



Research  
Smart Grid and Energy Internet—Article

## Super Resolution Perception for Improving Data Completeness in Smart Grid State Estimation



Gaoqi Liang<sup>a,b</sup>, Guolong Liu<sup>a</sup>, Junhua Zhao<sup>a,c,\*</sup>, Yanli Liu<sup>d,\*</sup>, Jinjin Gu<sup>a</sup>, Guangzhong Sun<sup>b</sup>, Zhaoyang Dong<sup>e</sup>

<sup>a</sup> School of Science and Engineering, Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China

<sup>b</sup> School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

<sup>c</sup> Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518172, China

<sup>d</sup> School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

<sup>e</sup> School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia

### ARTICLE INFO

#### Article history:

Received 8 May 2019

Revised 8 October 2019

Accepted 19 February 2020

Available online 22 June 2020

#### Keywords:

State estimation

Low-frequency data

High-frequency data

Super resolution perception

Data completeness

### ABSTRACT

The smart grid is an evolving critical infrastructure, which combines renewable energy and the most advanced information and communication technologies to provide more economic and secure power supply services. To cope with the intermittency of ever-increasing renewable energy and ensure the security of the smart grid, state estimation, which serves as a basic tool for understanding the true states of a smart grid, should be performed with high frequency. More complete system state data are needed to support high-frequency state estimation. The data completeness problem for smart grid state estimation is therefore studied in this paper. The problem of improving data completeness by recovering high-frequency data from low-frequency data is formulated as a super resolution perception (SRP) problem in this paper. A novel machine-learning-based SRP approach is thereafter proposed. The proposed method, namely the Super Resolution Perception Net for State Estimation (SRPNSE), consists of three steps: feature extraction, information completion, and data reconstruction. Case studies have demonstrated the effectiveness and value of the proposed SRPNSE approach in recovering high-frequency data from low-frequency data for the state estimation.

© 2020 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### 1. Introduction

The smart grid is a critical infrastructure that continuously provides a secure and economic electricity supply to modern society. State estimation in the smart grid plays a vital role in system monitoring and control, which helps the system operator to perceive the system's operational states and make accurate control decisions [1]. At present, the smart grid development poses new requirements for state estimation. On the one hand, with the fast-increasing penetration of renewable energy sources and new power appliances—such as wind power, solar power, and electric vehicles—much greater uncertainties are being introduced into the smart grid [2]. In order to mitigate the adverse impacts of intermittent renewable energy, the system operator needs to perceive the system's operational states more frequently and shorten the dispatch interval, which requires the support of high-frequency state estimation. On the other hand, with the rapid improvement

of computational capability, big data technologies are being widely applied to discover hidden knowledge by analyzing system measurement data [3]. State estimation, which directly monitors voltage, current, and real and reactive power values, is a basic tool for the perception of system working states. Therefore, more frequent state estimation results are helpful for discovering more hidden knowledge, which is beneficial for improved system security and efficiency.

Performing high-frequency state estimation poses new practical challenges and difficulties. First, the state estimation monitors system states based on thousands of meters deployed on nodes, generators, transmission lines, and so forth. In the existing supervisory control and data acquisition (SCADA) system, the majority of meters in a smart grid are traditional sensors that were deployed years ago. Traditional meters collect measurements with a relatively low frequency, such as every few minutes [4,5]. Second, some high-frequency sampling meters, such as phasor measurement units (PMUs), can gather measurement data at a much higher frequency than traditional meters [6–8]. However, a single measurement is insufficient for state estimation; at any specific time

\* Corresponding authors.

E-mail addresses: [zhaojunhua@cuhk.edu.cn](mailto:zhaojunhua@cuhk.edu.cn) (J. Zhao), [yanliliu@tju.edu.cn](mailto:yanliliu@tju.edu.cn) (Y. Liu).

point, the input of the state estimation must be a vector of measurements. PMUs are extremely expensive; it is financially impractical to replace all traditional meters with PMUs [7]. Third, even if all traditional meters can be replaced by PMUs, the capability of system state perception is very restricted by the capacity of communication channels. The high-frequency sampling data collected by PMUs cannot be transferred completely to the control center, but are usually stored at remote locations temporarily [5]. Therefore, PMUs' high-frequency data cannot be truly utilized. In summary, due to technical restrictions, performing high-frequency state estimation is still a difficult task.

Furthermore, both the data collected by traditional meters and those collected by PMUs may be lost or manipulated due to communication faults or cyber-attacks [9], such as false data injection attacks (FDIAs) [10,11], cyber topology attacks [12–14], and cyber-physical attacks [15,16]. Real-world cyber-attack events, such as the Ukraine blackout in December 2015 [17] and the Venezuelan power outage in March 2019 [18], indicate that cyber-attacks can result in serious consequences. In the literature, many studies have been conducted to detect abnormal data caused by cyber attackers. For example, Ashok et al. [19] proposed the detection of anomalies by describing a statistical characterization of the variation between SCADA-based state estimates and predicted system states based on load forecast information, generation schedules, and the synchro-phasor data. Esmalifalak et al. [20] utilized a distributed support vector machine (SVM)-based method to distinguish between attacked data and normal measurement data. However, so far there is very limited research on effective approaches to recover these manipulated data. Missing and tampered data also introduce significant challenges to accurate state estimation.

To tackle the challenges discussed above, novel methods are needed to support the high-frequency perception of system operational states based on the existing metering infrastructure. In this paper, we consider this problem as a data completeness improvement problem. We consider the original measurements received in the control center as incomplete data (i.e., low-frequency data), and the data that the system operator is expected to use to make high-frequency decisions as complete data (i.e., high-frequency data). Therefore, this problem is equivalent to the question of how to recover high-frequency data from low-frequency data in order to achieve data completeness improvement. Approaches may vary according to different research fields and different data quality attributes; it is of utmost importance to explore appropriate approaches to improve data completeness for smart grid state estimation.

In this paper, we achieve this goal by applying super resolution (SR) technology. SR is a technology that can recover high-resolution data from low-resolution data from both temporal and spatial perspectives [21,22]. Currently, the most effective methods to obtain high-resolution data from low-resolution data are mainly based on interpolation, reconstruction, and machine learning, respectively [23]. The machine-learning-based SR, which attempts to obtain *a priori* mapping between low-resolution and high-resolution image blocks by given samples, has become a hot topic in recent years due to its good performance [24]. For example, Dong et al. [25] proposed a novel deep learning approach—namely, a super-resolution convolutional neural network (SRCNN)—to learn the end-to-end mapping between low-resolution and high-resolution images; this approach shows superior performance in restoring high-resolution images. Wang et al. [26] proposed an enhanced super-resolution generative adversarial network (ESR-GAN) to recover images, which achieves consistently better perceptual quality than other SR methods.

The motivations for this paper are obvious. High-frequency perception of the system's operation status is of great importance for

the development of the smart grid. However, traditional meters' low sampling frequency, PMUs' high investment cost, the capacity limitation on communication channels, and the abnormal statuses caused by communication mistakes or cyber-attacks present obstacles and challenges in practical situations. The purpose of this paper, therefore, is to develop an super resolution perception (SRP) approach to improve the data completeness for smart grid state estimation. This paper makes the following two key contributions:

(1) We are among the first to study the data completeness problem for smart grid state estimation in this paper.

(2) We are among the first to propose an effective SRP approach to recover high-frequency data from low-frequency data for state estimation. This paper has proved the effectiveness and value of the proposed Super Resolution Perception Net for State Estimation (SRPNSE) approach.

This paper is organized as follows. Section 2 provides some background information on smart grid state estimation and its data completeness problem. Section 3 describes the SRP problem and presents the network structure and solving framework of a novel deep learning approach: SRPNSE. Section 4 demonstrates the effectiveness of the proposed approach by simulations. Finally, Section 5 provides conclusions and future work discussions.

## 2. The data completeness problem for state estimation

In this section, we give a brief introduction to smart grid state estimation and its data completeness problem.

### 2.1. Smart grid state estimation

In the smart grid, the key for the perception of system operational states is to obtain system measurements—that is, the vector of the steady-state voltage (magnitude and angle) at each bus of the network. Once the voltage information is grasped, all other system state variables can be readily calculated using power flow equations [4,5]. However, not all nodes' voltage magnitudes and phase angles can be easily telemetered. Other information, such as the real and reactive power flows of some transmission lines, or some real and reactive power injections, need to be monitored so as to satisfy different control purposes (e.g., providing alerts for emergency situations). In addition, not all telemetered data are reliable, due to measurement errors caused by disturbances or cyber-attacks. State estimation is a tool to estimate system state variables from all available system measurements. Therefore, state estimation in modern power systems plays a vital role in the online monitoring, analysis, and control of smart grids.

Usually, state estimation is a module embedded in the energy management system (EMS) of smart grids. In addition to necessary communication networks, the overall state-estimation-related modules contain three main components: sensors, a state estimator, and a bad data detector. Sensors measure system states, such as bus voltage magnitude, real and reactive power injections, real and reactive power flow, and so forth, at a certain sampling frequency [1]. The state estimator utilizes all collected data to estimate system state variables in order to obtain a snapshot of the power system in the steady state. The bad data detector thereafter detects and eliminates obvious errors in measurements.

The state estimation process can be considered as a generalized power flow calculation. As shown in the following model,  $\mathbf{z}$  represents the vector of measurement data with size  $[m, 1]$ ;  $\mathbf{x}$  represents the vector of system state variables with size  $[n, 1]$ , and  $m > n$ ;  $h(\mathbf{x})$  represents the functional relationship between measurement values and system state variables; and  $\mathbf{e}$  represents the vector of noises with size  $[m, 1]$ .

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{e} \quad (1)$$

The goal is to calculate the  $\hat{\mathbf{x}}$  vector from the  $\mathbf{z}$  vector, making the estimated  $\hat{\mathbf{x}}$  as close as possible to the actual  $\mathbf{x}$  based on certain estimate criteria. A widely used method for estimating  $\hat{\mathbf{x}}$  is the maximum likelihood (ML) estimation:

$$p(\mathbf{z}, \hat{\mathbf{x}}) = \max[p(\mathbf{z}, \mathbf{x})] \quad (2)$$

where  $p(\mathbf{z})$  is the probability distribution density function of  $\mathbf{z}$ .

Based on different distribution hypotheses of  $\mathbf{z}$ , there are different suitable estimators, such as the weighted least squares (WLS), weighted minimum absolute value (WLAV), least median of squares (LMS), least trimmed squares (LTS), and non-quadratic estimators [5]. If the measurement noises follow the normal distribution, WLS will be an optimal, unbiased, and consistent estimator. The goal is to find the extremum of the objective function  $J(\mathbf{x})$ , shown as follows:

$$J(\mathbf{x}) = [\mathbf{z} - h(\mathbf{x})]^T R^{-1} [\mathbf{z} - h(\mathbf{x})] \quad (3)$$

where  $R$  is the diagonal matrix of the measurement error variances.

## 2.2. Data completeness for state estimation

Data quality is a critical problem for solving any industrial problems. Without high-quality data, the performance of both analytical and data-driven models will be seriously compromised. When based on poor-quality data, model outcomes will encounter unpredictable deviations that can cause substantial economic losses and security risks. Most of the existing data quality studies focus on database systems [27]. Data completeness is an important attribute for assessing the quality of a dataset [27]. Completeness is usually defined as whether there are any gaps in the data from what was actually collected and what was expected to be collected—that is, whether there are missing data, damaged data, or manipulated data [28].

In this paper, the low-frequency measurements actually received by the control center are considered as incomplete. The control center perceives a big picture of the system state by processing the measurement vector  $\mathbf{z}$ . Let the discrete sequence  $z_k(\cdot)$  represent the value of the  $k$ th row of vector  $\mathbf{z}$ ; that is,  $z_k(\cdot)$  is the measurement value of the  $k$ th meter. The corresponding mathematical expression can be represented by a time series, as shown in Eq. (4).

$$z_k(t) = \begin{cases} f_k(t), & t \in \{t_1, t_2, \dots\} \\ 0, & \text{others} \end{cases} \quad (4)$$

where  $f_k(t)$  represents the function of the measurement of the  $k$ th meter at time  $t$ .

Due to the low sampling frequency of traditional meters, the capacity limitation on communication channels, and communication mistakes or even cyber-attacks, some part of  $z_k(t)$  will be missing. Similarly, let us define another time series  $\tilde{z}_k(t)$ , in which its time labels  $t$  differ from  $z_k(t)$ , as shown in Eq. (5).

$$\tilde{z}_k(t) = \begin{cases} g_k(t), & t \in \{\tilde{t}_1, \tilde{t}_2, \dots\} \\ 0, & \text{others} \end{cases} \quad (5)$$

where  $g_k(t)$  represents the function of the measurement of the  $k$ th meter at time  $t$ .

The high-frequency time series are data that are expected to be collected, and are considered complete. The aggregated time series represented by  $Z_k(t)$ , therefore, is  $Z_k(t) = z_k(t) + \tilde{z}_k(t)$ , as shown in Eq. (6).

$$Z_k(t) = \begin{cases} f_k(t) + g_k(t), & t \in \{t_1, \tilde{t}_1, t_2, \tilde{t}_2, \dots\}, t_i \neq \tilde{t}_i \\ 0, & \text{others} \end{cases} \quad (6)$$

For example, the time series  $z_k(t)$  and  $\tilde{z}_k(t)$  are shown in Figs. 1(a) and (b), respectively. The aggregated series is then shown in Fig. 1(c).

Data completeness for state estimation in this paper mainly refers to the completeness of the measurement vector  $\mathbf{z}$  in the temporal dimension. The data completeness improvement is to generate more data based on available information so that it is as close as possible to the actual measurements—that is, to recover the missing time series  $\tilde{z}_k(t)$ .

## 3. SRP model and the solving method

In this section, the SRP problem is first proposed. Second, the problem of data completeness improvement for state estimation is formulated. Third, due to the powerful feature extraction capability of deep neural networks, a deep learning approach—namely, SRPNSE—is proposed. Finally, the optimization algorithms for estimating the model parameters of SRPNSE are introduced.

### 3.1. SRP modeling

Regardless of the size of the system, the input of state estimation is a vector of the measurements collected by many meters at a specific time point. Given vector  $\mathbf{z}$ , the data completeness improvement problem aims to recover each meter's missing data. We can solve this problem one meter after another. Recovering the missing data of each specific meter is equivalent to a SRP problem. The SRP problem can be expressed as follows:

$$\mathbf{L}_\beta = \downarrow_\beta \mathbf{H} + \mathbf{e} \quad (7)$$

where  $\mathbf{L}_\beta$  represents the low-frequency data actually collected by meters;  $\mathbf{H}$  represents the original high-frequency data;  $\mathbf{e}$  represents the vector of noises, where the noises here are caused by the meters; and  $\downarrow_\beta$  represents the down-sampling function, in which  $\beta$  is the down-sampling factor. For example, suppose  $\mathbf{H}$  is a measurement vector sampled at 60 Hz; if  $\beta$  is equal to 10, then  $\mathbf{L}_\beta$  will be a measurement vector based on  $\mathbf{H}$  with a sampling rate of 6 Hz.

In this paper, the SRP problem is formulated as a maximum a posteriori (MAP) estimation problem. Based on the MAP estimation, the goal is to estimate an  $\mathbf{H}_\beta$  to maximize the posterior probability, shown as follows:

$$p(\mathbf{L}_\beta, \hat{\mathbf{H}}_\beta) = \max[p(\mathbf{L}_\beta, \downarrow_\beta \mathbf{H})p(\downarrow_\beta \mathbf{H})] \quad (8)$$

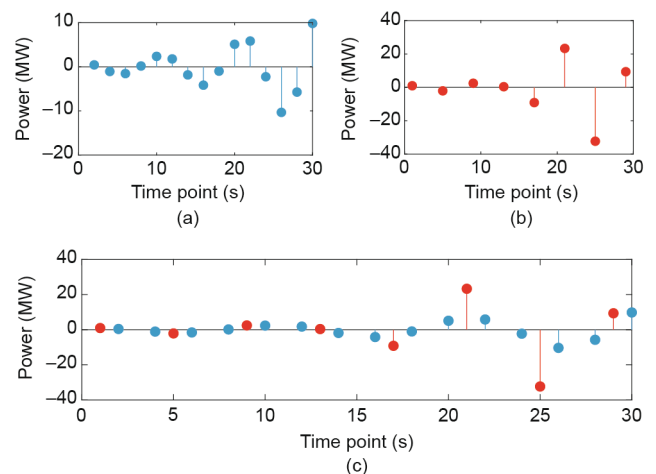


Fig. 1. The relationship of discrete sequences for meter  $k$ . Time series (a)  $z_k(t)$  and (b)  $\tilde{z}_k(t)$ ; (c) aggregated series  $Z_k(t)$ .

where  $p(L_\beta, \downarrow_\beta H)$  is the likelihood function and  $p(\downarrow_\beta H)$  is the prior probability of  $\downarrow_\beta H$ . In this situation, the prior probability mainly relates to the original sampling error caused by meters. Since the state estimation model assumes that the measurement noises follow the normal distribution, in this paper, we assume that the prior probability follows Gaussian distribution as well.

3.2. Problem formulation

The problem of improving data completeness for state estimation is equivalent to the problem of generating high-frequency data from low-frequency data. The high-frequency data is considered as complete data because it can recover the information lost in the incomplete data. The SRP problem formulation is shown in Fig. 2. Given a set of original data  $D$ , two down-sampling data sets  $L_\alpha$  and  $H_\beta$  are generated with down-sampling factors  $\alpha$  and  $\beta$ . The objective of SRP is to take the lower frequency data  $L_\alpha$  as the input and generate a set of estimated higher frequency data  $\hat{H}_\beta$  that is as close to the real down-sampling data  $H_\beta$  as possible.  $\ell_2$ -norm can be used to measure the difference between  $H_\beta$  and  $\hat{H}_\beta$ .

$$\ell_2 = \|\hat{H}_\beta - H_\beta\|_2 \tag{9}$$

3.3. SRPNSE framework

The network structure of the proposed SRPNSE method is shown in Fig. 3. The SRPNSE network directly uses the low-frequency data as the input, and then performs information enhancement to output the estimated high-frequency data. SRPNSE implements data quality improvement through the following three steps: feature extraction, information completion, and data reconstruction.

In the feature extraction stage, three one-dimensional (1D) convolutional layers [29] are used to extract features from low-frequency historical data. After obtaining the abstract features, the second part—the information completion stage—will supply higher resolution of features, based on the knowledge learned from

the relationship between the low-frequency data and the high-frequency data. The information completion stage consists of several SRPNSE blocks that are implemented by a residual structure. The residual structure consists of a big global residual connection and a number of local residual blocks. The global residual connection forces the network to learn the missing information rather than form the signal itself, and the local residual blocks provide the possibility to train deeper networks [29]. For better performance, this research used a total of 22 local residual blocks in the information completion stage. Fig. 4. shows the structure of the residual block used in this paper, where  $g$  represents the output of the previous layer; the rectified linear unit (ReLU) function [29] is used as the activation function; and identity represents the identity mapping. Then, the higher resolution features that contain more details of the system patterns are used to reconstruct the targeted high-frequency data in the third part—the data reconstruction stage—which is implemented by three 1D convolutional layers. In this part, the feature vectors outputted by the information completion are integrated into  $\beta$  sub-sequences with length  $l$ . The  $\beta$  sub-sequences are then rearranged into the reconstructed high-frequency sequence with length  $\beta \times l$ .

When training the proposed SRPNSE network, the mean squared error (MSE) is chosen as the loss function, which is shown as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{H}_{i\beta} - H_{i\beta})^2 \tag{10}$$

where  $\hat{H}_{i\beta}$  and  $H_{i\beta}$  are the  $i$ th data of  $\hat{H}_\beta$  and  $H_\beta$ , respectively.  $N$  is the size of the vector.

In this paper, we consider both the mean absolute percentage error (MAPE) [30] and the signal-to-noise ratio (SNR) [31] as evaluation metrics. The MAPE represents the degree of average absolute error compared with the actual value. A higher MAPE value means a larger difference between the actual value and the test one. The MAPE can be calculated as follows:

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{\hat{H}_{i\beta} - H_{i\beta}}{H_{i\beta}} \right| \tag{11}$$

In the signal processing field, the SNR represents the ratio of the average power of the signal to the average power of the noise. A higher SNR value indicates a smaller noise that the test value contains. The SNR can be calculated as follows:

$$SNR = \frac{\sum_{i=1}^N \hat{H}_{i\beta}^2}{\sum_{i=1}^N (\hat{H}_{i\beta} - H_{i\beta})^2} \tag{12}$$

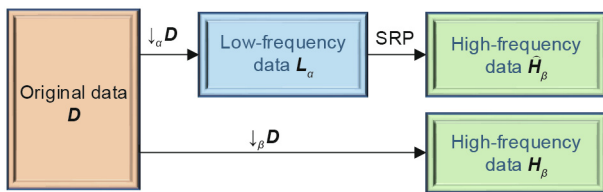


Fig. 2. Diagram of the SRP problem.

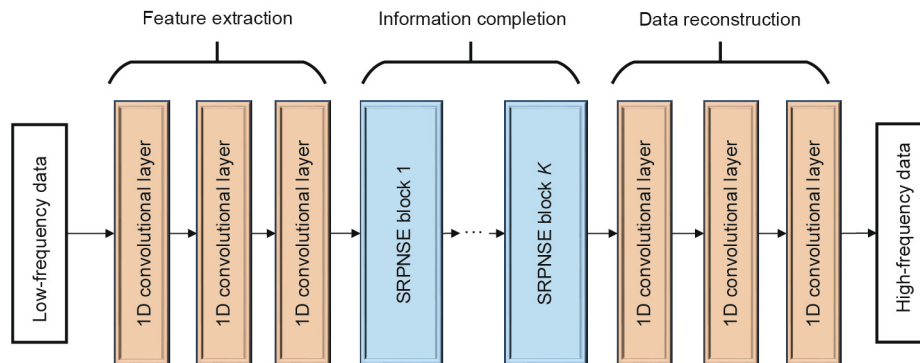


Fig. 3. Network structure of the proposed SRPNSE framework.

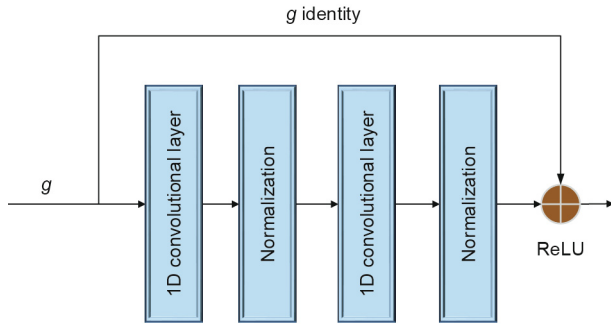


Fig. 4. Network structure of the SRPNSE block  $i$ , where  $i = [1, 2, \dots, K]$ .

### 3.4. Optimization method

In this paper, a novel neural network, SRPNSE, is proposed to solve the SRP problem. As a deep neural network, it consists of multiple residual blocks and has strong expressive power. That is, given a specific nonlinear function, the neural network can asymptotically approximate this function by appropriately adjusting its parameters. However, due to the existence of activation functions and multiple hidden layers in the network, the function underlying SRPNSE is highly nonlinear and nonconvex. The nonlinearity and nonconvexity of SRPNSE makes its parameter optimization problem extremely difficult. In this paper, we investigate efficient optimization algorithms for estimating the parameters of SRPNSE.

In the existing literature, gradient-based algorithms (as shown in **Algorithm 1**) remain the mainstream methods for the parameter estimation of deep neural networks. Typical examples are the batch gradient descent (BGD), stochastic gradient descent (SGD), and mini-batch gradient descent (Mini-BGD) algorithms [32]. In these algorithms, parameter updating is performed according to the following formulas:

$$W := W - \alpha_1 dW \tag{13}$$

$$b := b - \alpha_1 db \tag{14}$$

where  $W$  represents the weights;  $b$  is the bias;  $\alpha_1$  is the learning rate; and  $dW$  and  $db$  are the partial derivatives of cost function with respect to variables  $W$  and  $b$ , respectively.

---

#### Algorithm 1. The pseudo-code for gradient-based algorithms.

---

**Require:**  $f(\theta)$  Objective function with parameters  $\theta$

**Require:**  $\alpha_1$  Learning rate

**Require:**  $\theta_0 = 0, t = 0$  Initial parameters

**While** {not converged} **do:**

1. Calculate the gradient of the objective function with respect to the current parameter:

$$g_t = \nabla f(\theta_t)$$

2. Calculate the descending gradient at the current moment:

$$\eta_t = -\alpha_1 g_t$$

3. Update based on the descending gradient:

$$\theta_{t+1} = \theta_t + \eta_t$$

$$t = t + 1$$

**End**

**Return**  $\theta_{t+1}$

---

The main difference between these gradient-based algorithms is that when updating parameters in one iteration, BGD trains the network based on all batches of training data and SGD stochastically selects only one batch for training, while Mini-BGD selects only a portion of the batches. The biggest drawback of BGD is that the convergence speed is slow, especially when the number of batches is large, because it solves the gradient by calculating all batches. On the other hand, since SGD's and Mini-BGD's updating of the gradient direction is dependent on one or only a few data batches, their convergence trajectories are very unstable, resulting in continuous oscillation and local optima. Considering the drawbacks of traditional methods, based on Eqs. (13) and (14), many other methods such as Momentum [32], the adaptive gradient algorithm (Adagrad) [32], the root mean square prop (RMSProp) [32,33], and the adaptive moment estimation (ADAM) [32,34] have been proposed for improving the training process. In this paper, we will investigate the effectiveness of two new algorithms—RMSProp and ADAM—for estimating the parameters of SRPNSE.

#### 3.4.1. Root mean square prop

One main disadvantage of the dominant gradient descent methods is that the learning rate  $\alpha_1$  is a fixed value. Choosing a proper learning rate can be difficult. If it is too small, the convergence speed will be very slow, whereas if it is too large, the loss function will oscillate or even deviate significantly from the minimum value. RMSProp (as shown in **Algorithm 2**) is a variant of the dominant gradient descent methods that overcomes this shortcoming. Compared with Eqs. (13) and (14), RMSProp achieves an excellent adaptation of the learning rate by adding a moving average of the squared gradient over adjacent mini-batches [34]. For each iteration, as shown in Eqs. (17) and (18), the given learning rate  $\alpha_1$  is dynamically adjusted by the root mean square. The root mean square is actually the root of the exponential moving averages of squared past gradients. According to Eqs. (15) and (16), RMSProp limits the reliance of the update to only the past few gradients [34]. The root mean square in RMSProp aims to balance the oscillation amplitude of different dimensions. When the parameter space is relatively flat, the partial derivation is small; then, the exponential moving average is small and the learning rate speeds up as a result. When the parameter space is relatively steep, the partial derivation is large; then, the exponential moving average is large and the learning rate slows down as a result.

---

#### Algorithm 2. The pseudo-code for the RMSProp algorithm.

---

**Require:**  $f(W, b)$  Objective function with parameters  $W, b$

**Require:**  $\alpha_1$  Learning rate

**Require:**  $\beta_2$  Exponentially decaying average of squared gradients

**Require:**  $W_0, b_0, S_0(dW), S_0(db)$  Initial parameters

For each epoch  $t$ , **while** {not converged} **do:**

1. Calculate the gradient of the objective function with respect to the current parameter:

$$dW, db = \nabla f(W, b)$$

2. Calculate biased the second moment estimate from historical gradients:

$$S_{dW} := \beta_2 S_{dW} + (1 - \beta_2)(dW)^2, S_{db} := \beta_2 S_{db} + (1 - \beta_2)(db)^2$$

3. Calculate the descending gradient at the current moment:

$$\eta_W = -\alpha_1 \frac{dW}{\sqrt{S_{dW} + \epsilon}}, \eta_b = -\alpha_1 \frac{db}{\sqrt{S_{db} + \epsilon}}$$

4. Update based on the descending gradient:

$$W := W + \eta_W, b := b + \eta_b$$

**End**

**Return**  $w, b$

---

$$S_{dW} := \beta_2 S_{dW} + (1 - \beta_2)(dW)^2 \quad (15)$$

$$S_{db} := \beta_2 S_{db} + (1 - \beta_2)(db)^2 \quad (16)$$

$$W := W - \alpha_1 \frac{dW}{\sqrt{S_{dW} + \varepsilon}} \quad (17)$$

$$b := b - \alpha_1 \frac{db}{\sqrt{S_{db} + \varepsilon}} \quad (18)$$

where  $(dW)^2$  and  $(db)^2$  are the square of the gradient;  $\beta_2$  represents the exponential decay rate, which is usually set as 0.9 or 0.999;  $S_{dW}$  and  $S_{db}$  represent the exponential moving averages of squared past gradients; and  $\varepsilon$  is a very small number, say  $10^{-8}$ , to prevent the denominator from being 0.

### 3.4.2. Adaptive moment estimation

The other disadvantage of the dominant gradient descent methods is that the current gradient is the only factor to determine the descent direction. Once the current gradient is pointing in the opposite direction of the previous gradient, the loss function will oscillate or even deviate from the minimum value. Momentum [32] is a variant of the dominant gradient descent methods that overcomes this shortcoming. Compared with Eqs. (13) and (14), Momentum achieves the stability for faster learning by adding the accumulation of the exponential moving average of past gradients and then moving in that direction [32].

The ADAM algorithm (as shown in **Algorithm 3**) combines the ideas of both the Momentum and RMSProp algorithms. As shown in Eqs. (19) and (20), the exponentially decaying average of the gradients is calculated, which is the Momentum and is called the “first-order moment estimation” in ADAM; as shown in Eqs. (21) and (22), the exponentially decaying average of the squared gradients is calculated, which is the RMSProp and is called the “second-order moment estimation” in ADAM. In addition, as shown in Eqs. (23) and (24), ADAM computes a bias-corrected first-order moment estimate and second-order moment estimate to offset the deviation caused by the initialized zero vectors. As shown in

Eqs. (25) and (26), ADAM not only updates the descent direction by an exponentially decaying average of gradients, but also divides the learning rate by an exponentially decaying average of squared gradients. As a result, faster convergence and reduced oscillation are gained [34].

$$v_{dW} := \beta_1 v_{dW} + (1 - \beta_1)dW \quad (19)$$

$$v_{db} := \beta_1 v_{db} + (1 - \beta_1)db \quad (20)$$

$$S_{dW} := \beta_2 S_{dW} + (1 - \beta_2)(dW)^2 \quad (21)$$

$$S_{db} := \beta_2 S_{db} + (1 - \beta_2)(db)^2 \quad (22)$$

$$v_{dW}^c := \frac{v_{dW}}{1 - \beta_1^t}, v_{db}^c := \frac{v_{db}}{1 - \beta_1^t} \quad (23)$$

$$S_{dW}^c := \frac{S_{dW}}{1 - \beta_2^t}, S_{db}^c := \frac{S_{db}}{1 - \beta_2^t} \quad (24)$$

$$W := W - \alpha_1 \frac{v_{dW}^c}{\sqrt{S_{dW}^c + \varepsilon}} \quad (25)$$

$$b := b - \alpha_1 \frac{v_{db}^c}{\sqrt{S_{db}^c + \varepsilon}} \quad (26)$$

where  $\beta_1$  represents the exponential decay rate for the Momentum, which is usually set as 0.9;  $v_{dW}^c$ ,  $v_{db}^c$ ,  $S_{dW}^c$ , and  $S_{db}^c$  are the bias-corrected estimates; and  $\beta_1^t$  and  $\beta_2^t$  are defined as  $\beta_1$  and  $\beta_2$  to the power of the current timestep  $t$ .

## 4. Case studies

In this section, we conduct case studies based on a 9-bus system [35]. As shown in Fig. 5, three meters are deployed on nodes 5, 7, and 9 to record the real power loads; three meters are deployed on nodes 1, 2, and 3 to measure the real and reactive power outputs of the generators; five meters are deployed on the “from-end” of branches 1–4, 5–6, 6–7, 8–2, and 9–4 (e.g., the “from-end” of branch 1–4 is bus 1) to measure the real and reactive power flows; and four meters are deployed on the “to-end” of branches 4–5, 3–6, 7–8 and 8–9 (e.g., the “to-end” of branch 4–5 is bus 5) to measure the real and reactive power flows.

In order to better simulate the state estimation scenarios, the input values of state estimation in this paper are assumed to be some of the results of optimal power flow (OPF) calculations based on measured loads. As shown in Fig. 5, the first three meters, providing three real powers of loads, are considered as the input values of OPF; the remaining 12 meters, providing overall 24 real and reactive powers, are the input values of the state estimation for the system; and the values of the 12 meters come from the results of OPF.

It is assumed that on each load node, 1 MW electricity is set to supply approximately 200 households. Each household contains 11 types of appliances, such as air conditioners, heaters, washing machines, microwaves, and so on; each appliance's waveform comes from the plug load appliance identification dataset (PLAID) [36]. The PLAID samples 11 different types of appliances at  $3 \times 10^4$  Hz, which is down-sampled to 100 Hz in this paper. In this paper, we simulate 900, 1000, and 1250 households, with 100 times magnification for the load on nodes 5, 7, and 9, respectively. The super resolution perception state estimation dataset (SRPSED) was designed for testing the proposed SRPNSE; it has been released and can be found at

---

**Algorithm 3.** The pseudo-code for the ADAM algorithm.

---

**Require:**  $f(W, b)$  Objective function with parameters  $W, b$

**Require:**  $\alpha_1$  Learning rate

**Require:**  $\beta_1, \beta_2$  Exponentially decaying average of squared gradients

**Require:**  $W_0, b_0, S_0(dW), S_0(db), v_0(dW), v_0(db)$  Initial parameters

For each epoch  $t$ , **while** {not converged} **do**:

1. Calculate the gradient of the objective function with respect to the current parameter:

$$dW, db = \nabla f(W, b)$$

2. Calculate biased first and second moment estimate from historical gradients:

$$v_{dW} := \beta_1 v_{dW} + (1 - \beta_1)dW, v_{db} := \beta_1 v_{db} + (1 - \beta_1)db$$

$$S_{dW} := \beta_2 S_{dW} + (1 - \beta_2)(dW)^2, S_{db} := \beta_2 S_{db} + (1 - \beta_2)(db)^2$$

3. Update the bias-corrected of first and second order moment estimate:

$$v_{dW}^c := \frac{v_{dW}}{1 - \beta_1^t}, v_{db}^c := \frac{v_{db}}{1 - \beta_1^t}, S_{dW}^c := \frac{S_{dW}}{1 - \beta_2^t}, S_{db}^c := \frac{S_{db}}{1 - \beta_2^t}$$

4. Update based on the descending gradient:

$$W := W - \alpha_1 \frac{v_{dW}^c}{\sqrt{S_{dW}^c + \varepsilon}}, b := b - \alpha_1 \frac{v_{db}^c}{\sqrt{S_{db}^c + \varepsilon}}$$

**End**

**Return**  $W, b$

---

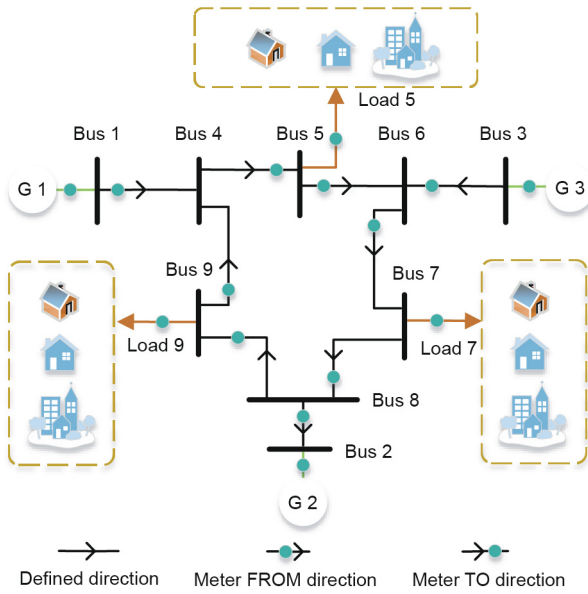


Fig. 5. Topology structure of the 9-bus system. G: generator.

<https://www.zhaojunhua.org/SRP/SRPSE/dataset/>. In generating this dataset, the user behavior of a normal office worker was adopted for each household; the user behavior has also been released with the dataset. The SRPSED contains a total of 60 d of high-frequency data at a frequency of 100 Hz, in which the first 45 d are used for training and validation, and the last 15 d are used for testing. Since the data completeness improvement problem for every single meter is independent in this paper, the proposed SRPNSE approach can be applied to larger systems in a similar manner.

As shown in Table 1, we conducted four case studies with a total of 16 different scenarios in this paper.  $f(L_x)$  and  $f(\hat{H}_\beta)$  represent the sampling rate in Hz, in which  $f(\hat{H}_\beta) = \eta f(L_x)$ , and  $\eta$  is the times. The down-sampling criterion is based on the interval values instead of average values. The case study programs are implemented in PyTorch 0.4.1 and executed on a GPU cluster with four GTX-1080Ti, a 16-core CPU, and 64 gigabytes RAM.

#### 4.1. Performance of SRPNSE in state estimation

Interpolation [37] is a popular way to fill vacancies and replace wrong data in many areas. In this paper, linear interpolation and cubic interpolation are applied as comparisons to the proposed SRPNSE approach. State estimation calculates the 9-bus system's state—that is, the voltage magnitude and angle—based on the given measurements. In this paper, we execute the state estimation based on the real down-sampled data, SRPNSE data, linear interpolation data, and cubic interpolation data, respectively. Four case

studies with a total of 16 scenarios were conducted. For each scenario, we calculated the MAPE and SNR of the voltage magnitude and angle.

##### 4.1.1. Performance evaluated with MAPE

The MAPE values for both voltage magnitude and angle for all scenarios under SRPNSE, linear interpolation, and cubic interpolation are shown in Appendix A Tables S1 and S2, respectively (also see Figs. 6 and 7). It should be noticed that each subfigure contains four scenarios. For example, Fig. 6(a) represents scenarios of recovering data from 1/60, 1/300, 1/600, and 1/900 Hz with  $\eta = 5$ , respectively.

##### 4.1.2. Performance evaluated with SNR

The SNR values for both voltage magnitude and angle for all scenarios under SRPNSE, linear interpolation, and cubic interpolation are shown in Appendix A Tables S3 and S4, respectively (also see Figs. 8 and 9).

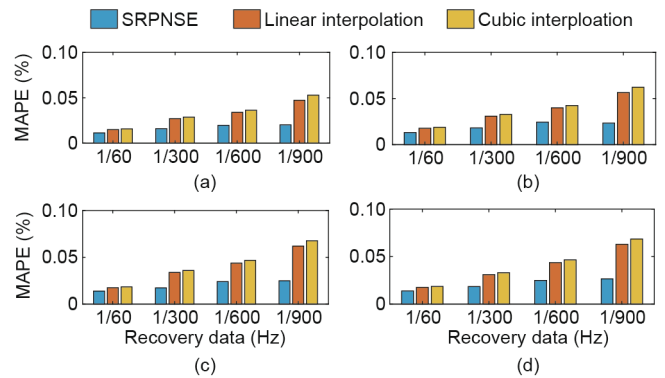


Fig. 6. MAPE comparisons among SRPNSE, linear interpolation, and cubic interpolation about the voltage magnitude. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

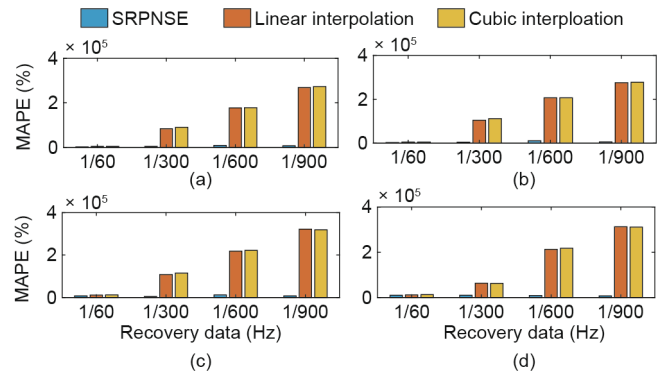


Fig. 7. MAPE comparisons among SRPNSE, linear interpolation, and cubic interpolation about the voltage angle. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

Table 1  
The down-sampling factors and SR factors used in this paper.

Case	$\alpha$	$f(L_x)$	$f(\hat{H}_\beta), \eta = 5$	$f(\hat{H}_\beta), \eta = 10$	$f(\hat{H}_\beta), \eta = 50$	$f(\hat{H}_\beta), \eta = 100$
1	6 000	1/60	1/12	1/6	5/6	5/3
2	30 000	1/300	1/60	1/30	1/6	1/3
3	60 000	1/600	1/120	1/60	1/12	1/6
4	90 000	1/900	1/180	1/90	1/18	1/9

4.2. Performances of the SRPNSE on load nodes

4.2.1. Performance evaluated with MAPE

The MAPE values in load nodes 5, 7, and 9 for all scenarios under SRPNSE, linear interpolation, and cubic interpolation are shown in Appendix A Tables S5, S6, and S7, respectively. Here, we take the scenarios on node 5 as a representative (Fig. 10).

4.2.2. Performance evaluated with SNR

The SNR values in load nodes 5, 7, and 9 for all scenarios under SRPNSE, linear interpolation, and cubic interpolation are shown in Appendix A Tables S8, S9, and S10, respectively. Here, we take the scenarios on node 5 as a representative (Fig. 11).

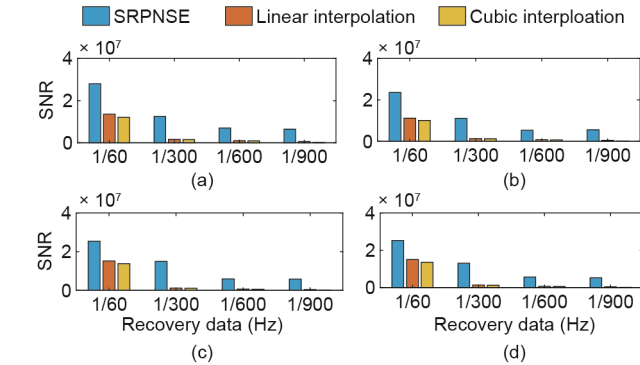


Fig. 8. SNR comparisons among SRPNSE, linear interpolation, and cubic interpolation about the voltage magnitude. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

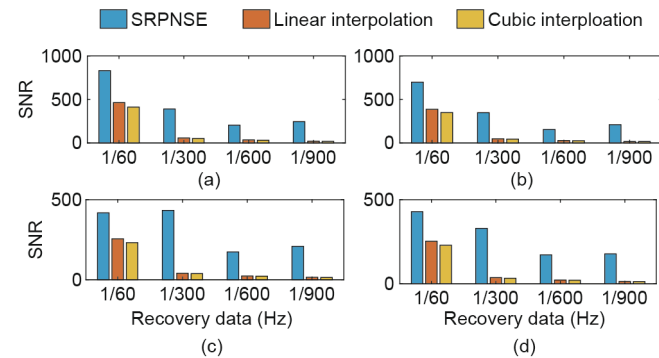


Fig. 9. SNR comparisons among SRPNSE, linear interpolation, and cubic interpolation about the voltage angle. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

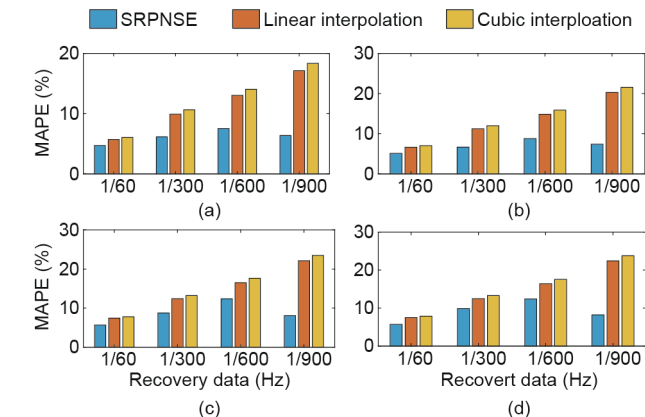


Fig. 10. MAPE comparisons among SRPNSE, linear interpolation, and cubic interpolation on load node 5. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

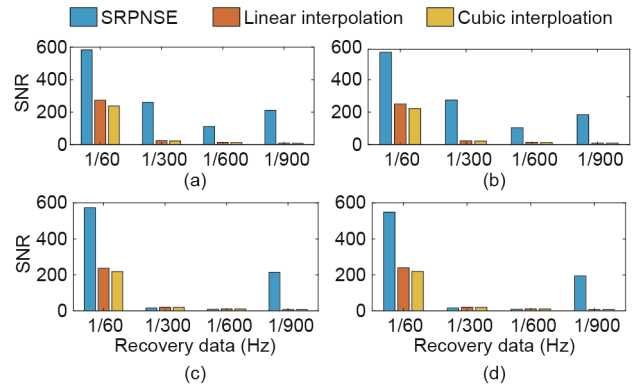


Fig. 11. SNR comparisons among SRPNSE, linear interpolation, and cubic interpolation on load node 5. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

4.3. Visualized comparison of state estimation

The 9-bus system has a total of three generators, nine nodes, and nine branches. Four case studies with the state estimation for a total of 16 scenarios were conducted; here, we randomly selected case 4, node 3, and branch 5–6 to show visualized comparisons of the state estimation results in more detail, as a representative. Specifically, we randomly chose a time period to show the fluctuation of voltage magnitude on node 3, voltage angle on node 3, power flow on branch 5–6, and generator output on node 3 with  $\eta = 5$ ,  $\eta = 10$ ,  $\eta = 50$ , and  $\eta = 100$ , respectively. In each figure, the true data, SRPNSE data, linear interpolation data, and cubic interpolation data are visualized for comparison.

4.3.1. Case 4 with  $\eta = 5$ : Voltage magnitude on node 3

In this scenario  $\alpha = 90\ 000$  and  $\eta = 5$ ; that is, the low-frequency data received in the control center is 1/900 Hz, and the goal is to restore 1/180 Hz data from the 1/900 Hz data. Based on the recovered data, state estimation is executed. Here, the voltage magnitude on node 3 is drawn. A comparison of the voltage magnitude after state estimation between the real down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation is shown in Fig. 12.

4.3.2. Case 4 with  $\eta = 10$ : Voltage angle on node 3

In this scenario,  $\alpha = 90\ 000$  and  $\eta = 10$ ; that is, the low-frequency data received in the control center is 1/900 Hz, and the goal is to restore 1/90 Hz data from the 1/900 Hz data. Based on the recovered data, state estimation is executed. Here, the

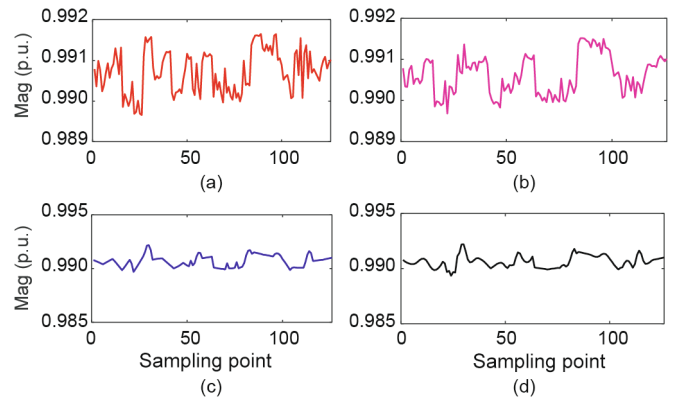


Fig. 12. Voltage magnitude (Mag) on node 3 after state estimation, recovering data from 1/900 to 1/180 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation. p.u.: per unit.



voltage angle on node 3 is drawn. A comparison of the voltage angle after state estimation between the real down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation is shown in Fig. 13.

4.3.3. Case 4 with  $\eta = 50$ : Power flow on branch 5–6

In this scenario,  $\alpha = 90\,000$  and  $\eta = 50$ ; that is, the low-frequency data received in the control center is 1/900 Hz, and the goal is to restore 1/18 Hz data from the 1/900 Hz data. Based on the recovered data, state estimation is executed. Here, the power flow on branch 5–6 is drawn. A comparison of the power flow on branch 5–6 after state estimation between the real down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation is shown in Fig. 14.

4.3.4. Case 4 with  $\eta = 100$ : Generator output on node 3

In this scenario,  $\alpha = 90\,000$  and  $\eta = 100$ ; that is, the low-frequency data received in the control center is 1/900 Hz, and the goal is to restore 1/9 Hz data from the 1/900 Hz data. Based on the recovered data, state estimation is executed. Here, the generator output on node 3 is drawn. A comparison of the generator output on node 3 after state estimation between the real down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation is shown in Fig. 15.

4.4. Visualized comparison of load nodes

Four case studies with a total of 16 scenarios were conducted. Here, we randomly selected case 1 on load node 5 and case 4 on load node 9 and discussed them in greater detail. In each figure,

the true data, SRPNSE data, linear interpolation data, and cubic interpolation data are visualized for comparison.

4.4.1. Load node 5: Data completeness improvement from 1/60 Hz

In this case,  $\alpha = 6000$  and  $\eta = 5, \eta = 10, \eta = 50, \text{ and } \eta = 100$ ; that is, the low-frequency data received in the control center is 1/60 Hz, and the goal is to restore 1/12, 1/6, 5/6, and 5/3 Hz data from the 1/60 Hz data, respectively. Here, we randomly chose a time period. Comparisons between the real 1/12, 1/6, 5/6, and 5/3 Hz down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation are shown in Figs. 16–19.

4.4.2. Load node 9: Data completeness improvement from 1/900 Hz

In this case,  $\alpha = 90\,000$  and  $\eta = 5, \eta = 10, \eta = 50, \text{ and } \eta = 100$ ; that is, the low-frequency data received in the control center is 1/900 Hz, and the goal is to restore 1/180, 1/90, 1/18, and 1/9 Hz data from the 1/900 Hz data, respectively. Here, we randomly chose a time period. Comparisons between the real 1/180, 1/90, 1/18, and 1/9 Hz down-sampled data and the estimated ones by SRPNSE, linear interpolation, and cubic interpolation are shown in Figs. 20–23.

4.5. Comparison of the SGD, RMSProp, and ADAM algorithms for solving the SRPNSE framework

Tables S11 and S12 in Appendix A provide comparisons of the MAPE and SNR values using the ADAM and RMSProp algorithms compared with the SGD algorithm for solving the proposed SRPNSE framework. Here, the MAPE values on load node 5 for three algorithms are shown as a representative (Fig. 24).

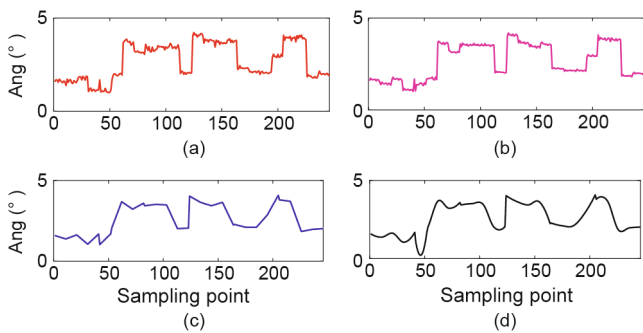


Fig. 13. Voltage angle (Ang) on node 3 after state estimation, recovering data from 1/900 to 1/90 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.

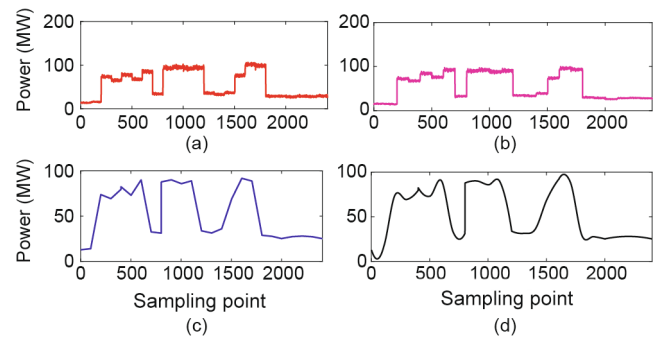


Fig. 15. Generator output on node 3 after state estimation, recovering data from 1/900 to 1/9 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.

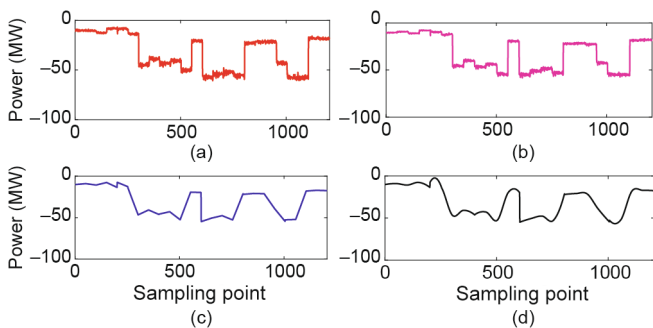


Fig. 14. Power flow on branch 5–6 after state estimation, recovering data from 1/900 to 1/18 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.

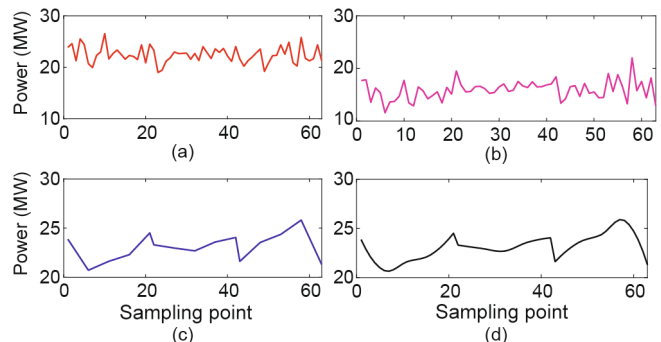
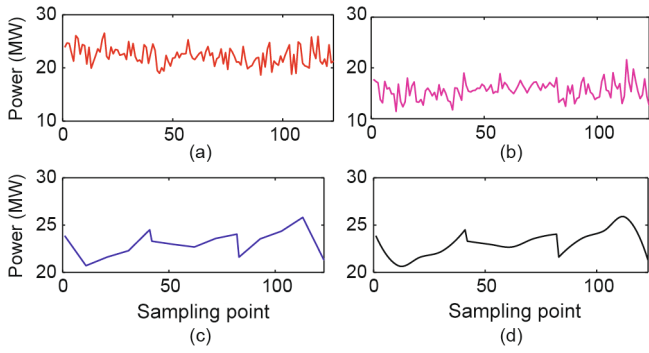
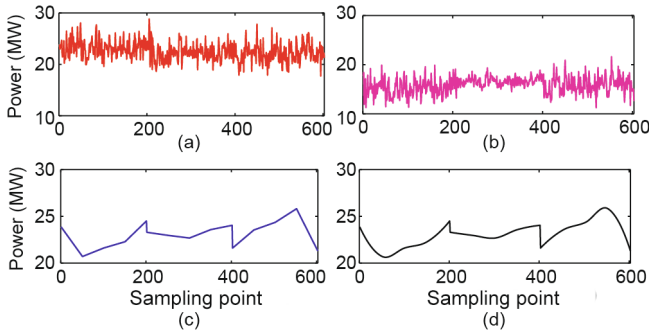


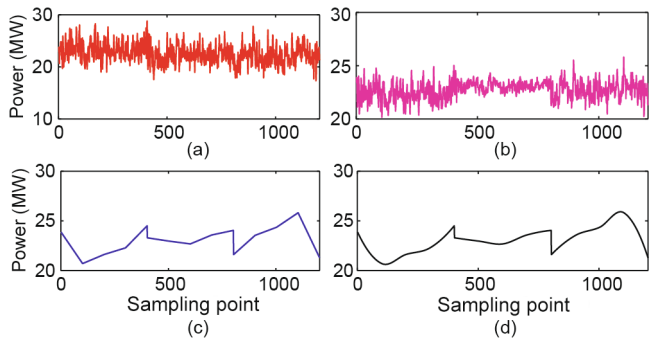
Fig. 16. Load measurements, recovering data from 1/60 to 1/12 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



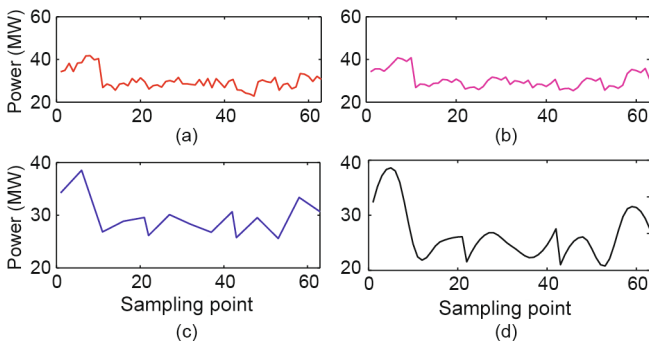
**Fig. 17.** Load measurements, recovering data from 1/60 to 1/6 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



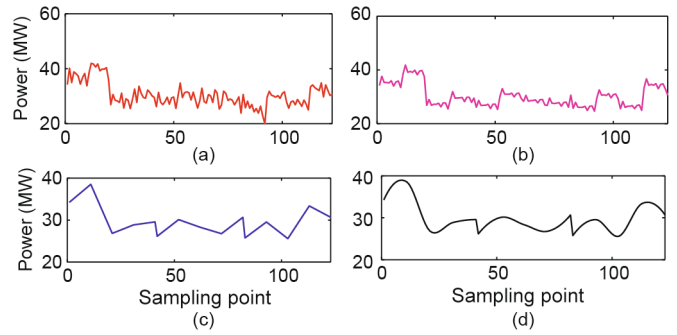
**Fig. 18.** Load measurements, recovering data from 1/60 to 5/6 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



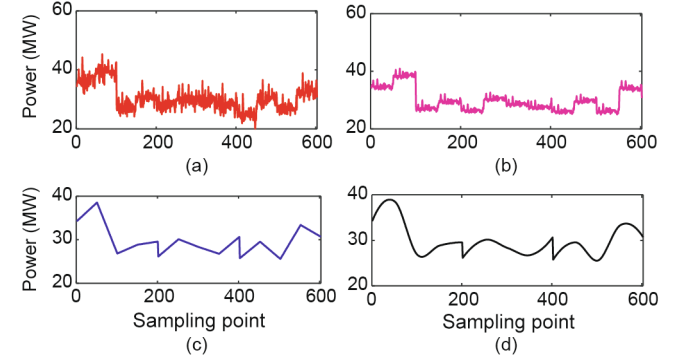
**Fig. 19.** Load measurements, recovering data from 1/60 to 5/3 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



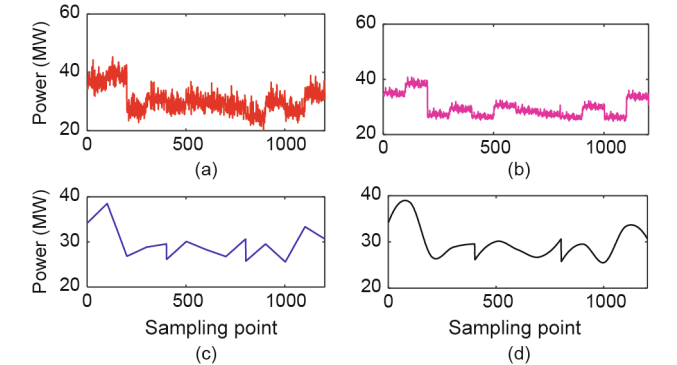
**Fig. 20.** Load measurements, recovering data from 1/900 to 1/180 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



**Fig. 21.** Load measurements, recovering data from 1/900 to 1/90 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



**Fig. 22.** Load measurements, recovering data from 1/900 to 1/18 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.



**Fig. 23.** Load measurements, recovering data from 1/900 to 1/9 Hz. (a) True data; (b) SRPNSE; (c) linear interpolation; (d) cubic interpolation.

We also compared the loss function in iterations using the RMSProp and ADAM algorithms compared with the SGD algorithm. Here, the MSE value for the scenario of case 4 with  $\eta = 100$  is selected as a representative (Fig. 25(a)). Fig. 25(b) provides magnified views of the first 80 iterations and the iterations from 300 to 380, respectively.

4.6. Result analysis

Tables S1–S4, Figs. 6–9, and Figs. 12–15 provide comparisons focused on the state estimation results. Tables S5–S10, Figs. 10 and 11, and Figs. 16–23 provide comparisons focused on the measurements of load nodes. It should be noted that the latter

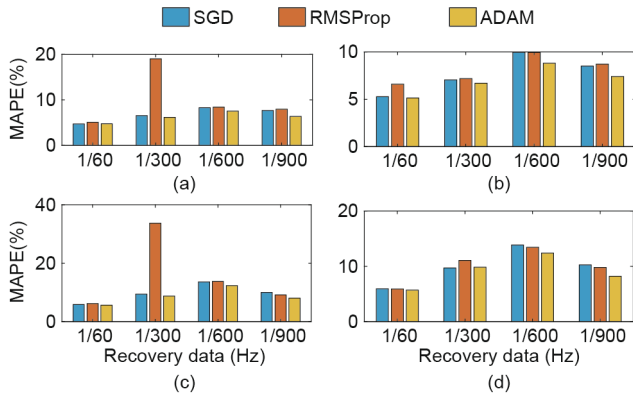


Fig. 24. MAPE comparisons using RMSProp and ADAM algorithm compared with SGD. (a)  $\eta = 5$ ; (b)  $\eta = 10$ ; (c)  $\eta = 50$ ; (d)  $\eta = 100$ .

comparisons are SR results of the meters' measurements, while the former comparisons illustrate the state estimation results, which are based on the SR results of the meters' measurements.

First, it is clear from Tables S1–S4 and Tables S5–S10 that the proposed SRPNSE significantly outperforms the linear and cubic interpolation methods. This indicates that the data supplemented by the SRPNSE approach can achieve a more accurate estimate of the actual situation, and thus helps to achieve a more accurate state estimation result. More importantly, the differences between using the SRPNSE, linear interpolation, and cubic interpolation methods on the state estimation are obvious. As shown in Tables S1–S4 and Figs. 6–9, the value differences are as high as one or two orders of magnitude. This indicates that the linear and cubic interpolations are weak in recovering lost information from relatively low-frequency data, while the proposed SRPNSE approach performs well.

Second, it is obvious from Tables S5–S7 that, no matter what the SR factor is, the MAPE values of the SRPNSE and interpolation methods keep increasing when the sampling frequency drops; and, more importantly, the lower the frequency is, the higher the MAPE difference between the SRPNSE and interpolation methods is. The reason for the small MAPE differences between the SRPNSE

and interpolation methods, as in case 1, is that relatively high-frequency data already contains enough information, which helps to improve the accuracy of the interpolations.

Third, by comparing the scenarios of different SR factors for a specific case, such as case 3 with  $\eta = 5$ ,  $\eta = 10$ ,  $\eta = 50$ , and  $\eta = 100$ , it is clear from Tables S5–S7 and Tables S8–S10 that a smaller SR factor will usually lead to better performance for both the SRPNSE and the interpolation methods.

Fourth, from Fig. 24 and Tables S11 and S12, it is clear that most MAPE values based on the ADAM algorithm are lower than those based on the RMSProp and SGD algorithms. As shown in Fig. 25(a), the SGD algorithm achieves a slow convergence speed and encounters continuous oscillation; as shown in Fig. 25(b), compared with RMSProp, the ADAM algorithm is slightly slow in convergence speed, but its stability is outstanding. The result is consistent with Section 3.4: The ADAM algorithm not only uses a dynamically adjusted learning rate to speed up convergence, but also uses an accumulated momentum to stay stable. Therefore, the ADAM algorithm performs better than the RMSProp and SGD algorithms in solving the proposed SRPNSE framework.

### 5. Conclusions and future works

In this article, we proposed a novel machine-learning-based SRP approach to improve data completeness for smart grid state estimation. The case studies demonstrated the effectiveness and value of the proposed approach.

Concerning the applicability of the SRPSNE approach in a larger system, please note that the SRPNSE is an algorithm that recovers high-frequency data for a single meter. In other words, when solving the SRP problem, the SRPNSE approach is applied to recover one meter after another, without using any information from neighboring meters. Therefore, when the SRPNSE approach is applied to a larger system, it is still possible to solve each meter one by one. Although we used the 9-bus system for the case study, the load data generated by this test system is big. The training data size of the 9-bus system used in the current case study is almost ten gigabytes. Adding a larger testing system into the case study would require substantial computational resources, which would need further investment in the hardware (including GPUs and

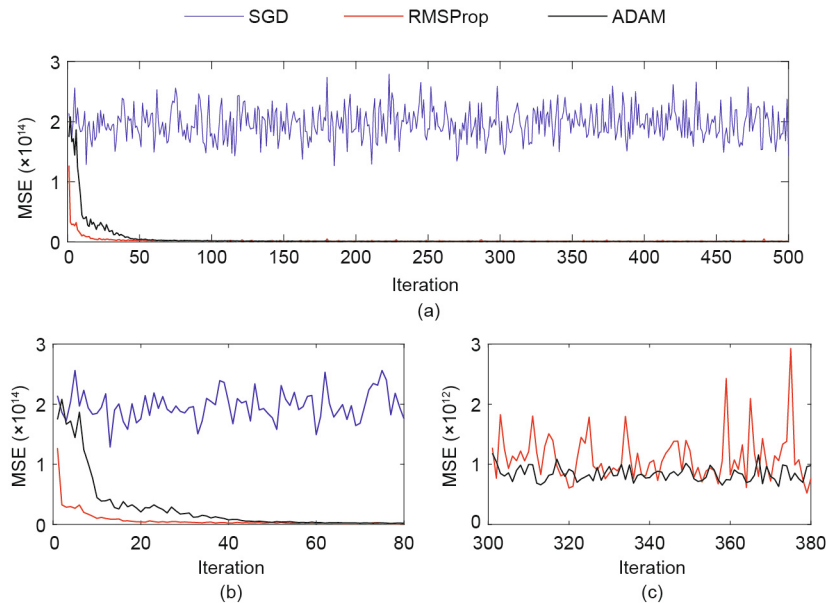


Fig. 25. Loss function comparisons using the RMSProp and ADAM algorithms compared with the SGD algorithm for case 4 with  $\eta = 100$ . MSE value in iterations (a) [1, 500]; (b) [1, 80], and (c) [300, 380].

larger memory). We would therefore like to leave this as our future work.

Furthermore, we will also perform trials on relatively higher frequency data in the future, such as recovering data from 100, 10, or 1 Hz. In fact, the SRPNSE approach can not only be applied in state estimation, but also in many other important modules in smart grids. The SRPNSE approach can help to improve data quality and thus overcome the obstacles and challenges caused by deployed meters, communication channels, and abnormal data intrusion. By applying the SRPNSE approach, the efficiency and security of existing industrial systems may be improved based on poor-quality data in practical situations without further investment and upgrading.

## Acknowledgements

This work was partially supported by the Training Program of the Major Research Plan of the National Natural Science Foundation of China (91746118), partially supported by the Shenzhen Municipal Science and Technology Innovation Committee Basic Research project (JCY20170410172224515), partially supported by funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society, and partially supported by Youth Innovation Promotion Association of Chinese Academy of Sciences.

## Compliance with ethics guidelines

Gaoqi Liang, Guolong Liu, Junhua Zhao, Yanli Liu, Jinjin Gu, Guangzhong Sun, and Zhaoyang Dong declare that they have no conflict of interest or financial conflicts to disclose.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.eng.2020.06.006>.

## References

- [1] Abur A, Gomez-Exposito A. Power system state estimation: theory and implementation. New York: Marcel Dekker; 2004.
- [2] Nagarajan A, Ayyanar R. Design and strategy for the development of energy storage systems in a distribution feeder with penetration of renewable resources. *IEEE Trans Sustain Energy* 2015;6(3):1085–92.
- [3] Tan S, De D, Song WZ, Yang J, Das SK. Survey of security advances in smart grid: a data driven approach. *IEEE Commun Surv Tutor* 2017;19(1):397–422.
- [4] Wu FF, Moslehi K, Bose A. Power system control centers: past, present, and future. *Proc IEEE* 2005;93(11):1890–908.
- [5] Monticelli A. State estimation in electric power systems: a generalized approach. Boston: Springer; 1999.
- [6] Yang Q, An D, Min R, Yu W, Yang X, Zhao W. On optimal PMU placement-based defense against data integrity attacks in smart grid. *IEEE Trans Inf Forensics Secur* 2017;12(7):1735–50.
- [7] Appasani B, Mohanta DK. A review on synchrophasor communication system: communication technologies, standards and applications. *Prot Control Mod Power Syst* 2018;3(1):37.
- [8] Gopakumar P, Mallikajuna B, Reddy MJB, Mohanta DK. Remote monitoring system for real time detection and classification of transmission line faults in a power grid using PMU measurements. *Prot Control Mod Power Syst* 2018;3(3):159–68.
- [9] Fan W, Liao Y. Wide area measurements based fault detection and location method for transmission lines. *Prot Control Mod Power Syst* 2019;4(4):53–64.
- [10] Liu Y, Ning P, Reiter MK. False data injection attacks against state estimation in electric power grids. *ACM Trans Inf Syst Secur* 2011;14(1):1–33.
- [11] Liang G, Zhao J, Luo F, Weller SR, Dong Z. A review of false data injection attacks against modern power systems. *IEEE Trans Smart Grid* 2017;8(4):1630–8.
- [12] Ashok A, Govindarasu M. Cyber attacks on power system state estimation through topology errors. In: Proceedings of the 2012 IEEE Power and Energy Society General Meeting; 2012 Jul 22–26; San Diego, CA, USA; 2012.
- [13] Kim J, Tong L. On topology attack of a smart grid: undetectable attacks and countermeasures. *IEEE J Sel Areas Commun* 2013;31(7):1294–305.
- [14] Liang G, Weller SR, Zhao J, Luo F, Dong Z. A framework for cyber-topology attacks: line-switching and new attack scenarios. *IEEE Trans Smart Grid* 2019;10(2):1704–12.
- [15] Li Z, Shahidepour M, Alabdulwahab A, Abusorrah A. Analyzing locally coordinated cyber-physical attacks for undetectable line outages. *IEEE Trans Smart Grid* 2018;9(1):35–47.
- [16] Deng R, Zhuang P, Liang H. CCPA: coordinated cyber-physical attacks and countermeasures in smart grid. *IEEE Trans Smart Grid* 2017;8(5):2420–30.
- [17] Liang G, Weller SR, Zhao J, Luo F, Dong ZY. The 2015 Ukraine blackout: implications for false data injection attacks. *IEEE Trans Power Syst* 2017;32(4):3317–8.
- [18] Pietrosevoli L, Rodríguez-Monroy C. The Venezuelan energy crisis: renewable energies in the transition towards sustainability. *Renewable Sustainable Energy Rev* 2019;105:415–26.
- [19] Ashok A, Govindarasu M, Ajjarapu V. Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Trans Smart Grid* 2018;9(3):1636–46.
- [20] Esmalifalak M, Liu L, Nguyen N, Zheng R, Han Z. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Syst J* 2017;11(3):1644–52.
- [21] Tsai RY, Huang TS. Multipleframe image restoration and registration. In: Huang TS, editor. *Advances in computer vision and image processing*. Greenwich: JAI Press; 1984. p. 317–39.
- [22] Shinde PD, Nabalwar SL. Image super-resolution. *Int J Sci Eng Technol Res* 2013;2(7):1505–8.
- [23] Borman S, Stevenson RL. Super-resolution from image sequences—a review. In: Proceedings of the 1998 Midwest Symposium on System and Circuits; 1998 Aug 9–12; Notre Dame, IN, USA; 1998.
- [24] Park SC, Park MK, Kang MG. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process Mag* 2003;20(3):21–36.
- [25] Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 2016;38(2):295–307.
- [26] Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, et al. ESRGAN: enhanced super-resolution generative adversarial networks. In: Proceedings of the 2018 European Conference on Computer Vision; 2018 Sep 8–14; Munich, Germany; 2018.
- [27] Fan W. Data quality: from theory to practice. *ACM SIGMOD Rec* 2015;44(3):7–18.
- [28] Schmid-Velasco M. Key principles of data quality [Internet]. Washington, DC: NAMATI; 2018 Apr [cited 2019 Mar 30]. Available from: <https://community.namati.org/t/key-principles-of-data-quality/42841>.
- [29] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521(7553):436–44.
- [30] Myttenaere AD, Golden B, Le Grand B, Rossi F. Mean absolute percentage error for regression models. *Neurocomputing* 2016;192:38–48.
- [31] Zhao Z, Li J, Wang L, Wei D, Gao K. Signal-to-noise ratio improvement of MAMR on CoX/Pt media. *IEEE Trans Magn* 2015;51(11):15303654.
- [32] Ruder S. An overview of gradient descent optimization algorithms. 2017. arXiv:1609.04747v2.
- [33] Hinton G, Srivastava N, Swersky K. Lecture 6e: RMSprop: Divide the gradient by a running average of its recent magnitude [Internet]. Coursera Lecture slides; [cited 2019 Apr. 15]. Available from: [https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf).
- [34] Rizwan M. Adam optimization algorithm [Internet]. engMRK; 2018 May 20 [cited 2019 Apr 15]. Available from: <https://engmrk.com/adam-optimization-algorithm/>.
- [35] Introduction of the 9-bus system (WSCC test case) [Internet]. AI-Roomi; 2015 Apr 11 [cited 2018 Nov 9]. Available from: <https://www.al-roomi.org/power-flow/9-bus-system>.
- [36] Gao J, Giri S, Kara EC, Bergés M. PLAID: a public dataset of high-resolution electrical appliance measurements for load identification research: demo abstract. In: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings; 2014 Nov 4–6; Memphis, TN, USA; 2014.
- [37] Bannore V. Iterative-interpolation super-resolution image reconstruction: a computationally efficient technique. Berlin: Springer-Verlag; 2009.