applied sciences

Article

Multi-Granularity User Anomalous Behavior Detection

Wenying Feng [1,†], Yu Cao [2,†], Yilu Chen [2], Ye Wang [1,2], Ning Hu [1], Yan Jia [1,2] and Zhaoquan Gu [1,2,*]

[1] Department of New Networks, Pengcheng Laboratory, Shenzhen 518055, China; fengwy@pcl.ac.cn (W.F.); wangye2020@hit.edu.cn (Y.W.); hun@pcl.ac.cn (N.H.); jiay@pcl.ac.cn (Y.J.)
[2] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China; 22s151131@stu.hit.edu.cn (Y.C.); 23b951004@stu.hit.edu.cn (Y.C.)
* Correspondence: guzhaoquan@hit.edu.cn; Tel.: +86-138-1148-2140
† These authors contributed equally to this work.

Abstract: Insider threats pose significant risks to organizational security, often going undetected due to their familiarity with the systems. Detection of insider threats faces challenges of imbalanced data distributions and difficulties in fine-grained detection. Specifically, anomalous users and anomalous behaviors take up a very small fraction of all insider behavior data, making precise detection of anomalous users challenging. Moreover, not all behaviors of anomalous users are anomalous, so it is difficult to detect their behaviors by standardizing with single rules or models. To address these challenges, this paper presents a novel approach for insider threat detection, leveraging machine learning techniques to conduct multi-granularity anomaly detection. We introduce the Multi-Granularity User Anomalous Behavior Detection (MG-UABD) system, which combines coarse-grained and fine-grained anomaly detection to improve the accuracy and effectiveness of detecting anomalous behaviors. The coarse-grained module screens all of the user activities to identify potential anomalies, while the fine-grained module focuses on specific anomalous users to refine the detection process. Besides, MG-UABD employs a combination of oversampling and undersampling techniques to address the imbalance in the datasets, ensuring robust model performance. Through extensive experimentation on the commonly used dataset CERT R4.2, we demonstrate that the MG-UABD system achieves superior detection rate and precision. Compared to the suboptimal model, the accuracy has increased by 3.1% and the detection rate has increased by 4.1%. Our findings suggest that a multi-granularity approach for anomaly detection, combined with tailored sampling strategies, is highly effective in addressing insider threats.

Keywords: insider threat detection; UEBA; anomaly detection; random forest

check for updates

Academic Editor: Douglas O'Shaughnessy

Received: 6 November 2024
Revised: 19 December 2024
Accepted: 22 December 2024
Published: 27 December 2024

Citation: Feng, W.; Cao, Y.; Chen, Y.; Wang, Y.; Hu, N.; Jia, Y.; Gu, Z. Multi-Granularity User Anomalous Behavior Detection. *Appl. Sci.* **2025**, *15*, 128. https://doi.org/10.3390/app15010128

1. Introduction

Insider threats represent a multi-faceted and evolving danger that spans both public and private realms within every critical infrastructure sector. Acknowledging and delineating insider threats is foundational to the development of strategies aimed at mitigating insider risks. According to the Cybersecurity and Infrastructure Security Agency (CISA) of America, an insider threat is characterized by the potential for an individual with legitimate access to misuse their privileges, either deliberately or inadvertently, causing detriment to the objectives, assets, personnel, premises, data, apparatus, networks, or technological frameworks of an organization [1].

As enterprises expand, the number of internal users and user activities increases, leading to a variety of threat scenarios against enterprise organizations. Threats may include external attackers hijacking employees or stealing user account credentials to launch attacks, internal employees negligently or malevolently causing corporate information leaks, and security personnel being understaffed, being thus unable to guard against both internal and external threats. All these factors contribute to making the detection for insider threats increasingly challenging. According to the 2023 Insider Threat Report [2], which has been produced by Cybersecurity Insiders, 74% of organizations said they are at least moderately vulnerable or worse to insider threats. Moreover, 68% of respondents were concerned or very concerned about insider risk as their organizations return to the office or transition to hybrid work and 53% said detecting insider attacks is harder in the cloud.

To effectively detect threats posed by internal users within an enterprise, it is necessary to employ anomaly detection techniques for user behaviors, which is known as User and Entity Behavior Analytics (UEBA), proposed by Gartner [3]. UEBA mines potential incidents that deviate from a user's normal behavior or profile within the actions of internal enterprise users. Basic UEBA approaches include signature-based rules or pattern matching, while more advanced approaches encompass supervised and unsupervised machine learning techniques. UEBA assists organizations in identifying anomalous behaviors that do not align with typical user activity, preventing unauthorized access and abuse of privileges. It guards against data breaches caused by insiders, promptly detecting anomalous users, thus reducing insider threats, and lowering security risks.

Existing UEBA research mainly focuses on the identification and analysis of anomalous behavior [4–11], dedicated to classifying normal and anomalous behavior data of users to enhance the accuracy of behavioral detection. Most of this research concentrates on feature representation and classification of user behavior data, which can be categorized as coarse-grained anomaly detection. Coarse-grained anomaly detection struggles to uncover the correspondence between anomalous behaviors and anomalous users. Specifically, it cannot determine whether an anomalous action is initiated by a known malicious user, nor can it ascertain if all actions taken by an anomalous user are indeed anomalous. Moreover, there is no validation of whether the detected anomalies encompass all anomalous users. In practice, businesses are concerned not only with the precision of anomaly detection but also with the individuals initiating such behaviors. They seek detailed tracking and in-depth analysis of anomalous users. Therefore, it is necessary to conduct fine-grained detection and analysis for user entity behaviors. In the context of fine-grained anomaly detection for user behaviors, the following issues need to be addressed:

- First, targeting the fine-grained anomaly detection needs of enterprises, there is a requirement for dual-dimensional and bi-directional anomaly detection, for both anomalous behaviors and anomalous users. This necessitates indexing anomalous behaviors back to the anomalous users who initiated them, and detecting anomalous behaviors from the actions of identified anomalous users. It is essential to ensure that the detected anomalous behaviors encompass all anomalous users.
- Second, the issue of imbalanced positive and negative samples is particularly severe, in practical anomaly detection scenarios. This is due to the extremely low proportion of anomalous users and behaviors among the total volume of data, causing classification models to overlearning the majority class while underlearning the minority class, leading to misclassification of anomalous behaviors as normal behaviors.
- Finally, historical behavior data should be used as training data, and validation or testing needs to be conducted on future behavior data, to meet practical application requirements. Datasets should be divided according to temporal sequence in experiments, rather than randomly, as is the case in most studies. This chronological division ensures that the model is trained on past data and tested on unseen future data, better simulating real-world conditions.

To address the aforementioned challenges and issues, this paper proposes a multi-granularity anomaly detection model, MG-UABD, combining coarse-grained and fine-grained approaches. MG-UABD consists of two parts: a coarse-grained detection module (at the behavior level) and a fine-grained detection module (at the user level). The coarse-grained detection module, based on a Random Forest classification model, performs anomaly detection on the entirety of user behaviors. This module identifies anomalous behaviors and indexes the anomalous users. The purpose of this module is to discover as many anomalous users as possible through anomalous behaviors. The main focus at this stage is to improve the recall rate for anomalous users. The fine-grained detection module constructs individual behavior classification models for each anomalous user to perform precise anomaly detection. The purpose of this module is to learn the behavior patterns of each anomalous user, conducting accurate anomaly detection based on each user's behavior. The coarse-grained anomaly detection emphasizes the recall rate of anomaly detection, whereas the fine-grained anomaly detection focuses on the precision rate of anomaly detection. To tackle the problem of imbalanced positive and negative samples, MG-UABD employs a combination of oversampling and undersampling techniques to balance the positive and negative samples during training. Additionally, to support practical applications, MG-UABD trains and learns models based on historical behavior data, and validates and tests on future behavior data, to prevent inflated detection results. Our study conducts experiments on the commonly used version of the publicly available user anomaly detection dataset CERT. The experiments demonstrates that the proposed MG-UABD model can achieve coarse-grained and fine-grained anomaly detection of user behaviors. MG-UABD outperforms baseline models in metrics such as DTR (Detection True Rate) and FPR (False Positive Rate) under the same condition.

The main contributions of this research work are as follows:

- For the lack of fine-grained anomaly detection in UEBA, we propose a multi-granularity anomaly detection model combining coarse and fine granularity detection capability. This model employs Random Forest for coarse-grained anomaly detection at the behavioral level, and then conducts fine-grained anomaly detection at the user level for users flagged by the coarse-grained module.
- To address the issue of imbalanced class distribution in anomaly detection behavior data samples, undersampling and oversampling techniques are applied to process normal and anomalous samples, significantly improving model accuracy in detecting anomalous behavioral data.
- For the problem in previous studies where datasets were not divided according to chronological order, thus violating the characteristics of the actual scene, we restructure the behavior data division. Historical behavior data of users are used to construct the training set, while future behavior data serve as the test set. Testing models on this benchmark more closely aligns with the practical application needs of anomaly detection.

The subsequent sections of this paper are arranged as follows. Section 2 offers an overview of prior studies of UEBA. Section 3 introduces our proposed methodology, detailing the data preparation stages and the two modules of MG-UABD. In Section 4, we describe the dataset employed, the experimental setup, and provide a comprehensive analysis of the results. The paper culminates in Section 5 with concluding remarks and a discussion.

2. Pilot Experiment

This section demonstrates an initial set of experiments of to quantify the proportion of anomalous users and behaviors in the dataset. Within the scope of this research, individuals who instigate anomalous activities are designated as anomalous users.

Table 1 illustrates the macro and micro proportions of anomalous users and anomalous behaviors within the user activity dataset. On a macro level, the quantity of anomalous activities in CERT R4.2 constitutes merely 0.02% of the total volume of user behaviors, making them exceedingly challenging to detect through direct training of a classification model. Within this dataset, the number of anomalous users stands at 70, accounting for 7% of the entire user base. At a micro level, focusing on a randomly selected single anomalous user, the proportion of its anomalous behaviors is found to be 0.02%.

Table 1. Proportion of anomalous users and behaviors in CERT R4.2.

Item	Anomaly	Total	Proportion
Number of users	70	999	7.01%
Number of behaviors	7273	32,770,222	0.02%
Average behavior number per user	10	45,192	0.02%
Distribution of behavior number of a single anomalous user	5∼350	2115∼56,452	0.02∼6.77%

In addition, we calculated the number and proportion of truly anomalous behaviors of all anomalous users compared to the total number of their behaviors. As can be observed from the table, the proportion of anomalous behaviors among anomalous users ranges from 0.02% to 6.67%. Even for the anomalous user with the highest ratio of initiating anomalous behaviors, this proportion does not exceed 7%.

This reveals that both anomalous behaviors and anomalous users are exceedingly rare. Even for anomalous users, their actual anomalous behavior only accounts for a very small part of all their behavior. This results in highly imbalanced positive and negative samples, which imposes high demands on the precision of anomaly detection models.

Figure 1 depicts the distribution of the 70 anomalous users across different proportions of anomalous behavior. It is evident from the figures that 85.7% (60 out of 70) of anomalous users exhibit less than 2% anomalous behavior relative to their total behavioral data, 78.6% (55 out of 70) have less than 1% of their actions classified as anomalous, and 40% (28 out of 70) display less than 0.1% anomalous behavior. This indicates that the majority of anomalous users engage predominantly in normal activities, with only a minuscule fraction of their actions being categorized as anomalous.

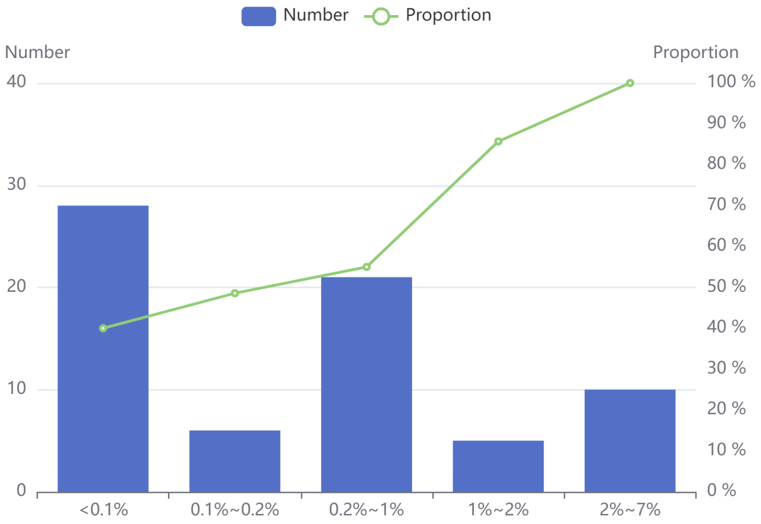


Figure 1. Distribution of anomalous behavior among anomalous users.

Following the pilot experiment, the following conclusions were reached:

- Both anomalous behaviors and anomalous users are significantly outnumbered by normal behaviors and normal users in terms of data volume. The imbalance between normal and anomalous sample sizes leads to considerable difficulty in detecting small-sample categories of anomalous behaviors.
- The overall ratio of anomalous behaviors initiated by anomalous users, compared to their total behavioral data, does not exceed 7%. Nearly 90% of anomalous users exhibit an anomalous behavior ratio of less than 2%. Conducting behavior detection for each anomalous user individually confronts similar issues of data distribution imbalance and detection challenges.

The aforementioned data distribution challenges in detecting user anomalous behaviors prompt us to propose a combined coarse-and-fine-grained anomaly detection model. Through staged detection, MG-UABD aims to screen the entirety of behavioral data while enhancing the detection efficacy for each individual anomalous user. Moreover, to address the scarcity of anomalous behavior samples, we apply random or SMOTE (Synthetic Minority Oversampling Technique) oversampling techniques (SMOTE is a popular method used to increase the size of minority classes in datasets where they are underrepresented). For the abundant normal behavior samples, undersampling is performed. By balancing the oversampling and undersampling, we equalize the ratio of anomalous to normal samples, enabling the model to comprehensively learn the characteristic patterns of both anomalous and normal behaviors. This approach mitigates overfitting to normal behavior categories and boosts the detection accuracy for anomalous behavior instances.

3. Related Work

Insider threat detection (ITD) uses a mitigation approach of detect and identify, assess, and manage to protect the organization. The foundation of ITD is the detection and identification of observable, concerning behaviors or activities. CISA, the cyber defense agency of the USA, has issued common guidelines to help mitigate insider threats in organizational environments [12]. Recent surveys [13–15] also provide comprehensive overviews for insider threat detection.

Most studies improve the accuracy and efficiency of anomaly detection by constructing and optimizing classification models [4], in the research of anomalous user behavior detection. We introduce research on UEBA from two aspects: coarse-grained and fine-grained user anomaly detection.

- Coarse-grained user anomaly detection. Early anomalous user detection mechanisms identified first-time system users as anomalous alarms. Tang et al. optimized the detection method for this problem by introducing different weights to different data indicators for processing, while alleviating the problem of data sparsity [16]. Another method that also uses weight processing is the anomaly detection method based on Mahalanobis distance judgment and singular value decomposition, as proposed by Shashanka et al., which considers sensitive anomaly detection requirements for anomaly alarms [17]. After machine learning and deep learning methods became mainstream, there have been many studies using supervised or unsupervised learning algorithms for detecting user anomalous behavior. Suresh et al. used fuzzy membership functions for feature aggregation and used the Random Weighted Majority Algorithm (RWMA) to transform traditional random forests into perceptron-like algorithms for detecting anomalous user behavior [18]. Singh et al. used Bi-LSTM for feature extraction and binary class support vector machine (SVM) for anomaly detection [19].

- Fine-grained user anomaly detection. In addition to coarse-grained classification of anomalous behavior, in recent years, user anomalous behavior detection has gradually shifted towards fine-grained analysis, detecting and analyzing anomalous behavior from multiple levels and perspectives. Le et al. proposed a user centered internal threat detection system that performs user behavior analysis at multiple data granularity levels under different training conditions [20]. Al-Mhiqani et al. proposed a multi-layer internal threat detection framework, which selects the most suitable model from multiple internal threat detection classification models based on multi-criteria decision-making techniques [21]. On the basis of detecting anomalous users and behaviors, Pantelidis et al. used Autoencoder and Variational Autoencoder for automatic defense against internal threats [22].

For fine-grained user anomalous behavior detection and analysis, existing research only focuses anomaly detection on behavior-level. However, constructing and training anomaly classification models based on only behavior data is not enough. Because the proportion of anomalous behavior is not consistent with the proportion of anomalous user behavior in practice. Existing research has not conducted in-depth research on the corresponding issues between anomalous users and anomalous behavior. In addition, existing research has not considered the temporal issue of behavioral data generation, and only divides the data into training and testing sets in proportion, which may lead to future time data entering the training set and causing testing leakage issues. Behavioral data usually have temporal correlations, and such experimental results do not have practical reference value.

To achieve accurate detection of fine-grained anomalous behavior for anomalous users, this study proposes a two-stage anomalous user behavior detection model, which is constructed from both behavior and user aspects. It not only detects anomalous behavior along time, but also detects the corresponding anomalous users. For each anomalous user, a specific anomalous detection model is constructed to achieve efficient and accurate detection of internal anomalous items at a fine-grained level.

4. Methodology

The goal of this study is to achieve accurate and efficient detection of internal anomalies through coarse-grained and fine-grained anomaly detection. Our method not only detects user behavior anomalies but also analyzes the behavior patterns of each anomalous user. This section mainly introduces our proposed detection method MG-UABD. Section 4.1 introduces the overall architecture of the model and the overall data processing flow. Section 4.2 introduces the types of user behavior data as well as the types of data preprocessing, such as data fusion, feature construction, and data partitioning. Section 4.3 introduces the coarse-grained anomaly detection module, which constructs models and detects anomalies based on all of the user behavior data. Section 4.4 introduces a fine-grained anomaly detection module, which constructs an individual behavioral anomaly detection model for each anomalous user.

4.1. Model Overview

Our proposed model MG-UABD for insider threat detection is illustrated in Figure 2. MG-UABD consists of three modules: data fusion and partition, coarse-grained anomaly detection, and fine-grained anomaly detection.

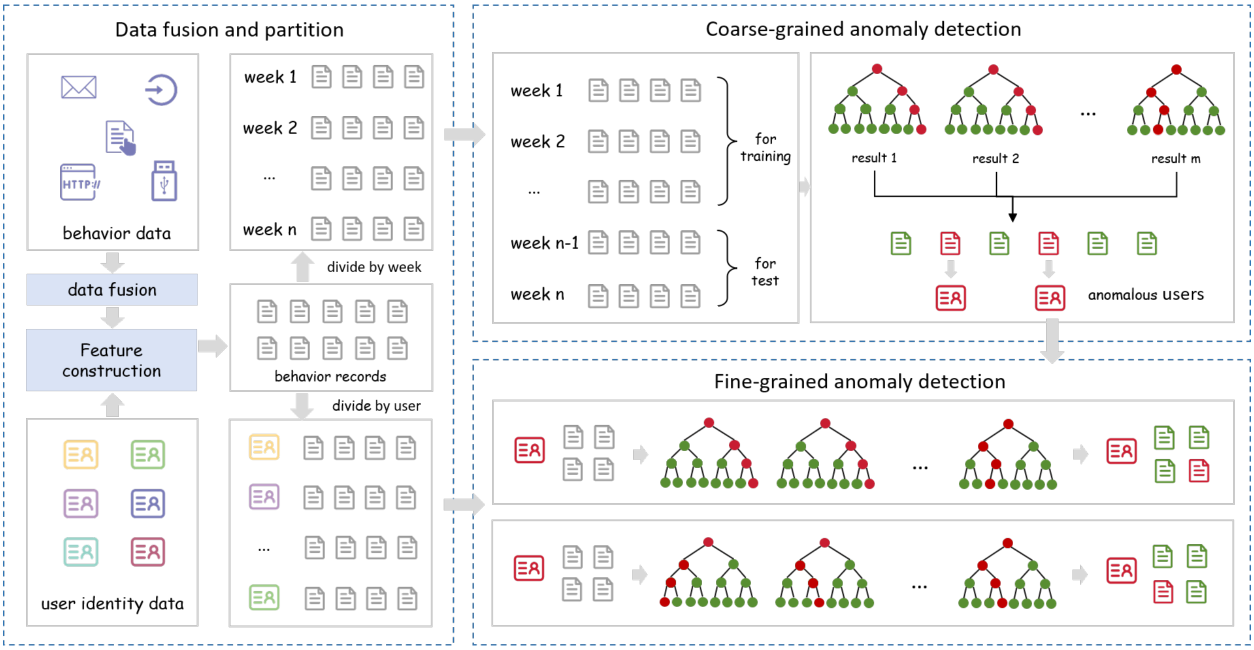


Figure 2. Overview of MG-UABD. In the data fusion and partition module, different user colors denote the user identities. In following modules, the anomalous users are marked in red, and the normal users are marked in green. The node color in each tree of the random forest model represent the decision process for the user.

Firstly, the data fusion and partition module integrates different types of user behavior data, including network traffic, login and logout, file operation, email transmission, and the use of USB devices. This module firstly designs features for the fused data and construct representation features for them. Then, this module split the behavior data after feature representation into training and testing sets based on the number of weeks, and performed coarse-grained anomaly detection, i.e., anomalous behavior. In addition, the behavior data of all users need to be divided according to the user to which the behavior belongs. Each user's behavior data need to be stored separately to support the fast reading of the fine-grained anomaly detection module.

Secondly, in the coarse-grained anomaly detection module, a random forest model is constructed based on the historical behavior data of all users. This module can detect anomalies in user behavior, and output classification labels for user behavior and its corresponding users. This module extracts all the behavior data of anomalous users and inputs them into the fine-grained anomaly detection module.

Finally, the fine-grained anomaly detection module extracts all the corresponding behavior data for each anomalous user, which are output by the coarse-grained anomaly detection module. This module establishes individual random forest classification model for each anomalous user. Each single model learns the historical behavior data of the anomalous user, and output the label of their behaviors. In the process of learning behavior for specific users, to address the problem of sparse anomalous behavior samples, we adopt random oversampling algorithms or SMOTE algorithms to oversample the training set for each anomalous user.

The reason for using the random forest algorithm is that in the random forest model, each decision tree is obtained based on random sampling, which can reduce the risk of oversampling and improve the generalization ability of the model. In addition, the random forest model can evaluate the importance of features and automatically identify features that have a significant impact on anomalous user behavior representation. After trying various machine learning algorithms, we found that random forest outperforms other classifications and achieves the best performance in user anomaly detection tasks. Therefore, in this study, random forest was adopted as the classification algorithm for anomaly detection.

4.2. Data Fusion and Preprocessing

The main work of this section is shown in the "Data fusion and partition" section of Figure 2. First, the module integrates user behavior data of different types. Then, it combines user identity information, constructing features for different types of behavioral data; Finally, the module divides the behavioral data according to two ways: by time and by user.

4.2.1. Behavior Data Fusion

UEBA conducts threat detection through various user behaviors. The user behavior analyzed in ITD typically includes the following types:

- Biometric behaviors: user biological behaviors that can be observed and recorded by instruments, such as mouse movement, keyboard input, etc.;
- Cyber activity behavior: cyber activity behaviors of users, such as login events, network packet, or network traffic;
- Psychosocial behaviors: user's psychological and emotional state, such as the power spectrum analysis of electroencephalogram (EEG) data, which can be used to identify insiders using brain wave features;
- Physical behaviors: such as door access, traffic sensor data, etc.;
- Other behaviors: other behavior features or combined behaviors from the above features.

Researchers from Carnegie Mellon University have constructed public datasets called CERT (computer emergency response team) for ITD and UEBA. The CERT dataset is an assembly of synthetic insider threat data compiled by CERT in collaboration with various partners (https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247, accessed on 6 November 2024). It is crafted through diverse scenarios incorporating traitorous actions and masquerading activities. This comprehensive dataset encompasses records of login activities, web browsing histories (including HTTP traffic), email communications, file access logs, device utilization metrics, LDAP (Lightweight Directory Access Protocol) data, as well as psychometric profiles of the users involved.

This study is based on CERT R4.2 (version 4.2 of the CERT dataset), released by Carnegie Mellon University [23]. The user behavior types of CERT R4.2 includes logs of login activities, HTTP or browsing histories, email exchanges, file access records, device utilization logs, LDAP data, and psychometric information [14].

The CERT R4.2 dataset includes two parts: user identity information and user behavior data. User identity information includes static information that is not frequently updated, such as user ID and entry time. User behavior data include user behavior extracted from various user activity logs. The five types, examples, and descriptions of user behavior data are listed in Table 2. Five different types of user behavior data come from different logs and are described with different attributes. Therefore, all activity logs need to be fused to store all user behavior information in a unified data format and order for easy use in subsequent modules.

4.2.2. Feature Construction

The five types of user behavior data are different and require the design and construction of different features. Feature construction can be divided into two types. (1) Constructing features separately based on behavior information, i.e., different features constructed according to different behaviors. For behaviors involving email sending, the feature is constructed according to information such as the number of recipients, email length, and whether it is an external email. (2) Constructing features combining user identity information and behavior information, such as extracting work time hazard rating features based on whether the occurrence time of user behavior is within normal working hours. We combine the features constructed in these two ways to obtain the complete behavior features. The features of each type of behavior are shown in Table 3.

Table 2. User behavior types, actions, and descriptions.

Activities	Action	Description
Logon/Logoff activity	Logon Logoff	User logged on a computer User logged off from a computer
File activity	Copy .exe file Copy .doc file Copy .pdf file Copy .txt file Copy .jpg file Copy .zip file	User copied an exe file to a removable media device User copied an doc file to a removable media device User copied an pdf file to a removable media device User copied an txt file to a removable media device User copied an jpg file to a removable media device User copied an zip file to a removable media device
HTTP activity	Neutral website Hacktivist website Cloudstorage website JobHunting website	User visited a neutral website User visited a hacktivist website User visited a cloudstorage website User visited a jobhunting website
Email activity	Internal email External email	All recipients are company email address There is an external address
Device activity	Connect Disconnect	User inserted a removable media device User removed a removable media device

Table 3. User behavior features.

Behavior Type	Features	Common Features
Logon/Logoff	Id, User, Date, Activity (logon or logoff)	Id, User, Date,
File	Id, User, Date, Filename, Content	
HTTP	Id, User, Date, PC, URL, Content	
Email	Id, User, Date, From, To, Cc, Bcc, Size, Attachment, Content	
Device	Id, User, Date, Activity (connect or disconnect)	

"Id", "User", and "Date" are the common features of all kinds of behavior data. "Id" denotes the unique identification code of the action record. "User" denotes the subject who make this action. "Date" denotes the time of the action record. For behavior data of type "Logon/Logoff", the unique feature is the activity description: "Logon" or "Logoff". For behavior data of type "File", the unique features include the "Filename" and "Content" of the file. For behavior data of type "HTTP", the unique features include "PC", the "URL" accessed by user, and the "Content" of the webpage. For behavior data of type "Email", unique features include the sender "From", the recipients "To", "Cc", and "Bcc". If the email contains attachments, the unique features includes the "Size", "Attachment", and "Content" of the attachment. For behavior data of type "Device", unique features is the activity description: "connect" or "disconnect".

After completing feature construction, it is necessary to perform numerical mapping on the features, represented by text. After converting feature values into numerical values, the original user identity information and user behavior data are transformed into a form that can be learned by the model for training.

4.2.3. Behavior Data Partition

As shown in the "Data fusion and partition" module in Figure 2, in order to detect coarse-grained and fine-grained behavior anomalies, the encoded user behavior data need to be divided into two ways: by time and by user.

Dividing data by time refers to the processing of data in the "extrapolation" testing mode. We extract time characteristics such as the date and week of the behavior based on the "Date" field of each behavior sample, and set a ratio for partitioning based on the overall time range of the dataset. For example, if the overall time range covers n weeks, we extract the behavior data from the first m weeks as the training set and the remaining (n-m) weeks as the testing set. The reason for processing the data in this way is that we have observed many studies using "interpolation" to randomly partition the data without considering the order in which behavioral actions occur. This will result in the model using behavior data from the future for training, while testing on behavior data from the past, which is inconsistent with the requirements of actual scenarios. Therefore, we optimize and use the "extrapolation" mode for model construction and testing, that is, training with historical data and testing with future data.

Dividing data by user refers to dividing user behavior data into subsets based on user identity. For example, if there is a user named Alexander, we extract all of their behavior records from the whole dataset. Furthermore, of course, the behavior data of each user also need to be divided according to the time range. The purpose of constructing a separate behavior dataset for each user is to consider the personalized behavioral characteristics of the user and perform fine-grained anomaly detection.

4.3. Coarse-Grained Anomaly Detection (CAD)

The purpose of the coarse-grained anomaly detection (CAD) module is to determine the presence of anomalous behavior based on the all behavior data of all users. This module trains and tests user behavior data at a weekly time granularity. Firstly, it calculates the number of weeks in which all user behavior starts and ends in the dataset, and determines the time range for inputting user behavior data into the model. This step needs to ensure that the range of weeks for the selected behavior data are earlier than the weeks of the earliest user who ended all behaviors, so as to ensure that no user in this stage of detection can be detected, because the behaviors end earlier and do not exist in the test set. At the same time, behavior data are selected from the first few weeks to form a training set of user historical data to train the model. Then, the training set data are processed by combining oversampling and undersampling, and the processed data are used as the training set and input into the random forest model for training. Finally, the remaining data are input in a weekly order as the test set to simulate detecting user behavior at a later time. When a behavior anomaly of a user is detected in a certain week, the fine-grained model is used to analyze and detect the anomalous behavior of this user in that week.

Coarse-grained anomaly detection takes the behavior data of all users in the first several weeks as input for the random forest model. We have tried various machine learning classification algorithms, and the random forest model has the best performance for anomaly detection. All user behavior data were divided into a training set $\mathcal{D}$ and a testing set $\mathcal{T}$ at weekly granularity as inputs to the random forest. A random forest is composed of multiple decision trees, each of which performs binary judgment on the current behavior and outputs the judgment result. The decision tree integrates the outputs of all decision trees as the final result of the current behavior judgment. If anomalous behavior is detected, the user corresponding to that behavior is input into the fine-grained anomaly detection module. The algorithm for the coarse-grained anomaly detection module based on random forest is shown in Algorithm 1, where algorithm input is a training set of the behavior data of all users.

Algorithm 1 Training and prediction of MG-UABD based on Random Forest.

```
procedure RANDOM FOREST($\mathcal{D}$, $T$, $m_{try}$) ▷ $\mathcal{D}$: Training set of the user behavior data, $T$: the number of tree, $m_{try}$: Maximum number of features
    $\mathcal{F} \leftarrow \emptyset$                                  ▷ Initialize forest as empty
    for $t = 1, \ldots, T$ do
        $\mathcal{D}_t \leftarrow$ Extract bootstrap sample from $\mathcal{D}$
        $T_t \leftarrow$ Building a decision tree($\mathcal{D}_t$, $m_{try}$)
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{T_t\}$                    ▷ Add trees to the forest
    end for
    return forest $\mathcal{F}$
end procedure
procedure PREDICT($\mathcal{F}$, test sample of user behavior data $x$)
    Initialize counter $C \leftarrow \emptyset$
    for each tree $T_t$ in $\mathcal{F}$ do
        $c_t \leftarrow T_t.predict(x)$                      ▷ Making Predictions for $x$
        Update Counter $C[c_t] \mathrel{+}= 1$
    end for
    Return the most frequently occurring category (normal or anomalous) in $C$ as the prediction result
end procedure
```

The coarse-grained anomaly detection module takes the behavior data of all users within a limited time range as input to a random forest, and outputs the binary classification judgment results for user behavior. Based on the judgment of normal and anomalous behavior data, it identifies the user who initiated anomalous behavior, i.e., the anomalous user, and inputs it into the subsequent fine-grained user behavior anomaly detection analysis.

4.4. Fine-Grained Anomaly Detection (FAD)

Fine-grained anomaly detection (FAD) is used to construct a random forest classification model for targeted user detection of coarse-grained detected users. This section constructs a random forest classification model for each anomalous user detected by the coarse-grained anomaly detection module. The processing flow for each anomalous user includes data oversampling, model construction, and user-level anomalous behavior detection.

First, for the anomalous users detected in the coarse-grained anomaly detection stage, all of their behavior data are extracted. Then, the training and testing sets of the user's behavior data are divided based on the time range. Data oversampling is conducted on the training set after partitioning. If there are no anomalous behaviors labeled in the user's behavior data, no model construction will be carried out. This is considering that the coarse-grained anomaly detection model may provide false positives for anomalous users. Because there is no anomalous behavior in the behavior data of normal users who are falsely detected as anomalous users, so there is no need to construct an anomaly detection model. Then, using the historical behavior data of each user as the training set, an anomaly detection and classification model is trained for the behavior pattern of each anomalous user. Finally, the remaining behavior data of the user are used as the test set to test the performance of the fine-grained anomaly detection model.

The fine-grained anomaly detection module algorithm based on random forest is similar to Algorithm 1. The only difference is that the input of the model is the behavior data training set of a certain user, and the output is the normal or anomalous state of that user's behavior. Building an independent user behavior model for each user helps the model learn the behavior patterns of each user, improve anomaly detection efficiency for specific users, and reduce false alarms of normal behavior.

5. Experiments

The MG-UABD model proposed in this study enables the detection of anomalous users and behaviors within an enterprise. We conducted experiments using the most widely used dataset CERT R4.2, and analyze the experimental results in detail.

Specifically, Section 5.1 details the experiment settings, including the introduce to the datasets, compared baseline models, and evaluation metrics. Section 5.2 shows the main experiments to demonstrate the anomaly detection effect of MG-AUBD. Section 5.3 explores the performance of MG-UABD on detecting anomalous behaviors and users under different training data ratios. Section 5.4 shows the ablation study of the model. Section 5.5 contains the correlation analysis of anomalous users and behaviors.

5.1. Experimental Settings

This section first introduces the datasets used in this study, and then introduces the baseline models for effect comparison. Finally, this section introduces the evaluation metrics used in the experiment.

5.1.1. Dataset Introduction

In this study, CMU-CERT [23], a publicly available and commonly used dataset in the field of user anomalous behavior detection, is used to test and validate the model proposed in this paper. CMU-CERT simulates and collects user behaviors within an enterprise organization, and the simulated enterprise generally contains 1000 to 4000 users. Several versions of CMU-CERT were released, including CERT R4.2, CERT R5.2, CERT R6.2, and CERT R6.2. We select the most commonly userd CERT R4.2 for our model evaluation. Table 4 show the statistics on the number of users and behaviors.

Table 4. Statistics of CERT R4.2.

Number of	Normal	Anomalous
users	930	70
behaviors	32,762,949	7273

The CERT R4.2 scenario models a hypothetical corporation consisting of 1000 staff members, among whom 70 individuals are simulated as malicious insiders, operating under the context of three distinct threat scenarios. The dataset contains several data files, including five behavioral record files, “logon.csv”, “device.csv”, “HTTP.csv”, “email.csv”, and “file.csv”, and the company’s employee organization structure LDAP folder. The behavior record file “logon.csv” records the behavior of users logging in and out of the host, the “file.csv” file records the behavior of employees copying files using USB flash drives, the “HTTP.csv” file records the behavior of users accessing the network, the “email.csv” file records the behavior of employees sending emails, and the “device.csv” file records the behavior of users connecting and disconnecting removable storage devices. The administrative records in the LDAP folder are used to record the active employee list for each month. If an employee leaves or is terminated from the activity, the user will no longer be included in the employee list for that month.

5.1.2. Compared Baselines

For CERT R4.2, we select the following baseline models:

- S-LSTM [6]: S-LSTM is based on the integration of sampling approach, which is the generation of synthetic samples to balance the two classes of learning by the SMOTE technique and LSTM algorithm for identify anomalous behavior.
- ITDBERT [10]: ITDBERT embeds temporal information into behavior and catches the fused semantic representation via pretrained language models. ITDBERT also leverages attention-based Bi-LSTM to provide behavior-level detection results.
- SPYRAPTOR [24]: A stream-based smart query system for real-time threat hunting within an enterprise. It is mainly composed of a graph construction subsystem, query synthesis subsystem, and query execution subsystem.
- SeqA-ITD [25]: This model focuses on user behavior sequences and proposes an Augmentation framework to boost the performance on Insider Threat Detection (SeqA-ITD). SeqA-ITD first embeds temporal information into user behavior sequences and then captures malicious user behavior's temporal and sequential patterns to generate discrete temporal sequences. A multi-granular enhanced Long Short-Term Memory (LSTM) model learns the original and generated temporal sequences with distinct temporal granularities to detect anomalous ones.
- RAP-Net [26]: A Resource Access Pattern Network (RAP-Net), which applies a reinforcement-learning-based Generative Adversarial Network, Word2Vec, Convolutional Neural Network, Recurrent Neural Network, and Attention Mechanism to insider threat detection. RAP-Net extracts user resource access pattern sequences from audit log files, and then performs data augmentation on the minority class sequences.
- LSTM-Autoencoder [27]: An LSTM-Autoencoder framework that has been proposed and tested on the CMU CERT r4.2 dataset and has shown to achieve high accuracy and low false alarm rates in detecting insider threats

5.1.3. Evaluation Metrics

The main experiment in this paper is a binary classification problem for user behavior samples. The trained model classifies the test samples into two categories: normal behavior (positive example data) and anomalous behavior (negative example data). We use the confusion matrix counting model to classify the normal behavior samples and anomalous behavior samples. The confusion matrix for the binary classification task contains the four data metrics True Positive (TP, correctly classified normal behavior samples), True Negative (TN, correctly classified anomalous behavior samples), False Negative (FN, misclassified anomalous behavior as normal behavior samples), and False Positive (FP, misclassified normal behavior as anomalous behavior sample). The five indicators commonly used in this study were calculated based on the four indicators in the confusion matrix as follows:

- Precision (*PRE*): Proportion of correctly predicted normal behavior as a percentage of actual positively predicted samples; the closer the value of this indicator is to 1, the more effective the model is:
$$PRE = \frac{TP}{TP + FP} \tag{1}$$
- Detection rate (*DTR*): also known as Recall; proportion of predicted correct normal behavior to total normal behavior—the closer the value of this indicator is to 1, the more effective the model is:
$$DTR = \frac{TP}{TP + FN} \tag{2}$$
- False positive rate (*FPR*): The probability that a sample of anomalous behavior will be misdiagnosed as a sample of normal behavior, which is a measure of the model's

ability to discriminate between negative case samples; the closer the value of this indicator is to 0, the more effective the model is:

$$FPR = \frac{FP}{FP + TN} \tag{3}$$

- F1-Score ($F1$): Harmonized mean of precision and recall, reflecting a combination of both; the closer the value of this indicator is to 1, the more effective the model is:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

- Accuracy (ACC): The proportion of correctly predicted behavioral samples to the total test set; the closer the value of this indicator is to 1, the more effective the model is:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

The coarse-grained detection module of MG-UABD is used to detect anomalous users, and the fine-grained detection module is used to detect anomalous behaviors of anomalous users. The results of the anomaly detection experiments shown in the following section combine the behavioral detection results of all users to get the overall results of the behavioral data test set and compare them with the baseline model.

5.2. Overall Results of Anomalous Behavior Detection (CAD)

MG-UEBA employs a combination of coarse-grained and fine-grained modules for the detection of anomalous user behavior. The code implementation for MG-UABD is available at https://github.com/cao-yuu/MG-UABD (accessed on 17 December 2024). Table 5 presents the detection outcomes of anomalous users and behaviors by the MG-UABD system. At the user level, out of a total of 1000 users, only 2 anomalous users were not detected and were instead classified as normal users. At the behavior level, from 6,554,045 behavior samples under examination, 141 anomalous behavior samples were misclassified as normal, thus remaining undetected, while 221 normal behavior samples were incorrectly categorized as anomalous. The majority of users and behavior samples were correctly identified by MG-UABD.

Table 5. Confusion matrix of MG-UABD on anomalous and normal user behavior detection.

Confusion Matrix	TP	TN	FP	FN
user level	930	68	2	0
behavior level	6,552,369	1314	141	221

The experimental results shown in Table 6 illustrate the performance differences between MG-UABD and other models in the task of anomaly detection at the behavior level. The baseline models listed in the table all conducted anomaly detection experiments on the CERT R4.2 dataset, and the results presented are directly sourced from their respective original publications. Unlike these models, our approach to coarse-grained anomaly detection did not involve random splitting of the training and testing sets. Instead, we used the first 25 weeks of user behavior data as the training set and the remaining data as the test set. This division method increases the difficulty of the anomaly detection task. However, as can be seen from the table, despite the more stringent testing conditions, MG-UABD outperforms all baseline models in all metrics except for FPR (False Positive Rate).

Table 6. Comparison results of anomaly detection on the behavior level. The bolded values are the optimal results, while the underlined values are the suboptimal results.

Model	**PRE (%)**	**DTR (%)**	**F1 (%)**	**ACC (%)**	**FPR (%)**
S-LSTM [6]	NA	96.00	95.45	99.00	2.00
ITDBERT [10]	93.00	91.87	92.43	NA	NA
SPYRAPTOR [24]	91.00	89.00	90.00	99.00	NA
SeqA-ITD [25]	95.79	95.64	95.63	NA	NA
RAP-Net [26]	95.71	96.01	95.86	NA	NA
LSTM-Autoencoder [27]	97.00	92.00	94.00	90.60	9.00
MG-UABD (ours)	**99.99**	**99.99**	**99.99**	**99.99**	9.69

Compared to the second-best model, MG-UABD achieved improvements of 3.1%, 4.1%, 4.3%, and 1.0% in PRE (Precision), DTR (Detection Rate), F1 (F1 Score), and ACC (Accuracy), respectively, calculated as follows:

$$PRE \uparrow = 3.1\% = \frac{(99.99 - 97.00)}{97.00}, \tag{6}$$

compared with LSTM-Autoencoder [27]:

$$DTR \uparrow = 4.1\% = \frac{(99.99 - 96.01)}{96.01}, \tag{7}$$

compared with RAP-Net [26]:

$$F1 \uparrow = 4.3\% = \frac{(99.99 - 95.86)}{95.86}, \tag{8}$$

compared with RAP-Net [26]:

$$ACC \uparrow = 1.0\% = \frac{(99.99 - 99.00)}{99.00}, \tag{9}$$

compared with SPYRAPTOR [24] and S-LSTM [6].

Due to the reorganization of the dataset according to chronological order, MG-UABD's FPR is slightly higher than that of other models, indicating that a small number of anomalous behavior samples were not detected.

This section of the experiment demonstrates that the MG-UABD, employing a two-stage approach of coarse-grained anomaly detection followed by fine-grained anomaly detection, significantly enhances the effectiveness of anomaly detection. It should be noted that the results shown in the table represent the models' detection of all samples of both normal and anomalous user behaviors. Since normal behavior samples constitute a vast majority, the models tend to predict this part of the sample accurately, hence the metrics are all above 90%. In the following sections, we will conduct a detailed analysis of the models' ability to detect anomalous users and anomalous behavior samples.

5.3. Anomaly Detection on Different Behavior Data Ratios

MG-UABD employs the coarse-grained anomaly detection module to screen all user behaviors for anomalies and the fine-grained anomaly detection module to examine each user's behavior individually. This section provides a detailed analysis of the behavior and user detection results, exploring the performance of MG-UABD's behavior-level and user-level anomaly detection under different training data conditions.

5.3.1. Results on Behavior Level (CAD)

We extracted the behavioral data of 70 anomalous users and a randomly selected subset of normal users from the CERT R4.2 dataset. We analyzed the impact of using different proportions of training data on the anomaly detection performance within the fine-grained anomaly detection module, for each individual user. The aggregated results across all users were considered as the overall detection outcome. In Figure 3, moving from left to right, the changes in the percentage of TN (True Negatives), TP (True Positives), and FN (False Negatives) samples in the anomaly detection results are depicted for training data proportions set at 20%, 40%, 60%, and 80% (Due to the model's accurate prediction of normal behavior (TP) occupying a significant proportion, which affects the visualization of other sample percentages and is not the focus of this analysis, TP is not shown). From the figure, it can be observed that as the proportion of the training set increases, the percentage of TNs notably rises, while the percentages of FP (False Positives) and FNs decrease significantly. This phenomenon suggests that with a larger training set, the model's capability to uncover anomalous samples improves, detecting more anomalies. Simultaneously, the percentages of misclassified anomalous and normal samples decrease. This observation indicates that a larger training dataset enhances the model's capability to perform anomaly detection.

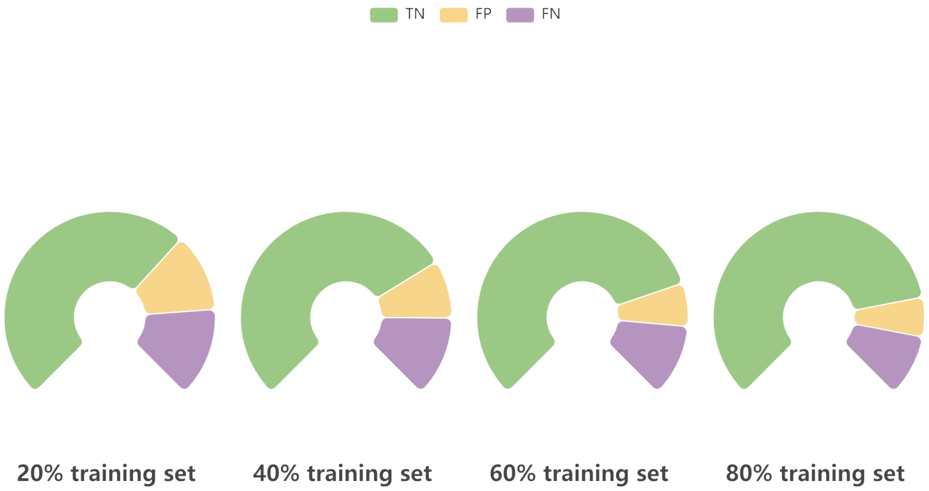
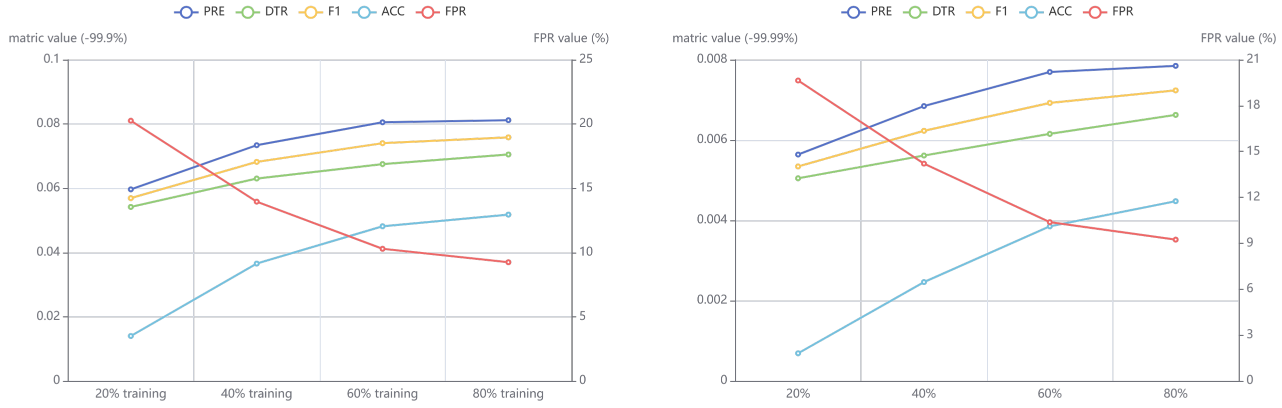


Figure 3. Fan chart of the sample distribution for user behavior detection.

We analyzed the changes in various anomaly detection metrics for both the fine-grained anomaly detection module and the MG-UABD system as a whole, as the proportion of the training set increased. As illustrated in Figure 4a,b, the performance across all metrics improved consistently as the training set proportion grew, for both the fine-grained anomaly detection module and the entire MG-UABD system. Specifically, Figure 4a shows that when the training set proportion was 20%, PRE, DTR, F1, and ACC were at their lowest levels. As the proportion of the training set increased, the marginal benefit of improvement in these metrics decreased. Among the four metrics, PRE, DTR, and F1 each increased by approximately 0.02%, while ACC, being a composite metric of PRE and DTR, increased by 0.04%. Additionally, the False Positive Rate (FPR) decreased from about 20% to around 10%. Combining the results of the coarse-grained anomaly behavior detection, we obtain the overall anomaly detection results for MG-UABD, as shown in Figure 4b. The trend of changes in the metrics as the proportion of training data varies is similar to that of the fine-grained module. However, due to the inclusion of the coarse-grained anomaly detection results, which serve to moderate the overall effect, the growth rates of the metrics are diluted. PRE, DTR, and F1 increase by 0.002%, 0.0015%, and 0.001% respectively, while ACC grows by 0.004%. The False Positive Rate (FPR) decreases from about 20% to approximately 9%.



(**a**) The performance change of FAD.

(**b**) The performance change of MG-UABD.

Figure 4. The performance changes of FAD and MG-UABD with increasing training set proportion.

The experimental results indicate that an increase in training data benefits the model's anomaly behavior detection performance, but the marginal benefit of improvement in the metrics gradually diminishes. On the CERT R4.2 dataset used in this study, the optimal balance between training volume and detection effectiveness is achieved with a training data ratio of approximately 60% to 80%.

5.3.2. Results on User Level (FAD)

In user-level anomaly detection, we focused specifically on the detection performance of MG-UABD for 70 anomalous users. Initially, we randomly selected four anomalous users and observed the impact of using different proportions of the training set on the detection performance for each user. As can be seen from Figure 5 and Table 7, the effect of the training set proportion varies among different users. For users TNM0961, CQW0652, and FSC0601, the detection performance generally improves as the training set increases, although there is a slight increase in the FPR for user CQW0652. User IJM0776 shows a pattern where the detection performance initially improves and then declines, reaching its best performance when the training data comprise 60% of the dataset. The unusual trends in the metric values may be due to overfitting by the model to these specific users' behavior patterns, where an increase in training data actually hinders the model's anomaly detection capabilities.

Table 7. Performance on four randomly selected users with the 80% training set.

User ID	PRE (%)	DTR (%)	F1 (%)	ACC (%)	FPR (%)
TNM0961	100	99.8962	99.9481	99.9011	0
CQW0652	99.6558	99.4656	99.5606	99.1327	28.1250
FSC0601	99.9911	99.9911	99.9911	99.9823	2.0833
IJM0776	99.3363	99.4830	99.4096	98.8555	20.4545

Generally, for most users, more training data lead to better anomaly detection performance; however, this rule does not apply uniformly across all users with certain behavior patterns. There was no clear correlation found between the variations in anomaly detection performance and the sample size for different users. This suggests that different users have distinct behavior patterns and varying distributions of normal and anomalous behavior samples, leading to the conclusion that the optimal setting for the amount of training data varies and needs to be customized based on the actual circumstances and requirements of each user.

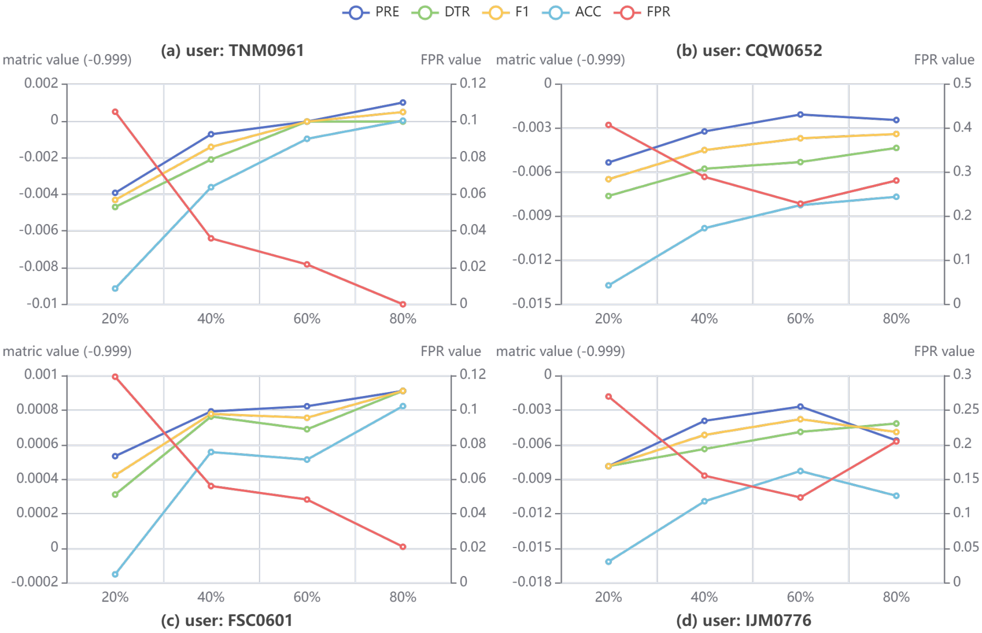


Figure 5. The anomaly detection performance of the model on randomly selected users varies with the proportion of the dataset.

Figures 6 and 7 represent the distribution of the 70 anomalous users in one-dimensional and two-dimensional metric spaces, with each point representing an anomalous user. Figure 6 displays the distribution of all 70 anomalous users along the PRE (Precision), DTR (Detection Rate), F1, and FPR (False Positive Rate) axes under training set proportions of 20%, 40%, 60%, and 80%. As the proportion of training data increases, the points representing anomalous users on the PRE, DTR, and F1 axes gradually cluster towards the right (indicating improvement), while the points on the FPR axis move towards the left (indicating reduction). However, not all user points are shifted towards the center of the clusters, and some still form long tails.

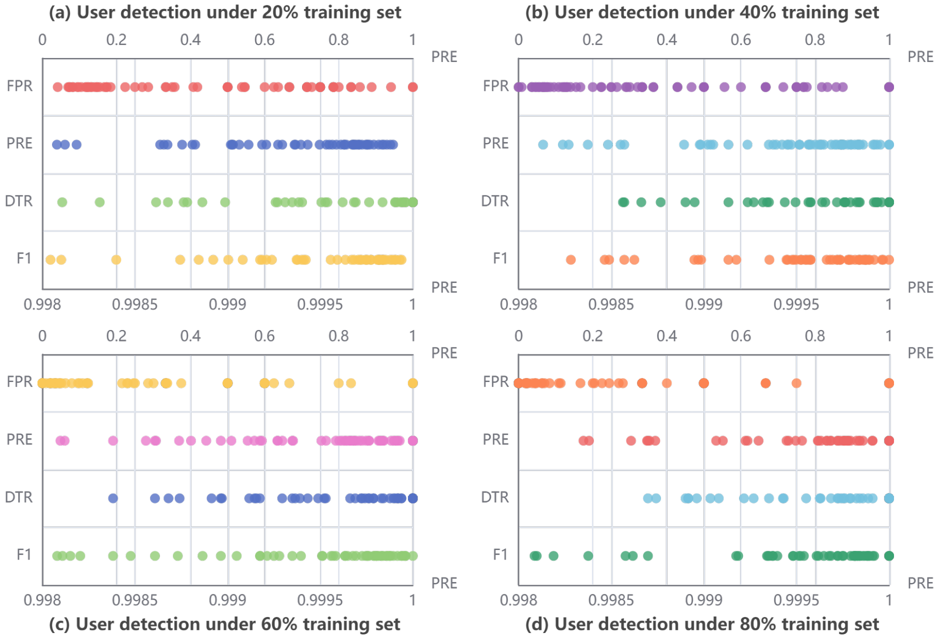


Figure 6. The one-dimensional spatial distribution of users in various indicators varies with the proportion of the dataset.

Figure 7 visualizes the distribution of anomalous user points in a two-dimensional space with PRE (Precision) on the x-axis and DTR (Detection Rate) on the y-axis. With 20% of the training data, most anomalous users are scattered throughout the space. As the proportion of training data increases, the anomalous users gradually shift towards areas with higher PRE and DTR. Under the 20% training set setting, no anomalous user samples fall on the vertical axis where PRE equals 1, but under the 80% setting, several anomalous users do fall on the "PRE = 1" axis. This phenomenon indicates that increasing the training set improves the overall detection performance for anomalous users.

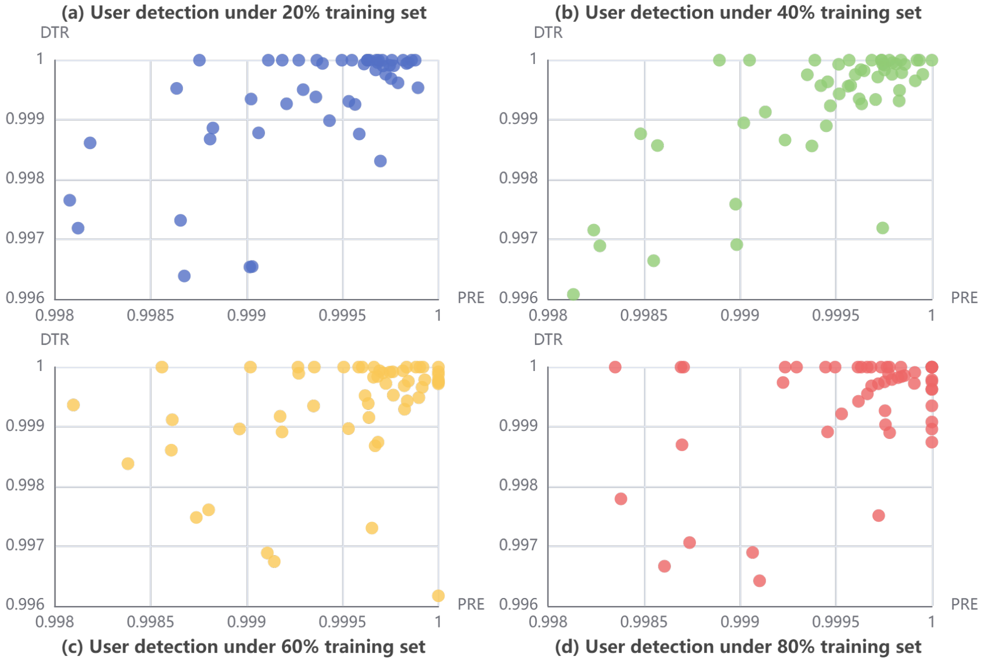


Figure 7. The two-dimensional spatial distribution of users in various indicators varies with the proportion of the dataset.

5.4. Ablation Study

The coarse-grained anomaly detection module of the MG-UABD trains on all behavioral data and identifies anomalous users based on detected anomalous behavior indices, which are then fed into the fine-grained anomaly detection module. This section explores the functions of each module and their contribution to enhancing the detection of anomalous users. Additionally, it compares the impact of using oversampling on the model's detection performance.

5.4.1. Ablation of CAD and FAD

This section of the experiment investigates the functions of the CAD and FAD modules. Figures 8 and 9 present the results of behavioral anomaly detection by each module under varying proportions of the training set. The CAD module screens all behavioral data for anomalies, while the FAD module further examines all actions associated with users flagged as anomalous by CAD. The MG-UABD represents the overall detection outcome, integrating the findings from both CAD and FAD. Given that the targets and volumes of data processed by CAD and FAD differ, the amounts of data they handle also vary with different training set configurations; specific details are provided in Table 8.

Table 8. Confusion matrix indicators.

Module and Model	TP	TN	FP	FN
CAD	20,962,409	161	7052	93
FAD (20%)	2,831,356	4500	1143	1298
FAD (40%)	2,329,977	3823	620	862
FAD (60%)	1,551,003	2634	302	504
FAD (80%)	749,084	1385	141	221
MG-UABD (20%)	26,209,061	4675	1143	1298
MG-UABD (40%)	19,656,907	3744	620	862
MG-UABD (60%)	13,104,676	2607	302	504
MG-UABD (80%)	6,552,369	1385	141	221

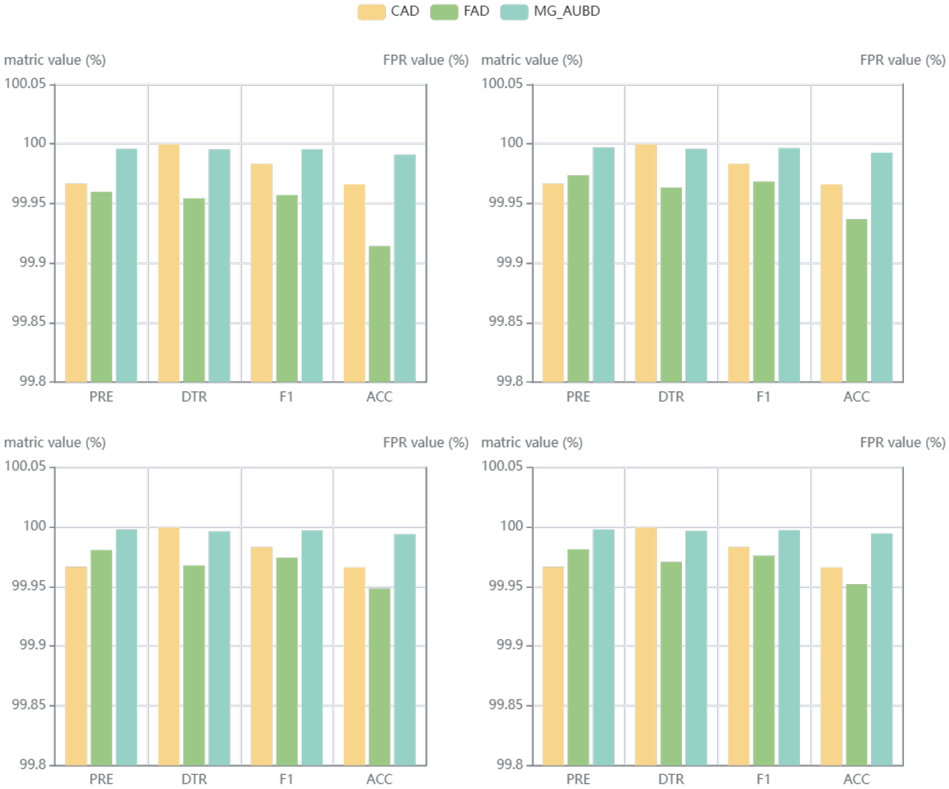


Figure 8. The performance of the coarse and fined granularity modules and the overall model is affected by the proportion of the training set (PRE, DTR, F1, and ACC).

From Figure 8, it can be observed that under varying proportions of the training data, the Precision (PRE) of the CAD module is not high, yet the Detection Rate (DTR) is relatively high. This indicates that the CAD module misclassifies some anomalous behavior samples as normal, but correctly classifies most normal behavior samples. In other words, the CAD module performs well in detecting normal samples. Additionally, as illustrated in Figure 9, the False Positive Rate (FPR) of the CAD module is notably high, exceeding 95%, which is consistent with the high DTR. This signifies that a substantial number of anomalous behavior samples are incorrectly classified as normal, and thus, not detected by the CAD module. The FPR represents missed detections of anomalous behaviors. Although the CAD module has a high recall rate for normal samples when they constitute a significant portion of the dataset, it also tends to classify anomalous samples as normal, leading to a very high false alarm rate for anomalous samples, and consequently, lower detection accuracy.

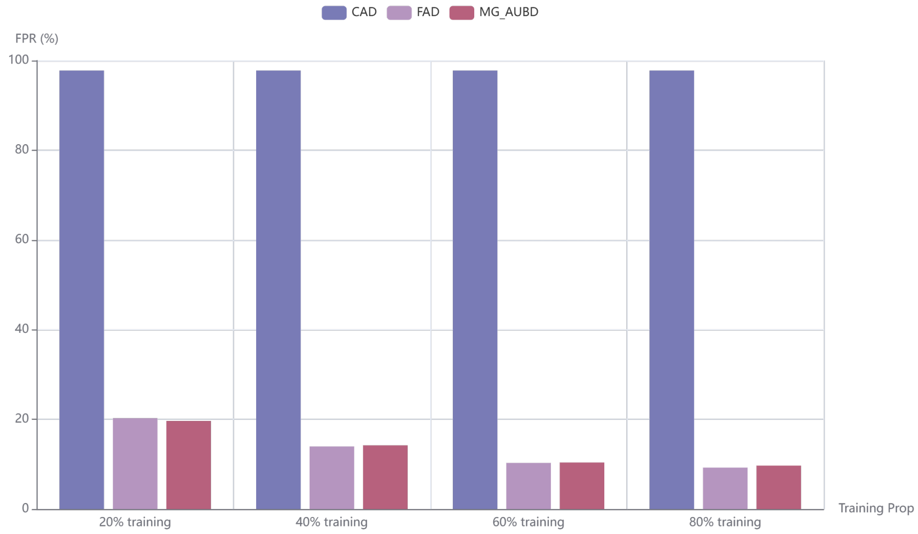


Figure 9. The performance of the coarse and fine granularity modules and the overall model is affected by the proportion of the training set (FPR).

The FAD module, although exhibiting a lower Detection Rate (DTR), shows an improvement in Precision (PRE) compared to the CAD module. This suggests that the FAD module performs better than the CAD module in detecting anomalous behavior samples, mitigating the misclassification of anomalous samples. Furthermore, the False Positive Rate (FPR) of the FAD module is reduced to below 20% compared to the CAD module, indicating a reduction in the misclassification of anomalous samples. The MG-UABD integrates the detection outcomes of both the CAD and FAD modules, combining the strength of the CAD module in detecting normal behavior samples with the strength of the FAD module in detecting anomalous behavior samples. This integration allows for the retention of effective detection of normal behavior samples, achieving a high DTR, while also improving the ability to accurately identify anomalous behavior samples, achieving a higher PRE.

It is important to note that the CAD module detects all user behavior samples, which include a larger proportion of normal users and normal behavior samples. Consequently, it develops a stronger capability to detect normal samples. Since in this study "positive" refers to normal samples, the CAD module's Detection Rate (DTR) even surpasses that of the FAD module. On the other hand, the FAD module operates on the basis of the CAD module's detection results, performing focused detection on each anomalous user individually. Although the FAD module does not outperform the CAD module in terms of the DTR metric, its detection targets contain fewer normal behavior samples that could interfere with the detection process. As a result, the FAD module exhibits superior detection capabilities for anomalous samples, leading to a higher detection accuracy.

5.4.2. Ablation of Oversampling

The MG-UABD system employs a combination of oversampling and undersampling techniques to address the imbalance between positive and negative samples. Specifically, for anomalous users with a relatively large number of anomalous behavior samples, the SMOTE algorithm is used for oversampling. However, since the SMOTE method requires a sufficient number of baseline anomalous samples, for anomalous users with very few anomalous behavior samples that do not meet the requirements for SMOTE, random selection and duplication of anomalous behaviors are employed to oversample the negative samples. To preserve the original data distribution, the oversampling ratio should not be too high, and the number of oversampled instances should not increase excessively. Correspondingly, undersampling is applied to the normal samples, which involves randomly removing positive samples to achieve balance between positive and negative samples in the training set.

To specifically validate the effectiveness of oversampling for anomaly detection tasks, we implemented comparative experiments with Random Forest, XGBoost, and MG-UABD models, applying oversampling, and visualized the changes in the False Positive Rate (FPR) in Figure 10. From the figure, it can be observed that after applying oversampling, the FPR values for all three models decreased, indicating a reduction in the misclassification of anomalous behavior samples in the test set. This demonstrates that data oversampling can address the issue of insufficient anomalous behavior samples during anomaly detection, thereby enhancing the detection performance of the models.

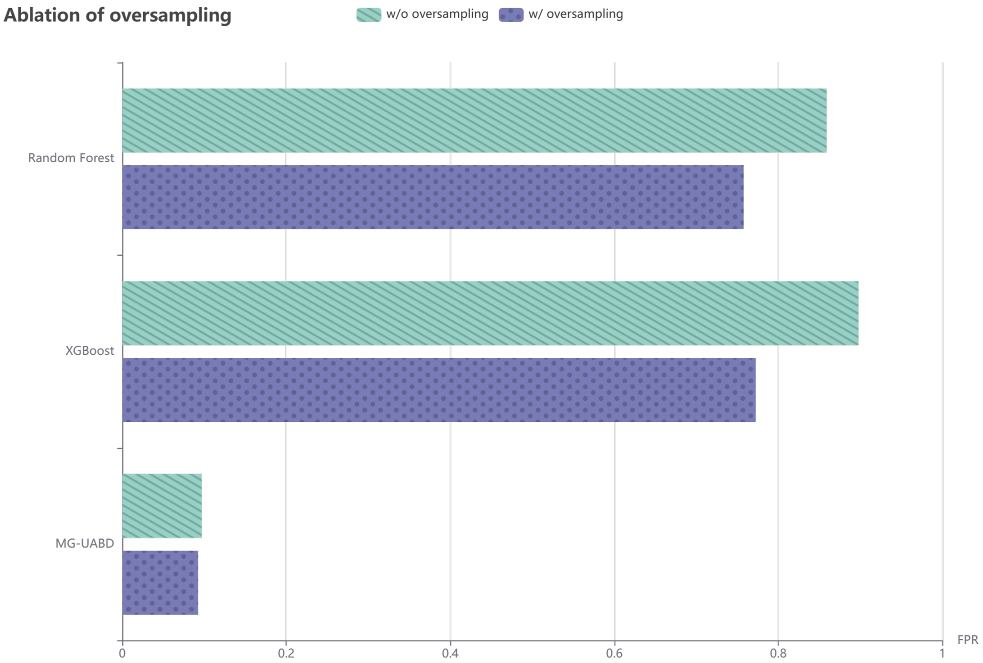


Figure 10. The impact of oversampling on model performance.

5.5. Correlational Analysis of Users and Behaviors

Users' anomalous behaviors encompass various types, and this part of the experiment investigates the distribution of anomalous user behaviors across these types during the fine-grained anomaly detection phase, aiming to understand the patterns among different types of behaviors. For this part, the coarse-grained anomaly detection module is trained separately with seven different data types as input, and users identified through the analysis of different behavioral data are sent to the corresponding fine-grained anomaly detection module for that behavior type.

Figure 11 illustrates the correspondence between normal and anomalous users across seven behavior types. For visualization purposes, we have selected only a portion of the normal users from the entire dataset, focusing our analysis on the distribution of behavior types for anomalous users. As shown in the figure, the behavior composition of normal users primarily consists of "logon", "logoff", and "http", whereas the behavior composition of anomalous users is mostly concentrated in "connect", "disconnect", and "email". The "http" type has the largest volume of data, with both normal and anomalous users engaging in this type of behavior.

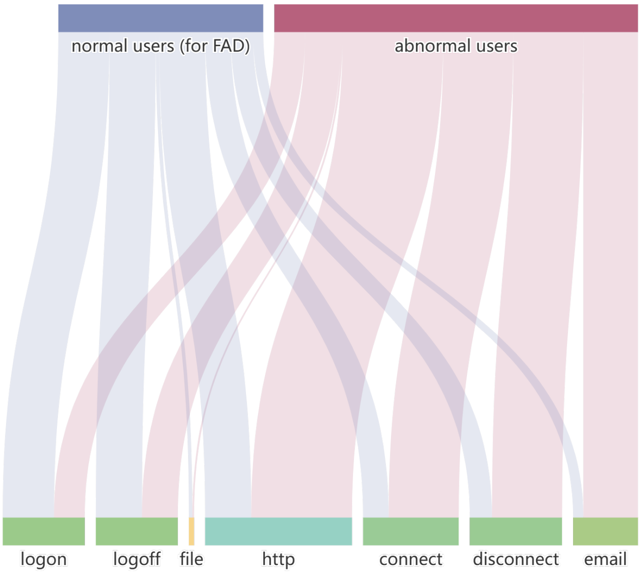


Figure 11. The correspondence between user types and behavior types.

We investigated the detection of anomalous users using different types of anomaly detection data. Figure 12 shows the number of anomalous users detected by the fine-grained anomaly detection module based on various behavior types. From the figure, it can be seen that the behavior types that detect the highest number of anomalous users are: "http", "connect", and "disconnect". Among these, "normal user" refers to the correction of anomalous users identified by the coarse-grained detection module; these users were initially flagged as anomalous by CAD but were subsequently determined to be normal after being processed by FAD. "Logon" and "logoff" have the highest number of misclassified anomalous users that were corrected, with relatively fewer anomalous users detected. Regardless of whether users are normal or anomalous, the "file" type of behavior has the least amount of data, and correspondingly, the number of anomalous users detected based on this type is also the lowest.

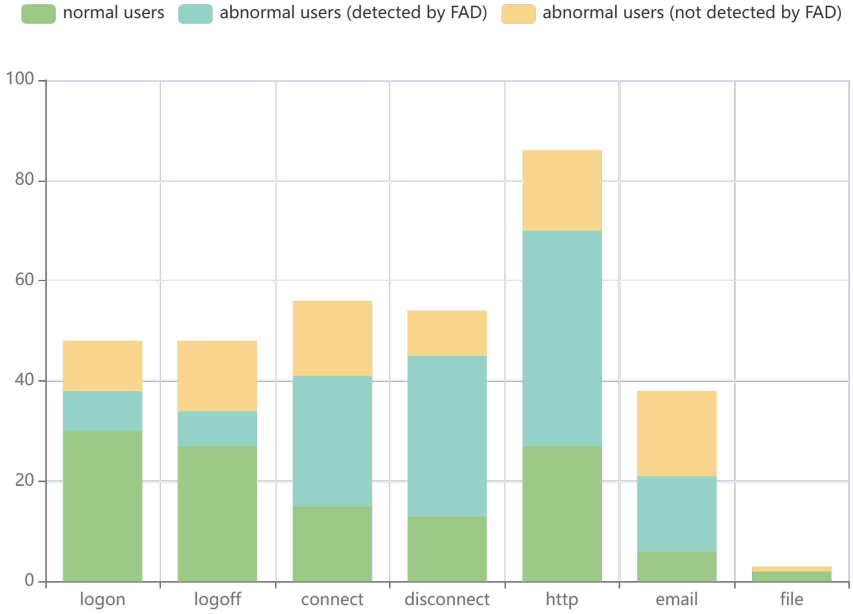


Figure 12. Detection correction of CAD by FAD.

Figure 13 displays the effectiveness of the MG-UABD fine-grained anomaly detection module in detecting anomalous behaviors across different behavior types. From the figure, it can be observed that the best-performing behavior types for anomalous behavior detection are “http” and “file”, followed by “logon” and “logoff”. The poorest performance is seen in “connect”, ”disconnect”, and “email”. Combining the results from Figure 12, we can infer that the behavioral characteristics of anomalous users are concentrated in “connect” and “disconnect”, but once individual anomalous users are subjected to anomalous behavior detection, the distinction between normal and anomalous features for these types is not clear, leading to a higher rate of missed detections. The “http” type has the largest volume of data, and the precision, detection rate, and low false-negative rate for behavior detection based on this type are all high. While the “file” type of behavior data are less prevalent, the detection results are also favorable.

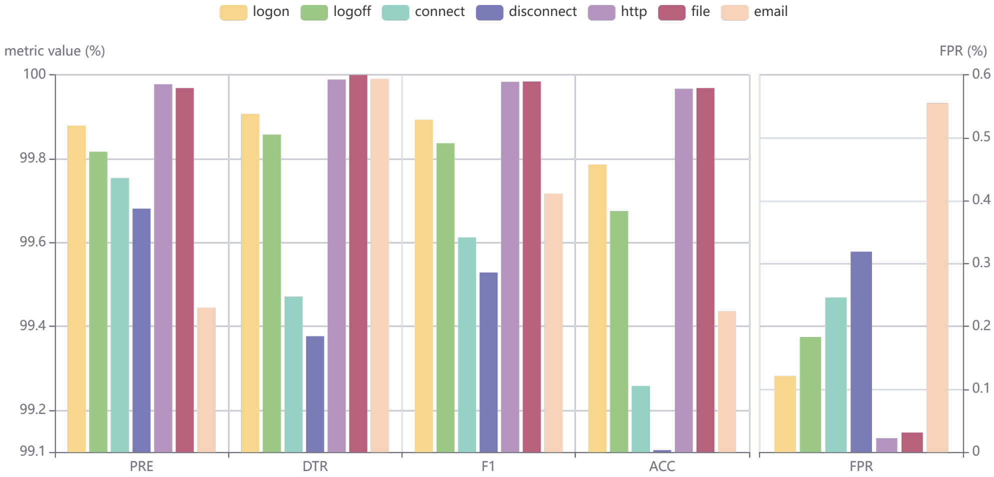


Figure 13. Detection effectiveness for each type of behavior.

In summary, the quantity and characteristics of anomalous users’ behaviors are more concentrated on device operations (“connect” and “disconnect”) and internet usage (“http”). However, the effectiveness of detecting anomalous behaviors based on device operation behaviors is not as good as that based on internet usage and file behaviors. The detection performance based on a single type of behavior is inferior to that achieved by integrating multiple types of behaviors for comprehensive detection. Therefore, in practical anomaly detection for users, it is recommended to focus not only on internet usage and file behaviors but also to incorporate various types of behaviors for anomaly detection.

6. Discussion

This section discusses and analyzes the challenges that MG-UABD may face in practical applications, including the complexity, real-time detection capability, generalization, and scalability of the model, as well as considerations of data privacy issues.

6.1. Complexity Analysis

Assuming that the initial number of training samples for MG-UABD is D, the maximum number of features among all types of behaviors is M, the number of decision trees in the random forest algorithm is set to T, and the depth of the decision trees is d, the time complexity of training a coarse-grained classification module is $O(D \times logD \times M \times T)$. If anomalous users which are detected in the coarse-grained anomaly detection stage is n, the maximum number of behavior samples corresponding to these users is P, and the time complexity of training the fine-grained detection module is $O(P \times logP \times M \times T)$. The time complexity of both coarse-grained and fine-grained modules during the detection phase is $O(d)$. As for the spatial complexity, if the number of nodes in each decision tree in the random forest is v, the spatial complexity of the coarse-grained detection module is $O(v \times T)$, and the spatial complexity of the fine-grained detection stage is $O(n \times vs. \times T)$.

It is obvious that MG-UABD builds detection models for each user in the fine-grained detection stage, which increases the computational time and space resources. The time complexity is directly proportional to the maximum number of anomalous user behavior samples detected in the coarse-grained detection stage, and the space complexity is directly proportional to the number of anomalous users. Therefore, it should be selected as needed in practical applications. Fortunately, the two modules of this model can be decoupled, and companies can choose which module to use based on their employee size and cost calculation. In addition, further research can explore the quantitative evaluation of the degree of user anomaly in the coarse-grained detection stage, and design a fine-grained detection model based on the degree of user anomaly, which is more targeted and cost-effective.

6.2. Real-Time Detection Analysis

Real-time detection of internal threats is an important factor in determining whether MG-UABD can be deployed for applications. There are three main factors that affect the efficiency of real-time detection: data push and basic processing speed, model incremental training speed, and representation and detection time of the test samples. Among them, data push and basic processing speed depend on the processing capabilities of deployed devices, and using high concurrency processing for streaming data can improve efficiency. Incremental training of the model refers to regularly tagging newly collected samples to maintain the model's capability to recognize new threats and alleviate the problem of concept drift. This part is a trade-off between training complexity and detection efficiency, and a suitable compromise needs to be found. After training the basic model, the time complexity of detecting each subsequent behavior sample in our method is $O(d)$, where d is the depth of the decision tree. Due to the much lower complexity of random forests compared to complex models such as neural networks, better real-time detection efficiency can be achieved while meeting the requirements for parallel data processing.

Apart from detection efficiency, real-time detection accuracy will vary depending on the domain and the model's generalization ability. After running the algorithm for a period of time, concept drift may occur, leading to a decrease in real-time detection accuracy. It is necessary to consider using methods such as continuous learning to incrementally train the model and regularly update its parameters. Enterprises with a single business, few personnel, and homogeneous behavior patterns require lower model construction costs to meet their needs, while enterprises with complex personnel and diverse behavior types require higher model construction costs.

6.3. Generalization Discussion

MG-UABD relies on the classification ability of random forest to detect anomalous behavior and users, and builds and applies the model based on historical user behavior data within the protected enterprise. MG-UABD relies on labeled data within the protected domain for training. If the user behavior data provided by the protected enterprise are not significantly different from the publicly available dataset used in this study, smaller adaptation and testing may be required for application. If the difference is significant, the model needs to be reconstructed based on the specific types and characteristics of behavioral data in order to obtain better results. In addition, adaptive domain transfer methods [28] or continuous learning [29] can be used for model adaptation in new data scenarios.

6.4. Scalability Discussion

The scalability of the model mainly depends on the computing power and data processing capability of hardware devices, as the random forest algorithm has good scalability and can handle large-scale data. When the dataset is large, some techniques can be used to accelerate the computation process, such as using random subsets to train each decision tree or using parallel computing techniques.

6.5. Data Privacy Discussion

MG-UABD is aimed at anomaly analysis of user entity behavior, which requires the collection of user behavior data within the enterprise. The more comprehensive the behavior data, the higher the credibility of the detection results. The implementation of MG-UABD is divided into two stages. The coarse-grained detection stage is behavior oriented and does not involve user identity. It can only collect behavior data without mapping to user identity. After detecting behavior anomalies in the coarse-grained stage, the user identity will be indexed and further detected. At this stage, privacy can be protected by setting a virtual user ID and erasing the correspondence between abnormal results and real users. This mapping may not be disclosed to the implementers and analysts of the scheme, but only retained internally by the enterprise to ensure privacy. Meanwhile, the deployment location of the algorithm should be designated by the protected enterprise and strict access control should be implemented to ensure privacy.

7. Conclusions and Future Plans

This article proposes a multi-granularity combined anomaly detection algorithm MG-UABD for internal threat detection. The model combines coarse-grained behavior detection with fine-grained user detection to improve the accuracy of detecting anomalous behaviors and users. The coarse-grained detection module detects all behavior data and identifies anomalous users through anomalous behavior. The fine-grained detection module constructs a separate detection model for each anomalous user and learns the specific behavior patterns of each user, and thus, improves the detection effect on the behavior of specific anomalous users. This study validated the effectiveness of the model through a large number of experiments, including comparing the accuracy of detecting anomalous behavior, conducting ablation experiments on various modules, and analyzing the correlation between anomalous users and anomalous behavior. The experimental results and analysis demonstrate that MG-UABD can effectively address challenges in detecting anomalous users and behaviors, and improve the accuracy and robustness of internal threat detection.

The future plan of this work includes the following two aspects: firstly, quantitatively evaluating the degree of abnormality of anomalous users or behaviors, such as adding a quantitative abnormality evaluation mechanism in the coarse-grained detection module. By setting an abnormality threshold value, the number of anomalous users sent to the fine-grained detection module can be determined to save the hardware resource cost of model deployment. Secondly, exploring the addition of association rules in statistical methods for detection. For example, if a user copies certain files after sending and receiving emails containing specific types, there may be a necessary association within the behavior sequence. Adding these rules to the system will help increase the capability to detect potential insider threats.

Author Contributions: Conceptualization, W.F.; methodology, Y.C. (Yu Cao); software, Y.C. (Yu Cao); validation, Y.W. and N.H.; formal analysis, Z.G.; investigation, Y.J.; resources, Y.C. (Yilu Chen); data curation, Y.C. (Yilu Chen); writing—original draft preparation, W.F.; writing—review and editing, Z.G.; visualization, W.F.; supervision, Y.J.; project administration, Z.G.; funding acquisition, Y.J., N.H. and Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Major Key Project of Pengcheng Laboratory (Grant No. PCL2024A05), the Shenzhen Science and Technology Program (No. KJZD20231023094701003), the opening project of science and technology on communication networks laboratory (No. FFX24641X007), and the National Natural Science Foundation of China (Grant No. 62372137).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in CERT of CMU at https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247 (accessed on 6 November 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MG-UABD	Multi-Granularity User Anomaly Behavior Detection
UEBA	User and Entity Behavior Analytics
CERT	Computer Emergency Response Team
CISA	Cybersecurity and Infrastructure Security Agency
ITD	Insider threat detection
SVM	Support vector machine
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
DTR	Detection True Rate
FPR	False Positive Rate
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
CAD	Coarse-grained anomaly detection
FAD	Fine-grained anomaly detection

References

1. CISA. *Defining Insider Threats*; CISA: Washington, DC, USA, 2020.
2. Cybersecurity Insiders. *2023 REPORT Insider Threat*; Cybersecurity Insiders: Baltimore, MD, USA, 2023.
3. Safety Internal Reference. UEBA Practical Implementation: Data Quality Determines Everything. 2021. Available online: https://www.secrss.com/articles/31990 (accessed on 18 June 2021).
4. AlSlaiman, M.; Salman, M.I.; Saleh, M.M.; Wang, B. Enhancing False Negative and Positive Rates for Efficient Insider Threat Detection. *Comput. Secur.* **2023**, *126*, 103066. [CrossRef]
5. Aldairi, M.; Karimi, L.; Joshi, J. A Trust Aware Unsupervised Learning Approach for Insider Threat Detection. In Proceedings of the 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science, IRI, Los Angeles, CA, USA, 30 July–1 August 2019; pp. 89–98.
6. Besnaci, S.; Hafidi, M.; Lamia, M. Dealing with Extremly Unbalanced Data and Detecting Insider Threats with Deep Neural Networks. In Proceedings of the 2023 International Conference on Advances in Electronics, Control and Communication Systems, ICAECCS, Blida, Algeria, 6–7 March 2023; pp. 1–6.

7. Ge, D.; Zhong, S.; Chen, K. Multi-Source Data Fusion for Insider Threat Detection Using Residual Networks. In Proceedings of the 2022 3rd International Conference on Electronics, Communications and Information Technology, CECIT, Sanya, China, 23–25 December 2022; pp. 359–366.
8. Hall, A.J.; Pitropakis, N.; Buchanan, W.J.; Moradpoor, N. Predicting Malicious Insider Threat Scenarios Using Organizational Data and a Heterogeneous Stack-Classifier. In Proceedings of the 2018 IEEE International Conference Big Data (Big Data), Seattle, WT, USA, 10–13 December 2018; pp. 5034–5039.
9. He, W.; Wu, X.; Wu, J.; Xie, X.; Qiu, L.; Sun, L. Insider Threat Detection Based on User Historical Behavior and Attention Mechanism. In Proceedings of the 2021 IEEE International Conference on Data Science in Cyberspace, Cyberspace DSC, Shenzhen, China, 9–11 October 2021; pp. 564–569.
10. Huang, W.; Zhu, H.; Li, C.; Lv, Q.; Wang, Y.; Yang, H. ITDBERT: Temporal-semantic Representation for Insider Threat Detection. In Proceedings of the 2021 IEEE Symposium on Computers and Communications, ISCC, Athens, Greece, 5–8 September 2021; pp. 1–7.
11. Igbe, O.; Saadawi, T. Insider Threat Detection Using an Artificial Immune System Algorithm. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON, New York, NY, USA, 8–10 November 2018; pp. 297–302.
12. CISA. *Detecting and Identifying Insider Threats*; CISA: Washington, DC, USA, 2020.
13. Yuan, S.; Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Comput. Secur.* **2021**, *104*, 102221. [CrossRef]
14. Al-Mhiqani, M.N.; Ahmad, R.; Zainal Abidin, 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.* **2020**, *10*, 5208. [CrossRef]
15. Raut, M.; Dhavale, S.; Singh, A.; Mehra, A. Insider Threat Detection using Deep Learning: A Review. In Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 3–5 December 2020; pp. 856–863. [CrossRef]
16. Tang, B.; Hu, Q.; Lin, D. Reducing False Positives of User-to-Entity First-Access Alerts for User Behavior Analytics. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops, Workshop ICDMW, New Orleans, LA, USA, 18–21 November 2017; pp. 804–811.
17. Shashanka, M.; Shen, M.Y.; Wang, J. User and Entity Behavior Analytics for Enterprise Security. In Proceedings of the 2016 IEEE International Conference on Big Data, Washington, DC, USA, 5–8 December 2016; pp. 1867–1874.
18. Varsha Suresh, P.; Lalitha Madhavu, M. Insider Attack: Internal Cyber Attack Detection Using Machine Learning. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT, Kharagpur, India, 6–8 July 2021; pp. 1–7.
19. Singh, M.; Mehtre, B.M.; Sangeetha, S. User Behaviour Based Insider Threat Detection in Critical Infrastructures. In Proceedings of the 2021 2nd International, Conference on Secure Cyber, Computing and Communications, ICSCCC, Jalandhar, India, 21–23 May 2021; pp. 489–494.
20. Le, D.C.; Zincir-Heywood, N.; Heywood, M.I. Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 30–44. [CrossRef]
21. Al-Mhiqani, M.N.; Ahmad, R.; Abidin, Z.Z.; Abdulkareem, K.H.; Mohammed, M.A.; Gupta, D.; Shankar, K. A New Intelligent Multilayer Framework for Insider Threat Detection. *Comput. Electr. Eng.* **2022**, *97*, 107597. [CrossRef]
22. Pantelidis, E.; Bendiab, G.; Shiaeles, S.; Kolokotronis, N. Insider Threat Detection Using Deep Autoencoder and Variational Autoencoder Neural Networks. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience, CSR, Virtual, 26–28 July 2021; pp. 129–134.
23. Glasser, J.; Lindauer, B. Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data. In Proceedings of the 2013 IEEE Security and Privacy Workshops, San Francisco, CA, USA, 23–24 May 2013; pp. 98–104. [CrossRef]
24. Zhu, D.; Sun, H.; Li, N.; Mi, B.; Huang, X. SPYRAPTOR: A Stream-based Smart Query System for Real-Time Threat Hunting within Enterprise. In Proceedings of the 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Rio de Janeiro, Brazil, 24–26 May 2023; pp. 1055–1062. [CrossRef]
25. Zhang, F.; Ma, X.; Huang, W. SeqA-ITD: User Behavior Sequence Augmentation for Insider Threat Detection at Multiple Time Granularities. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–7. [CrossRef]
26. Zhu, D.; Huang, X.; Li, N.; Sun, H.; Liu, M.; Liu, J. RAP-Net: A Resource Access Pattern Network for Insider Threat Detection. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8. [CrossRef]
27. Nasir, R.; Afzal, M.; Latif, R.; Iqbal, W. Behavioral Based Insider Threat Detection Using Deep Learning. *IEEE Access* **2021**, *9*, 143266–143274. [CrossRef]

28. Li, P.; Wang, Y.; Li, Q.; Liu, Z.; Xu, K.; Ren, J.; Liu, Z.; Lin, R. Learning from Limited Heterogeneous Training Data: Meta-Learning for Unsupervised Zero-Day Web Attack Detection across Web Domains. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Melbourne, Australia, 10–14 July 2023.
29. Chen, Y.; Ding, Z.; Wagner, D. Continuous Learning for Android Malware Detection. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), Anaheim, CA, USA, 9–11 August 2023; pp. 1127–1144.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.