classifiers for testing datasets of the 10-machine system and the 48-machine system, respectively.

Tables 5 and 6 show that the prediction errors of different CNN classifiers are different for different datasets, e.g., the CNN 1 (CNN with size 3 × 3 of the first convolution layer) performs the worst in 10-machine system but the best of the three CNN classifiers in 48-machine system. By using the ensemble method, the proposed CNN ensemble classifiers perform the best, with the highest G-mean for these two test power systems, and therefore the diverse errors are avoided.

(2) Cost Ratio Adjustment vs Without Cost Ratio Adjustment

Table 7 shows the prediction results of testing data using different cost ratios \( W_{us}/W_s \) in the loss function (6), where \( W_{us} \) and \( W_s \) are the weights of unstable and stable instances, respectively.

With the increase of cost ratios, the CNN-based ensemble classifier successfully identifies more unstable instances, whereas the number of accurately identified stable instances is reduced. Nevertheless, the performance is still satisfied due to the high G-mean and accuracy (above 98%). It can be seen that the classifier has the highest G-mean value when \( W_{us}/W_s = 2 \) and \( W_{us}/W_s = 4 \) for the 10-machine system and 48-machine system, respectively.

Having a careful check of the training data in Table 4, the ratio of stable instance number (\( N_s \)) and unstable instance number (\( N_u \)) is 10539/5779 = 1.82 for the 10-machine system, and \( N_s/N_u = 66744/17055 = 3.91 \) for the 48-machine system. It can be found that the best value of \( W_{us}/W_s \) is pretty close to the value of \( N_s/N_u \) in the training data. Therefore, it is recommended to directly use a similar value of \( N_s/N_u \) for \( W_{us}/W_s \) when the training data is imbalanced with less unstable cases.

(3) Ensemble CNN Classifier vs Other Kinds of Classifiers

An additional four classifiers are generated using the same training data in Table 4 for comparison: decision tree (DT), random forest (RF) with 100 trees, and multilayer perceptron (MLP) with four hidden layers. The decision tree is the classification and regression tree based on Gini impurity. The kernel size of the four fully-connected layers in MLP are 100, 50, 10 and 2 respectively. Considering the class imbalance in the training data, the extended data [19] is used to train the DT, RF and MLP. For the 10-machine system, the unstable cases are replicated one time to obtain the extended data. For the 48-machine system, the unstable cases are replicated three times. Then the DT, MLP and RF generated by the extended data are called as the extended DT, extended MLP and extended RF, respectively. Two layers of LSTM memory blocks are employed in the LSTM layer, where each layer has 128 memory blocks [29]. The network of LSTM and CNN are trained for up to 100 epochs. The three CNNs are trained in parallel. All these methods are run on a PC with an Intel Core i7-4790 3.6-GHz processor, 16 GB of RAM and a Titan X Pascal GPU. Tables 8 and 9 show the performance of different classifiers for the testing datasets of the 10-machine system and 48-machine system, respectively.

Tables 8 and 9 show that the proposed CNN ensemble classifiers perform the best, with the highest G-mean values for these two test power systems. The order of the test time for one instance is 10 \(^{-3}\) s. The memory space of the CNN ensemble classifiers for the two test system are the same because the parameter numbers are equal (Table 2). The training time of CNN ensemble classifier is a little long but still acceptable because the training process is conducted offline.

On the other hand, the training time of 48-machine system increases from 417.38 s to 2056.27 s mainly because the number of training instances increases. Fig. 8 shows the training time of the CNN ensemble classifier using different numbers of training instances.

It can be seen from Fig. 8 that, when using the same number of training instances, there are minor differences between the training time of the 10-machine system and 48-machine system because the parameter numbers are equal in Table 2. For a larger power system, therefore, the training time mainly depends on the number of instances when using the same CNN ensemble classifier with the same epochs.

(4) Effect of Noise

The input features can be provided from measurements and computations, whereas errors may be involved for practical application.

<table>
<thead>
<tr>
<th>Table 5</th>
<th>Prediction results of testing data of the 10-machine system using different CNN classifiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>G-mean</td>
</tr>
<tr>
<td>CNN 1</td>
<td>97.90%</td>
</tr>
<tr>
<td>CNN 2</td>
<td>98.19%</td>
</tr>
<tr>
<td>CNN 3</td>
<td>98.45%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>98.55%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Prediction results of testing data of the 48-machine system using different CNN classifiers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>G-mean</td>
</tr>
<tr>
<td>CNN 1</td>
<td>98.15%</td>
</tr>
<tr>
<td>CNN 2</td>
<td>97.81%</td>
</tr>
<tr>
<td>CNN 3</td>
<td>98.07%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>98.22%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Description of the training and testing data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test system</td>
<td>Training data</td>
</tr>
<tr>
<td></td>
<td>Total Stable Unstable</td>
</tr>
<tr>
<td>10-machine system</td>
<td>16,318 10,539 5,779</td>
</tr>
<tr>
<td>48-machine system</td>
<td>83,799 66,744 17,055</td>
</tr>
</tbody>
</table>
More case studies are conducted to test the prediction performance considering the input errors. The errors were simulated by adding white noises following Gaussian distribution $N(0, \sigma)$ to each variable in the original testing set to generate a new testing set. The new variables $x_{ij}'$ in the testing set is

$$x_{ij}' = x_{ij} \times (1 + \delta), \delta \sim N(0, \sigma) \quad (19)$$

The range of $\sigma$ is selected from 0.01 to 0.08, the prediction results of different classifiers in Tables 8 and 9 are tested. The G-mean value of the testing data are presented in Fig. 9. It can be found that the proposed CNN ensemble classifier has the best performance under the noise condition.

(5) Performance of the CNN Ensemble Classifiers with Different Response Time

For post-contingency application, the total prediction time of the CNN ensemble classifier is related to the response time of inputs, computation time of CNN ensemble classifier and the latency. The latency is unavoidable for any methods. The computation time for one instance after the inputs obtained is 0.004 s. Therefore, the response time is a key aspect for the total prediction time. The performance of the CNN ensemble classifiers using the features with different response times is tested, shown in Fig. 10.

It can be seen that the performance of CNN ensemble classifier is better when using a longer response time. To balance the rapidity and accuracy of transient stability prediction, the hierarchical scheme in [43] can be used for the CNN ensemble classifier.

Take the CNN ensemble classifier with 0.05 s response time as an example, the instability occurrence time histogram of the total testing unstable instances, the detected and undetected unstable instances are compared in Fig. 11.

Fig. 11(a) shows the instability occurrence time histogram of the total testing unstable instances. Fig. 11(b) shows that the CNN ensemble classifier can even detect some cases that instability occur after 3 s after the fault clearance. In Fig. 11(c), the instability occurrence times of the most of the undetected unstable instances are longer than 1 s. It can be found that the proposed CNN ensemble classifier can rapidly detect some instabilities. The average of the instability occurrence time in all the detected unstable instances is 0.8242 s, the response time is 0.05 s, the computation time of CNN is 0.004 s. Therefore, the average preemptive time of the proposed method before instability occurs is 0.7702 s regardless of the latency.

(6) Comparisons among different classifiers with different input features

In the literatures, the voltage-related variables are always used as the inputs because these variables can be obtained directly by phasor measurement units (PMUs). These features can also be used for the proposed CNN ensemble classifier. In this section, the voltage magnetic and phase angles of all the generator buses over the time span of 150 ms after fault clearing are used as the inputs. Then the extended DT, MLP, RF, LSTM and CNN ensemble classifiers are obtained using the same training data in Section 5.3 (3). The prediction results of different classifiers for the test data in 10-machine system are shown in Table 10. It can be seen from Table 10 that, when using the voltage variables as the inputs, the ensemble CNN classifier still perform the best with the highest G-mean value.

In [50], 34 features are extracted from the original trajectories. These features are also used to train a SVM from the same training data in Table 4 for comparison. Besides, the extended SVM is trained from the extended data. The response time is 150 ms after the fault clearing. The parameters of SVM are search by a cross validation with grid-search process. Table 11 shows the performances of different classifiers for testing datasets of the 10-machine system.

It should be noted that the data in this research and Ref. [50] are different. Although the SVM classifier has good performance than the other shallow models, the proposed CNN ensemble classifier (in Table 8) still possesses the best performance with the highest G-mean.

(7) Performance of the CNN ensemble classifier using the features of part of generators

For the post-contingency application, the inputs can only be obtained from real-time measurements. Considering the PMU locations, only the measurements of some generators can be obtained. Therefore, the prediction performance of the CNN ensemble classifier using the information of parts of generators should be considered.

To verify the effectiveness of the proposed approach, the CNN ensemble classifier is retrained only using the features of part of generators. When the features of any one generator are removed from the original inputs, the G-mean and accuracy of the CNN ensemble classifier for the test data in 10-machine system are shown in Fig. 12.

When the features of multiple generators are removed from the original inputs, the situations are more complex because of the randomness. For simplicity, only the situations when the generators’ features are removed according to the descending order of the G-mean

---

**Table 7**

<table>
<thead>
<tr>
<th>$W_{ui}/W_s$</th>
<th>10-machine system</th>
<th>48-machine system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Reliability</td>
</tr>
<tr>
<td>1</td>
<td>98.63%</td>
<td>98.27%</td>
</tr>
<tr>
<td>2</td>
<td>98.58%</td>
<td>98.69%</td>
</tr>
<tr>
<td>3</td>
<td>98.16%</td>
<td>99.03%</td>
</tr>
<tr>
<td>4</td>
<td>98.16%</td>
<td>99.03%</td>
</tr>
<tr>
<td>5</td>
<td>98.16%</td>
<td>99.03%</td>
</tr>
</tbody>
</table>

**Table 8**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Training time</th>
<th>Test time (one instance)</th>
<th>Memory</th>
<th>G-mean</th>
<th>Reliability</th>
<th>Security</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended CART</td>
<td>19.28 s</td>
<td>0.001 s</td>
<td>49 K</td>
<td>95.68%</td>
<td>94.67%</td>
<td>96.70%</td>
<td>95.98%</td>
</tr>
<tr>
<td>Extended MLP</td>
<td>10.80 s</td>
<td>0.001 s</td>
<td>1590 K</td>
<td>97.37%</td>
<td>97.37%</td>
<td>97.38%</td>
<td>97.38%</td>
</tr>
<tr>
<td>Extended RF</td>
<td>57.68 s</td>
<td>0.009 s</td>
<td>4092 K</td>
<td>97.86%</td>
<td>97.86%</td>
<td>97.76%</td>
<td>97.80%</td>
</tr>
<tr>
<td>LSTM, $W_{ui}/W_s = 2$</td>
<td>2220.20 s</td>
<td>0.022 s</td>
<td>4194 K</td>
<td>98.20%</td>
<td>98.34%</td>
<td>98.07%</td>
<td>98.16%</td>
</tr>
<tr>
<td>Ensemble CNN, $W_{ui}/W_s = 2$</td>
<td>417.38 s</td>
<td>0.004 s</td>
<td>3489 K</td>
<td>98.60%</td>
<td>98.69%</td>
<td>98.52%</td>
<td>98.58%</td>
</tr>
</tbody>
</table>
value in Fig. 12. Ten generator sets are shown in Table 12. The prediction results of the CNN ensemble classifier using different generator sets are shown in Fig. 13. Fig. 13 shows that the more generators are considered as the input features, the better the CNN ensemble classifier performs. Besides, the prediction results show that the proposed approach is effective (higher than 97%) when only using the measurements of three generators.

5.4. Performance of the CNN ensemble classifier considering the variation in operation conditions

In Section 5.3, because the training data and testing data are randomly selected from a total dataset, the operating conditions in testing dataset may exist in the training dataset. However, in practical application, the operating conditions cannot be the same as those of the offline simulation scenarios. Therefore, it is more reasonable to verify the generalizability of the proposed classifier under different generations, loads and network configurations.

(1) New England 10-machine 39-bus system

Two new untrained datasets are generated to verify the generalizability of the proposed classifier under different scenarios. Details of the power flow and network configuration are shown in Table 13. For the new test data 1, the new power flows within a similar range can be used to simulate the normal variation in operating situation. The new test data 2 is used to simulate the major variations. Fig. 14 shows the prediction results of different classifiers for these two new datasets. Fig. 14(a) shows that the proposed CNN ensemble classifier performs the best under the normal variation in power flow because it has the highest G-mean for this unbalanced testing set. Furthermore, the accuracy of the proposed CNN ensemble classifier is still up to 97.7%. Therefore, the proposed method can cope with the normal changes of operation conditions. When the power flow and network configuration change substantially, the performance of all the classifiers is reduced notably (Fig. 14(b)). The CNN-based ensemble classifier must adjust itself to make it suitable to the new operating condition. The performance of the proposed updating scheme will be validated in Section 5.5.

### Table 9: Prediction results of testing data of the 48-machine system using different classifiers.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Training time</th>
<th>Test time (one instance)</th>
<th>Memory</th>
<th>G-mean</th>
<th>Reliability</th>
<th>Security</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended CART</td>
<td>858.01 s</td>
<td>0.001 s</td>
<td>149 K</td>
<td>95.96%</td>
<td>93.32%</td>
<td>98.67%</td>
<td>97.54%</td>
</tr>
<tr>
<td>Extended MLP</td>
<td>527.98 s</td>
<td>0.002 s</td>
<td>6934 K</td>
<td>97.55%</td>
<td>98.69%</td>
<td>96.42%</td>
<td>96.89%</td>
</tr>
<tr>
<td>Extended RF</td>
<td>1047.52 s</td>
<td>0.009 s</td>
<td>18862 K</td>
<td>98.25%</td>
<td>97.61%</td>
<td>98.89%</td>
<td>98.63%</td>
</tr>
<tr>
<td>LSTM, $W_o/W_s = 4$</td>
<td>11680.91 s</td>
<td>0.022 s</td>
<td>5334 K</td>
<td>98.06%</td>
<td>98.13%</td>
<td>97.99%</td>
<td>98.08%</td>
</tr>
<tr>
<td>Ensemble CNN, $W_o/W_s = 4$</td>
<td>2056.27 s</td>
<td>0.004 s</td>
<td>3489 K</td>
<td>98.59%</td>
<td>98.71%</td>
<td>98.47%</td>
<td>98.52%</td>
</tr>
</tbody>
</table>

Fig. 8. The training time using different numbers of training instances.

Fig. 9. Comparison of different methods considering the effect of noise.

Fig. 10. Prediction results of the CNN ensemble classifier with different response time of 10-machine system.
The variations in operating conditions are also considered in the larger 48-machine system. Details of the new untrained data are shown in Table 14. The prediction results of different classifiers to these new test datasets are shown in Fig. 15. Fig. 15(a) shows that the proposed CNN-based ensemble classifier performs the best because it has the highest G-mean for these unbalanced testing sets. However, the performance of all the classifiers is reduced greatly in Fig. 15(b) when the operation conditions change substantially. Therefore, the classifier should be updated using new instances.

5.5. Performance of the proposed updating scheme integrating active learning and fine-tuning

According to the previous simulation results, the performance of the classifier will decrease when the operating condition of the power system changes greatly. In fact, these variations can be predicted by generation and load forecasting, dispatching, scheduled maintenance, etc. Based on this information, the classifier should be updated in a limited time. A fast updating scheme can make sure the practical application. To verify the performance of the proposed updating strategy, the comprehensive simulation will be conducted on the two test systems.

(1) New England 10-machine 39-bus system

An additional 11,832 instances are randomly generated using similar configurations as used in testing set 2 (Table 13) to construct the new updating dataset. The power flow of the updating and test datasets are completely different because of the stochastic property in Eqs. (10)–(13). Three different learning strategies are considered to compare the effectiveness of the proposed updating strategy using the same

Table 10
Prediction results of testing data of the 10-machine system using the voltage variables as the inputs.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>G-mean</th>
<th>Reliability</th>
<th>Security</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended CART</td>
<td>92.24%</td>
<td>91.70%</td>
<td>92.79%</td>
<td>92.41%</td>
</tr>
<tr>
<td>Extended MLP</td>
<td>94.09%</td>
<td>95.37%</td>
<td>92.83%</td>
<td>93.73%</td>
</tr>
<tr>
<td>Extended RF</td>
<td>96.43%</td>
<td>96.27%</td>
<td>96.59%</td>
<td>96.47%</td>
</tr>
<tr>
<td>LSTM, $W_u/W_s = 2$</td>
<td>96.41%</td>
<td>95.71%</td>
<td>97.12%</td>
<td>96.62%</td>
</tr>
<tr>
<td>Ensemble CNN, $W_u/W_s = 2$</td>
<td>97.51%</td>
<td>98.20%</td>
<td>96.81%</td>
<td>97.31%</td>
</tr>
</tbody>
</table>

Table 11
Prediction results of testing data of the 10-machine system using SVM and 34 features in [50].

<table>
<thead>
<tr>
<th>Classifier</th>
<th>G-mean</th>
<th>Reliability</th>
<th>Security</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>97.82%</td>
<td>97.51%</td>
<td>98.14%</td>
<td>97.92%</td>
</tr>
<tr>
<td>Extended SVM</td>
<td>97.90%</td>
<td>98.20%</td>
<td>97.61%</td>
<td>97.82%</td>
</tr>
</tbody>
</table>

Table 12
Generator sets when removing multiple generators’ information.

<table>
<thead>
<tr>
<th>Generator set</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>${G1, G2, G3, G4, G5, G6, G7, G8, G9, G10}$</td>
</tr>
<tr>
<td>2</td>
<td>${G1, G2, G3, G5, G6, G7, G8, G9, G10}$</td>
</tr>
<tr>
<td>3</td>
<td>${G2, G3, G5, G6, G7, G8, G9}$</td>
</tr>
<tr>
<td>4</td>
<td>${G2, G3, G5, G6, G7, G8}$</td>
</tr>
<tr>
<td>5</td>
<td>${G2, G3, G6, G7, G8, G9}$</td>
</tr>
<tr>
<td>6</td>
<td>${G2, G3, G7, G8, G9}$</td>
</tr>
<tr>
<td>7</td>
<td>${G3, G7, G8, G9}$</td>
</tr>
<tr>
<td>8</td>
<td>${G7, G8, G9}$</td>
</tr>
<tr>
<td>9</td>
<td>${G8, G9}$</td>
</tr>
<tr>
<td>10</td>
<td>${G9}$</td>
</tr>
</tbody>
</table>

Fig. 12. Prediction results when removing one generator’s information from the feature set.
number of training instances.

Scheme 1: Fine-tuning the pre-trained classifier based on the instances selected by the active learning strategies.

Scheme 2: Retraining a new classifier based on the instances selected by the active learning strategies.

Scheme 3: Retraining a new classifier based on the instances randomly selected.

The number of selected labelled instances are 204, 408, 612, 816, ..., 11,832 ($N_{\text{new}} = 204$ in the Algorithm 1). Since the new test data 2 in Table 13 contains more unstable instances, the values of $W_{\text{un}}$ and $W_{\text{s}}$ are set as 1 during the updating process. The performances of classifiers obtained from different learning strategies are shown in Fig. 16.

Fig. 16(a) shows that the accuracies of the training set (including unlabelled instances) of Scheme 1 and 2 are higher than that of Scheme 3, therefore, active selection strategy can select representative instances from unlabelled data. Fig. 16(b) shows that the prediction accuracy of Scheme 1 using the classifier trained by only 816 labelled instances is 98.01%. Therefore, only 816 labelled instances are needed to be obtained by long-term simulation. In contrast, Scheme 2 and Scheme 3 cannot achieve this level of performance even when using all the instances. As a result, the proposed method can obtain an accurate classifier using fewer instances, thereby dramatically reducing the computational time, including the time for time-domain simulation and classifier training.

To investigate the distribution of the instances selected by three schemes, the t-distributed stochastic neighbour embedding (T-SNE) [51] is used to provide the visualization of the top 816 instances selected by different schemes, as shown in Fig. 17. The high-dimensional features of the top 816 instances selected by different schemes are reduced into two dimensions by T-SNE. There are some overlapping areas in the visualization results. The overlapping is the main reason for classification difficulties, and the instances in overlapping area are more important to obtain the new classifier. Compared with the Scheme 2 and Scheme 3, shown in Fig. 17, Scheme 1 can select more instances in the overlapping areas, allowing the proposed method to obtain an accurate classifier with fewer training instances.

Table 15 shows the detailed results of different schemes using 816 instances and those of Scheme 3 using all 11,832 instances.

Table 15 shows that the accuracy, reliability, security and G-mean of Scheme 1 are all the highest. Most importantly, the total computation time $T_{\text{total}}$ of Scheme 1 (including the selection of important instances, training time using different numbers of instances during the iteration in algorithm 1) is satisfied for practical application. The training time of Scheme 2 is longer than that of Scheme 3 because the active learning should select important instances and train multiple classifiers using different numbers of instances. The fine-tuning technique enables the total training time to be dramatically reduced, making the proposed method quite suitable for online updating. Moreover, parallel computation techniques can be used with the proposed ensemble classifier to further reduce the training time.

On the other hand, data generation is a very time-consuming step in the classifier updating process. To obtain an instance with label, the time-domain simulation should be conducted. However, in the proposed approach, only the selected instances are needed to annotate a label (shown in Fig. 18). Therefore, the computation time of time-domain simulation can be significantly reduced, which makes the proposed approach more suitable for practical application.

(2) NPCC 48-machine 140-bus system

An additional 74,520 instances are randomly generated with the same configuration as in the new testing set 2 in Table 14. The fast updating process described in Section 4.2 is used to obtain the new classifier using these new instances. Three different learning strategies in Section 5.5(1) are considered to compare the effectiveness. Since the new test data 2 in Table 14 contains less unstable instances, setting $W_{\text{un}} = 3$ and $W_{\text{s}} = 1$ during the updating process. The performances of classifiers obtained using different learning strategies and different numbers of labelled instances are shown in Fig. 19.

Fig. 19(b) shows that the accuracy of the classifier on the test data is up to 98% when using only 2070 (2.78%) labelled training instances. In contrast, the performances of Scheme 2 and 3 are not satisfactory. Therefore, the proposed strategy can obtain an accurate classifier using fewer labelled instances with a shorter computational time.

Table 16 shows that detailed results of different schemes using 2070

![Fig. 14. Prediction accuracy of the new testing data in Table 13 using different classifiers.](image-url)
instances and Scheme 3 using all the 74,520 instances.

Table 16 reveals that the performance of Scheme 1 is the best, exhibiting the shorter total computation time and best evaluation index. Therefore, the proposed method is highly suitable for online application.

5.6. Discussions

(1) Selection of the input features

The reasonability of the input features has a great influence on the performance of a transient stability classifier [43]. It seems that the input features for transient stability prediction are always different in different studies. There is not a universal feature set for transient stability prediction.
stability prediction. And it is hard to say that any features can always perform the best in any cases.

In practical, the selection of input features should consider the locations of measurement units. On the other hand, some variables have strong correlation with the transient stability status. If the important measurements cannot be obtained, the prediction performance will reduce significantly. Therefore, new measurement units should be installed at new locations. It can be said that the selection of input features and the location of measurement units interact each other. Moreover, the input features are not fixed as the development of science and technology. Some variables may become measurable with the development of measurement technology. These variables should be added as the input features. On the other hand, when new progress has been achieved in the mechanism analysis of transient stability, new features should be utilized as the inputs for the data-driven methods.

(2) Considering new contingencies

For the pre-contingency application, only the credible contingencies are taken into consideration for the sake of economy. For the post-contingency application, whereas, the number of possible disturbances is infinite. Therefore, the generated classifier should possess good generalization for different contingencies. To obtain a CNN ensemble classifier with good performance under different contingencies, a comprehensive dataset should be obtained during the training process. Because the post-fault responses carry information about the influence of the faults on the power system, the generated classifier may have generalization with respect to similar contingencies. On the other hand, some contingencies can be predicted by the information of weather, equipment monitoring, forest fire prevention information, etc. Based on the new dataset considering new contingences, the CNN ensemble classifier can be updated to have good performance considering these new contingencies.

(3) Effect of stability criterion

In this research, the transient stability criterion is whether the difference between any two generator rotor angles exceeds 360° with 4 s after the fault clearance. It is reasonable because the time frame of interest in transient stability studies is usually 3–5 s [2]. However, the power system may lose the synchronous through multi swings after a longer time period. Under this condition, the criterion with shorter time period will label those multi-swing unstable instances as stable. Different from the model-based methods start from the model solving or theoretical analysis, the data-driven method is learning from massive data. When the criterion considers longer time of period, the criterion can be reflected on the label and then learned by the data-driven method. To verify this viewpoint, case studies are conducted on the 10-machine system using 10 s criterion, and the results still show that the proposed CNN ensemble classifier performs the best among the compared methods and some multi-swing instabilities can be detected. However, similar to the results in Fig. 11, some unstable instances with longer instability occurrence time cannot be detected with shorter response time. To balance the rapidity and accuracy, the hierarchical scheme in [43] should be used.

(4) Considering different types of stability problems

The power system stability status should be determined by comprehensive consideration of different kinds of stability. In practical
The stability criterion in this research can describe the instability phenomenon, but this criterion cannot reveal the instability mechanism. In practical application, for example, the voltage and rotor angle instability are always intertwined. When the power systems show several instability characteristics, addition method such as [53] should be used to recognize the dominant instability mechanism.

6. Conclusions

This paper presents a CNN-based ensemble classifier and its fast updating scheme for transient stability prediction. Through comprehensive comparisons among different methods of two test power systems, the following conclusions can be drawn:

1. The proposed CNN-based ensemble classifier can process multi-dimensional data directly and provide accurate prediction even under the measurement errors. For unbalanced data with fewer unstable instances, the reliability (accuracy of unstable instances) of the proposed classifier can be improved by improving the cost weighting of the unstable instances.

2. When the operating conditions change substantially, the proposed updating method can obtain an accurate classifier using only a few labelled instances. The computational time of both the time-domain simulation and classifier training can be greatly reduced, making the proposed method more suitable for online application.

Acknowledgment

The authors gratefully acknowledge the financial supports from the National Key R&D Program of China (2018YFB0904500), the China Postdoctoral Science Foundation Funded Project (2018M630156), the National Natural Science Foundation of China (51577009), and the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References


