# Reconstruction Residuals Based Long-term Voltage Stability Assessment Using Autoencoders

Haosen Yang, Robert C. Qiu, Fellow, IEEE, and Houjie Tong

Abstract-Real-time voltage stability assessment (VSA) has long been an extensively research topic. In recent years, rapidly mounting deep learning methods have pushed online VSA to a new height that large amounts of learning algorithms are applied for VSA from the perspective of measurement data. Deep learning methods generally require a large dataset which contains measurements in both secure and insecure states, or even unstable state. However, in practice, the data of insecure or unstable state is very rare, as the power system should be guaranteed to operate far away from voltage collapse. Under this circumstance, this paper proposes an autoencoder based method which merely needs data of secure state to evaluate voltage stability of a power system. The principle of this method is that an autoencoder purely trained by secure data is expected to only create precise reconstruction for secure data, while it fails to rebuild data of insecure states. Thus, the residual of reconstruction is effective in indicating VSA. Besides, to develop a more accurate and robust algorithm, long short-term memory (LSTM) networks combined with fully-connected (FC) layers are used to build the autoencoder, and a moving strategy is introduced to bias the features of testing data toward the secure feature domain. Numerous experiments and comparison with traditional machine learning algorithms demonstrate the effectiveness and high accuracy of the proposed method.

*Index Terms*—Reconstruction loss, autoencoders, voltage stability, long-short-term memory (LSTM), feature moving strategy.

## I. INTRODUCTION

WITH the continuous increase of the penetration of renewable generations and flexible consumers, power systems are more likely to operate near the voltage collapse point (VCP), which makes the voltage stability more critical for the security and economy of modern power systems [1]-[3]. Hence, an accurate and robust real-time evaluation method for voltage stability is urgently required, which benefits system utilities to timely take certain control actions to

DOI: 10.35833/MPCE.2020.000526

avoid potential accidents.

In recent years, with the wide deployment of phasor measurement unit (PMU), a huge amount of high-resolution data has been collected, which prompts many researchers to study data-driven methods for VSA. Data-driven methods analyze the behaviour of power systems from the perspective of measuring data, and they require no prior knowledge of the complex model and parameters of power systems. Therefore, it can circumvent the information loss caused by manual assumptions and simplifications which generally occur when using traditional model-based methods [4]-[6]. In addition, the emerging deep learning methods provide increasing opportunities to create more flexible and precise algorithms for VSA. In [7], an artificial neural network (ANN) based method is proposed to estimate the voltage stability margin (VSM) based on temporal measuring variables such as nodal voltages, branch currents and nodal injective power. References [8] and [9] examine the relationship between reactive power reverse (RPR) and VSM, and use multilinear regression methods to fit the relationship in order to predict VSM. In [10], a random forest based method is presented to classify the secure region and insecure region defined as different ranges of distance to VCP. It is tolerant for data loss and is adaptive to topology changes. Reference [11] utilizes maximal information coefficient (MIC) [12] to select the most representative variable, following which a polynomial fitness is implemented for VSA. But it requires to simulate a database of a considerable size that is potentially unrealistic to be built. Besides, a missing-data-tolerant approach based on the overlapped grouping of PMUs is proposed to judge the stability when some transient faults occur [13]. In [14], the spectrum estimation of the spatial-temporal matrix of PMU measurements is used for online evaluation of voltage stability. It investigates the random fluctuations included in PMU data by random matrix theory (RMT), and cleans these uncertainties by modifying the eigenvalues. In [15], an ensemble learning method constituting many extreme learning machines (ELM) is proposed to analyze pattern-recognitionbased transient stability (PRBTS). It operates faster than traditional neural network based approaches by a large margin, since no iterative updates for parameters are required. And then, to adapt to unpredicted changes of system topology and parameters, an active learning framework is proposed to update the learning model and training data pool [16].

JOURNAL OF MODERN POWER SYSTEMS AND CLEAN ENERGY



Manuscript received: July 27, 2020; accepted: October 26, 2020. Date of CrossCheck: October 26, 2020. Date of online publication: November 26, 2020. This article is distributed under the terms of the Creative Commons Attribu-

tion 4.0 International License (http://creativecommons.org/licenses/by/4.0/). H. Yang (corresponding author), R. C. Qiu, and H. Tong are with the Depart-

ment of Electrical Engineering, Center for Big Data and Artificial Intelligence, Shanghai Jiaotong University, Shanghai 200240, China, and R. C. Qiu is also with School of Electronic Information and Communication, Huazhong University of Science and Technology, Wuhan 430000, China (e-mail: 31910019@sjtu. edu.cn; reqiu@sjtu.edu.cn; thj 926@sjtu.edu.cn).

Even though most of literature has obtained promising accuracy, one critical drawback is that their performance highly depends on a large training data set, which must contain the data of each stage of the voltage stability deterioration process. Namely, from the view of P-V curve, data set is required to include certain data of each operation point above the VCP. However, in practice, the data of insecure state, i. e., at a point close to the VCP, is very rare, since power systems normally operate in secure state [17]. To implement the above-mentioned methods in reality, digital simulation or data argument is required to generate large number of data of insecure state. But there must exist certain difference between data coming from digital simulation and real measured data, since the simulation model and parameters are difficult to be precisely constructed. The similar comments are raised for data argument [18]-[20]. And by the experience of real validation, directly transferring a learning model trained by simulation data into actual environment yields heavy effectiveness loss. Therefore, the lack of data of insecure state is a major limit for traditional learning methods for VSA to operate in real environment.

Faced with this dilemma, this paper proposes a method which avoids using the data of insecure state, i.e., this method is purely trained by the data of secure state which is easily accessible. The idea behind the proposed method comes from a new perspective that the reconstruction loss of a wellmodified autoencoder is effective to indicate the change of data distributions [21]-[23]. Autoencoder is typically a kind of representation learning method, which is originally proposed to extract nonlinear representative features or compress large-dimensional data. In the past, autoencoders were widely applied in many areas in power systems, including state estimation [24], load forecasting [25], malicious data detection [26], and voltage stability [27]. Reference [27] forms VSA as a problem to discover representative latent variables, and utilizes variational autoencoder (VAE) to acquire probabilistic features. But it also needs massive simulations to obtain the data of insecure state. These methods are all direct applications of autoencoders. They focus on the extracted features at the middle layer of autoencoders. Different from them, we investigate the information contained in the reconstruction loss to design a criterion to divide the secure state and insecure state. Specifically, an autoencoder purely trained by secure data is expected to only recover secure data properly, while it fails to reconstruct insecure data. Namely, the data of secure state is expected to produce lower reconstruction residual, while the data of insecure state is prone to yield higher reconstruction loss. Based on this principle, data-driven VSA can be implemented without any data of insecure or unstable state. Besides, to develop the proposed method, we introduce a moving strategy for middle features of autoencoders to narrow the difference of current data from the feature domain of secure data. Thus, insecure features are adjusted to be more similar to secure ones, leading to higher error rate to recover the insecure data. In addition, to better utilize the spatial-temporal correlation, long short-term memory (LSTM) layers combined with fully-connected (FC) layers are utilized to construct the internal structure of our autoencoder. Through these improvements, the reconstruction residual becomes more representative and the classification accuracy is greatly improved. Numerous experiments and comparison with other well-known machine learning algorithms illustrate the effectiveness and accuracy of the proposed method.

The main contributions of our work are summarized as follows:

1) A novel data-driven framework based on the reconstruction residual of autoencoders is proposed to evaluate voltage stability of power systems. The training of this method merely requires the data of secure state which is easy to collect, thus it is no longer subjected to the limitation that the practical data of insecure state is not sufficient.

2) To enhance the performance of the proposed method, a moving strategy for middle features is utilized to enhance the similarity between the features of testing data and the secure feature domain formed by the training data.

3) The proposed method is compared with other machine learning methods in imbalanced data manner and other types of autoencoders. The results demonstrate that the proposed method outperforms traditional algorithms for imbalanced data.

4) Multiple tests in different power systems are conducted. In addition to the classification accuracy, the computation cost and effects of measurement errors are analyzed empirically.

The remainder of this paper is organized as follows. Section II reviews the basics of voltage stability and introduces the background knowledge of autoencoders and the main principle. Section III introduces the entire methodology and improvements of the proposed method, including LSTM layers and feature moving strategy. Section IV introduces case studies, and Section V summarizes the research.

## II. BACKGROUND KNOWLEDGE AND BASICS

## A. Voltage Stability

Long-term voltage stability, mostly suffering from load demand increments and unexpected changes of slowly acting equipment, involves the steady-state power system model that is described by an algebraic equation [28]:

$$F(\boldsymbol{V}_t, \boldsymbol{\lambda}_t, \boldsymbol{P}_0) = 0 \tag{1}$$

where  $V_t$  is the vector of state variables including nodal voltage magnitudes and angles observed from PMU at time t;  $\lambda_t$ is the loading factor portraying the gradual increase of load demand; and  $P_0$  is the initial load level. In general, active load, reactive load and generator outputs at different buses increase at different rates, hence, the load growing model is commonly written as linear equations:

$$\begin{cases} P_{i}^{t} = P_{i0} (1 + \lambda_{i} k_{Pi}) \\ Q_{i}^{t} = Q_{i0} (1 + \lambda_{i} k_{Qi}) \\ G_{j}^{t} = G_{j0} (1 + \lambda_{i} k_{Gj}) \end{cases}$$
(2)

where  $P_i^t$  and  $Q_i^t$  are the active power and reactive power demand of load *i* at a certain time *t*, respectively;  $G_j^t$  is the power output from generator *j* at time *t*;  $P_{i0}$  and  $Q_{i0}$  are the load demand of initial status;  $G_{j0}$  is the generator output in the basic case; and  $k_{Pl}$ ,  $k_{Ql}$ ,  $k_{Gj}$  are multiplicative factors describing the growth rates of various variables as mentioned above.

The *P-V* curve, as shown in Fig. 1, is the most straightforward way for VSA. The VCP  $(\lambda_{\max}, V^{\lambda_{\max}})$ , as the maximum point of the loading factor, is a critical boundary separating stability and instability domains. Exceeding the VCP, i. e., power system is operating at the lower part of *P-V* curve, will lead to a collapse of power system. Therefore, the distance of current load demand to the VCP, namely the VSM, has been widely studied as an indicator to assess voltage stability [29]. According to the voltage stability requirements formulated by western electricity coordinating council (WECC), an operation state is identified as secure state of voltage stability if current VSM is no less than 7% of a basic operation state:

$$\frac{m_t}{m_b} \ge 7\% \tag{3}$$

where  $m_i$  and  $m_b$  are the VSM at current time t and a base value, respectively. Otherwise, the operation state is judged as insecure state if  $m_i/m_b < 7\%$ , as shown by the red area in Fig. 1.



Fig. 1. Visualization of P-V curve.

## B. Measurement Data for VSA

The high-resolution and high-accuracy PMU measurements are used to assess voltage stability in this paper. PMU measurements contain nodal voltage magnitudes and angles, nodal injective active and reactive power, and branch currents. Voltage magnitudes and angles are selected for VSA because nodal voltages are the most representative measurement variables for voltage stability, and they are sensitive to the change of operation state. So the measurement vector at time t is  $\mathbf{x}_t = [V_{t1}; V_{t2}; ...; V_{tn}; \theta_{t1}; \theta_{t2}; ...; \theta_m] \in \mathbb{R}^{2n \times 1}$ , where n is the number of buses.

We use a split window that slides on measurement sequence of nodal voltage magnitudes and angles to collect a period of data  $x_{t-L:t} = [x_{t-L+1}, x_{t-L+2}, ..., x_t] \in \mathbb{R}^{2n \times L}$ , where *L* is the length of the split window. Each two-dimensional data slice  $x_{t-L:t}$  is used as a data unit to input into the proposed method.

### C. Autoencoder

Autoencoder is an auto-associative neural network that recovers the input data in the output from a compressed representation in low dimension. Autoencoder cascades an encoder and a decoder that are used to extract low-dimensional features and reconstruct the input data from these features, respectively. As shown in Fig. 2, the output of the middle layer is a low-dimensional feature vector supposed to reflect the important characteristics of the data, and the input and output of an autoencoder are identical. The calculation involved in each neural layer in an autoencoder can be described as:

$$\boldsymbol{x}_{(k)} = \sigma(\boldsymbol{W}_{(k)}\boldsymbol{x}_{(k-1)} + \boldsymbol{b}_{(k)}) \tag{4}$$

where  $\mathbf{x}_{(k)}$  and  $\mathbf{x}_{(k-1)}$  are the outputs of the  $k^{\text{th}}$  layer and the  $(k-1)^{\text{th}}$  layer, respectively;  $\mathbf{W}_{(k)}$  and  $\mathbf{b}_{(k)}$  are the weight matrix and bias which need to be optimized, respectively; and  $\sigma(\cdot)$  is the activation function. The training of a simple autoencoder is to minimize the reconstruction loss that is usually described by root mean square error (RMSE):

$$L = \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2 \tag{5}$$

where  $\hat{x}$  and x are the reconstruction data and the input data, respectively.



Fig. 2. A schematic diagram to visualize structure of an autoencoder.

Autoencoders have been widely studied and modified to quite a few variants such as sparse autoencoder (SAE) [30], variational autoencoder (VAE) [31], adversarial autoencoder (AAE) [32]. And the internal structure of the encoder and the decoder can also be formed by convolutional neural network (CNN) and recurrent neural network (RNN) to adapt to different tasks. But these studies to enhance autoencoder mainly concern how to extract more representative features, while the reconstruction loss is not of interest. In this study, we employ autoencoders from the opposite perspective that utilizes the information contained in reconstruction loss.

## D. Reconstruction Residual Based VSA

For a data set  $S = \{x_{t-L:t} | t \in L + 1:T_s\}$  which only contains the data of secure state, where  $T_s$  is the total length of measuring time series in S, each element  $x_{t-L:t}$  is input into the encoder to generate a feature vector  $z_{t-L:t} \in \mathbb{R}^{n_z \times 1}$ , where  $n_z \ll 2n$  is the length of a feature vector. And then the decoder outputs the reconstructed data  $\hat{x}_{t-L:t}$  from  $z_{t-L:t}$ . Thus, we have a set of features  $\mathcal{Z}_S = \{z_{t-L:t} | t \in L + 1:T_S, x_{t-L:t} \in S\}$ , and a set of reconstruction loss  $\mathcal{L}_S = \{||x_{t-L:t} - \hat{x}_{t-L:t}||_2|t \in L + 1:T_S, x_{t-L:t} \in S\}$ . Consider a data set of insecure state  $\mathcal{I} = \{u_{t-L:t} | t \in L + 1:T_t\}$ , where  $T_t$  is the total length of measuring time series in  $\mathcal{I}$ . We obtain a set of feature vectors  $\mathcal{Z}_{\tau}$  =  $\{v_{t-L:l} | t \in L+1: T_l, u_{t-L:l} \in \mathcal{I}\}$ , and a set of reconstruction residuals  $\mathcal{L}_{\mathcal{I}} = \{ \| \boldsymbol{u}_{t-L:t} - \hat{\boldsymbol{u}}_{t-L:t} \|_2 | t \in L + 1 : T_t, \boldsymbol{u}_{t-L:t} \in \mathcal{I} \}$ , where  $\boldsymbol{v}_{t-L:t}$  is the feature vector obtained by  $\boldsymbol{u}_{t-L:t}$ . Autoencoders are essentially learning an identity function, but they first compress data into low-dimensional features and then reconstruct it. The low-dimensional features must lose certain information compared with the input data, hence, the reconstruction residuals cannot reach zero. By only inputting secure data for training, autoencoders learn how to recover secure data but are unfamiliar with the data of insecure state. Specifically, after proper training of autoencoders by mere secure data, the reconstruction loss of each element in the set S is expected to become lower values. However, for the data of insecure state which is not included in the training set S, autoencoders are expected to fail to recover it, i.e., the reconstruction loss is significantly greater. According to the reconstruction loss, a threshold is introduced to classify the secure and insecure states.

Remark: Traditional studies based on autoencoders unanimously attempt to enhance the representative ability of feature vectors [25]-[27]. For the issue of VSA, to obtain representative features, their training must use a certain amount of insecure data. However, for the proposed method, since the insecure data set  $\mathcal{I}$  is not used for training, the features in  $\mathcal{Z}_{\mathcal{I}}$  are distributed disorderly and are therefore not clearly separated from the features of secure state  $\mathcal{Z}_{\mathcal{S}}$ . Thus, it is infeasible to divide the secure and insecure states by the difference of these feature vectors.

## III. PROPOSED METHODOLOGY

Now we introduce the enhancement and details for the proposed methodology, including the explicit structure of the LSTM autoencoder and the moving strategy for compressed features.

#### A. Potential Disadvantages

Although we can implement VSA by using a simple autoencoder, the accuracy and robustness still need to be improved. The drawbacks are explicitly listed as follows.

1) VSA is typically a multi-variate time series issue that not only the spatial correlation between different measuring variables exists but also the temporal correlation is very critical [17]. Since VSMs of the same operation point with different load increment directions are totally dissimilar, VSA using measurements of only one time point brings certain level of estimation error [33]. However, simple autoencoder, as shown in Fig. 2, can only input an  $n_{in} \times 1$  vector of input data, where  $n_{in}$  is the number of neural units in the first layer. The temporal correlation is not considered.

2) The proposeed method is based on the principle that, for the data quite different from the training data, its reconstruction is expected to be very poor. Nevertheless, in practice, autoencoders may generalize well, i. e., insecure data can also be properly represented and recovered. Specifically, even though the data corresponding to secure and insecure state has completely different distributions, the insightful operation mechanism of the power system shown in (1) remains unchanged. Thus, it is likely that the autoencoderbased model also understands the operation pattern of the power system, and the reconstruction loss of the insecure data is still lower, which will cause the decrease of the effectiveness of the proposed method.

## B. LSTM Autoencoder

Aiming to solve the first issue of temporal correlation, a multiple-layer LSTM is embedded in this work before FC layers to construct the encoder. While in the decoder, multiple-layer LSTM is successively connected after FC layers, as the opposite of the encoder.

LSTM is an enhanced variant of RNN that not only connects output and input data but also connects current cell state with the previous state. LSTM has achieved state-ofthe-art performance in many research areas associated with time sequence analysis. In the field of power system, it has been successfully applied for load forecasting [34], solar generation forecasting [35], transient stability analysis [36] and energy disaggregation [37]. The detailed knowledge of RNN is given in [38]. An LSTM cell, whose structure is shown as Fig. 3, consists of an input gate, a forget gate, and an output gate, where  $\sigma$  is a single FC layer with sigmoid activation function; tanh is an FC layer of tanh activation function; "x" with a green circle denotes the element-wise multiplication;  $h_{i}$ ,  $c_{i}$ ,  $h_{i-1}$ , and  $c_{i-1}$  are the hidden states and cell states at time t and the last time t-1, respectively; and  $x_t$  is the input data.



Fig. 3. Structure of an LSTM cell.

The explicit calculation of an LSTM cell is listed as follows:

$$\begin{cases} z_{i}^{t} = \sigma(W_{i}\boldsymbol{x}_{t} + U_{i}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{i}) \\ z_{t}^{f} = \sigma(W_{f}\boldsymbol{x}_{t} + U_{f}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{f}) \\ \hat{\boldsymbol{c}}_{t} = \tanh(W_{c}\boldsymbol{x}_{t} + U_{c}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{c}) \\ z_{t}^{o} = \sigma(W_{o}\boldsymbol{x}_{t} + U_{o}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{o}) \\ \boldsymbol{c}_{t} = z_{t}^{f} \cdot \boldsymbol{c}_{t-1} + z_{t}^{i}\hat{\boldsymbol{c}}_{t} \\ \boldsymbol{h}_{t} = z_{t}^{o} \cdot \tanh(\boldsymbol{c}_{t}) \end{cases}$$
(6)

where  $z_i^t$ ,  $z_i^c$ ,  $z_i^o$  are outputs of the input gate, forget gate, output gate, respectively; and  $W_i$ ,  $W_f$ ,  $W_c$ ,  $W_o$ ,  $U_i$ ,  $U_f$ ,  $U_c$ ,  $U_o$ ,  $D_i$ ,  $b_f$ ,  $b_c$ ,  $b_o$  are the weights and bias to be optimized.  $\sigma(x) = 1/(1+e^{-x})$  and  $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$  are the sigmoid and tanh activation function, respectively. The input gate aims to extract favorable information from the input data  $x_i$  and the hidden state of the last time  $h_{i-1}$ . The forget gate is used to

decide to drop or deliver the variables of the last cell state  $c_{t-1}$ , and the output gate is used to construct the current hidden state  $h_{t}$ .

## C. Feature Moving Strategy

To solve the second issue mentioned in Section II-A, we introduce a moving strategy for extracted features to make insecure features more similar to secure features, so that the reconstruction of insecure data is impeded. As shown in Fig. 4, 500 sampled spatial-temporal matrices, including 134 insecure data and 366 secure data, are used to visualize the distribution of extracted features. To facilitate the visualization, extracted features are set as two-dimensional vectors. As shown in Fig. 4, the features of secure state are formed in an around diagonal line by the LSTM autoencoder, as shown by the green points in a rectangle domain. Since the LSTM autoencoder is well trained by the data of pure secure state, the features residing in secure domain  $Z_s$  are expected to generate corresponding secure data similar to the input. Namely, one location in the domain  $Z_s$  is mapped into a particular input data slice by the decoder, and hence, for the data of secure state, the input data can be reconstructed precisely. But since insecure data is not included in the training data set, the insecure feature domain  $\mathcal{Z}_{I}$  is not well formed and may be overlapped with secure domain  $Z_s$ , as shown by the red points in Fig. 4.



Fig. 4. Visualization of two-dimensional extracted features by LSTM autoencoder with a total of 500 feature points.

The features of insecure data contain two types: one is overlapped with secure domain  $\mathcal{Z}_{s}$ , i.e.,  $v_{t-L:t} \in \mathcal{Z}_{t} \cap \mathcal{Z}_{s}$ , the other is not overlapped with  $Z_s$ . If a feature of insecure data falls in the domain  $\mathcal{Z}_1 \cap \mathcal{Z}_s$ , it will yield a reconstruction data matrix that belongs to secure states. Thus, the reconstruction residual is relatively great and the insecurity is detected. At the opposite, it is very difficult to judge whether the features outside the secure domain (i.e.,  $v_{t-L,t} \in \mathcal{Z}_t - \mathcal{Z}_s$ ) are capable of producing high reconstruction loss. The reason is that autoencoders are likely to possess promising generality to understand the insightful operation pattern of VSA, so that a small part of insecure data may also be recovered accurately. Faced with this dilemma, we propose a method to move these insecure features to secure feature domain, and thus, they are prone to yield very high reconstruction residuals.

The moving algorithm of extracted features should satisfy some rules as follows: ① the moving operation requires to

be embedded into the LSTM autoencoder, and they are optimized jointly with the entire autoencoder model; ② not only are insecure features processed by our moving strategy but also secure features are moved in the identical way, for no prior knowledge about the testing data is available. Therefore, the moving strategy should change insecure features significantly with little impact on the features of secure data.

In this work, an enhanced *K*-nearest neighbor (KNN) method is proposed to move extracted features to the center of a certain number of their nearest neighbors in the training data set. Given a pre-trained LSTM autoencoder and an incoming insecure data matrix  $u_{t-L:t}$ , the extracted feature vector  $v_{t-L:t}$  is obtained by the encoder. The distance between  $v_{t-L:t}$  and a feature vector in  $Z_s$  is measured by the cosine similarity:

$$d(\mathbf{v}_{t-L:t}, \mathbf{z}_{i}) = \frac{\mathbf{v}_{t-L:t}^{\mathrm{T}} \cdot \mathbf{z}_{i}}{\|\mathbf{v}_{t-L:t}\| \|\mathbf{z}_{i}\|}$$
(7)

where  $z_i$  is the *i*<sup>th</sup> feature vector in the domain  $Z_s$  generated by the training data set;  $v_{t-L:t}^{T}$  is the transpose of  $v_{t-L:t}$ ; and  $\|\cdot\|$  is the 2-norm. Based on the cosine similarity, feature vectors in  $Z_s$  are ranked, and the top *K* features are selected. The weighted center of these features is:

$$\boldsymbol{c}_{vt} = \sum_{i=1}^{K} \boldsymbol{w}_i \boldsymbol{z}_i \tag{8}$$

$$w_{i} = \frac{\exp(d(v_{t-L:t}, z_{i}))}{\sum_{i=1}^{K} \exp(d(v_{t-L:t}, z_{i}))}$$
(9)

where  $c_{vt} \in \mathbb{R}^{n_z \times 1}$  is the center of the selected *K* features; and  $d(\cdot)$  is the distance between  $v_{t-L:t}$  and  $z_t$ , which is used as the weights. The center is a linear combination of selected features, thus it must reside into the domain  $\mathcal{Z}_{s}$ .

In order to establish a more flexible moving strategy, a temperature parameter is introduced to decide the degree of moving towards the center  $c_{vv}$ . The final point after moving is:

$$\boldsymbol{v}_{t-L:t}^{m} = \alpha \boldsymbol{c}_{vt} + (1-\alpha)\boldsymbol{v}_{t-L:t}$$
(10)

where  $\alpha$  is the temperature parameter to control the moving degree. The principle of this moving strategy is that the influence for secure features by this moving is very little, because secure features in the domain  $Z_s$  must have numerous very adjacent points in the training data set, i.e., the nearest neighbors in the training data set of a secure feature are very close to this feature. Therefore, the distance between a secure feature vector and the center  $c_w$  of its neighbors is not great. As for the feature of partial insecure data outside the feature domain  $Z_s$ , its nearest neighbors in the training data set are distant. Therefore, the moving strategy enables insecure features to move to the secure domain  $Z_s$ , which biases the data reconstruction by the decoder to improve the reconstruction loss.

## D. Entire Framework

This moving strategy is embedded in the middle of the LSTM autoencoder, as shown in the block diagram in Fig. 5, and the entire framework of the proposed method is shown in Fig. 6.



Fig. 5. Block diagram of proposed method.

The operation of the moving strategy requires a set of secure features, and it is inevitable that the moving strategy

will slightly change the secure features. Hence, we need to obtain a set of secure features before operating the feature moving strategy. A two-stage training process is designed to tackle this issue, as shown in Fig. 7, which contains the main training procedure and the slight tuning after the moving strategy. The main training is used to optimize the weights by pure secure data. Thus, massive extracted features corresponding to the training data set are acquired. This main training process is implemented without the moving strategy, i.e., the features extracted from the encoder are directly input into the decoder. The tuning of moving strategy is a successive training procedure based on the weight parameters obtained from the main training process. It considers the moving strategy and optimizes the weight parameters to enable the autoencoder model to obtain low reconstruction loss of secure data. The training is based on Adam algorithm [39], and the initialization of weight parameters utilizes the Xavier initialization approach [40].



Fig. 6. Entire framework of proposed method.



Fig. 7. Diagram of training and testing process.

However, there exist some limits for the application of the proposed method. At first, a certain number of PMUs are required, because autoencoders, or more general, learning algorithms need sufficient data to acknowledge the operation information of power systems and implement classification tasks. Secondly, for large-scale systems which yield a large amount of data, sufficient computation resources are required for offline training and online testing.

## **IV. CASE STUDIES**

In this section, we prove the effectiveness and accuracy of the proposed method by numerous experiments. The first case elaborates the calculation process, including data generation, parameter settings and training. While the second case compares our method with other autoencoders based methods to verify the effectiveness of the proposed feature mov-

ing strategy. And the performances in different testing systems are demonstrated, including IEEE 30-bus, 57-bus, 118bus systems, and European high-voltage transmission 1354bus network [41]. In the third case, some traditional machine learning methods which also need a small amount of insecure data are compared.

## A. Explicit Procedures

This case aims to introduce the explicit process and configuration of our method by using IEEE 57-bus system.

# 1) Data Pool Generation

The data pool contains three types of load changing directions. One is random increase rate of randomly selected growing load. We randomly select around 30% active and reactive load to increase at different rates. And the ascending rates are randomly assigned by a uniform distribution  $U(0, 0.001P_b)$ , where U is the uniform distribution; and  $P_b$  is a base value of corresponding load. The second increasing model is that a single load grows while other load keeps unchanged. The final way is that the load of a specific area ascends simultaneously, and every load increases in its respective rate which is randomly selected by  $U(0, 0.001P_{h})$  w.r.t. the loading factor. The segmentation for IEEE 30-bus and 57-bus systems is based on the partition described in their documents, while the segmentation of IEEE 118-bus system

is described in [42]. MATPOWER 6.0 is used to generate the data set by continuous power flow (CPF), and our autoencoder based model is coded by PyTorch.

# 2) Model Configuration

Now we list the explicit structure and hyper-parameters of our method as shown in Table I. The rules of the choice of hyper-parameters include: (1) even though the objective of the proposed method is not to pursue high reconstruction accuracy but to reveal the difference of secure data and insecure data, the method is still subjected to the notorious overfitting. The reason is that overfitting not only hinders the reconstruction of insecure data, but also causes high reconstruction loss of secure data in testing, as it is mainly caused by the stochastic noise imposed on measurements. Therefore, like normal training of deep neural networks, the number of neural layers and the number of neural units in each layer should be as small as possible only if the reconstruction accuracy is ensured. However, in practice, similar to most of methods based on neural network, the choice of explicit values of hyper parameters mainly depends on the experience and numerous experiments; (2) as for the length of the split window, it involves a trade-off that short sliding window only contains limited information, which is not beneficial for the proposed model. A too long split window will bring more computation burden. Therefore, numerous simulations should be conducted to decide the optimal length of sliding window. The similar situation exists for the batch size when the proposed model is trained; 3 regarding the learning rate, it also involves a trade-off that low learning rate is prone to cause a too slow training process, while high learning rate makes it difficult to reach the optimal solution.

 TABLE I

 Configuration of Proposed Method in Section IV-A

Block	Hyper parameters	
Input data	Number of nodes: 57 Number of PMUs: 17 Split window length: 25 Input matrix size: 34×25	
Encoder	Number of LSTM layers: 2 Size of cell state: 16 Size of hidden state: 16 Number of FC layers: 3 Number of units in each FC layer: 16, 8, 2	
Decoder	Number of LSTM layers: 2 Size of cell state: 16 Size of hidden state: 16 Number of FC layers: 3 Number of units in each FC layer: 8, 16, 25	
Moving strategy	Number of nearest neighbors: 4 Temperature parameter: 0.75	
Batch size: 10 Optimization method: Adam Learning rate: 0.00006 Parameter initialization method: Xav		
Criterion	Reconstruction loss: 0.2836	

## 3) Assessment Indicators

The indicator to evaluate the performance of our method follows the well-known f1 index calculated by the classification precision and recall rate. The precision  $Pr_0$  is defined as:

$$Pr_{0} = \frac{T_{0}}{T_{0} + F_{0}}$$
(11)

where  $T_0$  and  $F_0$  are the number of correct and false classification results, respectively, if judging them into secure class.  $Pr_0$  is to measure how many correct classification results if the proposed model recommends secure states. The recall rate  $Rc_0$  of secure state is:

$$Rc_0 = \frac{T_0}{T_0 + F_1}$$
(12)

where  $F_1$  is the number of incorrect results of the insecure class.  $Rc_0$  essentially measures how much secure data is correctly clustered w.r.t. the total number of secure data. Then the  $fl_0$  index is defined as:

$$fl_{0} = \frac{2Pr_{0} \cdot Rc_{0}}{Pr_{0} + Rc_{0}}$$
(13)

Similarly, the precision, recall rate and  $fl_1$  index for insecure data are defined as:

$$\begin{cases}
Pr_{1} = \frac{T_{1}}{T_{1} + F_{1}} \\
Rc_{1} = \frac{T_{1}}{T_{1} + F_{0}} \\
fl_{1} = \frac{2Pr_{1} \cdot Rc_{1}}{Pr_{0} + Rc_{1}}
\end{cases}$$
(14)

where  $T_1$  is the number of correct results if judging them into insecure class. The weighted f1 index, which assigns different concerns to secure and insecure class according to their respective number of samplings, is used as the final indicator to evaluate the performance:

$$fl = \frac{(T_0 + F_1) \cdot fl_0 + (T_1 + F_0) \cdot fl_1}{T_1 + T_0 + F_1 + F_0}$$
(15)

Also, the weighted precision and weighted recall rate are defined as:

$$\begin{cases} Pr = \frac{(T_0 + F_1) \cdot Pr_0 + (T_1 + F_0) \cdot Pr_1}{T_1 + T_0 + F_1 + F_0} \\ Rc = \frac{(T_0 + F_1) \cdot Rc_0 + (T_1 + F_0) \cdot Rc_1}{T_1 + T_0 + F_1 + F_0} \end{cases}$$
(16)

4) Criterion

Based on the training by pure secure data, the proposed LSTM autoencoder with moving strategy successively divides the secure and insecure data. As shown in Fig. 8, by embedding moving strategy, the distribution of features of insecure data is totally changed and fully falls into the domain of secure data. Thus, the reconstruction loss of insecure data significantly ascends, and it is easier for us to divide the secure and insecure data. To visualize the obtained reconstruction residuals, 500 samplings are randomly selected and the probability density function (PDF) of reconstruction residuals is shown in the form of histogram. As shown in Fig. 9, most of reconstruction residuals from secure domain and insecure domain are separated well.



Fig. 8. Visualization of two-dimensional extracted features with moving strategy.



Fig. 9. Histogram of distribution of reconstruction residuals.

The criterion of reconstruction loss serving as the classification boundary is critical for the accuracy of the proposed method. We test different criterions and visualize the precision, recall rate and f1 index of both secure and insecure data as shown in Fig. 10(a), and weighted averages of these indicators are shown in Fig. 10(c). To demonstrate the improvement of the presented moving strategy, the performance of simple LSTM autoencoder without moving strategy is also shown for comparison, as shown in Fig. 10(b) and (d), respectively.



Fig. 10. Changing trends of precision, recall rate and f1 index with criterion growing, using moving strategy or not. (a) Classification report. (b) Classification report without moving strategy. (c) Weighted average of classification report. (d) Weighted average of classification report without moving strategy.

#### 5) Discussions

Based on the simulation results mentioned above, several properties of the proposed method are revealed. At first, our method effectively implements VSA by pure secure data, which overcomes the disadvantage that insecure data or even unstable data is extremely rare. And through the proposed moving strategy, the classification accuracy is significantly improved. The best f1 index without moving strategy is 0.9279, while the best f1 index of our method embedding moving strategy is 0.9787, demonstrating the effectiveness and high accuracy of our method. The best criterion for classification also increases, since the feature moving strategy encourages higher reconstruction loss rate for insecure data, so secure data and insecure data are separated more clearly. The best criterion for simple LSTM autoencoder is 0.2836.

# 6) Impact of Measurement Noise

Even if PMU has achieved enormous popularity for its high accuracy and resolution, measurement noise inevitably exists in the collected data of PMU. Higher measurement error will hinder the learning of the proposed method and may lead to overfitting. Therefore, it is infeasible to ignore the existence of measurement noise. To show the affect of measurement noise on our method, the classification results with different magnitudes of measurement noise are tested, as shown in Table II. According to the results, with the growth of measurement noise magnitude, the classification accuracy evidently decreases. But even with 5% measurement error which is larger than the normal error rate of PMU by a large margin, the proposed method still has 0.8960 accuracy. This shows that the proposed method maintains robustness when imposing higher measurement noise, even the existence of this noise incurs a certain extent of effectiveness loss.

TABLE II CLASSIFICATION RESULTS OF VARIOUS MAGNITUDES OF MEASUREMENT NOISE

Noise magnitude (%)	Precision	Recall rate	f1 index
0.0	0.9792	0.9782	0.9787
0.2	0.9487	0.9545	0.9516
0.5	0.9568	0.9411	0.9489
1.0	0.9538	0.9403	0.9470
2.0	0.9239	0.8905	0.9069
5.0	0.9089	0.8834	0.8960

## B. Comparison with Other Autoencoders

In this subsection, a large number of tests are implemented to compare the proposed method with other autoencoders using IEEE 30-bus, 57-bus, 118-bus systems, and European high-voltage transmission 1354-bus system. The criterions are selected by numerous tests as the first case. The detailed structure and hyper-parameters are listed in Table III and the results are shown in Table IV.

Block	IEEE 30-bus	IEEE 118-bus	European 1354-bus
Input data	Number of nodes: 30	Number of nodes: 118	Number of nodes: 1354
	Number of PMUs: 10	Number of PMUs: 32	Number of PMUs: 478
	Split window length: 25	Split window length: 32	Split window length: 64
	Input matrix size: 20×25	Input matrix size: 64×32	Input matrix size: 956×64
Encoder	Number of LSTM layers: 1	Number of LSTM layers: 3	Number of LSTM layers: 5
	Size of cell state: 12	Size of cell state: 16	Size of cell state: 16
	Size of hidden state: 12	Size of hidden state: 16	Size of hidden state: 16
	Number of FC layers: 2	Number of FC layers: 5	Number of FC layers: 5
	Number of units in FC layers: 8, 2	Number of units in FC layers: 16, 12, 8, 4, 2	Number of units in FC layers: 32, 16, 8, 4, 2
Decoder	Number of LSTM layers: 1	Number of LSTM layers: 3	Number of LSTM layers: 5
	Size of cell state: 12	Size of cell state: 16	Size of cell state: 16
	Size of hidden state: 12	Size of hidden state: 16	Size of hidden state: 16
	Number of FC layers: 2	Number of FC layers: 5	Number of FC layers: 3
	Number of units in FC layers: 8, 25	Number of units in FC layers: 4, 8, 12, 16, 32	Number of units in FC layers: 4, 8, 16, 32, 64
Moving	Number of nearest neighbors: 4	Number of nearest neighbors: 4	Number of nearest neighbors: 4
strategy	Temperature parameter: 0.75	Temperature parameter: 0.75	Temperature parameter: 0.75
Training	Batch size: 10	Batch size: 10	Batch size: 20
	Optimization method: Adam	Optimization method: Adam	Optimization method: Adam
	Learning rate: 0.00006	Learning rate: 0.0001	Learning rate: 0.0005
	Parameter initialization method: Xavier	Parameter initialization method: Xavier	Parameter initialization method: Xavier
Criterion	Reconstruction loss: 0.0760	Reconstruction loss: 0.2760	Reconstruction loss: 0.5134

=

 TABLE III

 CONFIGURATION OF PROPOSED METHOD IN DIFFERENT TESTING SYSTEMS

 TABLE IV

 Weighted fl Indices of Numerous Unsupervised Methods Using

 Different Testing Systems

	Weighted <i>f</i> 1 index			
Method	IEEE 30-bus	IEEE 57-bus	IEEE 118-bus	European 1354-bus
Proposed	0.9912	0.9787	0.9956	0.9671
LSTM-AE	0.9467	0.9279	0.9234	0.9005
CNN-AE-MS	0.9678	0.9478	0.9760	0.9270
SAE-MS	0.9033	0.8757	0.8312	0.7459
VAE-MS	0.8956	0.9204	0.9079	0.8345

The structure complexity of our model is changed according to different scales of testing systems. Larger systems such as European 1354-bus system and IEEE 118-bus system, which have installed 478 and 32 PMUs, respectively, have more complicated physical relationships that require more neural layers and more neural units to fit. For IEEE 118-bus system, we use three LSTM layers and five FC layers to construct the encoder and the decoder, and the criterion of reconstruction loss is 0.276. For European 1354-bus network, five LSTM layers and five FC layers are utilized, while the criterion of reconstruction residual is 0.5134. As for the IEEE 30-bus system containing 10 PMUs, only one LSTM layer and two FC layers are employed, and the criterion of reconstruction residual is 0.076. By comparison, the proposed method greatly outperforms other autoencdoerbased methods, including CNN-based autoencoder, SAE, and VAE, illustrating that the LSTM layer focusing on the temporal correlation is more beneficial for VSA. And the time of 500 operations is tested on Nvidia GeForce GTX 1080 (8G) GPU, as shown in Table V.

 TABLE V

 COMPARISON RESULTS OF COMPUTING EFFICIENCY OF 500 OPERATIONS

Method	Operation time (s)
Proposed	0.5770
LSTM-AE	0.4165
CNN-AE-MS	0.5239
SAE-MS	0.3391
VAE-MS	0.8560

### C. Comparison with Other Methods

In this case, the proposed method is compared with other machine learning based methods, including one-sided support vector machine (OS-SVM), cost-sensitive decision tree (CSDT), and cost-sensitive random forest (CSRF). More information on these machine learning methods can be found in [43]. OS-SVM, CSDT and CSRF unanimously require a small amount of insecure data, which is not similar to the proposed method that operates without any insecure data. Therefore, in addition to the classification precision, the requirement on data amount and computation efficiency is also under great concern.

# 1) OS-SVM

OS-SVM is an enhanced SVM algorithm to tackle the problem of imbalanced training data set. Compared with traditional soft SVM, it ensures absolute classification correctness of the main class by restricting its slack variables [44]. The mathematical formation of OS-SVM for VSA is:

$$\begin{cases} \min_{\boldsymbol{w},\boldsymbol{b},\xi_{i}} \frac{1}{2} \|\boldsymbol{w}\|^{2} + C \sum_{i=1}^{n} \xi_{i} \\ \text{s.t. } y_{i}(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_{i} - \boldsymbol{b}) \geq 1 - \xi_{i} \quad \boldsymbol{x}_{i} \in \mathcal{S} \cup \mathcal{I} \\ \xi_{i} \geq 0 \quad \boldsymbol{x}_{i} \in \mathcal{S} \cup \mathcal{I} \\ \xi_{i} \leq 1 \quad \boldsymbol{x}_{i} \in \mathcal{S} \end{cases}$$
(17)

where  $\boldsymbol{w} \in \mathbb{R}^{2n}$  and  $\boldsymbol{b} \in \mathbb{R}$  are tuned parameters defining the boundary and supported vectors;  $\boldsymbol{x}_i \in \mathbb{R}^{2n}$  and  $y_i \in \mathbb{R}$  are a sampling vector and the corresponding label, respectively; Sand  $\mathcal{I}$  are the secure and insecure data domains, respectively; C is a parameter governing the affecting level of slack variables;  $\xi_i$  is the slack variable for  $\boldsymbol{x}_i$ ; and n is the total number of sampling vectors in the data set.

The enhancement of OS-SVM is achieved by the third restriction for slack variables of secure data. If the classification is correct for  $\mathbf{x}_i$ ,  $y_i(\mathbf{w}^T\mathbf{x}_i - \mathbf{b})$  will be greater than zero; while an incorrect result leads to  $y_i(\mathbf{w}^T\mathbf{x}_i - \mathbf{b}) < 0$ . Hence, the third restriction ensures  $y_i(\mathbf{w}^T\mathbf{x}_i - \mathbf{b}) \in [0, 1]$  for  $\mathbf{x}_i \in S$ , i. e., OS-SVM is guaranteed to obtain correct results for secure data no matter how much insecure data is misclassified. OS-SVM is particularly suitable for extremely imbalanced data problem where the data of one class is very rare. For the VSA of binary classification, very limited insecure data but a large number of secure data is used for OS-SVM. This is different from the operation condition of the proposed method, which is totally free for insecure data. However, the comparison profoundly illustrates the accuracy and advantages of the proposed method.

## 2) Cost-sensitive Decision Tree

Decision tree is a traditional and widely-investigated machine learning method which iteratively selects the most representative feature. In this paper, we employ classification and regression trees (CART), one algorithm of decision trees, to compare with the proposed method. To enhance the performance when processing imbalanced data set, a costsensitive loss function is used, which assigns different weights for classes according to their occurrence frequency.

$$\begin{cases} w_s = \frac{n}{n_c n_s} \\ w_I = \frac{n}{n_c n_I} \end{cases}$$
(18)

where  $w_s$  and  $w_l$  are the weights for secure and insecure data, respectively; *n* is the total number of sampling vectors in the data set;  $n_s$  and  $n_l$  are the numbers of data points in these classes, respectively.  $n_c = 2$ , which denotes the number of classes, and these two weights satisfy  $1/w_s + 1/w_l = 1$ . Since secure data is much more than insecure data in amount,  $w_l$  is largely greater than  $w_s$ . The loss function is defined as:

$$L = \frac{1}{n} \left( w_S \sum_{\boldsymbol{x}_i \in S} f(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) + w_I \sum_{\boldsymbol{x}_i \in \mathcal{I}} f(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) \right)$$
(19)

where  $\hat{y}_i$  and  $y_i$  are the estimation and the true class label corresponding to  $x_i$ , respectively;  $f(\cdot)$  is the loss function, which is the well-known cross-entropy in this paper. CSDT also requires a small set of insecure data, thus it is not totally free for insecure data.

## 3) Cost-sensitive Random Forest

Random forest is an ensemble learning method that aggregates numerous decision trees assigned by different input data, and then uses voting to obtain the final averaged result. It significantly mitigates the overfitting problem of decision tree and improves the performance. In this paper, we combine the weighted loss function with random forest, and use it to compare to the proposed method. More information on these machine learning methods can be found in [43].

# 4) Comparison Results

For comparison, we use IEEE 57-bus system and the same preprocessing and data generation process. OS-SVM, CSDT, and CSRF require a small amount of insecure data, and more insecure data in training leads to more accurate classification. Thus, we use 15% insecure data, i.e.,  $w_s =$ 0.5882,  $w_1$  = 3.3333, to implement OS-SVM, CSDT and CS-RF, while the proposed method operates without any insecure data. The comparison results and data amount are listed in Table VI. According to Table VI, the proposed method has achieved better accuracy than other machine learning methods in imbalanced fashion. Even these machine learning methods are adjusted by one-sided restriction or cost-sensitive loss function, the lack of insecure data also incurs large loss of effectiveness. Therefore, the advantage of the proposed method that requires no insecure data is very profound, and particularly suitable for practical validation. Besides, the proposed method operates in an unsupervised way that no human effort is called to create labels. In contrast, these traditional machine learning methods mentioned above are supervised, which needs manual annotation to acquire the labels for training data set. Therefore, the proposed method not only gains high-accuracy result for this imbalanced data, but also enjoys easy operation condition that neither insecure data nor manual working is required.

TABLE VI Weighted *f*1 Indices of Numerous Unsupervised Methods Using Different Testing Systems

Method	Insecure data (%)	Precision	Recall rate	fl index
Proposed	0	0.9792	0.9782	0.9787
LSTM-AE	0	0.9270	0.9288	0.9279
OS-SVM	15	0.8499	0.9134	0.8805
CSDT	15	0.7631	0.7805	0.7717
CSRF	15	0.8267	0.8781	0.8516

## V. CONCLUSION

This paper presents a reconstruction residual based VSA method that only requires the data of secure state. This work utilizes the well-known LSTM to form a spatial-temporal autoencoder, which is purely trained by secure data. Hence, the autoencoder is prone to produce lower reconstruction loss for secure data, while insecure data will encounter higher loss rate. To enhance the classification accuracy, a feature moving strategy for the middle features extracted from the autoencoder is proposed to enable insecure features to reside in the secure feature domain. This feature moving strategy guides the features of insecure data to move towards secure features, thus the insecure data is difficult to be properly recovered, i.e., higher reconstruction loss rate is obtained. The final reconstruction loss from the decoder is used as a proper indicator to detect the insecure operation state of VSA. Our method is particularly suitable for real validation, since

no insecure data, which is very infrequent in practice, is required. In addition, it also has the priority of high accuracy and robustness for measurement noise. Further investigation could be considered to approximate directly VSM by only secure data. Some invariant features of operation states for VSA will be constructed, and insecure features will be adjusted to share the same changing trend with secure features. Thus, it is likely to estimate VSM by certain kind of principles of the extracted features from autoencoders.

## REFERENCES

- [1] Y. Qiu, H. Wu, Y. Zhou et al., "Global parametric polynomial approximation of static voltage stability region boundaries," *IEEE Transac*tions on Power Systems, vol. 32, no. 3, pp. 2362-2371, May 2017.
- [2] J. M. Lim and C. L. DeMarco, "SVD-based voltage stability assessment from phasor measurement unit data," *IEEE Transactions on Pow*er Systems, vol. 31, no. 4, pp. 2557-2565, Jul. 2016.
- [3] Z. Wang and J. Wang, "A practical distributed finite-time control scheme for power system transient stability," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3320-3331, Sept. 2020.
- [4] X. Shi, R. Qiu, Z. Ling *et al.*, "Spatio-temporal correlation analysis of online monitoring data for anomaly detection and location in distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 995-1006, Mar. 2020.
- [5] H. Yang, R. C. Qiu, L. Chu et al., "Improving power system state estimation based on matrix-level cleaning," *IEEE Transactions on Power* Systems, vol. 35, no. 5, pp. 3529-3540, Sept. 2020.
- [6] Y. Zhang, G. Pan, B. Chen *et al.*, "Short-term wind speed prediction model based on GA-ANN improved by VMD," *Renewable Energy*, vol. 156, pp. 1373-1388, Aug. 2020.
- [7] D. Q. Zhou, U. D. Annakkage, and A. D. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1566-1574, May 2010.
- [8] B. Leonardi and V. Ajjarapu, "Development of multilinear regression models for online voltage stability margin estimation," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 374-383, Jan. 2011.
- [9] S. Li, V. Ajjarapu, and M. Djukanovic, "Adaptive online monitoring of voltage stability margin via local regression," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 701-713, Jan. 2018.
- [10] H. Su and T. Liu, "Enhanced-online-random-forest model for static voltage stability assessment using wide area measurements," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6696-6704, Nov. 2018.
- [11] Y. Fan, S. Liu, L. Qin *et al.*, "A novel online estimation scheme for static voltage stability margin based on relationships exploration in a large data set," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1380-1393, May 2015.
- [12] D. N. Reshef, Y. A. Reshef, H. K. Finucane *et al.*, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518-1524, Dec. 2011.
- [13] Y. Zhang, Y. Xu, R. Zhang *et al.*, "A missing-data tolerant method for data-driven short-term voltage stability assessment of power systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5663-5674, Sept. 2019.
- [14] F. Yang, Z. Ling, M. Wei et al., "Real-time static voltage stability assessment in large-scale power systems based on spectrum estimation of phasor measurement unit data," *International Journal of Electrical Power & Energy Systems*, vol. 124, p. 106196, Jan. 2021.
- [15] Y. Li and Z. Yang, "Application of EOS-ELM with binary Jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092-23101, Oct. 2017.
- [16] V. Malbasa, C. Zheng, P. Chen *et al.*, "Voltage stability prediction using active machine learning," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3117-3124, Nov. 2017.
- [17] L. Zhu, C. Lu, Z. Y. Dong *et al.*, "Imbalance learning machine-based power system short-term voltage stability assessment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2533-2543, Sept. 2017.
- [18] R. C. Qiu and P. Antonik, *Smart Grid and Big Data: Theory and Practice*. Hoboken: Wiley, 2015.
- [19] O. Kochan, "The technique to prepare a training set for a neural network to model the error of a thermocouple leg," in *Proceedings of*

2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech, Aug. 2019, pp. 101-104.

- [20] H. Yang, K. Ding, R. C. Qiu *et al.*, "Remaining useful life prediction based on normalizing flow embedded sequence-to-sequence learning," *IEEE Transactions on Reliability*. doi: 10.1109/TR.2020.3010970
- [21] D. Gong, L. Liu, V. Le et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, Korea, Nov. 2019, pp. 1705-1714.
- [22] S. C. P. E. D. Dehaene and O. Frigo. (2020, Feb.). Iterative energybased projection on a normal data manifold for anomaly localization. [Online]. Available: https://arxiv.org/abs/2002.03734
- [23] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD Internation*al Conference on Knowledge Discovery and Data Mining, Halifax, Canada, Aug. 2017, pp. 665-674.
- [24] L. Wang, Q. Zhou, and S. Jin, "Physics-guided deep learning for power system state estimation," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 4, pp. 607-615, Jul. 2020.
- [25] S. Ryu, H. Choi, H. Lee et al., "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Trans*actions on Power Systems, vol. 35, no. 2, pp. 1048-1060, Mar. 2020.
- [26] J. Wang, D. Shi, Y. Li et al., "Distributed framework for detecting pmu data manipulation attacks with deep autoencoders," *IEEE Trans*actions on Smart Grid, vol. 10, no. 4, pp. 4401-4410, Jul. 2019.
- [27] H. Yang, R. C. Qiu, X. Shi et al., "Unsupervised feature learning for online voltage stability evaluation and monitoring based on variational autoencoder," *Electric Power Systems Research*, vol. 182, pp. 1-11, May 2020.
- [28] P. Kundur, N. J. Balu, and M. G. Lauby, *Power System Stability and Control*. New York: McGraw-Hill, 1994.
- [29] T. Van Cutsem and C. Vournas, Voltage Stability of Electric Power Systems. Berlin: Springer Science & Business Media, 2007.
- [30] A. Rangamani, A. Mukherjee, A. Arora *et al.* (2017, Oct.). Critical points of an autoencoder can provably recover sparsely used overcomplete dictionaries. [Online]. Available: https://arxiv.org/abs/1708.03735
- [31] D. P. Kingma and M. Welling. (2013, Dec.). Auto-encoding variational bayes. [Online]. Available: https://arxiv.org/abs/1312.6114
- [32] A. Makhzani, J. Shlens, N. Jaitly et al. (2015, Nov.). Adversarial autoencoders. [Online]. Available: https://arxiv.org/abs/1511.05644
- [33] L. Zhu, C. Lu, and Y. Sun, "Time series shapelet classification based online short-term voltage stability assessment," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1430-1439, Mar. 2016.
- [34] W. Kong, Z. Y. Dong, Y. Jia *et al.*, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions* on Smart Grid, vol. 10, no. 1, pp. 841-851, Jan. 2019.
- [35] A. Gensler, J. Henze, B. Sick et al., "Deep learning for solar power forecasting: an approach using autoencoder and lstm neural networks," in Proceedings of 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, Oct. 2016, pp. 2858-2865.
- [36] M. S. Azman, Y. Isbeih, M. S. El Moursi et al., "A unified online deep learning prediction model for small signal and transient stability," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4585-4598, Jun. 2020.
- [37] M. Kaselimi, N. Doulamis, A. Voulodimos et al., "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054-3067, Jul. 2020.
- [38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge: MIT Press, 2016.
- [39] D. P. Kingma and J. Ba. (2014, Dec.). Adam: a method for stochastic optimization. [Online]. Available: https://arxiv.org/abs/1412.6980
- [40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, May 2010, pp. 249-256.
- [41] C. Josz, S. Fliscounakis, J. Maeght *et al.* (2016, Mar.). AC power flow data in MATPOWER and QCQP format: itesla, RTE snapshots, and PEGASE. [Online]. Available: https://arxiv.org/abs/1603.01533
- [42] X. He, Q. Ai, R. C. Qiu *et al.*, "A big data architecture design for smart grids based on random matrix theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 674-686, Mar. 2017.
- [43] J. F. T. Hastie and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer, 2009.
- [44] M. Wytock, S. Salapaka, and M. Salapaka, "Preventing cascading failures in microgrids with one-sided support vector machines," in *Pro-*100 (1997) 100 (19

ceedings of 53rd IEEE Conference on Decision and Control, Los Angeles, USA, Dec. 2014, pp. 3252-3258.

**Haosen Yang** received the B.S. degree from South China University of Technology, Guangzhou, China, in 2017, and M.S. degree in Shanghai Jiaotong University, Shanghai, China, in 2019. He is pursuing the Ph.D. degree in the School of Electronics and Electrical Engineering, Shanghai Jiaotong University. His research interests include voltage stability and state estimation of power grid, machine learning and data science.

**Robert C. Qiu** received the Ph.D. degree in electrical engineering from New York University, New York, USA. He currently serves as a Professor in the Research Center for Big Data Engineering and Technologies, State Energy Smart Grid R&D Center, Department of Electronics and Electrical Engineering, Shanghai Jiaotong University, Shanghai, China. He was with GTE Laboratories, Inc., Waltham and Bell Labs, Lucent Technologies, Southern Tier, New York, USA. He was the Founder Chief Executive Officer and the President of Wiscom Technologies, Inc., manufacturing and marketing wideband code division multiple access (WCDMA) chipsets, Murray Hill, New Jersey, USA. In 2008, he became a Professor at the Center for Manufacturing Research, Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, USA. He was named a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) in 2015 for his contributions to ultra-wideband wireless communications. His current research interests include wireless communication and networking, random matrix theory based theoretical analysis for deep learning, and smart grid.

**Houjie Tong** received the B.S. degree in North China Electric Power University, Beijing, China, in 2017. He is currently pursuing the M.S. degree in Shnaghai Jiaotong University, Shanghai, China. His research interests include graph neural networks and power system stability.