Transfer learning based dynamic security assessment

Hoi Andy Lam1 | Zhao Yang Dong2

1 School of Computer Science, The University of Sydney, Sydney, Australia
2 School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia

Correspondence
Hoi Andy Lam, School of Computer Science, The University of Sydney, Building J12, 1 Cleveland Street, Darlington, NSW 2008, Australia.
Email: andylamhoi@gmail.com

Abstract
With the increasing deployment of wide-area monitoring systems (WAMS) and phasor measurement units (PMUs), along with artificial neural network (ANN) and high-performance distributed computation technique for smart grid and smart metering environment, online dynamic security assessment (DSA) plays a key role for early unstable event detection on power system security. It is especially important at a post-fault operation that the timing by DSA to detect an unstable event is critical to emergency remedial control action. However, excessive update training is one of the constraints for ANN to be effectively performed at pre-fault and post-fault operations on online DSA. This paper describes how transfer learning is successfully employed to shorten the training time for online DSA. It also helps to improve the validation accuracy if the training dataset of scratch ANN model is insufficient. Besides, a new approach of using the densely connected convolutional network with kernel principal component analysis (KPCA) is proposed to eliminate the traditional step of dimensionality reduction.

1 INTRODUCTION

In recent years, major power blackouts have occurred in various countries. Together with the increasing integration of renewable energy in the power supply system and the active response of the demand side, these uncertainties bring an unprecedented level of complexity to power system stability. Traditional DSA technology has not been able to provide an effective and timely system security assessment to meet this challenge.

The conventional DSA analysis methods are based on time-domain simulation (T-D) techniques and energy functions. However, it is not efficient to meet the high-speed online assessment if a large-sized model parameter and contingency set is encountered. The corresponding remedial control actions will no longer be reliable to satisfy the dynamic system condition.

With the wide application of technologies such as WAMS and PMU in power system, fast online DSA has been adopted as an effective means of detecting the unstable status of power system stability by continuous inputting the features data of operating parameters of the power system in real-time [1–3].

The DSA is applicable to the assessment of the power system security status for both pre-fault and post-fault stages. It monitors if the system will continue to be stable with respect to an anticipated but not yet occurred fault conditions through evaluating the pre-fault steady-state of the power system features, and also predict the system security status under post-fault ongoing disturbance. As such, the appropriate remedial control action is activated to prevent and avoid the power system operating at insecure regions when it encounters an unexpected fault condition or an ongoing disturbance.

In a smart grid, the data-driven method based on artificial intelligence (AI) architecture and data mining technology can effectively achieve rapid evaluation for online DSA [4]. It is a mapping relationship to correlate the input of power system features and the corresponding secure/insecure operating regions of the power system. This method can rapidly assess potential faults as a secure category or an uninsured category.

In [5, 6], decision tree (DT) and random forest (RF) were used for online DSA. In [7], support vector machine (SVM) was developed for rotor angle stability analysis. In [8, 9], artificial neural network (ANN) was proposed to perform DSA. In [10], SVM and different neural network (NN) including probabilistic, multilayer perceptron (MLP), recurrent with a long short term memory (RNN-LSTM), extreme learning machine (ELM) and stacked auto-encoders (SAE) and convolutional neural network (CNN) were tested in power systems with very high accuracy. The author could not conclude which method was more suitable for transient stability assessment (TSA). In [11], a binary classification CNN model was generated for transient stability prediction. And a multi-class classification CNN model was...
built for unstable generator identification. The author demonstrated that the proposed CNN models provided more accurate results than DT, MLP, RF and k-nearest neighbor (KNN) for transient stability prediction and unstable generator identification. In [12], transfer learning technique was employed to minimise the marginal and conditional distribution differences between the trained data and unknown data in power system for utilising one trained model to assess unlearned faults.

The AI is capable of predicting the security status of the power system in real-time after being trained by a dynamic security database of the power system. However, if the training data set is too small, it results in a poor approximation. An over-constrained model will underfit the small training dataset. If the training data set is too large, it will slow down the rapid update of the online AI model to identify the potential risk of the power outage. The timing of an insecure event identification is crucial to the effectiveness of emergency remedial control action. The earlier the unstable status of the power system is detected, the greater the likelihood that the power system will return to a stable condition. Transfer learning technology can tackle this challenge to improve the validation accuracy by finetuning technique if the training dataset is too small and rapidly train the AI model to shorten its online update time by feature extraction technique if the training dataset is too large.

Using ImageNet pre-trained CNN features, impressive results were obtained on several image classification datasets, as well as object detection, action recognition, human pose estimation, image segmentation, optical flow, image captioning, and others. In [13], the author demonstrated that training of finetuning on ImageNet classes that were not present in PASCAL-DET was sufficient to learn features that were also good for PASCAL classes. This concept is employed in this paper, that is, the pretrained non-electrical features from ImageNet on an AI model are transferred to improve the performance on the same model which is based on the training dataset obtained from the simulation of the power system. As such, the feature data obtained from the simulation of stability studies after being selected by KPCA or Relieff is converted into the two-dimensional (2D) image input to train the scratch AI model. It is in the form of a two-dimensional graph whose two coordinates are feature samples and normalised weight properties of the feature sample to represent the distribution of data in a histogram. This simpler approach has the dimension lower than the feature dataset selected by KPCA and Relieff.

In [14], the author reported that DenseNet outperformed the VGG19, ResNet50 and Xception on a biomedical image binary classification in an application of transfer learning method with a small dataset. In this paper, a binary classification DenseNet model is used to demonstrate the effectiveness of transfer learning approach even though the DenseNet can be used for multiclass classification. As DenseNet has a powerful capacity for feature analysis on different image patterns, this advantage is applied to analyse the 2D histograms of simulated OPs as different image inputs.

The pre-fault input data used in the work is converted to 2D histogram image data that is different but related to the dataset on ImageNet. The inductive transfer learning is applicable to this scenario. Then, the performance of the transfer learning approach with ImageNet is compared to scratch DenseNet trained by the operation points (OPs) obtained from the Western System Coordinating Council (WECC) 3-machine, 9 bus system [15–17]. Other smaller datasets from three scenarios on New England 39 bus system [18] are tested for further validation.

In addition, the transient stability status of a power system following a fault can be predicted early based on the measured post-fault values of the feature data obtained from the simulation of stability studies [7]. The feature data collected immediately after clearing a fault is used as inputs to an SVM classifier, predicting the transient stability status. As these feature data are used as training data for SVM, the proposed approach from this paper can convert the training data to be 2D data. Then, transfer learning techniques can be applied.

The main contributions of this paper are as follows. To the best of our knowledge, similar applications have not been reported in the field of DSA.

1. The non-electrical feature dataset on ImageNet is used to improve the performance for an AI model training on the power system.
2. The DenseNet with KPCA is proposed to eliminate the traditional step of dimensionality reduction.
3. The feature extraction technique of transfer learning is applied to shorten the training completion time on DenseNet if the training dataset is too large.
4. The finetuning technique of transfer learning is applied to improve the validation accuracy on DenseNet if the training dataset is too small.
5. The 2D histogram is used to represent the feature data of power system for the training of AI model.

2 | COMMON LEARNING ALGORITHMS

In deep learning, CNN as one of the major computational applications for image recognition and image classification is widely used in different fields [19–23]. As CNNs draw representational power from extremely deep or wide architectures, it emerges the new problems of gradient vanishing and model degradation. The recent network frameworks of ResNet [24], Highway Networks [25] and Fractal Nets [26] vary in the network topology to tackle these drawbacks; they all contain the same core ideas that they create short paths from early layers to later layers.

DenseNet [27] has deviated from the above concepts that the network performance is improved through feature reuse and bypass settings of information flow. It connects each layer to every other layer in a feed-forward fashion that some high-level features may be produced in the deeper final layers of the network. This approach makes it fewer parameters than ResNet for easier training and alleviates gradient vanishing and model degradation.

Assume a photo of $X_0$ to be input on a traditional convolutional feed-forward network which comprises $L$ layers, $t$ indexes the layer for implementing a non-linear transformation $H_t(\cdot)$. 


\( H(\star) \) can be a composite function of operations such as BN, ReLU, Pooling or Conv. The feature output at the \( t \)th layer is denoted as \( X_t \).

\[
\mathcal{X}_t = H(\{X_0, X_1, ..., X_{t-1}\}) \quad (1)
\]

where \( \llbracket \) refers to the concatenation of the feature-maps produced from layer 0 to layer \( t - 1 \). The input of the \( t \)th layer is not only related to the output of the \( t - 1 \)th layer but also accessible to the output of all preceding layers. The feature size must be the same in the same Denseblock for performing concatenation of feature maps on the different layers. Transition layers used for indirect information flow are set between different Denseblocks to achieve down-sampling. A single classifier on top of the network supervises all layers through transition layers. It makes the model structure and gradient calculation simpler than deep-supervised nets (DSN) which have classifiers attached to every hidden layer.

If the feature maps are viewed as the global state of the network, the training goal of each layer is to determine the updated value that needs to be added to the collective knowledge of its Denseblock. Assuming that the output of each non-linear transformation \( H \) is \( \kappa \) feature maps, the \( t \)th layer has \( k_0 + (t - 1) \times \kappa \) input feature maps. The growth rate \( \kappa \) also determines how much information each layer updates to the global state. DenseNet can have very narrow layers that a relatively small growth rate is sufficient to achieve state-of-art performance. This characteristic makes it significantly different from the other neural network.

The DenseNet-B uses a \( 1 \times 1 \) Conv (Bottleneck) before each \( 3 \times 3 \) Conv to reduce the computational burden that the improved non-linear transformation of input feature maps becomes BN-ReLU-Conv(1 \( \times \) 1)-BN-ReLU-Conv(3 \( \times \) 3). If a Denseblock contains \( m \) feature maps, the transition layer generates \( \beta m \) output feature maps. It is called DenseNet-C if the compression factor \( \beta < 1 \). If the Bottleneck is also used, it is called DenseNet-BC. DenseNet-BC is the most parameter efficient variant of DenseNet. It requires only about one-third of the number of parameters of ResNet with the same model accuracy. Comparing ResNet with 1001 layer over 10 M parameter and DenseNet-BC with the only 0.8 M parameter for 100 layers in training, they converge on the same training epoch.

### 3.1 A TRANSFER LEARNING BASED TRANSIENT STABILITY ASSESSMENT FRAMEWORK

Transfer learning is not a specific new concept for deep learning. Instead, it is entirely different from the traditional approach of machine learning to train a model. Users can re-use knowledge (i.e. features, weights, etc.) from another pretrained model for a related task in a new model. Reference [28] gave the definitions of “Transfer Learning” as follows.

Given a source domain \( D_S \) and learning task \( T_S \), a target domain \( D_T \) and learning task \( T_T \), transfer learning aims to improve the learning of the target predictive function \( f_T(\cdot) \) in \( D_T \) using the knowledge in \( D_S \) and \( T_S \), where \( D_S \neq D_T \) or \( T_S \neq T_T \). If the target task \( T_T \) is different from the source task \( T_S \), it is the definition of inductive transfer learning no matter when the source and target domains are the same or not.

A domain \( D \) consists of two components: a feature space \( \mathcal{X} \) and a marginal probability distribution \( P(X) \), where \( X = \{x_1, ..., x_n\} \in \mathcal{X}, D = \{\mathcal{X}, P(X)\} \).

A task consists of two components: a label space \( \mathcal{Y} \) and an objective predictive function \( f(\cdot) \) (denoted by \( T = \{\mathcal{Y}, f(\cdot)\} \)), which is not observed but can be learned from the training data of pairs \((x_i, y_i)\), where \( x_i \in \mathcal{X} \) and \( y_i \in \mathcal{Y} \). The function \( f(\cdot) \) can be used to predict the corresponding label \( f(x) \) of a new instance \( x \). From a probabilistic viewpoint, \( f(\cdot) \) can be written as \( P(y|x) \), and \( T = \{\mathcal{Y}, P(Y|X)\} \).

When the learning tasks \( T_S \) and \( T_T \) are different, then either (1) the label spaces between the domains are different, i.e. \( \mathcal{Y}_S \neq \mathcal{Y}_T \), or (2) the conditional probability distributions between the domains are different, i.e. \( P(Y_S|X_S) \neq P(Y_T|X_T) \), where \( Y_{S,i} \in \mathcal{Y}_S \) and \( Y_{T,i} \in \mathcal{Y}_T \).

When the domains are different. Then either (1) the feature spaces between the domains are different, i.e. \( \mathcal{X}_S \neq \mathcal{X}_T \) or (2) the feature spaces between the domains are the same, but the marginal probability distributions between domain data are different, i.e. \( P(X_S) \neq P(X_T) \), where \( X_{S,i} \in \mathcal{X}_S \) and \( X_{T,i} \in \mathcal{X}_T \).

When the target and source domains are the same, i.e. \( D_S = D_T \), and their learning tasks are the same, i.e. \( T_S = T_T \), it becomes a traditional machine learning problem.

The inductive transfer learning can be considered as multi-task learning if there are ample labelled data available from the source domain for target labelled data to induce an objective predictive model in the target domain. It helps to achieve high performance in the target task by transferring knowledge from the source task.

The feature-representation approach to the inductive transfer learning problem aims at finding “good” feature representations to minimise domain classification error. If a lot of labelled data in the source domain are available, supervised learning methods can be used to construct a feature representation. In [29], a two-stage approach was presented to domain adaptation. A generalisable feature representation with appropriate weights across different domains is selected to train a general classifier. Those features specifically useful for the target domain is picked up by employing semi-supervised learning.

The feature-projection mapping approach is used to map the data of each domain from the high-dimensional feature space to the low-dimensional latent feature space. After projection mapping, the marginal distributions of data between the source domain and target domain are close to each other in the low-dimensional latent feature space. In this way, the tagged source domain sample data can be used to train the classifier and predict the target test data. The training model of a supervised learning algorithm can be used in this case. In [30], a new dimension reduction method was proposed to solve the dimensionality reduction by minimising the Maximum Mean Discrepancy of the source domain data and the target domain data in the latent space. In [31], a structural corresponding learning (SCL) algorithm was proposed to learn a projection mapping of pivot.
features from the feature spaces of the source domain and target domain to a shared, low-dimensional real-valued feature space through defining a set of pivot features on the unlabelled data from both domains.

The idea of supervised feature construction methods for the inductive transfer learning setting is similar to multi-task learning. A low-dimensional representation that is shared across related tasks is learned. The common features can be learned by solving an optimisation problem as follows.

$$\arg \min_{\mathbf{A}, \mathbf{U}} \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} L \left( y_{ti}, \langle a_t, \mathbf{U}^T \mathbf{x}_{ti} \rangle \right) + \gamma \| \mathbf{A} \|_{2,1}^2 \quad \text{s.t.} \quad \mathbf{U} \in \mathcal{O}^d$$ (2)

In this equation, $S$ and $T$ denote the tasks in the source domain and target domain, respectively. $\mathbf{A} = [a_S, a_T] \in \mathbb{R}^{d \times 2}$ is a matrix of parameters. $\mathbf{U}$ is a $d \times d$ orthogonal matrix for mapping the original high-dimensional data to low-dimensional representations.

The learned new representation can reduce the classification error of each task. The $(r, p)$-norm of $\mathbf{A}$ is defined as

$$\| \mathbf{A} \|_{r,p} := \left( \sum_{i=1}^{d} \| a_i \|_r^p \right)^{1/p}$$ (3)

The above optimisation problem estimates the low-dimensional representations $\mathbf{U}^T \mathbf{X}_T$, $\mathbf{U}^T \mathbf{X}_S$ and the parameters, $\mathbf{A}$, of the model at the same time. It can be further transformed into an equivalent convex optimisation formulation and be solved efficiently.

### 4 | A SCRATCH DENSENET MODEL FOR TRANSIENT STABILITY ASSESSMENT

The later the training on the AI model is completed, the less the likelihood that the risk of blackouts can be detected earlier. In practice, it is common to pretrain a DenseNet on a very large dataset and then use it either as an initialisation or a fixed feature extractor. It helps to tackle the challenge if a scratch deep neural network of stability prediction takes too long to complete the training from a large feature dataset of the power system. Also, if the training data is not sufficient, e.g. the power system is under the situation of topology changes, it results in poor performance on validation accuracy.

In this paper, the ImageNet is used to pretrain the proposed DenseNet. Then, the techniques of transfer learning are applied to improve the performance that it outperforms the scratch DenseNet model to be trained by the dataset obtained from the simulation of stability studies. The training dataset is described in Section 5. The two major transfer learning scenarios are given as follows:

a. DenseNet as a fixed feature extractor: The weights for all the DenseNet is frozen except that of the final fully

layer. This last fully connected layer is replaced with a new one with random weights, and only this layer is trained. The feature codes for all images are extracted to train this layer as a classifier. The rest of the DenseNet is treated as a fixed feature extractor for the new dataset.

b. Finetuning the DenseNet: Instead of random initialisation, it finetunes the weights of the pretrained DenseNet using data from ImageNet by continuing the backpropagation. As the earlier layers of a DenseNet contain a more generic feature; it is an option on finetuning some higher-level portion of the network rather than all the layers. Rest of the training is running as usual.

As described in Section 3, if the feature dataset of the power system is used to pretrain the DenseNet before feature extraction of transfer learning is performed, it becomes a traditional machine learning approach. If the DenseNet has already been trained by a feature dataset, the proposed transfer learning approach provides flexibility to re-use this DenseNet model as a pretrained deep neural network. Then, the weights of the DenseNet are updated from its last checkpoint through fine-tuning or feature extraction techniques to save for later use.

Since the target dataset is similar in context to the original dataset of the pretrained DenseNet at the last checkpoint, the feature extraction technique can rapidly train the DenseNet with validation accuracy close to the pretrained model. If the new dataset is very small, training a DenseNet on a very small dataset from scratch may affect its ability to generalise, often cause overfitting concerns. The finetuning technique can tackle this problem to get a better result.

The flow chart of transfer learning-based framework is shown in Figure 1.
5 | DATABASE GENERATION

In [32], five indices were reviewed for the assessment of power system transient stability. They are the maximum rotor angle deviation, maximum speed deviation, maximum acceleration, transient rotor angle severity index (TRASI) and transient stability index (TSI). The indices of maximum rotor angle deviation, maximum speed deviation and maximum acceleration are specific to the individual generator that the running characteristics of each generator are feedbacked. But they do not provide the specific instability value of the entire system. The TRASI and TSI are the indices to measure the whole system stability status. TRASI measures the maximum rotor angle difference before and after a system fault to determine the severity. But the research for this subject has commonly used the angle-based stability index of TSI as described in Equation (4) [32]. Also, its concept is not complicated to be implemented by Power System Toolbox (PST) software. Therefore, it is used as the transient stability index in this paper. According to the criterion of Equation (5), secure and insecure labels are used to classify transient stability status.

The popular WECC 3-machine, 9 bus system and New England 39 bus system are considered in this paper. The simple 9 bus system well demonstrates how the proposed DenseNet and transfer learning techniques are tested for transient stability classification. The generator at bus #2 is assigned as a wind farm, and the power system network and parameters are obtained from [33]. Then, the effectiveness of the proposed approach is further validated by New England 39 bus system.

Power angle-based stability index is defined as follows:

\[ \eta = \frac{360 - \delta_{\text{max}}}{360 + \delta_{\text{max}}} \times 100 \quad 100 < \eta < 100 \quad (4) \]

\[ f\left\{ \begin{array}{ll}
\eta > 0 & \rightarrow \text{stable (secure label)} \\
\eta < 0 & \rightarrow \text{unstable (insecure label)}
\end{array} \right. \quad (5) \]

where \( \delta_{\text{max}} \) is the maximum angle separation of any two generators at the same time in the post-fault response. The \( \eta > 0 \) and \( \eta < 0 \) correspond to stable and unstable conditions [34].

Under the above transient stability criterion, 4,900 OPs are sampled evenly distributed across the variation of the generator output power at bus #2 and load demand that 3,500 pre-fault secure OPs and 1,400 pre-fault insecure OPs of the 9 bus test system are produced, respectively. It covers various load patterns ranging from 60% to 180% and the electricity output capacity of the wind farm ranging from 0% to 100% of its base OP. A three-phase fault is simulated at bus #4 by PST software, and the fault is cleared after 0.05 s.

Other three smaller datasets of 500 OPs are generated from New England 39 bus system by DigSilent PowerFactory software as shown in Table 1. Monte Carlo technique is applied to generate the random load ratio within the pre-determined range between 0.52 to 1.2.

Table 1 records the protection measures on three scenarios that the three phases fault occurs on buses at 0.5 s to obtain different stability dataset. In the scenario of bus #3, the fault is clear at 0.75 s. It generates 207 pre-fault secure OPs and 293 pre-fault insecure OPs. No transmission line is tripped. On scenarios of bus #19 and #22, the faults are clear at 0.7 and 0.6 s. The transmission lines between bus #19 - #33 and #21 - #22 are tripped respectively on each scenario. The scenario of bus #19 generates 280 pre-fault secure OPs and 220 pre-fault insecure OPs. The scenario of bus #22 generates 217 pre-fault secure OPs and 283 pre-fault insecure OPs. The validation data is selected from 25% of the above datasets.

5.1 | Dimensionality reduction

The advantage of using the feature selection technique has recently attracted much attention within the deep learning studies that it helps to reduce training time, overfitting and complexity of the feature datasets, and also improve data understandability and validation accuracy. The cost of data acquisition may be significantly reduced if the right subset is chosen.

The power system has a high dimension of features in nature. Only those features which can characterise the system stability status are served as useful inputs. ReliefF algorithm [35, 36], which is well known applied to genetic analyses where epistasis is common is used in this paper to identify feature interactions without having to check every pairwise interaction exhaustively. It significantly reduces the initial dimensionality of the feature datasets with less calculation time than exhaustive pairwise search. The features are assigned with different weights according to the relevance of each feature and category. Also, KPCA for dimensionality reduction in non-linear space is used to compare the effectiveness of the ReliefF algorithm through the validation accuracy of the DenseNet model for transient stability prediction.

Dimensionality reduction has been studied widely for image processing in the deep learning community. Traditional dimensionality reduction methods try to project the original data to a 2D coordinate while preserving the critical properties of the original data for a CNN model. In [37], the maximum variance unfolding (MVU) technique was proposed to maximise the variance of the embedding while preserving the local distances between neighbouring observations for extracting a low-dimensional representation of the data. The MVU applies principal component analysis (PCA) [38, 39] to kernel matrix \( K \) to

| Fault location | bus #3 | bus #19 | bus #22 |
|---------------|-------|--------|--------|
| Fault duration (s) | 0.25  | 0.2    | 0.1    |
| Tripped line    | N/A   | 19-33  | 21-22  |
| Secure OPs     | 207   | 280    | 217    |
| Insecure OPs   | 293   | 220    | 283    |


choose the base eigenvectors for mapping data on KPCA. The concepts of PCA and KPCA are summarised as follows.

5.2 | PCA

It is a transformation to diagonalise the covariance matrix of the feature data \( x_j \), \( j = 1, \ldots, m \) for \( x_j \in \mathbb{R}^d \) that the data is centred \( \sum_{j=1}^m x_j = 0 \) to be defined as:

\[
C = \frac{1}{m} \sum_{j=1}^m x_j x_j^T
\]

(6)

The above problem can be solved by the following eigenvalue equation [40]:

\[
\lambda \nu = C \nu
\]

(7)

where \( \|\nu\| = 1, \lambda \geq 0 \) are eigenvalues and \( \nu \in \mathbb{R}^d \) are eigenvectors.

The orthogonal projection on the eigenvector \( \nu^k \) of principle component for a test point \( x \) can be found as:

\[
x^k = \nu^k \cdot x
\]

(8)

\[
x_i' = \sum_{j=1}^{d'} a_{k_i} \nu^k
\]

(9)

where \( d' \leq m \) and \( x_i' \) are vectors after PCA transforms in \( d' \)-lower-dimensional space.

Refer to [41, 42]. The scatter matrix \( S \) is defined as a statistical expectation operator:

\[
S = E \left( x_j, x_j^T \right)
\]

(10)

where \( j = 1, \ldots, m \) for \( x_j \in \mathbb{R}^d \), combine Equation (9) and Equation (10) that \( a_{k_i} \) are the projections of the \( x_i \) on \( \nu^k \).

\[
\sigma^2 (\nu^k) = E [a_{k_i}^2] = (\nu^k)^T S \nu^k
\]

(11)

Given by [42], the variance \( \sigma^2 \) is a function of eigenvector \( \nu^k \). Equation (11) can be represented as an eigenvalue problem with nontrivial solutions at local maximum or minimum as shown in Equation (12)

\[
S \nu^k = \lambda \nu^k
\]

(12)

\[
E_{d'} = \frac{1}{2} \sum_{j=d'+1}^{m} \lambda_k
\]

(13)

where \( \lambda_1 > \lambda_2 > \ldots > \lambda_k > \ldots > \lambda_m \), \( E_{d'} \) is the error in the representation of original data \( x_j \) compared to data \( x_j' \) after dimensionality reduction by PCA, and \( \lambda_k \) is the eigenvalue corresponding to \( \nu^k \). As the eigenvectors \( \nu_1, \nu_2, \ldots, \nu_m \) are corresponding to \( \lambda_1 > \lambda_2 > \ldots > \lambda_k > \ldots > \lambda_m \), the lowest order component of eigenvectors with the largest eigenvalues gives the smallest error in representation. It implies the variance to be maximum in the direction of the eigenvectors.

5.3 | KPCA

The dynamic security problem is in the form of a large number of non-linear differential-algebraic equations. PCA is applicable for linear correlation among the variables that its performance is degraded in a non-linear problem. Kernel PCA is more suitable to transform data which are confined to low dimensional non-linear subspace in power system [43, 44]. It is a non-linear extension of PCA [45] to be described as follows.

First step: The \( x_i \in \mathbb{R}^d \) which is non-linearly separable is mapped to feature space \( F \), dimensionally higher (> \( d' \)), \( \Phi : \mathbb{R}^d \rightarrow \Phi \), so that the new non-linear subspace is linearly separable to perform PCA in \( F \) for a specific choice of \( \Phi \) [46]. As such, \( x_i \in \mathbb{R}^d \) are mapped through \( \Phi (x_i) \in F \). The Mapping \( \Phi : \mathbb{R}^d \rightarrow F \) is defined as dot product \( K(x_i, x_j) = \Phi (x_i) \cdot \Phi (x_j) \), which is similar to Equation (8). The dot product of Gaussian kernel is:

\[
K_{ij} = K (x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{2\gamma^2} \right) \gamma > 0, \gamma : \text{bandwidth}.
\]

(14)

Second step: Suppose the feature space \( F \) which is defined by the vectors \( \{ \Phi (x_1), \Phi (x_2), \ldots, \Phi (x_m) \} \) to be centralised, i.e. \( \sum_{j=1}^m \Phi (x_j) = 0 \), covariance for these vectors is shown as:

\[
\hat{\Sigma} = \frac{1}{m} \sum_{j=1}^m \Phi (x_j) \Phi (x_j)^T
\]

(15)

The eigenvalues \( \lambda \geq 0 \) and nonzero eigenvectors \( \nu \in F \) that satisfy \( \lambda \nu = \hat{\Sigma} \nu \) is required to do PCA for Equation (15).

\[
\lambda (\Phi (x_k) \cdot \nu) = (\Phi (x_k) \cdot \hat{\nu}), \forall k = 1, 2, \ldots, m
\]

(16)

where all eigenvectors \( \nu \) in the feature space \( F \) can be expressed as a linear combination of \( \Phi (x_k) \) in Equation (16). There exist coefficients \( \alpha_i \) where \( i = 1, \ldots, m \), in such, \( \nu = \sum_{i=1}^m \alpha_i \Phi (x_i) \). From Equation (16):

\[
\lambda \sum_{i=1}^m \alpha_i \Phi (x_k) \cdot \Phi (x_i) = \frac{1}{m} \sum_{i=1}^m \alpha_i \left\{ \sum_{j=1}^m \left( \Phi (x_k) \cdot \Phi (x_j) \right) \Phi (x_j) \right\}
\]

(17)

\( \forall k = 1, 2, \ldots, m \)

The Equation (17) is simplified by defining the \( m \times m \) Gram matrix \( K \) for \( K_{ij} = \Phi (x_i) \cdot \Phi (x_j) \). It is also known as Kernel
where \( \mathbf{a} = (a_1, a_2, ..., a_m)^T \), \( \mathbf{a} \) is not considered if \( \lambda = 0 \). PCA can be performed in the feature subspace \( \mathcal{F} \) by solving the Equation (19) for eigenvectors \( \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_p \) with eigenvalues \( \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m \). If the eigenvectors solution from the range of \( p+1 \) to \( m \) are reduced, the dimensionality reduction for PCA can be operated for the eigenvectors \( \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_p \) with eigenvalues \( \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \), where \( \lambda_p \geq \lambda_m \). The \( \mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_p \) are normalised if the corresponding vectors \( \mathbf{v}_k \in \mathcal{F} \) be normalised, i.e. \( \mathbf{v}_k \cdot \mathbf{v}_k = 1 \) and \( \mathbf{v}_k = \sum_{i=1}^{m} \alpha_i^k \mathbf{a}_i \mathbf{F}(x_i) \). Apply the Gram matrix \( \mathbf{K} \)

\[
1 = \sum_{i,j=1}^{m} \alpha_i^k \alpha_j^k (\mathbf{F}(x_i) \cdot \mathbf{F}(x_j)) \tag{20}
\]

\[
= \alpha_k \cdot \mathbf{K} \alpha_k \tag{21}
\]

\[
= \lambda_k (\alpha_k \cdot \alpha_k) \text{ where } k = 1, ..., p \tag{22}
\]

The projection on the normalised eigenvector \( \mathbf{v}_k \) in feature subspace \( \mathcal{F} \) for a test point \( x \) can be found as:

\[
\mathbf{v}_k \cdot \mathbf{F}(x) = \sum_{j=1}^{m} \alpha_j^k (\mathbf{F}(x_j) \cdot \mathbf{F}(x)) \tag{23}
\]

The kernel function \( \mathbf{F}(x_k) \cdot \mathbf{F}(x_j) = \mathbf{K}(x_k, x_j) \), which acts as a dot product to satisfy the Mercer’s theorem for mapping into the feature space \( \mathcal{F} \), can be used to reduce the computational cost for non-linear information of process. Using the kernel function of the Gaussian kernel in Equation (14), the dot product in feature space can be reduced to a function in input space.

Referring to Equation (14), the performance of a Gaussian kernel is highly dependent on the choice of bandwidth. As such, the simulations on different bandwidths of the Gaussian kernel were carried out using the stability data of different operating scenarios as described in Section 5. Nine bandwidths sizes were selected as 0.05, 0.75, 1, 10, 20, 40, 60, 80 and 100.

Using ReliefF, the values of all the features from the simulation of the 9 bus system were calculated and shown in Table 2 in descending order. Their distinct values imply the ability to classify stability instance.

| TABLE 2 | Features and their ReliefF weights on 9 bus system |
|---------|--------------------------------------------------|
| \( P_{G2} \) | 0.3234 |
| \( Q_{ST} \) | 0.3233 |
| \( P_{T4} \) | 0.2576 |
| \( \theta_2 \) | 0.2432 |
| \( \theta_8 \) | 0.2327 |
| \( \theta_1 \) | 0.2312 |
| \( \theta_7 \) | 0.2306 |
| \( \theta_9 \) | 0.2106 |
| \( P_{T2} \) | 0.2068 |
| \( Q_{T2} \) | 0.1984 |
| \( P_{T1} \) | 0.1976 |
| \( Q_{T1} \) | 0.1976 |
| \( Q_{T3} \) | 0.1952 |
| \( \theta_4 \) | 0.1904 |
| \( \theta_6 \) | 0.1880 |
| \( \theta_5 \) | 0.1792 |
| \( \theta_2 \) | 0.1774 |
| \( \theta_7 \) | 0.1774 |
| \( \theta_9 \) | 0.1774 |

| TABLE 3 | Selected features |
|---------|-------------------|
| Buses | Bus voltage magnitude (\( V_i \)) |
| Transmission lines | Bus voltage angle (\( \theta_i \)) |
| Generators, loads | Real power flow (\( P_i \)) |
| | Reactive power flow (\( Q_i \)) |
| | Net real power injection (\( P_{GT} \), \( P_{LT} \)) |
| | Net reactive power injection (\( Q_{GT}, Q_{LT} \)) |

To verify the benefits of the proposed transfer learning approach, the techniques of feature extraction and finetuning were repeated with the same training and validation datasets for comparison with the scratch model. The classification results of different techniques for the above test data are shown in the Case 1 and 2.

As compared to the effectiveness of ReliefF, the data obtained by using KPCA method for non-linear transformation is used to train the scratch model of DenseNet.

Also, the selected features as listed in Table 3 are reduced to the same dimensions as ReliefF did. The respective results of validation accuracy are shown in Table 4. It indicates that the bandwidth parameter \( \gamma = 20 \) has the best validation accuracy of 0.990283. This value is higher than the best validation accuracy of 0.982843 obtained from the simulation data of ReliefF, as shown in Table 5. The bandwidth \( \gamma = 20 \) was selected to proceed with the transfer learning approach.

According to Section 4, the simulation results of the scratch DenseNet model with data from ReliefF and KPCA are compared with the transfer learning approach, as shown in Figures 2 and 3 and Tables 5–7.

The DenseNet obtained scratch train and validation results with data from ReliefF and KPCA at Epoch 14/14, as shown

| TABLE 4 | Respective best validation accuracy for the scratch model of DenseNet based on bandwidth selection of KPCA |
|---------|--------------------------------------------------|
| Bandwidth (\( \gamma \)) | 0.05 |
| Best Val Acc | 0.73509 |
| Bandwidth (\( \gamma \)) | 0.75 |
| Best Val Acc | 0.95548 |
| Bandwidth (\( \gamma \)) | 1 |
| Best Val Acc | 0.972403 |

6 | NUMERICAL SIMULATION

6.1 | WECC 3-machine 9 bus power system

The proposed scratch DenseNet model for transient stability prediction was trained using the data obtained from the dimensionality reduction by ReliefF and KPCA method as described in Section 5.
TABLE 5 Simulation results of scratch DenseNet model with data obtained from ReliefF and KPCA

| Data source       | Train Loss | Acc | Val Loss | Acc | Training complete in | Best Val Acc |
|-------------------|------------|-----|----------|-----|----------------------|--------------|
| (ReliefF)         | 0.1466     | 0.9409 | 0.0537   | 0.9796 | 633 m 24s        | 0.962843     |
| (KPCA, γ = 20)   | 0.3953     | 0.8100 | 0.1187   | 0.9822 | 802 m 54s        | 0.990283     |

FIGURE 2 Simulation for transfer learning of feature extraction and finetuning, and scratch model with the dataset from the ReliefF technique in Table 5. It is the scratch model that does not use the transfer learning technique during the model training.

As described in Section 3, the proposed transfer learning approach can improve validation accuracy or reduce training completion time compared to the simulation result of scratch DenseNet model, which does not use the transfer learning as illustrated in Figures 2 and 3. The following two case studies of transfer learning demonstrate the effectiveness on a 9 bus system if the original pretrained dataset comes from ImageNet.

Case 1: Feature extraction

Apply feature extraction technique of transfer learning to verify if training completion time is reduced. The DenseNet obtained train and validation results with data from ReliefF and KPCA at Epoch 14/14 are shown in Table 6.

FIGURE 3 Simulation for transfer learning of feature extraction and finetuning, and scratch model with the dataset from the KPCA technique

TABLE 6 Simulation results of feature extraction technique with data obtained from ReliefF and KPCA

| Data source       | Train Loss | Acc | Val Loss | Acc | Training complete in | Best Val Acc |
|-------------------|------------|-----|----------|-----|----------------------|--------------|
| (ReliefF)         | 0.3116     | 0.9634 | 0.0627   | 0.9796 | 626 m 1s       | 0.995098     |
| (KPCA, γ = 20)   | 0.6100     | 0.7108 | 0.263    | 0.8794 | 801 m 26s      | 0.995951     |

TABLE 7 Simulation results of finetuning technique with data obtained from ReliefF and KPCA

| Data source       | Train Loss | Acc | Val Loss | Acc | Training complete in | Best Val Acc |
|-------------------|------------|-----|----------|-----|----------------------|--------------|
| (ReliefF)         | 0.0994     | 0.9634 | 0.0627   | 0.9796 | 626 m 1s       | 0.995098     |
| (KPCA, γ = 20)   | 0.263      | 0.8794 | 0.0192   | 0.9943 | 801 m 26s      | 0.995951     |

So, the feature extraction technique reduces training completion time from 633 m 24s to 313 m 47s and 802 m 54s to 310 m 39s for data obtained from ReliefF and KPCA, respectively. As previously described, if the feature dataset of the power system is used to pretrain the DenseNet before feature extraction of transfer learning is performed, it becomes a traditional machine learning approach. The validation accuracy is close to the pretrained DenseNet.

Case 2: Finetuning

Apply the finetuning of transfer learning to verify if validation accuracy is improved. The DenseNet obtained train and validation results with data from ReliefF and KPCA at Epoch 14/14, as shown in Table 7.

So, the best validation accuracies are increased from 0.982843 to 0.995098 and 0.990283 to 0.995951 for data obtained from ReliefF and KPCA, respectively. Figures 2 and 3 illustrate the effects if the DenseNet model applies transfer learning techniques.

The dark blue line shows that the scratch DenseNet model, which does not use the transfer learning technique, is trained to classify the pre-fault stability status, as described in Section 5. It attains validation accuracy 0.9796, or 2.04% error for source data from ReliefF and validation accuracy 0.9822, or 1.78% error for source data from KPCA, respectively, at Epoch 14/14.

The orange dash line shows the effect if DenseNet model applies the feature extraction technique. The training completion time drops from 633 m 24s to 313 m 47s or 49.54% for source data from ReliefF and 802 m 54s to 310 m 39s, or 38.69% for source data from KPCA, respectively, at Epoch 14/14.

The orange dash line shows the effect if DenseNet model applies the feature extraction technique. The training completion time drops from 633 m 24s to 313 m 47s or 49.54% for source data from ReliefF and 802 m 54s to 310 m 39s, or 38.69% for source data from KPCA, respectively, at Epoch 14/14. Its performance is dropped below the dark blue line of the scratch DenseNet model, as shown in Figures 2 and 3. The feature transfer from ImageNet trained model to the layer of DenseNet model reflects the specificity of representation to dominate the drop in performance. However, when the target and source domains are the same, and their learning tasks are the same, it becomes a traditional machine learning problem to resolve the drop-in performance.
The red dash line shows a very positive effect if the ImageNet trained model transfers features to DenseNet model and finetune them. It generalises better performance than the scratch DenseNet model, which does not use the transfer learning technique. It has the advantage not only to enable training without overfitting on small target datasets but also boost generalisation performance even if the target dataset is small as 4900 OPs.

### 6.2 New England 39 bus power system

The proposed transfer learning approach which uses the ImageNet as pretrained dataset is further validated on New England 39 bus power system. The simulation results are shown in Tables 1, 8–10 and Figures 4–6. Tables 8–10 and Figures 4–6 illustrate the simulation results to validate the findings on Cases 1 and 2 that the feature extraction technique reduces training completion time, and finetuning technique improves validation accuracy.

In the scenario of bus #3, the feature extraction technique reduces the training completion time from 373 m 27s to 163 m 7s or 43.68% at Epoch 29/29. The best validation accuracy also increases from 0.616 to 0.696. The finetuning technique improves validation accuracy from 0.616 to 0.824, as shown in Table 8 and Figure 4.

In the scenario of bus #19, the feature extraction technique reduces the training completion time from 354 m 50s to 155 m 16s, or 43.76% at Epoch 29/29. The finetuning technique improves validation accuracy from 0.936 to 0.952, as shown in Table 9 and Figure 5.

In the scenario of bus #22, the feature extraction technique reduces the training completion time from 347 m 18s to 158 m 47s or 45.72% at Epoch 29/29. The finetuning technique improves validation accuracy from 0.848 to 0.872, as shown in Table 10 and Figure 6.

The simulation results using the dataset obtained from New England 39 bus system as shown in Tables 8–10 and Figures 4–6 have further validated Section 6.1, that the feature extraction
technique shortens the training completion time and finetuning technique improves validation accuracy for pretrained ImageNet dataset. It also shows that the finetuning technique can reach high validation accuracy compared to scratch model training or feature extraction technique after a few training epochs. The curves shown on Figures 4–6 also match the observation from Figures 2 and 3.

In the scenario of bus #3, no transmission line is tripped as protection measures. In the scenarios of bus #19 and #22, transmission lines are tripped, respectively. On a large complex power system, it is a general practice to trip the fault-related transmission lines to isolate the fault location and stabilise the power system under severe fault condition. The scenarios of bus #19 and #22 which achieve high validation accuracy are more applicable to industrial practice.

7 | CONCLUSION

The simulation results of transient stability prediction supported by feature extraction and finetuning techniques for pretrained ImageNet dataset and target dataset obtained from the WECC 9 bus system with dimensionality reduction by ReliefF and KPCA are shown in Figures 2 and 3 and Tables 5–7. The results illustrate that the feature extraction technique reduces training completion time and finetuning technique improves validation accuracy if transfer learning approach is applied, respectively. It also demonstrates that the finetuning technique maintains stronger convergence ability with higher validation accuracy than the feature extraction technique or scratch DenseNet model. The simulation results using the target dataset obtained from the New England 39 bus system with pretrained ImageNet dataset, as shown in Tables 8–10 and Figures 4–6 have further validated the above conclusion.

As the ReliefF and KPCA present the target dataset in a different feature space, the DenseNet model with finetuning technique is capable of achieving the high validation accuracy around 0.995098 to 0.995951 in Section 6.1. The feature extraction technique can reduce training completion time to 49.54% and 38.69%, respectively.

Besides, it demonstrates that KPCA is more effective than ReliefF to process features for dimensionality reduction. Also, the DenseNet successfully extracts features from the 2D histogram images of simulated OPs for further processing of transient stability assessment. The proposed approach works for pre-fault DSA, and the procedure is also applicable to post-fault DSA.

ACKNOWLEDGEMENT

This work was partially supported by ARC Research Hub for Integrated Energy Storage Solutions.

REFERENCES

1. Xu, Y., et al.: Real-time transient stability assessment model using extreme learning machine. IET Gener. Transm. Distrib. 5(3), 314–322 (2011).
2. Sun, K., et al.: An online dynamic security assessment scheme using phasor measurements and decision trees. IEEE Trans. Power Syst. 22(4), 1935–1943 (2007).
3. He, M., et al.: Online dynamic security assessment with missing PMU measurements: A data mining approach. IEEE Trans. Power Syst. 28(2), 1969–1977 (2013).
4. Zhang, P., et al.: Next-generation monitoring, analysis, and control for the future smart control center. IEEE Trans. Smart Grid 1(2), 186–192 (2010).
5. Gene, I., et al.: Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems. IEEE Trans. Power Syst. 25(3), 1611–1619 (2010).
6. Liu, C., et al.: An accurate online dynamic security assessment scheme based on random forest. Electric Power Energy Syst. MDPI 11(7), 1914 (2018).
7. Gomez, F.R., et al.: Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements. IEEE Trans. Power Syst. 26(3), 1474–1483 (2011).
8. Liu, C.W., et al.: Application of a novel fuzzy neural network to real-time transient stability swings prediction based on synchronized phasor measurements. IEEE Trans. Power Syst. 14(2), 685–692 (2002).
9. Siddiqui, S.A., et al.: Real-time monitoring of post-fault scenario for determining generator coherency and transient stability through ANN. IEEE Trans. Ind. Appl. 54(1), 685–692 (2018).
10. Baltas, G.N., et al.: Support vector machine and neural network applications in transient stability. In: Proc. Int. Conf. Renewable Energy Research and Application, Paris, France, pp. 14–17 (2018).
11. Zhou, Y., et al.: Online identification of transient stability and unstable generators based on deep learning. In: IEEE Conf. Energy Internet and Energy System Integration, pp. 1–6 (2018).
12. Ren, C., Xu, Y.: Transfer learning based power system online dynamic security assessment: Using one model to assess many unlearned faults. IEEE Trans. Power Syst. 35(1), 821–824 (2020).
13. Hub, M., et al.: What makes ImageNet good for transfer learning? In: Proc. CVPR, Las Vegas, NV, pp. 1–10 (2016).
14. Brodzicki, A., et al.: Transfer learning methods as a new approach in computer vision tasks with small datasets. FCDS 45, 179–193 (2020).
15. Sauer, PW, Pai, M.A.: Simulation of multimachine power system dynamics. In: C.T. Leondes (Ed.) Control and Dynamic System. Academic Press (1991).
16. Stott, B.: Power system dynamic response calculations. Proc. IEEE 67, 219–241 (1979).
17. Dembart, B., et al.: Power system dynamic analysis-phase I. EPRI Report, pp. EL-484 (1977).
18. Pai, M.A.: Energy Function Analysis for Power System Stability. Kluwer Academic Publishers (1989).
19. Goodfellow, I., et al.: Deep Learning. MIT Press (2016)
20. Gupta, A., et al.: An online power system stability monitoring system using convolutional neural networks. IEEE Trans. Power Syst. 34(2), 864–872 (2019)
21. Tan, B., et al.: Representational learning approach for power system transient stability assessment based on convolutional neural network. J. Eng. 2017(13), 1847–1850 (2017)
22. Hou, J., et al.: Power system transient stability assessment based on voltage phasor and convolution neural network. In: Proc. IEEE Int. Conf. Energy Internet, Beijing, China, pp. 247–251 (2018)
23. Wang, Y., et al.: Online analysis of voltage security in a microgrid using convolutional neural networks. In: Proc. Power Energy Soc. General Meeting, Chicago, IL, pp. 1–5 (2017)
24. He, K., et al.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Las Vegas, NV, pp. 1–12 (2016)
25. Srivastava, R.K., et al.: Training very deep networks. In: Proc. Int. Conf. NIPS, Montreal, Canada, pp. 2377–2385 (2015)
26. Larsson, G., et al.: Fractalnet: Ultra-deep neural networks without residuals. Proc. Int. Conf. Learning Representations, Palais des Congrès Neptune, Toulon, France, pp. 1–11 (2017)
27. Huang, G., et al.: Densely connected convolutional networks. In: Proc. IEEE Conf. CVPR, Honolulu, HI, pp. 2261–2269 (2017)
28. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22(10), 1345–1359 (2010)
29. Jiang, J., Zhai, C.X.: A two-stage approach to domain adaptation for statistical classifiers. In: Proc. ACM Conf. Information and Knowledge Management, Lisbon, Portugal, pp. 401–410 (2007)
30. Pan, S.J., et al.: Transfer learning via dimensionality reduction. In: Proc. AAAI, Chicago, IL, pp. 677–682 (2008)
31. Blitzer, J., et al.: Domain adaptation with structural correspondence learning. In: Proc. Empirical Methods in Natural Language Processing, Sydney, NSW, Australia, pp. 120–128 (2006)
32. Zha, Y.: Power system loads and power system stability. PhD thesis, Manchester University (2020)
33. Sauer, P.W., et al.: Multimachine simulation. In: Power System Dynamics and Stability with Synchrophasor Measurement and Power System Toolbox, 2nd ed., p. 143. IEEE Press (2018)
34. Pavella, M., et al.: Transient Stability of Power Systems: A Unified Approach to Assessment and Control, 1st ed. Kluwer Academic Publishers (2000)
35. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proc. Int. Workshop Mach. Learn., Aberdeen, UK, pp. 249–256 (1992)
36. Sikoa, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RRelief. Mach. Learn. 53(1), 23–69 (2003)
37. Weinberger, K.Q., et al.: Learning a kernel matrix for nonlinear dimensionality reduction. In: Proc. Int. Conf. Machine Learning, Banff, AB, Canada (2004)
38. Wahab, N.L.A., et al.: Feature selection and extraction methods for power systems transient stability assessment employing computational intelligence techniques. Neural Process. Lett. 35(1), 81–102 (2011)
39. Pearson, K.: On lines and planes of closest fit to systems of points in space. PhilosMag 2(11), 559–572 (1901)
40. Hyvarinen, A., et al.: Independent Component Analysis, 1st ed. John Wiley & Sons (2001)
41. Malli, A., Gao, R.: PCA-based feature selection scheme for machine defect classification. IEEE Trans. Instrum. Meas. 53(6), 1517–1525 (2004)
42. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd ed. Prentice-Hall (1998)
43. Kundur, P.: Power System Stability and Control. McGraw-Hill (1994)
44. Zhong, B.X.: A method of simplified modeling based on kernel function principal component analysis. In: Proc. Conf. CC, Yantai, China, pp. 1525–1528 (2011)
45. Scholkopf, B., et al.: Kernel principal component analysis. In: Proc. Int. Conf. ANN, Lausanne, Switzerland, pp. 583–588 (1997)
46. Boser, B.E., et al.: A training algorithm for optimal margin classifiers. In: Proc. Annual Workshop, COLT, Pittsburgh, PA, pp. 144–152 (1992)