Hindawi
Scientific Programming
Volume 2020, Article ID 1953461, 16 pages
https://doi.org/10.1155/2020/1953461

Research Article

High-Performance Machine Learning for Large-Scale Data Classification considering Class Imbalance

Yang Liu,[1] Xiang Li,[1] Xianbang Chen,[1] Xi Wang,[2] and Huaqiang Li[1]

[1]*College of Electrical Engineering, Sichuan University, Chengdu 610065, China*
[2]*State Grid Sichuan Economic Research Institute, Chengdu 610041, China*

Correspondence should be addressed to Huaqiang Li; lihuaqiang@scu.edu.cn

Received 23 December 2019; Revised 24 February 2020; Accepted 5 May 2020; Published 18 May 2020

Academic Editor: Rahman Ali

Currently, data classification is one of the most important ways to analysis data. However, along with the development of data collection, transmission, and storage technologies, the scale of the data has been sharply increased. Additionally, due to multiple classes and imbalanced data distribution in the dataset, the class imbalance issue is also gradually highlighted. The traditional machine learning algorithms lack of abilities for handling the aforementioned issues so that the classification efficiency and precision may be significantly impacted. Therefore, this paper presents an improved artificial neural network in enabling the high-performance classification for the imbalanced large volume data. Firstly, the Borderline-SMOTE (synthetic minority over-sampling technique) algorithm is employed to balance the training dataset, which potentially aims at improving the training of the back propagation neural network (BPNN), and then, zero-mean, batch-normalization, and rectified linear unit (ReLU) are further employed to optimize the input layer and hidden layers of BPNN. At last, the ensemble learning-based parallelization of the improved BPNN is implemented using the Hadoop framework. Positive conclusions can be summarized according to the experimental results. Benefitting from Borderline-SMOTE, the imbalanced training dataset can be balanced, which improves the training performance and the classification accuracy. The improvements for the input layer and hidden layer also enhance the training performances in terms of convergence. The parallelization and the ensemble learning techniques enable BPNN to implement the high-performance large-scale data classification. The experimental results show the effectiveness of the presented classification algorithm.

1. Introduction

Classification is one of the most effective approaches in enabling the analysis of the digital data in quite a number of academia and research fields, for example, the medical researches [1–6] and the power system researches [7–12]. Multiple applications including image processing, pattern recognition, and pattern matching benefit from the accurate and efficient classification algorithms [1–6]. Li et al. [13] implemented the medical image classification using convolutional neural network (CNN). Their classification strategy can automatically and efficiently learn the graphical characteristics of the interstitial lung disease. As a result, the strategy is able to supply accurate and efficient classification performances. Jiao et al. [14] also employed and improved CNN to classify the mental load data. The experimental results show the effectiveness of their classification algorithm. Yang and Shen [15] reviewed the load classification researches of the electrical power system. The paper pointed out that classification is an effective way of researching the complicated load behaviors, which significantly affects the power production and consumption. In order to implement the natural language processing (NLP), Zhang et al. [16] employed deep learning as the underlying methodology to execute the text classification. Their work is able to identify the semantic labels of the texts successfully.

In the recent years, a number of classification algorithm researches have been presented, which are mainly based on k-means [17], fuzzy c-means (FCM) [18], neural networks (NNs) [19], support vector machine (SVM) [20], and so on.

For example, Peng et al. [7] employed a number of clustering algorithms containing k-means, k-medoids, self-organizing maps (SOM) [21], and FCM to recognize the patterns of the electrical load data. Xu et al. [8] improved and optimized k-means algorithm using canopy algorithm. The authors claimed that the clustering accuracy and efficiency of their algorithm have been significantly enhanced. Niazmardi et al. [22] presented an improved FCM algorithm which can achieve accurate results for clustering the hyperspectral data. However, quite a number of researches [23–25] pointed out that the flaws, for example, the sensitivity of outliers, difficulty of clustering nonlinear-separable classes, and empirical values-based parameters, existing in the abovementioned unsupervised machine learning algorithms prevent them from being effectively used. As a result, the supervised machine learning algorithms, for example, the neural networks, have been widely employed in the classification researches [26]. Gu et al. [27] optimized the learning rate and inertial factor for the traditional BPNN. Their improved self-adaptive BPNN algorithm shows great performances in terms of data modeling. Li et al. [9] combined BPNN and FCM to implement the electrical load forecasting. The authors reported that the load forecasting accuracy can be significantly improved. Based on the conclusions of the researches [10,11,23–25], BPNN has been proved that it is quite suitable for the classification tasks. Although BPNN has a number of advantages, it still has flaws, for example, the slow convergence issue in its training caused by the sensitivity of initial weights, sensitivity of learning rate, gradient exploding, and gradient vanishing. However, several researches [28–33] suggested that batch-normalization [28] and ReLU [34] have great potentials to solve these issues.

Additionally, current digital data collection benefits from the developments of the smart meters, data communication systems, and data storage technologies, which results in sharply increasing of the data scale and data dimension. Tang et al. [35] pointed out that to overcome the issue, machine learning algorithms need focus on dimension reduction and sample selection techniques to improve the algorithm efficiencies. Farrell et al. [36] also proved that their improvements of the maximum entropy principle, random forest, and SVM can achieve satisfied performances for processing the high-dimensional data. Xu et al. [37] claimed that the data labeling and training efficiency are two main challenges for the large-scale data classification. Therefore, they presented a k-means and SVM-based strategy to implement the large-scale data classification. Their experimental results show that based on their strategy, the size of the training dataset can be reduced with maintaining the classification accuracy. Liu et al. [23] further pointed out that BPNN encounters extremely low efficiency when it classifies the large-scale and high-dimension data. The research also stated that the distributed computing could be a suitable solution to overcome the low-efficiency issue. Su et al. [12] employed the Hadoop framework [38] to implement the efficiency improvement for BPNN. The authors reported that their solution achieves satisfied precision for the data forecasting. Liu et al. [25] also presented a Spark-based distributed BPNN algorithm [39] which shows remarkable efficiency for classifying large-scale data. However, Liu et al. [23] pointed out that the algorithm decoupling-based parallelization of BPNN may generate a great number of iterations, which deteriorates the processing efficiency. The authors also claimed that the data separation-based parallelization may reduce the final classification accuracy. Therefore, to figure out a parallelized BPNN with high efficiency and accuracy is a valuable work.

It also should be noticed that due to multiple classes and uneven data distribution, the class imbalance issue [40] frequently exists in the training data. The issue could significantly affect the training effect, which further impacts the final classification accuracy. Zhang et al. [41] reported that the image recognition ability of the deep CNN declines in the case of unbalanced training data. Therefore, the authors presented a classification method which recognizes a query image by comparing distances between category centers of CNN features of the whole training dataset and the corresponding CNN feature of the query image. Li et al. [42] also pointed out that although CNN supplies very high performances, it is still suffering from the problem of class imbalance. By adding an extra class-imbalance aware regularization, the authors presented a new loss function enabling CNN more sensitive to the samples of the minority classes. Zhang et al. [43] also claimed that the imbalanced class issue leads to significant misclassification of deep belief network (DBN). To address the issue, the authors unequalize the misclassification costs between classes, and then the costs are applied to DBN to achieve the accurate classification. Although the efforts contributed by the abovementioned researches are able to solve the class imbalance issue, Han et al. [40] suggested that the sampling technique should be an effective way of rebuilding the samples for the minority class and further solves the class imbalance issue efficiently.

As a result, this paper presents a BPNN-based high-performance large-scale data classification method considering class imbalance. The paper firstly improves the Borderline-SMOTE [40] algorithm using Fréchet distance [44] to solve the potential class imbalance issue in the training data. Secondly, in order to solve the slow convergence issue in the training phase of BPNN, zero-mean [45], batch-normalization, and ReLU are employed to improve the input layer, hidden layer, and activation function. Thirdly, this paper also presents a MapReduce-based parallelization method for the improved BPNN. Based on the data separation and ensemble learning techniques, the parallelization of the improved BPNN can be implemented.

The rest of the paper is organized as follows: Section 2 presents the details of the methodologies for the BPNN improvements; Section 3 shows the experimental results; Section 4 concludes the paper.

2. Class Balance-Based Improved BPNN in Enabling Large-Scale Classification

This section firstly presents the Fréchet distance-based Borderline-SMOTE which is able to solve the class imbalance issue in the training dataset, and then, the section presents the details of BPNN improvement using zero-

mean, batch-normalization, and ReLU. Finally, the section presents the ensemble learning-based parallelization for the improved BPNN using the Hadoop framework.

2.1. Fréchet Distance and Borderline-SMOTE-Based Class Balance. Class imbalance issue may significantly impact the training of BPNN, which finally leads to the misclassification. Borderline-SMOTE [40] is proved to be an effective solution for balancing the classes in the training dataset. However, Borderline-SMOTE measures the similarity between data instances using Euclidean distance which lacks the ability for representing the shape features and sequential characteristics of the data instances. Therefore, this paper employs Fréchet distance [44] instead of Euclidean distance to measure the similarity between the data instances.

2.1.1. Fréchet Distance Computation for Two Data Instances. Let R^2 denote the metric space; $A, B: [0, 1] \longrightarrow R^2$ represent two continuous curves in the Fréchet space [46]; and $\alpha, \beta: [0, 1] \longrightarrow [0, 1]$ denote two continuous nondecreasing functions with independent variable t in the unit interval. Therefore, the Fréchet distance $\delta_{\mathrm{dF}}(A, B)$ is defined as the following equation:

$$\delta_{\mathrm{dF}}(A, B) = \inf_{t\in[0,1]} \max \|A(\alpha(t)) - B(\beta(t))\|_P, \quad (1)$$

where $||.||_P$ represents the Euclidean norm; inf represents the infimum of the set; $\alpha(0) = \beta(0) = 0$, $\alpha(1) = \beta(1) = 1$ [46].

The parameter t is continuous which cannot adapt to the computation for the discrete parameters. Therefore, the discrete Fréchet distance is presented by researches [47,48]. Let $P = (p_1, p_2, ..., p_u)$ and $Q = (q_1, q_2, \ldots, q_v)$ denote two discrete curves, and k denote the total number of the Fréchet permutations [47,48]; therefore, for one Fréchet permutation $W_j = \{(P_i, Q_i)\}$, $1 \le I \le u$, $1 \le j \le k$, the max value of the max distances in W_j can be represented by $d_F^{W_j}(P, Q) = \max_i \max_{(p,q)\in P_i\times Q_i} \mathrm{dis}(p, q)$. As a result, the discrete Fréchet distance for P and Q can be represented by the following equation:

$$d_F(P, Q) = \min_W d_F^W(P, Q). \quad (2)$$

Fréchet distance has better ability to represent the shape features and sequential characteristics of the data instances [44]. Therefore, it is employed by this paper to compute the nearest neighbors for the Borderline-SMOTE algorithm.

2.1.2. Borderline-SMOTE in Enabling Class Balance. Borderline-SMOTE is able to balance the classes for the imbalanced dataset [40]. It balances the dataset based on two remarkable advantages including the identification of the border between the major and minority classes, and the sample synthesis around the border.

Step 1. In one dataset T, for each point $p_i (i = 1, ..., \mathrm{pnum})$ in the minority class P, compute a set of m nearest neighbors using Fréchet distance (equation (2)). Inside the set, the number of the points belonged to the major class is m' $(0 \le m' \le m)$.

Step 2. If m' equals m which indicates the m nearest neighbors are majority examples, p_i is regarded as noise and neglected. Otherwise, if $0 \le m' \le m/2$, p_i has less chance to be misclassified, which does not need to be further processed. If $m/2 \le m' \le m$, p_i has higher chance to be misclassified. Therefore, for all p_i in the minority class, a borderline set $E = \{p_1', p_2', ..., p_{\mathrm{dnum}}'\}$, $0 \le \mathrm{dnum} \le \mathrm{pnum}$ can be achieved, where p_i' denotes a nearest neighbor of p_i.

Step 3. For each p_i' in E, compute a number of k nearest neighbors from the minority class P, and then a number of s points are randomly selected from the k neighbors. As a result, let $r_j (j = 1, 2..., s)$ denote a random value between 0 and 1, and p_j' denote one of the s neighbors. Therefore, a new instance synthetic$_j = p_i' + r_j \times (p_i' - p_j')$ can be ultimately synthesized.

A demonstration performed by Fréchet distance-based Borderline-SMOTE is shown in Figure 1. Firstly, the algorithm is able to identify the border between two imbalance classes. Secondly, new instances can be synthesized to balance the classes and further highlight the border to distinguish two classes. Therefore, if class imbalance issue exists in the training dataset of BPNN, the Fréchet distance-based Borderline-SMOTE has great potential to improve the training performance. However, it also should be noted that the values of two parameters k and m affect the performance of Borderline-SMOTE. Therefore, in the later algorithm evaluation sections, the optimal values of k and m are selected from a series of preexecuted experiments.

2.2. Improved BPNN Using Zero-Mean, Batch-Normalization, and ReLU. As an effective classification algorithm, BPNN has been successfully employed in quite a number of researches [10,11,26,49]. However, issues, for example, the sensitivity of initial weights, sensitivity of learning rate, gradient exploding, and gradient vanishing, are still needed to be handled. Therefore, this paper firstly employs zero-mean to normalize the input data instances. Secondly, batch-normalization and ReLU activation function are employed to overcome the convergence issues in the training phase.

2.2.1. Brief Introduction of the Standard BPNN. Figure 2 shows the architecture of a typical BPNN, which contains the input layer, several hidden layers, and the output layer. In the hidden and output layers, a number of neurons exist, respectively. In Figure 2, $x_1, \ldots, x_n$ represent the input of BPNN; w_{ij}, b_i, and g represent the weight, bias, and activation function of a neuron; a_i represents the output of a neuron in the hidden layer; y represents the output of BPNN ($a = y$ in the output layer). To facilitate the presentation, the aforementioned parameters are represented in the compact form of matrices X, W, b, A, and Y. In terms of the network training, two phases including feed forward and back propagation are employed. Basically, feed forward computes the input X using each layer of the network to achieve the

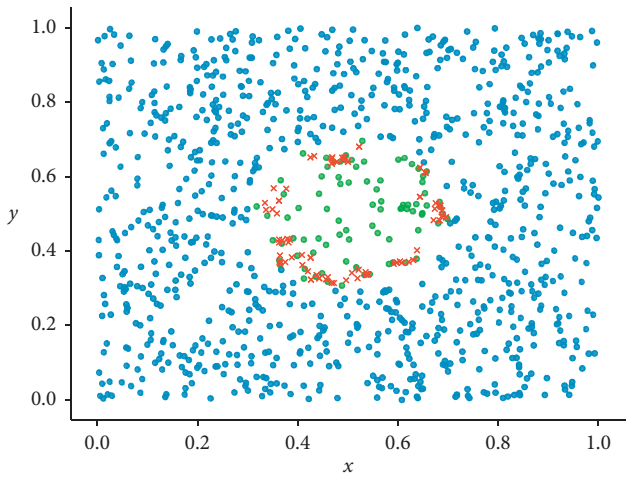


Figure 1: Borderline-SMOTE-based data instance synthesis (• majority class; • minority class; × synthetic data).

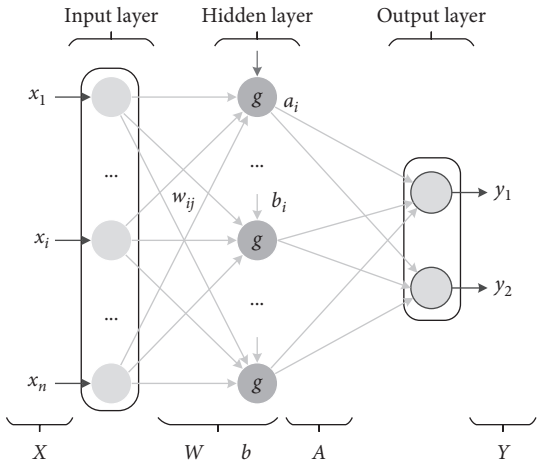


Figure 2: The architecture of a typical BPNN.

output Y according to $Y = g(W^T X + b)$. Back propagation employs the loss function *loss* to compute the variance J between the output Y and the actual value $Y\_$ according to $J = \text{loss}(Y, Y\_)$. The neural network chooses a proper optimization algorithm, for example, the stochastic gradient descent (SGD) [29] algorithm, to update W and b using $W = W - \alpha(\partial J/\partial W)$, $b = b - \alpha(\partial J/\partial b)$, where α is the learning rate [50].

In terms of classification using BPNN, firstly the training phase should be carried out. Let $\text{instance}_i = [a_1, a_2, \ldots, a_n]$ indicate the i^{th} data instance in the training dataset; a denote one feature of instance_i; and c_j denote the class that instance_i is belonged to. Firstly, each feature of instance_i is normalized. Secondly, BPNN inputs instance_i to run the feed forward to compute the output Y, and then, the coded c_j is regarded as the actual value $Y\_$ to run the back propagation to update the weights and biases. As long as all the training instances in the training dataset have been processed with several epochs and iterations, the training phase terminates. In the classification phase, let instance_k denote the k^{th} testing instance in the testing dataset. BPNN inputs instance_k and computes the output using only feed forward. The output is the class which instance_k is belonged to. As long as all the testing instances in the testing dataset have been processed, the classification phase terminates.

Figure 3 shows the BPNN improvements using zero-mean, batch-normalization, and ReLU. The details of the improvements are presented in the following sections.

2.2.2. Zero-Mean-Based Input Layer Improvement. Zero-mean is a normalization technique which is able to improve the data distribution to accelerate the gradient descent [45]. Let the input matrix X denote a number of batch-size input data instances in each iteration. Processed by the zero-mean layer shown in Figure 3, the zero-centered data instances become the actual input. Let the matrix X_{mean} denote the average value of the number of batch-size input data instances, and the matrix $X_{\text{zero-centered}}$ represent the zero-centered input; therefore, equations (3) and (4) indicate the computations for X_{mean} and $X_{\text{zero-centered}}$:

$$X_{\text{mean}} = \begin{bmatrix} \overline{x}_1 \\ \vdots \\ \overline{x}_{\text{batch-size}} \end{bmatrix}, \tag{3}$$

$$X_{\text{zero-centered}} = X - X_{\text{mean}}. \tag{4}$$

2.2.3. Batch-Normalization-Based Hidden Layer Improvement. Briefly, in the training phase of a multiple-hidden-layers neural network, the data distributions of different layers may vary, which is no longer independent and identically distributed (IID). Therefore, the internal covariate shift (ICS) occurs [28], which causes two main issues impacting the training. The first one is that the variation of the parameters in the current layer may cause the distribution variation of the input data in the last layer. As a result, the last layer has to tune itself to adapt to the distribution variation so that the learning performance is deteriorated. The second one is that the values of W and b may keep enlarging, which leads to larger value of $W \times x + b$ in each layer. Consequently, the gradient saturation of the activation function may occur. Therefore, the value of the updated gradient may be extremely small in the back propagation, which leads to the deterioration of the network convergence.

However, batch-normalization [28] employs the normalization operations to recover the input data distribution for each neuron in the hidden layers to the standard normal distribution $N(0, 1)$. Therefore, the activation function is able to work sensitively. Generally, in batch-normalization, the parameters mean μ and variance σ of the input data enable the normalization. Additionally, the tiny variations of μ and σ in each mini-batch can also improve the generalization of the neural network. The initial scale factor γ and the displacement factor β are adopted to implement the linear transformation. Both of the factors can be trained in batch-normalization. As a result, batch-normalization is capable of dealing with larger initial weights, larger learning rates,

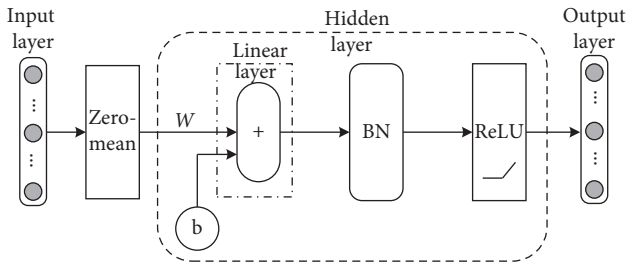


Figure 3: The improvements for the input, hidden, and output layers.

gradient issues, and overfitting issue, which significantly improves the training performance of the neural network [28].

This paper employs batch-normalization to improve the hidden layer of BPNN due to its remarkable advantages. It normalizes the linear output of the hidden layer, and then the output of batch-normalization is input into the nonlinear ReLU. The details of batch-normalization are shown in Algorithm 1.where D represents the output of the linear computation of the hidden layer; y_i denotes the output of batch-normalization, which is the input of the nonlinear ReLU; ε is a constant value to stabilize the training; μ_D and σ_x^2 represent the mean and variance of the input data instances; $\widehat{d}_i$ represents the standardized intermediate data. Similar to the weight and bias of BPNN, γ and β are also updated in the back propagation phase according to $\gamma = \gamma - \alpha(\partial L/\partial y_i)(\partial y_i/\partial \gamma)$ and $\beta = \beta - \alpha(\partial L/\partial y_i)(\partial y_i/\partial \beta)$, where L represents the back propagated loss function of the hidden layer.

2.2.4. ReLU Activation Function. To overcome the saturation of the commonly used *Sigmoid* and *Tanh* activation functions, as well as to improve the convergence of the network, this paper employs ReLU shown by equation (5) as the activation function for the neural network, which is able to handle the gradient issues [29]:

$$\mathrm{ReLU}(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0, \end{cases} \tag{5}$$

where x presents the input of ReLU.

2.3. Ensemble Learning-Based Parallelization of BPNN. In order to implement the large-scale data classification, the Hadoop framework based on MapReduce computing model [51] is employed to parallelize the improved BPNN. This paper firstly separates the entire training dataset into a number of data chunks which are saved in HDFS (Hadoop Distributed File System), and then, each participated mapper initializes one sub-BPNN and inputs one data chunk, respectively. Each mapper trains its own classifier individually so that a number of different classifiers can be finally achieved in parallel. Therefore, for one testing data instance, these classifiers may generate different decisions. Finally, this paper also presents a weighted voting strategy to decide the ultimate classification result for the testing data instance.

2.3.1. Data Separation-Based Parallelization for the Improved BPNN. Although BPNN can be directly decoupled and parallelized in Hadoop, the research [23] suggested that the efficiency of the parallelized algorithm is extremely low due to the imperfect iteration support of the Hadoop framework. Therefore, the parallelization in this paper is based on the data separation. Firstly, the entire training dataset is separated into a number of data chunks using the random sampling. The data chunks are saved in HDFS. However, in each data chunk, the class imbalance issue may be intensified or alleviated due to the random sampling and the separation. Therefore, if the class imbalance exists in one data chunk, Borderline-SMOTE balances the classes, and then, a number of mappers start, each of which individually initializes one improved BPNN as a sub-BPNN and inputs one data chunk to train the parameters of the sub-BPNN. The workflow of each sub-BPNN in a mapper is shown in Figure 4.

Finally, as long as the training for each mapper terminates, multiple different classifiers can be achieved in parallel. In the classification phase, one testing instance is input into all the classifiers. As a result, different classifiers may generate different classification results. One reducer collects all the results from the mappers to form an aggregation, in which the weighted voting is carried out to achieve the final classification result for the testing instance.

2.3.2. Weighted Voting. In order to ensemble the multiple classification results into one final result in the reducer, this paper presents a classification reliability-based weighted voting, which can achieve the final classification result according to the reliabilities of the results from the multiple classifiers.

It is known that for different classes, each classifier has different classification accuracies. Therefore, let C represent the reliability matrix; n represent the number of classes in the training dataset; and $r = [r_1, r_2, ..., r_n]$ represent the classification accuracy for each class of the classifier. Consequently, *softmax* function [52] can be adopted to compute the reliability c_i for the i^{th} class of the classifier using the following equation:

$$c_i = \frac{e^{r_i}}{\sum_{j=1}^{n} e^{r_j}}. \tag{6}$$

Therefore, for a number of m classifiers, the reliability matrix C can be represented by the following equation:

$$C_{m\times n} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}. \tag{7}$$

The output coding of each classifier is based on one-hot encoding [44]. So, the output p of the classifier is the probability distribution of each class p_i ($i = 1, 2, \ldots, n$) indicated by the following equation:

$$p = [p_1, p_2, ..., p_n]. \tag{8}$$

Therefore, the outputs of m classifiers form a probability distribution matrix P shown by the following equation:

Input: a batch of input $D = \{d_1, d_2, ..., d_{\text{batch-size}}\}$
Initial scale and displacement factors: γ, β
Output: $\{y_i = \text{BN}_{\gamma,\beta}(d_i)\}$
$\mu_D = (1/\text{batch} - \text{size}) \sum_{i=1}^{\text{batch-size}} d_i$
$\sigma_D^2 = (1/\text{batch} - \text{size}) \sum_{i=1}^{\text{batch-size}} (d_i - \mu_D)^2$
$\widehat{d}_i = (d_i - \mu_D)/(\sqrt{\sigma_d^2 + \varepsilon})$
$y_i = \gamma \widehat{d}_i + \beta \equiv \text{BN}_{\gamma,\beta}(d_i)$,

Algorithm 1: Batch-normalization.

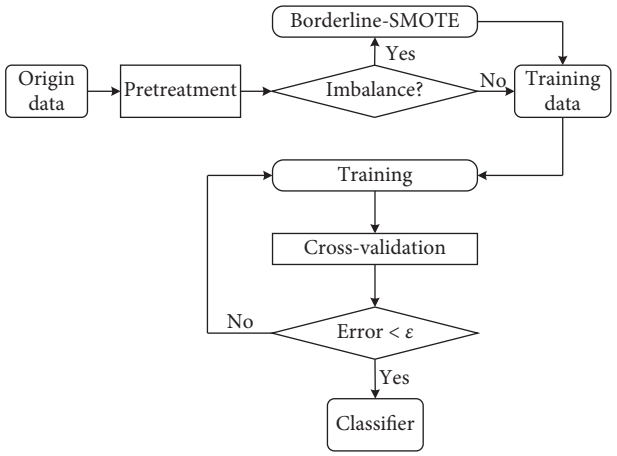


Figure 4: The data balancing and training processes for one sub-BPNN in a mapper.

$$P_{m\times n} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mn} \end{bmatrix}. \tag{9}$$

An intermediate matrix Q shown by equation (10) can be achieved by multiplying the reliability matrix C and the probability distribution matrix P:

$$Q_{m\times n} = \begin{bmatrix} c_{11}p_{11} & \cdots & c_{1n}p_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1}p_{m1} & \cdots & c_{mn}p_{mn} \end{bmatrix}. \tag{10}$$

The sum r_i' of the elements in the i^{th} column in Q finally forms a weight matrix R denoted by the following equation:

$$R_{1\times n} = [r_1', r_2', ..., r_n']. \tag{11}$$

Ultimately, the final classification result class_label for the input data instance can be identified according to the following equation:

$$\text{class\_label} = \arg\max_i R. \tag{12}$$

Figure 5 shows the logical flow of the BPNN parallelization using Hadoop.

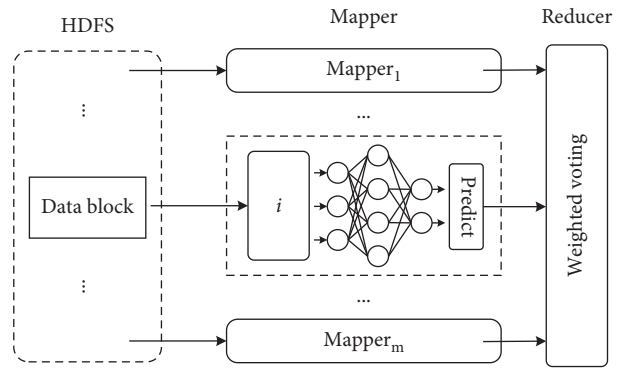


Figure 5: Parallelization of the improved BPNN using MapReduce.

3. Experimental Result

The experiments are organized into three parts. The first part evaluates the performances of the Fréchet distance-based Borderline-SMOTE using the randomly generated linearly inseparable two-dimensional semiannular dataset. The second part evaluates the performances of the improved BPNN in the standalone environment using the Iris dataset [53], Wine dataset [52], and Vehicle Silhouettes dataset [54]. The third part evaluates the performances of the parallelized improved BPNN in the Hadoop distributed computing environment. The details of the experimental environments are listed in Table 1.

3.1. Evaluation for Fréchet Distance-Based Borderline-SMOTE. A randomly generated dataset is employed to evaluate the Fréchet distance-based Borderline-SMOTE. The dataset has two classes, each of which contains 150 data instances, respectively. Based on the original dataset shown in Figure 6, an imbalanced dataset is generated, which is shown in Figure 7. Four class balance algorithms are implemented including Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 5$), random oversampling, random undersampling, and SMOTE. After the class balance for the imbalanced dataset, the improved BPNN carries out the classification using a number of 100 randomly selected training instances and a number of 90 testing instances. The classification results are shown in Table 2.

Table 2 indicates that without the class balance, the classification accuracy is only 66.67%. However, all the class balance algorithms can help to improve the classification

Table 1: Details of the experimental environment.

Standalone environment	Distributed environment
	CPU: Intel Core i5 2.5 GHz
CPU: Intel Core i5-8250U 1.6 GHz	MEM: 16 GB
MEM: 8 GB	OS: Ubuntu 16.04 64 bit
OS: Win 10 64 bit	Hadoop: 2.9.1
Python: 3.6.6	Python: 3.5.2
	Node no.: 4

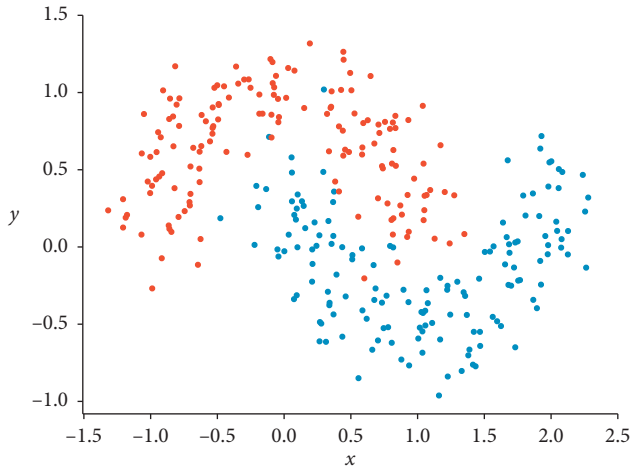


Figure 6: The randomly generated linearly inseparable two-dimensional semiannular dataset with 300 instances.

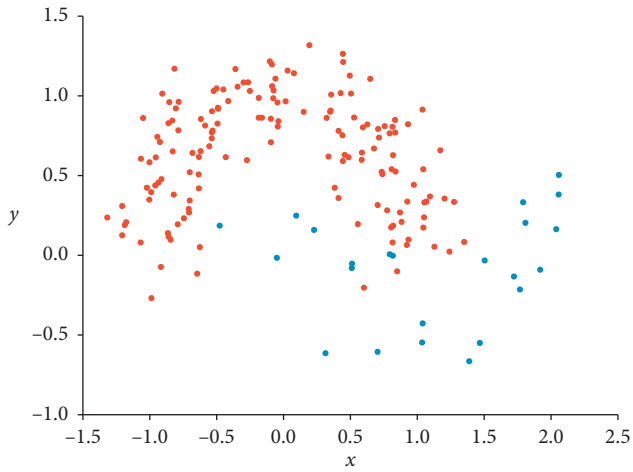


Figure 7: The imbalanced dataset based on the original dataset.

accuracy, among which the Fréchet distance-based Borderline-SMOTE significantly outperforms the other algorithms.

To further evaluate the potential of the Fréchet distance-based Borderline-SMOTE, a number of 20000 instances which averagely belonged to two classes are generated using the random sampling. Firstly, a number of 5000 instances are randomly selected from class 1, and then a number of 1000, 500, 50, and 10 instances are randomly selected from class 2, respectively, to compose 4 imbalanced training datasets. Balanced by the Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 5$), the improved BPNN is trained, and then carries out the classification. The number of the testing instances is 10000. The classification accuracies are listed in Table 3.

Table 2: The comparisons of the classification accuracy for different class balance algorithms.

Class balancing algorithm	Classification accuracy (%)
Borderline-SMOTE	95.556
Random oversampling	78.889
Random undersampling	74.444
SMOTE	86.667
Without balancing	66.667

Table 3 shows that the imbalanced training data significantly impacts the classification accuracy. However, benefitting from the Fréchet distance-based Borderline-SMOTE, the classification accuracy can be improved substantially. Table 3 also indicates that the imbalance ratio can impact the classification accuracy. The slightly imbalanced training dataset achieves better classification performance as long as it is balanced by the Fréchet distance-based Borderline-SMOTE. Contrarily, the extremely imbalanced training dataset leads to low classification accuracy even if it is balanced.

3.2. Evaluation for the Improved BPNN. This section evaluates the performances of the improved BPNN in the standalone environment. Iris dataset, Wine dataset, and Vehicle Silhouettes dataset are employed. The details of the training and testing instances are listed in Table 4. In terms of comparison, SVM, traditional BPNN, and self-adaptive BPNN [55] are also implemented.

Table 5 shows the average classification accuracy based on 50 times experiments for each algorithm using the Iris dataset. The traditional BPNN performs the lowest classification accuracy with the largest epochs. The presented improved BPNN and the self-adaptive BPNN perform the similar classification accuracy. However, in terms of the number of epochs, self-adaptive BPNN slightly outperforms the improved BPNN.

Tables 6 and 7 show the average classification accuracy based on 50 times experiments for each algorithm using the Wine dataset and Vehicle Silhouettes dataset. Firstly, the traditional BPNN also shows the worst performance. Secondly, in the Wine dataset-based experiments, although the self-adaptive BPNN slightly outperforms the improved BPNN in terms of epochs, the classification accuracy of the improved BPNN is higher than that of the self-adaptive BPNN. Thirdly, in the Vehicle Silhouettes dataset-based experiments, the improved BPNN performs the best in terms of accuracy and epochs.

Figures 8(a)–8(c) show the convergences of the traditional BPNN, the improved BPNN, and the self-adaptive BPNN using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. For the simple Iris dataset, three algorithms perform relatively close convergences. However, for the complicated Wine dataset, both the

Table 3: The comparisons of the classification accuracy using Borderline-SMOTE.

Imbalance class ratio	Balanced dataset accuracy (%)	Imbalanced dataset accuracy (%)
5000 : 1000	98.09	73.15
5000 : 500	95.19	68.32
5000 : 50	80.53	56.20
5000 : 10	67.45	53.13

Table 4: The training instances and the testing instances of the datasets.

Dataset	Training instance number	Testing instance number
Iris	120 (class1: 40; class2: 40; class3: 40)	30 (class1: 10; class2: 10; class3: 10)
Wine	124 (class1: 36; class2: 52; class3: 36)	54 (class1: 23; class2: 19; class3: 12)
Vehicle	592 (Class1: 137; Class2: 155; Class3: 159; Class4: 141)	254 (Class1: 62 Class2: 62; Class3: 59; Class4: 71)

Table 5: The accuracy comparisons using the Iris dataset.

Algorithm	Testing dataset accuracy	Epochs
Improved BPNN	96.667	188
Self-adaptive BPNN	96.667	172
Traditional BPNN	93.333	342
SVM	96.667	/

Table 6: The accuracy comparisons using the Wine dataset.

Algorithm	Testing dataset accuracy	Epochs
Improved BPNN	94.370	723
Self-adaptive BPNN	91.444	537
Traditional BPNN	67.317	1531
SVM	96.296	/

Table 7: The accuracy comparisons using the Vehicle Silhouettes dataset.

Algorithm	Testing dataset accuracy (%)	Epochs
Improved BPNN	83.465	245
Self-adaptive BPNN	80.709	465
Traditional BPNN	74.803	993
SVM	44.488	/

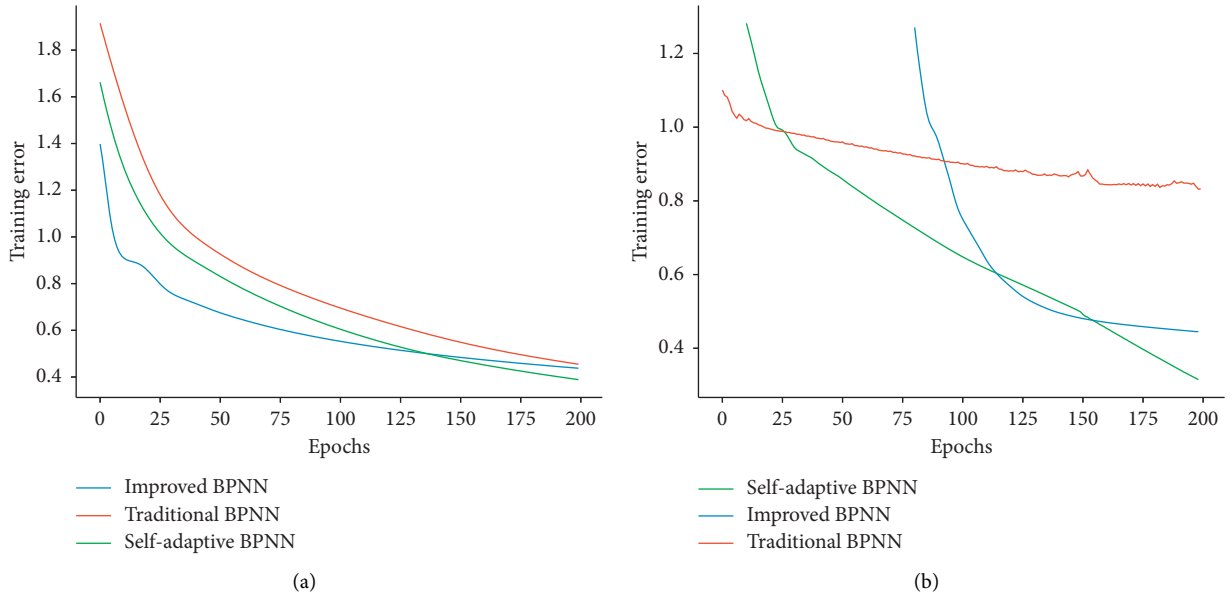
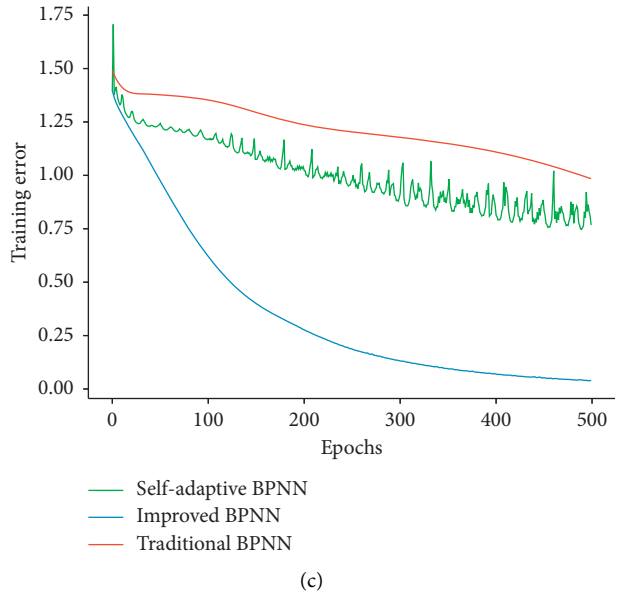


Figure 8: Continued.



(c)

Figure 8: (a) Comparison of the convergence using the Iris dataset. (b) Comparison of the convergence using the Wine dataset. (c) Comparison of the convergence using the Vehicle Silhouettes dataset.

improved BPNN and the self-adaptive BPNN significantly outperform the traditional BPNN. Until the error becomes small, the self-adaptive BPNN converges slightly faster than the improved BPNN. However, Figure 8(c) indicates that the convergence of the self-adaptive BPNN shows a number of fluctuations. The algorithm converges slowly and unstably. Contrarily, the convergence of the improved BPNN is fast and stable.

Figures 9(a)–9(c) show the max and min number of epochs in the 50 times experiments using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. The improved BPNN outperforms the traditional BPNN and performs the similar performances to those of the self-adaptive BPNN based on Iris and Wine datasets. In terms of the Vehicle Silhouettes dataset, the improved BPNN performs the best.

Figures 10(a)–10(c) show the max and min classification accuracies in the 50 times experiments using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. For the simple Iris dataset, the three algorithms perform closely. The presented improved BPNN slightly outperforms the other two algorithms. However, in terms of the Wine dataset, the traditional BPNN performs the worst. Although the classification accuracy of the self-adaptive BPNN can reach to 98%, its minimal accuracy is only 55%, which indicates the unstable performance of the algorithm. Contrarily, the improved BPNN performs the highest accuracy and the most stable performances. In terms of the Vehicle Silhouettes dataset, the improved BPNN also supplies the best accuracies.

To further indicate the performance of the improved BPNN, the statistical indices of evaluating the epoch and accuracy using three datasets for 50 times experiments are listed in Tables 8–10.

In the Iris dataset-based experiments, in terms of the epoch and accuracy tests, the improved BPNN and the self-adaptive BPNN perform similarly, though the improved BPNN shows slightly higher variance of the epoch, and also, both of them outperform the traditional BPNN.

In the Wine dataset-based experiments, although the traditional BPNN performs the least mean and variance of the epoch, it shows the lowest mean accuracy with the highest variance. Contrarily, the improved BPNN significantly outperforms the self-adaptive BPNN in terms of both the epoch and accuracy.

In the Vehicle Silhouettes dataset-based experiments, the traditional BPNN shows the worst performance. The improved BPNN outperforms the self-adaptive BPNN in terms of accuracy. For the epoch, the improved BPNN shows the least mean, and its variance is slightly higher than that of the self-adaptive BPNN.

Figures 11(a)–11(c) indicate that for the improved BPNN, the classification accuracy is impacted by batch size, and also, in terms of different datasets, the optimal batch size which leads to the highest accuracy is different. Therefore, a proper batch size is able to improve the classification accuracy for the presented improved BPNN.

Figures 12(a) and 12(b) indicate the numbers of epochs, nonconvergence, and overfitting of the traditional BPNN and the improved BPNN with varying learning rates (lr = 0.1, 0.01, 0.001, and 0.0001; 50 times experiments for each lr).

Firstly, Figure 12(a) shows that if the learning rate is small (lr = 0.0001), the traditional BPNN cannot converge totally, which results in the loss of the experimental result. However, Figure 12(b) shows that even if the learning rate is 0.0001, the improved BPNN can still converge. Secondly, for each learning rate, respectively, the number of exceptions (nonconvergence and overfitting) of the improved BPNN is

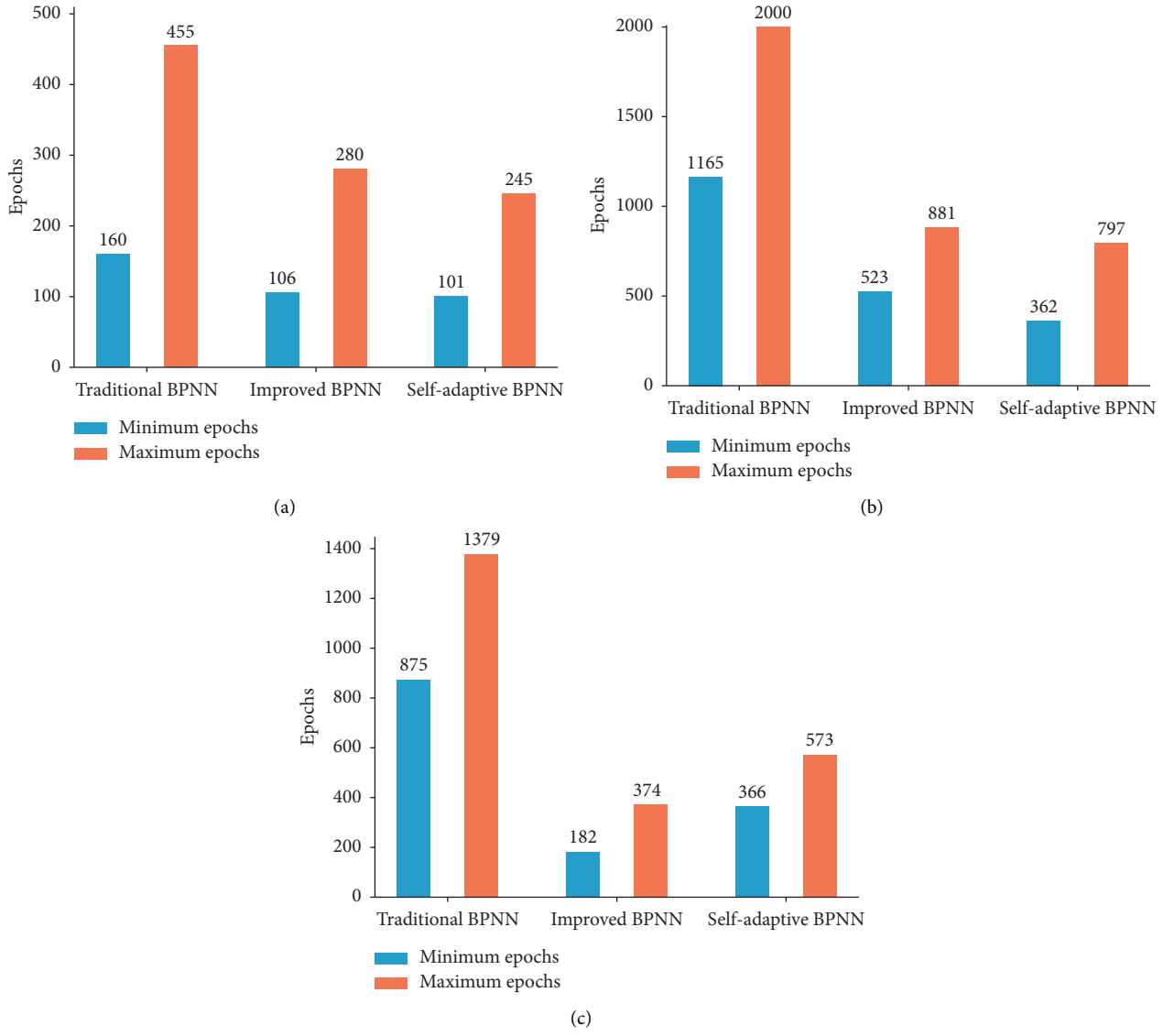


Figure 9: (a) The max and min number of epochs in 50 times experiments using the Iris dataset. (b) The max and min number of epochs in 50 times experiments using the Wine dataset. (c) The max and min number of epochs in 50 times experiments using the Vehicle Silhouettes dataset.

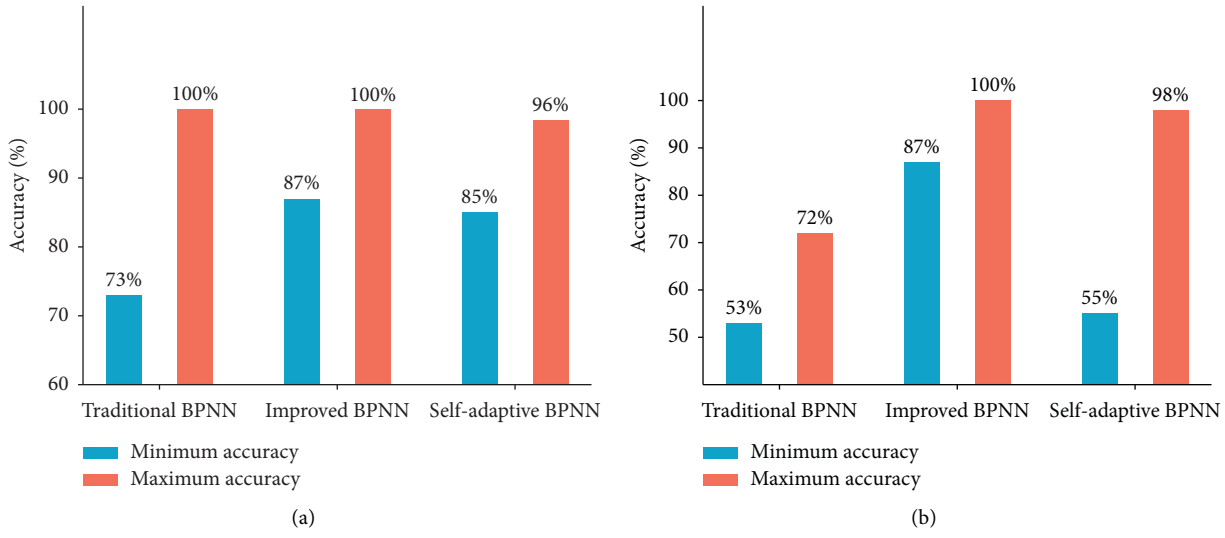
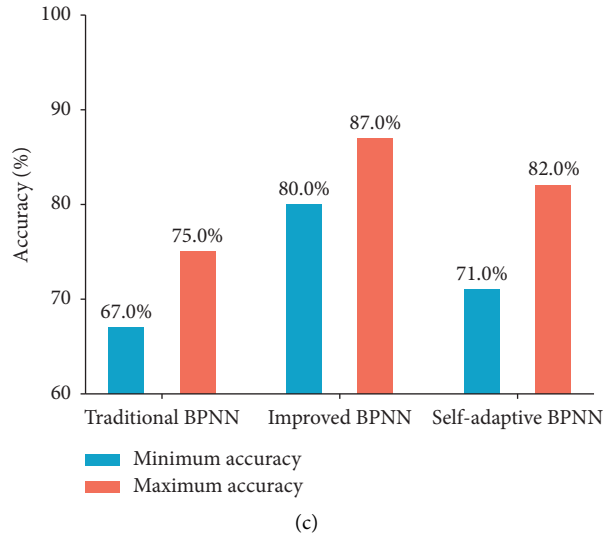


Figure 10: Continued.



(c)

Figure 10: (a) The max and min classification accuracies in 50 times experiments using the Iris dataset. (b) The max and min classification accuracies in 50 times experiments using the Wine dataset. (c) The max and min classification accuracies in 50 times experiments using the Vehicle Silhouettes dataset.

Table 8: The statistical indices of evaluating the epoch and accuracy using the Iris dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	188.82	2557.62	96.7	9.23
Traditional BPNN	342	4096.43	93.3	63.27
Self-adaptive BPNN	172.3	1093.96	96.7	11.73

Table 9: The statistical indices of evaluating the epoch and accuracy using the Wine dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	723.26	6266.69	94.4	28.31
Traditional BPNN	536.8	5974.37	67.3	372.87
Self-adaptive BPNN	1531	31924.12	91.4	55.2

Table 10: The statistical indices of evaluating the epoch and accuracy using the Vehicle Silhouettes dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	245	3243.19	83.5	21.06
Traditional BPNN	993	23542.08	74.8	187.21
Self-adaptive BPNN	465	2806.74	80.7	35.19

Table 11: The details of the training instances and the testing instances.

Dataset	Training instance number	Testing instance number
Iris	80 (class1: 36; class2: 7; class3: 37)	45 (class1: 14; class2: 18; class3: 13)
Wine	79 (class1: 36; class2: 7; class3: 36)	54 (class1: 23; class2: 19; class3: 12)
Vehicle	450 (Class1: 30; Class2: 140; Class3: 140; Class4: 140)	254 (Class1: 62; Class2: 62; Class3: 59; Class4: 71)

less than that of the traditional BPNN. Thirdly, for each learning rate, respectively, the improved BPNN performs a smaller number of epochs than that of the traditional BPNN. The experimental results prove that the improvements done by this paper can significantly improve the convergence of BPNN.

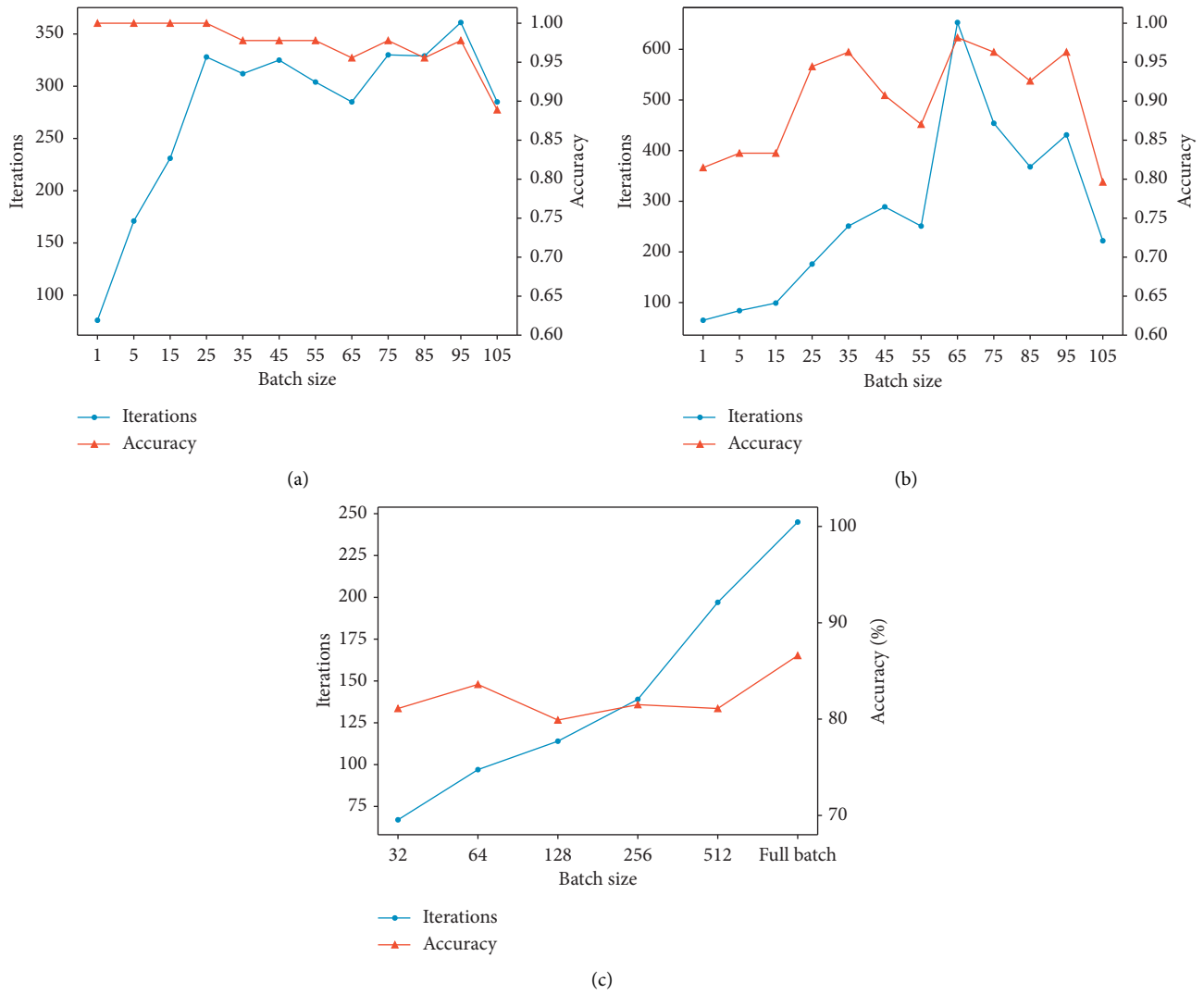


Figure 11: (a) Impact of batch sizes on classification accuracy of the improved BPNN using the Iris dataset. (b) Impact of batch sizes on classification accuracy of the improved BPNN using the Wine dataset. (c) Impact of batch sizes on classification accuracy of the improved BPNN using the Vehicle Silhouettes dataset.

In next experiments, imbalanced training datasets are generated based on the original Iris dataset, Wine dataset, and Vehicle Silhouettes dataset. The details of the training and testing instances are listed in Table 11. The imbalanced training datasets are firstly processed by the Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 10$), and then the improved BPNN carries out the classifications based on both the balanced and imbalanced training data for 50 times. The average classification accuracies are shown in Figures 13(a)–13(c).

Both Figures 13(a) and 13(b) indicate that the imbalanced training datasets significantly impact the training of the improved BPNN, which finally leads to severe misclassifications. However, balanced by the Fréchet distance-based Borderline-SMOTE, the network can be correctly and sufficiently trained. Therefore, the classification accuracies can be greatly improved. However, due to the complicated attributes of the instances of the Vehicle Silhouettes dataset, although the classification based on the balanced dataset also outperforms the imbalanced dataset, its average accuracy is only 79.53%. This point indicates that the dimension of the data may severely impact the performance of the class balance algorithm.

3.3. Evaluation for the Parallelized Improved BPNN in Hadoop Cluster. Firstly, the original Iris dataset is employed to evaluate the classification accuracy of the parallelized improved BPNN. A number of 105 instances are the training instances which will be separated for the parallelized training. The other 45 instances are the testing instances. A number of three mappers start in parallel. Each mapper initializes one sub-BPNN. The standalone BPNN and the parallelized long short-term memory network (LSTM) are also implemented in terms of comparison. Especially, the configuration of the parallelized LSTM is as the same as that of the parallelized BPNN. The number of the training

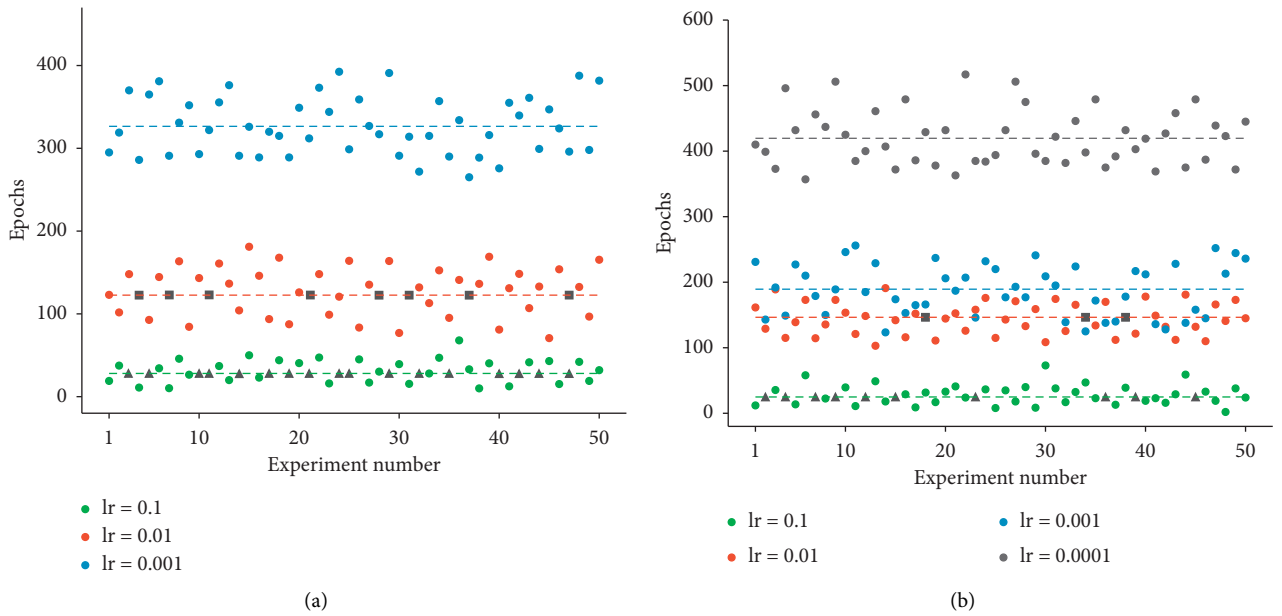


Figure 12: (a) The convergence of the traditional BPNN with various learning rates (▲: nonconvergence, ■: overfitting). (b) The convergence of the improved BPNN with various learning rates (▲: nonconvergence, ■: overfitting).

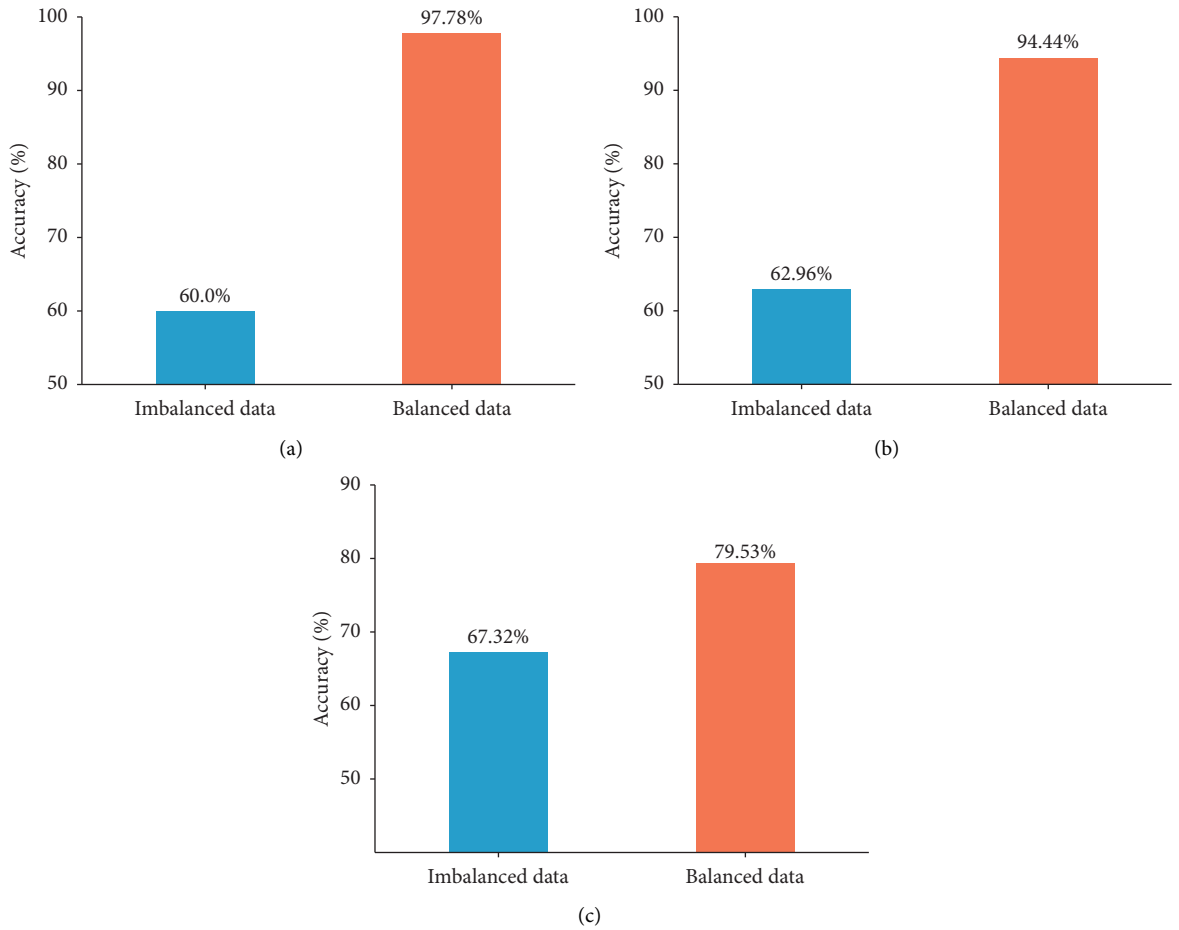


Figure 13: (a) Comparisons of the classification results using the Iris dataset. (b) Comparisons of the classification results using the Wine dataset. (c) Comparisons of the classification results using the Vehicle Silhouettes dataset.

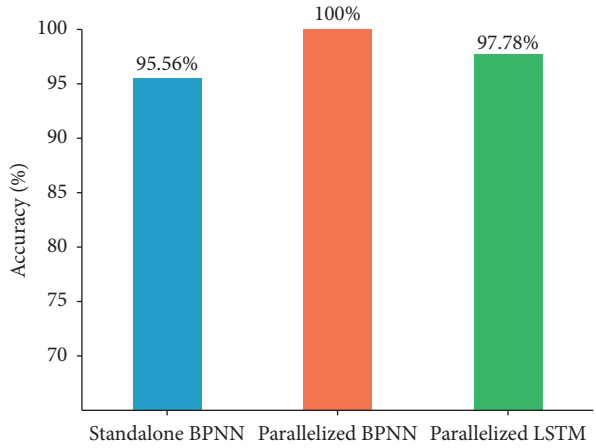


Figure 14: The accuracy comparison for the standalone BPNN, the parallelized BPNN, and the parallelized LSTM.

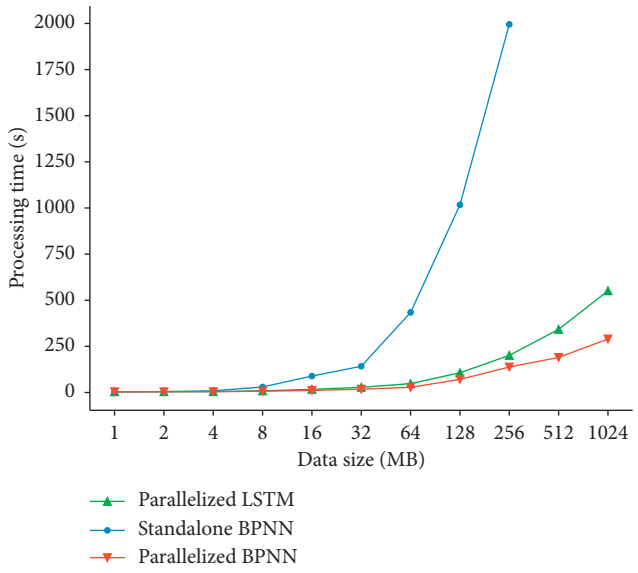


Figure 15: Efficiency comparison for the standalone BPNN, the parallelized BPNN, and the parallelized LSTM.

instances is 105 and the number of the testing instances is 45 for the standalone BPNN.

Figure 14 indicates that the parallelized BPNN outperforms the standalone BPNN and the parallelized LSTM. Benefitting from the weighted voting, the potential accuracy loss issue in the data separation is properly handled. Actually, the classification accuracies for the 3 sub-BPNNs are 96.88%, 94.44%, and 91.89%, respectively. However, the accuracy for the aggregated result can reach to 100%. It further proves that the weighted voting can help to aggregate the weak classifiers to a strong classifier. Additionally, the data separation may intensify the class imbalance issue of the training dataset in each sub-BPNN. However, the Fréchet distance-based Borderline-SMOTE can effectively solve the issue, which guarantees the classification accuracy. Moreover, the parallelized LSTM performs slightly worse than the parallelized BPNN. It is well known that LSTM is suitable for processing the time-series data. However, the Iris dataset employed for the experiments is not related to time series, which indicates that the type of the dataset may affect the performance of the parallelized LSTM in terms of accuracy.

In terms of the classification efficiency evaluation, this paper duplicates the size of the Iris dataset from 1 MB to 1024 MB. The cluster starts 16 mappers in parallel. The processing times of the standalone BPNN, the parallelized BPNN, and the parallelized LSTM are listed in Figure 15.

Figure 15 indicates that the performances of three algorithms are quite close with smaller data sizes. Due to the overhead of the Hadoop cluster, the standalone BPNN even outperforms the parallelized BPNN and the parallelized LSTM. However, along with the increasing data sizes, because of the computing resource limitations of the standalone environment, the processing time of the standalone BPNN increases sharply. Contrarily, due to less computations, the parallelized BPNN outperforms the parallelized LSTM slightly and processes the large volume of data relatively efficiently.

4. Conclusion

In order to serve the classifications for large-scale data, this paper presents a parallelized improved BPNN algorithm. The parallelization is based on the data separation, and the parallelization is implemented using the Hadoop framework. To overcome the classification accuracy loss issue caused by the separation, the weighted voting is presented to improve the classification accuracy. Based on the experimental results, the parallelization shows effectiveness for dealing with the large-scale data. However, there are other two issues existing. The first is that the class imbalance issue in the training dataset significantly impacts the training effect of BPNN, which finally leads to the deterioration of the classification accuracy. Therefore, this paper presents the Fréchet distance-based Borderline-SMOTE algorithm in enabling the class balance. According to the experimental results, the balanced training dataset can tremendously improve the classification accuracy. The second is that the convergence issue may exist in BPNN. Therefore, this paper improves the input layer, hidden layer, and activation function of BPNN by employing zero-mean, batch-normalization, and ReLU, respectively. Based on the comparisons to the traditional BPNN and the self-adaptive BPNN, the presented improved BPNN has great potential to serve the classification tasks accurately and efficiently.

Data Availability

The data of the models and algorithms used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by the State Grid Science and Technology Project (SGAH0000TKJS1900091).

References

[1] G. Wang, Z. Deng, and K.-S. Choi, "Tackling missing data in community health studies using additive LS-SVM classifier," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 579–587, 2018.

[2] M. Skubic, R. D. Guevara, and M. Rantz, "Automated health alerts using in-home sensor data for embedded health assessment," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 3, pp. 1–11, 2015.

[3] L. Chen, X. Li, Q. Z. Sheng et al., "Mining health examination records—a graph-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2423–2437, 2016.

[4] M. Huang, H. Han, H. Wang, L. Li, Y. Zhang, and U. A. Bhatti, "A clinical decision support framework for heterogeneous data sources," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1824–1833, Nov. 2018.

[5] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, and M. Buckland, "A hybrid feature selection with ensemble classification for imbalanced healthcare data: a case study for brain tumor diagnosis," *IEEE Access*, vol. 4, pp. 9145–9154, 2016.

[6] Y. Tseng, X. Ping, J. Liang, P. Yang, G. Huang, and F. Lai, "Multiple-time-series clinical data processing for classification with merging algorithm and statistical measures," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 1036–1043, 2015.

[7] X. Peng, W. Lai, and Y. Chen, "Application of clustering analysis in typical power consumption profile analysis," *Power System Protection and Control*, vol. 42, no. 19, pp. 68–72, 2014.

[8] Y. Xu, G. Li, K. Guo, S. Guo, and W. Lin, "Research on parallel clustering of power load based on improved *k*-means algorithm," *Computer Engineering and Applications*, vol. 53, pp. 260–265, 2017.

[9] Z. Li, B. Zhou, and N. Lin, "Classification of daily load characteristics curve and forecasting of short-term load based on fuzzy clustering and improved BP algorithm," *Power System Protection and Control*, vol. 40, no. 3, pp. 56–60, 2012.

[10] O. Noureldeen and I. Hamdan, "A novel controllable crowbar based on fault type protection technique for DFIG wind energy conversion system using adaptive neuro-fuzzy inference system," *Protection and Control of Modern Power Systems*, vol. 3, no. 1, pp. 328–339, 2018.

[11] A. A. Majd, H. Samet, and T. Ghanbari, "*k*-NN based fault detection and classification methods for power transmission systems," *Protection and Control of Modern Power Systems*, vol. 2, no. 2, pp. 359–369, 2017.

[12] X. Su, T. Liu, H. Cao et al., "A multiple distributed BP neural networks approach for short-term load forecasting based on Hadoop framework," *Proceedings of the CSEE*, vol. 37, pp. 4966–4972, 2017.

[13] Q. Li, W. Cai, X. Wang, Y. Zhou, D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 844–848, Singapore, December 2014.

[14] Z. Jiao, X. Gao, Y. Wang, J. Li, and H. Xu, "Deep convolutional neural networks for mental load classification based on EEG data," *Pattern Recognition*, vol. 76, pp. 582–595, 2018.

[15] S. Yang and C. Shen, "A review of electric load classification in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 103–106, 2013.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," 2015, https://arxiv.org/abs/1509.01626.

[17] C. Deng, Y. Liu, L. Xu et al., "A MapReduce-based parallel *K*-means clustering for large-scale CIM data verification," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 11, pp. 3096–3114, 2016.

[18] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Design of reinforced interval type-2 fuzzy C-means-based fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3054–3068, 2018.

[19] X. Wang, Y. Guo, Y. Wang, and J. Yu, "Automatic breast tumor detection in ABVS images based on convolutional neural network and superpixel patterns," *Neural Computing and Applications*, vol. 31, no. 4, pp. 1069–1081, 2019.

[20] Q. Jiang and F. Chang, "A novel rolling-element bearing faults classification method combines lower-order moment spectra and support vector machine," *Journal of Mechanical Science and Technology*, vol. 33, no. 4, pp. 1535–1543, 2019.

[21] T. Kohonen, "Physiological interpretation of SOM," in *Self-Organizing Maps*, vol. 30, Springer, Berlin, Germany, 1997.

[22] S. Niazmardi, S. Homayouni, and A. Safari, "An improved FCM algorithm based on the SVDD for unsupervised hyperspectral data classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 831–839, 2013.

[23] Y. Liu, J. Yang, Y. Huang, L. Xu, S. Li, and M. Qi, "MapReduce based parallel neural networks in enabling large scale machine learning," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 297672, 13 pages, 2015.

[24] Y. Liu, W. Jing, and L. Xu, "Parallelizing backpropagation neural network using MapReduce and cascading model," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 2842780, 11 pages, 2016.

[25] Y. Liu, L. Xu, and M. Li, "The parallelization of back propagation neural network in MapReduce and spark," *International Journal of Parallel Programming*, vol. 45, no. 4, pp. 760–779, 2017.

[26] N. Almaadeed, A. Aggoun, and A. Amira, "Speaker identification using multimodal neural networks and wavelet analysis," *IET Biometrics*, vol. 4, no. 1, pp. 18–28, 2015.

[27] D. Gu, Q. Ai, C. Chen, and S. Shen, "Application of adaptive neural network in dynamic load modeling," *Proceedings of the CSEE*, vol. 16, pp. 31–36, 2007.

[28] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[29] G. Wang, G. B. Giannakis, and J. Chen, "Learning ReLU networks on linearly separable data: algorithm, optimality, and generalization," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, 2019.

[30] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks-a comparative study with CIFAR-10 data," in *Proceedings of the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, Kolkata, India, January 2018.

[31] A. N. Abbasi and M. He, "Convolutional neural network with PCA and batch normalization for hyperspectral image classification," in *Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium IGARSS-2019*, Yokohama, Japan, July–August 2019.

[32] S. Wu, G. Li, L. Deng et al., "L1-Norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2018.

[33] K. Tachibana and K. Otsuka, "Wind prediction performance of complex neural network with ReLU activation function," in *Proceedings of the 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Nara, Japan, September 2018.

[34] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *Proceedings of the 2015 International Joint Conference on Neural Networks*, pp. 1–8, IEEE, Killarney, Ireland, July 2015.

[35] T. Tang, S. Chen, M. Zhao, W. Huang, and J. Luo, "Very large-scale data classification based on *K*-means clustering and multi-kernel SVM," *Soft Computing*, vol. 23, no. 11, pp. 3793–3801, 2019.

[36] A. Farrell, G. Wang, S. A. Rush et al., "Machine learning of large-scale spatial distributions of wild turkeys with high-dimensional environmental data," *Ecology and Evolution*, vol. 9, no. 10, pp. 5938–5949, 2019.

[37] X. Xu, T. Liang, J. Zhu, D. Zheng, and T. Sun, "Review of classical dimensionality reduction and sample selection methods for large-scale data processing," *Neurocomputing*, vol. 328, pp. 5–15, 2019.

[38] Apache Hadoop, 2019, http://hadoop.apache.org/.

[39] Apache Spark, 2019, http://spark.apache.org/.

[40] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, pp. 878–887, Springer, Berlin, Germany, 2005.

[41] Y. Zhang, L. Shuai, Y. Ren, and H. Chen, "Image classification with category centers in class imbalance situation," in *Proceedings of the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nanjing, China, May 2018.

[42] F. Li, S. Li, C. Zhu, X. Lan, and H. Chang, "Class-imbalance aware CNN extension for high resolution aerial image based vehicle localization and categorization," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, China, June 2017.

[43] C. Zhang, K. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, July 2016.

[44] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk, "Fréchet distance for curves, revisited," in *Lecture Notes in Computer Science*, pp. 52–63, Springer, Berlin, Germany, 2006.

[45] W. D. Mulder, G. Molenberghs, and G. Verbeke, "An interpretation of radial basis function networks as zero-mean Gaussian process emulators in cluster space," *Journal of Computational and Applied Mathematics*, vol. 363, pp. 249–255, 2020.

[46] D. Jiao, H. Wang, J. Zhu, Z. Chi, and S. Zeng, "EV battery SOH diagnosis method based on discrete Fréchet distance," *Power System Protection and Control*, vol. 44, no. 12, pp. 68–74, 2016.

[47] J. Zhu, Z. Huang, and X. Peng, "Curve similarity judgment based on the discrete Fréchet distance," *Journal of Wuhan University of Technology*, vol. 55, no. 2, pp. 227–232, 2009.

[48] C. Chen, G. Huang, Y. Fan, J. Wu, and X. Wang, "Short-term load forecasting based on discrete Fréchet distance and LS-SVM," *Power System Protection and Control*, vol. 42, no. 5, pp. 142–147, 2014.

[49] D. Fan, J. Yang, J. Zhang et al., "Effectively measuring respiratory flow with portable pressure data using back propagation neural network," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 6, Article ID 1600112, pp. 1–12, 2018.

[50] Y. Liu, X. Chen, L. Xu, H. Li, and M. Li, "A resource aware parallelized back propagation neural network in enabling efficient large-scale digital health data processing," *IEEE Access*, vol. 7, 2019.

[51] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[52] Wine Dataset, 21019, http://archive.ics.uci.edu/ml/datasets/Wine.

[53] Iris Dataset, 2019, https://archive.ics.uci.edu/ml/datasets/Iris.

[54] Vehicle Silhouettes Dataset, 2020, https://archive.ics.uci.edu/ml/datasets/Statlog+(Vehicle+Silhouettes).

[55] X. Yang, X. Y. Jin, and J. F. Shen, "A PVC identification method of ECG signal based on improved BPNN," *Applied Mechanics and Materials*, vol. 738-739, pp. 578–581, 2015.