Hindawi
Scientific Programming
Volume 2020, Article ID 8828745, 20 pages
https://doi.org/10.1155/2020/8828745

Research Article

Intelligent Detection and Recovery of Missing Electric Load Data Based on Cascaded Convolutional Autoencoders

Xin Wang,[1] Yuanyi Chen,[2] Wei Ruan,[3] Qiang Gao,[1] Guode Ying,[1] and Li Dong[4]

[1]*State Grid Zhejiang Electric Power Co., Ltd., Taizhou Power Supply Company, Taizhou 318000, China*
[2]*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*
[3]*College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*
[4]*State Grid Zhejiang Electric Power Co., Ltd., Hangzhou 310007, China*

Correspondence should be addressed to Wei Ruan; 2191724107@qq.com

Received 28 August 2020; Revised 19 October 2020; Accepted 15 November 2020; Published 7 December 2020

Academic Editor: Ting Yang

Under the background of Energy Internet, the ever-growing scale of the electric power system has brought new challenges and opportunities. Numerous categories of measurement data, as the cornerstone of communication, play a crucial role in the security and stability of the system. However, the present sampling and transmission equipment inevitably suffers from data missing, which seriously degrades the stable operation and state estimation. Therefore, in this paper, we consider the load data as an example and first develop a missing detection algorithm in terms of the absolute difference sequence (ADS) and linear correlation to detect any potential missing data. Then, based on the detected results, we put forward a missing recovery model named cascaded convolutional autoencoders (CCAE), to recover those missing data. Innovatively, a special preprocessing method has been adopted to reshape the one-dimensional load data as a two-dimensional matrix, and hence, the image inpainting technologies can be conducted to address the problem. Also, CCAE is designed to reconstruct the missing data grade by grade due to its priority strategy, which enhances the robustness upon extreme missing situations. The numerical results on the load data of the Belgium grid validate the promising performance and effectiveness of the proposed solutions.

1. Introduction

Nowadays, measurement data are the foundation of the power system. The massive collected data especially the quality of electricity such as voltage, current, and load are tightly associated with safe operation and economic dispatch [1]. However, due to the growing size of the power grid and the massive number of field sensors, the absence and anomalies of data measurements cannot be avoided, which is mainly due to the failures and disabilities of terminal equipment or performance degradation of transmission channels [2, 3]. The problem of missing data may lead to serious consequences including stability, optimization, and fault prevention [4]. In addition, the measured values of the missing data can be replaced by unknown noise, which makes them more difficult to be perceived and diagnosed. Thus, the efficient and accurate data detection of data anomalies and recovery of the missing data is a fundamental for the development of data-driven analysis and advanced algorithmic solutions.

Essentially, the detection of missing data can be classified as a branch of abnormal or outlier detection, and the related researches have been investigated in the previous decades [5]. In the conventional algorithms, the methods involving residual and sudden change are discussed [6, 7]. Based on statistical analysis, 3σ criteria [8], Z-score [9], and clustering [10] are introduced. Specifically, in the power system, scholars suggest approaches combined with the correlativity of multimeasured data to improve accuracy [11]. However, these solutions are susceptible to data pollution due to the existence of missing data and lead to unsatisfactory performance.

In recent years, with the remarkable development of artificial intelligence (AI) technologies, more and more

scholars concentrate on the application of AI technologies in data recovery [12]. In literature [13], an unsupervised learning framework based on Wasserstein generative adversarial network (WGAN) is proposed to repair the missing data of active power, reactive power, voltage amplitude, and phase in power system, which achieves high accuracy, but the missing mask is required to be detected in advance, and the processing efficiency is relatively low due to the one-dimensional convolution. In [14], the adaptive neural fuzzy inference system (ANFIS) model is developed to recover the missing data of wind power. It performs better compared with traditional empirical methods but is difficult to generalize to other data. Furthermore, many other solutions have been adopted (e.g., [15–20]), but almost all the discussed solutions will deal with the missing data indiscriminately and without priority, which brings bad performance in extreme situations.

This paper will take Belgium load data (http://www.elia.be) as an instance to propose a new model to detect and recover the missing data. Firstly, beginning with the analysis of the characteristics of missing data, we present a detection method with ADS and linear correlation to detect the potential missing mask from the input incomplete data. Secondly, preprocessing will be applied to the incomplete data and the detected missing mask as well, which reshapes them as images (matrices). Finally, we demonstrate a CCAE model to address the missing regions in the grade of the image by grade with defined priority.

The rest of this paper will be organized as follows. In Section 2, related work and basic theories are introduced. In Section 3, the method of missing detection is investigated in Section 3.1 and then the missing recovery algorithm is developed in Sections 3.2 and 3.3. Section 4 gives some numerical results and discussion based on the load data of the Belgium grid, where a missing mask generation model is designed and employed for testing. Finally, Section 5 will conclude this paper and list the strength and weaknesses of the proposed model, and also, future work is discussed.

2. Related Work

In this chapter, the related work including abnormal detection, convolutional neural network (CNN), and autoencoder (AE) is introduced, which are the basic technologies applied in this paper.

2.1. Abnormal Detection. There many abnormal detection models widely used in the literature. The elementary idea is to model the pattern of data and then set a proper threshold or condition to pick out abnormal data in datasets. In this part, the 3σ criteria will be explored.

For the dataset that obeys the Gaussian distribution or known as a normal distribution [21], $N(\mu, \sigma^2)$, as shown in Figure 1. The mean μ and standard deviation σ can be estimated through maximum likelihood estimation (MLE). According to the features of Gaussian distribution, the possibility of data lying in range $(\mu - 3\sigma, \mu + 3\sigma)$ is 99.7% [22]. Hence, the data out of that range could be labeled as an outlier. Even though the original data maybe do not obey the Gaussian distribution strictly but just approximately, we can adjust the 3σ properly to 2.5σ or 3.5σ for examples, which still makes it work well.

2.2. Convolutional Neural Network. A convolutional neural network (CNN) is known as a feedforward neural network with convolutional computation that is a typical image processing paradigm in deep learning [23]. CNN is capable of representation learning and can process the input image with shift-invariant classification. Therefore, it is called shift-invariant artificial neural network (SIANN) as well. An example of CNN-based classification is illustrated in Figure 2.

The convolutional computation in CNN illustrated in Figure 3 differs from that in equation (1). In CNN, the convolution is done for two-dimensional input and does not require the reverse operation to the final output:

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(\tau) g(x - \tau) \mathrm{d}\tau. \tag{1}$$

The reason we employ CNN instead of artificial neural network (ANN) [24] to process pictures is that the parameter sharing in CNN enables us to analyze images with much fewer parameters. This is because the fully connected layers are only used in the last several layers of CNN. Hence, its training efficiency is better than ANN, when tackling with the semantics of images. Also, the training strategies can be supervised or unsupervised, which depend on the targets.

Typically, there are different categories of layers in CNN, e.g., convolutional layer, pooling layer, and fully connected layer. The convolutional layer applies the convolutional kernel to the inner product the input, region by region, and the features can be extracted in the output, as demonstrated in Figure 3.

The pooling layer is designed to reduce the output size of the convolutional layer and then diminish the required parameters in the following convolutional layers as well. Another important benefit is that the pooling layer can alleviate overfitting and increase generalization ability. The major kinds of pooling layers include max pooling and average pooling, and the computation is shown in Figure 4.

The fully connected layer is similar to the layers in ANN. The only difference is that we will first flatten the two-dimensional output of the convolutional layer or pooling layer, as a one-dimensional vector, and then, the fully connected layer is employed, as described in Figure 5. Thus, the fully connected layer is also named the flatten layer on CNN.

CNN has been widely adopted in computer vision, such as image classification [25] and object recognition [26]. And the development of autopilot is also firmly incorporated with CNN [27].

2.3. Autoencoder. Autoencoder (AE) is a kind of supervised or unsupervised ANN used for data compression,

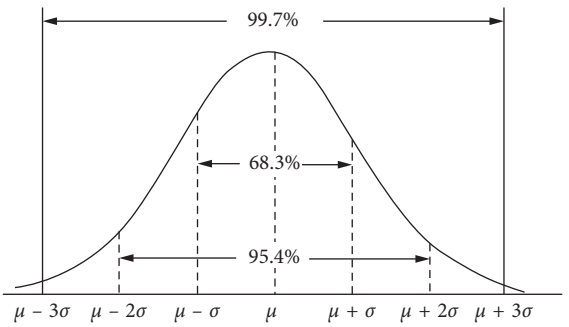


Figure 1: Gaussian distribution.

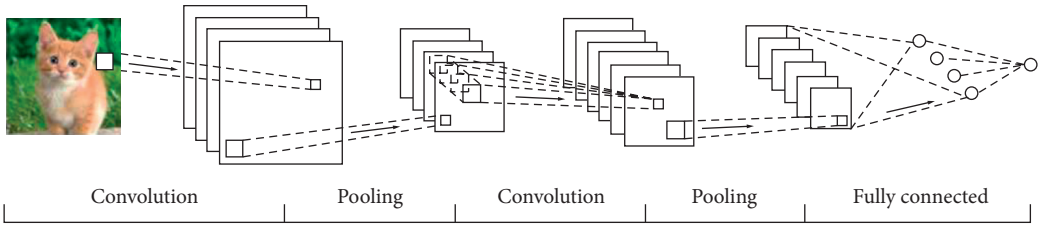


Figure 2: An example of CNN for classification.

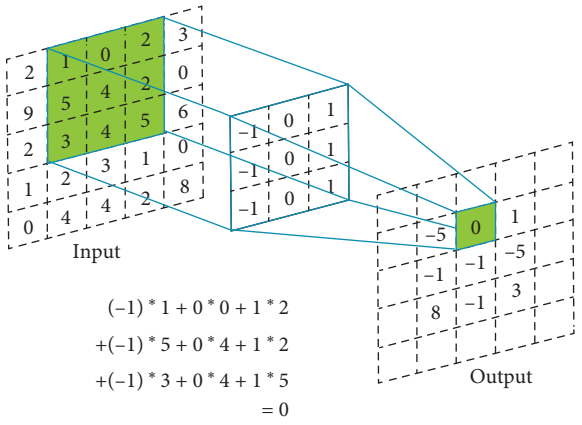


Figure 3: Convolutional computation on CNN.

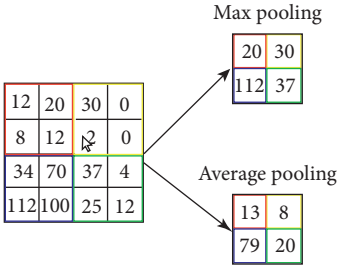


Figure 4: Two kinds of the pooling layer.

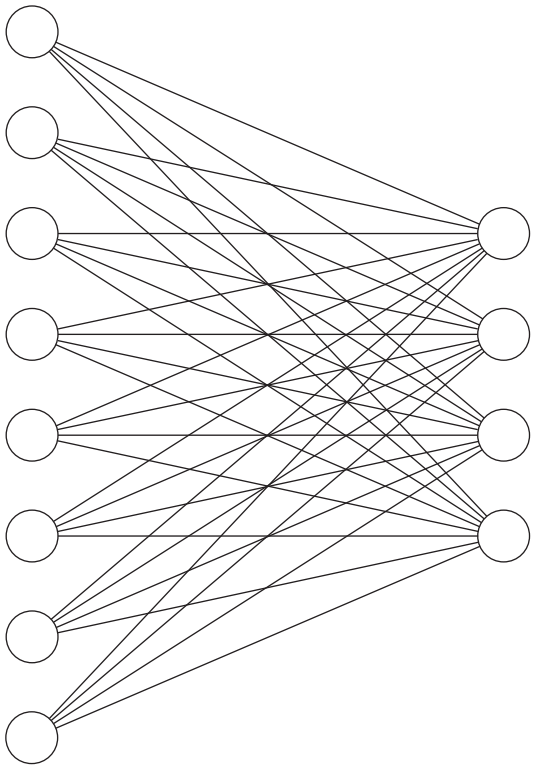


Figure 5: Fully connected layer.

representation learning, dimensionality reduction [28], and image denoising [29] which was first proposed by Rumelhart in 1986 [30]. An example of AE is presented in Figure 6.

The output of AE is required to be the same as the input as possible, as defined in (2) to (4). After training, AE can efficiently encode the features of input data:

$$f: X \longrightarrow Z, \tag{2}$$

$$g: Z \longrightarrow X, \tag{3}$$

$$f, g = \arg\min_{f,g} \|X - g[f(X)]\|^2. \tag{4}$$

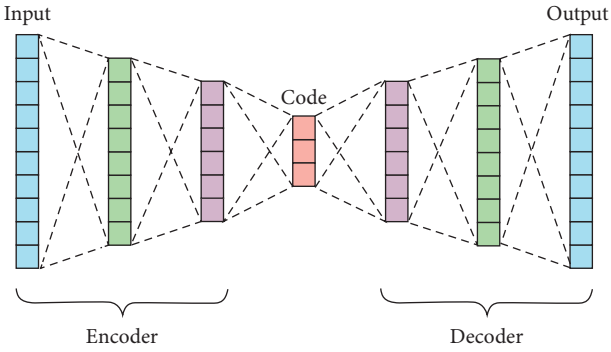


Figure 6: Illustration of an AE example.

As shown in Figure 6, AE has an encoder and a decoder, which essentially are fully connected layers. The encoder is responsible for feature learning, while the decoder should be able to reconstruct the input from the encoded features. Notice that when dealing with image problems, AE also can be made of convolutional layers [31] and that is exactly the basic structure of this paper.

3. Methodology

In this section, we will firstly discuss a missing detection algorithm to obtain the missing mask from the input incomplete data. Then, the input data and the detected missing mask will be further preprocessed as matrices, respectively. After that, we build a CDCAE model to recover the missing data in the matrices and reshape the matrices back as one-dimensional time series.

3.1. Missing Detection

3.1.1. Classification of Missing Segments. Usually, the missing data caused by the faults of sampling and communication equipment are mainly manifested as discrete missing points, continuous missing segments, or the combination of the two [32], as demonstrated in Figure 7. When the contact of the sampling terminal is loose or other disturbance occurs, the waveform of the data may appear as discrete missing points. However, in each link of transmission, the loss of data packet by temporary communication failure will result in a continuous missing segment.

The discrete missing points can be regarded as special cases of the continuous missing segments with a length of one. Hence, we only consider the missing segments.

In fact, the measurement values at the missing segments are usually not exactly zero or NA, but the noise is distributed near zero. This kind of noise includes not only the background noise from sampling equipment and transmission channel but also the noise due to failures or faults as well. To simplify the model, it is reasonable to assume that the noise at the missing segments is subjected to the Gaussian distribution with zero mean and relatively smaller variance than that of the normal data signal.

In this work, the missing segments are classified as typical missing segments and atypical missing segments, as illustrated in Figures 8 and 9 . Since the normal data are very likely to be far away from zero while the noise at missing segments is distributed near zero, for the beginning and the end of a missing segment, the curve will show an abnormal jump. We call those missing segments the typical missing segments, which are the most cases as well. On the contrary, there is a very low possibility that the normal data before or after a missing segment are also distributed near zero, which leads to the overlap of the ranges of the normal data and the missing data. In this situation, the curve may have an inapparent jump at the beginning or end of the missing segment. Therefore, we name those missing segments as atypical missing segments.

Here, the threshold of the abnormal jump is defined as the abnormal values in the ADS of the input sequence. The differential sequence (DS) for the load data is naturally subjected to the Gaussian distribution as well with zero mean; thus, we can simply apply the 3σ criteria to specify the abnormal values in ADS and then to locate all the abnormal jumps. However, because there might also exist abnormal jumps in atypical missing segments as indicated in Figure 9, it is still unable to locate the two kinds of missing segments.

In addition to the sudden jump of the curve, another feature might not be so noticeable. Under the small window size, the segments of normal data usually show a shape of a regular curve which has a high linear correlation for time, while the missing data segments will have a much lower linear correlation. This could be another important factor when distinguishing the missing data from normal data.

3.1.2. The Criterion of Sigma and Linear Correlation. Based on the definition in Section 3.1.1, the detection problem can be separated as two subproblems for typical and atypical missing segments, respectively. This paper will firstly propose a method to diagnose the typical missing segments. And then with the results, the remaining atypical missing segments can be detected.

Assume the ground truth data as $X = (x_1, x_2, \ldots, x_{n-1}, x_n)$, the input incomplete data as $X' = (x_1', x_2', \ldots, x_{n-1}', x_n')$, the missing mask as $M = (m_1, m_2, \ldots, m_{n-1}, m_n)$, and the noise signal as $W = (w_1, w_2, \ldots, w_{n-1}, w_n)$, then

$$X' = X - X \odot M + W \odot M, \tag{5}$$

where m_i is a binary number and $m_i = 1\,(0)$ represents x_i is missing (normal), noise w_i is subjected to the Gaussian distribution $N(0, \sigma_N^2)$, and $\odot$ is the element-wise production operator.

The steps to detect the missing mask M from the input incomplete data X' are described as follows:

(1) Define the DS and ADS of X' as

$$\begin{aligned} \mathrm{DS}(X') &= \{d_i | d_i = x_i' - x_{i+1}', x_i' \in X'\}, \\ \mathrm{ADS}(X') &= \{a_i | a_i = |x_i' - x_{i+1}'|, x_i' \in X'\}. \end{aligned} \tag{6}$$

(2) Assume d_i is subjected to $N(0, \sigma_D^2)$ and calculate the standard deviation σ_D. Label all the a_i in $\mathrm{ADS}(X')$

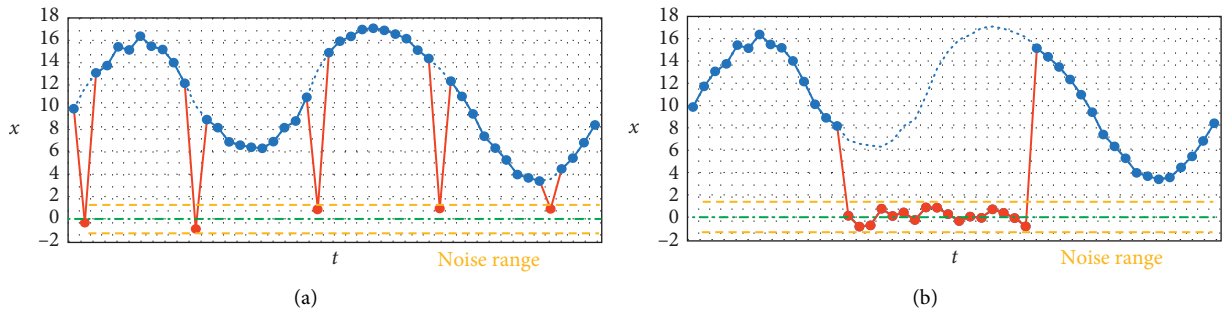


Figure 7: The (a) discrete and (b) continuous missing data (red) in the noise range.

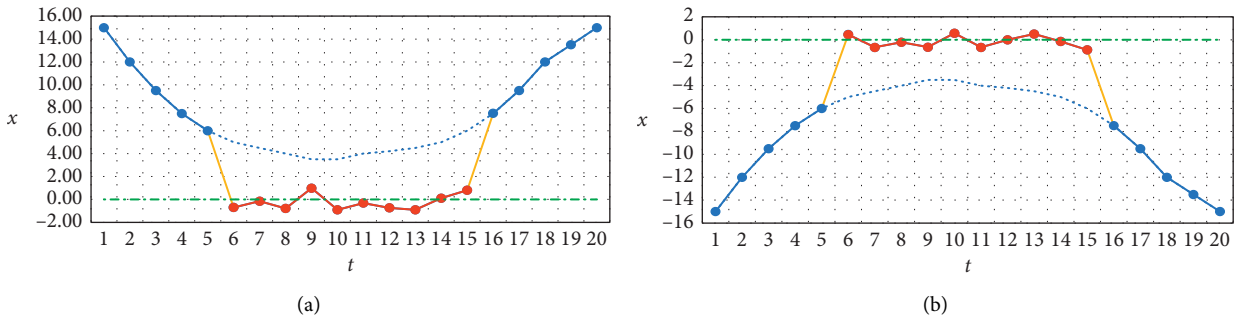


Figure 8: Examples of typical missing segments with two obvious jumps (yellow): (a) typical missing segment 1; (b) typical missing segment 2.

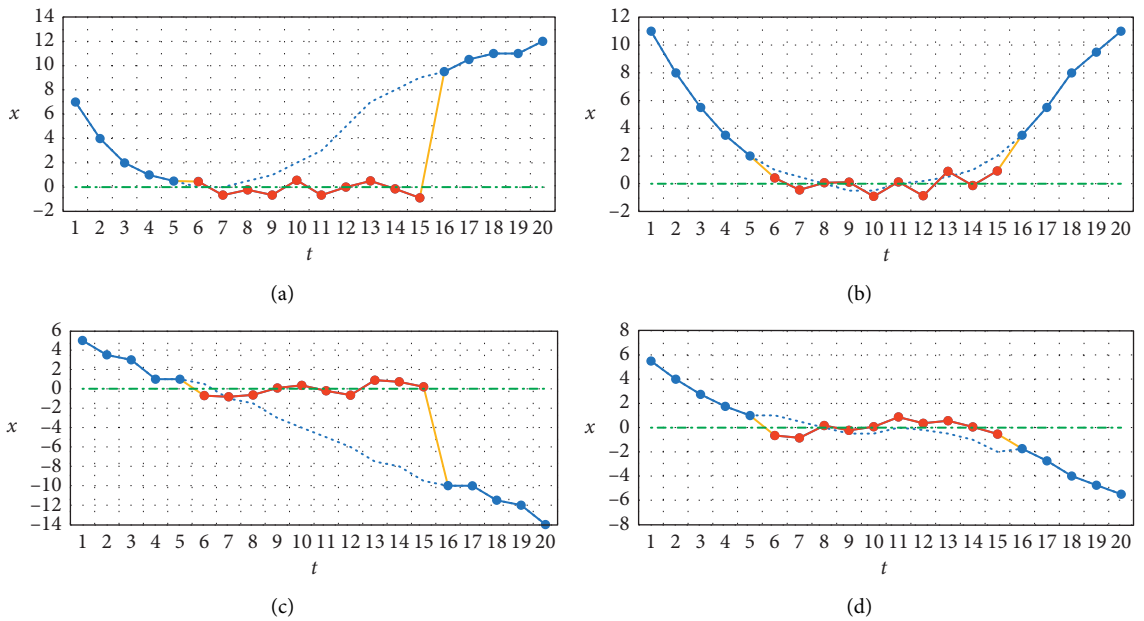


Figure 9: Examples of atypical missing segments with only one or none obvious jump (yellow): (a) atypical missing segment 1; (b) atypical missing segment 2; (c) atypical missing segment 3; (d) atypical missing segment 4.

that $a_i \geq 3.5\sigma_D$ as abnormal values due to abnormal jumps.

(3) Compute the linear correlation r for every segment between the adjacent labeled a_i for time. If $r \leq 0.3$, the corresponding segment is labeled as a typical missing segment.

(4) Since the noise signal w_i in missing segments obeys $N(0, \sigma_N^2)$, we can estimate the unknown σ_N through the detected typical missing segments.

(5) Locate all the segments of X' within $(-2.5\sigma_N, +2.5\sigma_N)$ except the detected typical missing segments in (3), and check the linear correlation r

again. For those with $r \leq 0.3$, label the corresponding segments as atypical missing segments.

(6) Set m_i on the detected typical and atypical missing segments as 1 and the other as 0. Obtain the detected missing mask M.

To evaluate our detection algorithm, we will refer to the confusion matrix [33] and calculate the precision P, recall R, and F_1 score [34], which is defined in 1 and in the following equations:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{7}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{8}$$

$$F_1 = \frac{2\mathrm{PR}}{P + R}. \tag{9}$$

3.2. Preprocessing

3.2.1. Normalization. After the missing mask M in X' has been detected, it is necessary to normalize X' as $\widehat{X'} = (\widehat{x}_1', \widehat{x}_2', ..., \widehat{x}_{n-1}', \widehat{x}_n')$ out of convenience, and a possible choice is shown as follows [35]:

$$\widehat{x}_i' = \frac{x_i' - \min(X')}{\max(X') - \min(X')}. \tag{10}$$

But due to the pollution from the Gaussian noise in the missing segments, we may not gain the real maximum and minimum values, which will lead to a gap in values distribution after normalization, and the values of data cannot fill this normalized range because the is possible even higher (lower) than the real maximum (minimum).

To avoid this problem, we will calculate the maximum and minimum values only in the normal data $\dot{X} = \{x_i' | x_i' \in X', m_i\}$ and then normalize x_i' as

$$\widehat{x'}_i = (1 - m_i)\left(\frac{x_i' - \min(\dot{X})}{\max(\dot{X}) - \min(\dot{X})} 0.99 + 0.01\right). \tag{11}$$

The values of $\widehat{x}_i'$ can well fully fill the range (0, 1), and there is no more apparent gap in the distribution of the values as well. Notice that a potential advantage by doing so is that noise in x_i'will be replaced by zero which represents missing data uniquely and vice versa. So, the following recovery algorithm can easily know which data are the missing data by just reading the zero values, even without knowing the missing mask M.

When preparing the training datasets, the ground truth data X also will be normalized as $\widehat{X} = (\widehat{x}_1, \widehat{x}_2, \ldots, \widehat{x}_{n-1}, \widehat{x}_n)$, where

$$\widehat{x}_i = \frac{x_i - \min(\dot{X})}{\max(\dot{X}) - \min(\dot{X})} 0.99 + 0.01. \tag{12}$$

It should be noted that the local maximum and minimum values used here for normalization may not be the global maximum and minimum of the normal data before missing happens, for the reason the real maximum and minimum could be blocked by the missing mask. Thus, when we normalize those blocked data, the results are probably beyond range (0, 1). But fortunately, since the data discussed in this paper are load data with obvious periodicity, the local maximum and minimum would be very close to the global maximum and minimum.

Table 1: Confusion matrix in detection.

	Missing	Normal
Predicted missing	True positive (TP)	False positive (FP)
Predicted normal	False negative (FN)	True negative (TN)

3.2.2. Grade. Repairing the missing data is to figure out how to make the best estimation for the missing data based on the adjacent normal data. Hence, making full use of the adjacent normal data is the key to solve the problem.

We note that, for different missing data at the same missing segment, the specific locations of the missing data are different, and also the numbers of adjacent available normal data are different. In detail, the missing data near the beginning or the end of the missing segments are closer to the normal data, which makes them easier to be recovered, while the missing data at the center of the missing segments are far away from the normal data and difficult to be addressed.

To design more targeted recovery algorithms for different missing situations, this paper will introduce powerful improvement on the detected missing mask M in Section 3.1.2. The core idea is that the missing data at the center should be recovered based on the recovery of the missing data at edges, which indicates the edge missing data have a higher priority and are before being recovered:

$$T = \mathrm{Grade}(M). \tag{13}$$

To distinguish the missing data at different positions, we grade M into K submissing masks $T = (G_1, G_2, \ldots, G_{K-1}, G_K)$ separately, where K is the grade parameter and the j-th submissing mask is $G_j = (g_1^{(j)}, g_2^{(j)}, \ldots, g_{n-1}^{(j)}, g_n^{(j)})$. The missing segments in M will be divided into smaller submissing segments from two ends toward the inside. The ratio of the submissing segments in G_j concerning that in M is defined as R_j:

$$\begin{aligned} \sum_{j=1}^{K} G_j &= M, \\ \sum_{j=1}^{K} g_i^{(j)} &= m_i, \\ \sum_{j=1}^{K} R_j &= 1, \end{aligned} \tag{14}$$

where $g_i^{(j)}$ is a binary number similar as m_i.

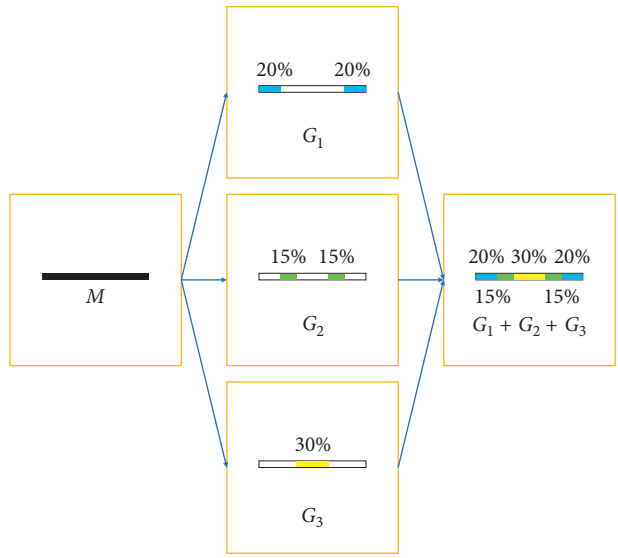


Figure 10: Submissing masks G_1, G_2, and G_3 from the missing mask M.

In this paper, the hyperparameters $K = 3$ and corresponding $R_1 = 40\%$, $R_2 = 30\%$, and $R_3 = 30\%$, as shown in Figure 10.

3.2.3. Reshape. The load data in the power system change along with the patterns of society operation and production. Hence, it will show evident multiple periodicities in days, weeks, quarters, and years. When repairing this kind of data, only referring to the continuity between the directly adjacent normal data and the missing data (e.g., data before and after half an hour) is not enough and leads to bad performance. Instead, the periodicity should be taken into consideration, meaning that we have to also refer to the data in adjacent cycles (e.g., the data at the same time but different periods), since those data, to some extent, are indirectly adjacent and share very similar patterns, as presented in Figure 11.

In terms of semantic intensity, direct adjacency is stronger than indirect adjacency, but they can be combined as a reference. For the data at the edges of the missing segments, due to the constraint of the waveform continuity, the directly adjacent data are nearly the most important repair reference; but for the missing data inside the missing segment, there will not be any directly adjacent data for reference, while the indirect adjacent data become a very significant factor instead. However, traditional mathematical estimation and interpolation methods can only perceive data inside a very limited window around the missing data, so they cannot make full use of the information from indirect adjacent data. As a result, the repair accuracy is low, especially for the long continuous missing segments.

When dealing with similar problems, literature [35] proposes a method to transfer the one-dimensional harmonic data into a two-dimensional grayscale image by

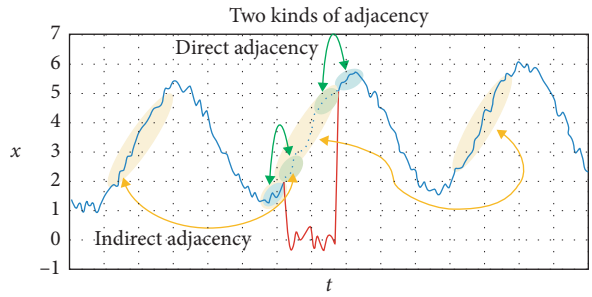


Figure 11: Direct and indirect adjacency relationship in load data.

periodic truncation and reshaping, as demonstrated in Figure 12$n = km$. Inspired by this, this paper will reshape the one-dimensional incomplete data $\widehat{X'}$ and corresponding submissing masks G_j into matrices.

Take $\widehat{X'}$, for example, assume the number of sampling points per day is m, for a dataset with k days, and the size . Then, reshape $\widehat{X'}$ into a k-by-m matrix $A_{\widehat{X'}}$:

$$A_{\widehat{X'}} = \text{reshape}\left(\widehat{X'}\right) = \left[a_{i,j}\right]_{k\times m}, \tag{15}$$

where $a_{i,j} = \widehat{x}'_{(i-1)m+j}$. For the load data of Belgium grid, $k = 2000$ and $m = 96$.

And similarly, matrices A_{G_j} can be obtained. The two-dimensional structure of the matrix enables the direct and indirect adjacency to be compatible with each other in rows and columns separately. And the shaped matrix can be understood as a special "generalized" image. Based on the above analysis, when we reshape the data, the problem of repairing the missing one-dimensional data becomes the problem of inpainting a two-dimensional image. In

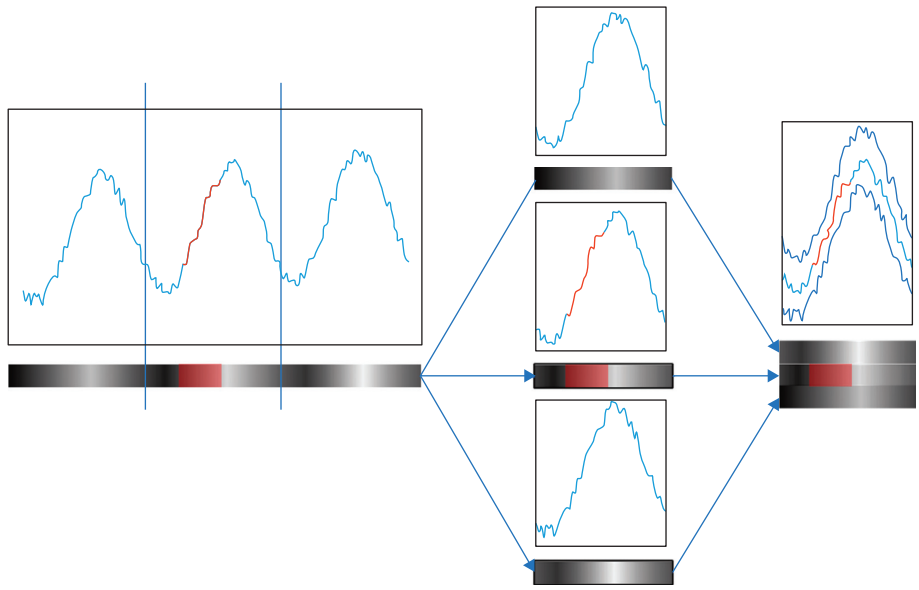


Figure 12: Reshaping of the one-dimensional data to the two-dimensional image.

the following discussion, we will combine the deep learning and image processing techniques to address this problem.

3.2.4. Edge Padding. After reshaping, the data located in the center of the matrices will have more available adjacent data than that at the edge of the matrices, which indicates the number of available adjacent data is radially attenuated outward from the core of the matrices, making the recovery of edges data more difficult than that of the central data.

To solve the problem of an unbalanced distribution of the available adjacent data in the radial direction, the most direct and effective method is to arrange some additional data on the edges. When we truncate the original one-dimensional data and reshape it into matrices, two adjacent data at the truncated point will be separated and then arranged at the end of this row and the beginning of the next row accordingly. So, the left edge and the right edge of the matrix are - "adjacent- " but misplaced by one row, as illustrated in Figure 13. Thus, the data in the left and right edges can be used as padding data for each other.

Take $A_{\widehat{X'}}$, for example, to improve this unbalanced distribution, two k-by-p padding matrices $B_{\widehat{X'}}$ and $C_{\widehat{X'}}$ are designed, and p is the padding depth:

$$B_{\widehat{X'}} = \left[b_{i,j}\right]_{k\times p}, C_{\widehat{X'}} = \left[c_{i,j}\right]_{k\times p}, \tag{16}$$

where

$$b_{i,j} = \begin{cases} a_{i-1,m-p+j} & i > 1 \\ 0 & i = 1 \end{cases}, \quad c_{i,j} = \begin{cases} a_{i+1,j} & i < k \\ 0 & i = k \end{cases} \tag{17}$$

Define the ratio of p and m as hyperparameter padding ratio η:

$$\eta = \frac{p}{m}. \tag{18}$$

Then, arrange $B_{\widehat{X'}}$ and $C_{\widehat{X'}}$ to the left and right sides of $A_{\widehat{X'}}$, respectively, to get a k-by-L padding matrix $Z_{\widehat{X'}}$ as

$$Z_{\widehat{X'}} = \text{Padding}\left(A_{\widehat{X'}}\right) = \left[B_{\widehat{X'}} A_{\widehat{X'}} C_{\widehat{X'}}\right] = \left[z_{i,j}\right]_{k\times L}, \tag{19}$$

where $L = m + 2p$.

And similarly, padding matrix Z_{G_j} can be attained.

3.2.5. Slice. Because there are no padding data on the upper and lower sides of padding matrices, we will cut the padding matrices into smaller L-by-L slices and set proper overlapped rows as padding data on the upper and lower sides.

Take $Z_{\widehat{X'}}$, for example, the slices are defined as $S_{\widehat{X'}}$, where the t-th slice $S_{\widehat{X'}}(t)$ is a L-by-L matrix:

$$S_{\widehat{X'}} = \text{Slice}\left(Z_{\widehat{X'}}\right), \quad S_{\widehat{X'}}(t) = \left[s_{i,j}^{(t)}\right]_{L\times L}, \tag{20}$$

where $s_{i,j}^{(t)} = z_{(t-1)m+i,j}$ and $t \le n_s = \lfloor k - 2p/m \rfloor$.

For adjacent slices $S_{\widehat{X'}}(t)$ and $S_{\widehat{X'}}(t+1)$, there are $2p$ overlapped rows on the lower side of $S_{\widehat{X'}}(t)$ and the upper side of $S_{\widehat{X'}}(t+1)$. Hence, the first and last p rows and columns are redundant data. And as a result, when recovering a slice, we only need to consider its center m-by-m region which is defined as the core region:

$$\text{Core}\left(S_{\widehat{X'}}(t)\right) = \left[u_{i,j}^{(t)}\right]_{m\times m}, \tag{21}$$

where $u_{i,j}^{(t)} = s_{i+p,j+p}^{(t)}$.

In particular, the core areas of the first (last) slice will include the upper (lower) p rows, which leads to a $(m+p)$-by-m matrix:

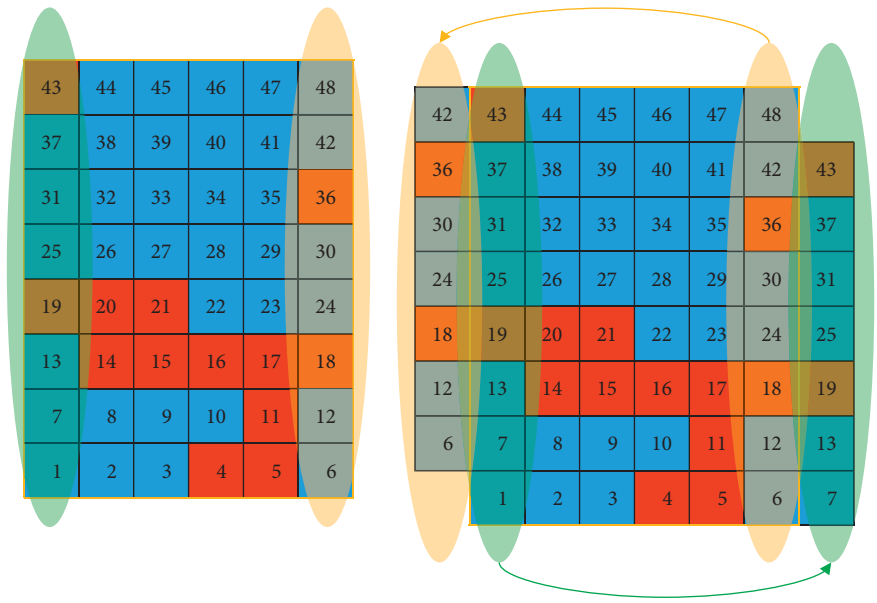


Figure 13: Misplacement adjacency between the left and right sides and self-padding.

$$\begin{aligned}\mathrm{Core}\left(S_{\widehat{X'}}(1)\right) &= \left[u_{i,j}^{(1)}\right]_{(m+p)\times m},\\ \mathrm{Core}\left(S_{\widehat{X'}}(n_s)\right) &= \left[u_{i,j}^{(n_s)}\right]_{(m+p)\times m},\end{aligned} \tag{22}$$

where $u_{i,j}^{(1)} = s_{i,j+p}^{(1)}$ and $u_{i,j}^{(n_s)} = s_{i+p,j+p}^{(n_s)}$.

Similarly, slices S_{G_j} can be obtained.

3.3. Missing Recovery. When we use a two-dimensional matrix to represent one-dimensional data, the problem of recovering one-dimensional data becomes the problem of image inpainting. Recently, CNN and GAN technologies in the deep learning field have excellent performance on image inpainting. In [36], the author uses the previous five convolutional layers from the AlexNet [23] as an encoder to extract the features of images with missing areas and then uses six deconvolutional layers as a decoder to restore the missing regions from the learned features. Inspired by this, this paper will put forward a convolutional autoencoder-based network to recover the missing data in the preprocessed matrices.

As we know, the feature learning in the AlexNet is designed for the object classification problem, where the object positions will not matter since the category of an object does not depend on its position. And because of the use of the max pool, "valid" padding, and flatten layer, the size of the tensor will be compressed, which will blur the position information. Therefore, it is nearly impossible to trace back to the original positions of features in deep layers, especially when the input data are damaged heavily.

For reshaped matrices, because the semantic information is not uniformly distributed in rows and columns, the original position of the feature is even more important when extracting the features. If the convolution and deconvolution framework is applied directly, the fuzzy location of features in deep layers will further degrade the situations. Moreover, since the "generalized" images are usually low-dimensional with low rank, the context encoder in [36] will have bad performance as the author reminded.

Thus, this paper presents a CCAE network based on the context encoder as follows: only use convolutional layers without any flatten, pooling, or fully connected layers, replace the padding mode from "valid" to "same," and set the stride as one which keeps the height and width of the output tensors in every layer equal to the input, namely, L-by-L. Finally, the output matrix will be restored to one-dimensional.

3.3.1. Network Structure. To recover the preprocessed incomplete data $S_{\widehat{X'}}$ with S_{G_j}, a CCAE model is proposed with the structure shown in Figure 14.

There are K convolutional autoencoders (CAE_j) blocks cascaded in CCAE corresponding to K submissing masks S_{G_j}. Each CAE_j has an encoderE_j, a decoder D_j, and a filter F_j. Encoder E_j and decoder D_j are made of Q convolutional layers, respectively, where the stride is one, padding mode is "same," and activation function is Relu. The filter F_j is used to update the recovery results of the missing segments in S_{G_j}, that is,

$$F_j = F_{j-1} + D_j \odot S_{G_j}. \tag{23}$$

And then, F_j will be the input of E_{j+1} in CAE_{j+1}, which ensures $S_{G_{j+1}}$ will be recovered based on the recovery result of S_{G_j}. In particular,

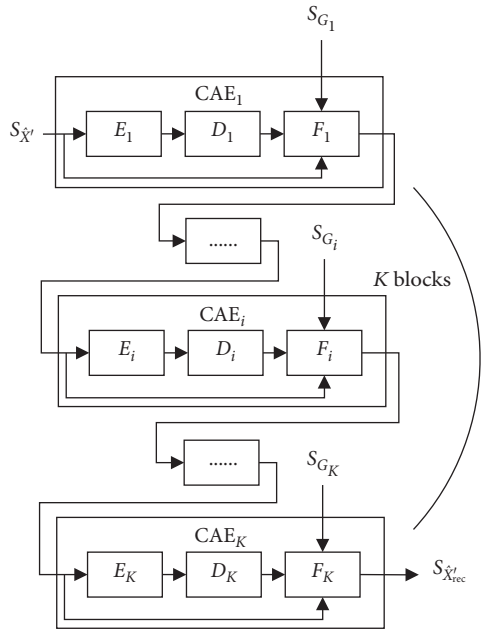


Figure 14: General structure of CCAE.

$$F_1 = S_{\widehat{X'}} + D_1 \odot S_{G_1},$$
$$S_{\widehat{X'_{\text{rec}}}} = F_K, \tag{24}$$

where $S_{\widehat{X'_{\text{rec}}}}$ is the recovery result of $S_{\widehat{X'}}$.

In this paper, $K = 3$ as mentioned and $Q = 2$. As a result, there are four convolutional layers in each CAE$_j$. The number of convolutional kernels in each layer is 64, 96, 32, and 1 in CAE$_1$, 32, 64, 16, and 1 in CAE$_2$, and 32, 64, 16, and 1 in CAE$_3$. The corresponding kernel size is (5, 5), (11, 11), (5, 5), and (3, 3) in CAE$_1$; (7, 7), (5, 5), (3, 3), and (3, 3) in CAE$_2$; and (5, 5), (5, 5), (3, 3), and (3, 3) in CAE$_3$, as demonstrated in Figure 15. The red rectangles represent the core areas in Section 3.2.5.

Without a flatten layer and fully connected layers, we only employ two convolutional layers for feature learning. As a result, the learned feature will locate at its original position, which means this network will encoder a feature vector for every point in the input matrix and just put that feature vector at the same position as the computed point. Through that, the position information can be retained during feature learning to the most extend.

3.3.2. Loss Function. One thing we should notice is that the size of the output matrices of CCAE is still L-by-L, in which only their m-by-m core areas matter as mentioned in Section 3.2.5. Hence, the loss function $\mathscr{L}$ is defined as the root mean squared error (RMSE) of the missing data within core areas:

$$\mathscr{L}\left(S_{\widehat{X'_{rec}}}, S_{\widehat{X}}\right) = \sqrt{\frac{1}{\sum_{i=1}^{n} m_i} \sum_{t=1}^{n_s} \left\| \text{Core}\left(S_{\widehat{X'_{rec}}}(t) - S_{\widehat{X}}(t)\right)\right\|_2^2}, \tag{25}$$

where $S_{\widehat{X}}$ is obtained through the same preprocessing in Section 3.2.

3.3.3. Restore. Since $S_{\widehat{X'_{\text{rec}}}}$ is a set of normalized L-by-L matrix slices $S_{\widehat{X'_{\text{rec}}}}(t)$, it should be restored back to one-dimensional time series, which means the reverse operations of preprocessing in Section 3.2. Assume $n_s = \lfloor k - 2p/m \rfloor = k - 2p/m$, then reorganize the core areas of slices $S_{\widehat{X'_{\text{rec}}}}(t)$ as a k-by-m matrix A_F:

$$A_F = \begin{bmatrix} \text{Core}\left(S_{\widehat{X'_{\text{rec}}}}(1)\right) \\ \vdots \\ \text{Core}\left(S_{\widehat{X'_{\text{rec}}}}(n_s)\right) \end{bmatrix} = \left[f_{i,j}\right]_{k \times m}. \tag{26}$$

After that, reshape A_F into a n-by-1 time series $\widehat{Y}_{\text{rec}} = (\widehat{y}_1, \widehat{y}_2, ..., \widehat{y}_{n-1}, \widehat{y}_n)$, where $\widehat{y}_h = f_{i,j}$ for those $(i-1)m + j = h$. Finally, reverse the normalization in Section 3.2.1 to get the restored results $Y_{\text{rec}} = (y_1, y_2, ..., y_{n-1}, y_n)$ in which

$$y_h = \left(\widehat{y}_h - 0.01\right) \frac{\max(\dot{X}) - \min(\dot{X})}{0.99} + \min(\dot{X}). \tag{27}$$

4. Experimental Results and Discussion

In this section, we will validate our detection and recovery algorithms. Load data from the Belgium grid will be conducted as training and test data, and a missing mask generation model is presented to produce generative missing masks under different parameters.

4.1. Experimental Design

4.1.1. Missing Mask Generation Model. In the original load data of the Belgium grid, there are very few missing segments; therefore, it could be regarded as ground truth data X. Then, we have to manually generate missing masks M for detection and recovery testing.

Define the missing rate as γ where

$$\gamma = \frac{\sum_{i=1}^{n} m_i}{n}. \tag{28}$$

If we just randomly select some segments in X as missing segments, the segments collision might happen, as demonstrated in Figure 16, which results in a lower missing rate than the given γ. Therefore, a stratified sampling model is proposed to solve the problem, as shown in Figure 17.

Define the number of missing data as NMD, namely,

$$\text{NMD} = \sum\nolimits_{i=1}^{n} m_i = n\gamma. \tag{29}$$

Then, divide NMD as NMS segments where NMS is the number of missing segments and

$$\text{NMS} = \text{random}\left([\text{NMD}\alpha], [\text{NMD}\beta]\right), \tag{30}$$

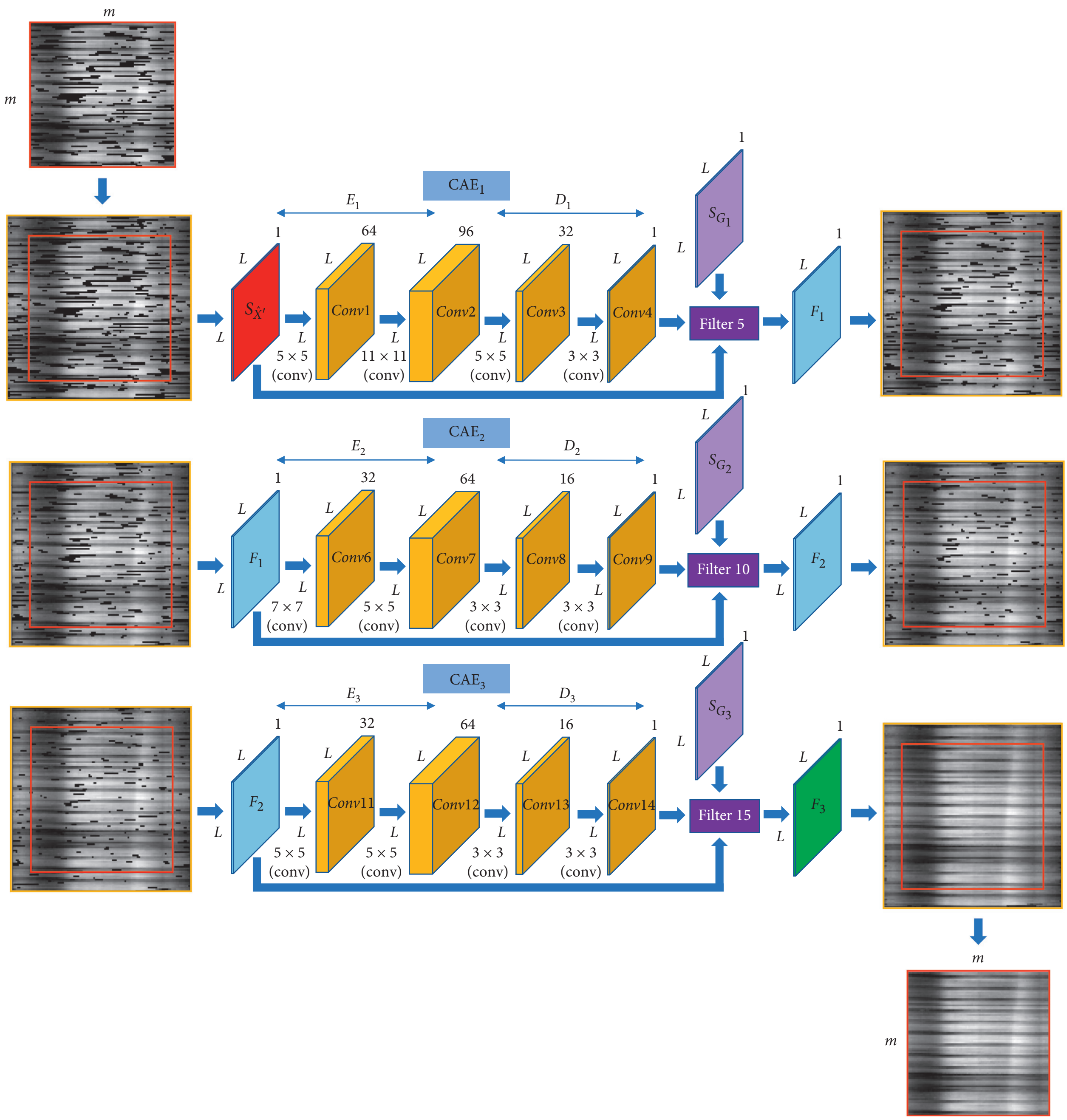


FIGURE 15: An instance of CCAE with $K = 3$ and $Q = 2$.

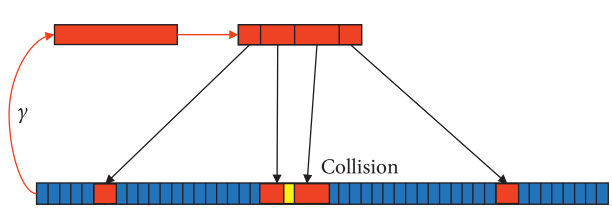


FIGURE 16: Segments collision in random generation of the missing mask.

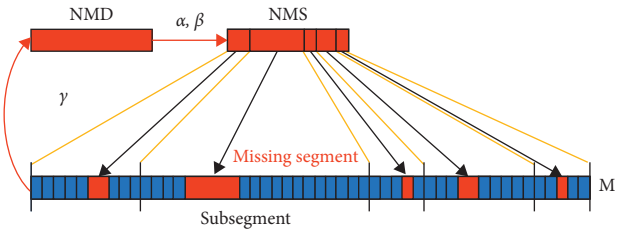


Figure 17: Stratified sampling generation model of the missing mask.

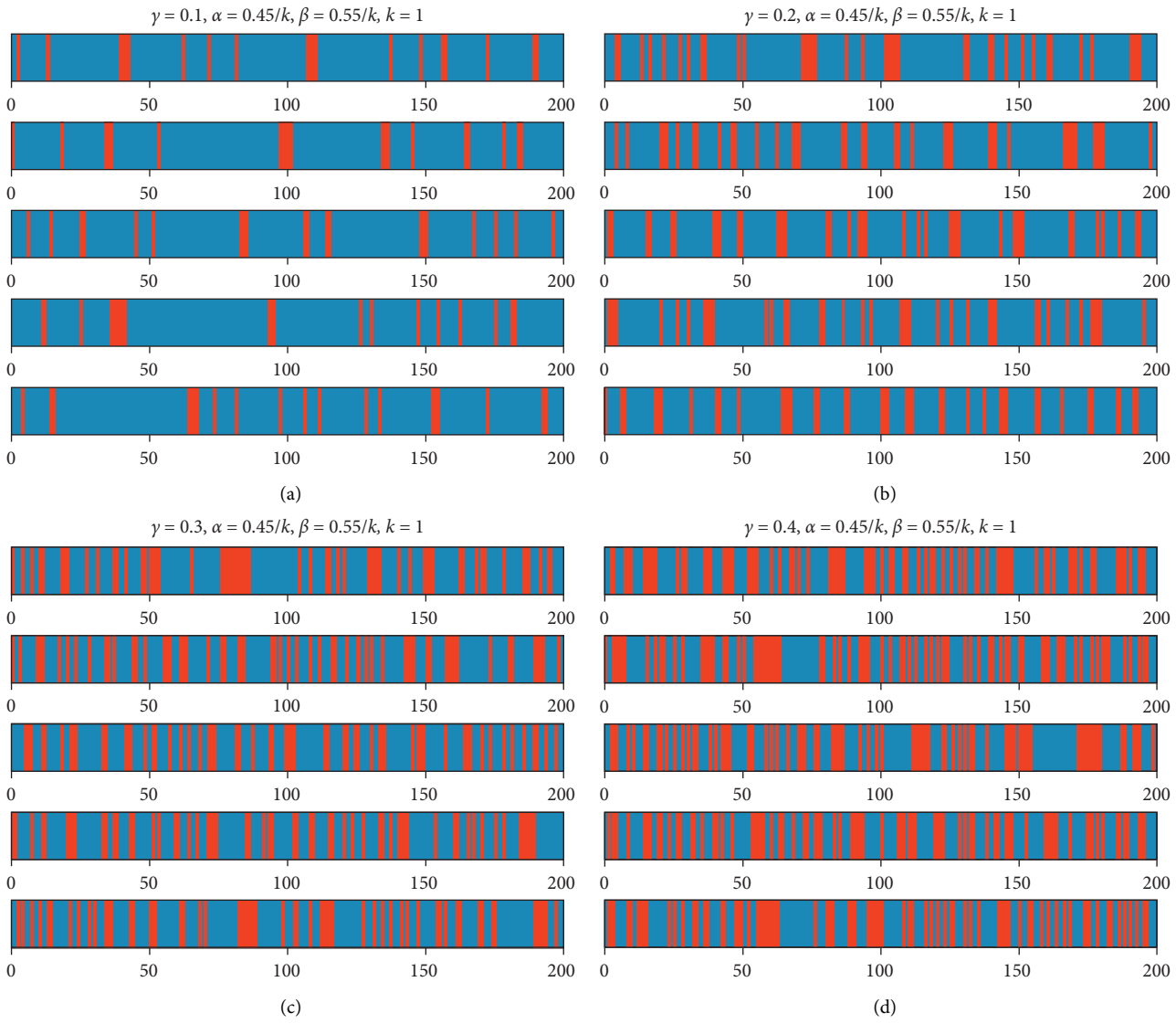


Figure 18: The missing masks in different missing rate γ.

where α and β are length parameters of missing segments which determine the average length of missing segments ALMS:

$$\text{ALMS} = \frac{2}{\alpha + \beta}. \tag{31}$$

Randomly generate an integer between NMDα and NMDβ as NMS and then stochastically divide NMD as NMS segments; then, according to the proportion of each NMS segment, divide M as NMS subsegments, too. Finally, within every subsegment in M, we independently select a missing segment and set those bits inside the subsegments as 1 and

Figure 19: The missing masks in different α and β (ALMS).

Table 2: Dataset configurations for missing detection.

#	Range	γ	SNR
1	(−500, 0)	{5%, 10%, 20%}	(15 dB, 40 dB)
2	(−250, 250)	{5%, 10%, 20%}	(15 dB, 40 dB)
3	(0, 500)	{5%, 10%, 20%}	(15 dB, 40 dB)

Table 3: Dataset configurations for missing recovery.

#	p	γ	(α, β)
1	0	{5%, 10%, 20%}	{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)}
2	7	{5%, 10%, 20%}	{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)}
3	9	{5%, 10%, 20%}	{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)}
4	11	{5%, 10%, 20%}	{(0.1, 0.15), (0.05, 0.075), (0.025, 0.0375)}

others as 0. This pipeline ensures γ and the distribution of missing segments are independent.

Some of the generative missing masks with given γ, α, and β are shown in Figures 18 and 19 . For each set of parameters, four missing masks are generated independently, and the total number of data is $n = 200$; the blue regions represent normal data with $m_i = 0$, while red regions mean missing data with $m_i = 1$.

4.1.2. Dataset Configurations. As mentioned, we will take the load data of the Belgium grid during 2014–2020 as an example, which involves 2000 days in total and 96 sampling points per day. The original data values range from 7000 MW to 14000 MW.

To better evaluate the proposed model, the detection and recovery components will be tested independently, which means the missing mask for the recovery component is the generative missing mask instead of the detected missing mask by the detection component. Hence, different datasets configuration will be used for the two components.

For the detection component, we assume noise W obeys the Gaussian distribution $N(0, \sigma_N^2)$ and define SNR for the normal data signal power P_{data} and noise signal power P_{noise} as

$$\text{SNR} = 10\lg\frac{P_{\text{data}}}{P_{\text{noise}}}, \tag{32}$$

where $P_{\text{data}} = 1/n\|X\|_2^2$ and $P_{\text{noise}} = \sigma_N^2$.

In the experiment, we consider SNR ranges from 15 dB to 40 dB. Since the original data range from about 7000 to 14000, if we directly generate a missing mask on that, nearly all the missing segments will be typical missing segments. To comprehensively evaluate the missing detection algorithm, we linearly map the original data values into (-500, 0), (-250, 250), and (0, 500), respectively. The factor of missing rate γ should be investigated as well, which is set as 10%, 20%, and 40% separately. Besides, length parameters are fixed as $\alpha = 0.1$ and $\beta = 0.15$, then ALMS = 8. The configurations are shown in Table 2. The indexes to assess the detection results are precision P, recall R, and F_1 score in Section 3.1.2.

Table 4: Specifications of the software.

Item	Version
Python	3.7.6
TensorFlow	1.14

Table 5: Specifications of hardware.

Item	Specifications
System	Ubuntu 19.04
CPU	i9-9820x
GPU	Nvidia Titan Xp 12G ×4
RAM	64G

For the recovery component, because the noise will be cleared as zero in Section 3.1.2, we do not care about the influence of noise W. Instead, parameters γ, α, β, and p are studied. γ will be set as 5%, 10%, and 20%, while α and β will be set as (0.1, 0.15), (0.05, 0.075), and (0.025, 0.0375), corresponding to ALMS of 8, 16, and 32. Furthermore, padding depth p will be set as 0, 7, 9, and 11. For each p, the missing mask contains mixed 3×3 combinations of (γ, α, β). The configurations are listed in Table 3. The index to assess the recovery results is RMSE in Section 3.3.2.

In addition, the missing masks with the above configurations will be generated for ten times independently to avoid the stochastic disturbance in results.

4.1.3. Training Settings. For each set of configurations in Table 3, 80% will be used as training sets and the remaining 20% will be testing sets. The specifications of software and hardware are presented in Tables 4 and 5 . The optimizer is "Adam," the learning rate is set to descend exponentially along the epochs, and the batch size is 20 (slices of $S_{\widehat{X'}}(t)$ with corresponding missing masks).

4.2. Results and Discussion of Missing Detection. As illustrated in Figure 20, for the data mapped in the positive range (0, 500) or the negative range (−500, 0), all the precision, recall, and F_1 are all nearly 100% when the SNR is over 20 dB, while the SNR needs to be more than 30 dB to reach the same result for the data mapped in (−250, 250). This is

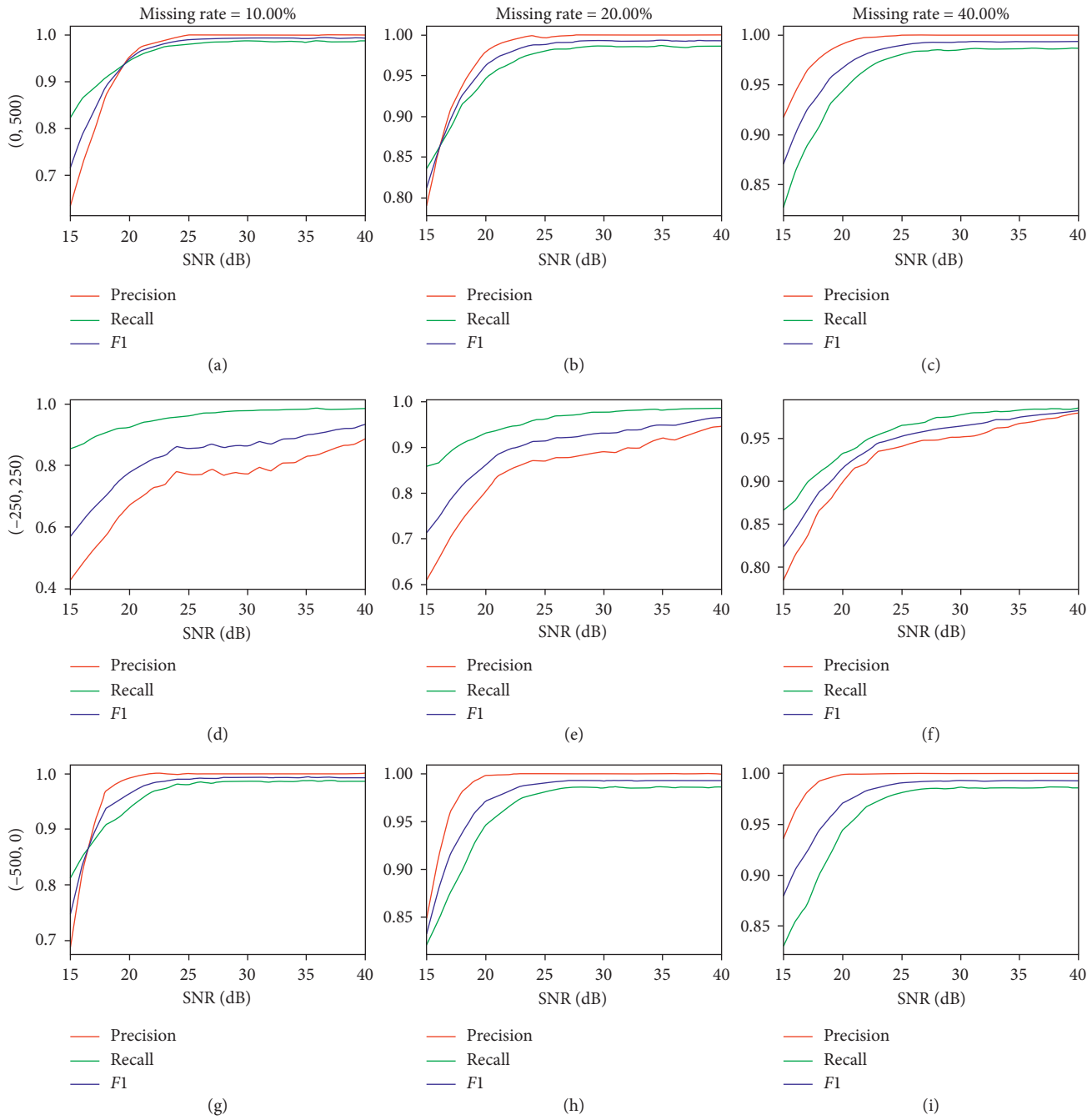


Figure 20: Missing detection results for different missing rates and normalization ranges.

because the ratio of the atypical missing segments for data mapped in (−250, 250) is much higher than the other two, and the noise on the missing segments in that situation will have a very high possibility to be overlapped with the normal data. Moreover, we might be unable to obtain enough typical missing segments for noise estimation, which influences the detection performance.

When the SNR decreases from 20 dB to 15 dB, there is significant deterioration in the results. F_1 of data mapped in (−500, 0) and (0, 500) can drop to 0.7. And it will be even worse for the data mapped in (−250, 250), where F_1 can be lower than 0.6. But fortunately, in most cases, the data are always positive or negative, which is far away from zero, and the noise on missing data is slight enough which ensures high SNR. Thus, the detected mask could be regarded as the ground truth missing masks. This is also

Table 6: RMSE of CCAE for different padding depths.

#	p	CCAE ($\times 10^{-2}$)
1	0	4.15
2	7	3.94
3	9	3.79
4	11	3.84

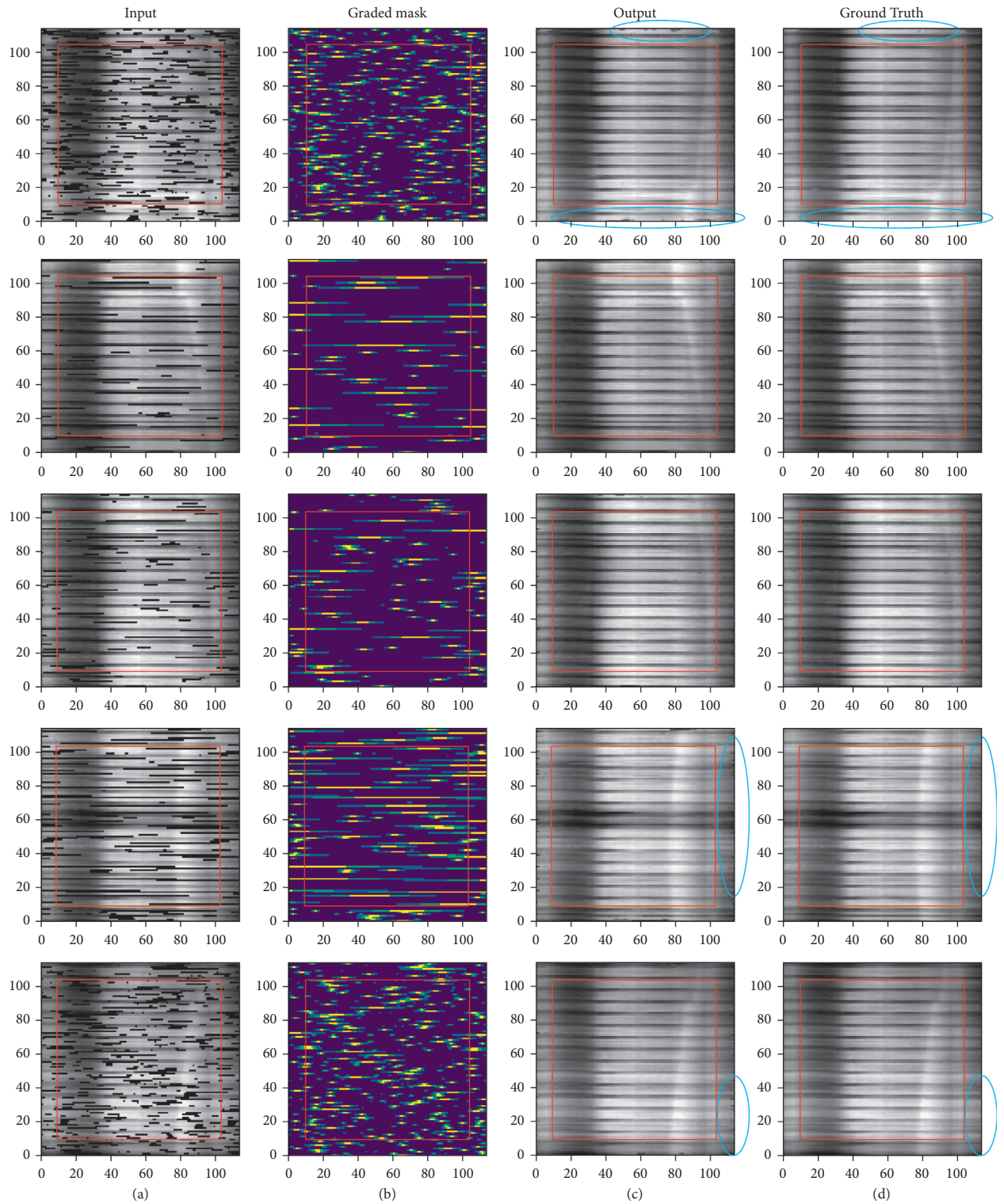
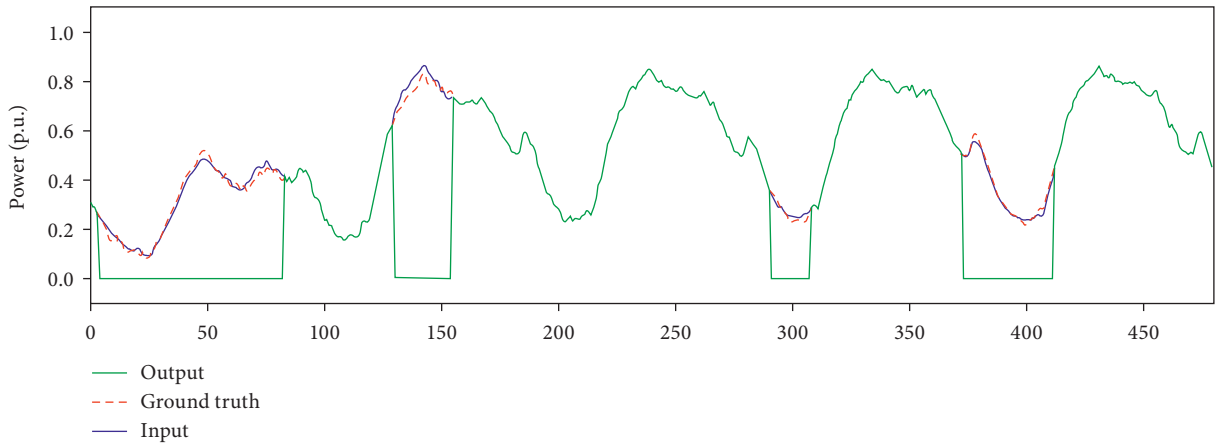


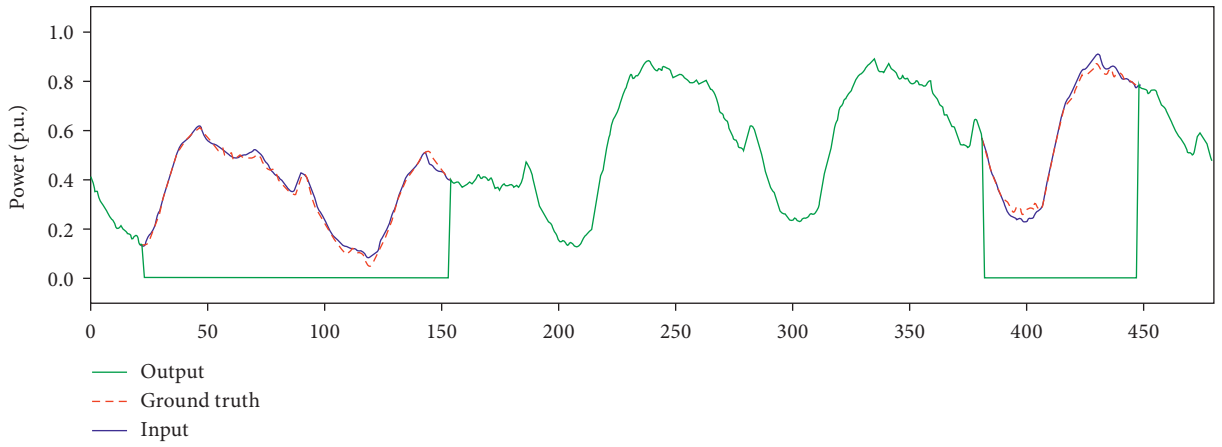
Figure 21: Missing recovery results with core areas inside red rectangles.

the reason for just using the generative missing mask rather than the detected missing mask to test the sequential recovery component.

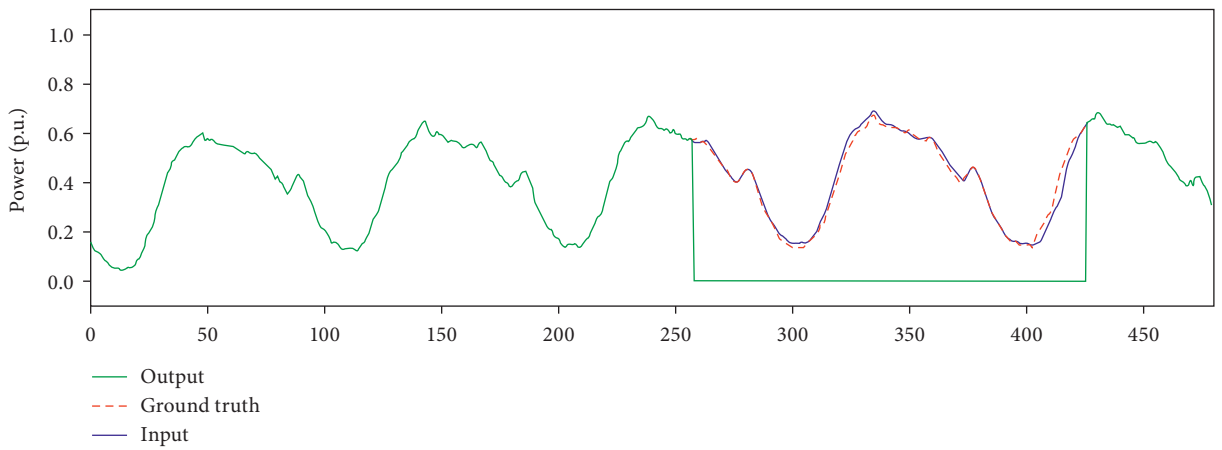
Another phenomenon is that, in the high SNR region, the missing rate seemingly makes no difference to the detection, while in the low SNR region, the higher the missing rate is, the better the detection performs. It may be not so intuitive. But our further analysis indicates that when the SNR is low, the ratio of the false-positive samples will increase greatly because of the overlap of normal data and noise, even when the missing rate is zero. Therefore, a higher missing rate will bring more positive



(a)

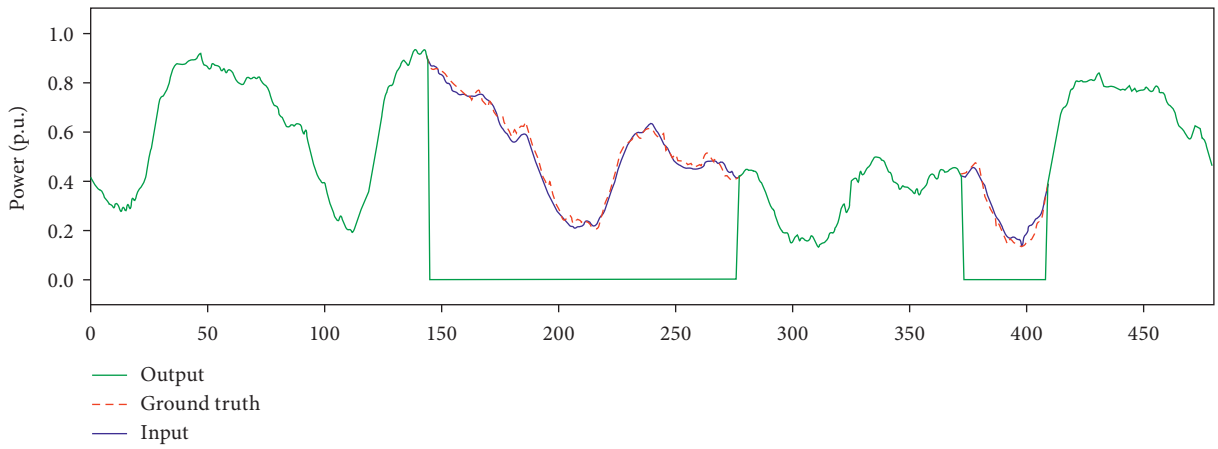


(b)



(c)

Figure 22: Continued.



(d)

Figure 22: Missing recovery results in one-dimensional (96 points/day).

samples in turn, which decreases the false-positive samples to some degree.

4.3. Results and Discussion of Missing Recovery. The RMSE for different padding depths is shown in Table 6. Generally, the CCAE model performs well on the missing recovery problem, and RMSE is pretty low for the normalized range (0, 1). Besides, the edge padding technique can further improve the recovery error, but too deep padding depth may lead to a drop in the performance, and in this experiment, the optimal choice is 9 with the padding ratio of 9.375%.

During training, CCAE is allowed to output high error on those padding areas, which makes the loss function converge more easily and efficiently. And without the padding areas, those errors would appear in the core areas unavoidably, which is also the reason we design the edge padding technique.

Theoretically, the deeper the padding depth is, the more the adjacent data can be used for the edges data in the core area, but with the increase in the depth, the distance between the padding data and core data will grow linearly, too. And once beyond a certain distance, the padding data can no longer provide any useful information, which instead causes a lower proportion of the core area in the padding slice and brings low computation efficiency.

To demonstrate the recovery performance of CCAE with $p = 9$, we randomly chose some output slices of $S_{\widehat{X}_{\text{rec}}'}$ compared with the input slices $S_{\widehat{X}'}$ and the ground truth $S_{\widehat{X}}$ in Figure 21, where those matrices are virtualized by grayscale images. The size of each slice is 114-by-114, and the blue, green, and yellow segments in submissing masks represent the submissing segments in G_1, G_2, and G_3, respectively. The areas inside red rectangles are core areas.

The core areas of the output images are nearly the same as core areas of the ground truth, even for those inputs with high missing rate and long missing segments. We even cannot tell the difference between the cores of output and ground truth. Only when zooming up the output, we can find some slight difference in the textures for the ground truth.

While in the edge padding area outside the core areas, there are obvious black holes as the blue circles shown in Figure 21 because of ignoring those areas when defining the loss function in Section 3.3.2. In Figure 22, we restore some rows of the output to one-dimensional to get $\widehat{Y}_{\text{rec}}$, and the results are consistent with the previous discussion.

5. Conclusion and Future Research

This paper proposes a missing load data detection and recovery model based on CCAE. In the detection issue, we combine ADS and the linear correlation as a criterion to detect the potential missing segments. And based on the detection results, we further divide the detected missing mask into submissing masks with priority and then reshape the original one-dimensional data and mask into two-dimensional matrices for data enhancement. The constructed matrices are regarded as "generalized" images, which transform the recovery problem to images inpainting. Furthermore, the deep learning technologies are conducted, and we have designed a CCAE model to repair the input damaged matrices. To assess the algorithms, we build a missing mask generation model to generate missing masks. Numerical results on the load data of the Belgium grid indicate that the developed detection and recovery algorithms have satisfactory performance under different missing situations. It should be highlighted that the proposed intelligent detection and recovery solution can be used for other forms of time-series dataset.

Here, the strength of the proposed detection and recovery algorithms can be summarized as follows: it can be found that the missing detection is nearly 100% accurate for most situations; the missing segments can be recovered grade by grade with priority in submissing masks strategy, which ensures the recovery accuracy even for long-missing segments. Also, the reshaping from one-dimensional time series to the two-dimensional image is a powerful data enhancement method for the load data, which enables the

CNN to understand the semantics of one-dimensional data. Finally, the structure of CCAE is not sensitive to the input size, so it is easy to make transfer learning to datasets with different periods.

On the contrary, the proposed solution is still needed for further investigation as it has the following potential limitations: firstly, under the condition of low SNR, some of the normal data distributed around zero may be wrongly labeled as missing data. The training process requires a large amount of historical data, which is difficult for some problems. Also, the number of hyperparameters is too many to be optimized and demands for the expert experience.

In future work, further research effort is required to further improving the proposed algorithmic solution from two aspects. Firstly, the models can be further enhanced through the adoption of more sophisticated deep learning models. Also, the solution can be incorporated with a hybrid model that consists of multiple different machine learning algorithms. In addition, the proposed solution can be applied and validated for different time-series data in other application domains.

Data Availability

The Belgium load data used to support the findings of this study are available at http://www.elia.be.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Project of State Grid Zhejiang Electric Power Co., Ltd. and Fundamental Research Funds for the Central Universities (Zhejiang University NGICS Platform).

References

[1] S. Chattopadhyay, M. Mitra, and S. Sengupta, *Electric Power Quality*, Springer, Dordrecht, The Netherlands, 2011.

[2] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2016.

[3] J. Sun, H. Liao, and B. R. Upadhyaya, "A robust functional-data-analysis method for data recovery in multichannel sensor systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1420–1431, 2014.

[4] J. Stones and A. Collinson, "Power quality," *Power Engineering Journal*, vol. 15, no. 2, pp. 58–64, 2001.

[5] L. Liu, D. Zhai, and X. Jiang, "Current situation and development of the methods on bad-data detection and identification of power system," *Power System Protection and Control*, vol. 38, no. 5, pp. 143–147, 2010.

[6] S.-J. Huang and J.-M. Lin, "Enhancement of anomalous data mining in power system predicting-aided state estimation," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 610–619, 2004.

[7] R. Baldick, K. A. Clements, Z. Pinjo-Dzigal, and P. W. Davis, "Implementing nonquadratic objective functions for state estimation and bad data rejection," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 376–382, 1997.

[8] M. Yang, L. Meng, D. Li et al., "Identification of abnormal data of photovoltaic power based on class 3σ," *Renewable Energy Resources*, vol. 36, no. 10, pp. 1443–1448, 2018.

[9] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, and N. Sharma, "Detection of spatial outlier by using improved Z-score test," in *Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 788–790, Tirunelveli, India, 2019.

[10] H. Yang, Y. Wang, S. Fotso Tagne, and Q. Huang, "Abnormal pre-detection algorithm based on time series," in *Proceedings of the 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 262–265, Newark, CA, USA, 2019.

[11] Y. Huang, J. Xiao, Y. Li et al., "A new method to detect and identify bad data based on correlativity of measured data in power system," *Power System Technology*, vol. 30, no. 2, pp. 70–74, 2006.

[12] M. Sohail Ibrahim, W. Dong, and Q. Yang, "Machine learning driven smart electric power systems: current trends and new perspectives," *Applied Energy*, vol. 272, 2020.

[13] S. Wang, H. Chen, and Z. Pan, "A reconstruction method for missing data in power system measurement using an improved generative adversarial network," *Proceedings of the CSEE*, vol. 39, no. 1, pp. 56–64, 2019.

[14] M. Yang, Y. Sun, G. Mu et al., "Data completing of missing wind power data based on adaptive neuro-fuzzy inference system," *Automation of Electric Power Systems*, vol. 38, no. 19, pp. 16–21, 2014.

[15] S. Zhao and C. Wang, "Research and appreciation of the intelligent recovery of missing data in power system measurement," *Science and Technology Innovation Herald*, vol. 15, no. 18, pp. 96–98, 2018.

[16] S. Quan and Y. Cao, "Interpretation method research application," *Science & Technology Information*, vol. 36, pp. 413-414, 2007.

[17] P. Shi and L. Zhang, "A missing data complement method based on K-means clustering analysis," in *Proceedings of the IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–5, Beijing, China, 2017.

[18] V. Miranda, J. Krstulovic, H. Keko, C. Moreira, and J. Pereira, "Reconstructing missing data in state estimation with autoencoders," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 604–611, 2012.

[19] C. T. Concepción, S. L. Fernando, C. R. José et al., "A new missing data imputation algorithm applied to electrical data loggers," *Sensors*, vol. 15, no. 12, pp. 31069–31082, 2015.

[20] C. K. Enders, *Applied Missing Data Analysis*, Guilford Press, New York, NY, USA, 2010.

[21] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, Wiley, New York, NY, USA, 1958.

[22] L. Li, Z. Wen, and Z. Wang, "Outlier detection and correction during the process of groundwater lever monitoring base on pauta criterion with self-learning and smooth processing," in *Proceedings of the Asian Simulation Conference SCS Autumn Simulation Multi-Conference*, Berlin, Germany, 2016.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1 (NIPS'12)*,

pp. 1097–1105, Curran Associates Inc., Red Hook, NY, USA, 2012.

[24] B. L. Yoon, "Artificial neural network technology," *ACM SIGSMALL/PC Notes*, vol. 15, no. 3, pp. 3–16, 1989.

[25] M. Tropea and G. Fedele, "Classifiers comparison for convolutional neural networks (CNNs) in image classification," in *Proceedings of the 23rd IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT '19)*, pp. 310–313, IEEE Press, New York, NY, USA, 2012.

[26] K. Yanai, R. Tanno, and K. Okamoto, "Efficient mobile implementation of a CNN-based object recognition system," in *Proceedings of the 24th ACM International Conference on Multimedia (MM '16)*, pp. 362–366, Association for Computing Machinery, New York, NY, USA, 2016.

[27] M. Hu, H. Guo, and X. Ji, "Automatic driving of end-to-end convolutional neural network based on mobilenet-V2 migration learning," in *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction (VINCI'2019)*, pp. 1–4, Association for Computing Machinery, New York, NY, USA, 2019.

[28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[29] P. Vincent, H. Larochelle, Y. Bengio et al., "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, Helsinki, Finland, 2008.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] J. Masci, U. Meier, D. Ciresan et al., "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proceedings of the 21st International Conference on Artificial Neural Networks*, pp. 52–59, Espoo, Finland, 2011.

[32] J. Sun, Y. Jin, and M. Dai, "Discussion on testing the mechanism of missing data," *Mathematics in Practice and Theory*, vol. 43, no. 12, pp. 166–173, 2013.

[33] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340-341, pp. 250–261, 2016.

[34] H. Huang, H. Xu, X. Wang, and W. Silamu, "Maximum F1-score discriminative training criterion for automatic mispronunciation detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 787–797, 2015.

[35] T. Yang, Z. He, D. Zhao et al., "FSOM neural network based on the network harmonic measurement missing data repair algorithm," *Power System Technology*, vol. 44, no. 5, pp. 1941–1949, 2020.

[36] D. Pathak, "Context encoders: feature learning by inpainting," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, Las Vegas, NV, USA, 2016.