Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4805035, 10 pages
https://doi.org/10.1155/2022/4805035

Research Article

Research on Security Anomaly Detection for Big Data Platforms Based on Quantum Optimization Clustering

Lijuan Deng, Long Wan, and Jian Guo

Zhaotong Power Supply Bureau of Yunnan Power Grid Co., Ltd, Zhaotong 657100, Yunnan, China

Correspondence should be addressed to Long Wan; 2006030328@st.btbu.edu.cn

Received 20 June 2022; Revised 20 July 2022; Accepted 23 July 2022; Published 26 August 2022

Academic Editor: Zaoli Yang

Due to the explosive growth of data in the Internet, more and more applications are being deployed on Big Data platforms. However, as the scale of data continues to increase, the probability of anomalies in the platform is also increasing. However, traditional anomaly detection techniques cannot effectively handle the massive amount of historical data and can hardly meet the security requirements of big data platforms. In order to solve the above problems, this paper proposes a security anomaly detection method for big data platforms based on quantum optimization clustering. Firstly, a framework of big data platform anomaly detection system is designed based on distributed software architecture through Hadoop and Spark big data open source technology. The system achieves effective detection of network anomalies by collecting and analyzing big data platform server log data. Secondly, an offline anomaly detection algorithm based on quantum ant colony optimized affinity propagation clustering is designed for various anomalies mined from historical data. The bias parameters of the affinity propagation clustering are treated as individual ants to construct an ant colony, and the clustering accuracy is set as fitness. Finally, in order to improve the accuracy of the optimal path search of the ant colony, quantum bit encoding is applied to the ant colony position to refine the granularity of the individual ant colony position update. The experimental results show that the proposed method can effectively complete the anomaly clustering detection of massive data. With a reasonable threshold, the quantum ant colony–based affinity propagation clustering has high detection accuracy.

1. Introduction

With the rapid development of big data technologies, big data platform architectures are becoming more complex, and the security requirements of big data platforms for new risks continue to increase. However, as the size of data and the functionality of modules continue to increase, the probability of anomalies in the platform grows [1–4]. For example, DDoS attacks on websites by hackers can bring down big data platform servers. Bursts of access traffic generated by large numbers of users during the festive season cause big data applications to crash. The proliferation of viruses and Trojan horses can lead to the leakage of personal information from applications [5, 6]. These security issues can cause incalculable financial losses to individuals and society. Therefore, accurate and timely detection of anomalies in big data platforms is of great practical importance.

Currently, the security of a Big Data platform is mainly provided by its infrastructure. However, due to the lack of necessary semantic interpretation and upper layer security mechanisms, the lower layer infrastructure is not capable of fully detecting anomalous events in the platform. For example, traditional firewalls and other security devices are unable to effectively detect and prevent anomalous events [7–9]. In order to achieve anomaly detection in big data platforms, we need to perform data mining on server logs and network traffic. Due to the fast, infinite, variable, and continuous nature of these massive data, it is very difficult to analyze these data manually. Over the years, research on automated anomaly detection has received much attention and has been widely used in areas such as intrusion detection, fault diagnosis, identity recognition, and e-mail filtering [10–14]. Trojans, viruses, and system vulnerabilities are now well resolved by software such as firewalls and

security assistants. However, there are still some limitations to traditional detection methods for network anomalies such as burst flow anomalies and DDoS attacks. This is because the data in the platform are highly variable and it is difficult to detect both of these anomalies completely with expert rule-based detection methods. In addition, software such as firewalls and security assistants can consume a lot of computational resources to detect large amounts of data. In summary, in order to ensure the security of big data platforms and to overcome the limitations of traditional anomaly detection algorithms, it is promising to study key technologies for anomaly detection in big data platforms.

The definition of anomaly refers to a special sample (pattern) which is inconsistent with most observed values in the observed set and produced by completely different mechanisms. For example, in regression algorithms, attribute values with significant deviations from expectations are regarded as anomalies [15, 16]. In statistical models, sample data that are distant from the series and do not obey the distribution are considered as anomalies. Assuming that there are qualitatively (or quantitatively) describable differences between normal and abnormal patterns, then anomaly detection is the process of identifying the differences that exist from the observed set of samples using statistical methods, data mining, and other theories. Anomaly detection algorithms are widely relevant and applicable to various fields, such as cyber-attack detection, credit card fraud detection, financial loan approval, medical drug research, etc. [17–20].

Early research into anomaly detection has mainly used misuse detection. Misuse detection constructs detection features based on existing network attacks and matches them to the corresponding attacks. The key step in misuse detection is the filtering and labeling of logs, which requires the extraction of important information from a large number of files. Therefore, misuse detection requires expert knowledge of appropriate WEB attacks to be able to detect predefined attacks. Misuse detection appears to be powerless against unknown anomalies. Anomaly detection methods mainly use normal WEB log data for analysis and training, and then build anomaly detection models that can distinguish unknown anomalous behavior. The current attack strategies faced by big data platforms mainly include [21–23]: DDoS, protocol vulnerability attacks, application service vulnerability attacks, and Trojan horses, which are all unknown anomalies. Therefore, Intrusion-Detection Systems (IDS) based on misuse detection cannot solve the security problems faced by Big Data platforms.

The anomaly detection data source of the Big Data platform is mainly based on server hosts and network traffic. Server host-based detection uses user logs and server access logs as the data source, and analyses them online or offline using relevant anomaly detection algorithms. Server host-based detection does not require external equipment and is insensitive to traffic data. The data source for network traffic-based anomaly detection is mainly network traffic from devices such as routers. Network traffic-based anomaly detection does not have access to the real-time status of the host system, resulting in less accurate detection.

There are currently three main types of anomaly detection algorithms [24, 25]: (1) Statistical analysis, (2) rule fields, and (3) data mining. Statistical analysis-based anomaly detection algorithms will assume that the data to be tested obeys a certain random distribution and identify anomalies through inconsistency detection methods. Said et al. [26] proposed a session anomaly degree calculation model based on the information of request URL, request time and other fields. Aissaoui's et al. [27] used log files to segment session attributes and used Bayesian parameter estimation to determine the session anomaly level. In the low-dimensional case, we can use statistical knowledge for distribution determination. However, when faced with large amounts of data, the data are usually high-latitude and therefore cannot be statistically analyzed.

Rule-based anomaly detection algorithms use expert experience to build up a rule base and complete a pattern matching process based on the rule base information to determine whether an anomaly is present. Rule-based anomaly detection algorithms are often used in misuse detection systems, such as the widely used Snort system, which has over 20,000 rules, each of which is a summary of expert experience. Metcalfe [28] construct a complete rule set based on the same elements in the data sequence under normal conditions. Sequences that are not identical to the rule set are defined as anomalies when detected. Rule-based anomaly detection algorithms are unable to detect unknown anomalies and require a priori knowledge to detect intrusion anomalies. However, in a Big Data environment, where anomalous attacks occur every day and expert experience is limited, building and maintaining a comprehensive rule base is a very difficult task.

Data mining–based anomaly detection algorithms preprocess the data to be tested and then extract the appropriate patterns from these data. If the extracted patterns do not match the normal behavior, they are categorized as anomalies. Data mining–based anomaly detection algorithms are divided into three main categories: detection algorithms based on correlation sequences, detection algorithms based on classification analysis, and detection algorithms based on cluster analysis, the most popular of which is cluster analysis, which belongs to the field of unsupervised detection. Anomaly detection algorithms based on cluster analysis will divide the data into multiple clusters and classify the anomaly clusters based on the similarity within and between clusters to perform the determination of anomaly detection. For example, Sunardi et al. [29] proposed a clustering-based anomaly detection algorithm for DDoS attacks that can automatically identify web crawlers. However, the method is too complex to train and has high complexity.

In summary, anomaly detection in big data platforms is a complex problem, especially for the anomalies present in massive historical data. Therefore, this paper proposes a quantum-optimized clustering-based anomaly detection method for big data platform security. The aim of the research was to accurately detect multiple anomalies present in the massive data of a big data platform in a reasonable time using an improved clustering analysis algorithm without the need for expert experience and rule bases.

The main innovations and contributions of this paper include.

(1) A framework for a big data platform anomaly detection system is designed on the basis of distributed software architecture through Hadoop and Spark open source technologies for big data. The system achieves effective detection of network anomalies by collecting and analyzing the server log data of the big data platform.

(2) To improve the performance of affinity propagation (AP) clustering, a quantum ant colony–based bias parameter optimization strategy is introduced, thereby enhancing the applicability of clustering. The similarity matrix and random values of the bias parameters are calculated after sample initialization. The selection paths are continuously optimized through changes in the pheromone values in the quantum ant colony optimization algorithm, resulting in stable clustering results.

The rest of the paper is organized as follows: In Section 2, the anomaly detection system framework for big data platforms are studied in detail, while Section 3 provides the offline anomaly detection method based on quantum ant colony optimised AP clustering. Section 4 provides the experimental results and analysis. Finally, the paper is concluded in Section 5.

2. Design of an Anomaly Detection System Framework for Big Data Platforms

The development of new open source technologies such as Hadoop/Spark has made it easy for Big Data–related companies to process and analyze all kinds of Big Data. Traditional anomaly detection systems are only oriented towards the anomaly detection segment and do not form an organic combination with the big data framework. Most of the traditional anomaly detection systems are rule-based anomaly detection algorithms. As there is no mature solution yet, it is important to study the system framework for big data anomaly detection and analysis.

2.1. System Logical Architecture Design. The objective of this paper was to design an intelligent security analysis system for big data platforms based on Hadoop and Spark technologies. The logical architecture of the system is shown in Figure 1. The intelligent security analysis system based on the Lambda framework consists of an offline processing layer, an online processing layer, and a service layer [30, 31]. The architecture combines the data detection functions of offline and online environments, and has the advantages of high fault tolerance, high scalability, and fast processing speed. In addition, the architecture supports various big data components such as Hadoop, Spark and kafka, and is suitable for the deployment of anomaly detection in big data platforms.

The offline processing layer is responsible for the storage and processing of large-scale data. This layer is mainly implemented using Hadoop and Spark. The collected historical data are stored on HDFS, while the data processing results are stored on HBase. The data are preprocessed and anomaly detected in an offline environment using Spark's efficient computing capabilities. The online processing layer is responsible for storing the incoming data streams with the distributed message cache kafka. The service layer is responsible for implementing fast user-interactive queries using Spark SQL.

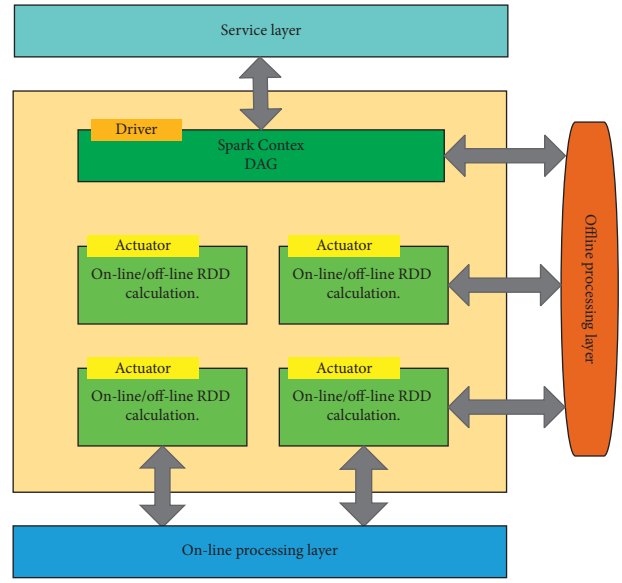

Figure 1: Overall system logic architecture.

2.2. System Physical Architecture Design. The physical architecture of the system is shown in Figure 2, which mainly accomplishes the tasks of log collection, log storage, and data processing. Firstly, the collection of standalone logs is accomplished by deploying Flume Agent on each server in the Big Data platform. Secondly, for the log storage task, the online streaming data are cached into kafka to ensure secure data transfer. For large batches of offline data, HDFS is used for storage. Finally, the Spark-Streaming technology in the Spark framework is used to process the real-time data. The offline data are then detected and analyzed using clustering algorithms.

The key modules of the system include the log preprocessing module, the monitoring and alerting module, and the data presentation module. The core function of the log preprocessing module is responsible for the collection of logs and the normalization of logs. In designing the log management module, this paper uses the Apache Flume open source log collection system to collect data from the WEB servers on the Big Data platform. The data collected include request logs from the WEB server, user access logs from the application server, and bulk WEB logs from the file server. The workflow of Apache Flume is shown in Figure 3.

The monitoring and alerting module is mainly divided into two parts: anomaly rule management and anomaly detection, the core part of which is anomaly detection, which is the focus of the full text. The proposed system uses

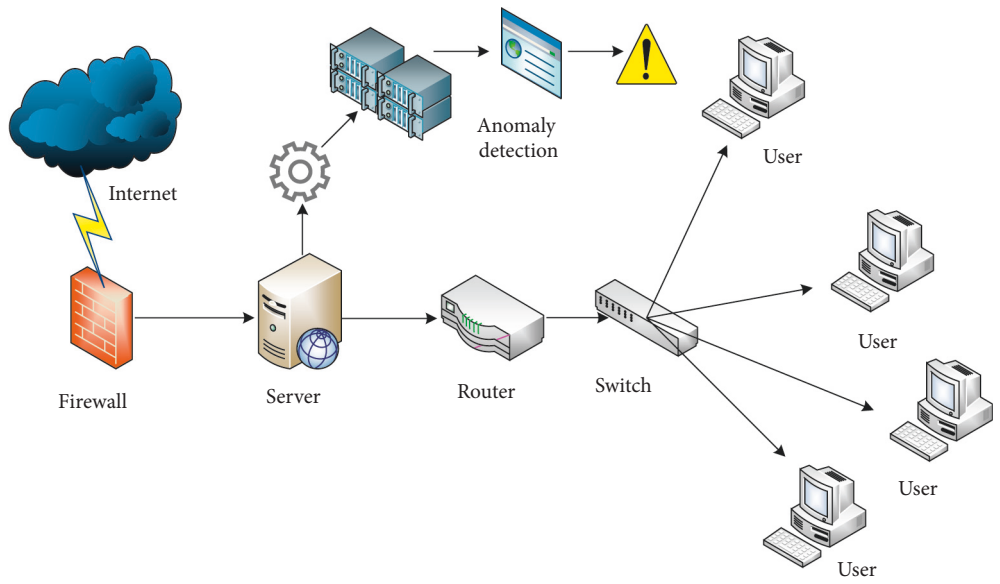

Figure 2: Physical system architecture.

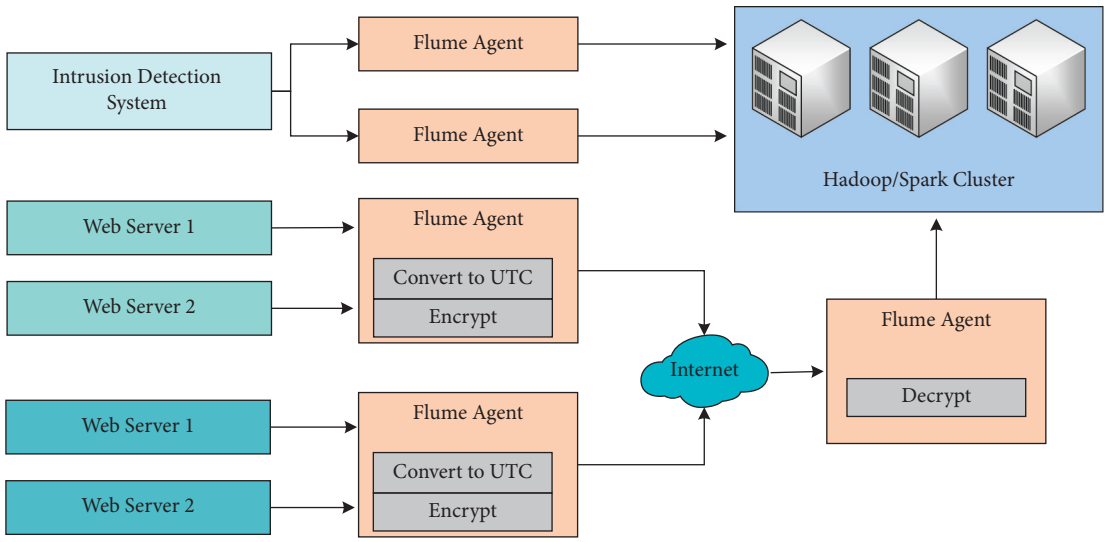

Figure 3: Workflow of Apache flume.

quantum optimization clustering algorithms to perform anomaly detection on offline historical data. The anomaly rule management part uses Snort as a subsystem for rule-based anomaly detection of online data. When an exception is detected, the system flags the exception and sends a processing message to the data presentation module for an exception warning.

The data presentation module uses a basic C/S (client/server) framework to deliver data to the front-end interface using the Json format. The module uses statistical charts to visualize the messages to the user. The data presentation module includes sub-modules for message push, data query, and data interface. WebSocket technology is used to implement active pushing of messages, taking into account the data transfer rate and bandwidth utilization. Spark SQL is used for fast and interactive queries.

3. Offline Anomaly Detection Based on Quantum Ant Colony Optimised AP Clustering

As a technique that does not require manual labeling, cluster analysis occupies an important position in data mining. Through clustering, hidden connections in massive amounts of data can be effectively mined, thus enabling value processing of large-scale data. As a kind of unsupervised learning, clustering can group similar objects into the same cluster. The more similar the objects within a cluster, the more effective the clustering will be. The advantage of anomaly detection algorithms based on clustering analysis is that the data categories do not have to be manually labeled, thus reducing the cost of the training process of the anomaly detection algorithm. Without the aid of a priori knowledge,

clustering often fails during data mining due to high dimensionality or heterogeneity.

The most commonly used anomaly detection algorithm based on cluster analysis is the K-Means algorithm [32]. However, the data sources in anomaly detection usually contain a large number of target objects, resulting in a large computational effort for anomaly detection based on the K-Means algorithm. Also, the selection of K values in the K-Means algorithm has to be tested repeatedly, thus further increasing the complexity of detection.

3.1. Affinity Propagation Clustering. The affinity propagation (AP) clustering algorithm [33] is a relatively new clustering method. Compared with traditional clustering methods, AP clustering algorithm does not require a predetermined number of clusters and has better clustering performance and efficiency. However, the accuracy of AP clustering is also often constrained by various factors such as the number of samples, the degree of sample balance, and the number of cluster centers. Therefore, in order to obtain better clustering results in large-scale data, the clustering method must be continuously improved according to the actual clustering needs.

Currently, there are more studies on optimized clustering algorithms. Shao et al. [34] used the density peaking algorithm to complete clustering and the whale algorithm for density peaking core parameter optimization to enhance the clustering performance. This study provides a new research direction for the improvement of clustering performance. Recently, various quantum population intelligence optimization algorithms have been proposed and have shown even better global and local optimization seeking capabilities. At present, no research has emerged on the use of quantum population intelligence optimization algorithms to enhance the performance of AP clustering.

AP cluster first needs to calculate that degree of similarity between two samples.

$$S(i,j) = -\left\|x_i - x_j\right\|^2, \tag{1}$$

where x_i and x_j represent the x dimensional functions of the samples i and j, respectively. The distance between any two of the samples is calculated for all samples and output as a matrix. The value on the diagonal of the matrix is called the bias parameter P.

In the similarity dimension function of a sample, $r(i,j)$ and $a(i,j)$ represent attract dimensionality and affiliation dimensions, respectively. Both can be represented in matrix form.

$$\begin{aligned} r(i,j) &= s(i,j) - \max_{j's.t.j' \neq j}\left\{a(i,j') + s(i,j')\right\},\\ \mathbf{R} &= [r(i,j)]_{N\times N},\\ \mathbf{A} &= [a(i,j)]_{N\times N},\\ a(i,j) &= \min\left\{0, r(j,j) + \sum_{i's.t.i' \notin \{i,j\}} \max\{0, r(i',j)\}\right\}, \end{aligned} \tag{2}$$

where $r(j,j)$ represents the attractiveness of j. Add $a(j,j)$ to both the left and right sides of equation (2).

$$r(i,j) + a(i,j) = s(i,j) + a(i,j) - \max_{j's.t.j' \neq j}\left\{a(i,j') + s(i,j')\right\}. \tag{3}$$

The calculation of the similarity of the AP clustering algorithm takes into account $r(i,j)$ and $a(i,j)$. The value of $r(i,j) + a(i,j)$ was chosen to measure the degree of similarity between i and j.

Let $\mathbf{E} = [e(i,j)]_{N\times N} = [r(i,j) + a(i,j)]_{N\times N} = \mathbf{R} + \mathbf{A}$. Keep optimally solving $\mathbf{E}$ to obtain the similarity of the samples to complete the clustering.

The factor ϕ ($\phi \in [0,1)$) is added to attenuate the oscillation effect in the update of $\mathbf{R}$ and $\mathbf{A}$. The calculation of $\mathbf{R}$ and $\mathbf{A}$ at the time of T is shown as follows:

$$\begin{aligned} \mathbf{R}_T &= (1-\phi)\mathbf{R}_T + \phi\mathbf{R}_{T-1},\\ \mathbf{A}_T &= (1-\phi)\mathbf{A}_T + \phi\mathbf{A}_{T-1}. \end{aligned} \tag{4}$$

3.2. Affinity Propagation Clustering for Quantum Ant Colony Optimization. In this paper, the AP clustering algorithm is used to implement data mining. In order to prevent the problem of degraded clustering performance due to unreasonable bias parameter settings, we introduced the ant colony optimisation (ACO) strategy [35]. The ant colony algorithm was used to optimize the bias parameters and quantum bit coding was used to encode the individual positions of the ant colony to improve the applicability of AP clustering. The essence of the ant colony algorithm is to find the optimum through ant path selection. For n nodes of m ants, the next node path is determined by the pheromone of the selectable nodes. When the ant k is at node position i and the set of optional nodes is M_i, then the next node is selected as follows:

$$J = \arg\max_{s\in M_i} \tau(i,s), \tag{5}$$

where $\tau(i,s)$ denotes the pheromone of the node s and node i. τ_0 denotes the initial pheromone. The pheromones in the ant path selection process were calculated in a probabilistic manner.

$$P_{ij} = \frac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}{\sum_{s\in M_a}[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}, \quad j \in M_a, \tag{6}$$

where $\tau(i,j)$ is the pheromone to (i,j), α is the coefficient of the pheromone, $\eta(i,j)$ is the inspired intensity, and β is the coefficient of the inspired intensity. The pheromone needs to be updated after the ant moves to the next node.

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho\tau_0, \tag{7}$$

where ρ is the evaporation factor.

$$\tau(i,j) = (1-\rho)\tau(i,j) + \Delta\tau(i,j), \tag{8}$$

where $\Delta\tau(i,j)$ indicates the best path pheromone value.

The bit representation of a quantum is shown as follows:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{9}$$

where $|\cdot\rangle$ represents the superposition state. $|\alpha|^2$ and $|\beta|^2$ represent the probability of quantum collapse to the "0" and "1" states, respectively.

$$|\varphi\rangle = [\alpha, \beta]^{\mathrm{T}}. \tag{10}$$

Let $\alpha = \cos(\theta)$ and $\beta = \sin(\theta)$.

$$|\varphi\rangle = \cos(\theta)|0\rangle + \sin(\theta)|1\rangle = [\cos(\theta),\ \sin(\theta)]^{T}. \tag{11}$$

We need to code all individual positions of the colony.

$$\mathbf{P}_i = \left[\begin{vmatrix} \cos(\theta_{i1}) \\ \sin(\theta_{i1}) \end{vmatrix} \begin{vmatrix} \cos(\theta_{i2}) \\ \sin(\theta_{i2}) \end{vmatrix} \cdots \begin{vmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{vmatrix} \begin{vmatrix} \cos(\theta_{\mathrm{im}}) \\ \sin(\theta_{\mathrm{im}}) \end{vmatrix}\right], \tag{12}$$

where $\theta_{ij} = 2\pi \cdot \mathrm{Rand}$, $\mathrm{Rand} \in (0, 1)$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, m\}$, n are the total number of individuals in the colony and m represents the location dimension.

Let θ be the phase of $[\alpha, \beta]^T$ on the component $|0\rangle = [1, 0]^{\mathrm{T}}$.

$$\begin{cases} |\varphi\rangle = \cos\ \theta + i\ \sin\ \theta = e^{i\theta} = r(\theta), \\ \cos^2\ \theta = |\alpha|^2,\ \sin^2\ \theta = |\beta|^2, \end{cases} \tag{13}$$

where θ meets $0 < \theta \leq (\pi/2)$.

3.3. Flow of Offline Anomaly Detection. The quantum ACO-AP clustering described above allows the dataset to be measured to be efficiently divided into clusters, and then the anomaly index is constructed according to the quantitative approach, thus completing the quantitative description of the anomaly clusters. For clusters with a large value of anomaly index, they can be regarded as anomalous clusters, which means that all objects in that cluster are anomalous data. The dataset is first clustered using quantum ACO-AP clustering, and then the anomaly index is calculated for each cluster. Then, the clusters are sorted according to the anomaly index and combined with the corresponding threshold judgement to finally identify the anomaly clusters, thus completing the anomaly detection. The flow of offline anomaly detection is shown below.

Step 1. Solving the similarity matrix for the offline historical data to be tested in the big data platform in order to initialize the bias parameters.

Step 2. Building several individuals of the ant colony optimisation algorithm with random values of the bias parameters.

Step 3. Quantizing the position of individual ants.

Step 4. Performing ant colony optimization to solve the ant colony individual with the highest fitness (optimal deflection parameter).

Step 5. Completing the clustering using the best bias parameter AP algorithm.

Table 1: Simulation sample set.

Datasets	Dimension	Number of samples	Number of categories
Wine	13	3791	3
Seeds	7	4621	3
Iris	4	5317	3
Flowers	12	3289	5
Glass	10	4433	6

4. Experimental Results and Analysis

In this paper, two sets of experiments are designed. The aim of the first set of experiments is to demonstrate that the quantum ACO-AP algorithm has better clustering performance than the K-Means algorithm. The purpose of the second set of experiments is to verify the effectiveness of anomaly detection based on the quantum ACO-AP algorithm. The experimental running environment experimental machine is a Lenovo desktop computer with Intel Core i7 processor, 3.30 GHz CPU, 16 GB memory, and CentOs 7.0 operating system type. The commonly used clustering accuracy and clustering Silhouette values are selected as the evaluation metrics for clustering performance in this paper.

4.1. Clustering Performance Evaluation

4.1.1. Clustering Performance with Different Bias Parameters. The first set of experiments used the UCI dataset, as shown in Table 1.

In AP clustering, the common bias parameter selection methods are the median and minimum of the similarity matrix. These three bias parameter selection strategies were used for clustering simulation, respectively, and the results are shown in Table 2.

It can be seen that the number of categories obtained by AP clustering when P = median is significantly greater than the other two methods. This is mainly because when P = median, too many class centres are obtained in the clustering process, which results in a significantly distorted number of categories. The number of classes obtained by AP clustering when P is equal to the minimum is also significantly more than the actual classes, which indicates that the conventional AP clustering algorithm is poorly adapted to the UCI dataset. Therefore, it becomes critical to optimize the bias parameters of AP.

The number of categories obtained by the quantum ACO-AP clustering algorithm is consistent with the actual categories, which indicates its strong adaptive capability. Next, the effects of the three bias parameter setting strategies on the AP clustering accuracy will be analyzed, and the results are shown in Figure 4.

It can be seen that the use of different bias parameter setting strategies has a greater impact on the Silhouette values. The Silhouette of the quantum ACO-AP algorithm is significantly higher than the other 2 algorithms in the 5-class dataset. The cross-sectional comparison revealed that each of the 3 algorithms obtained the highest Silhouette value in the Iris set. Therefore, the setting of the bias parameter has a greater impact on the performance of AP clustering. When it

Table 2: Number of clustering categories.

Datasets	Actual category	AP clustering		Quantum ACO-AP clustering
		P = median	P = minimum	
Wine	3	16	6	3
Seeds	3	15	10	3
Iris	3	26	11	3
Flowers	5	23	11	5
Glass	6	25	9	6

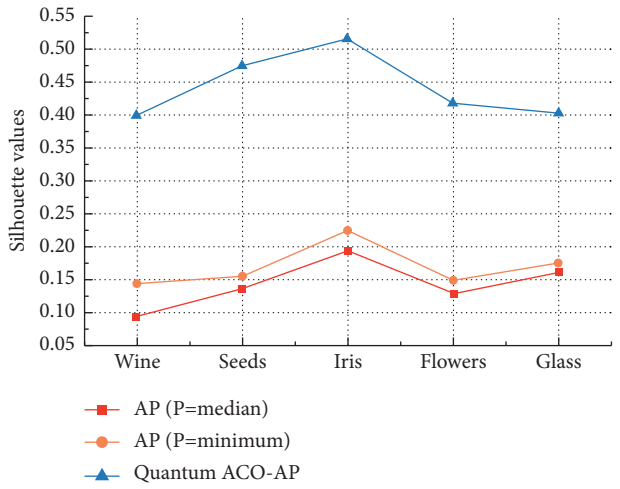

Figure 4: Silhouette values corresponding to different bias parameter setting strategies.

cannot be set manually and reasonably, it is more appropriate to use ACO algorithm to adaptively optimise the bias parameters.

4.1.2. Performance Verification of the Quantum ACO. To further verify the optimization performance of Quantum ACO for AP clustering, the clustering tests were carried out using the AP, ACO-AP, and Quantum ACO-AP algorithms, respectively, and the results are shown in Table 3 and Figure 5.

It can be seen that the clustering categories obtained using the AP algorithm are significantly larger than the actual categories, while the quantum ACO-AP and ACO-AP algorithms obtain clusters that are closer to the actual categories. This indicates that the clustering effect is significantly improved after the introduction of the ACO algorithm to optimize the bias parameters. The classes obtained by the quantum ACO-AP clustering are all equal to the actual classes, whereas ACO-AP only obtains the same results as the actual values on the wine and glass datasets. The biased parameter optimization performance of ACO was further improved by quantum bit encoding, resulting in higher accuracy for quantum ACO-AP clustering.

In terms of Silhouette performance, the AP algorithms all stayed below 0.25. The Quantum ACO-AP algorithm all stayed above 0.4, while the ACO-AP algorithm stayed between [0.3, 0.36]. This is because the QACO algorithm results in a more reasonable distribution between the

Table 3: Clustering categories of three algorithms.

Datasets	Actual category	Clustering categories		
		AP	ACO-AP	Quantum ACO-AP
Wine	3	6	3	3
Seeds	3	10	5	3
Iris	3	11	4	3
Flowers	5	11	6	5
Glass	6	9	6	6

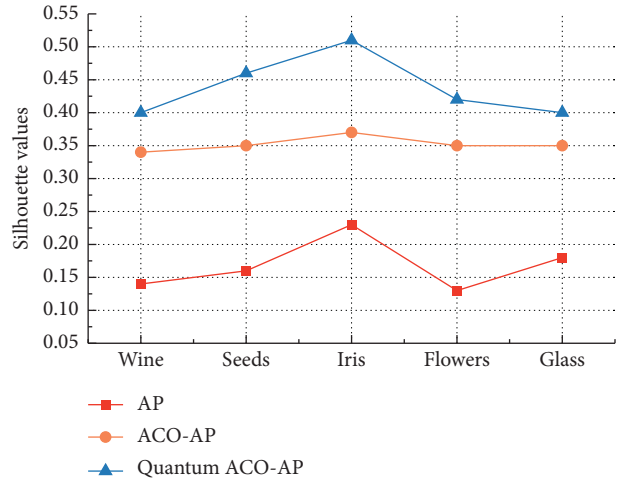

Figure 5: Silhouette values of three algorithms.

different cluster classes in the dataset. The performance comparison of the clustering accuracy of the three algorithms is shown as Table 4.

For the 5-class sample set, Quantum ACO-AP has the highest clustering accuracy. After the introduction of QACO, the performance of AP clustering was more stable. This is mainly due to the excessive number of class centres when the bias parameters of AP are not set appropriately, which tends to cause oscillations in the clustering results. A comparison of the convergence performance of the three algorithms is shown in Figure 6.

In terms of the number of iterations, Quantum ACO-AP requires fewer iterations, mainly because the AP algorithm requires more iterations to solve for the highest clustering accuracy before the bias parameters are optimized. However, with the quantum ACO algorithm, the efficiency of AP clustering is significantly improved. In terms of convergence curves, both ACO-AP and AP briefly fall into local optima, while the quantum ACO-AP algorithm has a very smooth downward trend.

Table 4: Clustering accuracy and performance of three algorithms.

Algorithms	Datasets	Accuracy	Standard deviation
AP	Wine	0.6953	$3.156e-5$
	Seeds	0.7326	$3.142e-5$
	Iris	0.6647	$3.113e-5$
	Flowers	0.7065	$3.246e-5$
	Glass	0.7328	$3.303e-5$
ACO-AP	Wine	0.7930	$2.341e-5$
	Seeds	0.8438	$2.361e-5$
	Iris	0.7727	$2.411e-5$
	Flowers	0.8224	$2.252e-5$
	Glass	0.8152	$2.273e-5$
Quantum ACO-AP	Wine	0.8237	$1.341e-5$
	Seeds	0.8982	$1.252e-5$
	Iris	0.7921	$1.411ee-5$
	Flowers	0.8556	$1.252e-5$
	Glass	0.8473	$1.273e-5$

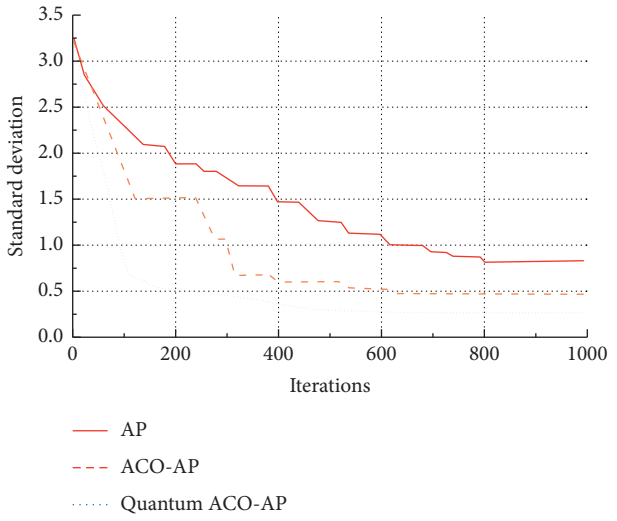

Figure 6: Convergence of three algorithms.

4.1.3. Clustering Performance of Commonly Used Algorithms. To further validate the clustering performance of the Quantum ACO-AP algorithm, it was compared with Decision tree, K-medoid, and K-Means, and the results are shown in Figure 7.

The cross-sectional comparison revealed that all four algorithms had the highest clustering accuracy in the Seeds set and generally poorer clustering accuracy in the Iris set. It can be seen that for the same sample set, the quantum ACO-AP algorithm has the highest clustering accuracy. With a comprehensive analysis of the above results, we can see that Quantum ACO-AP has a more obvious advantage in terms of clustering accuracy.

4.2. Anomaly Detection Results. The second set of experiments uses the KDD CUP 99 dataset, which contains mainly 4900000 network protocol connection records. Each record consists of 42 fields, where the 42nd field indicates whether the

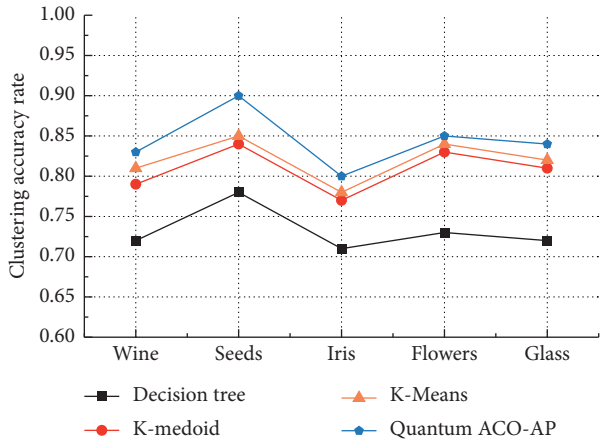

Figure 7: Clustering accuracy.

Table 5: Detection performance at different thresholds.

Threshold	Detection rate DR (%)	False detection rate FAR (%)
0.05	98.4	7.7
0.075	96.2	6.4
0.1	84.1	5.8
0.125	65.4	3.3
0.15	58.3	1.2

record is an anomaly or not. To check the effectiveness of the Quantum ACO-AP algorithm, 40,000 records were randomly selected from the dataset to form the dataset to be examined, namely, 38,447 normal records and 1,553 anomalous records.

Detection accuracy is usually evaluated using the detection rate and the false alarm rate. The detection rate DR is calculated as shown as follows:

$$\mathrm{DR} = \frac{\mathrm{Num}_{\mathrm{TP}}}{\left(\mathrm{Num}_{\mathrm{TP}} + \mathrm{Num}_{\mathrm{FP}}\right)}. \tag{14}$$

The false detection rate FAR is calculated as shown as follows:

$$\mathrm{FAR} = \frac{\mathrm{Num}_{\mathrm{TN}}}{\left(\mathrm{Num}_{\mathrm{TN}} + \mathrm{Num}_{\mathrm{FN}}\right)}, \tag{15}$$

where $\mathrm{Num}_{\mathrm{TP}}$ indicates the number of correctly detected abnormal samples, $\mathrm{Num}_{\mathrm{FP}}$ indicates the number of incorrectly detected abnormal samples, $\mathrm{Num}_{\mathrm{TN}}$ indicates the number of correctly detected normal samples, and $\mathrm{Num}_{\mathrm{FN}}$ indicates the number of incorrectly detected normal samples.

Firstly, different thresholds are used to detect KDD CUP99 experimental subsets, as shown in Table 5 and Figure 6.

It can be seen that the detection rate of the Quantum ACO-AP algorithm decreases when the threshold value is taken to be larger. However, the false detection rate also decreases at the same time. When threshold = 0.05, the detection rate can reach 98.4%, but the false detection rate is higher at this time. This indicates that the threshold value has a certain influence on the experimental results and should be chosen carefully. The best results are obtained when the threshold value is equal to [0.05, 1].

Table 6: Performance comparison results for anomaly detection.

	Quantum ACO-AP	PLC	CE
Detection rate DR	58%–98%	35.7%–88%	28%–93%
False detection rate FAR	1.2%–7.7%	1.44%–8.14%	0.5%–10%

Finally, experimental comparisons were performed on the KDD CUP99 dataset using the Quantum ACO-AP algorithm, the PLC algorithm, and the CE algorithm, respectively, and the comparison results are shown in Table 6.

It can be seen that the Quantum ACO-AP algorithm outperforms the other two anomaly detection algorithms in terms of both detection rate and false detection rate. In addition, the quantum ACO-AP algorithm has a smaller range of variation in detection rate and false detection rate. Combining the results of the first and second sets of experiments, we can conclude that the quantum ACO-AP algorithm has a higher detection performance for anomalies under reasonable threshold conditions.

5. Conclusion

Without the need for expert experience and rule bases, this paper uses an improved clustering analysis algorithm to accurately detect multiple anomalies in a large amount of data from a big data platform in a reasonable amount of time. The quantum ACO algorithm is used to optimally solve the bias parameters of AP clustering, which improves the accuracy of AP clustering. Reasonable settings of the main parameters of quantum ACO can obtain better bias parameter optimization of individuals and enhance the applicability of AP clustering. The experimental results show that compared with other anomaly detection algorithms, the quantum ACO-AP algorithm shows certain advantages in terms of both detection rate and false detection rate. The next step of the research will be to further optimize the core parameters of the quantum ACO algorithm in order to reduce the solution time of quantum ACO. An attempt is made to improve the real-time performance of large-scale sample processing by improving the clustering efficiency of the quantum ACO-AP.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu, and X. Shen, "Synergy of big data and 5G wireless networks: opportunities, approaches, and challenges," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 12–18, 2018.

[2] T. P. Liang and Y. H. Liu, "Research landscape of business intelligence and big data analytics: a bibliometrics study," *Expert Systems with Applications*, vol. 111, no. 11, pp. 2–10, 2018.

[3] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 32–38, 2018.

[4] Y. Chen and Y. Chi, "Harnessing structures in big data via guaranteed low-rank matrix estimation: recent theory and fast algorithms via convex and nonconvex optimization," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 14–31, 2018.

[5] G. R. Burmester, "Rheumatology 4.0: big data, wearables and diagnosis by computer," *Annals of the Rheumatic Diseases*, vol. 77, no. 7, pp. 963–965, 2018.

[6] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426–434, 2017.

[7] Y. Zhu and X. Liu, "Big data visualization of the quantification of influencing factors and key monitoring indicators in the refined oil products market based on fuzzy mathematics," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 5, pp. 1–11, 2020.

[8] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: approaches and challenges," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 94–101, 2018.

[9] X. L. Meng, "Statistical paradises and paradoxes in big data (I): law of large populations, big data paradox, and the 2016 US presidential election," *Annals of Applied Statistics*, vol. 12, no. 2, pp. 685–726, 2018.

[10] T. G. Kim and S. Yu, "Big data analysis of the risk of intracranial hemorrhage in Korean populations taking low-dose aspirin," *Journal of Stroke and Cerebrovascular Diseases*, vol. 30, no. 8, Article ID 105917, 2021.

[11] A. M. Al-Salim, T. E. El-Gorashi, A. Q. Lawey, and J. M. Elmirghani, "Greening big data networks: velocity impact," *IET Optoelectronics*, vol. 12, no. 3, pp. 126–135, 2018.

[12] A. Chehri, I. Fofana, and X. Yang, "Security risk modeling in smart grid critical infrastructures in the era of big data and artificial intelligence," *Sustainability*, vol. 13, no. 6, p. 3196, 2021.

[13] E. Vayena and A. Blasimme, "Health research with big data: time for systemic oversight," *Journal of Law Medicine & Ethics*, vol. 46, no. 1, pp. 119–129, 2018.

[14] S. Nadal, O. Romero, A. Abelló, P. Vassiliadis, and S. Vansummeren, "An integration-oriented ontology to govern evolution in Big Data ecosystems," *Information Systems*, vol. 79, no. 9, pp. 3–19, 2019.

[15] Y. Cao, H. Song, O. Kaiwartya et al., "Mobile edge computing for big-data-enabled electric vehicle charging," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 150–156, 2018.

[16] P. Tanuska, L. Spendla, M. Kebisek, R. Duris, and M. Stremy, "Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0," *Sensors*, vol. 21, no. 7, p. 2376, 2021.

[17] P. Cichosz, "Unsupervised modeling anomaly detection in discussion forums posts using global vectors for text representation," *Natural Language Engineering*, vol. 26, no. 5, pp. 551–578, 2020.

[18] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for

structural health monitoring," *Structural Health Monitoring*, vol. 18, no. 2, pp. 401–421, 2019.

[19] Y. M. Zhang, H. Wang, H. P. Wan, J. X. Mao, and Y. C. Xu, "Anomaly detection of structural health monitoring data using the maximum likelihood estimation-based Bayesian dynamic linear model," *Structural Health Monitoring*, vol. 20, no. 6, pp. 2936–2952, 2021.

[20] B. D. M. Lopes, L. C. B. Silva, I. M. Blanquet, P. Georgieva, and C. A. F. Marques, "Prediction of fish mortality based on a probabilistic anomaly detection approach for recirculating aquaculture system facilities," *Review of Scientific Instruments*, vol. 92, no. 2, Article ID 025119, 2021.

[21] L. Kulanuwat, C. Chantrapornchai, M. Maleewong et al., "Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series," *Water*, vol. 13, no. 13, p. 1862, 2021.

[22] L. Silva, M. Coutinho, and C. Santos, "Hardware architecture proposal for TEDA algorithm to data streaming anomaly detection," *IEEE Access*, vol. 9, pp. 103141–103152, 2021.

[23] R. Kabore, A. Kouassi, R. N'goran, O. Asseu, Y. Kermarrec, and P. Lenca, "Review of anomaly detection systems in industrial control systems using deep feature learning approach," *Engineering*, vol. 13, no. 01, pp. 30–44, 2021.

[24] T. Wei, Q. Guo, X. Zhang, C. Zhang, and W. Jing, "A nested residual encoder-decoder network for overhead contact system fastener anomaly detection," *IEEE Access*, vol. 9, pp. 74959–74968, 2021.

[25] I. Apostol, M. Preda, C. Nila, and I. Bica, "IoT botnet anomaly detection using unsupervised deep learning," *Electronics*, vol. 10, no. 16, Article ID 1876, 2021.

[26] A. M. Said, A. Yahyaoui, and T. Abdellatif, "Efficient anomaly detection for smart hospital IoT systems," *Sensors*, vol. 21, no. 4, Article ID 1026, 2021.

[27] O. E. Aissaoui, Y. E. A. El Madani, L. Oughdir, and Y. E. Allioui, "Combining supervised and unsupervised machine learning algorithms to predict the learners' learning styles," *Procedia Computer Science*, vol. 148, pp. 87–96, 2019.

[28] G. Metcalfe, "An Avron rule for fragments of R-mingle," *Journal of Logic and Computation*, vol. 26, no. 1, pp. 381–393, 2018.

[29] S. Sunardi, I. Riadi, and A. Sugandi, "Forensic analysis of docker swarm cluster using grr rapid response framework," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 2, pp. 459–466, 2019.

[30] L. Teng, H. Li, S. Yin, and Y. Sun, "A modified advanced encryption standard for data security," *International Journal on Network Security*, vol. 22, no. 1, pp. 112–117, 2020.

[31] S. Zeebaree, H. M. Shukur, L. M. Haji, and R. R. Zebari, "Characteristics and analysis of Hadoop distributed systems," *Technology Reports of Kansai University*, vol. 62, no. 4, pp. 1555–1564, 2020.

[32] B. S. Aski, A. T. Haghighat, and M. Mohsenzadeh, "Evaluating single web service trust employing a three-level neuro-fuzzy system considering k-means clustering," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 1, pp. 1–15, 2021.

[33] W. A. Lin, A. Sw, Y. B. Zhe, and P. C. Lu, "Analyzing potential tourist behavior using PCA and modified affinity propagation clustering based on Baidu index: taking Beijing city as an example - ScienceDirect," *Data Science and Management*, vol. 2, pp. 12–19, 2021.

[34] M. Shao, D. Qi, and H. Xue, "Big data outlier detection model based on improved density peak algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 9, pp. 1–10, 2020.

[35] B. Ikhlef, C. Rahmoune, B. Toufik, and D. Benazzouz, "Gearboxes fault detection under operation varying condition based on MODWPT, Ant colony optimization algorithm and Random Forest classifier," *Advances in Mechanical Engineering*, vol. 13, no. 8, pp. 4463–4478, 2021.