scientific reports

Check for updates

OPEN

MITD-Net: Markov image-based threat detection network

Malek Algabri[1,2,6], Firdaus Alhrazi[1,6], Cavazos Quero Luis[3], Ahmed Abdu[4], Yeong Hyeon Gu[5]✉ & Redhwan Algabri[3,6]✉

The increasing sophistication of malicious activities within applications emphasizes the need for advanced predictive technologies. Malicious user behavior (MUB) is a concern in organizations, as it is a significant source of security breaches caused by employees within the organization. Although previous studies in user activity detection have demonstrated some success, these technologies have been insufficient in identifying new or unfamiliar security threats. To improve the detection of insider threats, this study introduces MITD-Net, a novel method based on a MobileNet convolutional neural network (CNN) architecture to predict the MUB effectively and efficiently. MITD-Net is faster and accurate than its counterparts, leveraging the computational efficiency and adaptability of deep neural networks in low-resource environments. Our model addresses the challenge of predicting harmful behavior. MITD-Net contributes to the proactive identification and mitigation of potential threats, thereby enhancing overall system security. The proposed method aims to extract features from the CERT r4.2 dataset, converting them into a Markov image to detect the MUB from authorized parties. Experimental evaluations conducted on CERT r4.2 datasets demonstrate the effectiveness of the proposed model. Moreover, this paper compares the results of previous studies. The experimental findings show that the proposed approach outperforms or achieves state-of-the-art techniques. Ablation studies were also performed to evaluate the significance of each individual component of the model.

In today's digital landscape, securing computer systems against malicious user behavior (MUB) has become a critical concern[1–3]. Insider threats, such as data manipulation and system misuse, originate from within the organization, typically involving employees who abuse their authority. These threats pose significant risks to both data integrity and system security. Unlike external attacks, insider threats are difficult to predict and detect, as they are carried out by authorized users with legitimate access. Given the evolving nature and diversity of insider threats, traditional security mechanisms may be insufficient to ensure data and system integrity. Consequently, researchers have increasingly explored unconventional methods to secure systems and defend against insider threats.

Insider threats pose a formidable challenge in cybersecurity, leveraging sophisticated techniques to evade detection and exploiting privileged access for malicious activities[4–7]. Insider threats also represent a significant challenge for cybersecurity due to their ability to bypass traditional security measures and exploit access to sensitive information. Recent studies have increasingly focused on advanced detection systems that leverage machine learning (ML), deep learning (DL), and behavioral analysis to enable more accurate and proactive identification of potential threats[5,8–11]. Meng et al.[12] proposed advanced detection techniques based on ML models, particularly long short-term memory (LSTM) networks, to detect subtle variations in user behavior that may indicate insider threats. Similarly, Lin et al.[13] highlighted the effectiveness of combining multiple behavioral domain features with deep belief networks (DBNs) to improve detection accuracy. In addition, Zhang et al.[14] investigated the use of adaptive optimization within DBNs to enhance the detection of covert malicious activities. Collectively, these studies underscore the growing reliance on DL frameworks to tackle the complex challenges of insider threat detection (ITD).

Recent advances in ML have improved the ability to distinguish insider threats from normal user behavior, thereby enhancing the accuracy and effectiveness of detection systems. Advanced analytical techniques, such

[1]Department of Computer Science, Faculty of Computer and Information Technology, Sana'a University, P.O. Box 33039 Sana'a , Yemen. [2]Department of Cybersecurity, College of Engineering and Information Technology, Emirates International University, 16881 Sana'a, Yemen. [3]Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea. [4]School of Information, Xi'an University of Finance and Economics, Xi'an 710100, China. [5]Department of Artificial Intelligence and Data Science, College of Software and Convergence Technology, Sejong University, Seoul 05006, Republic of Korea. [6]Malek Algabri, Firdaus Alhrazi and Redhwan Algabri have contributed equally to this work. ✉email: yhgu@sejong.ac.kr; redhwan@sejong.ac.kr

as ensemble learning and user behavior analytics, have become essential components of modern cybersecurity architectures, providing powerful tools for the early detection and prevention of insider-related threats[15–17]. CNNs have been widely successful in numerous applications, such as robot[18], defect prediction[19], pose estimation[20], image recognition, natural language processing, and defect detection, due to their ability to automatically and adaptively learn spatial hierarchies of features. The resulting CNN-MobileNet model is designed to improve both accuracy and efficiency for insider threat classification (ITC). The research seeks to validate the effectiveness of an image-based transfer learning approach, specifically leveraging MobileNet-CNN, in securing computer systems against MUB. The results are based upon the CERT r4.2 Dataset. The labeled images represent both normal and malignant behavior of users. A MobileNet-CNN trained on this dataset to build a reliable and efficient system to detect and classify potentially malicious activities from a visual perspective of user interactions. Furthermore, the study assesses the ITC performance of the image-based MobileNet-CNN model for its relevance in improving the system security. This study proposed a proactive and effective method to mitigate the security threats from malicious users. The key contribution of this work is its capability to enhance the detection and classification of MUB using the strength of MobileNet-CNN and visual information analysis. Our proposed approach has the capability to capture complicated patterns as well as accurate visual signals to security threats, which is a bit more proactive in threat mitigation, providing system administrators with more prompt and effective alerts to potential attacks.

The key contributions of this study can be summarized as follows:

1. A novel approach has been proposed called MITD-Net to convert text data into Markov images, making it easier to implement using CNN technologies. MITD-Net is fast and accurate, enhances system security by ensuring rapid and effective detection in the classification of user behavior on systems
2. A Markov image-based DL technique was applied in the system security domain by leveraging visual cues derived from user interactions.
3. MobileNet technology was applied to leverage its lightweight architecture and speed in image classification, providing a fast approach for predicting threats that can be used on mobile devices.
4. New features were extracted from the dataset, such as work hours, visit-to-job-search site count, and presence file.exe malware, in order to obtain better accuracy for ITD.
5. A comprehensive comparison with state-of-the-art methods was conducted[21–39], along with the experimental results obtained using the proposed method.

The rest of this work is organized as follows. The section "Related work" reviews related works, the section "Image-based feature representation" describes the proposed MITD-Net framework, the section "Results and discussion" discusses the results obtained, and finally, the section "Conclusions" is the conclusions.

Related work

In this section, several studies on ITD and prediction are reviewed, focusing on the challenges posed by malicious activities carried out by authorized personnel within organizations[40]. The following paragraphs highlight significant contributions to this field. Insider threats present a substantial challenge due to the authorized access these individuals have to sensitive information and systems. These threats are distinct as they behave maliciously, which makes them hard to observe and forecast, since they emerge within one's organization. Knowledge of user behavior and assessment of that behavior is essential in order to prevent risks before they become security incidents. Recent developments in ML, particularly DL, offer promising techniques to discern and anticipate such behaviors. However, the complexity of insider activities necessitates the development of more sophisticated and computationally efficient predictive models.

In[41], the authors focused on classifying untrusted users in a cloud environment employing ML techniques. Four different classical ML classifiers, namely Decision Tree (DT), Logistic Regression, Random Forest Classifier, and K-nearest Neighbor (KNN), along with a DL model. The DL model outperformed the other models, achieving an accuracy of 88% and an AUC value of 0.856. In[42], the authors proposed a technique for detecting insider threats that combines stacked LSTM and stacked GRU attention models, demonstrating strong robustness in identifying insider threats. The multi-edge weight relational graph neural network (MEWRGNN) is a novel method introduced in[43] to enhance the anomaly detection of insider threats. Accurate anomaly detection relies on the ability of this method to capture the changing contextual connections of user activities across time. The MEWRGNN approach also provides interpretability by value assignment of the different edge-representation properties. It also proved that it could learn from less training data, ensuring the adoption of quick and accurate insider threat analysis. It also provided plain language findings for security researchers, which helped examine the discovered risks.

The authors in[22] employ a multi-step procedure for data preprocessing, followed by bi-directional LSTM (bi-LSTM) networks and binary support vector machines (SVM) for ITD from malicious users within the organization. Comparing this method with existing methods demonstrates significant performance improvements. Its main advantages are that it is simple, flexible, and can identify malicious users. The CMU-CERT dataset's imbalance problem is the focus of future research. In addition, DL methods have been shown to enhance the performance of intrusion detection systems. DL models that can detect and classify various forms of cyberattacks include recurrent neural networks (RNNs) and CNNs. As these models support the complexity and unpredictability of network traffic data, it results in fine and accurate intrusion detection. Combining the DL solutions with the conventional intrusion detection solutions can considerably improve the performance of the system[44].

One of the approaches proposed by Kim et al. for identifying insider threats is to combine it with other ones[45] focuses on anomaly detection algorithms and user behavior modeling. Reading user logs, the researchers

generated three kinds of sets, and four kinds of algorithms for anomaly detection were used to detect abnormal behavior. Results of their experiment further show the efficiency of the proposed method for discovering insider threats, especially using an unbalanced dataset with few domain expert opinions. Le & Zincir-Heywood [46] applied an ML-based ITD system from a user-centric point of view to the various studies. Their method analyzes system performance in terms of each data instance, regular insider, and malicious insider at different data granular levels. The authors in[47] introduced the ADSAGE method that focuses on irregularities in audit log events that are represented as graph edges in the context of fine-grained ITD. ADSAGE supports text, category, and numeric data, as well as edge sequences and characteristics. Wang & El Saddik[28] presented a DL-based framework to combined a Digital Twins and self-attentionfor predicting insider threats. The approach corrects the imbalanced data and enhances the detection accuracy, and makes use of advanced data augmentation techniques, such as BERT and GPT-2. Similarly, Hu et al.[48] proposed a method based on a DL model for ITD. This model dynamically captures the biobehavioural signature of the mouse over time to verify the user's identity. By processing and classifying these dynamics with a DL framework, the method achieves lower false acceptance and rejection rates and better detection performance. Furthermore, Yang et al.[49] developed an ITD system that combines scenario-driven alarm filters along with baseline anomaly detection. Based on multi-domain behavioral mode characteristics and an abnormal frequency anomaly degree based on time, the proposed method effectively distinguishes normal and malicious anomalies and provides high-accuracy threat detection in arbitrary situations.

A User and Entity Behavior Analytics (UEBA) model for anomaly detection in an enterprise security employing data science techniques is proposed in [50]. This model leverages user profiles as a major focus to identify disruptions in activities. Liu et al.[10] used an ensemble of deep autoencoders to construct an ITD system that is able to discern non-normal activities of audited user activities, including file operations, USB device usage, and logon/logoff activities. The approach demonstrates excellent accuracy in identifying hostile insider activity and dramatically decreases false positives. Raut et al. [51] examined how to use DL to identify insider threats and provided a thorough analysis of DL model comparisons using the CERT dataset. Saaudi et al.[52] proposed a CNN-LSTM model for ITD, using character embeddings to model user behavior with high precision and recall, demonstrating its effectiveness on the CERT dataset.

Proposed MITD-Net framework

This section provides a proposed MITD-Net framework and objectives for this study.

Figure 1 shows an MITD-Net framework of the proposed method. MITD-Net framework consists of six main stages: (1) dataset CERT r4.2, (2) data preprocessing, (3) image-based feature representation, (4) split the dataset, (5) MobileNet-CNN Model, and (6) classification. Each stage is explained in the next subsection. The following are stages of the proposed MITD-Net framework.

Dataset CERT r4.2

The purpose of the CERT r4.2 dataset[53] is to help with research on ITD. The Carnegie-Mellon University Software Engineering Institute's CERT Insider Threat Center created multiple datasets, and one of them is CERT r4.2. This dataset is an intriguing collection for the simulation of a user activity environment, which allows detecting insider threats in the organization while doing it. It is a dataset with a complex structure distributed between several .csv files in such a way that each of the files captures a separate dimension of user behavior that is important to consider when analyzing and modeling insider threats. The CERT r4.2 dataset comprises multiple files, each containing processed data that captures specific aspects of user activity within a controlled environment. These aspects include email communications, file operations, logon sessions, device usage, and other behavioral indicators[54].

The dataset comprises detailed logs that represent typical activities within an organization, which include the following in Table 1: 1) Logon.csv: This file records user logon and logoff activities, including the user ID, the computer used for access, logon type (e.g., interactive or remote), and timestamps. The columns include id, date, user, pc, and activity (logon or logoff). Key features derived from this data include the frequency and timing of user logins. This file captures patterns in user authentication behavior. Anomalies such as frequent logins during non-working hours or irregular access patterns may indicate suspicious or unauthorized activity. 2) Device.csv: This file tracks the connection and disconnection of USB devices, which may be used for unauthorized data transfer. It logs the user, the device, the time of connection or disconnection, and the computer where the device was used. The columns include *id, date, user, pc, and activity* (connect or disconnect). This file records instances of USB device activity. Unusual patterns, such as USB usage outside regular working hours or on unauthorized machines, may suggest attempts at physical data exfiltration or policy violations. 3) File.csv: This file logs file-related operations, such as opening, copying, deleting, or creating files. Each entry includes details about the user, file path, type of operation, and timestamp. The columns include *id, date, user, pc, filename, and activity* (e.g., read, write, copy, delete). The key feature is the number of executable files downloaded. This file monitors user interactions with files, particularly the download of executable files. A high number of such downloads may signal an attempt to install unauthorized software or execute potentially malicious code. 4) Email.csv: Captures email activity, documenting each instance an email is sent or received. Information includes sender, recipients, timestamp, and sometimes a summary of the email content. The columns include *id, date, user, pc, to, cc, bcc, from, and activity* (send or receive), and sometimes *size, attachments*. The key feature is the number of emails sent. Measures the number of emails a user sends; a spike in email activity could suggest attempts to exfiltrate data. 5) HTTP.csv: Contains logs of HTTP requests made by users, which are essential for understanding web browsing behavior. The entries include the user, URL, timestamp, and the computer used for browsing. The columns include *id, date, user, pc, and url*. The key feature is the count of website visits. Tracks the number of times a user visits different websites, which can indicate either normal browsing behavior or visits to suspicious sites.

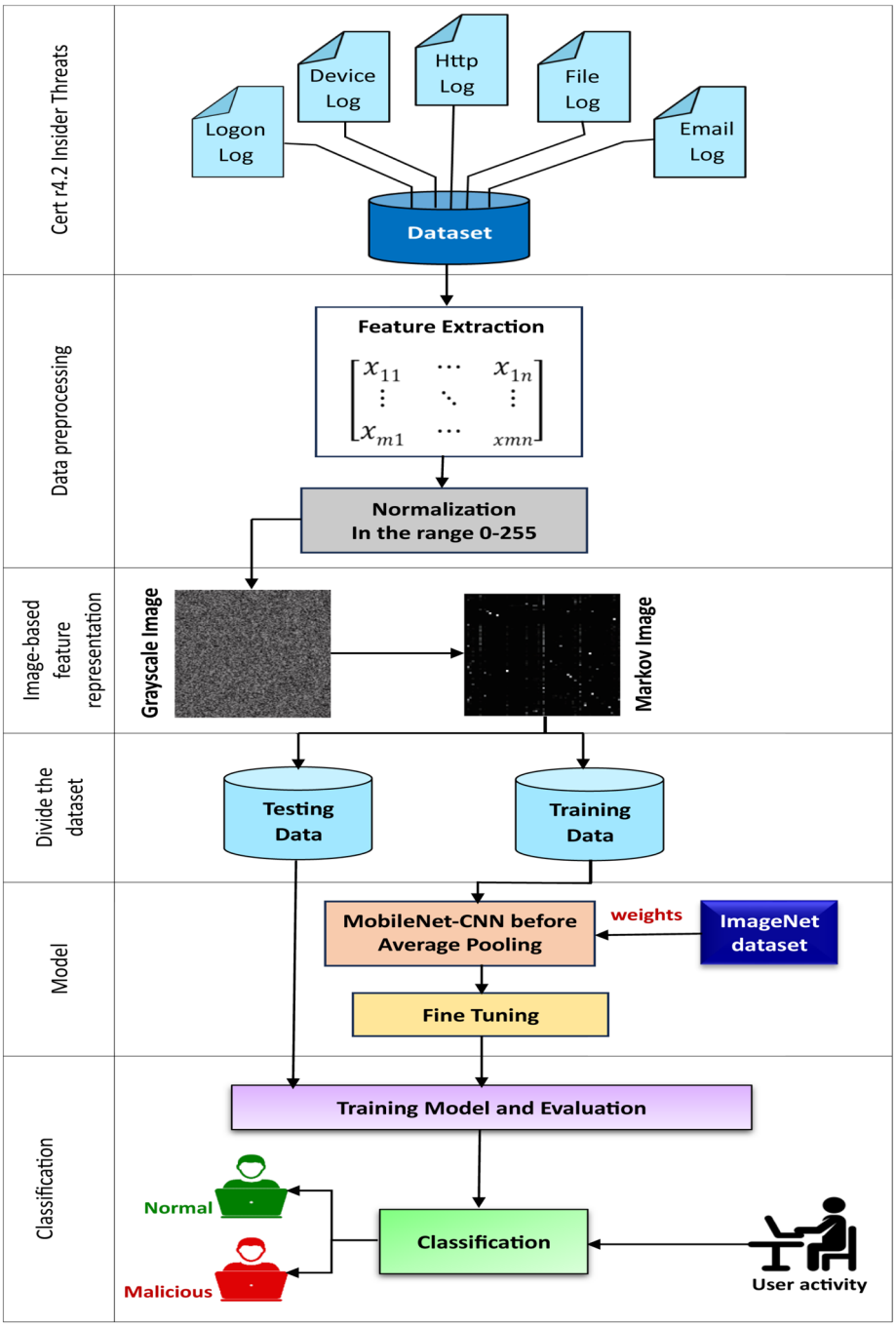


Fig. 1. Overall the MITD-Net framework of the proposed method.

Files details	Count of instances	Malicious users	Normal users
Logon.csv	854,859	33	330
Device.csv	405,380	31	310
File.csv	445,581	29	290
Email.csv	2,629,979	27	270
http.csv	28,434,423	34	340
Total	32,770,222	154	1540

Table 1. Detailed information on the organization of the CERT r4.2 dataset.

Column name	Description
user_id	Identifier for each user in the dataset, crucial for tracking individual behavior patterns over time.
date	The date of the logged activity is useful for analyzing activity trends and identifying temporal anomalies in access patterns.
activity_type	Type of activity (e.g., login, file access, email sent) is important for understanding common or unusual actions within the dataset.
file_id	Identifiers for files accessed or modified are helpful in identifying potentially unauthorized or unusual file access patterns.
pc	The workstation or computer used for the activities is useful for anomaly detection in usage patterns like access from multiple machines.
to/from/cc/bcc	Email fields in email events and analyzed to detect unusual communication patterns like emails to external domains or a high number of recipients.
URL	The URL accessed in web browsing events can be analyzed for access frequency and nature to potentially risky or unauthorized websites.
logon_type	Type of logon (e.g., remote or local), which helps identify unusual remote access that could indicate unauthorized access attempts.
content_size	The size of the file or email content in file/email events, with large transfers flagged for potential data exfiltration attempts.

Table 2. Representation of the key columns in the CERT r4.2 dataset.

Category	Log files	Size	Total users	Total activities	Work days	Total datasets
Employees	logon.csv	55.8 MB	1000	32,770,227	500	Five Files
	email.csv	1.26 GB				
	http.csv	13.5 GB				
Removable devices	device.csv	27.6 MB				
Files	file.csv	184 MB				

Table 3. CERT r4.2 dataset statistics and categories of log files.

Table 1 displays information on the CERT r4.2 dataset. The number of normal users in Logon.csv, http.csv, File.csv, Email.csv, and Device.csv is 330, 340, 290, 270, and 310, respectively, while the number of malicious users in Logon.csv, http.csv, File.csv, Email.csv, and Device.csv is 33, 34, 29, 27, and 31, respectively.

Table 2 provides a brief summary of each of the key columns in the CERT r4.2 dataset and Table 3 presents the statistics and categories of log files in the CERT r4.2 dataset.

Data preprocessing

Preprocessing data is an essential first step in any dataset analysis, but it is especially important for ITC since the precision of the findings has a big impact on how well the suggested works. This section outlines the key preprocessing steps tailored to enhance the CERT r4.2 dataset, focusing on handling missing values, removing outliers, and addressing class imbalance through the over-sampling technique (SMOTE) as:

1) ***Handling missing values:***

Missing data can introduce a substantial amount of bias and uncertainty, affecting the performance of ML models. The approach to handling missing values depends significantly on the nature of the dataset and the extent of missingness. The imputation method was used to handle missing values. The missing values were calculated using mean and average embedding strategies for numeric data, and the KNN technique was also applied.

2) ***Removing outliers:***

Outliers can significantly distort the results of data analysis, especially in statistical computations. The maximum values were replaced with the nearest non-outlier values to remove outliers.

3) ***SMOTE technique:***

ITD frequently involves imbalanced datasets. Instances of malicious activities are significantly rarer than those of normal activities.

This stage of data preprocessing involves two steps. The following are the steps followed in this stage.

A) *Feature extraction*

Feature extraction begins with a thorough analysis of the dataset to identify which attributes are most indicative of insider threat behaviors. This involves distinguishing between normal and malicious events by understanding the scenario and the mechanics of user actions within the organization. The features extracted need to be relevant, informative, and non-redundant to optimize the performance of the classification model. There are 154 malicious and 1540 normal users.

The data preparation process involves selecting relevant parts of the dataset and converting text data into a format suitable for DL models. This step is critical for balancing the trade-off between detection accuracy and

computational resources. Encoding techniques are employed to manage the high dimensionality and variability of the dataset.

The selected features were chosen based on their relevance and contribution to improving model performance:

- Relevance: Features showing a strong relationship with the target variable are considered significant predictors.
- Redundancy Reduction: Ensure that selected features provide unique information and are not highly correlated with each other.
- Performance Improvement: Features that enhance the model's accuracy, precision, recall, and AUC-ROC during cross-validation and testing are considered valuable.

In this study, the dataset was classified into two categories: normal activities and malicious activities. The features identified by [29,38,55] were used, along with additional features, for the ITC, as shown in Table 4. Our methodology involved collecting data over a single day. Feature extraction was then conducted on the data collected to create new instances. Each has a fixed set of features [29,55,56].

After r4.2 dataset preprocessing and feature extraction, this study utilized one-dimensional feature vectors to characterize employee behavior on one day, employing these vectors for behavioral analysis. Subsequently, we transformed these feature vectors into grayscale images, a critical advancement in representing user behavior. This transformation acts as bridging the gap between insider threat analysis and computer vision methods. At present, no computer vision methods are specifically tailored for ITD, which necessitates either modifying existing methodologies or redefining the problem altogether. In this approach, we redefine the problem of insider threat analysis as a computer vision challenge by converting user-behavior feature vectors into Markov images. These feature vectors are derived by extracting pertinent attributes from organizational usage logs, which encapsulate all necessary information to delineate user behavior. The transformed feature vectors into images enable the application of deep neural network techniques for image classification to detect malicious activity. Any deviation from the established behavioral model is identified as malicious activity[38,57–59].

User activity logs are initially encoded into one-dimensional feature vectors that represent various features derived from user activities. These vectors can also be used directly as input for traditional classifiers.

We adopted dictionary techniques in Python for the feature extraction process. As shown in Table 4, several key features were identified and extracted from various log files, focusing on different aspects of user behavior.

This paper utilizes feature selection techniques based on statistical methods. These techniques assist in extracting relevant features from user activity logs, ensuring a comprehensive representation of user behavior. Min-max normalization is used to bring the chosen features to a common scale. The feature selection criteria involve choosing features that effectively distinguish between non-malicious and malicious activities. Comparison

Log files	Features	Description of features selected for one day
http.csv	H1	Count of visits to wikileaks.org
	H2	Count of visits to key-logger sites
	H3	Count of visits to job-search sites
file.csv	F1	Number of executable files downloaded
	F2	"Is file.exe Malware?" F2 was generated using a rule-based automated labeling mechanism designed to simulate malware detection behavior as performed by an antivirus program.
device.csv	D1	Usage count of USB drives outside office hours
	D2	Usage count of external devices
email.csv	E1	Number of emails sent to external domains
	E2	Count of internal emails that a supervisor's account has sent
	E3	The total amount of sent email attachments
	E4	Average size of sent emails
	E5	How many people received the email
logon.csv	L1	Time variation between the initial login and the office start
	L2	Time variation between office finish and last login
	L3	Total login count
	L4	Login count after hours in the office
	L5	The sum of different systems accessed
	L6	The sum of systems used after hours in the office
	L7	Average of sessions conducted after hours in the office
	L8	Number of working hours
	L9	Total login
	L10	Login count outside office hours
	L11	Number of sessions

Table 4. Feature set used in the proposed method.

results show that the proposed approach, which employs image-based feature representation, achieves higher accuracy compared to other methods.

The selected features are chosen for their ability to capture significant patterns in user behavior, which are essential for accurate ITD.

Feature Extraction: Aggregating data for generating numerical vectors, or instances, of a predetermined length N. This process includes features such as averages, counts, and time differences that characterize behavior.

Normalization: Min-max normalization scales features to a common range of [0,1], ensuring data normalization for seamless log analysis and preventing the model from being biased by features with a wide range of values.

Equal Weightage: Giving equal weight to all features during training ensures that the relative comparison of feature values is not influenced by their absolute ranges.

B) *Normalization*

In this approach, we must have a strong normalization process on the extracted features to facilitate the analysis of user activities. Normalization is a crucial technique in the preprocessing of data, particularly in the context of feature extraction for analyzing user activities. This method rescales the features to a fixed range, usually 0 to 1, which helps in standardizing the scale of different variables and enhancing the comparability between distinct types of data. By applying Min-Max normalization, each value in a feature is transformed using Eq.(1).

$$\text{Newvalue} = \frac{value - Min}{Max - Min} \tag{1}$$

where *'Max' and 'Min'* indicate the maximum and minimum values in the features, respectively. This scaling process not only aids in reducing the potential bias that might arise due to the varying scales of raw data but also improves the convergence speed of ML algorithms, which can be sensitive to the scale of input data. Furthermore, normalization ensures that features with smaller ranges do not get overshadowed by features with wider ranges, which is essential for maintaining the integrity of analytical models. As a result, min-max normalization is an indispensable step in preparing data for robust and effective analysis, particularly when dealing with diverse user activities in an organizational context. Figure 2 shows the process of normalizing features of the 1D feature vector.

Image-based feature representation

This stage describes the daily resource utilization patterns for each user. The feature set, includes both frequency-based and statistical features that capture user activities. Initially, these features are grouped by columns to improve understanding and visualization. While the sequence of features used to generate image representations can vary, maintaining a consistent order throughout the process is essential [29]. The images created from the feature extraction process were resized to a standard dimension suitable for input into the MobileNet-CNN model. This resizing allows the model to process the data efficiently while maintaining sufficient detail for accurate classification. To enhance the effectiveness of the suggested method, the 23-featured generated images were resized to 64 × 64. Multiple trials were carried out using different image sizes: (20 × 20), (32 × 32), (64 × 64), (128 × 128), and (256 × 256). Images of smaller sizes, 20 × 20 and 32 × 32, were observed to result in restricted multilevel decomposition, limiting the analysis to single-level and two-level, respectively, leading to inconsistent results. In contrast, testing the MobileNet-CNN model with images sized 64 × 64, 128 × 128, and 256 × 256 yielded improved and consistent results. Among these, the (64 × 64) size was selected for this paper as it balances performance with reduced computing and memory requirements compared to the larger sizes of 128 × 128 and 256 × 256. To prevent distortion of the results, the numeric features require normalization[60]. To scale these features between 0 and 1, we employ the Min-Max approach [38]. After being normalized, these features are presented as 1D feature vectors[61], which depict everyday user activity. Furthermore[62], features retrieved from log files are first represented as grayscale images and subsequently transformed into Markov images. While these images are challenging to differentiate with the naked eye, CNN-based DL models are capable of identifying

Features Extracted	H1…H3	F1…F2	D1…D2	E1…E5	L1…L11
Values after Min-Max Normalizing Process	0.3082541, 0.5, 0.615386	1, 0.5	0.90, 0.25	0.075213, 0.25, 0.507728, 0.517245, 0.5	0.5, 0.25, 0.394736845, 0.5, 0.137931038, 0.398118339, 0.389830509, 0, 0.25, 0.5, 0

Fig. 2. Process of normalizing features of the 1D feature vector.

malicious patterns within these images[38,61]. The proposed image-based approach using MobileNet-CNN offers significant advantages over the two-dimensional matrix approach, particularly in terms of feature preservation, detection accuracy, and adaptability to complex insider threat scenarios. By capturing visual patterns through image representations, the MobileNet-CNN model provides a more detailed and accurate analysis. Although the two-dimensional matrix approach is one of the simplest methods, it may not offer the same level of detail and accuracy. This underscores the importance of using advanced image-based methodologies, as demonstrated in this study.

An example illustrating how employee behavior can be extracted from the dataset is shown in Fig. 3.

This figure displays a sample collection of images, chosen at random, that were generated using the extracted feature vectors. It is evident that the images belonging to (a) malicious and (b) normal classes exhibit a range of non-zero intensity values and a combination of patterns. However, a higher number of non-zero intensity values in user behavior is not always indicative of harmful conduct. Consequently, distinguishing these images by the human eye is challenging. Nevertheless, DL models based on CNNs are capable of identifying malicious patterns in the provided images. In this study, 1D feature vectors with 23 features were normalized and reshaped into 64 $\times$ 64 matrices to create grayscale images. Each feature value is mapped to a pixel intensity, with 0 representing black and 1 representing white. Bilinear interpolation is used to maintain image quality during resizing. These (64 $\times$ 64) pixel grayscale images are then used as inputs for the MobileNet-CNN model, enhancing the ITC by leveraging visual analysis techniques. The model uses both images and 1D feature vectors as inputs to differentiate between malicious and normal activities. Each pixel in a Markov image represents a normalized feature value scaled to a range of 0 to 255. The images are categorized into two groups: normal and malicious. Additionally, 23 features are extracted per user, which can be reshaped into a 16 $\times$ 8 matrix to form a grayscale image. To enhance the performance of this work, the image incorporating these 23 features is resized to 64 $\times$ 64. Figure 4 illustrates the values converted into Markov images after the Min-Max normalization process[63,64].

Normalization to the range 0–255 is performed using the following formula:

$$\text{Normalized in the range 0–255} = \text{Newvalue} \times 255 = \frac{\text{value} - \text{Min}}{\text{Max} - \text{Min}} \times 255. \quad (2)$$

Split the dataset

Creating and verifying models for ITD requires splitting the collected dataset into training and testing subsets. By ensuring that the model is trained on a fraction of the data and assessed on an entirely different subset, this procedure makes it easier to accurately determine how well the model generalizes to new, untested data. Typically, the dataset is partitioned in a 70-30 or 80-20 ratio. 70% or 80% of the data is allocated for training and the remainder for testing. This approach ensures that both subsets are likely to be representative of the overall dataset.

Techniques for Balancing:

- *Random Undersampling:* Applied at different levels to reduce the number of non-malicious instances, making the dataset more balanced but at the risk of losing potentially useful information.
- *SMOTEENN:* Used to not only balance the dataset by augmenting the minority class with synthetic data but also refine the dataset by removing any noisy instances generated through the oversampling process.

Results of using both techniques for the dataset taken from the CERT r4.2 dataset are shown in Figs. 5 and 6.

MobileNet-CNN model

MobileNet is a specialized type of CNN specifically engineered for image classification tasks on mobile devices in real time[65–67]. Its key characteristic is its efficiency, which enables it to operate with minimal processing power. This

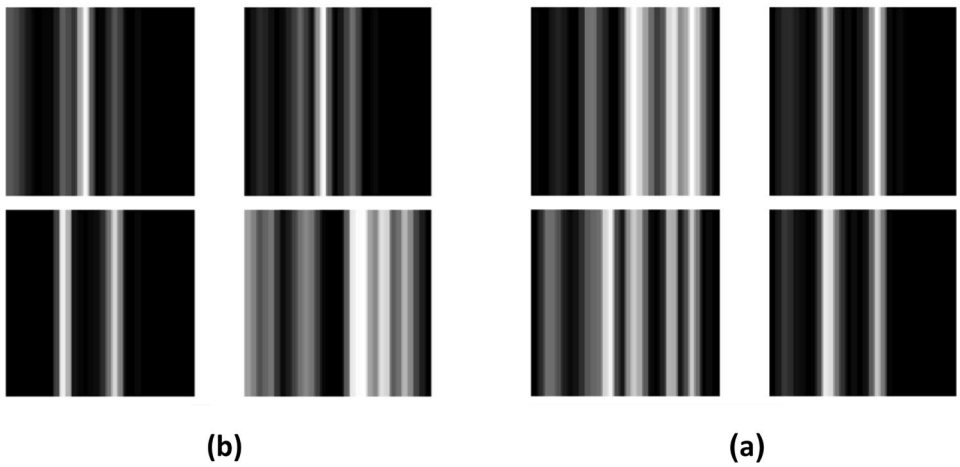


Fig. 3. Samples of feature representation depicting the instances: **a** malicious and **b** normal[38].

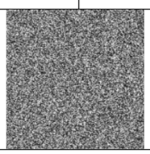
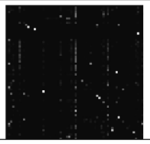
Features Extracted	H1…H3	F1…F2	D1…D2	E1…E5	L1…L11
Values after Min-Max Normalizing Process	0.3082541, 0.5, 0.615386	1, 0.5	0.90, 0.25	0.075213, 0.25, 0.507728, 0.517245, 0.5	0.5, 0.25, 0.394736845, 0.5, 0.137931038, 0.398118339, 0.389830509, 0, 0.25, 0.5, 0
Normalized in the range 0-255	79, 128, 157	255, 128	230, 64	19, 64, 129, 132, 128	128, 64, 100, 128, 35, 101, 99, 0, 64, 128, 0
Grayscale Image Representation					
Conversion into Mrkov Image Representation					

Fig. 4. Values converted into Markov images after the Min-Max normalization process.

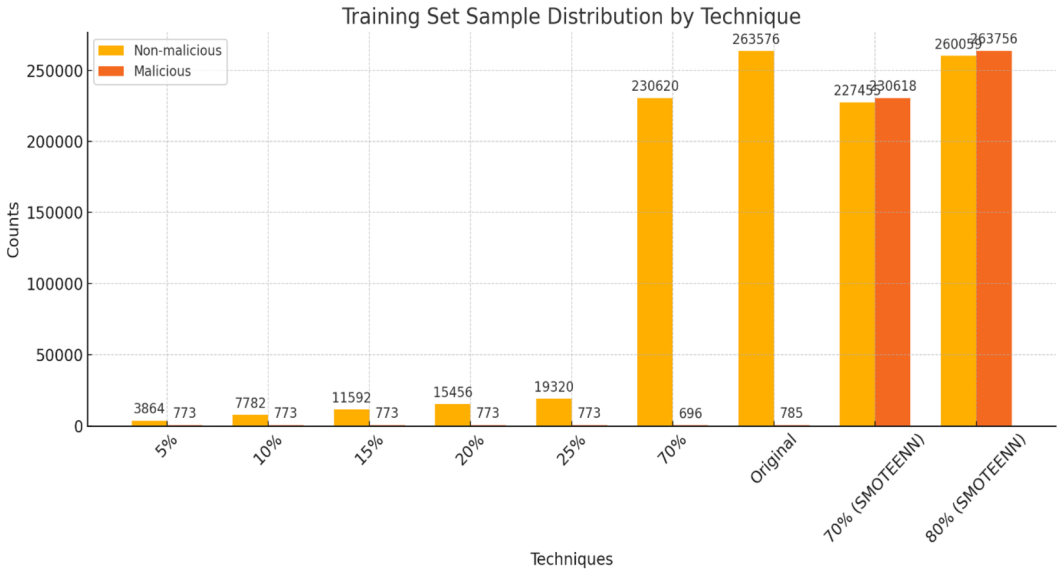


Fig. 5. Applying undersampling and SMOTEENN on the dataset.

makes MobileNet particularly well-suited for implementing transfer learning in environments where resources are limited. The architecture of MobileNet includes subsystems such as depth-wise separable convolution, which itself is divided into two distinct types, Depthwise convolution and pointwise convolution[68,69].

MobileNet utilizes a sophisticated architecture, as depicted in Fig. 7, which features deeply separable convolutions. This design results in a lightweight neural network comparable to an agile terrestrial nervous system. A critical component, the primary layer, consists of a deeply separable filter that serves as the cornerstone of MobileNet. The effectiveness of the network also relies on its thoughtfully crafted structure. Additionally, it offers users the ability to modify width and resolution parameters, providing the flexibility needed for fine-tuning to improve prediction accuracy and precision[70].

CNN is utilized to extract domain-independent image features, enhancing their transferability between the source and target domains. A classification head is used in place of the higher layers of a CNN that has already been trained on ImageNet. This change drastically cuts down on training time, which, in appropriate cases, makes pre-trained models quite desirable. The key objective of transfer learning is to reduce the overhead involved in training new models. There are two ways to modify pre-trained models: fine-tuning and feature extraction. To accommodate new classes, more layers can be added during fine-tuning. In this study, we employ the fine-tuning method, which adapts a pre-trained model to the requirements of a new dataset without the need

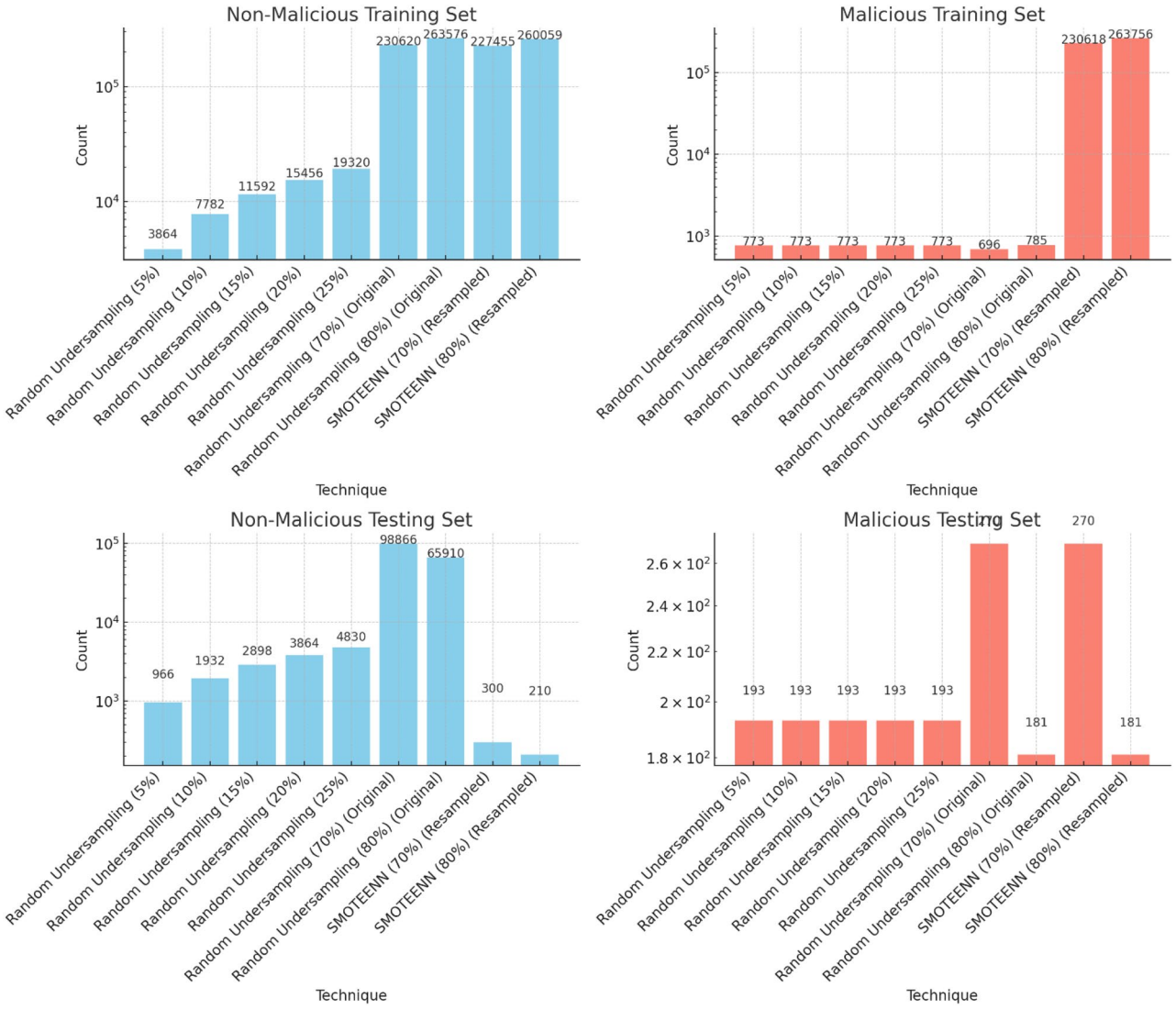


Fig. 6. Results of applying techniques for training and testing.

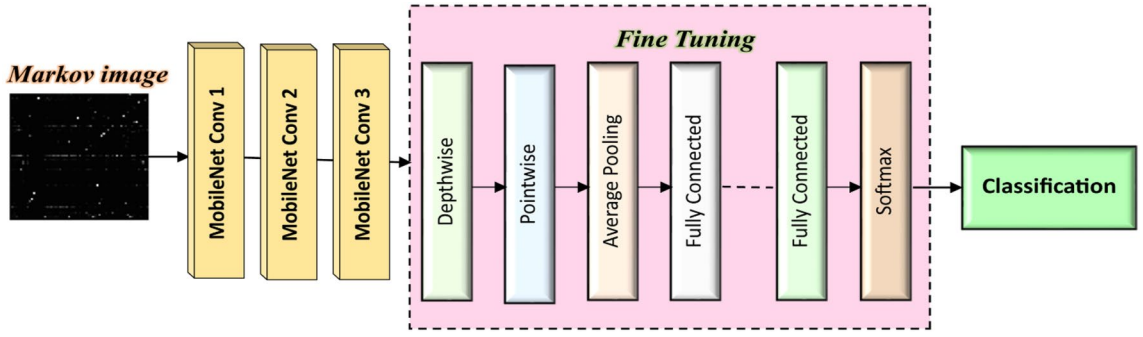


Fig. 7. Architecture of MobileNet-CNN.

for training from scratch, significantly reducing training time. This involves retraining select layers by updating their weights to better align with the features of the new dataset.

Typically, the lower layers of convolutional networks learn generic and simple features applicable to a wide range of images, while the higher layers become increasingly specialized to the dataset used in training. The primary aim of fine-tuning is to leverage these specialized features for the desired dataset instead of completely

relearning basic features. This approach is employed to enhance the performance of model[71]. To fine-tune the models, fully linked layers that function as classification layers were added after a global average pooling layer. In this method, depthwise and pointwise convolutions were followed by the implementation of global average pooling. The global average pooling layer is used to replace the top fully connected layers in CNNs, either totally or partially, in contrast to usual pooling layers that make small adjustments to minimize picture size. The training process is accelerated by this operation's simplicity and absence of trainable characteristics. In order to categorize a picture during the classification process, probabilistic values between 0 and 1 are assigned using the Softmax function. The Softmax layer receives the aggregated values from a feature map in the global average pooling layer. The ImageNet weights from the original model's training are used to initialize the weights. It uses the binary cross-entropy logarithmic loss function because there are only two class variables. The Softmax function is used in classification tasks to assign probabilistic values, facilitating the classification of an image. Table 5 indicates the details of the hyper-parameters.

Classification

This stage addresses the classification of created images as normal or malicious is the focus of this stage. Two sets of tests were carried out. In the first experiment, grayscale image classification was done using MobileNet. The produced Markov image data were classified using the MobileNet-CNN model in the second experiment[72,73]. Markov image representations are processed as input by the MobileNet-CNN model, which then passes them through a number of computational units to produce representations with the proper classification score. The MobileNet-CNN model uses three convolution layers to reduce the size of the feature maps, as shown in Fig. 7. Without compromising accuracy, this technique substitutes max pooling[74].

In the ITC model, a global average pooling layer is placed immediately before the fully connected layers to optimize network performance while minimizing the parameter count. This layer acts as an energy layer by computing the average of the activated outputs across each feature map. The tested model incorporates generalized convolution and pooling layers, utilizes four decomposition levels with wavelets for multi-resolution analysis, and includes a global average pooling layer functioning as an energy layer. This configuration is followed by two dense layers. The results section presents a comparative analysis of the proposed setup and its network performance relative to previous models.

Results and discussion

In this section, the CERT r4.2 dataset is used to assess the MITD-Net model. With 98.11% accuracy, the MobileNet-CNN model accomplished an insightful breakthrough in identifying insider threats.

The present analysis excludes adversarially constructed insider behaviors and longitudinal temporal variation. Because of this, MITD-Net might be susceptible to concept drift or complex evasion techniques. Future studies will concentrate on giving the framework adversarial robustness (e.g., data perturbation analysis, adversarial training) and adaptive learning capabilities (e.g., online learning, continual training, domain adaptation). Furthermore, in order to measure performance deterioration and recovery over time, a temporal evaluation benchmark utilizing red-teamed simulations or progressive log updates would be necessary.

Experiment environment

The publicly available CERT r4.2 dataset was used to validate the proposed method. It has 32,770,227 activities performed during 500 days of work and includes 1,540 normal users and 154 malicious users. The experiment hardware specification: Intel Core i9 processor, 32 GB of RAM, and Nvidia GeForce RTX 3060 with 12 GB of memory. The model has a computational complexity of 1.519 million parameters and a size of 13.5 MB, with response times of 13 ms on a CPU and 4 ms on a GPU.

Evaluation metrics

In the context of developing and validating ITD systems, the Evaluation Metrics section of this paper is crucial for quantifying the effectiveness of the proposed models. We used common metrics and visualizations to examine MITD-Net's false positive behavior in order to measure its operational utility. The model rarely misclassifies benign users, as evidenced by its low false alarm rate of 1.17%. High AUC and precision values are also displayed by the ROC and Precision-Recall curves, indicating strong robustness in differentiating insider threats from typical behavior. Depending on organizational tolerance for alert fatigue, modifying the detection threshold in a practical deployment can further reduce false positives at the expense of somewhat lower recall.

This section should thoroughly detail the metrics used to assess model performance, emphasizing how these metrics capture the accuracy, precision, and reliability of anomaly detection. Commonly used evaluation metrics in this domain include false alarm rates, accuracy, recall, precision, and F1-score, which are calculated as follows:

Hyper-parameters	MobilNet-CNN
Input image	(64,64)
Weight	Initialized to ImageNet dataset
Batch size	64
Epochs	15
Classifier	Softmax

Table 5. MobileNet-CNN hyperparameter details.

$$\text{FAR} = \frac{FP}{FP + TN} \tag{3}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{F1} - \text{Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

Table 6 shows the results obtained from the confusion matrix of MITD-Net using the test split of the CERT r4.2 dataset:

Based on the values in Table 6, we have calculated to be the false alarm rates (FPR) from Eq. (3) was 1.17%. Compared to conventional rule-based detection systems, which often display false positive rates of more than 5–10%, this is operationally acceptable for a large number of enterprise environments.

This metric reflects the percentage of benign user actions that are incorrectly flagged as malicious.

The counts of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) are shown in Fig. 8a. This breakdown is particularly crucial in the security domain, where the differentiation between these categories can help stakeholders understand the model's strengths and weaknesses.

The confusion matrix that was obtained from the proposed model's testing phase is shown in Fig. 8a. The confusion matrix demonstrates the model's high performance, indicating its success in learning and identifying malicious activity. Additionally, Fig. 8b displays the ROC curve. AUC (Area Under Curve) is 0.991, this demonstrates excellent separability between benign and malicious behavior. Figure 8c and d illustrate the model accuracy graph and model loss graph, respectively.

Experiment results

In this section, we performed a series of comparative experiments using the CERT r4.2 dataset. As detailed in the section "Proposed MITD-Net framework", features were encoded both as one-dimensional feature vectors and as images. Random sampling and SMOTEENN techniques were employed with various sampling ratios to address the class imbalance. In order to identify the optimal configuration, we experimented with various ratios to divide the data into training and testing sets.

This study's main goal is to find out how well the suggested MITD-Net framework uses MobileNet-CNN architecture and image-based feature representation to identify insider threats. Therefore, to ensure fair comparison with previous methods and to highlight classification performance metrics like accuracy, precision, recall, and F1-score, the study was designed and evaluated in a controlled experimental setting with fixed datasets (CERT r4.2). With separable convolutions at depth and reduced parameter complexity, the MobileNet architecture is known to be efficient and suitable for low-resource environments. However, we did not conduct thorough evaluations pertaining to real-time system integration, computational throughput under memory or CPU constraints, or handling dynamic streaming data. Despite their importance, these practical aspects were purposefully left out of the focus of this research stage and will be discussed in future research.

Figure 9a–d show curves for performance evaluation on the grayscale image using MobileNet-CNN.

The performance evaluation presented in Fig. 9 illustrates the capabilities of the MobileNet-CNN when utilized for the analysis of grayscale images at varying levels of training intensity, specifically 70% and 80%. These insights are crucial, as they demonstrate how the handling of training data volume affects the overall effectiveness of the model in practical scenarios. The findings revealed that there is an increasing trend in performance for all the assessed measurements when the percentage of training data increases from 70% to 80%. This generalization is quite interesting, particularly for neural networks: if there's more data available, a larger model should, in general, be able to use it better. This is evidenced by the increased Accuracy and Recall rates, suggesting that the model becomes better at classifying images correctly and retrieving a higher proportion of relevant instances. Its robustness is marked by a high accuracy. Nevertheless, the accuracy increases with more training data, emphasizing the importance of enough training to reach optimal performance, especially for transferring to grayscale images, for which the separation of the feature is greater than the separation of texts. Precision, indicating the model's capability to avoid false positives, increases as the size of the training dataset grows. This means the model is better at separating different classes as it's being trained. Similarly, Recall also improves, indicating the model's enhanced capability to identify all relevant samples from the dataset. The F1-score, which balances Precision and Recall, is notably higher with larger training datasets. This metric is particularly important in scenarios where maintaining a balance between precision and recall is crucial, such as

Prediction ground truth	Malicious	Non-malicious
Malicious	948	12
Non-Malicious	28	1,012

Table 6. Results obtained from the confusion matrix.

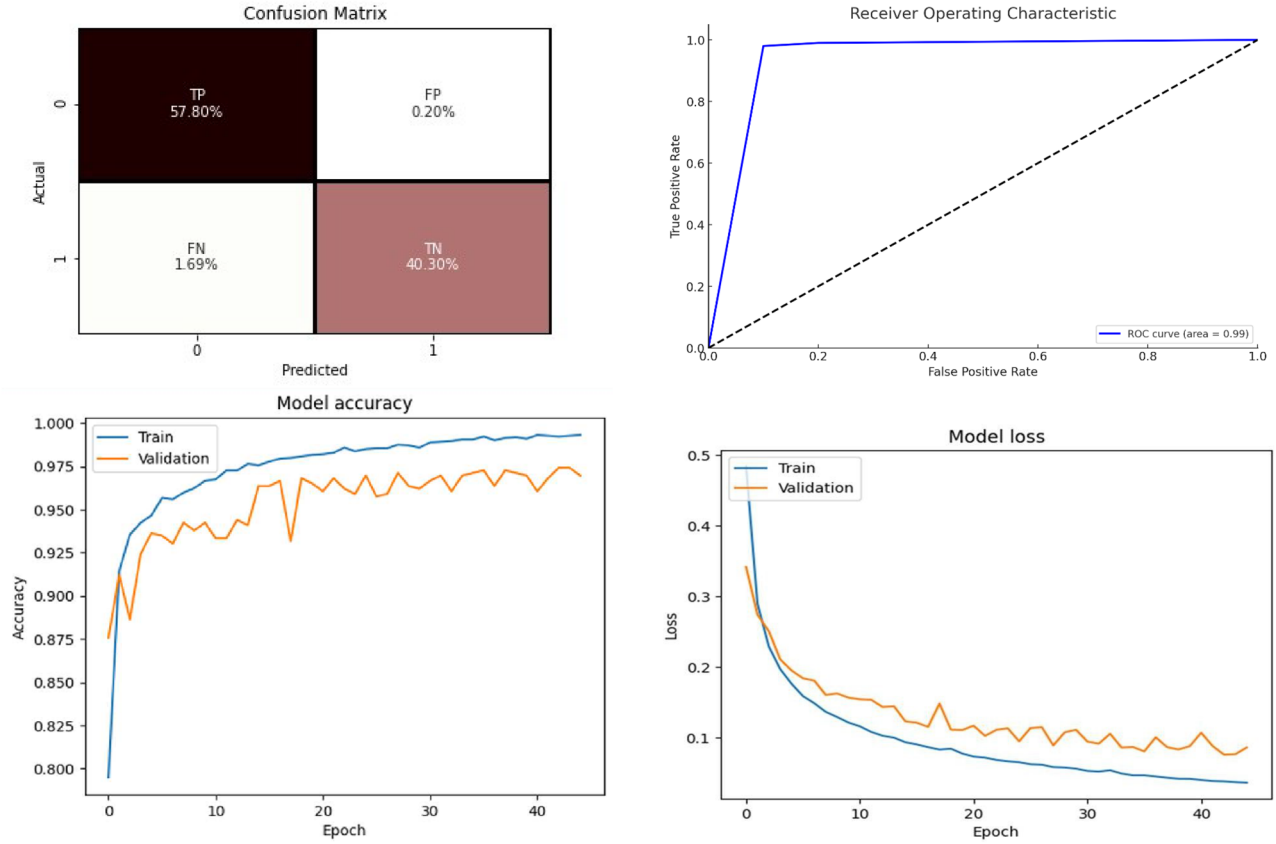


Fig. 8. Evaluation of the proposal model: **a** Confusion matrix, **b** Roc curve, **c** Accuracy, and **d** Loss.

in automated systems where the costs of false positives and false negatives are equally prohibitive. However, the increasing complexity of these models makes them less suitable for real-time applications or devices with limited resources, as they require greater memory and processing power.

Figure 10a–d show curves for performance evaluation on the Markov image using the proposed model. The performance evaluation presented in Fig. 10 illustrates the high capabilities of the MobileNet-CNN when utilized for the analysis of Markov images at varying levels of training intensity, specifically 70% and 80%. The data shows a high trend of improved performance across all the metrics evaluated compared to grayscale images. Where the percentage of training increases from 70% to 80%. This improvement is particularly important in the context of neural networks, where the availability of a larger dataset generally enhances the model's ability to generalize. This is evidenced by the high increase in resolution and recall rates, which suggests that the model becomes very much better at correctly classifying images and retrieving a higher percentage of relevant cases. High accuracy levels in model durability. However, the increase in accuracy with more training data underscores the crucial role of comprehensive training in achieving optimal performance, especially when dealing with the fine details of Markov images, where feature recognition may be clearer than in grayscale images.

Comparison of performance with similar works

Analysis based on Markov images has shown to be highly promising. In this phase, state-of-the-art studies are compared with the proposed study. This approach utilizes a pre-trained DL model, adjusted to generate new labels, motivating us to frame ITC as a problem in transfer learning. By leveraging the pre-trained model and adapting it to obtain new labels, we have structured ITC as a transfer learning issue.

Table 7 provides a detailed performance comparison of recent state-of-the-art methods with the proposed method in insider threat prediction. The MobileNet-CNN model enhances overall performance by extracting both spectral and spatial features. This proposed method yields superior results compared to the most recent advanced technologies. We compare the performance of our proposed model with the following baselines.

Bi-LSTM with SVM[22]: This approach employs a bi-LSTM model for efficient feature extraction to present a user behavior-based method using a flexible, anomaly detection approach enhanced by augmented decision-making.

LSTM autoencoder[23]: This model uses an LSTM Autoencoder through behavioral analysis. It classifies user activity as normal or malicious using a rich feature set based on events like logon/logoff, user roles, and functional units.

SENTINEL[24]: This approach introduces a model that uses a dynamic user behavior interaction graph to capture both spatial and temporal aspects of user activity. Leveraging a spatio-temporal graph neural network

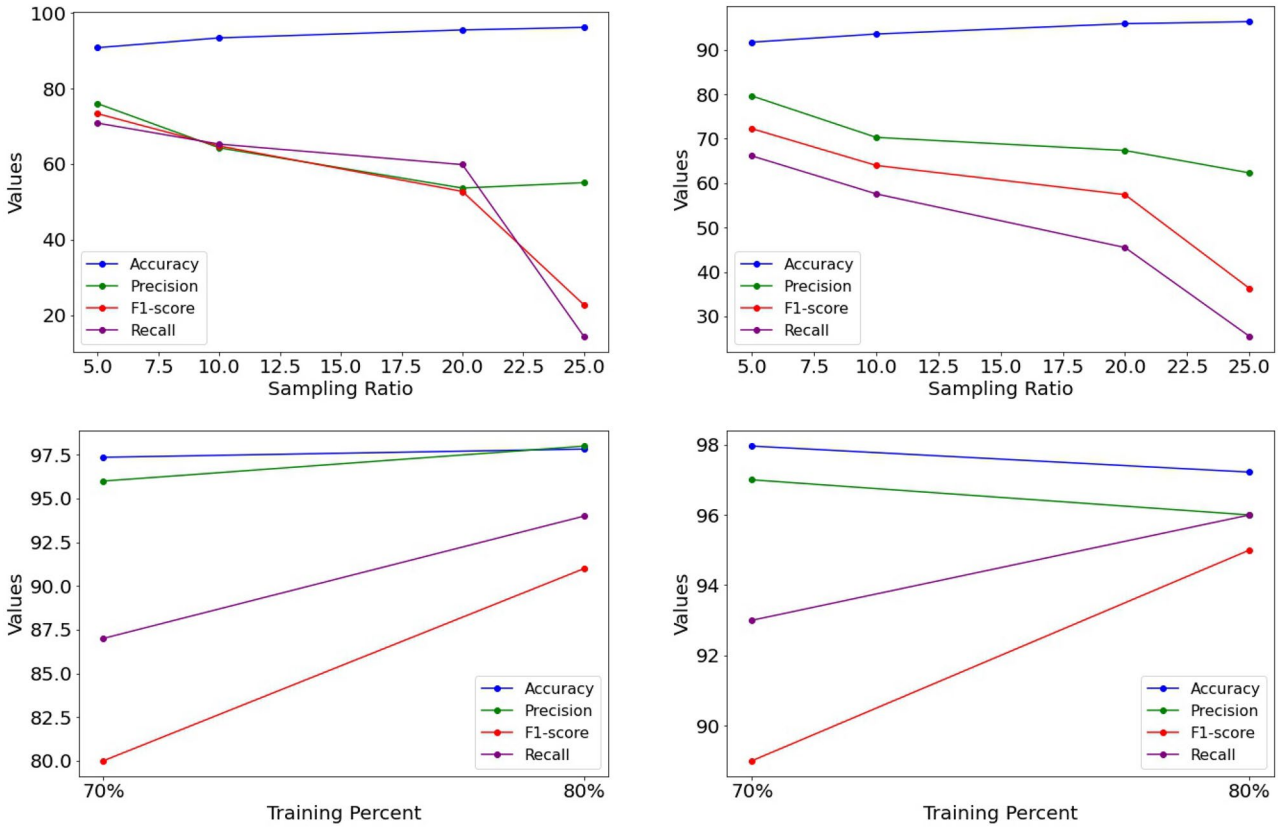


Fig. 9. Performance evaluation of training by the Markov image: **a** 70% for sampling, **b** 80% for sampling, **c** 70% for SMOTEENN, and **d** 80% for SMOTEENN.

and a multi-timescale fusion mechanism to detect both abrupt and persistent threats at the log-entry level without needing attack samples during training.

ResHybnet[25]: In this model, integrate a feature engineering approach by combining manual and automatic features from daily user activities. An LSTM auto-encoder extracts sequential features, while a residual hybrid network with GNN and CNN components enhances detection using an organizational graph, where each node represents a user-day combination.

M-EOS[35]: This approach introduces a Stacked CNN–Attentional BiGRU model using user activity logs and information. It performs time-series classification by learning temporal and numerical features through stacked generalization. The combined model enhances feature representation and classification accuracy with reduced complexity. Additionally, a Modified-Equilibrium Optimization strategy is evaluated using non-linear time control, chaos-based, and opposite-based learning techniques.

AUTH[27]: This paper presents a model, an unsupervised method based on an adversarial autoencoder. The model incorporates both time and event features and combines a temporal convolutional network with an LSTM in a TL-AAE model to better detect covert threats.

DTITD[28]: This paper proposes DTITD that leverages digital twins and self-attention-based deep learning to monitor and analyze insider risks in real time. It introduces advanced language data augmentation using BERT and GPT-2 to address data imbalance, with results showing GPT-2–based sentence embedding performs best.

DTGI[33]: This model utilizes a continuous-time dynamic heterogeneous graph network and a behavior context-constrained anomaly sample generator to augment attack samples, thereby enhancing the performance of supervised models in highly imbalanced real-world data.

ITDSTS[30]: This approach presents a scenario-oriented model targeting privilege abuse, identity theft, and data leakage. It serializes and vectorizes daily user behavior, incorporates contextual features, and reconstructs abnormal behavior backgrounds to improve detection accuracy. A time series model with a multi-head attention mechanism captures long-distance dependencies and behavior correlations, enhancing feature representation for robust anomaly detection.

Ripple2Detect[31]: This model is a multi-step framework that builds an evidence sequence library from known attacks and uses a knowledge graph to assess behavioral correlations. A BERT-based semantic similarity model is trained for operation sequences, while contrastive learning addresses data imbalance. A preference propagation mechanism is then used to predict attack behaviors within the graph.

SMOTEENN-WCNN[29]: This paper presents the WCNN model, which transforms one-dimensional user activity features into image-based representations to capture visual patterns. By combining spectral and spatial

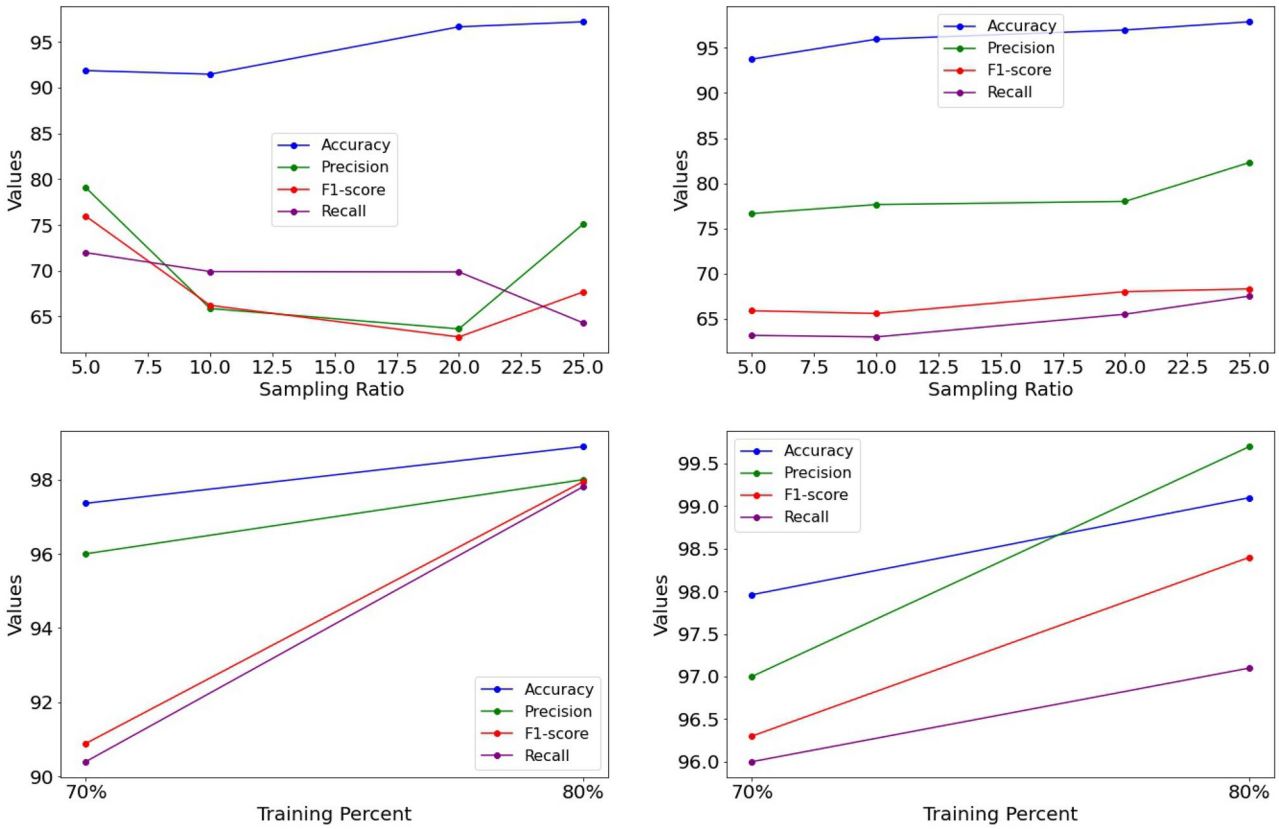


Fig. 10. Performance evaluation of training by the grayscale image: **a** 70% for sampling, **b** 80% for sampling, **c** 70% for SMOTEENN, and **d** 80% for SMOTEENN.

Model	Accuracy	Precision	Recall	F1-score	Input type
Bi-LSTM with SVM[22]	87.5	85.1	82.3	82.7	Sequence
LSTM Autoencoder[23]	90	97	✗	94	
ResHybnet[25]	92.62	91.52	92.87	92.19	
M-EOS[35]	92.52	98	95	96	
AUTH[27]	93.2	✗	✗	✗	
DTITD[28]	93.53	98.13	80.77	88.61	
ITDSTS[30]	96	94	92	93	
Ripple2Detect[31]	96.4	96.69	96.09	96.39	
LSTM with Gray Encoding [32]	97.23	99.27	94.15	96.64	
DS-IID[26]	97.31	79	86	82	
SENTINEL[24]	92.96	✗	✗	✗	Graph
DTGI[33]	95.52	95.52	96.79	96.32	
DD-GCN[34]	98.21	✗	✗	93.04	
ITDE[21]	**99**	99	97	98	
SITD[36]	✗	93.62	88	90.72	
Ge et al.[37]	✗	88.43	✗	86.19	
DCNNs[38]	96.34	96.80	97.12	98.59	
SMOTEENN-WCNN[29]	97.19	95	**99**	97	Image
Rohini et al.[39]	97.54	97.3	96.22	97.45	
MITD-Net (ours)	98.11	**99.66**	97.16	98.39	

Table 7. Performance comparison of recent state-of-the-art techniques with the proposed approach.

Component removed	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
MITD-Net	98.11	99.66	97.16	98.39
Feature vector	92.10	90.00	86.50	88.21
Feature selection	88.30	86.20	82.70	84.41
Grayscale images	93.40	92.80	87.30	89.96
Markov Images	91.20	89.00	85.10	87.00
Without MobileNet-CNN (using VGG19)	90.16	94.70	94.13	94.10
Without MobileNet-CNN (using ResNet18)	95.60	96.41	94.98	95.69
Without MobileNet-CNN (using EfficientNet-B0)	96.50	97.64	90.48	93.92

Table 8. Performance of the proposed model for the CERT r4.2 dataset using different components.

analysis, WCNN effectively identifies malicious behavior. The SMOTEENN sampling technique is used to address class imbalance.

SITD[36]: This paper introduces a method that identifies threats by comparing sample pairs rather than direct classification using ResNet18. By focusing on whether pairs belong to the same category, it avoids common classification issues. An improved contrastive loss function emphasizes differing-category pairs, significantly boosting detection performance.

Rohini et al.[39] addressed the high false alarm rates in insider threat detection methods, which can disrupt organizational workflow and lower morale. To mitigate this, it proposes a new methodology that categorizes insider threats using image-based classification techniques to improve detection accuracy and reduce false positives.

Jiao et al.[75] proposed a deep transfer learning model that converts single-day user behavior data into grayscale images for intuitive analysis and enhanced insider threat detection, utilizing architectures such as AlexNet, VGG16, ResNet18, and ResNet50. However, the accuracy of all four models did not exceed 85% based on the comparison figure.

Ge et al.[37] propose an insider threat detection method using residual convolutional neural networks for multi-source data fusion. It transforms tabular data into neighborhood-correlated feature maps for contextual information extraction and employs skip connections for multi-level feature fusion, enabling the detection of complex, previously undetectable threat patterns.

LSTM with gray encoding [32]: The LSTM-based deep learning model is used. It distinguishes normal from malicious user activities by combining sentiment analysis and gray encoding to preserve temporal correlations. The system reformats activity data into variable-length samples for effective training and improved threat detection.

DS-IID[26]: This model uses deep feature synthesis to create detailed user profiles and binary deep learning to evaluate generative algorithms, and differentiates real from synthetic abnormal profiles. The model also applies weighted random sampling to manage data imbalance.

DD-GCN[34]: A Dual-Domain Graph Convolutional Network for adaptive and accurate ITD is proposed. It models user features and relationships as heterogeneous graphs and applies a weighted feature similarity mechanism to fuse user behavior and structural information. Graph embeddings from both original and fused structures are extracted, and an attention mechanism assigns adaptive importance to user features. Combination and difference constraints further enhance the embeddings' ability to capture shared and unique threat indicators.

ITDE[21]: This model uses a graph neural network with two-layer attention. It represents user features and relationships as heterogeneous graphs derived from behavior analysis. Node-level and semantic-level attention mechanisms are used to capture complex structural information and generate informative embeddings by aggregating features from meta-path-based neighbors

Ablation study

An ablation study is crucial to highlight the contribution of each component in the framework. This will help in understanding the significance of feature selection, feature vectors, grayscale images, Markov images, and the pre-trained MobileNet-CNN using transfer learning. We conducted experiments on the CERT r4.2 dataset, using the same training and testing split for consistency. The evaluation metrics include accuracy, precision, recall, and F1-score. The ablation study investigates the impact of different components in the MITD-Net framework on the overall performance.

We incrementally remove or modify components and measure the changes in performance metrics such as accuracy, precision, recall, and F1-score. Table 8 provides a detail of this study.

The best feature selection, second-best MobileNet-CNN, third-best Markov images, fourth-best feature vector, and fifth-best Grayscale images on the Cert r4.2 dataset, respectively.

Conclusions

In this study, we introduced MITD-Net, a novel deep Markov image-based model for threat detection. MITD-Net underscores the critical need for advanced predictive technologies to combat the increasing sophistication of malicious activities within organizational applications. The adoption of the MobileNet-CNN model marks a significant advancement in detecting MUB. This model effectively addresses the limitations of traditional ML and DL methods, which often struggle to classify new or unknown threats and tend to produce high rates

of false negatives and positives. The MobileNet-CNN leverages the MobileNet architecture, renowned for its computational efficiency and adaptability, making it particularly useful in resource-constrained environments. This enables real-time prediction of malicious behaviors, significantly enhancing system security by providing system administrators with a robust tool for preemptive threat identification and mitigation. Experimental evaluations show the model's superior performance, achieving an accuracy of 98.11%, precision of 99.66%, recall of 97.16%, and f1-score of 98.39%, all while maintaining low false positive and false negative rates. These findings highlight the potential of the MobileNet-CNN model in threat classification on insider threats, and also provide evidence that it may outperform the current state-of-the-art methods, and provide potential directions for future work.

Although the proposed approach demonstrates a high success rate on the CERT r4.2 dataset, it does have certain limitations. For instance, the computational cost is due to using the CNN. Moreoverr, its generalizability to real-world environments remains an open research direction.

Future work will involve validating the framework across diverse organizational datasets and insider threat scenarios to ensure robustness, adaptability, and operational reliability in dynamic and heterogeneous settings.

Data availability

The datasets are available from the corresponding author upon reasonable request.

Received: 3 July 2025; Accepted: 8 September 2025
Published online: 09 October 2025

References

1. Jeon, G., Jin, H., Lee, J. H., Jeon, S. & Seo, J. T. Iwtw: A framework for iowt cyber threat analysis. *CMES-Comput. Model. Eng. Sci.* **141**, 1575 (2024).
2. Latif, M. A. et al. Oversampling-enhanced feature fusion-based hybrid vit-1dcnn model for ransomware cyber attack detection. *Comput. Model. Eng. Sci. (CMES)* https://doi.org/10.32604/cmes.2024.056850 (2025).
3. Alzaabi, F. R. & Mehmood, A. A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. *IEEE Access* **12**, 30907–30927 (2024).
4. Wisnubroto, D. S., Khairul, K., Basuki, F. & Kristuti, E. Preventing and countering insider threats and radicalism in an Indonesian research reactor: Development of a human reliability program (hrp). *Heliyon* **9**, e15685 (2023).
5. Yuan, S. & Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Comput. Secur.* **104**, 102221 (2021).
6. Erola, A., Agrafiotis, I., Goldsmith, M. & Creese, S. Insider-threat detection: Lessons from deploying the citd tool in three multinational organisations. *J. Inf. Secur. Appl.* **67**, 103167 (2022).
7. Ferreira, P., Le, D. C. & Zincir-Heywood, N. Exploring feature normalization and temporal information for machine learning based insider threat detection. In *2019 15th International Conference on Network and Service Management (CNSM)*, 1–7 (IEEE, 2019).
8. Li, D. et al. Image-based insider threat detection via geometric transformation. *Secur. Commun. Netw.* **2021**, 1–18 (2021).
9. Jiang, J. et al. Anomaly detection with graph convolutional networks for insider threat and fraud detection. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*, 109–114 (IEEE, 2019).
10. Liu, L., De Vel, O., Chen, C., Zhang, J. & Xiang, Y. Anomaly-based insider threat detection using deep autoencoders. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 39–48 (IEEE, 2018).
11. Lavanya, P., Glory, H. A. & Sriram, V. S. Mitigating insider threat: a neural network approach for enhanced security. *IEEE Access* **12**, 73752–68 (2024).
12. Meng, F., Lou, F., Fu, Y. & Tian, Z. Deep learning based attribute classification insider threat detection for data security. In *2018 IEEE third international conference on data science in cyberspace (DSC)*, 576–581 (IEEE, 2018).
13. Lin, L., Zhong, S., Jia, C. & Chen, K. Insider threat detection based on deep belief network feature representation. In *2017 international conference on green informatics (ICGI)*, 54–59 (IEEE, 2017).
14. Zhang, J., Chen, Y. & Ju, A. Insider threat detection of adaptive optimization dbn for behavior logs. *Turk. J. Electr. Eng. Comput. Sci.* **26**, 792–802 (2018).
15. Le, D. C. & Zincir-Heywood, A. N. Machine learning based insider threat modelling and detection. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 1–6 (IEEE, 2019).
16. Jiang, J. et al. Prediction and detection of malicious insiders' motivation based on sentiment profile on webpages and emails. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, 1–6 (IEEE, 2018).
17. Singh, M., Mehtre, B. M. & Sangeetha, S. User behavior profiling using ensemble approach for insider threat detection. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, 1–8 (IEEE, 2019).
18. Algabri, R. & Choi, M.-T. Online boosting-based target identification among similar appearance for person-following robots. *Sensors* **22**, 8422 (2022).
19. Abdu, A. et al. Cross-project software defect prediction based on the reduction and hybridization of software metrics. *Alex. Eng. J.* **112**, 161–176 (2025).
20. Algabri, R., Shin, H., Abdu, A., Bae, J.-H. & Lee, S. Wquatnet: Wide range quaternion-based head pose estimation. *J. King Saud Univ. Comput. Inf. Sci.* **37**, 24 (2025).
21. Tian, T. et al. Insider threat detection based on heterogeneous graph neural network. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 628–635 (IEEE, 2023).
22. Singh, M., Mehtre, B. & Sangeetha, S. User behaviour based insider threat detection in critical infrastructures. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 489–494 (IEEE, 2021).
23. Nasir, R., Afzal, M., Latif, R. & Iqbal, W. Behavioral based insider threat detection using deep learning. *IEEE Access* **9**, 143266–143274 (2021).
24. Xiao, F. et al. Sentinel: Insider threat detection based on multi-timescale user behavior interaction graph learning. *IEEE Trans. Netw. Sci. Eng.* **12**(2), 774–790 (2024).
25. Hong, W. et al. A graph empowered insider threat detection framework based on daily activities. *ISA Trans.* **141**, 84–92 (2023).
26. Kotb, H. M., Gaber, T., AlJanah, S., Zawbaa, H. M. & Alkhathami, M. A novel deep synthesis-based insider intrusion detection (ds-iid) model for malicious insiders and ai-generated threats. *Sci. Rep.* **15**, 207 (2025).
27. Zhu, X. et al. Auth: An adversarial autoencoder based unsupervised insider threat detection scheme for multisource logs. *IEEE Trans. Ind. Inform.* **20**(9), 10954–65 (2024).
28. Wang, Z. Q. & Saddik, A. Dtitd: An intelligent insider threat detection framework based on digital twin and self-attention based deep learning models. *IEEE Access* **11**, 114013–30 (2023).

29. Randive, K., Mohan, R. & Sivakrishna, A. M. An efficient pattern-based approach for insider threat classification using the image-based feature representation. *J. Inf. Secur. Appl.* **73**, 103434 (2023).
30. Tian, T., Zhang, C., Jiang, B., Feng, H. & Lu, Z. Insider threat detection for specific threat scenarios. *Cybersecurity* **8**, 17 (2025).
31. Liu, H., Liu, M., Han, L., Sun, H. & Fu, C. Ripple2detect: A semantic similarity learning based framework for insider threat multi-step evidence detection. *Comput. Secur.* **154**, 104387 (2025).
32. AlSlaiman, M., Salman, M. I., Saleh, M. M. & Wang, B. Enhancing false negative and positive rates for efficient insider threat detection. *Comput. Secur.* **126**, 103066 (2023).
33. Gao, P. et al. Deep temporal graph infomax for imbalanced insider threat detection. *J. Comput. Inf. Syst.* **65**, 108–118 (2025).
34. Li, X. et al. A high accuracy and adaptive anomaly detection model with dual-domain graph convolutional network for insider threat detection. *IEEE Trans. Inf. Forensics Secur.* **18**, 1638–1652 (2023).
35. Anju, A. & Krishnamurthy, M. M-eos: modified-equilibrium optimization-based stacked cnn for insider threat detection. *Wirel. Netw.* **30**, 2819–2838 (2024).
36. Zhou, S., Wang, L., Yang, J. & Zhan, P. Sitd: Insider threat detection using Siamese architecture on imbalanced data. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 245–250 (IEEE, 2022).
37. Ge, D., Zhong, S. & Chen, K. Multi-source data fusion for insider threat detection using residual networks. In *2022 3rd International Conference on Electronics, Communications and Information Technology (CECIT)*, 359–366 (IEEE, 2022).
38. Gayathri, R., Sajjanhar, A. & Xiang, Y. Image-based feature representation for insider threat classification. *Appl. Sci.* **10**, 4945 (2020).
39. Rohini, V., Mohan, R. & Sivakrishna, A. M. Insider threat detection on cert data using pre-trained resnet. In *2024 Global Conference on Communications and Information Technologies (GCCIT)*, 1–6 (IEEE, 2024).
40. Priyadarshi, P. & Kumar, P. A comprehensive review on insider trading detection using artificial intelligence. *J. Comput. Soc. Sci.* **7**(2), 1645–64 (2024).
41. Bharathi, S. & Balasubramanian, C. Non-trusted user classification-comparative analysis of machine and deep learning approaches. In *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 316–324 (IEEE, 2022).
42. Pal, P., Chattopadhyay, P. & Swarnkar, M. Temporal feature aggregation with attention for insider threat detection from activity logs. *Expert Syst. Appl.* **224**, 119925 (2023).
43. Xiao, J. et al. Robust anomaly-based insider threat detection using graph neural network. *IEEE Trans. Netw. Serv. Manag.* **20**(3), 3717–33 (2022).
44. Karatas, G., Demir, O. & Sahingoz, O. K. Deep learning in intrusion detection systems. In *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 113–116 (IEEE, 2018).
45. Kim, A. et al. Sok: A systematic review of insider threat detection. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **10**, 46–67 (2019).
46. Le, D. C. & Zincir-Heywood, N. Machine learning based insider threat modelling and detection. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* 1–6 (2019).
47. Garchery, M. & Granitzer, M. Adsage: Anomaly detection in sequences of attributed graph edges applied to insider threat detection at fine-grained level. ArXiv:2007.06985 (2020).
48. Hu, T. et al. An insider threat detection approach based on mouse dynamics and deep learning. *Secur. Commun. Netw.* **2019**(1), 3898951 (2019).
49. Yang, G., Cai, L., Yu, A. & Meng, D. A general and expandable insider threat detection system using baseline anomaly detection and scenario-driven alarm filters. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 763–773 (IEEE, 2018).
50. Khan, M. Z. A., Khan, M. M. & Arshad, J. Anomaly detection and enterprise security using user and entity behavior analytics (ueba). In *2022 3rd International Conference on Innovations in Computer Science & Software Engineering (ICONICS)*, 1–9 (IEEE, 2022).
51. Raut, M., Dhavale, S., Singh, A. & Mehra, A. Insider threat detection using deep learning: A review. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 856–863 (IEEE, 2020).
52. Saaudi, A., Al-Ibadi, Z., Tong, Y. & Farkas, C. Insider threats detection using cnn-lstm model. In *2018 International conference on computational science and computational intelligence (CSCI)*, 94–99 (IEEE, 2018).
53. Glasser, J. & Lindauer, B. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*, 98–104 (IEEE, 2013).
54. Shanmugapriya, D., Dhanya, C., Asha, S., Padmavathi, G. & Suthisini, D. N. P. Cloud insider threat detection using deep learning models. In *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, 434–438 (IEEE, 2024).
55. Chattopadhyay, P., Wang, L. & Tan, Y.-P. Scenario-based insider threat detection from cyber activities. *IEEE Trans. Comput. Soc. Syst.* **5**, 660–675 (2018).
56. 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.* **17**, 30–44 (2020).
57. Kancherla, K. & Mukkamala, S. Image visualization based malware detection. In *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 40–44. https://doi.org/10.1109/CICYBS.2013.6597204 (2013).
58. Bhodia, N., Prajapati, P., Di Troia, F. & Stamp, M. Transfer learning for image-based malware classification. arXiv preprint arXiv:1903.11551 (2019).
59. Ferreira, P., C. Le, D. & Zincir-Heywood, N. Exploring feature normalization and temporal information for machine learning based insider threat detection. In *2019 15th International Conference on Network and Service Management (CNSM)*, 1–7, https://doi.org/10.23919/CNSM46954.2019.9012708 (2019).
60. Algburi, R. N. A. et al. Hhlp-ssa: enhanced fault diagnosis in industrial robots using hierarchical hyper-laplacian prior and singular spectrum analysis. In *2024 8th international artificial intelligence and data processing symposium (IDAP)*, 1–8 (IEEE, 2024).
61. Duan, S.-M., Yuan, J.-T. & Wang, B. Contextual feature representation for image-based insider threat classification. *Comput. Secur.* **140**, 103779 (2024).
62. Sharma, O., Sharma, A. & Kalia, A. Windows and iot malware visualization and classification with deep cnn and xception cnn using markov images. *J. Intell. Inf. Syst.* **60**, 349–375 (2023).
63. Deng, H., Guo, C., Shen, G., Cui, Y. & Ping, Y. Mctvd: A malware classification method based on three-channel visualization and deep learning. *Comput. Secur.* **126**, 103084 (2023).
64. Mai, C. et al. Mobilenet-based iot malware detection with opcode features. *J. Commun. Inf. Netw.* **8**, 221–230 (2023).
65. Saqib, S. M. et al. Cataract and glaucoma detection based on transfer learning using mobilenet. *Heliyon* **10**(17), e36759 (2024).
66. Huang, C., Sarabi, M. & Ragab, A. E. Mobilenet-v2/ifho model for accurate detection of early-stage diabetic retinopathy. *Heliyon* **10**, e37293 (2024).
67. Algabri, R. & Choi, M.-T. Robust person following under severe indoor illumination changes for mobile robots: online color-based identification update. In *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 1000–1005 (IEEE, 2021).
68. Vasan, D. et al. Imcfn: Image-based malware classification using fine-tuned convolutional neural network architecture. *Comput. Netw.* **171**, 107138 (2020).

69. Chitrapu, P. & Kalluri, H. K. Mobilenet-powered deep learning for efficient face classification. In *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1–6 (IEEE, 2024).
70. Kumar, M. P., Hasmitha, D., Usha, B., Jyothsna, B. & Sravya, D. Brain tumor classification using mobilenet. In *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, 1–7 (IEEE, 2024).
71. Zhao, J., Shetty, S., Pan, J. W., Kamhoua, C. & Kwiat, K. Transfer learning for detecting unknown network attacks. *EURASIP J. Inf. Secur.* **2019**, 1–13 (2019).
72. Dhanya, K. et al. Obfuscated malware detection in iot android applications using Markov images and cnn. *IEEE Syst. J.* **17**(2), 2756–66 (2023).
73. Yu, Q. & Shi, C. An image classification approach for painting using improved convolutional neural algorithm. *Soft Comput.* **28**, 847–873 (2024).
74. Andrearczyk, V. & Whelan, P. F. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognit. Lett.* **84**, 63–69 (2016).
75. Jiao, J., Liu, Z. & Li, L. Intranet security detection based on image and deep transfer learning. In *Proceedings of the 2023 13th International Conference on Communication and Network Security*, 196–202 (2023).

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2024-00437191) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and by Sejong University Industry-University Cooperation Foundation under the project number (2025-0247-01).

Author contributions

Conceptualization, M.A., F.A., C.Q.L., A.A., Y.H.G., R.A.; methodology, M.A., F.A., C.Q.L., A.A., Y.H.G., R.A.; software, M.A., F.A.; validation, M.A., F.A., C.Q.L., A.A., Y.H.G., R.A.; formal analysis, M.A., F.A., C.Q.L., A.A., Y.H.G., R.A.; investigation, M.A., C.Q.L., A.A.; resources, M.A., H.A.A, A.A.; data curation, M.A., F.A.; writing- original draft preparation, M.A., F.A.; writing-review and editing, C.Q.L., A.A., Y.H.G., R.A.; visualization, M.A., F.A.; supervision, Y.H.G., R.A.; project administration, Y.H.G., R.A.; funding acquisition, Y.H.G., R.A. All authors have read and agreed to the published version of the manuscript. All authors discussed the results and commented on the manuscript.

Funding

This work was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2024-00437191).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.H.G. or R.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025, corrected publication 2025