Hindawi
Security and Communication Networks
Volume 2023, Article ID 9831958, 1 page
https://doi.org/10.1155/2023/9831958

Retraction

Retracted: Method of Cumulative Anomaly Identification for Security Database Based on Discrete Markov chain

Security and Communication Networks

Received 5 December 2023; Accepted 5 December 2023; Published 6 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Z. Xu, T. Yang, and M. L. Najafi, "Method of Cumulative Anomaly Identification for Security Database Based on Discrete Markov chain," *Security and Communication Networks*, vol. 2022, Article ID 5113725, 10 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5113725, 10 pages
https://doi.org/10.1155/2022/5113725

Research Article

Method of Cumulative Anomaly Identification for Security Database Based on Discrete Markov chain

Zhiying Xu,[1] **Ting Yang**,[1] **and Moslem Lari Najafi**[2]

[1]*Shaoxing University Yuanpei College, Shaoxing 312000, China*
[2]*Pharmaceutical Science and Cosmetic Products Research Center, Kerman University of Medical Sciences, Kerman, Iran*

Correspondence should be addressed to Ting Yang; yangting663123@163.com and Moslem Lari Najafi; m.larinajafi@kmu.ac.ir

Received 7 July 2021; Revised 30 August 2021; Accepted 2 February 2022; Published 21 March 2022

Academic Editor: Chinmay Chakraborty

There exists an enormous volume of data in the database system, which is accountable for the storage of data and organization of data. The intruders can breach the security system of database and steal the important information. Therefore, it is of great significance to carry out the cumulative anomaly identification of the security database. In view of the shortcomings of traditional anomaly detection methods in detection performance and poor effect of anomaly recognition, this paper proposes a cumulative anomaly recognition method based on discrete Markov chain for security database. First, the sniffer is used to read the user access behaviour data, and then, it is processed, that is, standardized processing. Then, the segmentation method is used to extract the user behaviour features, and the normal feature data and abnormal feature data are obtained. Finally, the state sequence generated by the discrete Markov chain is used to calculate the state probability, which is used to evaluate the abnormal process behaviour. The proposed method in this paper is based on the Markov chain and can be used for better anomaly recognition. The results are obtained in terms of sensitivity score, precision score, and F1-score. The results are also compared with the results obtained by using some of the state-of-the-art traditional techniques. The comparison clearly indicated that the proposed method is more effective as compared to the tradition methods. The proposed method has the highest F1-score of 0.8586, and then the traditional methods have F1-scores of 0.7233, 0.8236, and 0.7562 for methods 1, 2, and 3, respectively.

1. Introduction

Data are becoming a powerful tool for businesses and organizations. Some of these data are worth millions of dollars, and companies take great effort to limit who has access to them, both within the company and outside the company [1]. When it comes to concerns of privacy of personal data, data security is also critical, and firms and organizations that manage such data must provide solid guarantees about the confidentiality of these data to comply with legal requirements and policies [2]. In the context of the information security system, data security plays a critical role. The availability of the data allows for an agile reaction to consumers searching for improved service that is critical for the administration of a business [3]. The proper deployment of a database in public organizations aids in the achievement of the goal, thus security measures must be implemented. Stealing of relevant information, duplication of records, denial of service, and the inability to get information on time are all issues that public entities face [4]. Cyber attackers are seeking a way in through a system breach and have a variety of tools for gaining access to an organization's systems or databases [5]. Theft of information, duplication of data, denial of service, and the difficulty to obtain information particularly heathcare data on time are all difficulties that public bodies confront [6]. Cyber intruders and hackers are looking for innovative ways to breach the security of the system and have explored several methods for breaching the security of the databases of corporate organizations [7].

The issue is that security models used in databases in public organizations are vulnerable to cyber-attacks because of flaws in their security management systems [8]. Breaches are unavoidable, threats have become more complex, and database security has become more difficult. Furthermore,

many threats are undetectable by traditional policy-based or rule-based security systems [9]. Firewalls, access control levels, and rule-based management are useless in circumstances of stolen privileged user accounts or internal attacks. As a result, there is a pressing need for a new technique that can detect harmful activity beyond the capabilities of rule-based systems. Any good security solution needs an intrusion detection system (IDS) to detect anomalous access. The software monitors network data and operating system operations for malicious activity or policy violations and generates reports [10].

1.1. Background Study. Anomaly detection is a technology that generates hints of possibly incorrect data and potentially dangerous processes. In the first stage, an anomaly detector analyses a system's usual state and behaviour and generates a set of reference for healthcare data that represents its unique qualities [11]. The same computations are then performed on the operational system, and the current set is compared to the reference set. The anomaly detector indicates an anomaly, i.e., an uncommon deviation, whenever the difference exceeds a certain threshold [12]. On systems with unambiguous patterns of regularity, anomaly detection works best, i.e., creates the fewest false hints and alerts. The most challenging aspect in designing an anomaly detection system (ADS) for networks and operating systems is identifying or extracting these patterns with well-designed relational databases [13]. Many of them are available for free to identify the anomaly. Anomalies are distinct from the rest of the data in the data set by their very nature. They can be separated from other data points in multidimensional Cartesian space. Anomalies will have a greater value than typical data points if the measurement of the average distance of the nearest N neighbors is obtained [14]. This attribute is used by distance-based algorithms to find anomalies in data.

The density of a neighbourhood data point is inversely proportional to its distance from its neighbours. Anomalies are found in low-density areas, while standard data points are found in high-density areas. The reason is that the relative frequency of an external user is small compared with the regular data point's frequency [15]. Data points with a low probability of occurrence are anomalies. Consequently, it is easier to discover the anomalous data points if the sample is fitted into a statistical distribution. For modeling the data set, it can be used to calculate the mean and standard deviation of a basic normal distribution. Anomalies in a data set differ by definition from the remainder of the data [16]. They are unusual data points separated from typical data points and usually do not form a close cluster. They still have a large distance from other clusters even when they join a group. Almost all classification techniques may be utilized to discover anomalies when previously categorized data are available [17]. When using the classification model, the availability of the previously marked healthcare data is an impediment. Since outlier data are unusual, it can be hard to find the anamoly [18]. Using oversampling the outer data with the remaining data, this problem can be partially overcome by stratified samples [19].

1.2. Related Work. A lot of anomaly detection technology was concentrated in operating systems and networks. In recent years, many techniques have been established due to the importance of privacy and security of personal information in database systems.

In [5], the authors have introduced a database security anomaly detection method. This means that the user's access pattern is checked in the database log and anomalous access events are detected. They evaluated the model based on the analysis of the user's pattern, the analysis of the machine learning, and the control of the rules. Casas et al. [6] have described Big-DAMA, a big data analytics framework (BDAF) for NTMA applications. Big-DAMA is a versatile BDAF, which evaluates and saves enormous quantities, both in streaming and batch mode, of structured and unstructured heterogeneous data sources. They have used Big-DAMA to detect various forms of network assaults and anomalies, comparing numerous supervised machine learning models. The assessments are made using the WIDE backbone networks based on real network measurements, and assaults are labeled with a known MAWILab data set. The experimental analysis have been compared to a normal Apache Spark cluster, and Big-DAMA can speed up computations by a factor of ten. Michele et al. [7] have drawn attention to important emerging challenges in the computer system and network security, particularly the Internet. Li et al. [8] have proposed a kind that user security auditing solution is based on a one-class support vector machine (OCSVM). The detection rate of 3 kinds of anomalous behaviour is above 80%, which shows a higher detection accuracy, according to simulation trials.

Ranganathan et al. [9] have used the Diffie–Hellman key exchange technique and the advanced encryption standards (AES) technique to implement the concept of differential privacy, which are both quite powerful in terms of speed. The tests were conducted with Laplace and Gaussian methods, which are the techniques currently most commonly used. The methods have been examined in the context of a case in which an initial and end location had been determined, and these had been encrypted using the aforementioned techniques while maintaining anonymity. Thudumu et al. [10] have attempted to chronicle the current state of anomaly detection in high-dimensional big data by utilizing a triangle model of vertices to represent the distinct challenges: the problem (large dimensionality), techniques/algorithms (anomaly detection), and tools (big data applications/frameworks specially pertaining to healthcare data). Furthermore, the limits of old methodologies and contemporary high-dimensional data strategies are explored, as well as recent techniques and applications on big data that are necessary for anomaly detection improvement. In [12], authors have introduced an anomaly detection method based on user behaviour into the internal attack detection in the database system to address the problem of internal attack in the database system. The anomaly detection of a database system was done using the discrete-time Markov chain (DTMC). The results suggest that the proposed approach can more accurately describe user activity and detect anomalies.

Even though databases include access control methods, these alone are insufficient to ensure data security. They must be supplemented by appropriate identification measures; the deployment of such techniques is critical for preventing impersonation attacks and dangerous code placed in applications. Additionally, anomaly detection procedures may aid in the prevention of insider threats, a growing problem in today's enterprises for which few solutions have been discovered. Although developing anomaly detection systems for networks and operating systems has been a hot topic of research, there are few anomaly detection systems specially designed for databases.

1.3. Need for the Research. The purpose of the research work presented in this paper is to investigate the construction of a database anomaly identification system to meet the need of the hour. There are two basic requirements for designing and developing such identification systems. One is the database application should not act as destructive element for the network and operating system used by an organization. The second and most crucial reason behind it is that the network and operating system capabilities cannot protect databases against the threats within the organization but can protect from threats from outside world. These threats are harder to detect, and it is difficult to protect the database since these threats are raised by the system administrators or users who have direct access to information and data.

1.4. Contribution of the Research. The contribution of this work is to design a cumulative anomaly detection system using discrete Markov Chain customized mechanism for database systems:

The discrete-time Markov chain (DTMC) has been used to detect anomalies in a database system

The sniffer is used to read user access behaviour data, which is then processed in a standardized manner.

The user behaviour features are extracted using the segmentation approach, and normal and aberrant feature data are acquired.

The state probability is used to evaluate anomalous process behaviour created by the discrete Markov chain

1.5. Organization of the Paper. This research is designed as follows: background, literature, as well as the study's goal and scope are provided in Section 1. The data and its representation are then defined in the subsequent part, followed by a description of the suggested anomaly detection approach in Section 2. The experiment design is provided in Section 3, and the findings are presented in Section 4. Finally, in Section 5, the findings are examined and conclusions are drawn, as well as future research directions.

2. Basic Definitions

The main process of cumulative anomaly recognition comprises of two main processes, viz training process and detection process [19], as shown in Figure 1. It is clear from Figure 1 that the training process has four main steps in the sequence data reading, processing, sequencing, and feature extraction. The detection process mainly consists of the user data gathering, characteristic extraction, comparison of characteristics with normal, and to detect the abnormality. Both the processes are discussed in detail as follows:

(1) In the process of training, make the database system run for a period of time under normal conditions, collect data during normal operation, extract user behaviour characteristics, and establish normal user behaviour mode (the established behaviour characteristics mode should include normal system behaviours).

(2) In the detection process, make the database system run in the real environment, gathers the behaviour data of the existing user and extract the behaviour characteristics, compare the behaviour characteristics of the detected user with the normal behaviour characteristics, and judge whether there is any abnormality by comparing the deviation degree between the normal and the current behaviour characteristics.

Figure 1 shows a framework of cumulative anomaly identification method for security database based on discrete Markov chain, which is divided into four parts:

(1) Data reading part: the reading object is the behaviour data generated when the user accesses the database.

(2) Data processing part: it is to process user access behaviour data.

(3) Feature extraction part: it is used to extract the feature information from the packets with known attack types and store it in the database.

(4) Feature comparison part: it is used to compare or match the captured package information and feature information in the feature database. If the match is successful, it is an anomaly and the response module is called for processing; if the match fails, it is normal.

2.1. Reading User Access Behaviour Data. When the user accesses the database, once the access behaviour occurs, the system immediately records a record in the cache, including digital ID/user account, access time, source IP, access page, source page, dwell time, and whether to leave or not. Figure 2 displays the construction process of the access record. It is clear from Figure 2 that data from "n" users is collected in data base, then the cookies account is generated and account, source IP, visit time, upper page, and stay time are recorded.

The access record records all the user's behaviours, so as long as these behaviour data are read, we can analyse whether the user has abnormal behaviours according to these data. At present, the main method of reading user access behaviour data is sniffer.

Sniffer is used as a software equipment that monitors the network data and mainly focuses on the legal management

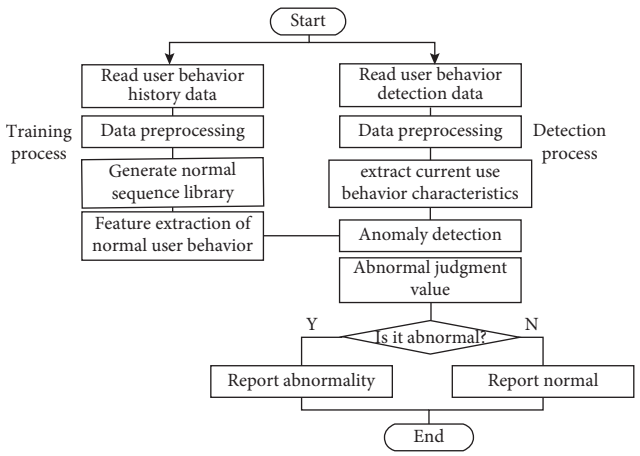

Figure 1: Process of cumulative anomaly identification of security database.

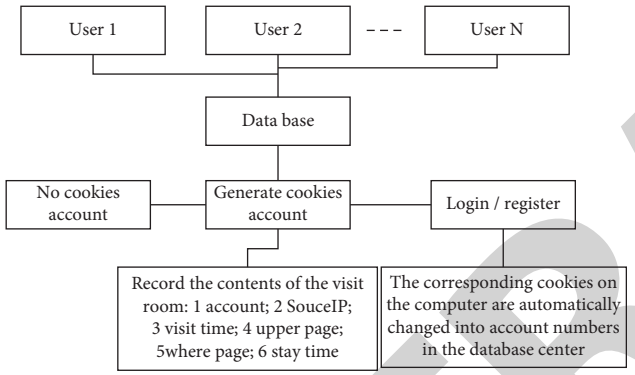

Figure 2: Formation process of access record.

of the networks. The management of Sniffer includes monitoring of the network traffic, analysis of the data packets, monitoring of the utilization of the network resources, implementation of security rules, diagnosing of network problems, and identification and analyses of network data. Sniffer is usually composed of four parts (Figure 3): (1) network hardware equipment; (2) monitor driver: to intercept data flow, filter and store data in buffer; (3) real-time analysis program: to analyse data contained in data frame in real time to find network performance problems and faults; it is different from intrusion detection system in that it focuses on network performance and fault, rather than on discovering hacker behaviour; (4) decoding program: to decrypt the received encrypted data, construct its own encrypted data package and send it to the network.

The sniffer used in this paper is a kind of sniffer designed with Win Pcap technology. WinPcap is derived from Berkeley's group capture library. It is mainly used in 32 bit windows operation platform. WinPcap is mainly used for packet truncation and filtering the captured packets.

WinPcap technology enables the user-level data package to operate under the common windows platform. WinPcap is a kind of architecture, which uses BPF model and Libpcap function library. WinPcap mainly consists of the following parts (Figure 4):

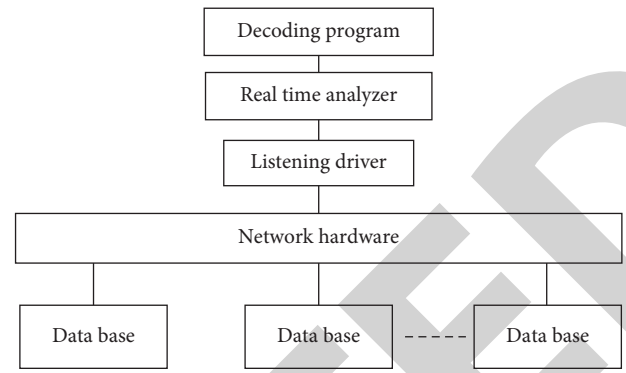

Figure 3: Structure of sniffer.

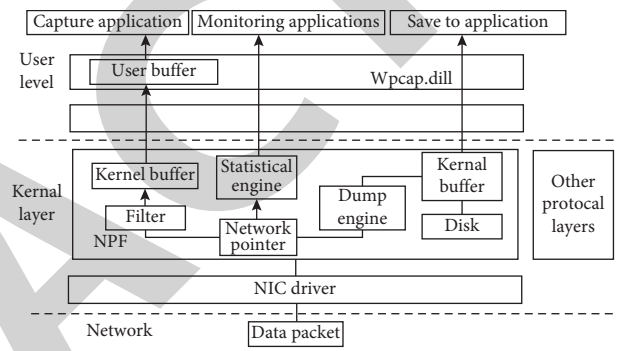

Figure 4: Structure of WinPcap.

NPF (Core Part). Net group Packet Filter, which is the network driver of the protocol, provides the function of intercepting and sending original packets for each operating system by calling NDIS. It is a virtual device driver file that filters packets and passes the original packets to the user.

Libpcap (Function Library). It is an upper level function library independent of the system and is more abstract.

Packet.dll (the Underlying Dynamic Link Section). It includes an application interface to access BPF and a function library conforming to the interface of high-level function library. Different operating systems have different kernels and user modules. This part provides a general interface for the platform in view of this phenomenon, thus saving the time of recompilation [16].

Among them, the underlying dynamic link part directly maps the kernel calls. In the dynamic link part, Wpcap.dll provides a more comprehensive and friendly function call. WinPcap's trump card lies in its standard interface for capturing packets. Moreover, WinPcap and Libpcap are compatible with each other. Therefore, for the network analysis tools supported by the original UNIX, it can be very compatible, which is very beneficial for development. At the same time, it also makes overall improvement in all aspects, making the operation more efficient. For example, it supports kernel level network packet filter and kernel state statistics mode.

WinPcap provides access to the bottom layer of the network on the application program of 32-bit operating system. It mainly includes the following aspects:

Interception function: it is used to effectively intercept the original datagram, mainly for all kinds of datagrams exchanged, sent and received by each host on the shared network

Filter function: it is used to provide user-defined rules, filter out the parts that meet the rules before sending the datagram according to the defined rules

Function of sending datagram: to support sending original datagram on shared network

Summary statistics function: in the process of active network communication, the collected information is summarized and counted

Figure 5 shows the flow chart of WinPcap sniffer reading user access behaviour data.

2.2. Data Processing. In order to make them comparable, it needs to use standardized methods to eliminate the deviation:

(1) Max-Min standardization/dispersion standardization:

Max-Min standardization, is also known as discrete standardization, is a linear transformation of the data and normalizing the values to [0,1]. The formula is shown in

$$x' = \frac{(x - \min)}{\max}, \tag{1}$$

where max represents the highest value of the sample and min represents the lowest value of the sample.

Deviation standardization keeps the relationship of the novel data and the normalized data. It is the method to eradicate the influence of dimension on the data range. The problem with this method is addition of new data that may cause changes of highest and lowest values in the sample and then the conversion function requires to be redefined [20].

(2) Z-score standardization/standard deviation standardization/zero mean standardization- Z-score is also a standard deviation standardization. The mean value is given by 0 of the processed data and the standard deviation value is 1. The formula is shown in

$$x' = \frac{(x - \mu)}{\sigma}, \tag{2}$$

where μ is the mean and σ is the standard deviation.

This method is not sensitive to outliers. It is very useful when the maximum and minimum values of the original data are unknown or the outliers control the Max-Min standardization. Z-score standardization is currently the most widely used standardization method [21].

(3) Log function conversion

By using the log function conversion, the scaling of data is also performed. The formula is shown in

$$x' = \frac{\log_{10}(x)}{\max}, \tag{3}$$

where max is the highest value of the sampling data.

2.3. Sequence Feature Extraction of User Behaviour. Feature extraction refers to the extraction of feature information from the data of known attack types and the behaviour data of current users. At present, there are multiple linear regression analysis algorithm and independent component analysis algorithm for user behaviour feature extraction. Among them, the former has a good filtering effect, but for large-scale information, the calculation process is more tedious, while the latter is within the error tolerance range, but takes a long time [22]. In view of the above situation, a user behaviour feature extraction based on time series is proposed in this section.

User access behaviour is a long series of sequential data in time sequence, so there must be some regularity, so as long as we grasp this regularity, that is to extract the sequence characteristics of user behaviour, we can achieve anomaly detection under the guidance of subsequent matching. At present, the method of feature extraction is mainly based on transform. Its principle is to transform the time series into the feature space and then use its feature mode to represent the time series. Its typical representatives are Fourier transform and discrete wavelet transform. However, this method can only be implemented on the premise of the same distribution of data groups. Once the data in the data flow are distributed differently, this method will lose its effectiveness [23]. In view of this situation, this section uses the segmentation method to extract the features of user access behaviour data. Compared with the traditional extraction method, the biggest feature of segmentation method is that it is faster and more accurate. The basic idea is the user behaviour sequence is separated into several segments and then the average value of each segment is determined. Finally, according to these average values, a vector is formed, that is, the feature representation after data dimensionality reduction, which is expressed as follows by mathematical formula:

Supposing that a time series is $G = \{g_1, g_2, \ldots, g_n\}$, where $g_.$ represents each data in the series and n is the number of data in the series, that is, the length of the series.

Let N represent the dimension of the feature space and $1 \le N \le n$, the time series with length is represented by the feature vector of N-dimension feature space as shown in

$$\overline{H} = \{h_1, h_2, \ldots, h_N\}, \tag{4}$$

where the ith element of $\overline{H}$ can be found out by

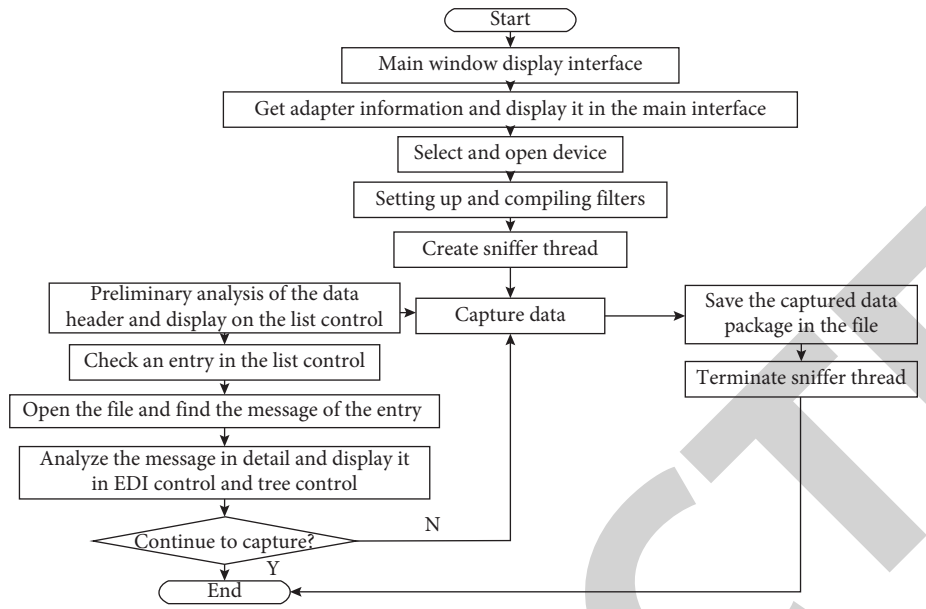

Figure 5: Flow chart of WinPcap sniffer reading user access behaviour data.

$$h_i = \frac{N}{n} \cdot \sum_{j=(N/n)(i-1)+1}^{(N/n)\cdot i} g_i. \tag{5}$$

Here, when $n = N$, the features of time series before transformation are the same as those after transformation; when $n = 1$, the features of time series after transformation are the same as the arithmetic mean of time series before transformation.

The above is the principle basis of segmented method for feature extraction. The following describes the specific process:

Step 1: set the input parameters, that is, determine the time series set and time series, the number of sequence segments k, and define the threshold value of local change mode.

Time series set:

$$G' = [G1, G2, \ldots, G_S]. \tag{6}$$

Time series:

$$G = \{g_1, g_2, \ldots, g_n\}. \tag{7}$$

Step 2: according to the frequent sequence of big data flow, the initial characteristic matrix is constructed as follows as shown in

$$F = \begin{bmatrix} (f_{11}, Q_1), & (f_{12}, Q_2), \ldots, (f_{1n}, Q_n) \\ (f_{12}, Q_1), & (f_{22}, Q_2), \ldots, (f_{2n}, Q_n) \\ & \ldots \\ (f_{m1}, Q_1), & (f_{m2}, Q_2), \ldots, (f_{mn}, Q_n) \end{bmatrix}. \tag{8}$$

Here, $f_i (i = 1, 2, \ldots, n)$ is the column vector of the characteristic matrix; $Q_i (i = 1, 2, \ldots, n)$ is the distance. The formula represents the local features of each variable dimension in each segment of the feature sequence F of user access behaviour data.

Step 3: divide each time series in the feature series of user access behaviour data into k subseries, as shown in

$$g_i = \{z_1, z_2, \ldots, z_i, \ldots, z_k\},\ i = 1, 2, \ldots, k, \tag{9}$$

where $z_1 = \{z_{b_1}, z_{b_2}, \ldots, z_{b_i}, \ldots, z_{b_{k+1}}\}$; $z_{b_i}, i = 1, 2, \ldots, k$ is the segmentation point.

Step 4: calculate the maximum value, minimum value, slope, and slope standardization value of the kth time series in the feature series of user access behaviour data. The formula is as follows:

Maximum value:

$$\max z_i = \max\{z_{b_1}, z_{b_2}, \ldots, z_{b_{i+1}}\}. \tag{10}$$

Minimum value:

$$\min z_i = \min\{z_{b_1}, z_{b_2}, \ldots, z_{b_{i+1}}\}. \tag{11}$$

Slope p_i:

$$p_i = \frac{z_{b_{i+1}} - z_{b_i}}{b_{i+1} - b_i}, \tag{12}$$

where $1 = b_1, b_2, \ldots, b_i, \ldots, b_{k+1} = n$

Slope standardization value a:

$$a = \frac{d - \overline{d}}{v_d}. \tag{13}$$

Here, d is the sequence feature; $\overline{d}$ is the average value of the sequence feature d; and v_d is the standard deviation of the sequence feature d.

Step 5: save the results from Step 4 above to the initial matrix.

Step 6: calculate the jump value of each subsequence after the k th time series is segmented.

Step 7: judge whether the jump value u between two adjacent subsequences is greater than the threshold e. If it is greater than e, continue to the next step, otherwise terminate.

Step 8: Add the subsequence larger than d into the initial feature matrix, and stsndardize it.

Step 9: Repeat the above steps, extract the mean value, variance and slope of each time in the frequent sequence set of big data stream, and then standardize them, and list them in the feature matrix to achieva sequence feature extraction.

2.4. Cumulative Anomaly Detection Based on DTMC. Markov process is a random process with no after effect. The so-called no after effect refers to that when the state of a random process at time t_0 is known, the state of a random process at time $t\,(t > t_0)$ is only related to the state of time t_0, but not to the state of a process before time t_0 [20, 21]. Those Markov processes with discrete time and state are called Markov chains, as shown in Figure 6.

Markov chain is a sequence of random variables with Markov property. If there is a random process $\{Y(t),\ t \in T\}$, the state of t at the time is Y_t, and the state of Y_{t+1} at $t+1$ is only related to the state of Y_t at t, but not to the state of $Y_{t-1}, Y_{t-2}, \ldots, Y_0$ at any time in the past, then $\{Y(t),\ t \in T\}$ is called Markov process. The state of Markov process is countable, as shown in

$$Z(Y_{t+1} = V_{t+1} | Y_t = V_t, Y_{t-1} = V_{t-1}, Y_1 = V_1) = Z(Y_{t+1} = V_{t+1} | Y_t = V_t), \tag{14}$$

where $V_1, V_2, \ldots, V_T \in (S_1, S_2, \ldots, S_N)$ is the value of the state and is called, as shown in

$$Y_{i,j}(t, t+1) = Y\left(Y_{t+1} = S_j | Y_t = S_i\right),\ 1 \le i,\quad j \le N. \tag{15}$$

$Y_{i,j}(t, t+1)$ is the probability of transition from state i to state j. i, j has N states, respectively. When $Y_{i,j}(t, t+1)$ has nothing to do with t, then Markov chain is called homogeneous Markov chain.

When the Markov chain is homogeneous and $Y_{i,j}(t, t+1)$ is recorded as b_{ij}, the state transition probability matrix is as follows, as shown in

$$B = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1N} \\ b_{21} & b_{22} & \ldots & b_{2N} \\ & \ldots & & \\ b_{N1} & b_{N2} & \ldots & b_{NN} \end{bmatrix}, \tag{16}$$

where $1 \le i, j \le N$ and $1 \le b_{ij} \le N$, $\sum_{j=1}^{N} b_{ij} = 1$ and B is called the state transition matrix.

It can be seen that matrix B represents the probability of state from t to $t+1$, but the probability of initial state distribution cannot be obtained. Therefore, in addition to matrix B, the initial probability vector $\pi = \{\pi_i\}$ must be obtained to represent the complete Markov chain process.

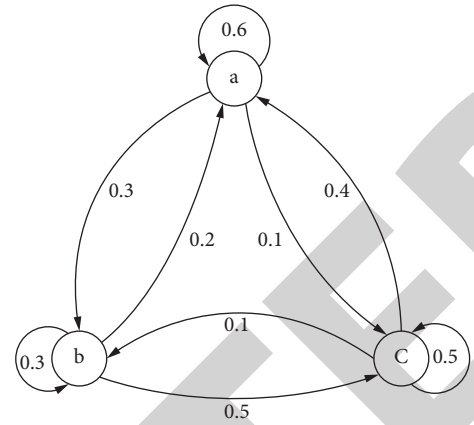

Figure 6: Markov chain.

$$\pi_i = Y\left(V_1 = C_i\right),\ 1 \le i \le N,\quad 0 \le \pi_i \le 1, k4 \sum_{i=1}^{N} \pi_i = 1. \tag{17}$$

In this case, (B, π) can represent a Markov chain.

On the basis of the above Markov chain principle, the cumulative anomaly recognition of security database is carried out, and the specific process is as follows:

Step: 1: execute a system call and add it to the end of the empty queue;

Step 2: match the system call sequence in the queue with the feature pattern in the feature library. If the sequence happens to be the feature pattern, go to Step 3; if the sequence matches a feature pattern, go to Step 1; if it cannot match, go to Step 4;

Step: 3: record the corresponding status number, add the status sequence, clear the queue, and go to Step 1;

Step: 4: add the status sequence corresponding to each system call in the queue, clear the queue, and go to Step 1.

The above steps are repeated until the end of the process. The system call sequence is transformed into a state sequence, the detection is based on the probability $p\,(L)$ of L consecutive states, and the method of local frame counting is used. The frame is a window with fixed length k [24, 25]. In the detection process, the frame window will slide forward with the detection point, which is used to record the number of k state sequences with probability less than the threshold v. The number of records less than the threshold v in the frame is counted here. When the count value is greater than 2, an anomaly is considered and an alarm is given.

3. Results

In order to check the viability and the effectiveness of the proposed cumulative anomaly recognition method for security database based on discrete Markov chain, it is compared with three anomaly recognition methods in

reference [3–5]. In this paper, the event log generated when the DARPA98 data set is replayed on the NT system is used as the experimental data for simulation experiment. The attack scenario of DARPA98 data set is shown in Figure 7.

The test data set of DARPA98 attack scenario comprises a series of attacks. The whole attack process is realized by DDoS attack. The invader first notices the active host through IP Sweep and then scans the port to find the host with sad-mind vulnerability. Then, the attacker attacks three hosts with this vulnerability: Pascal (172.16.112.50), Mill (172.16.112.20) and Locke (172.16.112.10) to make it a puppet machine. Then, the attacker installs the Trojan horse software to implement DDos attacks on the puppet machine and uses the controlled host to make DDos attack on the target.

3.1. Data Set. DARPA98 provided by Lincoln Laboratory of MIT is used as a data source. Because of the large amount of data, this experiment only selects part of the data for testing. In order to make the experiment comparable, five typical attacks are extracted as the experimental data of this model. Five attacks are Neptune (SYNFlooding), Satan, PortSweep, Buffe-overflow, and Guess-passwd. The attacks selected in this experiment include four categories of attacks, as shown in Table 1.

3.2. Development Environment. The development environment is java language platform (JDK1.6.2). It is an object-oriented programming language. This paper uses it as the development language mainly because it has the following characteristics:

(1) Java language is simple. Java discards redundant operations such as operator overloading, multiple-inheritance, and automatic cast. It does not make use of pointers.

(2) Java language is distributed. It supports the development of Internet applications by using network application programming.

(3) Java is portable language. In addition to it, Java strictly defines the length of each basic data type.

(4) Java language is multithreaded and provides the synchronization mechanism between multithreads (the keyword is synchronized).

3.3. Experiment Process. First, 60% of all data are used for training, including intrusion data and normal data; second, after the training, another 40% data are used to test the model; third, output results are generated.

3.4. Evaluation Index. The data in this paper can be divided into two categories after model detection, that is, positive data and negative data. Whether the payload data can be classified correctly is identified by true or false. The correct classification is true, and the error classification is false. Each model may produce four results for sample detection, which are, respectively, represented by TP, FP, TN, and FN, as shown in Table 2:

(i) TP indicates that the real category of data samples is positive, and the predicted outcome is also positive.

(ii) FP indicates that the real class of data samples is negative, but the predicted outcome is positive.

(iii) FN indicates that the real category of data samples is positive, but the final predicted outcome is negative.

(iv) TN indicates that the real category of data samples is negative, and the predicted outcome is also negative. According to the above indexes, precision and recall can be calculated, respectively.

Precision, the accuracy rate, indicates the probability of correct prediction of positive class in the prediction results and in the data samples of positive class, shown in

$$precision = \frac{TP}{TP + FP}. \tag{18}$$

TPR, also known as recall, indicates the probability of being correctly predicted as a positive class in the positive class of the original data sample, as shown in

$$TPR = \frac{TP}{TP + FN}. \tag{19}$$

In the experiment, we hope to get high precision and recall, but the precision and recall are mutually exclusive, so we need a compromise way Fl-score to express the effect of the experiment. Fl-score represents the harmonic average evaluation index of precision rate and recall rate, as shown in

$$Fl - score = \frac{2 \times TPR \times precision}{TPR + precision}. \tag{20}$$

3.5. Result Analysis. From Table 3, it can be observed that the proposed work in this paper is better than other methods in reference [3–5] in terms of cumulative anomaly recognition of security database, and the F1-score obtained is higher than the three anomaly recognition methods in other methods in reference [3–5], which shows that the recognition performance of the method in this paper is better. Table 4 shows precision achieved by all the methods, and the precision achieved by the method proposed in this article is the highest.

4. Discussion

In this paper, user behaviour anomaly recognition is establishing a normal behaviour mode of a legal user. By comparing the current behaviour and normal behaviour characteristics of the legal user, we can identify the abnormal behaviour. That is, if the present behaviour of the legal user deviates greatly from the normal behaviour characteristics in its history, it is considered that an anomaly has occurred. This anomaly may be caused by the unauthorized operation of the legal user itself, or by the illegal operation of other legal users or external intruders in the system. In the

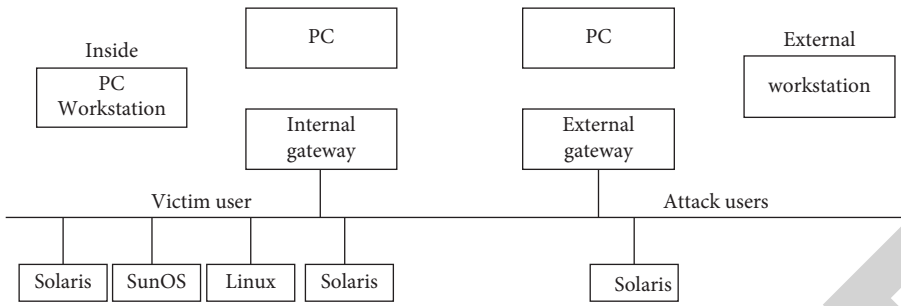

Figure 7: Attack scenario of DARPA98 data set.

Table 1: Attack data.

Attack categories	Attack selected in this experiment	Benign sample	Malicious samples
Dos	Neptune	31366	24225
Probing	PortsWeep	74463	62551
	Statan	35440	36641
U2L	Buffer-overflow	74450	82565
R2L	Guess-passwd	47123	36550

Table 2: Data classification.

Classification	Actual normal data	Actual malicious data
Forecast normal data	TP	FP
Predict malicious data	FN	TN

Table 3: F1-score.

Evaluation parameter	Article method	Reference [3] method	Reference [4] method	Reference [5] method
F1-score	0.8586	0.7233	0.8236	0.7562

Table 4: Precision obtained by various methods.

Evaluation parameter	Article method	Reference [3] method	Reference [4] method	Reference [5] method
Precision	0.92	0.75	0.85	0.78

database system, users mainly interact with the database management system through the access request to complete information query, modification, deletion, and other operations. Therefore, by analysing the execution sequence of the access request, we can more comprehensively explore the behaviour characteristics of users.

In order to improve the poor performance of traditional methods, this paper proposes a new method based on discrete Markov chain, which is proved to be more effective than traditional methods. The proposed method in this paper is based on the Markov chain and can be used for better anomaly recognition. The results are obtained in terms of sensitivity score, precision score, and F1-score. The results are also compared with the results obtained by using some of the state-of-the-art traditional techniques. The comparison clearly indicated that the proposed method is more effective as compared to the tradition methods. The proposed method has the highest F1-score of 0.8586 and then the traditional methods that have F1-score of 0.7233, 0.8236, and 0.7562 for methods 1, 2, and 3, respectively. The precision obtained by our method is 0.92, which is the highest among the comparative methods.

5. Conclusions

In this paper, a novel anamoly detection method based on discrete Markov chain is proposed to identify the cumulative anomaly in security database. This method not only considers the probability relationship between system calls but also considers the semantic relationship of system calls, that is, the short sequence of repeated system calls. After testing, the F1-score of the proposed method is higher than that of traditional methods, which proves the validity and feasibility of the method and achieves the purpose of research. This research provides a novel approach based on discrete Markov chain, which has been shown to be more successful than traditional methods in order to enhance the poor performance of traditional methods. This article's proposed method is based on the Markov chain and can be utilized to improve anomaly detection. The sensitivity score, precision

score, and F1-score are used to calculate the findings. The results are also compared to those acquired utilizing some of the most cutting-edge traditional methods. When compared to traditional methods, the comparison clearly showed that the proposed strategy is more effective. The proposed method has also achieved the highest precision among the techniques considered for comparative study. The proposed technique has an F1-score of 0.8586, which is higher than the standard methods, which have F1-scores of 0.7233, 0.8236, and 0.7562 for procedures 1, 2, and 3, respectively.

Data Availability

Data are available on request to the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. M. Toapanta, O. A. E. Quimis, L. E. M. Arellano, and R. Maciel, "Analysis for the evaluation and security management of a database in a public organization to mitigate cyber attacks," *IEEE Access*, vol. 8, pp. 169367–169384, 2020.

[2] E. Marina and N. M. Adams, "An anomaly detection framework for cyber-security data," *Computers & Security*, vol. 97, 2020.

[3] D. Kumar, J. C. Bezdek, S. Rajasegarar et al., "Adaptive cluster tendency visualization and anomaly detection for streaming data," *Acm Tran. on Knowledge Discovery from Data*, vol. 11, pp. 1–40.

[4] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A deep learning approach for anomaly detection and prediction in power consumption data," *Energy Efficiency*, vol. 13, no. 8, pp. 1633–1651, 2020.

[5] B. Abbasi, J. Calder, and A. M. Oberman, "Anomaly detection and classification for streaming data using PDEs," *SIAM Journal on Applied Mathematics*, vol. 78, no. 2, pp. 921–941, 2018.

[6] P. Casas, F. Soro, J. Vanerio, G. Settanni, and A. D'Alconzo, "Network security and anomaly detection with Big-DAMA, a big data analytics framework," in *Proceedings of the 2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, pp. 1–7, Prague, Czech, September 2017.

[7] V. Michele, C. Andrea, P. D. Elias, and A. Mahanti, "System and network security: anomaly detection and monitoring," *J. of Elec. and Comp. Engg*, vol. 2016, pp. 1-2, Article ID 2093790, 2016.

[8] Y. Li, T. Zhang, Y. Y. Ma, and C. Zhou, "Anomaly detection of user behavior for database security audit based on OCSVM," in *Proceedings of the 2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 214–219, Beijing, China, July 2016.

[9] C. Wu, S. Shao, and C. Tunc, "An Explainable and Efficient Deep Learning Framework for Video Anomaly Detection," *Cluster Computing*, 2021.

[10] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, no. 1, 2020.

[11] L. Jiang, S. R. Sakhare, and M. Kaur, "Impact of industrial 4.0 on environment along with correlation between economic growth and carbon emissions," *Int J Syst Assur Eng Manag*, 2021.

[12] I. M. Kulikovskikh, "Anomaly detection in an ecological feature space to improve the accuracy of human activity identification in buildings," *Computer Optics*, vol. 41, no. 1, pp. 126–133, 2017.

[13] I. Nevat, D. M. Divakaran, S. G. Nagarajan, P. Zhang, L. Su, and L. L. Ko, "Anomaly detection and attribution in networks with temporally correlated traffic," *IEEE/ACM Transactions on Networking*, vol. 26, pp. 131–144, 2016.

[14] L. Wang, L. Xu, Y. Xue, and G. Zhang, "Group behavior time series anomaly detection in specific network space based on separation degree," *Cluster Computing*, vol. 19, no. 3, pp. 1201–1210, 2016.

[15] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2315–2322, 2018.

[16] M. Lenning and J. Fortunato, T. Le, "Real-time monitoring and analysis of zebrafish electrocardiogram with anomaly detection," *Sensors*, vol. 18, p. 61, 2017.

[17] C. Wang, H. X. Yao, and X. S. Sun, "Anomaly detection based on spatio-temporal sparse representation and visual attention analysis," *Multimedia Tools and Applications*, vol. 76, pp. 6263–6279, 2017.

[18] M. Kaur and S. Kadam, "Bio-inspired workflow scheduling on HPC platforms," *Tehnički Glasnik*, vol. 15, no. 1, pp. 60–68, 2021.

[19] L. L. Zhang and C. H. Zhao, "A spectral-spatial method based on low-rank and sparse matrix decomposition for hyperspectral anomaly detection," *Infrared Physics & Technology*, vol. 38, pp. 4047–4068, 2017.

[20] D. Huang, J. B. Tristan, and G. Morrisett, "Compiling Markov chain Monte Carlo algorithms for probabilistic modeling," *AcmSigplan Notices*, vol. 52, pp. 111–125, 2017.

[21] G. Bayrak and E. Acar, "Reliability estimation using Markov chain Monte Carlo–based tail modeling," *AIAA Journal*, vol. 56, pp. 1–14, 2017.

[22] L. Ruff, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[23] X. Gu and P. Angelov, "Autonomous anomaly detection," *Evolving and Adaptive Intelligent Systems*, EAIS, Kings Lynn, UK, pp. 1–8, 2017.

[24] G. Zhao, Y. J. Liu, and Y. Shi, "Real-time assessment of the cross-task mental workload using physiological measures during anomaly detection," *IEEE Trans. on Human-Machine Systems*, vol. 48, pp. 149–160, 2018.

[25] X. Xu, H. Liu, and M. Yao, "Recent progress of anomaly detection," *Complexity*, vol. 2019, Article ID 2686378, 2019.