

## Article

# A Hierarchical Self-Adaptive Method for Post-Disturbance Transient Stability Assessment of Power Systems Using an Integrated CNN-Based Ensemble Classifier

# Ruoyu Zhang <sup>1,\*</sup>, Junyong Wu <sup>1,\*</sup>, Yan Xu <sup>2</sup>, Baoqin Li <sup>1</sup> and Meiyang Shao <sup>1</sup>

- <sup>1</sup> School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China
- <sup>2</sup> School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore
- \* Correspondence: 15117385@bjtu.edu.cn (R.Z.); wujy@bjtu.edu.cn (J.W.); Tel.: +86-185-1330-8095 (R.Z.); +86-139-1006-1748 (J.W.)

Received: 2 July 2019; Accepted: 13 August 2019; Published: 21 August 2019



Abstract: Data-driven approaches using synchronous phasor measurements are playing an important role in transient stability assessment (TSA). For post-disturbance TSA, there is not a definite conclusion about how long the response time should be. Furthermore, previous studies seldom considered the confidence level of prediction results and specific stability degree. Since transient stability can develop very fast and cause tremendous economic losses, there is an urgent need for faster response speed, credible accurate prediction results, and specific stability degree. This paper proposed a hierarchical self-adaptive method using an integrated convolutional neural network (CNN)-based ensemble classifier to solve these problems. Firstly, a set of classifiers are sequentially organized at different response times to construct different layers of the proposed method. Secondly, the confidence integrated decision-making rules are defined. Those predicted as credible stable/unstable cases are sent into the stable/unstable regression model which is built at the corresponding decision time. The simulation results show that the proposed method can not only balance the accuracy and rapidity of the transient stability prediction, but also predict the stability degree with very low prediction errors, allowing more time and an instructive guide for emergency controls.

**Keywords:** Transient stability assessment (TSA); intelligent system (IS); convolutional neural network (CNN); ensemble learning; hierarchical self-adaptive method; integrated decision-making rule; power systems

## 1. Introduction

Transient stability or large-disturbance rotor angle stability is referred to as the ability of an interconnected power system to maintain synchronism when subjected to a large disturbance, such as a three-phase short-circuit fault on a transmission line [1]. Transient instability may develop into catastrophes, such as cascading failures and/or widespread blackouts. Therefore, transient stability assessment (TSA) has significant importance in security monitoring of power systems. It is an essential requirement to maintain transient stability in power system operation.

To achieve rapid real-time TSA, the data-driven artificial intelligence (AI) approach [2–7] was identified as a novel and promising solution. First of all, through offline training on massive sets of data generated through time-domain (T-D) simulation, it can capture the potential useful knowledge to map the relationship between inputs (features of power system) and outputs (the corresponding dynamic security indices, such as stability status or stability margin/degree). Then, when it comes to online



application, the outputs can be given instantaneously once the inputs are available. Compared with other traditional TSA methods, such as T-D simulation [8], transient energy function method [9], and extended equal-area criterion [10], the merits of the data-driven AI approach are rapid decision-making speed, fewer system physical parameters required, strong generalization capability, and powerful data mining ability.

Post-disturbance TSA is different from pre-disturbance TSA. It predicts the system future stability status under an ongoing disturbance by using post-disturbance dynamic features such as variations of the rotor angles/speeds/accelerations/kinetic energy [11–13], voltage magnitudes [4,13,14], electromagnetic power [11,13], etc. On the other hand, the pre-disturbance TSA evaluates the current stability status for an anticipated fault which is yet to occur by using pre-disturbance steady-state power system operating variables such as line flow, P, Q load, P, Q generation of each generator [2,3], etc. The assessment result of post-disturbance TSA is used to trigger emergency controls to impede the propagation of instability, while the pre-disturbance TSA is utilized to activate preventive controls to preventively modify the system operating condition to withstand those possible potential severe disturbances.

This paper focuses on post-disturbance TSA, which has stricter requirements for prediction accuracy and speed because a fault is ongoing. Thanks to the development of wide-area measurement systems (WAMS), the dynamic response of power system operating conditions after a fault can be monitored by the widely deployed phasor measurement units (PMUs), enabling a data-driven AI approach to real-time post-disturbance TSA [5,14,15]. Recently, the main research focused on generating training and testing databases, feature selection, and improving the learning algorithms to increase prediction accuracy; however, it seldom considered the reliability or credibility of the prediction result and specific stability degree. It is a tough challenge to perfectly eliminate classification/prediction errors by using an AI learning algorithm. Thus, the prediction result may be wrong if the TSA error remains unavoidable. It is better to identify those unreliable prediction results as uncertain ones temporarily rather than use such wrong prediction results. In addition, the existing data-driven AI methods tend to use fixed duration times of the response data as predictors. They vary from 50 ms to 3 s [3,11,13,14,16] according to different inputs and classifiers. In reality, for different faults with different severity, the instability occurs at different times. Therefore, a constant response time may lead to inefficient decision-making. Concerning the above points, the contribution of this paper consists of developing a novel intelligent system (IS) that can both achieve more efficient and reliable post-disturbance TSA.

The proposed IS was developed based on a popular deep learning network called convolutional neural network (CNN) [11,17–19], which has strong recognition ability for highly nonlinear patterns, learns more useful features automatically, and reduces the number of network structure parameters dramatically. Constructing a CNN-based ensemble learning structure, the IS gathered several CNNs with an integrated decision-making rule. Under a hierarchical self-adaptive method [5,14,20,21], the decision can only be output when the prediction result meets the credibility requirement making the accurate TSA decision at an appropriately early time. Additionally, the regression models for stable and unstable instances are built, such that the transient stability degree of the instances can be predicted accurately, which will give instruction on subsequent emergency control measures.

The main contributions of this paper are summarized as follows:

- A hierarchical self-adaptive framework based on CNN-based ensemble classifiers for post-disturbance TSA is proposed. This framework is verified with high prediction accuracy and reliability, and can balance the accuracy and rapidity of the transient stability prediction.
- (2) A novel integrated decision-making rule based on multiple CNNs is presented. Through the suitable setting of thresholds, it can achieve as high prediction accuracy as possible, and no large false alarm (when predicting an actual stable instance as an unstable instance), with zero misdetection (when predicting an actual unstable instance as a stable instance).

(3) Through regression modeling and predicting of transient stability index (TSI), it provides the stability degree of each instance, which is capable of providing a basis for subsequent emergency control.

The remaining of this paper is organized as follows: Section 2 introduces the generation of the dataset for TSA. Section 3 describes the learning algorithm of the CNN. Section 4 presents the hierarchical self-adaptive IS model for TSA. Comprehensive case studies on a New England 10-machine 39-bus power system are conducted in Section 5. The discussion and conclusions are given in Sections 6 and 7, respectively.

#### 2. Generation of Dataset

In terms of data-driven AI methods for TSA, an offline stage of training on a massive amount of data generated by T-D simulations needs to be done at the first step. The methods aim to map the relationship between the inputs (i.e., features) and the outputs (i.e., stability status or stability degree) [4], expressed as follows:

$$y = f(x), \tag{1}$$

where *x* represents the input predictors, namely, features of the power system, *y* denotes the corresponding dynamic security indices, such as stability status or stability degree, and *f* represents the mapping function. The training dataset can be simulated via T-D simulation on a certain power system with various load/generation patterns, network topologies, and fault types, locations, and duration times. TSI [4,22] is defined as follows:

$$TSI = \frac{360^{\circ} - |\Delta\delta|_{max}}{360^{\circ} + |\Delta\delta|_{max}},$$
(2)

where  $|\Delta\delta|_{max}$  is the maximum of the absolute value of the angle separation between any two generators. For the classification of transient stability status in this paper, when TSI > 0, the system is considered as stable and vice versa. For the regression of transient stability degree in this paper, a continuous value of TSI is used as the label *y*. For a stable instance, the TSI is between 0 and 1. The greater the value of TSI is, the more stable the system becomes. For an unstable instance, the TSI is between -1 and 0. The smaller the value of TSI is, the more unstable the system becomes.

Since the label of each simulation instance is determined by TSI, which is calculated in terms of generator rotor angle, the rotor angle-related variables such as rotor angle/speed/acceleration and kinetic energy are usually used as the input features [11–13]. The trajectory of relative electromagnetic power, which represents the ratio of the electromagnetic and mechanical power, also contains abundant information of system transient stability status, because it reflects the recovery of the electromagnetic power after the fault clearance [13]. In the literature [4,14], the voltage magnitudes of all the generator buses are often used as inputs. In this research, the dynamic trajectories of voltage magnitude, relative rotor angles, speeds, accelerations, and kinetic energy, electromagnetic power, and relative electromagnetic power of all the generators are used as the inputs.

The input feature types are described in Table 1, where *n* is the number of total generators in the test system, and *s* is the number of feature sampling times, which reflects the response time after fault clearance. When s = 1, t1 represents the first sampling time after fault clearing. The calculation of input features and the data normalization process are explained in detail in Appendix A. Taking feature type 1 for example, the normalized input feature type 1 of each instance can be stored as an array which has a similar data structure to that of an image, as shown in Figure 1. Its two axes named generator and time can be viewed as the width and length axes of the image. This characteristic inspired us to apply CNN, a widely recognized deep learning method for image classification, to extract the mapping knowledge for TSA.

Feature Type	Feature Name	Feature Number	Description
1	Voltage magnitude	$k \times n$	$U_i(t_s), i = 1,, n; s = 1,, k$
2	Relative rotor angle	$k \times n$	$\tilde{\delta}_i(t_s), i = 1, \ldots, n; s = 1, \ldots, k$
3	Relative rotor speed	$k \times n$	$\tilde{\omega}_i(t_s), \ i = 1,, n; s = 1,, k$
4	Relative rotor acceleration	$k \times n$	$\tilde{\alpha}_i(t_s), \ i = 1,, n; s = 1,, k$
5	Relative kinetic energy	$k \times n$	$EK_i(t_s), i = 1,, n; s = 1,, k$
6	Electromagnetic power	$k \times n$	$P_{ei}(t_s), i = 1,, n; s = 1,, k$
7	Relative electromagnetic power	$k \times n$	$\tilde{P}_{ei}(t_s), \ i = 1,, n; s = 1,, k$

Table 1. Descriptions of different input features.



Figure 1. Time series of voltage magnitude of all the generators.

## 3. Convolutional Neural Network

As a well-known deep learning method, CNN works as a deep structure feedforward neural network. It consists of an input layer, multiple hidden layers, and an output layer. There are three types of common construction of hidden layers: convolutional layer, pooling layer, and fully connected layer. Referring to the LeNet-5 [23], it can be found that the order of the layers in a common structure of CNN is as follows: input layer–convolution layer–pooling layer–convolution layer–pooling layer–fully connected layer–output layer, as illustrated in Figure 2. The difference in the forward propagation process between the convolution layer and fully connected layer is the convolutional neuron, which shares the same weights as each receptive field (i,j), greatly reducing the number of parameters of network. The output g(n) of a convolution layer with input location (i,j) is as follows:

$$g(n) = f((\mathbf{W} \cdot \mathbf{X})_{i,i} + \mathbf{b}), \tag{3}$$

where W is the weight of a filter, b is the bias, " $\cdot$ " is the dot product, representing the sum of the product of the corresponding elements in the matrix, and *f* is the activation function Rectified linear unit (ReLU), where ReLU(*x*) = max(0, x). The forward propagation process of the convolutional layer is described in Appendix B which gives a simple example.



Figure 2. Common structure of a convolutional neural network (CNN).

A pooling layer is often added between the convolutional layers. It can effectively reduce the size of the matrix, thereby reducing the parameters in the final fully connected layer. Using a pooling layer can speed up calculations and prevent over-fitting problems. The most commonly used pooling

layer, max-pooling, is utilized here. In CNN, a larger convolution kernel size corresponds to more free parameters (i.e., weights and bias), which means more computation and more time to train the model. The most commonly used convolution kernel size is  $3 \times 3$ , and the size of max-pooling is usually selected as  $2 \times 2$ . After several convolutional and pooling layers, the output node matrix should be flattened and connected to a fully connected layer, whereby the calculation process is the same as an artificial neural network (ANN) with ReLU as the activation function. The training algorithm of the CNN used in this paper is outlined in Appendix C [24]. The final output layer uses the softmax function [11,23], described as follows:

$$P(C_k|X) = \frac{e^{y_k(X)}}{\sum_{k=1}^2 e^{y_k(X)'}}$$
(4)

where  $y_k(X)$  is the *k*-th input data of the softmax layer with instance *X*, and  $P(C_1|X)$  and  $P(C_2|X)$  are the probabilities of instance *X* identified as category-1 and category-2 for a binary classification problem (i.e., transient stability status prediction). The final output of CNN is  $\hat{y}_i = (P(C_1|X), P(C_2|X))$ ; if  $P(C_1|X) > P(C_2|X)$ , the instance will be identified as category-1 and vice versa.

In previous literature [11,17–19], it was validated that the CNN is effective on a number of benchmark models and actual life problems in both classification and regression fields. It shows stronger recognition ability for highly nonlinear patterns, learns more useful features automatically from a massive amount of time-series data, dramatically reduces the number of network structure parameters, and has better generalization capacity than some conventional techniques [25–27].

#### 4. The Proposed IS Model

In this section, following the basic idea illustrated in Section 3, we develop an IS model for TSA based on CNN. The workflow chart of the proposed IS model is shown in Figure 3.



Figure 3. The workflow chart of the proposed intelligent system (IS) model.

## 4.1. CNN-Based Ensemble Learning

Ensemble learning was widely applied in many research fields, such as machine learning [28,29], electrical power systems [3,11,20], computational biology [30–33], and so on. It is an effective strategy to increase accuracy. Thanks to the diversity of individual learners, the single learners can adequately compensate for each other and tend to reduce aggregate variance, achieving better results. Different dynamic response trajectories of power systems contain rich dynamic information. Making full use of this information helps better predict the transient stability of power systems. However, the

determination of a better classifier with favorable input features can only be done by repeated time-consuming trials. Moreover, it is hard to say the selected features are versatile for all application scenarios. With this in mind, different CNNs with the same model structure but different types of features as inputs were trained to construct an ensemble classifier. The obtained single CNN models were different even with the same network structure and learning rate, because the input features were different and the initial weights and bias were randomly selected. With this stochastic nature, comprehensively utilizing the diversity of these single classifiers can further improve classification performance. On the basis of previous work [3,11,14,20], this paper proposes a novel IS based on CNN-based ensemble classifiers to more efficiently achieve post-disturbance TSA of power systems. As mentioned in Section 2, seven kinds of input features are utilized, which means that there are seven individual CNN classifiers, and each is constructed using an individual input feature type. The ensemble classifier consisting of m (m = 1, 2, ..., 7) different individual CNNs is defined as type m. The performance of different types of classifiers is analyzed in Section 5.4.

## 4.2. Integrated Decision-Making Rule

Different from most ensemble models, which determine the final output by voting the majority or average value of the individual outputs [28,29,34], a tailored integrated decision-making rule is proposed in this paper. Each individual CNN classifier exports two probabilities  $\hat{y}_n = (\hat{y}_n^{(1)}, \hat{y}_n^{(2)})$ , where  $\hat{y}_n^{(1)} + \hat{y}_n^{(2)} = 1$  [11,23]. They represent the confidence of instance X belonging to each class. The specific integrated decision-making rule [35] for ensemble classification is described in Algorithm 1.

#### Algorithm 1: Integrated decision-making rule for CNN-based ensemble classification

Given m single CNNs with different types of input features, CNN1, CNN2, ..., CNNm, whose first outputs are represented by  $\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \ldots, \hat{y}_m^{(1)}$ : If  $\min(\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \ldots, \hat{y}_m^{(1)}) > \alpha$ , then it is a credible stable instance; Else if  $\max(\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \ldots, \hat{y}_m^{(1)}) < \beta$ , then it is a credible unstable instance; Else it is an uncertain instance at the current decision time, where  $\alpha$  and  $\beta$  are user-defined thresholds to judge whether the instance is credible stable/unstable or an uncertain instance. end

With the integrated decision-making rule, the ensemble classifier can identify the credible stable/unstable instances and uncertain instances. For those credible instances, they are sent into the corresponding regression model to regress TSI, which reflects their stability degree. Those uncertain instances need further identification at the next TSA decision time. The stable regression model and the unstable regression model were built based on CNN by training a stable dataset and unstable dataset, respectively, with inputs corresponding to time series of post-disturbance voltage magnitude trajectories of the duration from the first post-disturbance sampling to different decision times.

#### 4.3. Hierarchical Self-Adaptive Method for TSA

The structure of the proposed hierarchical self-adaptive IS is described in Figure 4. There are a series of individual CNN-based ensemble classifiers, each performing the TSA at a different response times (note that each of the classifiers is trained using different time series of features mentioned above from the first post-disturbance sampling time to the corresponding response time). For an ensemble classifier at response time t =  $T_s$ , only when the minimum value of these first outputs of these single CNN classifiers is greater than  $\alpha$  can this instance be determined as credible stable. Only when the maximum value of these first outputs of these single CNN classifiers is smaller than  $\beta$  can this instance be determined as credible unstable; then, the IS delivers the transient stability prediction result at t =  $T_s$ . Otherwise, this instance should be determined as an uncertain instance and the classification

continues at time  $t = T_{s+1}$ . The instances far away from the classification boundary are easier to identify even at a very short response time. As time goes by, the dynamic characteristics of the power system become more and more obvious and separable. Thus, the uncertain instances are recognized as credible instances at longer response times. Therefore, this proposed hierarchical self-adaptive method allows balancing the rapidity and accuracy of transient stability prediction. Then, the instances that were identified as credible stable and credible unstable instances can be predicted using the stable regression model and the unstable regression model, respectively.



**Figure 4.** The process of the hierarchical self-adaptive CNN-based ensemble classifier for post-disturbance transient stability assessment (TSA).

## 5. Case Study

The developed IS was tested on a popular benchmark system: IEEE 10-machine 39-bus system. Detailed parameters of this system can be found in Reference [36]. The system frequency was 60 Hz. The IS network was implemented in TensorFlow, a state-of-the-art open-source machine learning framework, by using the computer programming language of Python [37]. Experiments were carried out on a 64-bit personal computer with an Intel Core i5-7200U central processing unit and 4.00 GB of random-access memory. The T-D simulations were conducted by MATLAB Power System Toolbox 3.0.

#### 5.1. Data Generation

During the generation of the dataset, the system generation and load patterns were randomly varied within the 75–120% level of the initial operation condition. The contingencies were mainly three-phase permanent short-circuits at ten locations ranging from the beginning to 90% of the length of each transmission line with an increment of 10%. The simulation time was 5 s, and the simulation step was 0.0167 s (one cycle was 0.0167 s for the 60-Hz system). There were 11 assumed types of failure duration times, ranging from 0.0167 s to 0.1837 s with an increment of 0.0167 s. For this test system, 37,400 instances including 24,864 stable instances and 12,536 unstable instances were generated. For the proposed IS model, 22,400 instances were selected for model training, 5000 instances were selected to build the validation dataset which was used for model selection and prevented over-fitting, and the remaining 10,000 instances were selected to form the testing dataset.

#### 5.2. Indices for Performance Evaluation

Transient stability prediction is a typical imbalanced classification task, where the number of stable cases is more than that of unstable cases. In the meantime, the costs of misdetections and false alarms

are quite different. It is much more serious to misclassify an unstable instance as a stable instance. If a stable case is misclassified as unstable, it may cause malfunction of some control devices, but it has little effect on the safe and stable operation of the entire system. If an unstable case is misclassified as stable, no measure is taken to prevent the instability accident, which causes tremendous financial losses and disastrous consequences. Therefore, the accuracy of the overall dataset does not quite reflect the classification performance of the classifier. With this in mind, in addition to the accuracy of the whole dataset, a new index named HRP (harmonic mean of recall and precision) [4] is introduced to better evaluate the performance of unstable instance identification. It was modified to focus on the unstable instances. According to the confusion matrix of TSA shown in Table 2, the related indices are defined as follows:

• Accuracy, abbreviated as ACC, is the proportion of instances that are correctly predicted by the classifier.

$$ACC = \frac{(T_s + T_{us})}{(T_s + T_{us} + F_s + F_{us})}$$
(5)

• HRP is the abbreviation of harmonic mean of recall and precision. It is commonly used in evaluation classifiers.

$$\operatorname{recall} = \frac{T_{us}}{(T_{us} + F_s)},\tag{6}$$

$$precision = \frac{T_{us}}{(T_{us} + F_{us})},$$
(7)

$$HRP = 2 \times recall \times \frac{precision}{(recall + precision)} = \frac{2}{\frac{1}{recall} + \frac{1}{precision}}$$
(8)

Table 2. Confusion matrix of transient stability assessment (TSA).

<b>Confusion Matrix</b>	Stable (Actual)	Unstable (Actual)
Stable (Predicted)	$T_s$	$F_s$
Unstable (Predicted)	Fus	$T_{us}$

#### 5.3. Implementation Details

In this research, each individual CNN classifier consisted of two convolution layers, two pooling layers, and two fully connected layers. The detailed parameters of each individual CNN classifier are shown in Table A1 (Appendix C). Both the fully connected layer 1 and fully connected layer 2 use dropout technology with the dropout rate as 0.5. For specific details of dropout technology, the reader can refer to Reference [38]. Each individual CNN classifier was trained with a mini-batch stochastic gradient descent with an exponential decay learning rate. Its pre-set initial learning rate was set to 0.01 with an attenuation coefficient of 0.99. The exponential decay learning rate allows the model to quickly approach the best solution in the early stage of training, without too much fluctuation in the later stage of training. Thus, it will be much closer to the local optimum. For specific details of the exponential decay learning rate, the reader can refer to Reference [37]. The scripts of the proposed hierarchical self-adaptive post-disturbance TSA method and the training process of a single CNN are described in Appendix D, where Figure A2 illustrates the processes of offline training and online application of the proposed method.

#### 5.4. Parameter Determination

In this test, it was assumed that the generator voltage phasor measurements, rotor angles, electromagnetic power, and mechanical power were sampled at a rate of one sample per cycle. Therefore, we trained each individual CNN classifier with different input trajectories during the period of the first sampling time after the clearance of the fault for each response time. To build the

CNN-based ensemble classifier, the number of selected features m in Figure 4 should be determined. It is helpful to evaluate the performance of the CNN-based ensemble classifier without the integrated decision-making rule, just using the average value of the individual outputs [11]. We randomly selected m (m = 1, 2, ..., 7) individual CNNs from the seven kinds of individual CNNs to construct classifier type m and calculated the ensemble performance. Then, this operation was repeated many times. Lastly, the average value of these performances was calculated as the performance of classifier type m. The performances of these seven types of classifiers defined in Section 4.1 are shown in Figure 5. The values of the two evaluation indices ACC and HRP obviously increased with the increment of the response time. In addition, the type 7 classifier performed the best among the others. That means that the ensemble classifier with all seven individual CNN classifiers was a good choice. Therefore, m was set as 7 in this paper.



**Figure 5.** Performance of different types of CNN-based ensemble classifiers with different response cycles: (**a**) accuracy (ACC); (**b**) harmonic mean of recall and precision (HRP).

After determining the specific composition of the ensemble CNN, we further propose a hierarchical self-adaptive method using an integrated decision-making rule to improve the credibility of prediction results and balance the trade-off between TSA speed and accuracy. Before analyzing the results, for the sake of convenience, several definitions are introduced.  $T_i$  is the *i*-th decision time; CS(T) and CU(T) are the total numbers of credible stable instances and credible unstable instances as of the current decision time, respectively; U(T) and  $U_r(T)$  are the total number of uncertain instances and the rate of uncertain instances (the percentage of the uncertain instances with respect to the total testing instances) as of the current decision time;  $M(T_i)$  and M(T) are the total numbers of false alarms at and as of the current decision time, respectively;  $A(T_i)$  and A(T) are the current and accumulative TSA accuracy at and as of the current decision time, respectively;  $A(T_i)$  and A(T) are the current and accumulative TSA accuracy

$$A(T_i) = \frac{[CS(T_i) + CU(T_i) - M(T_i) - F(T_i)]}{[CS(T_i) + CU(T_i)]},$$
(9)

$$A(T) = \frac{[CS(T) + CU(T) - M(T) - F(T)]}{[CS(T) + CU(T)]}$$
(10)

The credibility estimation parameters  $\alpha$  and  $\beta$  in the integrated decision-making rule for CNN-based ensemble classification mentioned in Algorithm 1 are very important, since they directly determine the performance of the IS. The value of parameter  $\alpha$  is usually set as 0.5 to 0.95, and  $\beta$  is usually set as 0.05 to 0.5. We now use the control variable method to observe the effect of varying  $\alpha$  while fixing  $\beta$  as 0.05. The results are shown in Figure 6. We also observed the effect of varying  $\beta$  while fixing  $\alpha$  as 0.95. The results are shown in Figure 7. The results are based on response time *t* equal to 30 cycles (0.5 s).



**Figure 6.** Impact of  $\alpha$  on classification results ( $\beta = 0.05$ ): (**a**) rate of uncertain instances  $U_r(T)$ ; (**b**) accumulative accuracy A(*T*); (**c**) count of F(*T*) and M(*T*).



**Figure 7.** Impact of  $\beta$  on classification results ( $\alpha = 0.95$ ): (a) rate of uncertain instances  $U_r(T)$ ; (b) accumulative accuracy A(*T*); (c) count of F(*T*) and M(*T*).

It can be observed in Figure 6 that, with  $\alpha$  increasing, the rate of uncertain instances  $U_r(T)$  and the accumulative accuracy A(T) increased, the count of F(T) remained unchanged as three, and the count of M(T) decreased to 0 when  $\alpha$  increased to 0.9; the reason is that a larger  $\alpha$  means the credibility estimation criterion of stable instances is more strict; thus, there tend to be more classifications being evaluated as uncertain instances, while those classified as credible stable instances become much more accurate with less misdetection. According to Figure 7, it is shown that, with  $\beta$  increasing, the rate of uncertain instances  $U_r(T)$  and the accumulative accuracy A(T) reduced, the count of M(T) remained unchanged as zero, and the count of F(T) increased; the reason is that a larger  $\beta$  means the credibility estimation criterion of unstable instances is looser; thus, there tend to be fewer classifications being evaluated as uncertain instances, while those classified as credible unstable instances become not that accurate and, as such, the count of F(T) increases. As mentioned before, the costs of misclassification of unstable instances (misdetection) are enormous and unacceptable for real-time utilization. Thus, combining the impact of these two parameters on the classification results of TSA, we chose parameters with high accumulative accuracy, a low uncertain instance rate, few false alarms, and zero misdetection. Therefore, in this paper,  $\alpha$  and  $\beta$  were set as 0.9 and 0.1, respectively, achieving 99.97% accumulative accuracy, 8.87% uncertain instance rate, zero misdetection, and three false alarms at response time t equal to 30 cycles (0.5 s). The results are shown in Table 3.

i	Response Time (s)	Classified as Stable		Classified as Unstable			$A(T_i)$	A( <i>T</i> )		<b>U</b> <sub><i>r</i></sub> ( <i>T</i> )	
(Cycles)		CS(T)	<b>M(T</b> <sub><i>i</i></sub> )	<b>M(</b> <i>T</i> <b>)</b>	CU(T)	<b>F</b> ( <i>T<sub>i</sub></i> )	F( <i>T</i> )	(%)	(%)	0(1)	(%)
0	0	-	-	-	-	-	-	-	-	10,000	100
3	0.05	4265	0	0	1991	0	0	100	100	3744	37.44
6	0.10	516	0	0	251	0	0	100	100	2977	29.77
9	0.15	363	0	0	271	2	2	99.68	99.97	2343	23.43
12	0.20	276	0	0	214	1	3	99.80	99.96	1853	18.53
15	0.25	255	0	0	101	0	3	100	99.96	1497	14.97
18	0.30	110	0	0	34	0	3	100	99.97	1353	13.53
21	0.35	174	0	0	8	0	3	100	99.97	1171	11.71
24	0.40	42	0	0	22	0	3	100	99.97	1107	11.07
27	0.45	84	0	0	10	0	3	100	99.97	1013	10.13
30	0.50	102	0	0	24	0	3	100	99.97	887	8.87

Table 3. Prediction results of the hierarchical self-adaptive method.

## 5.5. Hierarchical Self-Adaptive Method for Transient Stability Prediction

To balance the rapidity and accuracy of TSA, the hierarchical self-adaptive method was used. The response times of each layer were set to three cycles (0.05 s), six cycles (0.10 ms), nine cycles (0.15 ms),  $\dots$ , 30 cycles (0.50 s), which can be adjusted in different situations. The prediction results are shown in Table 3. It is illustrated in Table 3 that the test instances can be identified at an earlier time with high accuracy, and there is no misdetection. In total, 4265 instances and 1991 instances out of 10,000 instances were classified as credible stable instances and credible unstable instances, respectively, at the first layer with response time as 0.05 s, and the accuracy was as high as 100%, without any misdetections and false alarms. The remaining 3744 uncertain instances moved to the next layer with response time as 0.10 s, where 516 and 251 of them were classified as credible stable instances and credible unstable instances, respectively, with an accuracy of 100%. Then, the remaining 2977 uncertain instances moved to the third layer with response time as 0.20 s, where 363 and 271 of them were classified as credible stable instances and credible unstable instances, respectively, with an accuracy of 99.68% (i.e., two false alarms). There still remained 887 uncertain instances at a response time of 0.50 s, of which 455 instances were unstable. Further observation of uncertain unstable instances is done through comparing the instability occurrence time, which is calculated by the difference between fault clearance and unstable occurrence. Figure 8 shows the instability occurrence time histogram for the total testing of unstable instances and uncertain unstable instances.



**Figure 8.** Histogram of instability occurrence time: (**a**) total testing unstable instances; (**b**) uncertain unstable instances remaining at a response time of 0.50 s.

For the sake of easy observation and comparison, the red line in Figure 8a illustrates the maximum vertical axis value in Figure 8b. It can be observed in Figure 8a that the majority of unstable instances in the total testing dataset had shorter instability occurrence times. Figure 8b shows that the instability of the remaining uncertain unstable instances occurred more than 3 s after the clearance of the faults. This implies that the proposed self-adaptive method can rapidly identify a large number of instances which are far away from the classification boundary. The sooner it identifies unstable instances, the more time will be reserved for emergency control. Those instances with longer instability times are always critical unstable instances near the classification boundary. It is very difficult to identify these instances quickly and accurately with the existing approaches. It is more reasonable to identify them as uncertain ones temporarily, rather than directly deliver wrong prediction results. Therefore, they need further identification at the next decision time. As time goes by, the dynamic characteristics of the power system become more and more obvious and separable. Thus, the uncertain instances are recognized as credible instances at longer response times.

## 5.6. TSI Regression Results

Through the above hierarchical self-adaptive transient stability prediction, credible stable and unstable instances were exported at each decision time. Then, the TSIs of credible stable and unstable instances were regressed using the stable regression model and unstable regression model, respectively. TSI is the transient stability index defined in Equation (2) reflecting the stability degree of power systems. The regression mean squared errors (MSEs) defined in Equation (A10) (Appendix C) at different decision times of the stable regression model for the total testing stable instances and the unstable regression model for the total testing unstable instances are shown in Table 4.

i (Cycles)	Response Time (s)	Total Testing Stable Instances	Total Testing Unstable Instances
0	0	-	-
3	0.05	0.0021	0.0103
6	0.10	0.0019	0.0101
9	0.15	0.0019	0.0098
12	0.20	0.0018	0.0095
15	0.25	0.0011	0.0045
18	0.30	0.0017	0.0092
21	0.35	0.0016	0.0091
24	0.40	0.0015	0.0088
27	0.45	0.0015	0.0083
30	0.50	0.0014	0.0080

**Table 4.** Mean squared errors (MSEs) at different decision times of total testing stable instances and unstable instances.

It can be observed in Table 4 that both the regression MSEs of the stable and unstable regression models had the trend of reducing with the increment of response time. The TSIs of the testing credible stable instances obtained until 30 post-disturbance cycles were predicted using the stable regression model. Figure 9 illustrates the detailed prediction results for the stable regression model at a response time of 0.50 s. It can be found in Figure 9a,b that the prediction results of the testing credible stable instances were closer to the actual TSI value than that of the total testing stable instances. The MSE of the testing stable instances. At the same time, the TSIs of the testing credible unstable instances obtained until 30 post-disturbance cycles were predicted using the unstable regression model. Figure 10 illustrates the detailed prediction results for unstable regression model. Figure 10 illustrates the detailed prediction results for unstable regression model. Figure 10 is found in Figure 10a,b that the prediction results of the testing credible unstable instances were closer to the actual TSI value than that of the total testing credible unstable instances obtained until 30 post-disturbance cycles were predicted using the unstable regression model. Figure 10 illustrates the detailed prediction results for unstable regression model at a response time of 0.50 s. It can be found in Figure 10a,b that the prediction results of the testing credible unstable instances were closer to the actual TSI value than that of the total testing unstable instances. The MSE of the testing credible unstable instances was 0.0023 which is much smaller than the 0.0080 for the total testing credible unstable instances was 0.0023 which is much smaller than the 0.0080 for the total testing credible unstable instances was 0.0023 which is much smaller than the 0.0080 for the total testing credible unstable instances was 0.0023 which is much smaller than the 0.0080 for the total testing credible unstable instances was 0.0023 whic

testing unstable instances. As mentioned in Section 5.5, those uncertain instances are always critical instances near the classification boundary. It is very difficult to identify these instances quickly and accurately with the existing approaches. Thus, it is more accurate and reasonable to predict the stability degree of the credible instances obtained through the proposed hierarchical self-adaptive method. Therefore, the first step of identifying the credible stable and unstable instances is of great help to the next step of predicting the transient stability degree. This method achieves not only predicting the stability of the instances, but also obtaining the stability degree of each instance, which is instructive for emergency control.



**Figure 9.** Detailed prediction results for the stable regression model at a response time of 0.50 s: (a) detailed prediction results of testing credible stable instances, mean squared error (MSE) = 0.0003; (b) detailed prediction results of total testing stable instances, MSE = 0.0014.



**Figure 10.** Detailed prediction results for unstable regression model at a response time of 0.50 s: (a) detailed prediction results of testing credible unstable instances, MSE = 0.0023; (b) detailed prediction results of total testing unstable instances, MSE = 0.0080.

## 6. Discussion

#### 6.1. Construction and Incompleteness of Input Features

The selection of reasonable input features has a significant impact on the performance of TSA classifiers [11,20]. Different researchers approached transient stability uniquely, and their feature extraction and feature selection methods were different. Therefore, there is no general feature set for TSA, and it is difficult to say that there exists a feature set that always has the best performance for any IS in any situation.

In addition, in practical applications, unexpected PMU failure, communication link delay, signal interruption, cyber-attack, etc. [39,40] limit the availability of the features. An incomplete feature input would detrimentally influence the accuracy of individual learning models or even make the transient stability assessment process unavailable under this condition. Some researchers proposed a decision tree with surrogate (DTWS) to solve missing features [41]. Others used the feature estimation method

to predict the missing data directly [42], or used an emerging deep-learning technique called generative adversarial network (GAN) to address the missing data problem [43].

#### 6.2. Classifier Updating for Performance Enhancement

In practical applications, it is necessary to update the classifier depending on the practical situation for performance enhancement. With the variations of load levels, network topologies, and so on, the operation conditions change very much. Through pre-disturbance TSA simulation, the data for updating can be obtained. However, this will be time-consuming for huge T-D simulations. In addition, if the classifier is retrained by the newly generated comprehensive dataset considering all the uncertainties associated with the new operation conditions, it also be time-consuming, especially for classifiers constructed by deep learning networks. Therefore, in order to solve the time-consuming problem of updates in this paper, active learning and fine-tuning methods can be adapted to select informative and representative instances [11]. Firstly, new instances are generated via short-term simulations. Secondly, these unlabeled instances are predicted by the pre-trained classifier to obtain those import instances for the process of updating. Only those judged to be uncertain by the integrated decision-making rule mentioned in Section 4.2 will be attached with target labels through a long-term simulation. As analyzed in Section 5.5, those uncertain instances are close to the classification boundary and are relatively indistinguishable. Thus, they are important instances for the process of updating. Through fine-tuning the pre-trained classifier using these new labeled instances, the computational time of both the T-D simulation and classifier training can be greatly reduced, making the proposed method more applicable for industry online applications.

## 6.3. Increment of System Size

With the increase of system size, the growth rates of the sample size and feature dimensions are exponential, leading to an increment in computational memory and computation time. However, for statistics, the larger the dataset is, the more sufficient the information is likely to be. With the development of computer technologies like graphics processing units (GPUs) and distributed techniques like the alternating direction method of multipliers (ADMM), the advantages of big data can be reflected. All the sub-classifiers can be trained by GPUs combined with the ADMM algorithm, making the proposed approach more suitable for larger test systems in industry.

#### 6.4. Misdetections and False Alarms

The costs of misdetections and false alarms are definitely different in TSA. The misclassification of stable samples as unstable samples (false alarm) may cause malfunction of some control devices, but it has little effect on the safe and stable operation of the entire system. On the other hand, the misclassification of unstable samples as stable samples (misdetection) may lead to a chain collapse or even catastrophic accidents, and the consequences are very serious. Therefore, the cost of misdetections is greater than that of false alarms. When using a data-driven AI method for TSA, we should not just pursue high accuracy but also focus on the counts of misdetections and false alarms, achieving as high a prediction accuracy as possible, with no large false alarms and zero misdetections. In this paper, through the suitable setting of thresholds, we could achieve 99.97% accumulative accuracy, with three false alarms and zero misdetections for credible instances. This indicates the effectiveness of the proposed hierarchical self-adaptive method and the importance of suitable threshold setting.

#### 7. Conclusions

Based on a well-known deep learning technology called CNN, this paper developed an IS for the rapid, accurate, and reliable post-disturbance TSA of power systems. It can not only export credible accurate classification results, but also provide the stability degree of each instance, which is instructive for emergency control. With the CNN algorithm and ensemble technologies, this IS uses a strategically designed learning and an integrated decision-making rule to achieve a hierarchical self-adaptive method which makes correct transient stability prediction results at an appropriate earlier time. Through the suitable setting of thresholds, it can achieve the goal of a high prediction accuracy, few false alarms, and zero misdetections, allowing more time for emergency control and reducing economic losses. Specifically, this two-step method (the first step is the identification of the credible stable/unstable instances, and the second step is the prediction of transient stability degree) of TSA can avert unreliable results, making all the decided instances accurate. More comprehensive case studies will be done on a larger power grid in the future.

**Author Contributions:** R.Z. was in charge of designing the proposed method, performing the experiments, and writing this paper; this work was further improved and revised with regular feedback from J.W. and Y.X.; B.L. and M.S. contributed to the simulations; all authors read and approved the final manuscript.

Acknowledgments: The authors appreciate the financial support from the National Key R&D Program of China, No. 2018YFB094500, and the National Natural Science Foundation of China, No. 51577009.

Conflicts of Interest: The authors declare no conflicts interest.

## Appendix A

All the input features and data normalization process in Section 2 are explained in the following description.

For a power system with n generators, the COI reference of the rotor angle at  $t = t_s$  after the fault clearance,  $\delta_{COI}(t_s)$ , is:

$$\delta_{COI}(t_s) = \sum_{i=1}^n M_i \delta_i(t_s) / \sum_{i=1}^n M_i$$
(A1)

The *i*-th generator relative rotor angle based on COI reference at  $t = t_s$  is:

$$\tilde{\delta}_i(t_s) = \delta_i(t_s) - \delta_{COI}(t_s) \tag{A2}$$

where  $M_i$  is the inertia coefficient of the *i*-th generator;  $\delta_i(t_s)$  is the *i*-th generator rotor angle at time  $t = t_s$ . The relative rotor speed, rotor acceleration and the kinetic energy of generator *i* at  $t = t_s$  are calculated by Equations (A3)–(A5) respectively and relative electromagnetic power of generator *i* at  $t = t_s$  is defined in Equation (A6), where  $P_{mi}(t_s)$  represents the mechanical power at  $t = t_s$ .

$$\tilde{\omega}_i(t_s) = \left[\tilde{\delta}_i(t_s) - \tilde{\delta}_i(t_{s-1})\right] / \Delta t \tag{A3}$$

$$\tilde{\alpha}_i(t_s) = \left[\tilde{\delta}_i(t_s) - \tilde{\delta}_i(t_{s-1}) + 2\tilde{\delta}_i(t_{s-2})\right] / (\Delta t)^2$$
(A4)

$$EK_i(t_s) = \frac{1}{2}M_i(\tilde{\omega}_i(t_s))^2 \tag{A5}$$

$$\tilde{P}_{ei}(t_s) = P_{ei}(t_s) / P_{mi}(t_s)$$
(A6)

Given what it is said above, each kind of feature can be written as a matrix as follows:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,s\times n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,s\times n} \\ \dots & \dots & \dots & \dots \\ x_{m,1} & x_{m,2} & \dots & x_{m,s\times n} \end{bmatrix}$$
(A7)

The rows of the matrix represent the number of instances, and the columns of the matrix represent the number of each kind of feature. Then normalization of the input data should be pre-processed. In

this research, maximum and minimum normalization as in Equation (A8) is used to normalize the input data.

$$x_{i,j}^{*} = \frac{x_{i,j} - \min_{i} (x_{i,j})}{\max_{i} (x_{i,j}) - \min_{i} (x_{i,j})}$$
(A8)

#### Appendix B

The following Figure A1 is a simple example showing the process of convolutional layer forward propagation.



Figure A1. A simple example showing the process of convolutional layer forward propagation.

## Appendix C

The training method of CNN is the same as that of ANN. Adjust the network connection weights and bias by mini-batch stochastic gradient descent and back propagation to minimize the loss function. In this research, in order to avoid over-fitting problems, the common cross-entropy loss [23] with L2 regularization [24] is used in classification problem shown as Equation (A9).

$$Loss_{classification} = -\frac{1}{N} \sum_{n=1}^{N} y_n log \hat{y}_n + \mu w_2^2$$

$$= -\frac{1}{N} \sum_{n=1}^{N} \left[ y_n^{(1)} log \hat{y}_n^{(1)} + y_n^{(2)} log \hat{y}_n^{(2)} \right] + \mu \sum_i |w_i^2|$$
(A9)

where *N* is the total number of training dataset instances,  $\mu$  is the regularization weight,  $w_i$  are weights of network that need to calculate the regularization loss.  $y_n = \left(y_n^{(1)}, y_n^{(2)}\right)$  is the true label vector and  $\hat{y}_n = \left(\hat{y}_n^{(1)}, \hat{y}_n^{(2)}\right)$  is the predicted label vector of the *n*-th instance.

The loss function of mean squared error (MSE) is adopted in regression problem shown as Equation (A10).

$$\text{Loss}_{\text{regression}} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$$
(A10)

where  $y_n$  and  $\hat{y}_n$  are the actual and predicted TSI of the *n*-th instance respectively.

Layer	Filter Size/Stride	Filter Number	Padding
Input Layer	-	-	-
Convolution Layer 1	$3 \times 3/1 \times 1$	32	Same
Max pooling Layer 1	$2 \times 2/2 \times 2$	32	Same
Convolution Layer 2	$3 \times 3/1 \times 1$	64	Same
Max pooling Layer 2	$2 \times 2/2 \times 2$	64	Same
Fully-connected Layer 1	120	-	-
Fully-connected Layer 2	30	-	-
Output Layer	2	-	-
- ·			

Table A1. Details of each individual CNN classifier.

## Appendix D

The script of the proposed hierarchical self-adaptive post-disturbance TSA method

Algorithm A1. Hierarchical Self-adaptive Post-disturbance TSA

1: Input input features and labels	
2. Normalize 7 kinds of input features	

3: Given 7 single CNNs with different types of input features: CNN1, CNN2, ..., CNN7, whose first outputs are represented by  $\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \dots, \hat{y}_7^{(1)}$ ; T<sub>i</sub>,  $i = 1, 2, \dots, s$ , is response time

4: <b>for</b> a	i = 1:s
5:	do
6:	Train 7 individual CNNs with each kind time-series input feature
	corresponding to T <sub>1</sub> to T <sub>i</sub> , use the integrated decision-making rule for CNN based ensemble
	classification on the 7 outputs of each remained uncertain instance in testing dataset
7:	if $\min(\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \dots, \hat{y}_7^{(1)}) > \alpha$
8:	credible stable instance, send into stable regression model to predict TSI
9:	else if $\max(\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \dots, \hat{y}_7^{(1)}) < \beta$
10:	credible unstable instance, send into unstable regression model to predict TSI
11:	else it is an uncertain instance at the current decision time
12:	if the number of remained uncertain instances is zero
13:	break
14	end
15:	Output predicted stability status and stability degree TSI

The script of the training process of a single CNN

Algorithm A2. CNN Training Process

- 3: The input data is forwardly propagated through the convolutional layer, the max-pooling layer, and the
- fully connected layer to obtain an output value
- 4: Calculate the error between the output value of the network and the target value
- 5: Adjust the network connection weights and bias by mini-batch stochastic gradient descent and back propagation to minimize the loss function
- 6: The weights and bias are updated according to the obtained error. Then go to the second step

7: End training when the error is equal to or less than our expected value

<sup>1:</sup> Input normalized input features and labels

<sup>2:</sup> Initialize weights and bias of the network randomly



Figure A2. The process of off-line training and on-line application of proposed method.

## References

- Kundur, P.; Paserba, J.; Ajjarapu, V.; Andersson, G.; Bose, A.; Canizares, C.; Hatziargyriou, N.; Hill, D.; Stankovic, A.; Taylor, C.; et al. Definition and classification of power system stability. *IEEE Trans. Power Syst.* 2014, 19, 1387–1401.
- 2. Xu, Y.; Dong, Z.Y.; Meng, K.; Zhang, R.; Wong, K.P. Real-time transient stability assessment model using extreme learning machine. *IET Gener. Transm. Distrib.* **2011**, *5*, 314–322. [CrossRef]
- 3. Xu, Y.; Dong, Z.Y.; Zhao, J.H.; Zhang, P.; Wong, K.P. A reliable intelligent system for real-time dynamic security assessment of power systems. *IEEE Trans. Power Syst.* **2012**, *27*, 1253–1263. [CrossRef]
- 4. Zhu, Q.M.; Chen, J.F.; Zhu, L.; Shi, D.Y.; Bai, X.; Duan, X.Z.; Liu, Y.L. A deep end-to-end model for transient stability assessment with PMU data. *IEEE Access* **2018**, *6*, 65474–65487. [CrossRef]
- 5. Yu, J.J.Q.; Hill, D.J.; Lam, A.Y.S.; Gu, J.T.; Li, V.O.K. Intelligent time-adaptive transient stability assessment system. *IEEE Trans. Power Syst.* **2018**, *33*, 1049–1058. [CrossRef]
- 6. Zhu, Q.M.; Dang, J.; Chen, J.F.; Xu, Y.P.; Li, Y.H.; Duan, X.Z. A method for power system transient stability assessment based on deep belief network. *Proc. CSEE* **2018**, *38*, 735–743.
- Zhang, R.Y.; Wu, J.Y.; Shao, M.Y.; Li, B.Q.; Lu, Y.Z. Transient stability prediction of power systems based on deep belief networks. In Proceedings of the IEEE Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–6.
- 8. Zadkhast, S.; Jatskevich, J.; Vaahedi, E. A multi-decomposition approach for accelerated time-domain simulation of transient stability problems. *IEEE Trans. Power Syst.* **2015**, *30*, 2301–2311. [CrossRef]
- 9. Bhui, P.; Senroy, N. Real-time prediction and control of transient stability using transient energy function. *IEEE Trans. Power Syst.* **2017**, *32*, 923–934. [CrossRef]
- Chiodo, E.; Lauria, D. Transient stability evaluation of multi-machine power systems: A probabilistic approach based upon the extended equal area criterion. *IET Gener. Transm. Distrib.* 1994, 141, 545–553. [CrossRef]

- Zhou, Y.Z.; Guo, Q.L.; Sun, H.B.; Yu, Z.H.; Wu, J.Y.; Hao, L.L. A novel data-driven approach for transient stability prediction of power systems considering the operational variability. *Int. J. Electr. Power Energy Syst.* 2019, 107, 379–394. [CrossRef]
- 12. Li, Y.; Yang, Z. Application of EOS-ELM with binary jaya-based feature selection to real-time transient stability assessment using PMU data. *IEEE Access* **2017**, *5*, 23092–23101. [CrossRef]
- You, D.H.; Wang, K.; Ye, L.; Wu, J.C.; Huang, R.Y. Transient stability assessment of power system using support vector machine with generator combinatorial trajectories inputs. *Int. J. Electr. Power Energy Syst.* 2013, 44, 318–325. [CrossRef]
- 14. Zhang, R.; Xu, Y.; Dong, Z.Y.; Wong, K.P. Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system. *IET Gener. Transm. Distrib.* **2015**, *9*, 296–305. [CrossRef]
- 15. Ree, J.D.L.; Centeno, V.; Thorp, J.S.; Phadke, A.G. Synchronized phasor measurement application in power system. *IEEE Smart Grid.* **2010**, *1*, 21–27.
- 16. Kamwa, I.; Grondin, R.; Loud, L. Time-varying contingency screening for dynamic security assessment using intelligent systems techniques. *IEEE Trans. Power Syst.* **2001**, *16*, 526–536. [CrossRef]
- 17. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; Volume 3361, pp. 1995–2009.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 2017, 60, 84–90. [CrossRef]
- 19. Gupta, A.; Gurrala, G.; Sastry, P.S. An online power system stability monitoring system using convolutional neural networks. *IEEE Trans. Power Syst.* **2019**, *34*, 864–872. [CrossRef]
- 20. Zhou, Y.Z.; Wu, J.Y.; Yu, Z.H.; Ji, L.Y.; Hao, L.L. A hierarchical method for transient stability prediction of power systems using the confidence of a SVM-based ensemble classifier. *Energies* **2016**, *9*, 778. [CrossRef]
- 21. Zhang, Y.C.; Xu, Y.; Dong, Z.Y.; Zhang, R. A hierarchical self-adaptive data-analytics method for real-time power system short-term voltage stability assessment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 74–84. [CrossRef]
- Zhou, Y.Z.; Zhao, W.L.; Guo, Q.L.; Sun, H.B.; Hao, L.L. Transient stability assessment of power systems using cost-sensitive deep learning approach. In Proceedings of the IEEE Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–6.
- 23. Bishop, C. *Pattern Recognition and Machine Learning (Information Science and Statistics Series);* Springer: Berlin, Germany, 2006; Available online: https://books.google.it/books?id=kTNoQgAACAAJ (accessed on 14 August 2019).
- 24. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
- Zhou, Y.Z.; Wu, J.Y.; Hao, L.L.; Ji, L.Y.; Yu, Z.H. Transient stability prediction of power systems using post-disturbance rotor angle trajectory cluster features. *Electr. Power Compon. Syst.* 2016, 44, 1879–1891. [CrossRef]
- 26. Ji, L.Y.; Wu, J.Y.; Zhou, Y.Z.; Hao, L.L. Using trajectory clusters to define the most relevant features for transient stability prediction based on machine learning method. *Energies* **2016**, *9*, 898. [CrossRef]
- 27. Zhou, Y.Z.; Sun, H.B.; Guo, Q.L.; Xu, B.; Wu, J.Y.; Hao, L.L. Data driven method for transient stability prediction of power systems considering incomplete measurements. In Proceedings of the IEEE Energy Internet and Energy System Integration (EI2), Beijing, China, 27–28 November 2017; pp. 1–6.
- Hansen, L.K.; Salamon, P. Neural network ensemble. *IEEE Trans. Patter Anal. Mach. Intell.* 1990, 12, 993–1001. [CrossRef]
- 29. Breiman, L. Random foresrts. Mach. Learn. 2001, 45, 5–32. [CrossRef]
- Boopathi, V.; Subramaniyam, S.; Malik, A.; Lee, G.; Manavalan, B.; Yang, D.C. mACPpred: A support vector machine-based meta-predictor for identification of anticancer peptides. *Int. J. Mol. Sci.* 2019, 20, 1964. [CrossRef] [PubMed]
- Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. Meta-4mCpred: A sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol. Ther. Nucleic Acids* 2019, 16, 733–744. [CrossRef] [PubMed]
- Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. mAHTPred: A sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics* 2018, in press. [CrossRef] [PubMed]

- Manavalan, B.; Basith, S.; Shin, T.H.; Wei, L.; Lee, G. AtbPpred: A Robust Sequence-Based Prediction of Anti-Tubercular Peptides Using Extremely Randomized Trees. *Comput. Struct. Biotechnol. J.* 2019, 17, 972–981. [CrossRef] [PubMed]
- 34. Kamwa, I.; Samantary, S.R.; Joos, G. Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices. *IEEE Trans. Smart Grid.* **2010**, *1*, 144–157. [CrossRef]
- 35. Tian, F.; Zhou, X.X.; Shi, D.Y.; Chen, Y.; Huang, Y.H.; Yu, Z.H. Power system transient assessment based on comprehensive convolutional neural network model and steady-state features. *Proc. CSEE* 2019. accepted.
- Zhou, Y. Transient Stability Analysis and Preventive Control of Power Systems Based on Data Mining Technique; Beijing Jiaotong University: Beijing, China, 2017; pp. 135–137.
- Zheng, Z.Y.; Liang, B.W. *TensorFlow Practical Application Google Deep Learning Framework*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2018; pp. 85–87.
- 38. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
- 39. He, M.; Vittal, V.; Zhang, J.S. Online dynamic security assessment with missing PMU measurements: A data mining approach. *IEEE Trans. Power Syst.* 2013, *28*, 1969–1977. [CrossRef]
- 40. Zhang, Y.C.; Xu, Y.; Dong, Z.Y. Robust ensemble data analytic for incomplete PMU measurements-based power system stability assessment. *IEEE Trans. Power Syst.* **2018**, *33*, 1124–1126. [CrossRef]
- 41. Guo, T.Y.; Milanovic, J.V. The effect of quality and availability of measurement signals on accuracy of on-line prediction of transient stability using decision tree method. In Proceedings of the Innovative Smart Grid Technologies Europe IEEE, Lyngby, Denmark, 6–9 October 2013.
- Li, Q.Q.; Xu, Y.; Ren, C.; Zhao, J. A hybrid data-driven method for online power system dynamic security assessment with incomplete PMU measurements. In Proceedings of the IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019.
- 43. Ren, C.; Xu, Y. A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data. *IEEE Trans. Power Syst.* **2019**. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).