Research Article

A Data Mining Method Using Deep Learning for Anomaly Detection in Cloud Computing Environment

Jin Gao,¹ Jiaquan Liu,² Sihua Guo,¹ Qi Zhang,³ and Xinyang Wang⁴

¹State Grid Chongqing Electric Power Research Institute, Chongqing, China

²State Grid Chongqing Electric Power Company, Chongqing, China

³State Grid Chongqing Electric Power Company Maintenance Company, Chongqing, China

⁴State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing, China

Correspondence should be addressed to Xinyang Wang; 20141886@cqu.edu.cn

Received 24 July 2020; Revised 7 September 2020; Accepted 24 September 2020; Published 20 November 2020

Academic Editor: Yi-Zhang Jiang

Copyright © 2020 Jin Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at problems such as slow training speed, poor prediction effect, and unstable detection results of traditional anomaly detection algorithms, a data mining method for anomaly detection based on the deep variational dimensionality reduction model and MapReduce (DMAD-DVDMR) in cloud computing environment is proposed. First of all, the data are preprocessed by a dimensionality reduction model based on deep variational learning and based on ensuring complete data information as much as possible, the dimensionality of the data is reduced, and the computational pressure is reduced. Secondly, the data set stored on the Hadoop Distributed File System (HDFS) is logically divided into several data blocks, and the data blocks are processed in parallel through the principle of MapReduce, so the k -distance and LOF value of each data point can only be calculated in each block. Thirdly, based on stochastic gradient descent, the concept of k -neighboring distance is redefined, thus avoiding the situation where there are greater than or equal to k -repeated points and infinite local density in the data set. Finally, compared with CNN, DeepAnt, and SVM-IDS algorithms, the accuracy of the scheme is increased by 10.3%, 18.0%, and 17.2%, respectively. The experimental data set verifies the effectiveness and scalability of the proposed DMAD-DVDMR algorithm.

1. Introduction

With the popularization of cloud computing, the “reliability of never down machine” in industrial applications has gradually changed from expectation to practical need. To solve this problem, how to improve the accuracy, sensitivity, and execution efficiency of anomaly detection algorithm in data mining becomes more important [1–3].

Aiming at the large volume and various types of the cloud computing environment, existing research starts from many aspects. The literature [4] proposed a cloud computing network traffic matrix estimation and anomaly detection model based on the Bayesian network. Because the ideal naive Bayesian model assumes that attributes are independent, this assumption is often not established in practical applications, so the effect is often not good enough in multiattribute situations. The literature [5] proposes

statistical learning of anomaly detection in the cloud server system based on the Markov chain. The Markov model is not suitable for long-term prediction of the system, so it can only judge the short-term changes of the system. The judgment of long-term operation is not accurate enough. Supervised anomaly detection algorithms need a large number of samples for model detection before monitoring anomaly data. For example, the wavelet soft threshold method proposed in the literature [6] is used to eliminate noise or errors in data streams to support the framework of anomaly detection in uncertain data streams. This scheme is based on the technology of effective period pattern recognition and feature extraction under large sample detection, so there is uncertainty in engineering practice to some extent.

For the anomaly detection of big data, there have been algorithm models for anomaly detection methods based on machine learning to classify data with different character

attributes through linear or nonlinear methods, for example, one type of support vector machine [7] (one-class SVM, OCSVM), and this model is simpler than the support vector machine training set. At the same time, the classification algorithm based on the neural network [8] has a high research value at this stage. Neural networks are generally divided into two types of convolutional neural networks [9] and deep neural networks [10], which are proposed in the article by He et al. [11]. A convolutional neural network for video classification is proposed, but the generalization ability and the huge amount of parameters are huge problems. Su et al. [12] proposed the shortcomings of deep neural networks, which are vulnerable to one-pixel attack, and one pixel may affect the output of the entire neural network.

Unsupervised anomaly detection gets rid of the shortcomings of the above schemes. It does not need labeled samples, so it has higher practical value. For example, the local outlier factor (LOF) algorithm proposed in the literature [13] can determine the abnormal degree of a data object by calculating the local outlier factor (LOF value) of each point. Compared with other algorithms, the algorithm is simple in theory and highly adaptable. It can detect global and local anomalies effectively. However, the LOF algorithm is designed based on local density, which has high complexity and assumes that there are no more than or equal to k -repeating points. Therefore, a new density-based outlier detection (DBOD) algorithm is proposed in the literature [14], which defines the point density of data as the nearest point to k divided by the distance of k . Although this algorithm reduces computational complexity and improves work efficiency, its data processing scale is limited by memory capacity and data complexity. Therefore, it is very important to design an anomaly detection algorithm which can not only guarantee the advantages of the LOF algorithm but also deal with a large number of data efficiently [15, 16].

The innovative points of the proposed DMAD-DVDMR in the cloud computing environment are as follows:

- (1) Based on the data preprocessing method of deep variational dimensionality reduction, through the training of labeled samples, a potential presentation layer with high predictive ability is constructed to ensure maximum information while reducing the dimension of data features, which is the next step. Anomaly detection provides a more complete preprocessing result.
- (2) The local anomaly factor detection method based on MapReduce avoids the excessive density of data caused by the excessive concentration of data and greatly improves the data processing capacity and work efficiency.
- (3) The hybrid model that combines deep variational dimensionality reduction and local anomaly factor detection improves the generalization ability, solves the calculation problem caused by the excessively high data dimension, and improves the retention of label information.

The rest of this article includes: the second section introduces the data preprocessing method based on the deep variational dimensionality reduction model; the third

section introduces the local anomaly factor detection algorithm based on MapReduce and stochastic gradient descent. Section 4 presents the discussion of experiments and numerical examples. Section 5 summarizes the outlook.

2. Data Preprocessing Based on Deep Variational Dimensionality Reduction Model

2.1. Deep Variational Learning. Sufficient dimensionality reduction (SDR) is a dimensionality reduction idea that aims to find a low-dimensional representation of the data while retaining predictive information about label variables. The original work of SDR proposed a method to quantify information using the concept of information theory and introduced an iterative algorithm to extract features that maximize information. The SDR method is usually applied to continuous target variables, but for discrete target variables, methods based on distance covariance can estimate the central subspace. The following regression target is often encountered in the field of machine learning, which is the predicted value of the object predictor label $y \in \mathbb{R}^D$ given the observation value $x \in \mathbb{R}^P$. In the high-dimensional field, traditional regression methods may require a large amount of training data to avoid overfitting. Therefore, it is urgent to use the dimensionality reduction method to replace the original covariate x with another variable $z \in \mathbb{R}^D$, which retains most or all of the information and changes of x . When z retains all relevant information about y , such a dimensionality reduction method is considered to be sufficient dimensionality reduction. The SDR problem can be explained by any model in Figure 1. The model in Figure 1(a) can use unlabeled samples to construct the latent space, so it can be used for semisupervised learning. The model in Figure 1(b) can only use labeled samples [17, 18].

Variational autoencoder is a deep learning model that can effectively maximize the lower bound of the variational log-likelihood of the joint distribution on a large scale. In this model, the conditional distribution is reparameterized by the neural network (reparameterization trick). Compared with the standard variational autoencoder, the proposed model pays more attention to the coding process and carries as much data as possible to distinguish the data during the coding process. Compared with the standard variational autoencoder, the proposed model pays more attention to the coding process and carries as much data as possible to distinguish the data during the coding process, so we hope to maximize the joint distribution probability $p(X, Y)$. Use the depth variational autoencoder to maximize this lower limit, so this part of the model used for data dimensionality reduction preprocessing is called the depth variational dimensionality reduction model (DVDR). Through the training of the labeled samples, a potential presentation layer with high predictive ability is constructed, and the dimensionality is reduced as much as possible on the premise of preserving more predictive label information.

2.2. Hybrid Model Combining Deep Variational Dimensionality Reduction. As the dimensionality of the generated data continues to increase and become increasingly complex,

FIGURE 1: Deep variational dimensionality reduction model.

dimensional disasters have become a common problem [19–21]. Models that reduce the dimensionality of the data and reduce the loss of original data features are very necessary for data mining tasks. The hybrid model provides a good idea for solving the problem of dimensional disaster and data mining [22, 23]. The depth variational dimensionality reduction algorithm is improved from the variational autoencoder. The autoencoder can be used as a feature dimensionality reduction algorithm because the dimensionality of the intermediate layer is smaller than that of the input data. After the process of learning and characterizing the input data, a high low-dimensional feature vector of the dimensional data and the reconstruction part is the test of the feature lifting the performance of the most coded part. Compared with the original input data, the reconstructed new data have a smaller reconstruction error, and the more it shows that the middle layer has learned more Good features, so the generation performance is also a comparative indicator of this type of model. Compared with some mainstream linear dimensionality reduction models, this type of generative model can better learn the nonlinear features in high-dimensional data, and can better express the original high-dimensional features. The feature reduction effect is better than the nonlinear dimensionality reduction model. Although the undercomplete autoencoder with the middle layer smaller than the input layer can be used to reduce the dimensionality of the data, the autoencoder only pays attention to the reconstruction error and has poor adaptability to noise vectors. At the same time, the middle layer has discrete values and poor generalization ability, unable to perform good feature representation on the original high-dimensional data. Therefore, a deep variational dimensionality reduction model is proposed, which improves the generalization ability, strengthens the encoder part, and improves the integrity of the predicted label information. After the depth variational dimensionality reduction model training is completed, the encoding part can be used as the processing part of the original data to reduce the feature dimensions of the original data, so that the anomaly detection model can be better used for anomaly detection. The structure of the hybrid model is as follows.

As shown in Figure 2, after the original data are reduced by the deep variational dimensionality reduction model, a latent vector with a dimension smaller than the original data is obtained, and the latent vector is used as the input of the improved anomaly detection model because the improved deep variational dimensionality reduction model The

features used for classification are well preserved, so the dimensionality disaster problem can be solved well after dimensionality reduction.

3. Suggested Anomaly Detection Algorithm

Based on the hybrid model shown in Figure 2, the data that have undergone dimensionality reduction preprocessing are calculated by calculating the local anomaly factor of each point to achieve the effective detection of global and local anomalies.

3.1. Anomaly Detection Framework Based on MapReduce.

The MapReduce parallel programming model proposed by Google has become the main model of large data processing because of its simplicity, scalability, and fault tolerance [24, 25]. The core idea of the parallel programming model is “divide and rule,” that is, dividing dense large data without intrinsic dependency into several fragments, then parallel computing, and processing by multiple subtasks, respectively. The results are aggregated to the control task for output [26].

The MapReduce programming model realizes the above idea. The distributed computing task is highly abstracted into two phases: Map and Reduce. The corresponding Map and Reduce processing functions in the developer’s implementation phase are shown in Figure 3 [27]. According to a specific slicing strategy, input data will be divided into multiple slices. Each slice is then transformed and processed by a Map task. Each row of data in the fragmentation applies a user-defined Map function. To reduce the network transmission consumption in aggregating intermediate results, users can specify a combined function to merge and simplify the output of Map tasks. These intermediate output results will be hashed to different partitions and sorted according to a specific partition strategy. Then, the intermediate output of the same partition number will be shuffled and copied to the corresponding node. Before running the Reduce task, the nodes will merge the intermediate results as the complete input. The merged data are reduced and processed by the Reduce function, and the final results are written to the file system.

The MapReduce programming model hides specific processes such as job control and process scheduling in cluster management. Therefore, developers concentrate on program development without or little consideration of fragmentation, partition, network transmission, and I/O details. In this way, the reliability, ease of use, and fault tolerance of parallel computing are ensured. MapReduce job uses master-slave architecture as its operation mechanism. It is generally composed of one master node and several slave nodes, with client nodes for submitting and monitoring MapReduce job. The master node initiates a JobTracker process. This process is responsible for tracking job progress, including receiving jobs submitted by clients, distributing jobs in the form of subtasks to slave nodes, and monitoring the execution of returned jobs from nodes [28]. The slave node initiates one or more TaskTracker processes to track the progress of tasks, including Map tasks or Reduce tasks assigned to it. The entire flow of MapReduce is shown in Figure 4.

FIGURE 2: Overall hybrid architecture model.

FIGURE 3: Working principle of MapReduce.

FIGURE 4: The entire flow of MapReduce.

Real-time online monitoring part is used to collect the current time data for anomaly detection. Then, data are compared with the threshold to determine whether the current time data are abnormal. The abnormal data are reported, and remaining normal data are added to R-tree. So, the oldest data are deleted from R-tree. In this way, the model can be adjusted according to the changes of the normal system, so as to achieve the purpose of self-adaptation.

LOF is a general and portable algorithm. We collect system information directly from cloud computing platform

for anomaly detection, such as CPU usage, memory usage, and other basic system information. In addition, LOF calculates anomaly scores for each detection data. Users can choose threshold according to the situation, and find a suitable compromise between detection rate and false alarm rate. In addition, LOF only needs to learn the current normal situation and does not need to train all kinds of anomalies. It has good adaptability and recognition for new anomalies.

LOF is used to describe the abnormal degree of an object. It calculates the degree of anomaly by comparing the density of this object and its neighbors.

3.2. Anomaly Detection Algorithm Based on Stochastic Gradient Descent. In order to get the optimal weight in most supervised learning models, we need to create cost loss function for the model. Then, choose the appropriate optimization algorithm to get the minimum function loss value. The gradient descent algorithm is the most widely used optimization algorithm at present [29]. Its core idea is to calculate the minimum loss value of function. Firstly, calculate the gradient of loss function, and then reduce the loss value of function gradually according to the direction of gradient. Finally, by constantly updating and adjusting the weight value, the loss value of function reaches the minimum, so as to obtain the optimal solution. The stochastic gradient descent (SGD) algorithm is an improved algorithm based on gradient descent [30]. SGD randomly selects one sample at a time to update iteratively, rather than for all samples. Therefore, the algorithm significantly reduces the computational complexity. SGD has the characteristics of fast training speed and easy convergence. It is also the most popular optimization algorithm for researchers at home and abroad. The SGD-related formulas are as follows:

$$\begin{aligned} g(\gamma) &= \sum_{j=0}^n \gamma_j x_j + \sum_{i=0}^n x_i, \\ f(\gamma) &= \frac{\sum_{i=1}^m (y^i - g\gamma(x^i))^2}{\sum_{i=1}^m \gamma^i}, \\ \gamma &= \eta \nabla \gamma f(\gamma), \\ \eta f(\gamma) &= \gamma - \frac{1}{2m} \eta \nabla. \end{aligned} \quad (1)$$

In the formula, the weights of network parameters are represented by γ ; the gradient is represented by $\nabla \gamma$; the loss function is represented by $f(\gamma)$; the objective function is represented by $g(\gamma)$; the sample value of the first sample is represented by y_i ; the total number of iterations is represented by m ; η denotes the step size in gradient descent; and the total number of parameters in KNN is represented by j . As described above, learning rate is very important for the gradient descent algorithm. If the setting of η is too small, it will need several iterations to find the optimal solution and reduce the convergence speed of the network. It may even lead to stagnation in the local optimal solution. Although the training speed of KNN will be accelerated, it will increase the probability of skipping the optimal solution with increased learning rate. KNN may not find the optimal solution. It can be seen that η is the key factor to decide whether the gradient descent algorithm is effective or not.

The function of the proposed algorithm is to achieve effective anomaly data detection based on the stochastic gradient descent algorithm and MapReduce. The method ensures the high efficiency of algorithm. That is to say, when abnormal records appear in the data set, the target model obtained by the stochastic gradient descent algorithm can realize fast detection. The basic idea of the algorithm is to distribute the data in the data set to each distributed computing node, perform random gradient descent

algorithm on each node through Map subtask, and use Reduce subtask to update the model merge operation.

The principle of subspace clustering is to reduce high-dimensional data to low-dimensional data, which makes the subsequent data analysis possible. Because of the existence of outliers, subspace clustering is disturbed. The solution is to introduce e_1 regularization coefficient. It should be noted that the initial outlier detection threshold should be set to a larger value to ensure the accurate determination of outliers. For the data to be discriminated, it is first determined whether it is an outlier. The corresponding outlier residual vectors are marked. Otherwise, the subspace is classified and the subspace is updated. The discriminant threshold is updated after each model iteration. Then, with the iteration of the model, the threshold of outlier discrimination decreases exponentially. Finally, all outliers will be detected accurately, and the subspace will be clustered appropriately. In the process of iteration, k subspaces are updated step by step. Since the algorithm processes data by data, the process of updating subspace is equivalent to the SGD. The memory requirement is quite low, so the problem of large memory occupation is successfully solved.

Many existing algorithms are too absolute in judging anomalies, either normal or abnormal. However, in practical application, many test data are difficult to judge absolutely, which will result in high false alarm rate or high missed detection rate. It is also difficult to adjust the severity of anomaly detection. So, we need a degree value to judge it. We can determine different thresholds according to different use environments when we finally output the anomaly. Eventually output the anomaly points which are larger than the threshold [31].

All kinds of clustering algorithms have certain ability of anomaly detection. The common problem is most clustering algorithms use a global distance criterion as the basis of detection. The anomaly itself has a certain locality, which is related to the distribution of neighbors within a certain range. Therefore, the mechanism of finding anomalies by the clustering algorithm is limited. LOF is based on the local density of the anomaly to determine the anomaly. To describe the local characteristics of LOF, a simple two-dimensional data set in Figure 5 is taken as an example. As can be seen from the graph, the data amount in cluster C_1 is larger than C_2 , but the data density in cluster C_2 is higher than C_1 .

Because of the low density of cluster C_1 , the distance between each data in cluster C_1 and its nearest neighbor is larger than p_2 and its nearest neighbor in C_2 . In this case, p_2 will not be considered as an exception. The global view clustering algorithm will cause false alarm, but LOF can be successfully detected.

3.3. Algorithmic Design. The algorithm of LOF is described as follows:

(1) Calculating k -distance of p

For any natural number k , the k -distance of the test object p is defined as the distance from the test object p to the

FIGURE 5: Advantage of the LOF approach.

k -nearest neighbor of p ($o \in D$). That is, o needs to satisfy two conditions at the same time:

- ① At least k elements $o' \in D$ satisfy $d(p, o') \leq d(p, o)$
- ② At most k elements $o' \in D$ satisfy $d(p, o') < d(p, o)$

(2) Calculating k -distance neighbor sets of p

The k -distance neighbor set of object p is the set of all elements that do not exceed k -distance, defined as

$$N_{k\text{-dis}(p)}(p) = \{q \in D \setminus \{p\} \mid d(p, q) \leq k\text{-dis}(p)\}, \quad o \in D. \quad (2)$$

The set of objects q satisfying the above formulas is the k -distance neighbor set of p .

(3) Calculating the reachable distance of object p relative to object o

For each object p , compute its reachable distance relative to object o :

$$r_dist_k(p, o) = \max\{k_dist(o), d(p, o)\}. \quad (3)$$

$d(p, o)$ represents the distance from object p to object o .

Figure 6 shows a diagram of the reachable distance when $k=4$. If an object is far from o (e.g., p_2 in Figure 6), the reachable distance between them is the real distance. If an object is close enough to o (e.g., p_1 in Figure 6), the reachable distance between them is the k -distance of o . In this way, the fluctuation of $d(p, o)$ of the object near o can be reduced. The smoothing intensity can be adjusted by adjusting k .

So far, we have calculated k -distance, k -distance neighbor aggregation, and reachable distance. In actual anomaly detection, MinPts is defined as k parameter, and all $o \in N_{\text{MinPts}}(p)$ are used to calculate $\text{reach_dist}_{\text{MinPts}}(p, o)$,

FIGURE 6: Reach-dist(p_1 , o) and reach-dist(p_2 , o), ($k=4$).

respectively, so as to determine the density around the object p .

(4) Computing the locally achievable density of object p

The local reachable density of object p is the reciprocal of the average achievable distance between object p and its MinPts neighbors:

$$\text{lrd}_{\text{MinPts}}(p) = \frac{1}{(\text{MinPts}(p, o) / |N_{\text{MinPts}}(p)|)}. \quad (4)$$

$\text{lrd}_{\text{MinPts}}(p)$ represents its local arrival density. If there are at least MinPts different objects with the same coordinates as object p , the local achievable density may be infinite. Because the sum of all achievable distances at this time is 0, so assume that there are not so many identical objects in the database, or by finding MinPts closest to p but different from p coordinates.

(5) LOF for computing object p

The LOF of object p can be calculated according to the following formula:

$$\text{LOF}_{\text{MinPts}}(p) = \left(\frac{\sum_{o \in N_{\text{MinPts}}(p)} ((\text{lrd}_{\text{MinPts}}(o) + 1) / \text{lrd}_{\text{MinPts}}(p))}{|N_{\text{MinPts}}(p)|} \right) + \frac{\text{lrd}_{\text{MinPts}}(p)}{|N_{\text{MinPts}}(p)|}. \quad (5)$$

The LOF of object p represents the abnormal degree of p . It is equal to the average of the ratio of local attainable density of p to the local attainable density of the MinPts-neighbor set of p . If the local achievable density of p is very low, but the local achievable density of MinPts-neighbor set is very high. Then, it indicates that p is very likely to be abnormal.

The LOF algorithm is a density-based anomaly detection algorithm, which has a large amount of computation. There

is a hypothesis about the definition of local reachable density in the LOF algorithm: there is no more than or equal to k -repeating points. When such repetitive points exist, the average reachable distance of these points is zero, and the local reachable density becomes infinite. Obviously, it affects the effectiveness of the Algorithm 1.

Because of the shortage of the LOF algorithm, the concept of k -neighborhood distance is redefined. Propose the anomaly detection algorithm combining depth learning with the MapReduce framework. The redefined concept of k -proximity distance is as follows, “ k -distinct-distance”: for any positive integer k , the k -adjacent distance of point p is defined as k -distance $(p) = d(p, o)$, if the following conditions are satisfied:

- (1) In the sample space, at least $d(p, g) \leq d(p, o)$ exists
- (2) In the sample space, at most $k-1$ is $0 < d(p, g) < d(p, o)$. Let $d(p, o)$ be the distance between point p and point o

This method of improving the proximity distance effectively realizes the data classification of big data scenes. Through a more accurate definition of k value, it realizes fast and effective data classification while ensuring calculation accuracy.

4. Experiments and Results Analysis

The configuration of experimental platform is as follows: 3 PCs (connected via LAN), node configuration for CentOS.7 under Windows VMware Workstation Pro 12.0.0, JDK 1.8, Hadoop 2.7.4. All the algorithms in this paper are implemented in JAVA language and eclipse compiler environment. The experimental environment is a Hadoop cluster based on the cloud platform. Using the KDD99 data set, the proposed DMAD-DVDMR algorithm is compared with the convolutional neural network algorithm (CNN) in the literature [9] from the five perspectives of model robustness, algorithm efficiency, algorithm response time, algorithm accuracy, and scalability. The overall performance of the deep learning algorithm (DeepAnt) in the literature [10] and the intrusion detection method based on support vector machine (SVM-IDS) in the literature [7].

The KDD99 data set is a data packet data set collected from a network connection. It contains a total of 41 attributes and a total of about 5 million data packet information. The data set is divided into a training data set with annotations and an unlabeled training data set. There are two parts of the data set. There are a total of 39 attack labels in the data set. The training set contains 22 label categories in the dimensionality reduction model 39 based on deep variational learning in Chapter 4. The test set contains 17 attack methods that are not in the training set. For the generalization ability of the detection algorithm model, that is, the model can better deal with and prevent unknown attacks, so as to better evaluate the detection performance of the model. Choose a subset of about 500,000 pieces of data as the experimental data set.

input $X = x_1, x_2, \dots, x_n$: data set
 k : number of nearest neighbor
 N : data block number
 θ : threshold for LOF
output Abnormal data and LOF values
 Get data set which $\text{lof}_i > \theta$ from algorithm 2 add in XX
 Calculate $\text{lof}_i > \theta$ of $x_j \in XX$
 return Abnormal data and LOF

ALGORITHM 1: DMAD-DVDMR algorithm

4.1. Basic Performance Verification of the Algorithm. The basic performance of the proposed DMAD-DVDMR method was verified from the three aspects of algorithm robustness, algorithm accuracy, and algorithm response time, and the performance of CNN, DeepAnt, and SVM-IDS methods is compared. The AUC indicator is used as the robustness evaluation standard of the data mining anomaly detection method. AUC (area under curve) is defined as the area under the ROC (receiver operating characteristic curve) curve. Classifiers with larger AUC have more robust anomaly detection performance which performed. As the label position of the abnormal data changes, the comparison of the AUC indicators on the KDD99 data set is shown in Figure 7.

Next, analyze the accuracy and response time of the algorithm. The detection results of normal data and abnormal data using anomaly detection algorithm are shown in Table 1. Normal data are expressed as “positive” and abnormal data as “negative.” Data detected as normal are expressed as “1,” and data detected as abnormal are expressed as “0.”

Accuracy (Ac) is defined as

$$Ac = \frac{P_1 + N_0}{P_1 + P_0 + N_1 + N_0}. \quad (6)$$

It is expressed as the ratio of normal data detected as normal and abnormal data detected as abnormal to the total data, that is, the probability of correct detection. According to the test data, the calculated accuracy of each algorithm is shown in Figure 8.

As shown in Figure 8, with the increase of data set size, the impact of accidental errors gradually decreases, and the accuracy of the algorithm continues to improve until it stabilizes to a more stable value. Compared with CNN, DeepAnt, and SVM-IDS, the accuracy of the proposed DMAD-DVDMR algorithm for anomaly detection can reach more than 94%. The accuracy of the algorithm is improved by 10.3%, 18.0%, and 17.2%, respectively.

As shown in Figure 9, it is the anomaly detection response time of different algorithms. When the size of the data set increases to 15,000, the response time of the algorithm will increase significantly. Compared with CNN, DeepAnt, and SVM-IDS methods, the response time of the proposed DMAD-DVDMR algorithm is reduced by 23.3%, 28.1%, and 36.1%, respectively.

```

input data set  $X = x_1, x_2, \dots, x_n$ 
 $k$ : number of nearest neighbor
 $\theta$ : threshold for LOF
 $N$ : data block number
output data set which  $\text{lof}_i > \theta$ 
Initialize a Hadoop Job
Set TaskMapReduce class
Logically divide  $X$  into multiple data blocks:  $D_1, D_2, \dots, D_n$ .
In the  $j$ -th TaskMapReduce
FirstMapper
input  $D = d_1, d_2, \dots, d_m$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i, [(o_k, \text{dis}(d_i, o_k)), k - \text{dis}(d_i)] \rangle$ 
for each data  $d_i, i = 1, 2, \dots, m$  do
Calculate  $\text{dis}_{ij}$  = distance  $(d_i, d_j), j = 1, \dots, m$ 
Sort  $\text{dis}_{ij}$  of  $d_i$ 
for each  $\text{dis}_{ij}$  of  $d_i$  do
if  $\text{dis}_{ij} \neq 0$  &  $|k - \text{neighbour}| < k$ 
add  $d_i$  and  $\text{dis}_{ij}$  in  $k$ -distinct-neighbor record  $(o_k, \text{dis}(d_i, o_k))$ 
end
Calculate  $k$ -distinct-distance record  $k\text{-dis}(d_i)$ 
end
First Record
input  $\langle \text{key}, \text{value} \rangle = \langle d_i, [(o_k, \text{dis}(d_i, o_k)), k\text{-dis}(d_i)] \rangle$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i, [(o_k, \text{dis}(d_i, o_k)), k\text{-dis}(d_i)] \rangle$ 
SecondMapper
input  $\langle \text{key}, \text{value} \rangle = \langle d_i, [(o_k, \text{dis}(d_i, o_k)), k\text{-dis}(d_i)] \rangle$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i, [(o_k, \text{reach-dis}(d_i, o_k))] \rangle$ 
for  $o_k \in k\text{-distinct-neighbor}$  do
if  $k\text{-dis}(d_i) < \text{dis}(d_i, o_k)$ 
reach-dis  $(d_i, o_k) = \text{dis}(d_i, o_k)$ 
else reach-dis  $(d_i, o_k) = k\text{-dis}(d_i, o_k)$ 
end
SecondReducer
input  $\langle \text{key}, \text{value} \rangle = \langle d_i, (o_k, \text{reach-dis}(d_i, o_k)) \rangle$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i, \text{lrd}(d_i) \rangle$ 
for value do
 $\text{lrd}(d_i) = k / (\sum \text{reach} - \text{disk}(d_i, o_k)),$ 
 $o_k \in k\text{-distinct-neighbor}$ 
end
ThirdMapper
input  $\langle \text{key}, \text{value} \rangle = \langle d_i, \text{lrd}(d_i) \rangle$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i, (\text{lof}(d_i) > \theta), \text{lof}(d_i) \rangle$ 
for  $o_k \in k\text{-distinct-neighbor}$  do
 $\text{lof}(d_i) = (\sum (\text{lrd}(o_k)/k)) / \text{lrd}(d_i),$ 
 $o_k \in k\text{-distinct-neighbor}$ 
end
if  $\text{lof}(d_i) > \theta$ 
output
ThirdReduce
input  $\langle \text{key}, \text{value} \rangle = \langle d_i, (\text{lof}(d_i) > \theta), \text{lof}(d_i) \rangle$ 
output  $\langle \text{key}, \text{value} \rangle = \langle d_i^*, \text{lof}(d_i^*) \rangle$ 
for value do
Sort  $\text{lof}(d_i)$  for  $d_i$  and record  $d_i^*$ 
End

```

ALGORITHM 2: Compute $\text{lof}_i > \theta$

4.2. *Analysis of Algorithmic Efficiency.* Comparing the execution time of the DMAD-DVDMR algorithm with CNN, DeepAnt, and SVM-IDS in dealing with data sets of the same size, the efficiency of the DMAD-DVDMR algorithm is verified.

As shown in Figure 10, when the amount of data is large, the execution efficiency of DMAD-DVDMR is obviously better than other three algorithms. The reason Hadoop will schedule multiple MapReduce tasks in

FIGURE 7: AUC indicators of different algorithms on the KDD99 dataset.

TABLE 1: Data test result.

Result	Detected normal	Detected abnormal
Normal data	P_1 (positive 1)	P_0 (positive 0)
Abnormal data	N_1 (negative 1)	N_0 (negative 0)

FIGURE 8: Accuracy comparison of the algorithm.

parallel after partitioning the data. Compared with CNN, DeepAnt, and SVM-IDS, the efficiency of the proposed algorithm is improved by 57.75%, 78.95%, and 79.03%, respectively.

FIGURE 9: Response time of the algorithm.

FIGURE 10: Efficiency comparison of the algorithm.

4.3. Analysis of Algorithmic Execution Efficiency. In order to verify the scalability of the DMAD-DVDMR algorithm, this paper compares the execution efficiency under different computing nodes by expanding the data scale.

As shown in Figure 11, under the same data set size, with the increase of cluster computing nodes, the execution efficiency of the algorithm improves. Therefore, when the data set increases, the DMAD-DVDMR algorithm is extensible to improve the execution efficiency by expanding the computing nodes in Hadoop cluster.

FIGURE 11: Execution efficiency comparison of the algorithm.

5. Conclusion

Based on the deep analysis of data characteristics in a cloud computing environment, this paper proposes DMAD-DVDMR. Through the deep variational dimensionality reduction preprocessing and parallel anomaly detection of the data, this method meets the requirement of computing efficiency for large data. And also it alleviates the computational pressure, improves the execution efficiency under various nodes, and ensures the availability of data.

The next steps are as follows: (1) based on the proposed algorithm, the parameter settings in the algorithm are further optimized. The internal relationship of each parameter on the efficiency of the algorithm is analyzed, and the efficiency of the algorithm is improved further; (2) the factors leading to the fluctuation of accuracy are studied. In the process of system modeling, the above factors are considered to reduce the negative impact of irrelevant factors on the efficiency and availability of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was financially supported by the National Social Science Foundation of China. The project name is "Online Estimation and Whole Process Management of Short-Circuit Current Level in Active Distribution Network" (no. 15BGL040 (51577018)).

References

- [1] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly detection in roads with a data mining approach," *Procedia Computer Science*, vol. 121, no. 2, pp. 415–422, 2017.
- [2] Z. Xing and G. Wen, "A fast and adaptive method for determining K_1 , K_2 , and K_3 in the tensor decomposition-based anomaly detection algorithm," *IEEE Geoscience & Remote Sensing Letters*, vol. 15, no. 1, pp. 3–7, 2018.
- [3] M. Jones, D. Nikovski, M. Imamura, and T. Hirata, "Exemplar learning for extremely efficient anomaly detection in real-valued time series," *Data Mining and Knowledge Discovery*, vol. 30, no. 6, pp. 1427–1454, 2016.
- [4] L. Nie, D. Jiang, and Z. Lv, "Modeling network traffic for traffic matrix estimation and anomaly detection based on Bayesian network in cloud computing networks," *Annals of Telecommunications*, vol. 72, no. 5–6, pp. 1–9, 2017.
- [5] W. Sha, Y. Zhu, and M. Chen, "Statistical learning for anomaly detection in cloud server systems: a multi-order Markov chain framework," *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 401–413, 2018.
- [6] J. Ma, S. Le, and W. Hua, "Supervised anomaly detection in uncertain pseudoperiodic data streams," *Acm Transactions on Internet Technology*, vol. 16, no. 1, pp. 1–20, 2016.
- [7] Q. Schueller, K. Basu, M. Younas et al., "A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center," in *Proceedings of the 28th IEEE International Telecommunication Networks and Applications Conference (ITNAC 2018)*, IEEE, Sydney, Australia, November 2018.
- [8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, San Jose, CA, USA, May 2017.
- [9] D. Kwon, K. Natarajan, S. C. Suh et al., "An empirical study on network anomaly detection using convolutional neural networks," in *Proceedings of the IEEE International Conference on Distributed Computing Systems*, IEEE, Vienna, Austria, July 2018.
- [10] M. Munir, A. Siddiqui, and S. Ahmed, "DeepAnT: a deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
- [11] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, CVPR, Salt Lake City, UT, USA, June 2018.
- [12] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.
- [13] Z. Lei, C. Yong, and S. Liao, "Algorithm optimization of anomaly detection based on data mining," in *Proceedings of the 10th International Conference on Measuring Technology and Mechatronics Automation*, pp. 402–404, (ICMTMA), Changsha, China, February 2018.
- [14] G. K. Jha, N. Kumar, and P. Ranjan, "Density based outlier detection (DBOD) in data mining: a novel approach," *Recent Advances in Mathematics, Statistics and Computer Science*, vol. 22, no. 4, pp. 403–412, 2016.
- [15] R. Laxhammar and G. Falkman, "Online detection of anomalous sub-trajectories: a sliding window approach based on conformal anomaly detection and local outlier factor," in *Proceedings of the IFIP Advances in Information & Communication Technology*, pp. 192–202, Halkidiki, Greece, June 2018.

- [16] A. Bechini, F. Marcelloni, and A. Segatori, "A MapReduce solution for associative classification of big data," *Information Sciences*, vol. 332, no. 5, pp. 33–55, 2016.
- [17] K. Chen, "Indirect PCA dimensionality reduction based machine learning algorithms for power system transient stability assessment," in *Proceedings of the 2019 IEEE Innovative Smart Grid Technologies—Asia (ISGT Asia)*, IEEE, Chengdu, China, May 2019.
- [18] J. Liu, C. Li, and W. Yang, "Supervised learning via unsupervised sparse autoencoder," *IEEE Access*, vol. 6, pp. 73802–73814, 2018.
- [19] T. Fujiwara, J. K. Chou, Shilpika, P. Xu, L. Ren, and K.-L. Ma, "An incremental dimensionality reduction method for visualizing streaming multidimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 418–428, 2020.
- [20] G. Jianxiao, W. Hongli, G. Yarong, and Z. Zhiwen, "A new data mining method of iterative dimensionality reduction derived from partial least-squares regression," in *Proceedings of the 2009 Third International Symposium on Intelligent Information Technology Application*, pp. 471–474, Shanghai, China, December 2009.
- [21] M. Naseer and S. Y. Qin, "Performance Comparison of Nonlinear Dimensionality Reduction Methods for Image Data Using Different Distance Measures," in *Proceedings of the International Conference on Computational Intelligence & Security*, pp. 41–46, IEEE, Las Vegas, NV, USA, August 2008.
- [22] Z. Li, B. Dong, and R. Vega, "A hybrid model for electrical load forecasting—a new approach integrating data-mining with physics-based models," in *Proceedings of the ASHRAE Atlanta Conference*, Atlanta, GA, USA, July 2015.
- [23] D. Chunxia, Z. Yuhang, Y. Dong et al., "Hybrid model for renewable energy and loads prediction based on data mining and variational mode decomposition," *Iet Generation Transmission & Distribution*, vol. 12, no. 11, pp. 2642–2649, 2018.
- [24] F. Kong and X. Lin, "The method and application of big data mining for mobile trajectory of taxi based on MapReduce," *Cluster Computing*, vol. 22, no. 6, pp. 11435–11422, 2017.
- [25] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel data processing with MapReduce," *ACM SIGMOD Record*, vol. 40, no. 4, pp. 11–20, 2012.
- [26] N. M. F. Qureshi, I. F. Siddiqui, and M. A. Unar, "An aggregate MapReduce data block placement strategy for wireless IoT edge nodes in smart grid," *Wireless Personal Communications*, vol. 106, no. 2, pp. 2225–2236, 2018.
- [27] I. Palit and C. K. Reddy, "Scalable and parallel boosting with MapReduce," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1904–1916, 2012.
- [28] Q. He, F. T. Shang, and Z. Shi, "Parallel extreme learning machine for regression based on MapReduce," *Neurocomputing*, vol. 102, no. 15, pp. 52–58, 2013.
- [29] P. Chaudhari and S. Soatto, "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks," *Annals of Telecommunications*, vol. 101, no. 3, pp. 1–10, 2017.
- [30] I. Chakroun, T. Haber, and T. J. Ashby, "SW-SGD: the sliding window stochastic gradient descent algorithm," *Procedia Computer Science*, vol. 108, pp. 2318–2322, 2017.
- [31] X. Luo, D. Wang, M. Zhou, and H. Yuan, "Latent factor-based recommenders relying on extended stochastic gradient descent algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 99, no. 7, pp. 1–11, 2019.