Song *et al. Cybersecurity* (2024) 7:2
https://doi.org/10.1186/s42400-023-00190-9

Cybersecurity

RESEARCH

Open Access

BRITD: behavior rhythm insider threat detection with time awareness and user adaptation

Shuang Song[1,2,3], Neng Gao[1,2*], Yifei Zhang[1,2] and Cunqing Ma[1,2]

Abstract

Researchers usually detect insider threats by analyzing user behavior. The time information of user behavior is an important concern in internal threat detection. Existing works on insider threat detection fail to make full use of the time information, which leads to their poor detection performance. In this paper, we propose a novel behavioral feature extraction scheme: we implicitly encode absolute time information in the behavioral feature sequences and use a feature sequence construction method taking covariance into account to make our scheme adaptive to users. We select Stacked Bidirectional LSTM and Feedforward Neural Network to build a deep learning-based insider threat detection model: Behavior Rhythm Insider Threat Detection (BRITD). BRITD is universally applicable to various insider threat scenarios, and it has good insider threat detection performance: it achieves an AUC of 0.9730 and a precision of 0.8072 with the CMU CERT dataset, which exceeds all baselines.

Keywords Insider threat detection, Behavior pattern mining, Time information, User adaptive, Deep learning

*Correspondence:
Neng Gao
gaoneng@iie.ac.cn
Full list of author information is available at the end of the article

Graphical Abstract

Introduction

Insider threat refers to the insider's behaviors that violate the security policy of the organization (Homoliak et al. 2019), and it is currently widely affecting various enterprises and organizations. User behaviors can be described as sequential decision-making processes (Pan et al. 2020). Users usually have specific decision preferences which can also be called behavioral patterns. In insider threat detection, the behavioral patterns of insider threats are typically considered to be different from benign behavioral patterns. Researchers mine user behavior patterns to distinguish threat behaviors from benign ones. The time information of user behaviors is an important concern of behavior pattern mining for insider threat detection. Benign behaviors at an unusual time can also signal an insider threat. For example, users may log into their accounts late at night to steal data.

Thanks to the deep structure, deep learning is considered more suitable for analyzing complex user behaviors. Deep learning techniques currently utilized for detecting insider threats include deep autoencoder (Liu et al. 2018; Tuor et al. 2017; Chattopadhyay et al. 2018), DBN (Lin et al. 2017), CNN (Hu et al. 2019; Bu and Cho 2020), RNN (Zhang et al. 2018; Yuan et al. 2019; Tuor et al. 2017; Nasir et al. 2021; Dr et al. 2022; Villarreal-Vasquez et al. 2023), Transformer (Yuan et al. 2020), isomorphic graph representation learning (Jiang et al. 2019), heterogeneous graph representation learning (Liu et al. 2019), etc. To the best of our knowledge, however, existing deep learning-based insider threat detection studies pay little attention to the behavior time information (Yuan and Wu 2020) that has a strong connection to insider threat behaviors, which makes the studies less effective at detecting insider threats.

The time information of user behavior can be classified into two types: relative time information and absolute time information. The relative time information of behaviors refers to the sequence relationship of user behaviors in the sense of time (for example, "<Logon> occurred before <File open>" is a kind of relative time information). The absolute time information of behaviors refers to the position of the behavior time in the 24 consecutive hours of a day (for example, "<Logon> occurred at 8:40" is a kind of absolute time information).

A few studies have made crude and mostly unsuccessful attempts to include user behavior time elements in insider threat detection. These studies have focused only on one type of time information, either absolute time information or relative time information. Ye et al. (2020), Liu et al. (2018) and Tuor et al. (2017) mainly focus on the absolute time information of user behavior. Ye et al. (2020), as shown in Fig. 1a, gather users' hourly activities and build separate models for each hour (for example, a detection model is trained using only the activities from 8:00 to 9:00, and this model can only be used to detect activity data from 8:00 to 9:00). They create 24 models

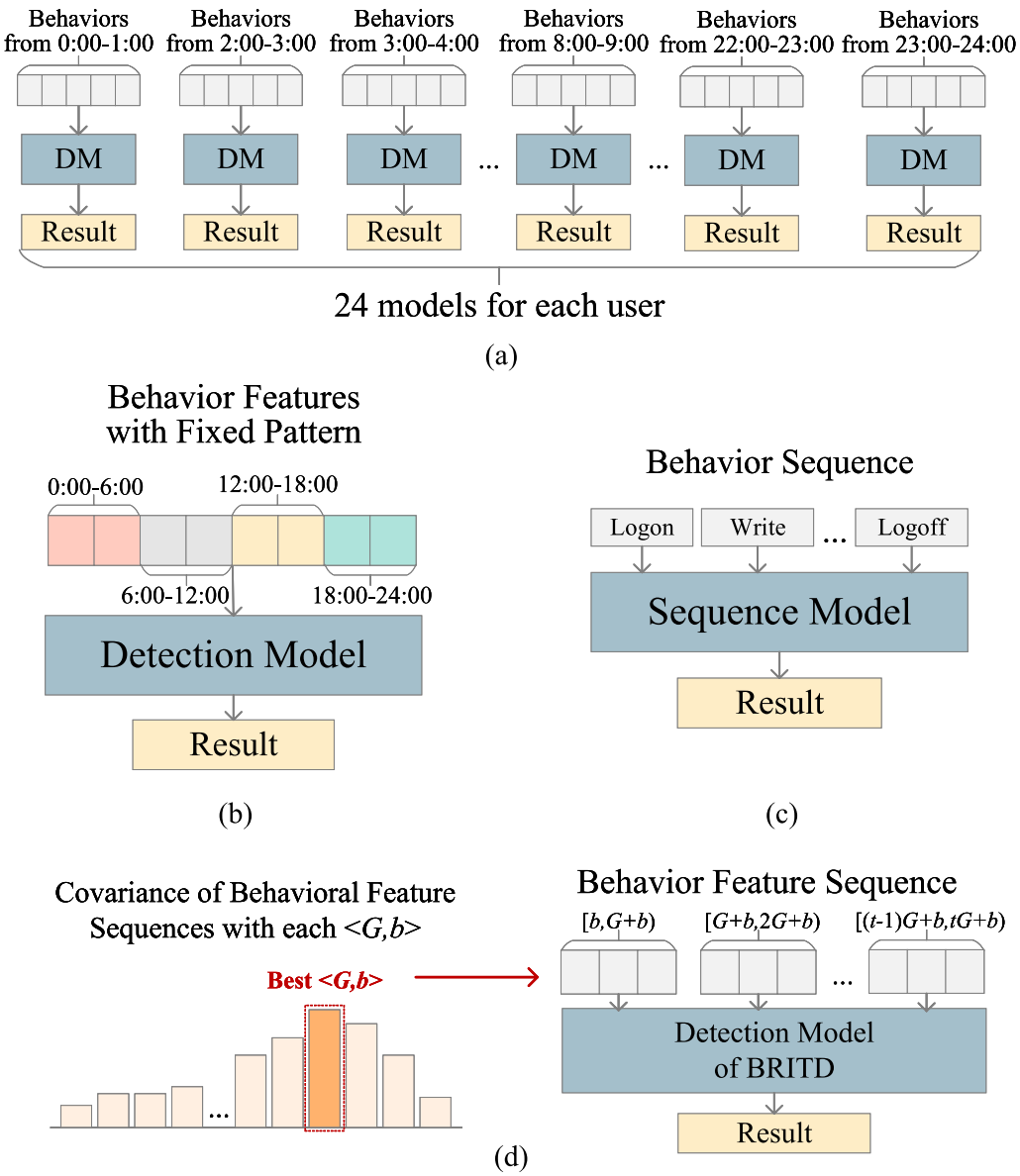


Fig. 1 Insider threat detection schemes that attempt to include user behavior time elements. **a** Illustrates the schemes that focus only on absolute time information (taking Ye et al. work (Ye et al. 2020) as an example); **b** illustrates another type of schemes that focus only on absolute time information (taking Tuor et al.'s work (Tuor et al. 2017) as an example); **c** illustrates the general pattern of schemes that focus only on relative time information; **d** illustrates our scheme, where G refers to the time granularity, b refers to the starting bias, t is the index of the time period, and the $< G, b >$ pair represents the construction scheme of the behavioral feature sequence

for each user, in other words. The scheme is unable to capture the relationship between user behaviors at various time periods since there is no information contact or communication between multiple models of the same user. Similarly, Liu et al. (2018) and Tuor et al. (2017) gather users' hourly or 6-hourly activities and combine the user activities over the course of a day into feature vectors (for example, as shown in Fig. 1b, when they gathered user activities every 6 h, the feature vectors would contain data for the user's activities from 0:00–6:00, 6:00–12:00, 12:00–18:00, to 18:00–24:00, respectively). Finally, an Feedforward Neural Network (FNN) is used for threat detection, which means that the model learns user behaviors from various time periods without taking the temporal order into account. The above approaches all fail to capture the relative time information of user behavior when learning user behavior patterns. Furthermore, the rules they utilize for gathering user behaviors

(e.g., gathering hourly or 6-hourly user behaviors on a fixed basis) are rigid and unable to adjust easily to diverse users.

Sequence models such as LSTM, Transformer, etc., or heterogeneous graphs are used to model user behavior by Zhang et al. (2018), Liu et al. (2019), Yuan et al. (2020), Nasir et al. (2021), etc, as shown in Fig. 1c. However, these methods mainly focus on relative time information of user activity and do not establish a direct connection with the specific timing of user activities. Due to this limitation, sequences occurring at different times or with significantly different activity intervals may be treated as one and the same.

The relative time information and absolute time information of user behaviors are not simultaneously captured by any of the aforementioned methods. Meanwhile, the periodic user behavior rhythm is not utilized or fully analyzed by these methods. Additionally, it is important but often neglected to carry out adaptive extraction and analysis of the behavior time distribution of particular users given that users have a variety of behavioral patterns.

To address the above issues, our goal is to construct an insider threat detection scheme that captures both absolute and relative time information of user behavior, mines user behavior rhythm, and is user adaptive.

In this paper, we propose an insider threat detection scheme called BRITD as can be seen in Fig. 1d. The scheme consists of two parts:

- Feature Extraction Method. We propose a feature extraction method that can encode the time information implicitly and reinforce the behavior rhythm adaptively. We create a sequence by calculating user behavior statistical vectors over predetermined time frames and concatenating the statistical vectors of a day in chronological order. The position of a vector in the sequence represents its corresponding absolute time periods (for example, when the behavior statistical vectors within the time periods of 0:00–6:00, 6:00–12:00, 12:00–18:00, to 18:00–24:00 are concatenated in chronological order to form a sequence, we can assert that the behavior contained in the first vector of the sequence must occur between 0:00–6:00, and the same logic applies to the other vectors). We choose the sequence construction scheme that maximizes the covariance of the behavioral feature sequences for each user to obtain better detection performance, which means our insider threat detection scheme is user adaptive.
- Insider Threat Detection Model. We propose a deep-learning based insider threat detection model to take full advantage of our feature extraction scheme. Our insider threat detection model is an autoencoder. We select stacked BiLSTM and FNN to construct the encoder of the model, and we use another FNN as the decoder of the model. Stacked BiLSTM is used to model the absolute time information and relative time information, and the FNN in the encoder improves the model's ability to capture other behavior features than time-related behavior features.

Our insider threat detection scheme fits the user-day behavioral rhythm. an intuitive observation is: Natural time takes one day as a cycle; affected by it, user behaviors also take one day as a cycle (in this paper, the user behaviors on weekends are not considered). The behavior distribution of specific users in each cycle is relatively stable (a typical example is that the daily working hours and off-duty hours of specific users do not vary significantly under normal circumstances). We call this periodic behavior distribution the user-day behavior rhythm. Both our feature extraction method and detection model follow the cycle and have the ability to fit the user-day behavior rhythm adaptively. Therefore, our scheme can capture behavioral time information, learn benign behavior patterns, and detect insider threat behaviors more effectively.

To summarize, we propose a time-aware and user-adaptive insider threat detection scheme. Our contributions in this paper are as follows:

- Our scheme includes the methods of absolute time information implicit encoding and behavioral rhythm fitting. The scheme can fit the natural cycle of user behavior, and mine the user-day behavior rhythm in depth, which makes up for the problem that existing insider threat detection schemes do not analyze the time information of user behavior sufficiently.
- Our scheme selects the appropriate feature sequence construction method for each user by calculating and comparing the covariance of the feature sequences constructed by different methods, which can extract and further reinforce the user-day behavior rhythms specific to each user. The positive correlation between the covariance of feature sequences and the detection performance is experimentally proved.
- Our insider threat detection model can learn and analyze both time-related and time-independent behavior features, which improves the scenario generalization of our detection scheme.
- We use the CMU CERT V6.2 dataset to evaluate the effectiveness of the insider threat detection model with IF, OCSVM, FNN, BiLSTM, and HMM as baselines. The performance of the model overperforms all the baselines: the AUC of the model reaches 0.9730, which is at least 0.0802 higher than baselines; and

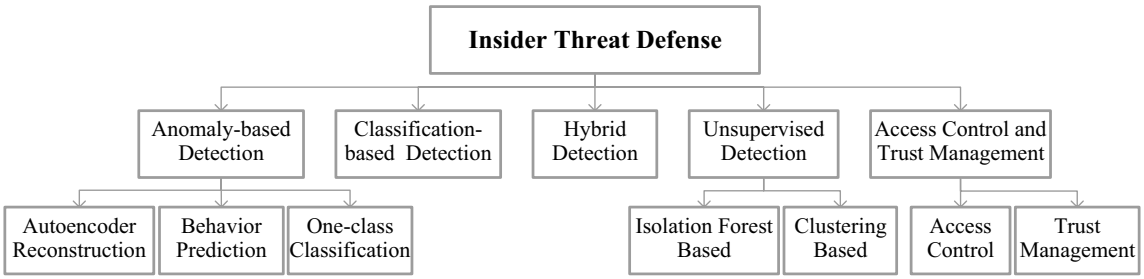


Fig. 2 Technical classification of insider threat detection, access control and trust management

the precision reaches 0.8072, which is at least 0.2746 higher than baselines.

Based on the above scheme, we further improve the accuracy and scenario generalization of insider threat detection. At the same time, we also provide a new idea of user behavior time analysis for other similar studies.

Related works

The defense techniques against information system threats have developed to date, and researchers have produced a large number of outstanding research results in the fields of vulnerability mining (Zhang et al. 2021; Ispoglou et al. 2020), malicious code defense (Lu et al. 2021a, b), APT attack detection (Wang et al. 2022; Alsaheel et al. 2021), and network traffic analysis (Khandait et al. 2021). Insider threat is a special and important information system threat, and there are many targeted detection methods against insider threat.

As shown in Fig. 2, insider threat detection techniques can be classified into 5 categories: anomaly-based, misuse-based, hybrid, classification-based, and unsupervised detection (Homoliak et al. 2019). Besides, access control and trust management are also important for defending against insider threats. In the following, we will discuss the related works in each of the aforementioned 6 aspects.

Anomaly-based detection

Anomaly-based detection is a scheme that models only benign behaviors and detects threats by capturing the differences between test data and benign behavior patterns. Anomaly detection techniques that only use benign behavior data as training data are frequently used because in real-world scenarios: (1). the number of threat behaviors is very small compared to benign behaviors; (2). threat data labels are difficult to obtain; and (3). threat behaviors are complex and varied.

Anomaly-based detection mainly includes three types of methods: one-class classification, behavior prediction, and autoencoder reconstruction. For one-class-based methods, Song et al. (2013) use the one-class conditional random field (OCCRF) to learn the normal behavior pattern of users and use hinge loss to increase the distinction between normal and abnormal behavior. Rashid et al. (2016) model the user's normal behavior feature sequence using the hidden Markov model (HMM). Lin et al. (2017) use the one-class support vector machine (OCSVM) to identify threats.

With the development and widespread application of deep learning, researchers began to use deep learning models to build anomaly-based detection schemes. Some of them use deep learning models to predict the user's next behavior and compare it with the actual behavior. Then, they regard behaviors with significant differences from the predicted results as abnormal behaviors. For example, Villarreal-Vasquez et al. (2023) use LSTM to model system event sequences and predict the probability of the next event, with low-probability events being considered anomalous events. A similar approach is also used by Yuan et al. (2019). Others use deep autoencoder to reconstruct user behavior data, and behaviors with high reconstruction error are considered anomalous. Schemes based on such tack include Liu Liu et al. work (Liu et al. 2018), Rida Nasir et al. work (Nasir et al. 2021), Jongmin Yu et al. work (Dr et al. 2022), etc. In particular, Nasir et al. (2021) use the LSTM-based autoencoder, which considers the historical behavior information when reconstructing user behavior data. Dr et al. (2022) propose an insider threat detection scheme based on Adversarial Recurrent Autoencoder (ARAE) that improves detection performance by making the encoding results close to a normal distribution.

It should be noted that Tuor et al. (2017) prove through experiments that the detection scheme based on autoencoder reconstruction has better detection performance than the prediction-based detection scheme. We build our insider threat detection model based on the deep autoencoder's fundamental structure in light of their research.

Misuse-based detection and classification-based detection

Misuse-based detection only models threat behaviors and detects threats by measuring the similarity between test data and threat behavior patterns. In reality, to our knowledge, there are currently few genuine misuse-based detection schemes in the field of insider threat detection due to the dearth of labeled threat behavior data and the diversity of insider threat behavior.

On the other hand, a lot of detection schemes opt to employ classification-based methods that model both benign and malicious behavior. Classification-based detection techniques introduce threat behavior information and can establish a direct connection between the detection model and insider threat activities, giving it an advantage of low false alarm rates compared to anomaly detection. However, the classification-based detection techniques still struggle with high training data imbalance since there is a dearth of labeled threat behavior data. Chattopadhyay et al. (2018) use random forest (RF) and multilayer perceptron (MLP) for classification, while using random under-sampling to address data imbalance issues. Azaria et al. (2014) construct seven algorithms combining support vector machine (SVM), Multinomial Naive Bayes (MultinomialNB), semi-supervised learning, and partially supervised classification, which can handle training data with the high imbalance and lacking labels. Duc C. Le et al. apply three semi-supervised learning techniques, namely label propagation (LP), label spreading (LS), and self-training (ST), to insider threat detection to address the issue of data imbalance (Le et al. 2021). Al-Shehari and Alsowail (2021) implement seven commonly used machine learning algorithms to detect data leakage, with the synthetic minority oversampling technique (SMOTE) to address the data imbalance issue. Wu and Li (2021) use an ensemble of NN and RF to detect threat activities after feature selection.

Well-constructed classification detection schemes can achieve good detection performance on the test set. However, these schemes are still limited by data hysteresis and lack sensitivity to unknown threat behaviors that are not included in the training set.

Hybrid detection

Hybrid detection is a combination of anomaly-based and misuse-based approaches. Similar to misuse-based detection methods, this type of detection method is also very rare in the field of insider threat detection. Fortunately, noticing this research gap, Mohammed Nasser Al-Mhiqani et al. propose a machine learning-based hybrid insider threat detection scheme (Al-Mhiqani et al. 2022). The scheme first uses the misuse-based model to detect known threats and subsequently uses the anomaly-based model to detect unknown threats that cannot be captured by the misuse model. The scheme solves the problem of insensitivity of misuse detection to unknown threats to some extent but still suffers from a lack of labels, data imbalance, and high false alarm problems.

Unsupervised detection

Unsupervised detection methods do not require data labels, and are mainly implemented through isolation forest (IF) and clustering. Gavai et al. (2015) use IF to capture anomalies in user behaviors relative to their own historical behaviors and relative to their colleagues' behaviors. Similarly, Soh et al. (2019) choose IF to capture potential threats from user sentiment profiles. Kandias et al. (2017) use clustering algorithms like K-means to classify social media data to evaluate user stress levels. Ning Hu et al. propose an abnormal traffic detection method based on multiple kernel clustering (MKC) algorithm (Hu et al. 2021). Fucheng Liu et al. capture malicious nodes in graphs by clustering the embedding vectors of nodes(Liu et al. 2019). Unsupervised detection methods usually have high requirements for the quality of feature extraction.

Access control and trust management

In addition to threat detection, access control and trust management for internal personnel are equally important for defending against insider threats. The occurrence of insider threats is closely related to knowledge, access authority, and trust (Probst et al. 2008). Developing a rigorous management strategy and implementing it effectively can reduce the abuse of the above three factors at the source, which is of great help in defending against insider threats. Takabi and Jafarian (2017) and Rauf et al. (2021) propose moving target defense (MTD) based and biological principle-based access control schemes, respectively. Nathalie Baracaldo et al. use geo-social feature-based user trustworthiness analysis to support authorization decisions (Baracaldo et al. 2019). Weizhi Meng et al. propose a trust management scheme based on Bayesian inference (Meng et al. 2017) to defend against insider attacks from medical smartphone networks (MSNs). They then propose a trust-based threat detection scheme for MSNs (Meng et al. 2020). Wu and Zhang (2022), Ayed et al. (2023) and Asif et al. (2022) on the other hand, propose trust management mechanisms based on blockchain for the supply chain, the Internet of Vehicles, and the Internet of Things, respectively.

Considering the current status and problems of insider threat detection research, in this paper, we propose an anomaly-based insider threat detection scheme. Combining with deep learning, the scheme acquires user behavior rhythm through in-depth analysis of user behavior data with time information, and on this basis,

it captures insider threat behaviors. Our insider threat detection scheme is suitable for multiple insider threat scenarios and can be applied in concert with trust management and access control mechanisms to jointly secure the organization's systems.

Our approach

The insider threat detection approach that we use is shown in Fig. 3. We first perform feature extraction on behavioral logs from different sources. Second, we feed the extracted feature sequences into our deep-learning-based autoencoder for reconstruction. The reconstruction error is the outlier. The bigger the outliers of a data instance, the more likely it is to be an insider threat instance. In the subsequent sections, we detail the feature extraction method and the construction of the deep learning autoencoder that is used in our insider threat detection approach.

User-adaptive feature extraction

The user-adaptive feature extraction method is divided into 2 steps as follows.

- We preprocess the data. We keep both the user id and time in each user behavioral log entry and simplify the rest to a behavior token. Then we aggregate the behavioral logs from different sources belonging to the same user according to the user id and arrange them chronologically to obtain a composite log.
- We construct feature sequences to convert the time information of the behavior token into location information and fit user behavioral rhythms. If in training, we find the best feature sequence construction scheme for each user and use the best scheme to construct feature sequences for them; if in testing, we directly use the best scheme to construct feature sequences for the users.

As for the data pre-processing, we classify or enumerate the attribute values in the original behavioral log, whereafter concatenate them together to obtain a token set. We convert log entries to behavior tokens based on the token set. As an example, Fig. 4. shows the process of converting Logon log entries in the CERT dataset to behavior tokens.

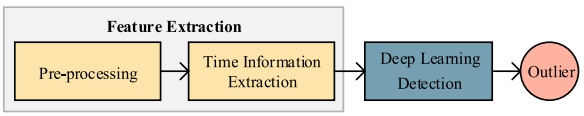


Fig. 3 The framework of our insider threat detection approach

As for the time information conversion, we first divide the consecutive 24 h into T time periods with a granularity of G. The range of each time period is $[0:00+b+Gt, 0:00+b+G(t+1))$, where t is the index of the time period and b is the starting bias. TG = 24 h = 1440 m = 86400 s. Figure 4. shows an example when G = 6h and b = 0h. Second, we classify the behavior tokens in a single day of a specific user into different time periods according to the corresponding time. Then, we count the behavior tokens in each time period by type respectively and obtain a M-dimension statistical vector $\boldsymbol{x}_t^{u,G,b} = (x_{t,0}^{u,G,b}, x_{t,1}^{u,G,b}, \ldots, x_{t,i}^{u,G,b}, \ldots, x_{t,M-1}^{u,G,b})$. i is the index of behavioral feature, $i = 0, 1, 2, \ldots, M-1$. M is the number of behavior token types ($M = 129$ in this paper). We concatenate the vectors corresponding to each time period in chronological order to obtain the user behavior feature sequence: $\mathbf{x}^{u,G,b} = (\boldsymbol{x}_0^{u,G,b}, \boldsymbol{x}_1^{u,G,b}, \ldots, \boldsymbol{x}_t^{u,G,b}, \ldots, \boldsymbol{x}_{T-1}^{u,G,b})$. t is the time index and the position index. The numerical time information in the original behavior log can be converted to position information through the above data processing method.

Converting log entries into tokens and counting each token separately is a common log data processing method in the field of insider threat detection. This method is able to extract non-time features of user behavior and focus on anomalies in behavior type and behavior frequency to help capture them in the subsequent deep-learning detection phase. However, the above traditional feature extraction methods ignore the time information of user behavior. We combine the traditional feature extraction methods with our original feature sequence construction scheme that considers absolute time information to comprehensively extract time features and non-time features of user behavior.

It should be noted that if the time granularity G is consistent with the time granularity in the original behavior log (for example, $G = 1s$), $\boldsymbol{x}_t^{u,G,b}$ is converted from the statistics value of the behavior token to the one-hot. However, considering that when the granularity G is too small, data sparseness will occur, we choose to use a large G, which can also reduce the noise in the original user behavior data. In this paper, $G = 1, 2, 3, 4, 6, 8, 12, 24\,h$ and $b = 0, 1, ..., (G-1)h$.

As for the user adaptation, for each user under each behavioral feature sequence construction scheme, we calculate the covariance of the feature sequences $c^{u,G,b} = covariance(\mathbf{x}^{u,G,b})$. We use a two-tuple to represent a specific behavioral feature sequence construction scheme $s =< G, b >$. We obtain the best feature sequence construction scheme corresponding to a specific user $s_{best} =< G_{best}^u, b_{best}^u >$, where $c^{u,G_{best}^u,b_{best}^u} = max(\{c^{u,G,b} | \forall < G, b >\})$.

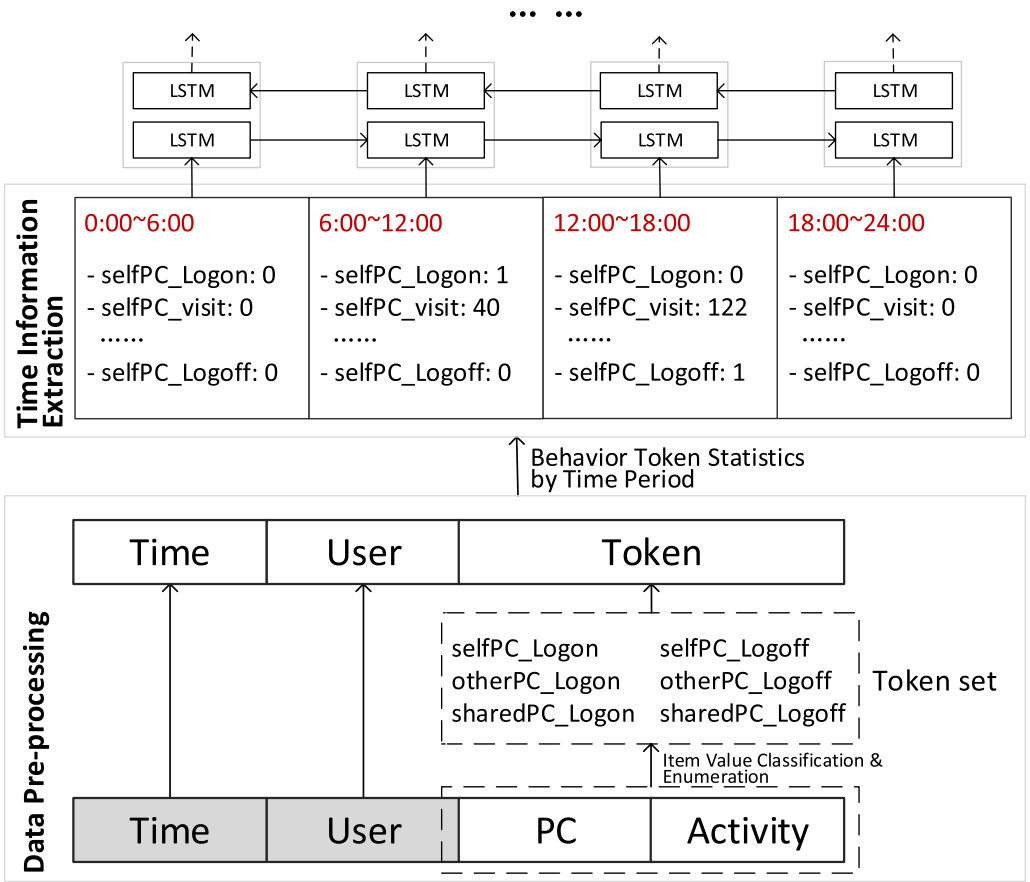


Fig. 4 Our feature extraction includes data preprocessing (take Logon log as an example) and time information extraction (take G = 6 h,b=0 h as an example)

Deep-learning-based insider threat detection model

Our goal is to propose an insider threat detection model that can find out time-related insider threat behaviors better while maintaining high detection performance against time-independent insider threat behaviors.

In order to achieve our goal, we select stacked bi-directional LSTM (BiLSTM) to capture user behavioral rhythm. Stacked BiLSTM has a sequential structure, which means that stacked BiLSTM can naturally model the sequence relationship of input data, that is, it can directly model the relative time information of behaviors. Combined with our feature extraction method, stacked BiLSTM can model both absolute time information and relative time information of behaviors. We also add an FNN layer to make our model have better detection performance for time-independent insider threat behavior.

Input data can be reconstructed using the encoder-decoder structure and insider threat behaviors can be effectively detected with the help of reconstruction errors (Malhotra et al. 2016). Therefore, we will build the model according to the encoder-decoder structure, where the encoder consists of stacked BiLSTM and FNN and the decoder is FNN. Taking $G = 6h$, $b = 0h$ as an example, the structure of BRITD is shown in Fig. 5, where x_1, x_2,x_3, and x_4 represent the statistical values of user behavior features at 0:00-6:00, 6:00-12:00, 12:00-18:00 and 18:00-24:00 respectively.

The loss function is the mean square error between the model reconstruction result and the original input data. As shown in formula (1), x_t^u represents the input x of the time step t and u represents the specific user. $x_t^{'u}$ represents the reconstruction value of x_t^u. T represents the number of the time step. The result of this function is also the outlier.

$$Loss = o^u = \frac{1}{T}\sum_{t=0}^{T-1}(\boldsymbol{x}_t^{'u} - \boldsymbol{x}_t^u)^2 \tag{1}$$

For the above model, we use a semi-supervised approach for training. The data without insider threat instances are used as training data. In the test, a larger outlier indicates a more significant anomaly in the input data.

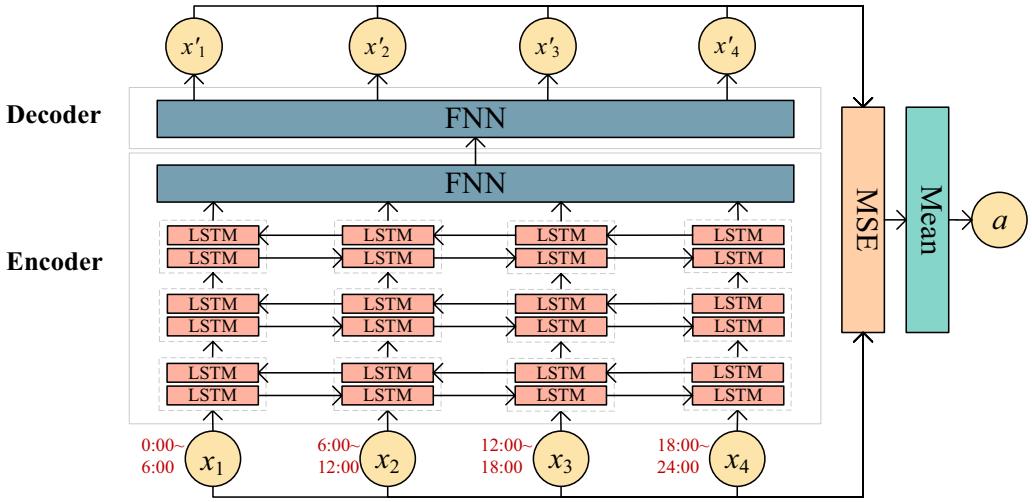


Fig. 5 The structure of our insider threat detection model (take G=6 h,b=0 h as an example)

Experiments

In this section, we will experimentally answer 4 research questions. We conduct a comprehensive analysis of our proposed insider threat detection scheme in our experiments. In the following, we detail our experimental settings and experimental results.

Experimental settings

Dataset

The CMU CERT dataset (Glasser and Lindauer 2013) is a synthetic insider threat dataset developed by the CERT division of Carnegie Mellon University. The dataset has now been released in 10 versions, of which we choose version 6.2 for our experiments. This dataset simulates a series of behavior logs (including login logs, HTTP logs, file logs, email logs, device logs, and so on) of 4,000 employees over 516 days. Insider threat events were injected into the normal behavior logs.

Referring to the classification of behavioral anomalies (Chandola et al. 2009), we classify the insider threat behaviors into the following four categories.

- Point Threat Behaviors: a single log entry is threatening.
- Contextual Threat Behaviors: a single log entry at a specific time is threatening, while the entry itself is benign.
- Coarse-grained Collective Threat Behaviors: the collection of multiple benign log entries is threatening; the related log entries refer to the same behavior, and the threat manifests in the frequency of the behavior.
- Fine-grained Collective Threat Behaviors: the collection of multiple benign log entries is threatening; the related log entries refer to multiple behaviors.

Among them, contextual threat behaviors and fine-grained collective threat behaviors are strongly correlated with time information. In particular, contextual threat behaviors are related to absolute time information. Coarse-grained collective threat behaviors are not related to absolute time information and the order of behaviors (relative time information). Thus, Coarse-grained collective threat behaviors are regarded as time-independent threat behaviors together with point threat behaviors in our experiments. In addition, it should be noted that each type of threat behavior will affect the time distribution of user behavior.

In this paper, we use the feature sequence construction method that is able to encode the absolute time information implicitly and fit the behavior rhythm to extract the time information of user behavior. We use the deep learning model with sequence structure to learn the behavior time information for the purpose of detecting time-related contextual threat behaviors and fine-grained collective threat behaviors. We use the method of converting behavior log entries into behavior tokens and counting the behavior tokens to extract the behavior type, and behavior frequency information. We use the deep learning model to learn the above information for the purpose of detecting time-independent point threat behaviors and coarse-grained collective threat behaviors (the FNN layer of the deep learning model is important for detecting time-independent threat behaviors).

Table 1 Summary of the dataset

Scenario	Category	Training set (days)	Testing set (days)
Scen. 1	Contextual anomalies	150	25
Scen. 2	Coarse-grained collective anomalies	150	166
Scen. 3	Point anomalies & Fine-grained Collective Anomalies	150	7
Scen. 4	Point anomalies	150	206
Scen. 5	Fine-grained collective anomalies	150	206

The insider threat events of CMU CERT v6.2 cover 5 threat scenarios. The categories of insider threat behaviors contained in each insider threat scenario are shown in Table 1. Scenario 1, scenario 3, and scenario 5 contain time-related threat behaviors, while scenario 2 and scenario 4 contain only time-independent threat behaviors.

We select a subset of the CMU CERT v6.2 dataset for model training and testing. Each user contained in this subset generated insider threat behaviors. These insider threat behaviors cover all five insider threat scenarios in the CMU CERT v6.2 dataset. The statistics of our training and testing sets are shown in Table 1. We extract each user's behavior logs from the original dataset and arrange them in chronological order. Subsequently, we select the first 150 days of behavior data in each user's behavior log as the training set and the rest as the test set.

Research questions

We try to answer the following four research questions experimentally.

- How to prove the rationality and effectiveness of our proposed adaptive algorithm for feature extraction?
- Whether our proposed insider threat detection scheme is more advantageous than the current common insider threat detection models.
- What is the role of each part of our proposed deep learning-based insider threat detection model?
- Can the accuracy and running speed of our proposed insider threat detection scheme adapt to real-world application environments?

We design a series of experiments to prove the rationality and effectiveness of our adaptive algorithm for feature extraction to answer Question 1. Then, we use the adaptive algorithm to select the best feature extraction scheme for each user to be tested. We set up a series of comparison experiments to demonstrate the advantages of our scheme in terms of accuracy and precision to answer Question 2. We verify the role of various parts of our model through ablation experiments to answer Question 3. We provide a comprehensive analysis of the accuracy and running speed of our scheme to answer Question 4.

Hyperparameters setting and evaluation indicators

The hyperparameters of our model are: $units = 64$, $lr = 0.001$, $batchsize = 16$, $epoch = 500$; the layers number of stacked BiLSTM is 3; Adam optimizer is selected. The main model evaluation metrics we choose include AUC and P (Precision).

Our P is the max precision when $recall = 1$. We use the minimum anomaly value of true anomaly instances as the threshold σ. Instances with anomaly values greater than σ are predicted anomaly instances, and instances with anomaly values less than σ are predicted normal instances. P is calculated based on the above prediction results. Due to the different amounts of data for 5 insider threat scenarios, we calculate AUC and P for each insider threat scenario respectively. The average of the test results of the 5 scenarios is calculated as the overall experimental result. The granularity of the evaluation is 24 h (day). The results of all experiments are the average of the results of 10 replicates.

RQ 1: effectiveness proof of adaptive algorithm

In response to Question 1, we propose the following conjecture: Using a time granularity consistent with the user-day behavior rhythm will help the model to learn the normal user behavior pattern more efficiently, thereby greatly improving the accuracy and precision of insider threat detection. We can use the covariance of feature sequences to assess the consistency of the time granularity with the user-day behavior rhythm. If the consistency between the time granularity and the user-day behavior rhythm is high, the user's active period and the user's rest period will be classified into different nodes in feature sequences, and the covariance of the feature sequences will be big.

We use the following experiment to prove this conjecture: First, we calculate the covariance of each feature sequence. We perform maximum normalization on the feature sequences to exclude the effect of the fluctuation of the total number of user-day behaviors on the average covariance. Second, we test our detection scheme at different time granularity G and starting bias b. Our conjecture is considered valid if the time granularity G and starting bias b that maximizes the feature sequence covariance is consistent with G and b that makes the detection scheme achieve optimal performance. We make the time granularity $G = 1, 2, 3, 4, 6, 8, 12, 24\,h$ and $b = 0\,h$. Such a setting can make the gap between the

Table 2 The Covariance of the pre-processed feature sequences

Time granularity (h)	1	2	3	4	6	8	12	24
Scen.1($\times 10^{-5}$)	0.86	1.14	1.40	1.32	**2.05**	1.67	0.51	0
Scen.2($\times 10^{-5}$)	1.09	1.48	1.86	2.11	2.41	**2.69**	1.19	0
Scen.3($\times 10^{-5}$)	0.77	1.04	1.18	1.50	2.00	**2.28**	0.44	0
Scen.4($\times 10^{-5}$)	0.81	1.12	1.23	1.65	1.56	**1.97**	0.95	0
Scen.5($\times 10^{-5}$)	0.93	1.36	1.55	1.75	2.35	**2.46**	0.67	0

*Bold indicates optimal results

covariance of feature sequences obvious enough to make the experimental results more clear.

The covariances of the feature sequences are shown in Table 2. In scenario 1, when the time granularity is 6 h, the covariance of the feature sequence is the highest; in scenarios 2, 3, 4, and 5, when the time granularity is 8 h, the covariance of the feature sequence is the highest. The AUC and P of our scheme are shown in Fig. 6. In scenario 1, the AUC and P reach the maximum values when the time granularity is 6 h; in scenarios 2, 4, and 5, the AUC and P reach the maximum values when the time granularity is 8 h; in scenario 3, the AUC and P reach the maximum values when the time granularity is 2, 3, 4, 6, and 8 h.

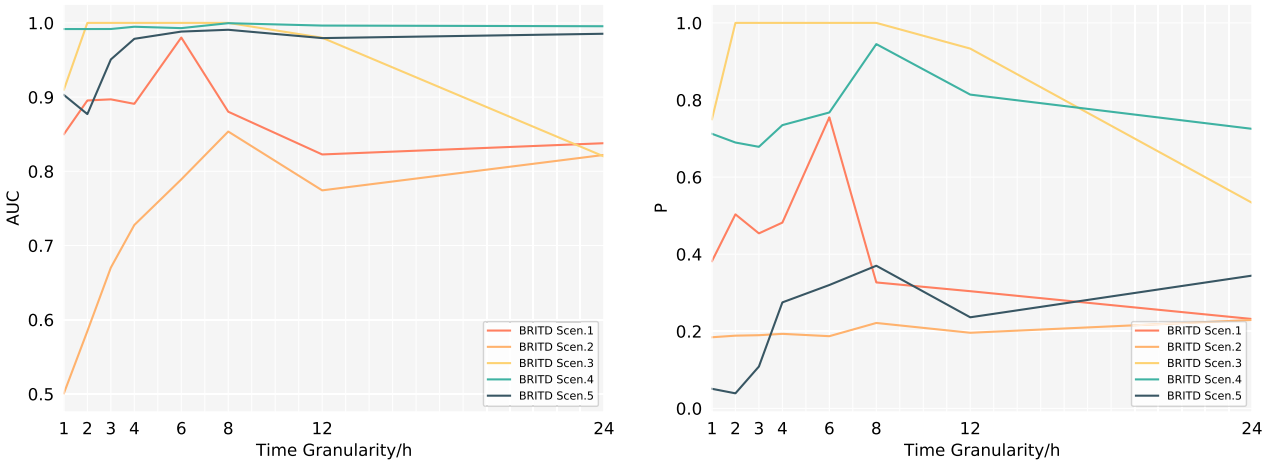
The results of the above two experiments are consistent, which means that using the feature sequence that is suitable for the user-day behavior rhythm can improve the accuracy and precision of insider threat detection, and we can determine the best time granularity and starting bias by calculating the covariance of the feature sequences to obtain more suitable feature sequence.

Based on the scheme proposed in this paper, we calculated the covariance of the feature sequence from each sequence construction scheme for each user to be tested. Based on the calculation results, we selected the optimal feature sequence construction scheme for each user (each threat scenario has one user): $s_{best}^{u_{Scen.1}} =< 12\,h, 7\,h >, s_{best}^{u_{Scen.2}} =< 12\,h, 8\,h >, s_{best}^{u_{Scen.3}} =< 12\,h, 5\,h >, s_{best}^{u_{Scen.4}} =< 12\,h, 8\,h >, s_{best}^{u_{Scen.5}} =< 12\,h, 8\,h >$. According to the optimal feature sequence construction scheme obtained, the optimal evaluation results of the scheme proposed in this paper can be obtained, as shown in Table 3.

RQ 2: comparison experiments

In response to Question 2, we set up a series of comparison experiments to demonstrate the effectiveness and superiority of our scheme.

First, We migrate various anomaly-based models including Gavai et al's work [53], Rashid et al's work (Rashid et al. 2016), Tuor et al's work (Tuor et al. 2017), etc. on the condition that the models are comparable to ours to analyze in detail where the advantages of our scheme lie. In addition, we also compare our scheme



(a). AUC (b). P

Fig. 6 The results of effectiveness proof of adaptive algorithm experiment for 5 insider threat scenarios

Table 3 Comparison experiment results of BRITD and 9 baselines

Model	Scen. 1		Scen. 2		Scen. 3		Scen. 4		Scen. 5		Overall	
	AUC	P	AUC	P	AUC	P	AUC	P	AUC	P	AUC	P
BRITD*	**0.9957**	**0.9333**	**0.8707**	**0.2629**	**1.0000**	**1.0000**	0.9999	0.9900	**0.9985**	**0.8500**	**0.9730**	**0.8072**
HMM-BRITD	0.9152	0.4267	0.6310	0.1861	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9049	0.0504	0.8902	0.5326
IF-6h-129×**4**	0.8970	0.5129	0.5224	0.1839	0.9100	0.7333	0.9668	0.3023	0.4439	0.0152	0.7480	0.3495
OCSVM-6h-129×**4**	0.9394	0.6000	0.6725	0.1758	**1.0000**	**1.0000**	0.8009	0.0763	0.4780	0.0093	0.7782	0.3723
FNN-6h-129×**4**	0.9197	0.4771	0.6240	0.2074	**1.0000**	**1.0000**	0.9917	0.7020	0.9288	0.0654	0.8928	0.4904
BiLSTM-6h-129×**4**	0.7000	0.2050	0.5753	0.1873	**1.0000**	**1.0000**	0.9474	0.3227	0.6180	0.0130	0.7681	0.3456
IF-24h	0.8545	0.3976	0.6737	0.2241	0.8800	0.6400	0.9999	0.9900	0.7541	0.0228	0.8325	0.4549
OCSVM-24h	0.8030	0.2308	0.6159	0.1758	0.8000	0.5000	0.9566	0.2813	0.9024	0.0476	0.8156	0.2471
FNN-24h	0.7561	0.2688	0.6384	0.1891	0.8600	0.6667	0.9983	0.9274	0.9298	0.2759	0.8365	0.4656
BiLSTM-24h	0.8942	0.3750	0.6628	0.2194	0.6998	0.4000	0.9958	0.7366	0.7026	0.0193	0.7910	0.3501

*Bold indicates optimal results; star indicates our model

directly with the works of Tuor et al. (2017), Meng et al. (2020), Dr et al. (2022) and Villarreal-Vasquez et al. (2023) to further demonstrate the superiority of our scheme.

Second, Yuan et al. (2020) propose a threat detection scheme based on the combination of Transformer and FNN considering time embedding; Wu and Li (2021) propose a threat detection scheme based on an ensemble of NN and RF. We compare our scheme with the above two classification-based schemes to demonstrate that our scheme still has advantages in the face of other schemes that consider user behavior time and schemes based on combined detection models.

Comparison experiment with anomaly-based schemes

In the comparison experiment with anomaly-based schemes, our baselines include Isolated Forest (IF), One-Class Support Vector Machine (OCSVM), Feedforward Neural Network (FNN), Bi-directional LSTM (BiLSTM), and Hidden Markov Model (HMM). The baseline HMM-BRITD uses the same input data as BRITD, only the detection model is replaced with HMM. BiLSTM uses the traditional setting that the location information of nodes in the input sequence is independent of absolute time information and only reconstructs the last node of the input sequence. All baselines (except HMM-BRITD) use 24-hour time granularity. In addition, the 6 h-129×4 behavioral statistics is added by referring to the work of Tuor et al. (2017). The 6-h-129×4 behavioral statistics refers to the division of the user behavior features into four more types according to the time of behavior occurrence (occurring between 0:00-6:00/6:00-12:00/12:00-18:00/18:00-24:00), so the number of user behavior features has changed from 129 to 129×4.

The hyperparameter settings for BiLSTM and FNN keep consistent with BRITD. The hyperparameters of the IF and OCSVM are determined using the GridSearchCV tool in the sklearn library. The model structure and hyperparameter settings of HMM are referred to in the work of Rashid et al. (2016). That is, for BiLSTM: units = 64, lr = 0.001, batchsize = 16, epoch = 500, the layers number of BiLSTM is 3; for FNN: units = 64, lr = 0.001, batchsize = 16, epoch = 500, the layers number of FNN is 3; for IF: hyperparameters are determined using the GridSearchCV tool of the sklearn library; for OCSVM: hyperparameters are determined using the GridSearchCV tool of the sklearn library; for HMM: states_num=15, max_iters=20, max_restarts=5, $\epsilon_{convergence}$=0.01, $\epsilon_{restarts}$=0.1, λ =0.05.

In the experiment, each model calculates the user-day behavioral outliers. We evaluate the model based on the outliers.

As can be seen in Table 3, the overall experimental results of BRITD outperform all the baselines. In scenarios 1, 2, 3, and 5, our model achieves better experimental results than any other baselines, especially in scenario 1 which contains the context threat behaviors. In scenario 4, the AUC and P of our model are very close to 1.0000.

The experimental results of HMM-BRITD outperform most other baselines, which indicates that our feature extraction approach is effective. However, HMM-BRITD cannot outperform BRITD overall, which indicates that our deep learning detection model has an advantage in behavioral pattern fitting.

Meanwhile, the experimental results of FNN are generally higher than those of BiLSTM, which indicates that the traditional setting of BiLSTM does not allow it to obtain more information than FNN.

Table 4 Comparison experiment results of BRITD and 4 anomaly-based works

Scheme	Year	AUC	P	F1
*BRITD**	–	**0.9730**	**0.8072**	**0.8540**
Tuor et al. (2017)	2017	0.8658	0.3827	0.4805
Meng et al. (2020)	2020	0.8388	0.4471	0.5406
Dr et al. (2022)	2022	0.8757	0.3850	0.4850
Villarreal-Vasquez et al. (2023)	2023	–	0.1608	0.1689

*Bold indicates optimal results; star indicates our model

Table 5 Comparison experiment results of BRITD and 2 classification-based works

Scheme	Numbers of Malicious Instances	P	Recall	F1
*BRITD**	0	0.8072	**1.0000**	**0.8540**
Yuan et al. (2020)	5	0.0047	0.8140	0.0096
	8	0.8824	0.3750	0.5263
	10	0.7333	0.5789	0.6471
	15	**0.9200**	0.6970	0.7931
Wu and Li (2021)	15	0.9091	0.6061	0.7273

*Bold indicates optimal results; star indicates our model

Moreover, in the time-related threat scenarios, the experimental results of the −6 h-129X4 models are generally higher than the experimental results of the −24 h models, but they still can not exceed the BRITD. This shows that considering time information can improve the sensitivity of the detection scheme to time-related threat behaviors, but existing feature extraction schemes are not enough for the scheme to fully capture the time-related behavioral features. The behavior feature extraction and detection model in BRITD coordinate with each other in capturing time information, and can also self-adjust according to the behavior patterns of specific users, which helps to obtain the time information of user behavior more comprehensively.

Besides, we find that baselines cannot be applied to all threat scenarios simultaneously. For example, IF-24 h does not apply to scenarios 3 and 5; FNN-6 h-129×4 does not apply to scenario 2; FNN-24 h does not apply to scenarios 1 and 3. In contrast, our model can be applied to all 5 insider threat scenarios.

Furthermore, we directly compare our scheme with 4 classical or novel anomaly-based insider threat detection schemes, which include the works of Tuor et al. (2017), Meng et al. (2020), Dr et al. (2022), and Villarreal-Vasquez et al. (2023). Table 4 shows the experimental results of our scheme and the other 4 schemes on the same CERT v6.2 dataset. Our scheme outperforms all other works, which again proves the advantage of our scheme.

Comparison experiment with classification-based schemes

Table 5 shows the detection performance comparison results of Yuan et al. (2020) scheme, Chunhui Wu et al. (2021) scheme, and our scheme. The table indicates the various numbers of malicious instances that these two classification-based schemes use for training. The classification-based schemes usually require a fully labeled training set. In reality, labeled user behavior data, especially malicious behavior data, is difficult to obtain. Without the need for labeled malicious instances, the F1 score of our scheme can still exceed the above two classification-based schemes. Therefore, our scheme is better.

RQ 3: ablation experiments

In the model design stage, we select stacked BiLSTM and FNN to construct the encoder. We expect the stacked BiLSTM to capture the absolute time information and relative time information, and the FNN to improve the ability of the encoder in capturing other features than time-related features in the sequence. In this section, we will verify whether the role of stacked BiLSTM and FNN are consistent with our expectations through ablation experiments.

The role of stacked BiLSTM

The experimental results are shown in Fig. 7. In this experiment, we use two baselines: BRITD-SS and BRITD-FNN. In BRITD-SS, we fixed the time step number of the BiLSTM layer in BRITD as 1. In BRITD-FNN, we replace the BiLSTM layer with the FNN layer. The results show that BRITD has the best experimental results in most of the threat scenarios, which is because BRITD-SS and BRITD-FNN cannot extract the time information of data like BRITD can. The only exception is scenario 4, which is due to slight overfitting: BRITD pays more attention to the behavior time information, while the threat behaviors in Scenario 4 are completely unrelated to the time information and time distribution of behaviors (the threat behaviors in Scenario 4 are all point threat behaviors). A small number of behaviors that are benign but inconsistent with the training data have a large reconstruction error in BRITD, which increases the false positive rate and reduces the accuracy of detection. BRITD mostly has the highest improvement with 6-h or 8-h time granularity, which

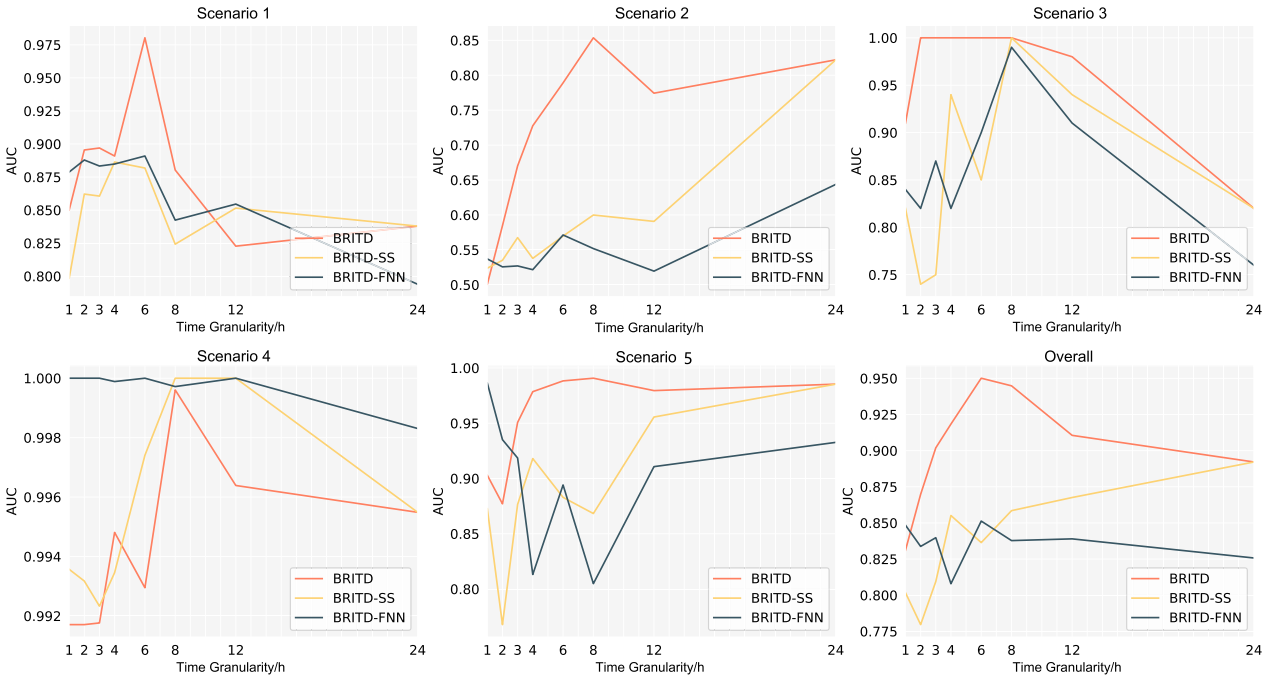


Fig. 7 Experimental results of BRITD, BRITD-FNN and BRITD-SS under 8 time granularitys and 5 insider threat scenarios

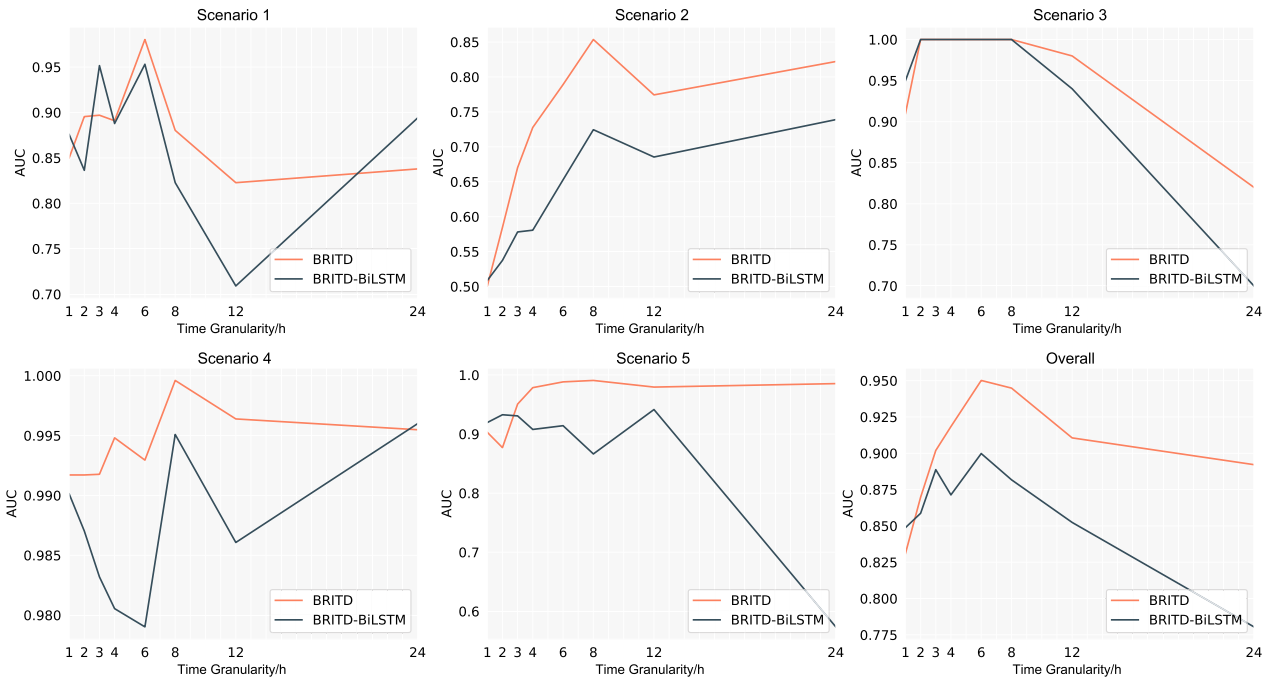


Fig. 8 Experimental results of BRITD and BRITD-BiLSTM under 8 time granularitys and 5 insider threat scenarios

is also consistent with the relevant findings in *Effectiveness Proof of Adaptive Algorithm*.

As a whole, the BiLSTM layer has a boosting effect on the detection of contextual threat behaviors (scenario 1) and collective threat behaviors (scenarios 2/3/5). And BRITD obtains the best overall detection results, which indicates that the Stacked BiLSTM layer can learn time-related behavioral features, and our time

information extraction scheme can improve model performance.

The role of FNN

The experimental results are shown in Fig. 8. In this experiment, we use one baseline: BRITD-BiLSTM. In BRITD-BiLSTM, we remove the FNN layer from the encoder in BRITD. According to Fig. 8, in scenario 1, The advantage of BRITD over BRITD-BiLSTM is not obvious. This is probably because for the detection of contextual threat behaviors in scenario 1 Stacked BiLSTM plays the main role. In the other scenarios, BRITD is significantly better than BRITD-BiLSTM, which indicates that FNN can indeed improve the detection effect of point threat behaviors, coarse-grained collective threat behaviors, and even fine-grained collective threat behaviors.

RQ 4: performance analysis

Accuracy and threshold setting

As shown in Fig. 9, The red dots representing the outliers of the insider threat instances basically coincide with the darker part of the histogram. This means that outliers of insider threat instances calculated by our model are generally higher than outliers of benign instances. However, the selection of the optimal threshold still depends on the proportion of insider threat instances on the basis of the existing experimental data because the number of instances to be tested is not large enough. In practical application environments, the proportion of insider threat behaviors in all user behaviors will be smaller, which is acceptable to security administrators.

Training and test time

As shown in Table 6, the total test time of all the test instances of a single user is less than 3 s. The training time of a single model is affected by the time granularity of the detection scheme: The longest training time is 426.71 s with 1-h time granularity, and the shortest training time is 65.86s with 24-h time granularity. The detection scheme with the 12-h time granularity we selected needs 78.49 s to train. The above training time and test time are within the acceptable range.

In conclusion, the accuracy and running time of our proposed insider threat detection scheme are acceptable in practical application environments. However, it should be noted that the CERT dataset we used in the experiments is a synthetic dataset. The CERT dataset simulates a series of operational behaviors of internal personnel in a virtual organization system using the relationship graph model, asset graph model, behavior model, psychological model, and topic model (Glasser and Lindauer 2013),

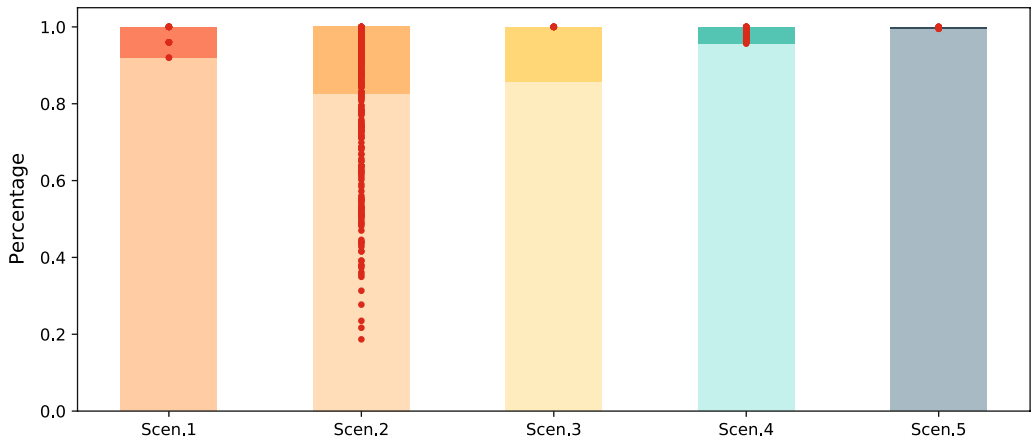


Fig. 9 The ordinate of the red dots represents the percentage of the outliers of the insider threat instances obtained by BRITD among the outliers of all the instances to be tested; The dark part of the histogram represents the percentage of insider threat instances in all instances to be tested, and the light colored part represents the percentage of benign instances to all instances to be tested

Table 6 The training time and test time of BRITD

Time granularity(h)	1	2	3	4	6	8	12	24
Training time(s)	426.71	218.90	166.31	158.81	100.10	93.08	78.49	65.86
Test time(s)	2.30	2.38	2.20	2.19	2.52	2.16	2.20	2.24

with the aim of achieving a comprehensive simulation of insider threat scenarios. As a synthetic dataset, there is still a discrepancy between its data distribution and scenarios from the real world. Therefore, the experimental results obtained under the CERT dataset cannot fully reflect the detection performance of our scheme in real-world scenarios. Unfortunately, due to the covertness of insider threat activities and privacy protection issues of real-world organizational employees, we have not yet found a well-labeled real-world insider threat dataset containing comprehensive user behavior data, which hinders us from rigorously proving the detection performance of our scheme in real-world scenarios. The experimental findings and conclusions that we came to using the CERT dataset remain useful for reference, though. Meanwhile, collecting real-world datasets and using them to demonstrate the availability of insider threat models in real environments will be an interesting issue for future exploration.

Discussion

Self-attention

Considering the difficulty of LSTM to capture the long-distance dependence of feature sequences, we try to capture it by adding a self-attention layer to the model. The results show that the self-attention layer can be useful when the time granularity is small and the feature sequence is long. When the time granularity becomes larger and the feature sequence becomes shorter, the stacked BiLSTM alone is sufficient to capture the time information. The additional self-attention layer may instead cause overfitting, increase the false positive rate and decrease the accuracy of the detection model.

Trust management and access control

The occurrence of insider threats is closely related to knowledge, access authority, and trust (Probst et al. 2008). Insider threat activities can be seen to a certain extent as the abuse of knowledge, access authority, and trust by organizational members. From the perspective of an organization, it means that careful management of privilege and trust is crucial to ensuring the security of organization systems. Our proposed insider threat detection scheme, as an indicator of threats, can be an effective tool for privilege and trust management.

Specifically, our detection scheme calculates the anomaly score of user behavior, which can serve as a decision-making basis for access control. Users with higher anomaly scores need to be granted permissions more cautiously or have certain key permissions disabled. It is worth mentioning that, with reference to the works of Meng et al. (2017, 2020), we found that our insider threat detection scheme can also support the trust management of a distributed intrusion detection system, which can help improve the robustness of the intrusion detection system. According to the research of Nurse et al. (2014), users with insider threat risks may disable security tools to provide convenience for their possible threat activities, which can negatively affect the reliability of intrusion detection systems of the organization. The anomaly scores of user behaviors calculated by us can be easily converted into trust scores (for example, taking the reciprocal). The higher the anomaly score of the user behavior, the less trustworthy the corresponding intrusion detection node. The detection results returned by the intrusion detection node with low trust scores should be assigned lower weights.

Table 7 The log attribute values included in the behavior features

Log Type	Attribute	Value
commonly used*	PC	selfPC
		otherPC
		sharedPC
logon	Activity	Logon
		Logoff
device	Activity	Connect
		Disconnect
email	Receive/Send	Send with cc
		Send with cc and bcc
		Receive
	correspond with	inside the organization
	Attach	outside the organization
file	Activity	copy
		delete
		write
		open
	Media	to USB
		from USB
	Decoy	disk
http	Activity	down
		up
		visit

Limitation

The detection granularity of our model is not fine enough. The accuracy and precision of the model, especially in Scenario 2, still have space for improvement. Moreover, user behavioral rhythms may change over time, but our user adaptive scheme fails to take these changes into account. Besides, the CERT dataset used in our experiments is a synthetic dataset, which makes it difficult to demonstrate the effectiveness of our method

in real-world situations. Finding or collecting a real-world dataset is crucial for research on insider threat detection. These problems need to be further studied and solved.

Conclusion

Aiming at the problem that the existing researches fail to make full use of the user behavior time information, we propose a deep learning-based insider threat detection scheme. In this scheme, the time information conversion method is adopted to convert numerical time information into location information, so that the model can obtain user behavior rhythm from user behavior data. The proposed scheme is also able to select the best feature extraction scheme for a specific user based on the covariance, which means that it is user adaptive. The effectiveness of our insider threat detection scheme is validated using the CMU CERT v6.2 dataset. The experimental results show that our scheme has obvious advantages over the baselines.

Appendix

Details of behavior features

Table 7 shows all the log attributes and their values that we use in behavioral feature extraction.

Details of baselines

In comparison experiments, we compare our proposed insider threat detection scheme with 15 baselines. Among them, there are 13 anomaly-based baselines, namely: IF-6 h-129×4, OCSVM-6 h-129×4, FNN-6 h-129×4, BiLSTM-6 h-129×4, IF-24 h, OCSVM-24 h, FNN-24 h, BiLSTM-24 h, HMM-BRITD as well as the work of Tuor et al. (2017), the work of Meng et al. (2020), the work of Dr et al. (2022) and the work of Villarreal-Vasquez et al. (2023). There are 2 classification-based baselines, which are the works of Yuan et al. (2020) and Wu and Li (2021).

9 of the anomaly-based baselines are named in the pattern of "model name-input data type". A detailed introduction to them is as follows:

- **IF & OCSVM**: IF Gavai et al. (yyyy) and OCSVM Schlkopf et al. (1999) are commonly used machine learning models for outlier detection tasks. We use these two as baselines for comparison experiments in this paper. The input data of IF-24 h and OCSVM-24 h is a single behavior feature vector with a data granularity of 24 h. The input data of IF-6 h-129×4 and OCSVM-6 h-129×4 is a single feature vector with a data granularity of 24 h, and each original behavior token is divided into four tokens according to the behavior time. The number of modified behavioral tokens is 129×4. IF and OCSVM directly outputs outliers after receiving input data. We utilize the GridSearchCV tool in the sklearn library to determine the hyperparameters of IF and OCSVM.
- **FNN**: We set the FNN as the baseline referring to Tuor et al. work (Tuor et al. 2017). Our FNN model is an autoencoder with a fully connected layer as the decoder. We take the autoencoder reconstruction error as the outlier. The input data for FNN-24 h is the same as for IF-24 h and OCSVM-24 h. The input data for FNN-6 h-129×4 is the same as for IF-6 h-129×4 and OCSVM-6 h-129×4.
- **BiLSTM**: We set BiLSTM as the baseline referring to Tuor et al.'s work (Tuor et al. 2017). The BiLSTM model is an autoencoder with a fully connected layer as the decoder. The BiLSTM model receives a sequence of behavior feature vectors as input and reconstructs the last node of the sequence. We take the autoencoder reconstruction error as the outlier. The length of the input sequence of the BiLSTM is fixed. The input data for BiLSTM-24 h is a sequence of behavior feature vectors with a data granularity of 24 h. The input data for BiLSTM-6 h-129×4 is a sequence with a data granularity of 24 h and a number of behavior tokens of 129×4.
- **HMM**: We construct the HMM-BRITD as the baseline referring to Rashid et al. work (Rashid et al. 2016). The input data for HMM-BRITD is the same as for BRITD. HMM-BRITD directly outputs outliers after receiving the input data.

The other 4 anomaly-based baselines are reproductions of related works. A detailed introduction to them is as follows:

- Tuor et al. (2017): Tuor et al. extract 408 behavior features from user behavior logs and obtain the user-day feature vectors by counting the behavior features. During the behavior feature extraction, Tuor et al. divide each type of behavior into four more types according to when the behavior occurred (i.e., 12am–6am, 6am–12pm, 12pm–6pm, and 6pm–12am). They use the deep feedforward neural network to reconstruct the user behavior feature vectors while fitting the user behavior feature distribution with a multivariate Gaussian distribution to obtain a decomposable reconstruction error (which is also the anomaly score).
- Meng et al. (2020): Weizhi Meng et al. use Euclidean distance and Euclidean norm to calculate the difference between two behavior profiles. The greater the difference between the test behavior profiles and the benign behavior profiles, the higher the abnormality

Table 8 The relationship between Epoch and AUC of BRITD

Epoch	AUC
100	0.8749
200	0.9186
300	0.9362
400	0.9373
500	0.9501
600	0.9486
700	0.9517
800	0.9568
900	0.9524
1000	0.9531

of the test behavior profiles, and the lower the trustworthiness of the corresponding user node.

- Dr et al. (2022): Jongmin Yu et al. use the RNN-based autoencoder to reconstruct behavior sequences extracted from security audit logs to detect anomaly behaviors. They use adversarial learning to make the distribution of hidden layer features closer to a normal distribution to reduce the uncertainty in the hidden layer feature space. The objective function of the model consists of both the reconstruction loss and adversarial loss.
- Villarreal-Vasquez et al. (2023): Miguel Villarreal-Vasquez et al. collect activity datasets from enterprise EDR systems and extract event sequences from the datasets. They then use LSTM to learn the patterns of benign event sequences and predict the probability of each event becoming the next event. The most repeated probability is considered the threshold. Events with probabilities above the threshold are benign events, while events with probabilities below the threshold are anomaly events.

There are 2 classification-based baselines which are described below:

- Yuan et al. (2020): Shuhan Yuan et al. treat the behavior token sequences as sentences, treat the behavior tokens as words, and use the Transformer to encode the behavior token sequence. During training, referring to the training method of BERT, the scheme first uses a method similar to the masked language model (MLM) to pre-train the Transformer model. Subsequently, the scheme performs supervised fine-tuning of the model: it treats the first vector of the encoded result as the representation vector of the entire behavior token sequence and then uses a fully connected network to perform binary classification on this representation vector. The scheme obtains behavioral time information by replacing the original position encoding of the Transformer with time encoding.
- Wu and Li (2021): Chunhui Wu et al. first use the Fast Correlation-Based Filter (FCBF) algorithm for feature selection on activity features extracted from system logs. They then use an ensemble of NN and RF to detect threat activities. Meanwhile, they construct a decision component to choose the optimal detection algorithm based on the detection accuracy.

The setting of epoch

We set Epoch to 500 in our experiments instead of setting epoch to a value less than 200 as in other common studies. It is because we find that small epoch tends to lead to underfitting problem in our experiments. The relationship between the value of Epoch and the detection performance of BRITD (measured in terms of AUC) is shown in Table 8.

In this experiment, the other hyperparameters are set as follows: units = 64, lr = 0.001, batchsize = 16, the layers number of BiLSTM is 3, the length of the input sequence is 4, and the time granularity is 6 h.

As can be seen in the table, with the increase of epoch, the AUC of detection results for the test set also shows an increasing trend basically. However, when the epoch is greater than 500, the value of AUC begins to level off. At the same time, the increase of epoch also leads to the increase of the training time of the model. Considering both the detection performance and training time of the model under different epochs, we decide to set the epoch to 500.

Abbreviations

BRITD	Behavior rhythm insider threat detection
Stacked BiLSTM	Stacked bidirectional LSTM
LSTM	Long short term memory
RNN	Recurrent neural network
FNN	Feedforward neural network
OCCRF	One-Class Conditional Random Field
ARAE	Adversarial recurrent autoencoder
RF	Random forest
MLP	Multilayer perceptron
MultinomialNB	Multinomial Naive Bayes
LP	Label propagation
LS	Label spreading
ST	Self-training
SMOTE	Synthetic minority oversampling technique
MKC	Multiple kernel clustering
MTD	Moving target defense
MSNs	Medical smartphone networks
CMU	Carnegie Mellon University
AUC	Area under curve
P	Precision
IF	Isolation forest
OCSVM	One-Class SVM
SVM	Support vector machine

HMM	Hidden Markov model
MLM	Masked language model
FCBF	Fast Correlation-Based Filter

Acknowledgements
I express my gratitude to my respected supervisor for her guidance, and to the other teachers and schoolmates who provided comments on this work.

Author contributions
SS proposed the BRITD scheme with careful experiments and wrote the manuscript. NG and YZ joined the discussion of the work and provided significant suggestions on the behavior feature extraction. NG, YF and CM reviewed the manuscript and gave suggestions on the revision of the details of the article. All authors read and approved the fnal manuscript.

Funding
This work is supported by the National Key Research and Development Program of China.

Availability of data and materials
The CMU CERT dataset analyzed during the current study is available in the kilthub repository, https://doi.org/10.1184/R1/12841247.v1.

Declaration

Competing interests
The author declares no competing interests.

Author details
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. [2]State Key of Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China. [3]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.

Received: 23 November 2022 Accepted: 11 September 2023
Published online: 02 January 2024

References

Al-Mhiqani MN, Ahmad R, Abidin ZZ, Abdulkareem KH, Mohammed MA, Gupta D, Shankar K (2022) A new intelligent multilayer framework for insider threat detection. Comput Electr Eng 97:107597

Alsaheel A, Nan Y, Ma S, Yu L, Walkup G, Celik ZB, Zhang X, Xu D (2021) ATLAS: A sequence-based learning approach for attack investigation. In: USENIX security symposium

Al-Shehari T, Alsowail RA (2021) An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. Entropy 23(10):1258. https://doi.org/10.3390/e23101258

Asif M, Aziz Z, Bin Ahmad M, Khalid A, Waris HA, Gilani A (2022) Blockchain-Based Authentication and Trust Management Mechanism for Smart Cities. Sensors 22(7):2604. https://doi.org/10.3390/s22072604

Ayed S, Hbaieb A, Chaari L (2023) Blockchain and trust-based clustering scheme for the IoV. Ad Hoc Netw 142:103093. https://doi.org/10.1016/j.adhoc.2023.103093

Azaria A, Richardson A, Kraus S, Subrahmanian VS (2014) Behavioral analysis of insider threat: a survey and bootstrapped prediction in imbalanced data. IEEE Trans Comput Soc Syst 1(2):135–155. https://doi.org/10.1109/TCSS.2014.2377811

Baracaldo N, Palanisamy B, Joshi J (2019) G-SIR: an insider attack resilient geo-social access control framework. IEEE Trans Dependable Secure Comput 16(1):84–98. https://doi.org/10.1109/TDSC.2017.2654438

Bu S-J, Cho S-B (2020) A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. Inf Sci 512:123–136. https://doi.org/10.1016/j.ins.2019.09.055

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15–11558. https://doi.org/10.1145/1541880.1541882

Chattopadhyay P, Wang L, Tan Y-P (2018) Scenario-based insider threat detection from cyber activities. IEEE Trans Comput Soc Syst 5(3):660–675. https://doi.org/10.1109/TCSS.2018.2857473

Dr J, Oh H, Kim M, Jung S (2022) Unusual insider behavior detection framework on enterprise resource planning systems using adversarial recurrent autoencoder. IEEE Trans Industr Inf 18(3):1541–1551. https://doi.org/10.1109/TII.2021.3090362

Gavai G, Sricharan K, Gunning D, Hanley J, Singhal M, Rolleston R. Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data, 17

Gavai G, Sricharan K, Gunning D, Rolleston R, Hanley J, Singhal M (2015) Detecting insider threat from enterprise social and online activity data. In: Proceedings of the 7th ACM CCS international workshop on managing insider security threats, pp. 13–20. ACM, Denver Colorado USA. https://doi.org/10.1145/2808783.2808784. https://dl.acm.org/doi/10.1145/2808783.2808784 Accessed 16 Sept 2022

Glasser J, Lindauer B (2013) Bridging the gap: a pragmatic approach to generating insider threat data. In: 2013 IEEE security and privacy workshops, pp 98–104. IEEE, San Francisco, CA. https://doi.org/10.1109/SPW.2013.37. http://ieeexplore.ieee.org/document/6565236/ Accessed 14 Sept 2021

Homoliak I, Toffalini F, Guarnizo J, Elovici Y, Ochoa M (2019) Insight into insiders and IT: a survey of insider threat taxonomies, analysis, modeling, and countermeasures. ACM Comput Surv 52(2):30–13040. https://doi.org/10.1145/3303771

Hu T, Niu W, Zhang X, Liu X, Lu J, Liu Y (2019) An insider threat detection approach based on mouse dynamics and deep learning. Secur Commun Netw 2019:1–12. https://doi.org/10.1155/2019/3898951

Hu N, Tian Z, Lu H, Du X, Guizani M (2021) A multiple-kernel clustering based intrusion detection scheme for 5G and IoT networks. Int J Mach Learn Cybern 12(11):3129–3144. https://doi.org/10.1007/s13042-020-01253-w

Ispoglou KK, Austin D, Mohan V, Payer M (2020) FuzzGen: automatic fuzzer generation

Jiang J, Chen J, Gu T, Choo K-KR, Liu C, Yu M, Huang W, Mohapatra P (2019) Anomaly detection with graph convolutional networks for insider threat and fraud detection. In: MILCOM 2019–2019 IEEE military communications conference (MILCOM), pp 109–114. IEEE, Norfolk, VA, USA. https://doi.org/10.1109/MILCOM47813.2019.9020760. https://ieeexplore.ieee.org/document/9020760/ Accessed 25 June 2021

Kandias M, Gritzalis D, Stavrou V, Nikoloulis K (2017) Stress level detection via OSN usage pattern and chronicity analysis: an OSINT threat intelligence module. Comput Secur 69:3–17. https://doi.org/10.1016/j.cose.2016.12.003

Khandait P, Hubballi N, Mazumdar B (2021) IoTHunter: IoT network traffic classification using device specific keywords. IET Netw 10(2):59–75. https://doi.org/10.1049/ntw2.12007

Le DC, Zincir-Heywood N, Heywood M (2021) Training regime influences to semi-supervised learning for insider threat detection. In: 2021 IEEE security and privacy workshops (SPW), pp. 13–18. IEEE, San Francisco, CA, USA. https://doi.org/10.1109/SPW53761.2021.00010. https://ieeexplore.ieee.org/document/9474297/ Accessed 16 Sept 2022

Lin L, Zhong S, Jia C, Chen K (2017) Insider threat detection based on deep belief network feature representation. In: 2017 international conference on green informatics (ICGI), pp 54–59. IEEE

Liu L, De Vel O, Chen C, Zhang J, Xiang Y (2018) Anomaly-Based Insider Threat Detection Using Deep Autoencoders. In: 2018 IEEE international conference on data mining workshops (ICDMW), pp. 39–48. IEEE, Singapore, Singapore. https://doi.org/10.1109/ICDMW.2018.00014. https://ieeexplore.ieee.org/document/8637390/. Accessed 16 June 2022

Liu F, Wen Y, Zhang D, Jiang X, Xing X, Meng D (2019) Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise, 1777–1794. https://doi.org/10.1145/3319535.3363224

Lu H, Jin C, Helu X, Zhu C, Guizani N, Tian Z (2021a) AutoD: intelligent blockchain application unpacking based on JNI layer deception call. IEEE Netw 35(2):215–221. https://doi.org/10.1109/MNET.011.2000467

Lu H, Jin C, Helu X, Zhang M, Sun Y, Han Y, Tian Z (2021b) Research on intelligent detection of command level stack pollution for binary program analysis. Mobile Netw Appl 26(4):1723–1732. https://doi.org/10.1007/s11036-019-01507-0

Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G (2016) Lstm-based encoder-decoder for multi-sensor anomaly detection. CoRR. arXiv:1607.00148

Meng W, Li W, Xiang Y, Choo K-KR (2017) A bayesian inference-based detection mechanism to defend medical smartphone networks against insider

attacks. J Netw Comput Appl 78:162–169. https://doi.org/10.1016/j.jnca.2016.11.012
Meng W, Li W, Wang Y, Au MH (2020) Detecting insider attacks in medical cyber-physical networks based on behavioral profiling. Futur Gener Comput Syst 108:1258–1266. https://doi.org/10.1016/j.future.2018.06.007
Nasir R, Afzal M, Latif R, Iqbal W (2021) Behavioral based insider threat detection using deep learning. IEEE Access 9:143266–143274. https://doi.org/10.1109/ACCESS.2021.3118297
Nurse JRC, Buckley O, Legg PA, Goldsmith M, Creese S, Wright GRT, Whitty M (2014) Understanding insider threat: a framework for characterising attacks. In: 2014 IEEE security and privacy workshops, pp. 214–228. IEEE, San Jose, CA. https://doi.org/10.1109/SPW.2014.38. http://ieeexplore.ieee.org/document/6957307/ Accessed 11 Jan 2022
Pan M, Huang W, Li Y, Zhou X, Liu Z, Song R, Lu H, Tian Z, Luo J (2020) DHPA: dynamic human preference analytics framework: a case study on taxi drivers' learning curve analysis. ACM Trans Intel Syst Technol 11(1):1–19. https://doi.org/10.1145/3360312
Probst CW, Hunker J, Gollmann D, Bishop M (2008) Countering insider threats
Rashid T, Agrafiotis I, Nurse JRC (2016) A new take on detecting insider threats: Exploring the use of hidden markov models, 47–56
Rauf U, Shehab M, Qamar N, Sameen S (2021) Formal approach to thwart against insider attacks: a bio-inspired auto-resilient policy regulation framework. Futur Gener Comput Syst 117:412–425. https://doi.org/10.1016/j.future.2020.11.009
Schlkopf B, Williamson RC, Smola A, Shawe-Taylor J, Platt J (1999) Support vector method for novelty detection. Adv Neural Inf Process Syst, 12
Soh C, Yu S, Narayanan A, Duraisamy S, Chen L (2019) Employee profiling via aspect-based sentiment and network for insider threats detection. Expert Syst Appl 135:351–361. https://doi.org/10.1016/j.eswa.2019.05.043
Song Y, Wen Z, Lin C-Y, Davis R (2013) One-class conditional random fields for sequential anomaly detection
Takabi H, Jafarian JH (2017) Insider threat mitigation using moving target defense and deception. In: Proceedings of the 2017 international workshop on managing insider security threats, pp 93–96. ACM, Dallas Texas USA. https://doi.org/10.1145/3139923.3139935. https://dl.acm.org/doi/10.1145/3139923.3139935 Accessed 16 Sept 2022
Tuor A, Kaplan S, Hutchinson B, Nichols N, Robinson S (2017) Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. WS-17
Villarreal-Vasquez M, Modelo-Howard G, Dube S, Bhargava B (2023) Hunting for insider threats using lstm-based anomaly detection. IEEE Trans Dependable Secure Comput 20(1):451–462. https://doi.org/10.1109/TDSC.2021.3135639
Wang S, Wang Z, Zhou T, Sun H, Yin X, Han D, Zhang H, Shi X, Yang J (2022) THREATRACE: detecting and tracing host-based threats in node level through provenance graph learning. IEEE Trans Inf Forensics Secur 17:3972–3987. https://doi.org/10.1109/TIFS.2022.3208815
Wu C, Li W (2021) Enhancing intrusion detection with feature selection and neural network. Int J Intell Syst 36(7):3087–3105. https://doi.org/10.1002/int.22397
Wu Y, Zhang Y (2022) An integrated framework for blockchain-enabled supply chain trust management towards smart manufacturing. Adv Eng Inform 51:101522. https://doi.org/10.1016/j.aei.2021.101522
Ye X, Hong S, Han M (2020) Feature engineering method using double-layer hidden markov model for insider threat detection. Int J Fuzzy Log Intel Syst 20(1):17–25. https://doi.org/10.5391/IJFIS.2020.20.1.17
Yuan S, Wu X (2020) Deep learning for insider threat detection: review, challenges and opportunities. arXiv:2005.12433. Accessed 17 Aug 2021
Yuan S, Zheng P, Wu X, Li Q (2019) Insider threat detection via hierarchical neural temporal point processes. In: 2019 IEEE international conference on big data (big data), pp 1343–1350. IEEE
Yuan S, Zheng P, Wu X, Tong H (2020) Few-shot insider threat detection, 2289–2292. https://doi.org/10.1145/3340531.3412161. Accessed 12 June 2021
Zhang H, Lu K, Zhou X, Yin Q, Wang P, Yue T (2021) SIoTFuzzer: fuzzing web interface in IoT firmware via stateful message generation. Appl Sci 11(7):3120. https://doi.org/10.3390/app11073120
Zhang D, Zheng Y, Wen Y, Xu Y, Wang J, Yu Y, Meng D (2018) Role-based log analysis applying deep learning for insider threat detection. In: Proceedings of the 1st workshop on security-oriented designs of computer architectures and processors, pp 18–20

Publisher's Note