Nonlinear Dyn
https://doi.org/10.1007/s11071-023-08954-1

ORIGINAL PAPER

User-level malicious behavior analysis model based on the NMF-GMM algorithm and ensemble strategy

Xiu Kan · Yixuan Fan · Jinjie Zheng · Aleksey Kudreyko · Chi-hung Chi · Wanqing Song · Albina Tregubova

Received: 7 December 2022 / Accepted: 15 September 2023

Abstract In the security supervision sector, it is the importance of accurate detection and analysis of insider threats. In this article, we propose a new concept of insider threat kill chain, which is capable to understand psychological and behavioral change process of malicious users. Meanwhile, a novel user-level malicious behavior analysis model is established based on non-negative matrix factorization-Gaussian mixture model (NMF-GMM). In particular, we carry out the analysis from three perspectives: typical malicious behavior characteristics, overall user behavior and temporal individual behavior change. New classification method suggests to use group users by targeting malicious users with typical malicious features. The Z-score method is applied to establish evaluation model of suspicious user behavior, and the threshold of normal behavior is also determined. Furthermore, a temporal individual behavior change model is established, malicious users are located by the Pettitt test method, and the time of the first malicious behaviors are given. Experimental results show that the proposed user grouping method and ensemble strategy is capable for detection of malicious users.

Keywords Insider threat · User behavior modeling · Non-negative matrix factorization · Gaussian mixture model · Ensemble strategy

Abbreviations

NMF	Non-negative Matrix Factorization
GMM	Gaussian Mixture Model
NMF-GMM	Non-negative Matrix Factorization-Gaussian Mixture Model
CERT	Computer Emergency Response Team
IF	Isolation Forest
OCSVM	One-Class Support Vector Machine
R	Recall
P	Precision
F1	F1-Score
FPR	False Positive Rate
LOF	Local Outlier Factor
CH	Calinski-Harabasz Index
EM	Expectation–Maximization
HMM	Hidden Markov Model

X. Kan (✉) · Y. Fan · J. Zheng · W. Song
School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
e-mail: xiu.kan@sues.edu.cn

A. Kudreyko · A. Tregubova
Department of Medical Physics and Informatics, Bashkir State Medical University, Lenina st. 3, Ufa, Russia 450008

C. Chi
Strategic Centre for Research in Privacy-Preserving Technologies and Systems, Nanyang Technological University, Level 4, BorderX Block, Research Techno Plaza, 50 Nanyang Drive, Singapore 637553, Singapore

Published online: 12 October 2023

1 Introduction

1.1 Background and significance

Insider threat is typically attributed to legitimate users who maliciously leverage their system privileges and familiarity and proximity to threat computational environment to compromise valuable information or inflict damage [1–3]. Accounts of insiders provide different levels of access to organization's information [4–6]. In addition, insiders are highly trusted and familiar with the internal environment and resources of the enterprise. As the result, it is easier to bypass internal security controls and achieve data leaks or data destruction [7,8]. According to insider threat report, insufficient data protection strategies and solutions are the main reasons why insider threats are rising [9]. Based on the above performance, the sustainable monitoring of insider threats and the provision of strategies are important technical means to ensure security of enterprise's information assets.

1.2 Literature review

In the Internet era, users' operation records of data are reflected in a large number of log streams. Therefore, the problem of information extraction from user generated data into behavior characteristics is suitable for present challenges. Looking at the problem with much broader perspective, we also consider represented behavioral changes of users [10,11]. However, behavior data have inherent problem of data imbalance, i.e., the amount of malicious data is much smaller than normal data. To address the data imbalance issue, over-sampling and under-sampling techniques were utilized to balance data sets [12–14]. At the same time, malicious user behavior is reflected in unusual frequency of other operations such as files or websites. In order to distinguish normal users and abnormal users, the operation times of each user in each domain are counted in days, and different behavior frequency characteristics of users on each day are constructed [15,16]. It can be referred from the mentioned studies that the process of collecting log information reflects the change of behavioral statistics.

Insider threat detection is mostly based on machine learning algorithms. Through abnormal detection of user behavior model, it achieves the goal of distinguishing normal and malicious users. Clustering method is based on the relationship between users in the organization and e-mail communication. Consequently, behavioral characteristics of each user can be extracted. Subspace and subgraph clustering algorithms are used to identify non-clustered or sparse outliers [17]. The baseline of normal behavior is trained in terms of preceding period, which includes normal data. The time-dependent relationship between samples is captured by deviation between the actual and predicted values, and then it is compared with anomaly detection [18,19]. For the sake of modeling the user's normal behavior model, one-class learning method, HMM, and some distance measures based on abnormal detection are studied [20,21] Simultaneously, insider threat detection is defined as the embedded learning problem of heterogeneous event sequences, which combines the activities on the same day into event sequence and the data of the first two months are utilized to train the model for realizing the final anomaly detection [22,23].

1.3 Problem statement

The concept of "cyber kill chain" introduced by Lockheed Martin completely describes the path of external attacks [24], i.e., it provides defense idea and framework for dealing with various external attacks. The analysis of possible damage to system at each node of the kill chain is analyzed in refs [25,26]. Besides, with the purpose of improving the detection rate of malicious samples, over-sampling techniques and under-sampling techniques are usually combined to generate new samples, and some more easily misclassified instances are saved [27]. On the other hand, the lack of labeled insider threats or attacks brings increased challenges to direct application of supervised machine learning algorithms. This problem is known as unsupervised anomaly detection [28–31]. Since the insider threat problem is not unique and complex, there is a need for organizations in its full description and development of algorithms, which reveal malicious user activity.

1.4 Contributions

Inspired by previous works, the ultimate goal of insider threat detection is to find specific malicious users

instead of suspicious ones with abnormal behavioral characteristics mainly from the perspective of unsupervised methods. Although the user-level behavior model is reliable, the computing cost is also very huge. In this paper, a suspicious user-level behavior detection framework is proposed including NMF-GMM based on similar user behavior grouping and ensemble strategy consisting of three sub-strategies, with the main contributions as follows: A concept of insider threat kill chain is defined to understand the psychological and behavioral change process of malicious users. A clustering model NMF-GMM is designed to solve the high-dimensional inseparability problem of the original data. A malicious users detection ensemble strategy is designed including malicious user screening scheme, suspicious user search scheme and malicious user's location scheme. This suspicious user-level behavior detection framework monitors changes in the user's behavior status during practical application. When user behavior deviates from normal, it can be detected by managers in time. Therefore, it has the role of screening and management, which is of great significance to the management of the enterprise.

1.5 Plan of the study

The article is organized as follows. In Sect. 2, we examine characteristics of insider threat and introduce the process of insider threat kill chain, then we describe the framework of user behavior detection. In Sect. 3, we introduce the NMF-GMM algorithm for suspicious user selection and discuss the ensemble strategy consisting of three sub-strategies. Simulation and comparative results are given in Sect. 4 to illustrate the performance of the algorithm. Our conclusions are summarized in Sect. 5.

2 Model of user behavior

2.1 Description of the problem and data acquisition

CERT is an organization that performs activities such as vulnerability coordination, scanning for new threats, informing different organizations or individuals of new threats and vulnerabilities. Having a CERT-IT data is a way to have a centralized capability for analyzing security events, coordinate incidents and ensure that information on these incidents and events is conveyed to those who need it. CERT-IT data include information about normal behavior and threat behavior data, which were obtained from an enterprise environment by the insider threat center of Carnegie Mellon University in cooperation with ExactData company. This dataset consists of log records of 1000 users from January 2, 2010 to May 17, 2011. In particular, multi-domain log includes device usage (device domain), file operation (file domain), PC login (logon domain), website browsing (http domain), and e-mail exchange (e-mail domain). According to routine system operation, frequency of certain actions of each user can be extracted.

Considering that for ordinary users, the remarkable characteristics of abnormal behavior include unusual high-frequency operation times during rest time or even using other people's computers. In this paper, the normal working time range is set as 7:00 to 22:00. Forty behaviors are selected to analyze each user, and the frequency of each user's every behavior on each day is counted. Finally, 40 kinds of behavior frequency characteristic data of 1000 users in 501 days are obtained. Among the 40 behavioral characteristics, 19 characteristics during working time and 19 characteristics during rest time, as well as characteristics $countwiki\_access$ and $countkeylog$. Characteristics during working time and meanings are shown in Table 1. The 19 characteristics during rest time have the same meaning, and the time is 22:00–7:00. Characteristics $countwiki\_access$ and $countkeylog$ represent the number of times that the user visits the website wikileaks.org and downloads keylogging, respectively.

According to the insider threat scenario instances (CERT4.2), the constructed dataset is re-labeled. With the labeled dataset, malicious behavior of users with different roles is shown in Fig. 1. One can see that production line workers have the largest number of entries with abnormal behavior (247, 26%). Thus, it is essential to consider behavior analysis of production line workers.

By analyzing the data characteristics of production line workers, delete 11 characteristic quantities with values of 0 including $countopc\_workemail$, $countopc\_rest\text{-}e-mail$, $countopcweb\_work$, $countopcweb\_rest$, $countop\text{-}cconnect\_work$, $countopcconnect\_rest$, $countopc\_work\text{-}file$, $countopc\_restfile$, $countlogopc\_rest$, $countoffopc\_rest$ and $countkeylog$. Furthermore, Pearson correlation coef-

Table 1 Feature extraction of each user during working time

Region	Feature extraction
File	***countspc_workfile*** (Number of access to files on personal computer)
	countopc_workfile (Number of access to files on other computer)
E-mail	***countspc_workemail*** (Number of e-mails sent with personal computer)
	countopc_workemail (Number of e-mails sent with other computers)
	countinpeople_workemail (Number of e-mails sent to insiders)
	countoutpeople_workemail (Number of e-mails sent to outsiders)
	countattachments_work (Number of attachments sent)
	countinclude_attach_work (Number of e-mails sent includiing attachment)
Logon	***countlogspc_work*** (Logon times on personal computer)
	countlogopc_work (Logon times on other computers)
	countoffspc_work (Logoff times on personal computer)
	countoffopc_work (Logoff times on other computers)
	countvisitpc_work (Number of computers visited)
Http	***countspc_webwork*** (Number of web visits on personal computer)
	countopc_webwork (Number of web visits on other computers)
	countURLs_work (Number of websites visits)
Device	***countspcconnect_work*** (Number of connections on personal computer)
	countopcconnect_work (Number of connections on other computers)
	countpc_device_work (Number of computers connected)

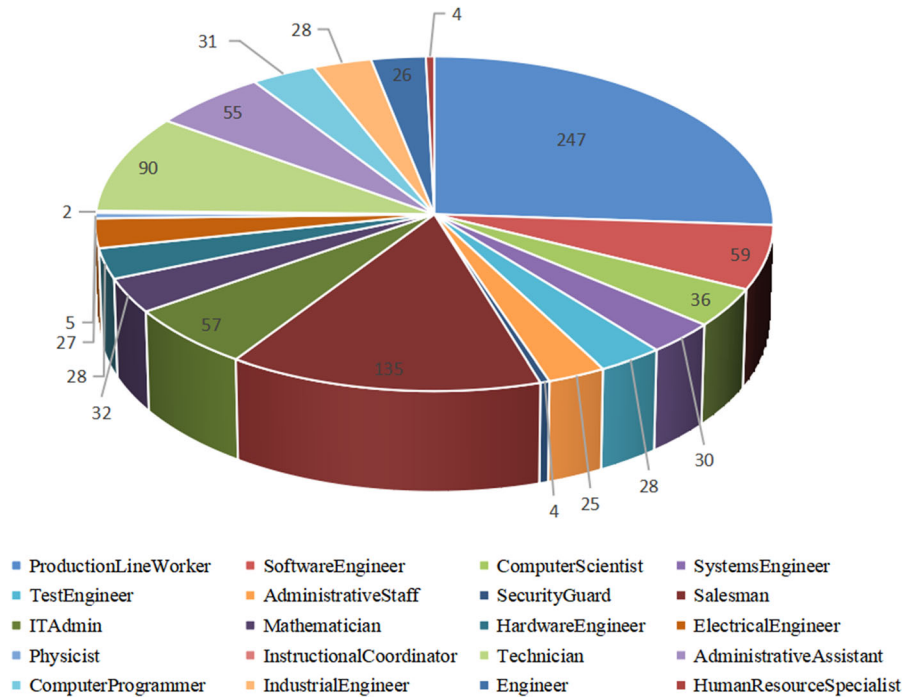


Fig. 1 Number of Malicious Behaviors in Each Position

Table 2 Insider threat kill chain

Grievances	Frequently browse job search websites and send complaint e-mails to superiorsor colleagues to reveal their dissatisfaction with the company and have the intention to leave.
⇓	
Reconnaissance	Access the shared network on the system, including personal information and key data from different departments of the organization. Detect accessible documents and analyze the weaknesses of the system.
⇓	
Weaponization	In order to successfully implement the transfer of confidential data and retaliation against the company, prepare removable storage devices and download keyloggers.
⇓	
Acquisition	Lock the insider threat target. Back up relevant confidential data, and obtain personal system login secret key and other information.
⇓	
Exfiltration	Masquerade supervisor's key to disturb the company's order by sending threatening e-mails. Leave the company with confidential data. Disclose or divulge the company's confidential data publicly.

ficient is used to reduce the feature redundancy, and remove $countoffopc\_work$ and $countURLs\_rest$. On the basis of the above operation, the data set of final production line workers is 90180 samples including 27 features. At this time, there are 89933 normal behaviors of 180 users and 247 malicious behaviors of 16 users.

2.2 User behavior detection framework

According to the characteristics of external attackers' intrusion systems, Lockheed Martin developed a cyber kill chain, which describes the attack route in detail in seven steps. Similarly, the occurrence of insider threat is essentially the evolution process of system information leakage and organization destruction, that is, the psychological and behavioral change process of malicious users. Aiming at the behavior characteristics of malicious users in each stage, this paper constructs a new insider threat kill chain, which is mainly divided into five stages as shown in Table 2.

From Table 2, we can see the evolution process of malicious activities of insiders. Based on the understanding of the change process of users' malicious behavior, aiming at the typical malicious characteristics of users in the exfiltration stage, this paper proposes an ensemble strategy including three sub-strategies to solve the problem of detection and analysis of malicious users. Among them, sub-strategy 1 first locks some malicious people according to the extremely malicious characteristics of publicly disclosed data; sub-strategy 2 considers the behavior evolution process of all employees and monitors the behavior change of malicious users compared with the baseline domain of normal behavior of all employees, from normal behavior in the early stage to violations in the later stage; sub-strategy 3 locates the abnormal behavior period from the user's own temporary individual behavior and analyzes the malicious evolution process of the corresponding behavior of the insider threat user. Therefore, this paper presents a novel user behavior detection framework which includes two parts.

(1) Data preprocessing

The data preprocessing phase mainly consists of two important steps: NMF is used to extract high-dimensional features and GMM clustering is applied to get low-dimensional suspicious clusters. Data preprocessing detects the range of suspicious abnormal users.

(2) Detection of malicious users: ensemble strategy

The ensemble strategy includes three progressive sub-strategies. Firstly, sub-strategy 1 locates malicious users according to the typical malicious characteristics. Secondly, sub-strategy 2 uses the behavior information of the first month to set the baseline of normal behavior, uses the Z-score to evaluate user behavior

changes, and narrates the range of suspicious users. Finally, sub-strategy 3 establishes temporal individual behavior change and uses the feature median and the Pettitt test method to judge malicious users.

3 Methodology: the study of user behavior

3.1 User grouping based on NMF-GMM algorithm

To summarize a large number of data and realize the subsequent grouping of user behavior data, GMM is optimized by NMF for clustering more similar suspicious behaviors.

(1) NMF

Non-negative matrix factorization is a technique for finding parts-based, linear representations of non-negative data [32]. The goal of this technique is to calculate approximate factorization of original data matrix into two non-negative matrices. Let the original non-negative data sample be $D_{n\times m}(n = 27, m = 90180)$, i.e., it can be factorized into two non-negative matrices:

$$D_{n\times m} \approx W_{n\times r} H_{r\times m} \tag{1}$$

where n denotes the feature dimension of the original data sample; m denotes the number of original data samples; r denotes dimension of sample feature reduction after NMF; $W = [w_1, w_2, ..., w_r]$ denotes the basis vector matrix; $H = [h_1, h_2, ..., h_m]$ denotes the coefficient matrix that the final m samples with dimension r.

For the sake of containing the essential characteristics of the original data samples with a small number of bases as far as possible, we usually set $r \ll in(m, n)$. With the iteration of updating $W_{n\times r}$ and $H_{r\times m}$ continuously, the distance between the reconstructed result $\hat{D}_{n\times m} \approx W_{n\times r} H_{r\times m}$ and the original data sample $D_{n\times m}$ is shortened. Lee proposed the quantitative measurement method with Euclidean distance, which is defined as follows:

$$\begin{aligned} &||D_{n\times m} - \hat{D}_{n\times m}||^2 = \sum_{j=1}^{m} (D_{ij} - \hat{D}_{ij})^2, \\ &i = 1, 2, ..., n \\ &j < m \end{aligned} \tag{2}$$

In order to conserve original information possible, the distance between the matrices of Eq. (2) must be minimized. Two non-negative matrices are updated as follows:

$$W_{ij} \leftarrow W_{ij} \frac{(DH^T)_{ij}}{(WHH^T)_{ij}}, i = 1, 2, ..., n, j = 1, 2, ..., r \tag{3}$$

$$H_{ij} \leftarrow H_{ij} \frac{(W^T D)_{ij}}{(W^T WH)_{ij}}, i = 1, 2, ..., r, j = 1, 2, ..., m \tag{4}$$

As shown in Eqs. (3) and (4), by using the known original non-negative data sample matrix, the basis vector matrix and coefficient matrix obtained in the last iteration, the new estimated basis vector matrix and coefficient matrix can be obtained when the loss function (2) is small enough.

Remark: The larger the value of r yields better the data fitting effect, and the small value of r give low complexity of the model. In general, the appropriate value r is selected according to the meaning of each basis vector in the basis matrix and the independence between the basis vectors. In this paper, cosine similarity is used to measure the correlation between basis vectors, so as to determine the appropriate value of r.

(2) GMM

Gaussian mixture model [33] assumes that feature sets are from different clusters and each cluster is subjected to different Gaussian distributions independently with the following probability distribution:

$$p(h|\theta) = \sum_{k=1}^{K} \alpha_k \varphi(h|\theta_k) \tag{5}$$

where α_k denotes the coefficient of the k-th cluster; $\alpha_k \geq 0, \sum_{k=1}^{K} \alpha_k = 1$; $\varphi(h|\theta_k)$ denotes the probability density function of Gaussian distribution; $\theta_k = (u_k, \sigma_k^2)$, and the k-th sub-model is defined as follows:

$$\varphi(h|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{(h-\mu_k)^2}{2\sigma_k^2}) \tag{6}$$

Therefore, originally feature sets can be mapped onto several Gaussian models. In addition, unknown parameters can be obtained from the EM algorithm.

Step1. Given the rough parameters $(\alpha_k, u_k, \sigma_k^2)$ of each Gaussian model, the probability of the sample generated by the k-th sub-model. When the sample h_j and the Gaussian distribution parameters $\theta_k = (u_k, \sigma_k^2)$ are given, the posterior probability $\hat{\gamma}_{jk}$ could be calculated as follows:

$$\hat{\gamma}_{jk} = \frac{\alpha_k \varphi(h|\theta_k)}{\sum_{k=1}^{K} \alpha_k \varphi(h|\theta_k)} \tag{7}$$

Step 2. Based on the estimated posterior probability, the parameters $(\alpha_k, u_k, \sigma_k^2)$ of the Gaussian model could be updated as follows:

$$\alpha_k = \frac{\sum_{j=1}^{m} \hat{\gamma}_{jk}}{m} \tag{8}$$

$$\mu_k = \frac{\sum_{j=1}^{m} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{m} \hat{\gamma}_{jk}} \tag{9}$$

$$\sigma_k^2 = \frac{\sum_{j=1}^{m} \hat{\gamma}_{jk} (y_j - \mu_k)^2}{\sum_{j=1}^{m} \hat{\gamma}_{jk}} \tag{10}$$

where m denotes the number of samples input into GMM. Keep iterating Step 1 and Step 2 until the maximum number of iterations is reached. Then, the parameters of each Gaussian distribution cluster are finally determined, so as to further determine the cluster to which each sample belongs. The division basis can be set as follows:

$$h_j \in \{i = k | argmax \gamma_{ji}, j = 1, 2, ..., m\} \tag{11}$$

(3) GMM clustering based on NMF samples

The goal of utilizing GMM is to group users with more similar behaviors. Because of the low discrimination of the original sample space, the original data cannot be grouped effectively when they are directly input into GMM. However, the original features are expressed linearly by NMF, and all the basis vectors contain the information of them as much as possible. Therefore, NMF is utilized here to optimize GMM user group learning, and the pseudo-code of the proposed NMF-GMM algorithm is described as Algorithm 1.

The decomposed low-dimensional feature samples are put into GMM, and the basis vector of each dimension includes part of the feature combination of the original data. Thus, the discrimination is better, which is convenient for the first step of suspicious behavior grouping. According to the distribution of the characteristics within the group given by the NMF-GMM algorithm, the most suitable group with dense distribution could be selected. The users who include the behavior in the selected group are suspicious users, which should be processed further. As well, the suspicious users' list $Suspicious\_list1$ can be achieved accordingly.

Algorithm 1 :NMF-GMM algorithm

Input: Original sample set $D_{n\times m}$, Number of basis vectors r, Number of Gaussian Mixture clusters K, NMF's maximum iterations $maxiter1$ and GMM's maximum iterations $maxiter2$.

1: Initialization: Matrix $W_{n\times r}, H_{r\times m}$ with any Non-negative element value, Gaussian Mixture Model's mixing coefficient α_k, Mean vector u_k, Covariance matrix σ_k.
2: **for** $i = 1 : maxiter1$ **do**:
3: Normalize the base vector of each column in $W_{n\times r}$.
4: Update each element in $W_{n\times r}$ according to Eq. (3).
5: Update each element in $H_{r\times m}$ according to Eq. (4).
6: Update $D_{n\times m} \approx W_{n\times r} H_{r\times m}$ and minimize the loss function value in Eq. (2).
7: **end for**
8: Transform the original high-dimensional sample set $D_{n\times m}$ into a low-dimensional data sample matrix $H_{r\times m}$.
9: **for** $i = 1 : maxiter2$ **do**:
10: **for** $j = 1 : K$ **do**:
11: Calculate the new mixing coefficient $\alpha_i^{'}$ according to Eq. (8).
12: Calculate the new mean vector $u_i^{'}$ according to Eq. (9).
13: Calculate the covariance matrix $\sigma_i^{'}$ according to Eq. (10).
14: **end for**
15: Update the Gaussian mixture model's parameters $(\pi_i, u_i, \sigma_i), i = 1, 2, ..., K$ to the new ones$(\pi_i^{'}, u_i^{'}, \sigma_i^{'})$, $i = 1, 2, ..., K$.
16: **end for**
17: **for** $i = 1, 2, ..., m$ **do**:
18: Mark the cluster to which each sample h_j belongs according to Eq. (11).
19: Divide the sample h_j into its corresponding cluster c_k.
20: **end for**

Output: Class cluster set $c_1, c_2, ..., c_K$.

3.2 Malicious users search strategy

In this paper, an ensemble strategy is proposed for searching for malicious users, which includes three sub-strategies to reduce the scope of suspicious user layer by layer. The details are described as follows:

(1) Sub-strategy 1: Malicious users search based on typical features.

The core asset of an enterprise is data information. One of the features extracted in this paper is related to data leakage, which is intolerable in the enterprise. When the value of feature $countwiki\_access$ is not 0, the user can be judged as a malicious user directly. The pseudo-code of sub-strategy 1 is described as Algorithm 2. As outlined in Algorithm 2, the suspicious user behavior data and filtered normal user behavior data are obtained by sub-strategy 1.

(2) Sub-strategy 2: Suspicious users search based on overall user behavior performance.

Algorithm 2 :Malicious users search based on typical features

Input: Original sample set $D_{n\times m}$, Suspicious users list after NMF-GMM: $Suspicious\_list1$

Initialization: Malicious users list:$malicious\_list = [\ ]$, Filtered normal user behavior data:$normal_data = [\ ]$, Suspicious user behavior data: $S1 = [\ ]$, Delete behavior samples with activity record 0 from D.
for $i = 1 : len(D)$ **do**:
  if $D.user[i]$ in $Suspicious\_list1$ **then**:
    Select i-th rows from D and add into $S1$.
  else:
    Select i-th rows from D and add into $normal\_data$.
  end if
end for
for $i = 1 : len(S1)$ **do**:
  if $S1.countwiki\_access[i] > 0$ and $S1.user[i]$ not in $malicious\_list$ **then**:
    Add $S1.user[i]$ into $malicious\_list$.
  end if
end for
for $i = 1 : len(S1)$ **do**:
  if $S1.user[i]$in $malicious\_list$ **then**:
    Delete i-th rows from $S1$.
  end if
end for

Output: Suspicious user behavior data: $S1$, Filtered normal user behavior data: $normal\_data$.

Based on practical experience and data analysis, users do not have any abnormal behavior in the first month of employment generally, and the working mode of users in the same profession is relatively fixed. Therefore, this paper chooses the data of the first month as the basic quantity for normal behavior analysis. Set an appropriate threshold and use the Z-score measure to further detect abnormal behaviors [34]. Then, an appropriate threshold is set, and all subsequent behaviors are monitored as a whole. For the sake of monitoring all users' behavior changes, a simple yet efficient change-detection algorithm based on the Z-score measure is utilized to automatically detect abnormal changes in user behavior in time series.

The median absolute value deviation of the j-th feature over the past N_{days} days can be calculated as follows:

$$mad^{j}_{N_{days}} = \frac{\sum_{i=1}^{N} abs(x_i^j - \text{median}(S1^{j}_{N_{days}}))}{N}, \quad j = 1, 2, ..., numf_1 \tag{12}$$

where $S1^{j}_{N_{days}}$ denotes the j-th behavior feature data involved in the past N_{days} days; N denotes the number of user behavior records involved in the past N_{days} days; x_i^j denotes the j-th feature value of the i-th behavior data, and $numf_1$ denotes the number of feature columns with the value not all zero.

The j-th feature value's Z-score for the i-th records after N_{days} days is defined as follows:

$$z_i^j = \frac{0.6745 \cdot (x_i^j - \text{median}(S1^{j}_{N_{days}}))}{mad^{j}_{N_{days}}}, \quad i = N+1, ..., len(S1) \tag{13}$$

The Z-score value is calculated on the basis of N behavior records to obtain $threshold1$:

$$z_i = \frac{0.6745 \cdot (x_i - \text{median}(S1_{N_{days}}))}{mad_{N_{days}}}, i = 1, 2, ..., N \tag{14}$$

$$Z = (z_{ij})_{N\times numf_1} = [z_1, z_2, ..., z_N]^T = [s_1, s_2, ..., s_{numf_1}] \tag{15}$$

$$threshold1 = (\text{median}(s_1), \text{median}(s_2), ..., \text{median}(s_{numf_1})) \tag{16}$$

where Z denotes the Z-score matrix of the normal sample in the past N_{days} days; $\text{median}(S1_{N_{days}}) = (\text{median}(S1^{1}_{N_{days}}), ..., \text{median}(S1^{numf_1}_{N_{days}}))$, $mad_{N_{days}} = (\text{median}(mad^{1}_{N_{days}}), ..., \text{median}(mad^{numf_1}_{N_days}))$, $s_j = [s_{1j}, s_{2j}, ..., s_{Nj}]$; $z_i = (z_{i1}, z_{i2}, ..., z_{inumf_1})$ and $x_i = (\text{median}(x_i^1), ..., me\,dian(x_i^{numf_1}))$.

For each user behavior in data $(S1 - S1_{N_{days}})$, the number of behavioral violations per record compared to the $threshold1$ is shown as follows:

$$AS1(i) = \sum_{j=1}^{numf_1} sign(z_i^j - threshold1(j)), \quad i = N+1, ..., len(S1) \tag{17}$$

where $sign(x) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$

Then, the maximum violation tolerance for each user behavior record is set as $threshold2$. If $AS(i) > threshold2$, this behavior is more suspicious. The pseudo-code of sub-strategy 2 is shown as Algorithm 3, which clearly describes how to further reduce the scope of suspicious users from the overall behavior change.

(3) Malicious user's determination based on temporal individual behavior change.

Temporal behavior information reflects the user's working behavior habits. Besides, abnormal behavior

Algorithm 3 :Suspicious users search based on overall user behavior performance

Input: Original sample set $D_{n\times m}$, Suspicious user behavior data: $S1$, Filtered normal user behavior data: $normal\_data$, Maximum violation tolerance for each user behavior record: $threshold2$.

Initialization: Suspicious behavior data: $S2 = [\,]$, Suspicious users list: $Suspicious_list2 = [\,]$, Receive $S1_{N_{days}}$ behavior data in the first N_{days} days from $S1$.
for $i = 1 : len(S1_{N_{days}})$ **do**:
  for $j = 1 : numf_1$ **do**:
    Calculate the median absolute value deviation $mad^j_{N_{days}}$ of the j-th feature over the past N_{days} days (see Eq. (12)).
  end for
  Calculate $threshold1$ of the user's normal behavior record according to Eqs (14), (15) and (16).
end for
for $i = (N + 1) : len(S1)$ **do**:
  Calculate the number of behavioural violations per record $AS1(i)$.
  if $AS1(i) > threshold2$ and $S1.user[i]$ not in $Suspicious\_list2$ **then**:
    Add $S1.user[i]$ into $Suspicious\_list2$.
  end if
end for
for $i = 1 : len(S1)$ **do**:
  if $S1.user[i]$ in $Suspicious\_list2$ **then**:
    Select i-th row from $S1$ and add into $S2$.
  else:
    Select i-th row from $S1$ and add into $normal\_data$.
  end if
end for

Output: Suspicious user behavior data:$S2$, Filtered normal user behavior data:$normal\_data$, Suspicious users list:$Suspicious\_list2$.

can be detected through behavioral changes. An individual abnormal behavior analysis sub-strategy 3 is developed based on the Pettitt test method.

Firstly, judge the change range of feature of each user. For the remaining suspicious users, $T_{N_{days}}$ is taken as the time window, and the moving step size is 1. The comparison threshold $threshold3$ of the range of user behavior changes is shown as follows:

$$\begin{aligned} &threshold3(j) = max(abs(a_k^j(l) - a_k^j(l-1))),\\ &\quad l = 1, 2, ..., N_{u_k} - T_{N_{days}} + 1,\\ &\quad k = 1, 2, ..., len(normal\_user) \end{aligned} \tag{18}$$

$$\begin{aligned} &a_k^j(l) = \text{median}(x_k^j(l), x_k^j(l+1),\\ &\quad ..., x_k^j(l + T_{Ndays})) \end{aligned} \tag{19}$$

$$\begin{aligned} &threshold3 = (threshold3(1), ...,\\ &\quad threshold3(numf_2)) \end{aligned} \tag{20}$$

where N_{u_k} denotes the total number of behavior records for the normal users; a_l^{kj} denotes the median of the j-dimensional feature in $T_{N_{days}}$ days from day l for the k-th user; x_l^{kj} denotes the j-dimensional feature for the k-th user on day l; $numf_2$ denotes the total number of features for normal behaviors.

Furthermore, compared with the $threshold3$, the violation times matrix $AS2$ of behavior changes range for all users is recorded as follows:

$$\begin{aligned} AS2(k, j) = &\sum_{i=1}^{N_{u_k} - T_{N_{days}} + 1} sign(abs(a_k^j(l)\\ &- a_k^j(l-1)) - threshold3(j)),\\ &k = 1, 2, ..., len(Suspicious\_list2),\\ &j = 1, 2, ..., numf_2 \end{aligned} \tag{21}$$

where M_{u_k} denotes the total number of behavior records of the k-th user to be tested in $Suspicious\_list2$.

Elements in matrix $AS2$ that are not zero correspond to suspicious characteristics of suspicious users. Next, the Pettitt test method is further used to detect malicious user behavior.

The Pettitt test [35] as a nonparametric test is used to detect the single unknown mutation point. Given the time series X_0 of a certain feature of each user, the null hypothesis of the test is made as H_0: the average value of the original time series $X(t)$ does not change. The data before and after τ are compared based on rank. If the characteristics of the test change on a certain day τ, the null hypothesis H_0 is rejected. Therefore, Pettitt statistic is defined as $k(\tau)$, and the judgment basis is given as follows:

$$k(\tau) = \sum_{i=1}^{\tau} \sum_{j=\tau+1}^{n} sgn(x_j - x_i) \tag{22}$$

$$\text{where} \begin{cases} sgn(x_j - x_i) = 1, if(x_j - x_i) > 0\\ sgn(x_j - x_i) = 0, if(x_j - x_i) = 0\\ sgn(x_j - x_i) = -1, if(x_j - x_i) < 0 \end{cases}$$

Furthermore, the rejection of the null hypothesis H_0 by significance probability P can be approximated as follows:

$$P \approx 2 \times exp[-6K^2(i^3 + i^2)] \tag{23}$$

where $K = \max\limits_{t\leqslant\tau\leqslant i}(|k(\tau)|)$. Meanwhile, the corresponding mutation time position is given as follows:

$$T_{mal} = \operatorname*{argmax}_{t\leqslant\tau\leqslant i}(|k(\tau)|) \tag{24}$$

As calculated in Eq. (23), if the result P is smaller than 0.05, the user characteristic behavior changes significantly. Therefore, each suspicious user with suspicious features could be tested whether malicious or not. The pseudo-code of sub-strategy 3 is outlined as Algorithm 4.

Algorithm 4 :based on Temporal individual behavior change for detection of malicious users

Input: Filtered normal user behavior data:$normal\_data$, Suspicious users list:$Suspicious\_list2$, Suspicious user behavior data:$S2$, Malicious users list:$malicious\_list$.

1: Initialization: Receive normal users from $normal\_data$: $normal\_user = distinct(normal\_data.user)$. Calculate the user behavior changes range $threshold3$ according to Eqs. (18), (19) and (20).
2: **for** $k = 1 : len(Suspicious\_list2)$ **do**:
3: **for** $j = 1 : numf_2$ **do**:
4: Calculate the violation times matrix $AS2$ of behavior changes range for each user according to Eq. (21).
5: **end for**
6: **end for**
7: **for** $k = 1 : len(Suspicious\_list2)$ **do**:
8: **for** $j = 1 : numf_2$ **do**:
9: **if** $AS2[k, j]! = 0$ **then**:
10: Select behavior data from the time series $X(t)$ from $S2.column[j]$ where $S2.user == Suspicious\_list2[k]$.
11: Calculate P according to Eqs. (22) and (23).
12: **if** $P < 0.05$ **then**:
13: Add $Suspicious\_list2[k]$ into $malicious\_list$.
14: **end if**
15: **end if**
16: **end for**
17: **end for**

Output: Malicious users list:$malicious\_list$.

4 Experimental analysis and discussion

We have chosen to focus on the CERT 4.2 dataset (see Section 3.1). The experimental test environment is: Intel(R) Xeon(R) Silver 4208 processor, Windows 10 operating system, and the programming environment used is the training and verification of the NMF-GMM algorithm performed on Python 3.7.

4.1 User grouping results of NMF-GMM

As given in Section 3.1, the first step is to cluster likely suspicious behavior. After decomposition of the original data set, the basis vectors have high-dimensional features. The training results of the original features with non-negative matrix factorization are related with the number of basis vectors. Let $r = 3$, $r = 4$ and $r = 5$, respectively, and the default value $maxiter1$ is 200. The expression distribution results of the original features are shown in Figs. 2, 3 and 4.

In principle, for the sake of expressing the original features completely, the similarity between the basis vectors is required to be as low as possible. Table 1 shows the statistics of cosine similarity of the basis vectors with different values of r.

From Figs. 2, 3 and 4 and Table 3, when the value of r is equal to 4 or 5, the cosine similarity of the basis vectors is relatively low. Considering the low complexity and low similarity of the basis vector, setting $r = 4$ is more suitable.

The effectiveness of user grouping is reflected in the clustering effect of GMM. The CH is one of the clustering algorithms evaluation measures [36]. The CH index is calculated as a ratio of the sum of inter-cluster dispersion and the sum of intra-cluster dispersion for all clusters:

$$CH = \frac{trB/(K-1)}{trW/(K-m)} \tag{25}$$

$$trB = \sum_{i=1}^{K} m_j ||u_j - u||^2 \tag{26}$$

$$trW = \sum_{j=1}^{K} \sum_{i=1}^{m} ||x_j - u_j||^2 \tag{27}$$

where trB denotes the trace of deviation matrix between groups; n_j denotes the number of elements in group j; u denotes the average distance between all sample data sets; u_j denotes the average distance between samples in group j; trW denotes the trace of deviation matrix within groups; m denotes the number of samples in the dataset, and K denotes the number of clusters.

A high CH index means a better clustering effect, that is, inter-group distance is large, and intra-group distance is small. The results based on different K values are shown in Table 4.

As given in Table 4 and Fig. 5, when $K = 8$, the CH index achieves relatively better value. Therefore, we suggest to set $K = 8$ as the appropriate number of clusters. The maximum number of iterations $maxiter2$ of GMM is set to 100. Algorithm 1 is implemented to

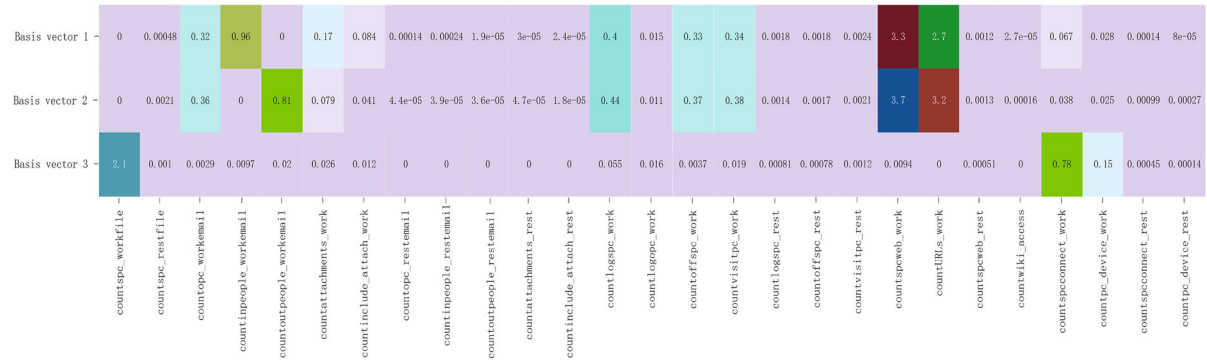


Fig. 2 Distribution of the original features in $r = 3$

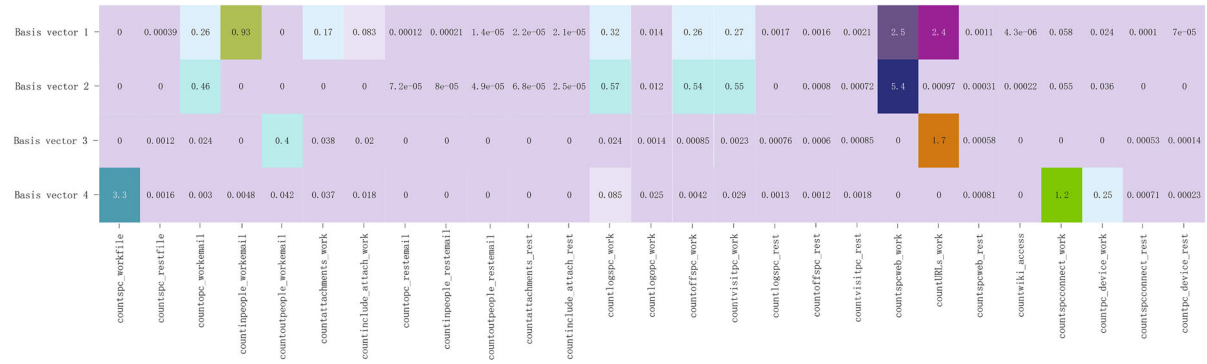


Fig. 3 Distribution of the original features in $r = 4$

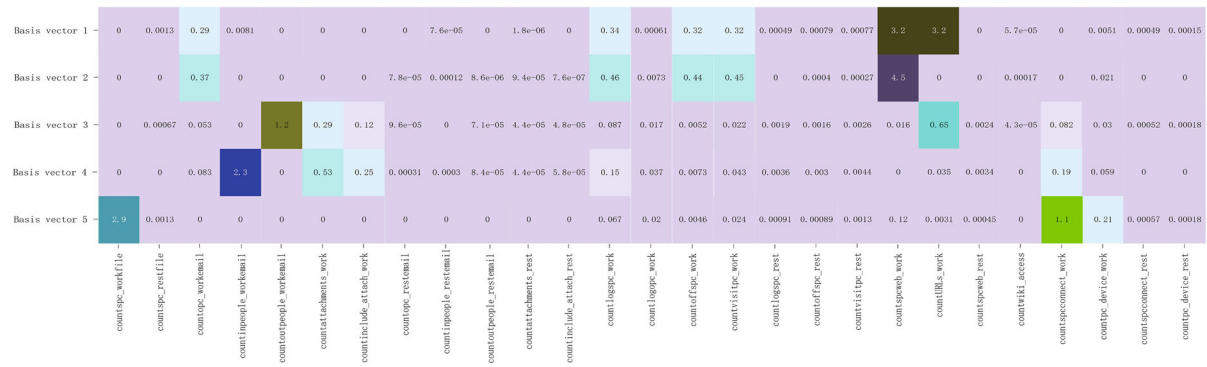


Fig. 4 Distribution of the original features in $r = 5$

Table 3 Cosine similarity statistics of the basis vectors

Cosine similarity	Maximum	Minimum	Mean
r=3	0.963	0.011	0.329
r=4	0.710	0.003	0.229
r=5	0.715	0.006	0.125

Table 4 CH index for different numbers of clusters

K	4	5	6	7	8	9
CH	27981.88	32958.23	22471.65	21011.25	49645.46	38047.11

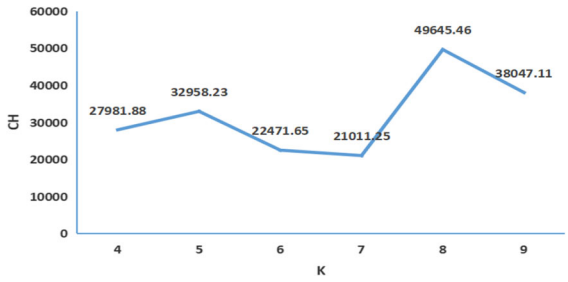


Fig. 5 CH index for different numbers of clusters

Table 5 Data distribution corresponding to basis vector

Class number	Sample size	Data distribution
0	4853	Basic vectors are all dense
1	30162	Basis vectors are all sparse
2	18283	Basis vector 4 is sparse
3	1072	Basis vectors 2, 3 and 4 are sparse
4	1950	Basic vectors are all dense
5	784	Basis vector 4 is sparse
6	30219	Basis vector 4 is sparse
7	2858	Basis vector 1 is sparse

group user behaviors. The simulation results are shown in Table 6.

Table 3 shows that the distribution of cluster 0 and 4 is relatively dense. As the result, the remaining 115 users included in clusters 0 and 4 are the suspicious users, which must be identified.

4.2 Suspicious users re-screening and comparative experiment

Firstly, sub-strategy 1 is used to target part of malicious users. Eight malicious users including DCH0843, EHB0824, EHD0584, GHL0460, KLH0596, MAR0955, MCF0600 and RAB0589 are identified based on Algorithm 2. These users all share a common characteristic of uploading confidential data to the website wikileaks.org in connection with document disclosure violations.

Secondly, since there are 20 working days in a month, set $N_{days} = 20$. In the first 20 days, there are five features that are all 0 including $countspc\_restfile$, $countou\ tpeople\_restemail$, $countwiki\_access$, $countspcconnect\_rest$ and $countspc\_device\_rest$. Thus, we delete these five features.

The Z-score of all user behavior data is calculated based on Eqs. (12)-(13), and the normal behavior threshold of the first 20 days is obtained by using Eqs. (15)-(16):

$$threshold1 = [0, 0, 0.513, 0, 0, 0, 0, 0, 0, 0, 0, \\ 0, 0, 0, 0, 0, 0, 0, 0.624, 0, 0, 0]$$

Then, we calculate the number of behavioral violations (17) of a user within the range of 20 days. Considering that there are 22-dimensional features, each domain has 4–5-dimensional features on average. In real life, violations of insider threat are generally sequential and include at least files and other actions, such as opening the computer, sending e-mails, uploading websites, and shutting down the computer. In other words, threatening behavior occurs across domains. Based on this thought, we set $threshold2 = 6$. As it is elaborated in Algorithm 3, the behavior with $AS1 > threshold2$ is suspicious.

Normal behavior baseline is trained based on the method proposed by sub-strategy 1 and sub-strategy 2. Meanwhile, one-class learning method can also train normal behavior baseline based on the data of the previous N_{days} days to detect suspicious users. In order to evaluate the effectiveness, the proposed sub-strategy 1 and sub-strategy 2 are compared with several one-class learning methods including OCSVM, IF and LOF. The average value of the index is obtained by using the method of 10% cross-verification. In light of known insider threat scenery information, the advantages of the proposed strategy are verified with the following indicators.

(1) P:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{28}$$

where P denotes the proportion of correctly classified samples; TP denotes the number of normal samples with correct classification (true positives); and FP denotes the number of normal samples with the wrong classification (false positives).

(2) R:

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{29}$$

where R is the number of true positives divided by the number of true positives plus the number of false neg-

Table 6 Comparison of average results for different methods

Indicator	OCSVM	IF	LOF	Sub Strategy 1,2
P	0.010	0.040	0	**0.566**
R	1	1	–	1
F1	0.020	0.079	–	**0.723**
FPR	0.860	0.856	0.861	**0.729**
FP+TN	114	111	115	**59**

The bold indicates optimal results

atives; FN denotes the number of malicious samples with the wrong classification (false negatives).
(3) $F1$:

$$F1 = \frac{2\text{PR}}{P + R} \tag{30}$$

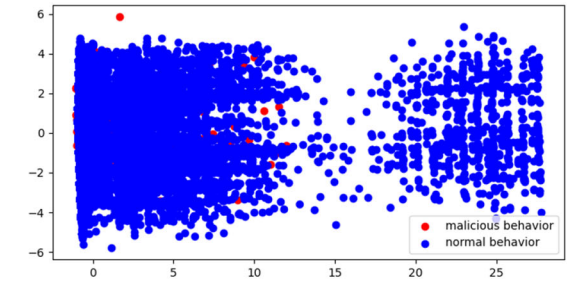
(4) FPR:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{31}$$

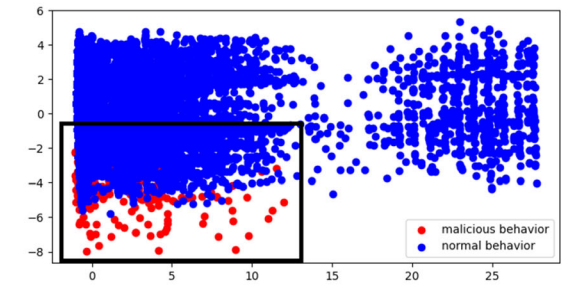
where FPR denotes the proportion of true normal samples in the predicted number of malicious samples. Here, the predicted number of malicious samples FP + TN are suspicious users who need to observe their behavior changes further.

Comparative results are illustrated in Table 6. It can be observed that sub-strategy 1 and sub-strategy 2 perform best with the result of 0.566 for P, 1 for R, 0.723 for $F1$, 0.729 for FPR and 59 for FP + TN, respectively. One-class learning methods may have better recognition accuracy for malicious behavior, but fail to locate malicious users. Sub-strategy 1 and sub-strategy 2 can obtain 59 suspicious users. This includes 8 malicious users identified by sub-strategy 1, and 51 users that need to be further identified .

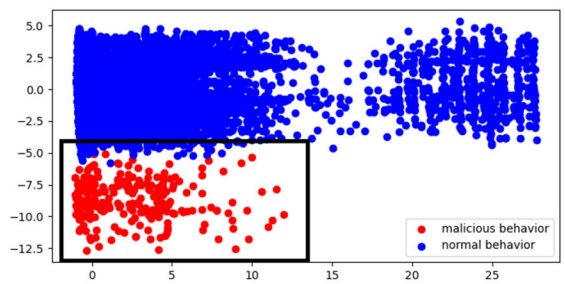
As shown in Fig. 6, the red dot label is abnormal threat behavior data, and the blue dot label is normal behavior data. It can be seen that the data of threat behavior is mixed in the normal behavior data and cannot be directly eliminated, so it is of great significance to analyze and study. Subgraph (b) is a visual representation of the results of sub-strategy 1 and sub-strategy 2 in our integration strategy, and it can be seen that after screening, the initial separation of malicious threat behavior data and normal behavior data is realized. Subgraph (c) is a visual display of the malicious person detection, tracking and positioning of sub-strategy 3, that is, the result of the entire strategy is realized.



(a) Original behavior visualization



(b) Partial filtering visualizations



(c) Fully filter visualizations

Fig. 6 Data visualization

4.3 Malicious users' location and analysis

Using sub-strategy 3 could analyze temporal individual behavior change and locate malicious users. According to Eqs. (18)-(20), the threshold value of individual behavior change range is determined:

$$threshold3 = [0.5, 1.5, 1, 0.5, 0.5, 0.5, 0.5, 0, 0.5, 0, 1]$$

Furthermore, according to Eq. (21), we calculate the violation times matrix $AS2$ of behavior changes range for all users. The rows and columns in the matrix that are not 0 correspond to the suspicious characteristics of the suspicious users. The suspicious users are the feature $countoutpeople\_workemail$ of

CJP0952, IKP0472, ILH0958 and MLM0950, and the feature *coun tspcweb_work* of EDB0714, EGD0132, HXL0968, MDH 0580, PNL0301, PSF0133, RAR0725 and TNM0961, respectively. Among them, several users are also suspicious in feature *countURLs_work*. Figure 7 shows temporal individual behavior change of four users under suspicious feature *countoutpeople_workemail*. Figure 8 shows temporal individual behavior change of eight users under suspicious feature *countspcweb_work*. As shown in Figs. 7 and 8, *countspcweb_work* is more significant in detecting abnormal change.

Furthermore, the Pettitt test method is used to detect and judge the mutation time points of each suspicious user, and the specific results are shown in Table 7. Users with the precision $P < 0.05$ reject the null hypothesis and are identified as malicious, but two of them (CJP0952 and MLM0950) are normal. It is clear that the P values of these two users differ by more than 10 times from the actual malicious users. If the significance level is set more strictly, such as 0.01, users CJP0952 and MLM0950 are judged as normal. The first malicious time of EDB0714, EGD0132, HXL0968, MDH0580, PNL0301, PSF0133, RAR0725 and TNM0961 is consistent with that of the mutation point in Fig. 8. If users upload confidential data to the website wikileaks.org and browse too many web pages, the production line workers have malicious purposes. Similarly, starting from these two features, when the mutation occurs for the first time, the authority management of the production line worker needs to be strengthened by the enterprise in time to prevent greater losses.

When setting the normal behavior baseline, 16 feature quantities with 0 value are deleted from sub-strategy 2 and sub-strategy 3. Excluding typical characteristics in sub-strategy 1, 15 characteristics are established for identifying normal users with temporal individual behavior change. For the actual malicious users detected in Fig. 8, violations are concentrated within a month, which is 10 times higher than the average situation. Similarly, Fig. 9 presents the temporal individual behavior change of two users with significant detection features. It can be seen that the changing trend of users' behavior is obviously different from that of malicious users. In the later stage, the change is still within the normal behavior, but the behavior frequency is reduced. It can be seen that the deleted features do not generally distinguish malicious behavior. The experimental results show that the proposed ensemble strategy could

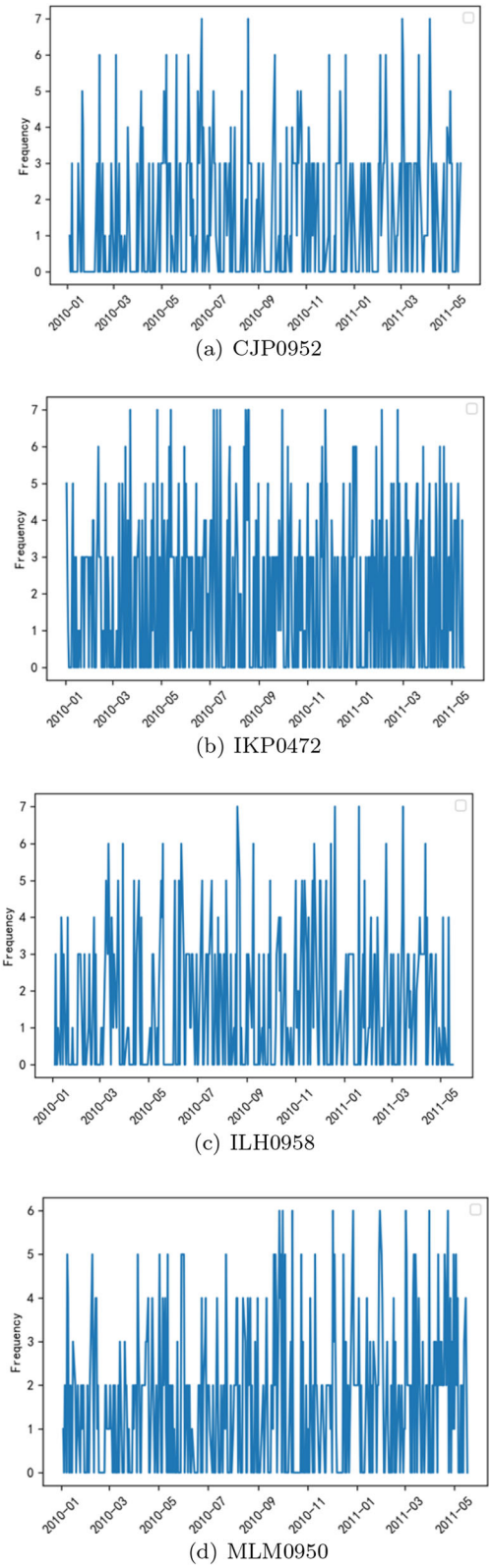


Fig. 7 Changes in Suspicious Feature countoutpeople_ workemail

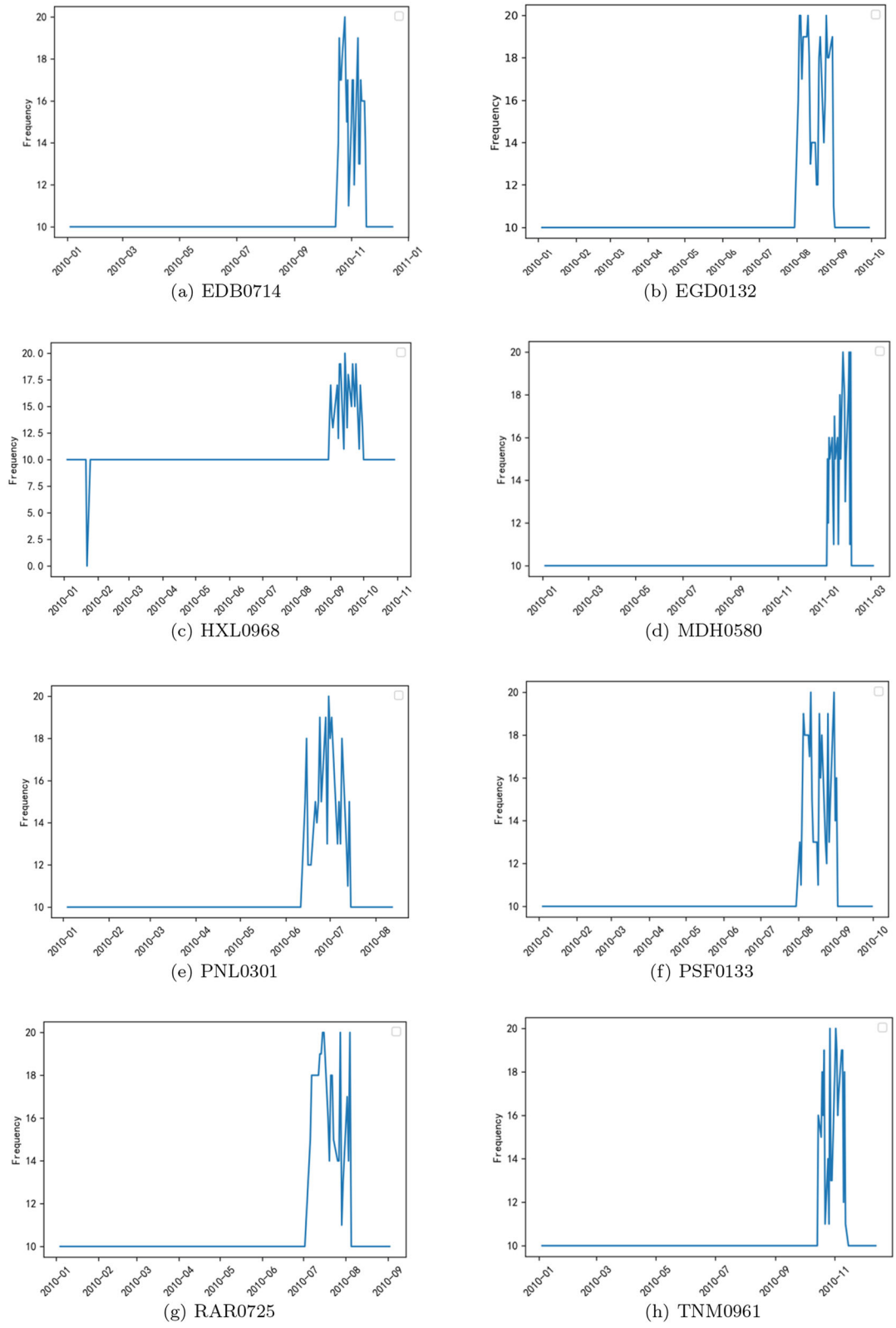
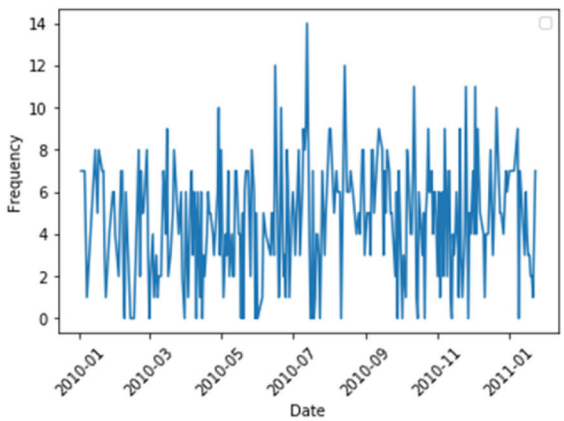


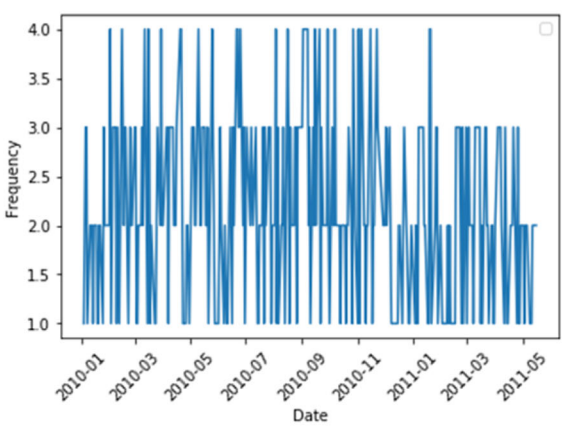
Fig. 8 Changes in Suspicious Feature countspcweb_work

Table 7 Pettitt test results

User id	P	Test result	Mutation time
CJP0952	0.0261	Significant	2010-04-30
IKP0472	1.2517	Nonsignificant	2010-05-08
ILH0958	0.8771	Nonsignificant	2010-06-02
MLM0950	0.0144	Significant	2010-12-15
EDB0714	0.0005	Significant	2010-10-18
EGD0132	0.0002	Significant	2010-08-02
HXL0968	0.0003	Significant	2010-08-31
MDH0580	0.0009	Significant	2011-01-04
PNL0301	0.0001	Significant	2010-06-14
PSF0133	9.57e–05	Significant	2010-08-02
RAR0725	0.0002	Significant	2010-07-06
TNM0961	0.0010	Significant	2010-10-15



(a) $countspc\_workfile$



(b) $countspcconnect\_work$

Fig. 9 Temporal individual behavior change of AOK0844 and IBB0696

effectively detect and analyze the malicious behavior of production line workers.

5 Conclusion

In this paper, the concept of insider threat behavior kill chain is proposed to analyze and characterize the psychological and behavioral change process of malicious users in the organization. Correspondingly, we give a user-level behavior detection framework including NMF-GMM based on similar user behavior grouping and ensemble strategy consisting of three sub-strategies. NMF is used to overcome the linear inseparability of high-dimensional data and optimize the user grouping results of the GMM clustering model. In addition, the three sub-strategies reduce the scope of malicious users layer by layer to achieve the final temporal individual behavior model. Among them, sub-strategy 1 uses typical characteristics to target part of malicious users. Sub-strategy 2 adopts the Z-score to score the user's overall behavior and performance, and then screens suspicious users according to the behavior baseline set within normal time. Sub-strategy 3 determines the suspicious characteristics of the suspicious users from the behavior change range, and further establishes the temporal individual behavior change model of the suspicious users. The Pettitt test method is used to detect malicious users, and the time of the first occurrence of abnormal behavior is given. The experimental results show that the proposed ensemble strat-

egy can effectively implement the location of malicious users. .

Acknowledgements This work was supported in part by the Scientific and Technological Innovation 2030—Major Project of New Generation Artificial Intelligence (2020AAA0109300), the Bashkir State Medical University Strategic Academic Leadership Program (PRIORITY-2030).

Funding The authors have not disclosed any funding.

Data availability The data comes from Carnegie Mellon University's Insider Threat Data Center (https://www.sei.cmu.edu). The experimental data can be provided by the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest concerning the publication of this manuscript.

References

1. Glasser, J., Lindauer, B.: Bridging the gap: a pragmatic approach to generating insider threat data. Proceedings of the 2nd IEEE CS security and privacy workshops, pp. 98-104 (2013)
2. Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., Ochoa, M.: Insight into insiders and IT: a survey of insider threat taxonomies, analysis, modeling, and countermeasures. ACM Comput. Surv. **52**(2), 30 (2019)
3. Oladimeji, T.O., Ayo, C.K., Adewumi, S.E.: Insider threat detection using binary classification algorithms. IOP Conf. Series **1107**, 012031 (2021)
4. Yu, J., Kim, M., Oh, H., Yang, J.: Real-time abnormal insider event detection on enterprise resource planning systems via predictive auto-regression model. IEEE Access **9**, 62276–62284 (2021)
5. Yuan, S., Wu, X.: Deep learning for insider threat detection: review, challenges and opportunities. Comput. Secur. **104**, 102221 (2021)
6. Al-Mhiqani, M.N., Ahmad, R., Abidin, Z.Z., Yassin, W., Hassan, A., Abdulkareem, K.H., Ali, N.S., Yunos, Z.: A review of insider threat detection: classification, machine learning techniques, datasets, open challenges, and recommendations. Appl. Sci-Basel **10**(15), 5208 (2020)
7. Zou, B., Yang, M., Guo, J., Wang, J.B., Benjiamin, E.R., Liu, H., Li, W.: Insider threats of physical protection systems in nuclear power plants: prevention and evaluation. Prog. Nucl. Energ. **104**, 8–15 (2018)
8. Meng, W.Z., Choo, K.K.R., Furnell, S., Vasilakos, A.V., Probst, C.W.: Towards Bayesian-based trust management for insider attacks in healthcare software-defined networks. IEEE Trans. Netw. Serv. Man. **15**(2), 761–773 (2018)
9. Holger, S.: 2020 insider threat report https://www.securonix.com/resources/2020-insider-threat-report/
10. Kim, D.W., Hong, S.S., Han, M.M.: A study on classification of insider threat using Markov chain model. KSII Trans. Internet Inf. Syst. **12**(4), 1887–1898 (2018)
11. Jang, M., Ryu, Y., Kim, J.S., Cho, M.: Against insider threats with hybrid anomaly detection with local-feature autoencoder and global statistics (LAGS). IEICE Trans. Inf. Syst. **E103D**(4), 888–891 (2020)
12. Bauder, R.A., Khoshgoftaar, T.M.: A study on rare fraud predictions with big Medicare claims fraud data. Intell. Data Anal. **24**(1), 141–161 (2020)
13. Wang, Z.C., Sun, Y.R.: Optimization of SMOTE for imbalanced data based on AdaRBFNN and hybrid metaheuristics. Intell. Data Anal. **25**(3), 541–554 (2021)
14. Dlamini, G., Fahim, M.: DGM: a data generative model to improve minority class presence in anomaly detection domain. Neural Comput. Appl. **33**(20), 13635–13646 (2021)
15. Kim, J., Park, M., Kim, H., Cho, S., Kang, P.: Insider threat detection based on user behavior modeling and anomaly detection algorithms. Appl. Sci-Basel **9**(19), 4018 (2019)
16. Le, D.C., Zincir-Heywood, N.: Exploring anomalous behaviour detection and classification for insider threat identification. Int. J. Netw. Manag. **31**(4), e2109 (2019)
17. Gamachchi, A., Boztas, S.: Insider threat detection through attributed graph clustering, In: Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 112-119 (2017)
18. Zou, S., Sun, H., Xu, G., Quan, R.: Ensemble strategy for insider threat detection from user activity logs. CMC-Comput. Mater. Con. **65**(2), 1321–1334 (2020)
19. Raman, M.R.G., Somu, N., Mathur, A.P.: A multilayer perceptron model for anomaly detection in water treatment plants. Int. J. Crit. Infr. Prot. **31**, 100393 (2003)
20. Rashid, T., Agrafiotis, I., Nurse, J.R.C.: A new take on detecting insider threats: exploring the use of hidden Markov models. CCS International workshop on managing insider security threats, pp. 47-56 (2016)
21. Lo, O., Buchanan, W.J., Griffiths, P., Macfarlane, R.: Distance measurement methods for improved insider threat detection. Secur. Commun. Netw. UNSP5906368 (2018)
22. Chen, T., Tang, L.A., Sun, Y.Z., Chen, Z.Z., Zhang, K: Entity embedding-based anomaly detection for heterogeneous categorical events. In: International joint conference on artificial intelligence, pp. 1396-1403 (2016)
23. Wang, J.R., Cai, L.J., Yu, A.M., Meng, D.: Embedding learning with heterogeneous event sequence for insider threat detection. In: 31st IEEE international conference on tools with artificial intelligence, pp. 947-954 (2019)
24. Hutchins, E., Cloppert, M., Amin, R.: Intelligence-Driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: 6th International conference on information warfare and security, pp. 80-81 (2011)
25. Kim, H., Kwon, H.J., Kim, K.K.: Modified cyber kill chain for multimedia service environments. Multimed. Tools Appl. **78**(3), 3153–3170 (2019)
26. Ning, C., Xi, Z.: Window-type detector for stealthy false data injection attack in cyber-physical systems. Int. J. Syst. Sci. (2023). https://doi.org/10.1080/00207721.2023.2186754
27. Gayathri, R.G., Sajjanhar, A., Xiang, Y.: Image-based feature representation for insider threat classification. Appl Sci-Basel **10**(14), 4945 (2020)
28. Oh, J., Kim, T.H., Lee, K.H.: Advanced insider threat detection model to apply periodic work atmosphere. KSII Internet Inf. **13**, 1722–1737 (2019)

29. Garchery, M., Granitzer, M.: Identifying and clustering users for unsupervised intrusion detection in corporate audit sessions. In: Identifying and clustering users for unsupervised intrusion detection in corporate audit sessions, pp. 19-27 (2019)
30. Aldairi, M., Karimi, L., Joshi, J.: A trust aware unsupervised learning approach for insider threat detection. IN: IEEE International conference on information reuse and integration for data science, pp. 89-98 (2019)
31. Lisboa, P.J.G., Saralajew, S., Vellido, A., Fernández-Domenech, R., Villmann, T.: The coming of age of interpretable and explainable machine learning models. Neurocomputing **535**(28), 25–39 (2023)
32. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–791 (1999)
33. Chen, Y., Ashizawa, N., Yeo, C.K., Yanai, N., Yean, S.: Multi-scale self-organizing map assisted deep autoencoding Gaussian mixture model for unsupervised intrusion detection. Knowl.-Based Syst. **224**, 107086 (2021)
34. Blaise, A., Bouet, M., Conan, V., Secci, S.: Detection of zero-day attacks: An unsupervised port-based approach. Comput. Netw. **180**, 107391 (2020)
35. Taïbi, S., Zeroual, A., Meddi, M.: Efect of autocorrelation on temporal trends in air temperature in Northern Algeria and links with teleconnections patterns. Theor. Appl. Climatol. **147**(3), 959–984 (2022)
36. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat-Thero. M. **3**, 1–27 (1974)