Annals of Telecommunications
https://doi.org/10.1007/s12243-024-01023-7

E-Watcher: insider threat monitoring and detection for enhanced security

Zhiyuan Wei[1] · **Usman Rauf**[2] · **Fadi Mohsen**[3]

Received: 30 December 2023 / Accepted: 11 March 2024

Abstract
Insider threats refer to harmful actions carried out by authorized users within an organization, posing the most damaging risks. The increasing number of these threats has revealed the inadequacy of traditional methods for detecting and mitigating insider threats. These existing approaches lack the ability to analyze activity-related information in detail, resulting in delayed detection of malicious intent. Additionally, current methods lack advancements in addressing noisy datasets or unknown scenarios, leading to under-fitting or over-fitting of the models. To address these, our paper presents a hybrid insider threat detection framework. We not only enhance prediction accuracy by incorporating a layer of statistical criteria on top of machine learning-based classification but also present optimal parameters to address over/under-fitting of models. We evaluate the performance of our framework using a real-life threat test dataset (CERT r4.2) and compare it to existing methods on the same dataset (Glasser and Lindauer 2013). Our initial evaluation demonstrates that our proposed framework achieves an accuracy of 98.48% in detecting insider threats, surpassing the performance of most of the existing methods. Additionally, our framework effectively handles potential bias and data imbalance issues that can arise in real-life scenarios.

Keywords Behavioral analysis · Information gain · Insider threat detection · Machine learning · Hybrid detection

1 Introduction

In the past two decades, the ultimate objective of the cybersecurity and forensic community has been the real-time detection and mitigation of known and unknown threats without human intervention. Although this desire has contributed to revolutionary advancements in AI and cyber defense, the existing state-of-the-art solutions only address the facet of many current challenges. Most of the work in the last decade was focused on developing efficient monitoring tools, i.e., SIEM/SEM [2]. While these tools excel at monitoring various system elements, they often lack the ability to connect and interpret the evidence cohesively. Therefore, the security administrator/analyst bears the responsibility of assembling the evidence and determining whether any malicious activity is occurring within the organization. The analysts typically rely upon pre-set alarms triggered by the violations of predefined policy. Once a threat is detected, the security analyst has two options: either report it to the authorized authorities (IT department), who can adjust access authorization, or manually implement security policies to address the ongoing threat. However, in many organizations, security analysts only report IT facilities and lack control over fine-tuning access control policies. This creates an additional barrier and time delay that can potentially benefit adversaries [3, 4]. Due to inherent constraints, there is a notable annual increase in the frequency of insider incidents. Recent surveys reveal that approximately 22% of security incidents can be attributed to insiders [2, 5]. Additionally, incidents related to insider threats have surged by 47% between 2018 and 2022. The challenges linked to identifying insider threats result in inevitable delays in mitigation, significantly extending the containment of incidents. As per a 2022 report, the average

✉ Usman Rauf
urauf@mercy.edu

Zhiyuan Wei
zwei@usrobotech.com

Fadi Mohsen
f.f.m.mohsen@rug.nl

[1] Rocky Mountain Robotech, Broomfield, CO, USA

[2] Department of Mathematics & Computer Science, Mercy University, Dobbs Ferry, NY, USA

[3] Bernoulli Institute for Mathematics, Computer Science & Artificial Intelligence, University of Groningen, Groningen, Netherlands

Published online: 04 April 2024

duration for resolving incidents related to insiders has risen from 77 days in 2020 to 85 days in 2022. This increase has led to a 32% rise in the average annual costs for such incidents, reaching $15.4 million [6]. Given the swift pace of digital transformation and the widespread adoption of remote work during and after the pandemic, organizations must urgently address the imperative to mitigate insider threats.

Recently, Machine Learning (ML) based methods have emerged as potential solutions for the aforementioned challenges [3, 7–9]. These methods focus on detecting abnormal activities by analyzing user and system logs. However, a significant requirement for these methods is having a sufficient amount of data, including malicious activities, to train an accurate ML model. The main limitation that has hindered the widespread adoption of ML methods is the difficulty in defining a scenario using diverse information extracted from logs. Existing literature suggests that ML methods perform well when the situation or scenario is known [3], but they struggle to comprehend unknown scenarios by effectively linking various log parameters.

In this paper, we introduce a user-centered hybrid framework called E-Watcher. This framework combines information theory, statistical analysis, and machine learning methods to achieve effective and early threat detection, covering both known and unknown scenarios. The proposed framework emphasizes the detection of behavioral anomalies within an individual's profile rather than relying on a comparison to a standard anomaly set or policy. This approach enables the development of a personalized detection framework that can analyze and monitor each individual in a distinct and isolated manner. Achieving personalized detection using machine learning and information theory is the most fundamental contribution of the proposed E-Watcher (Employee-Watcher) framework.

E-Watcher operates on the principle of the "Divide & Conquer" rule, deviating from the conventional (big data) approach of analyzing extensive aggregated data encompassing logs from numerous employees. Instead, we partition the data of each individual to gain a deeper understanding of their unique patterns and behavior, enabling us to create separate prediction models for each employee. This system is designed to support cybersecurity analysts by minimizing the need for extensive data labeling (unsupervised learning) while effectively detecting both known and unknown threats and providing valuable insights for mitigating insider threats. To demonstrate the efficacy of our proposed framework, we conducted a preliminary evaluation using a real-life threat dataset. Our results illustrate the potential benefits of combining machine learning and statistics-based methods in the field of threat detection.

The preliminary details on this framework can be accessed in previously published work [10], where we introduced the basics of this user-centered hybrid framework, with preliminary evaluation (single known scenario). In this paper, (1) we not only rigorously test our previously proposed framework, on unknown scenarios, (2) we also broaden the scope of our evaluation by testing the resilience and tolerance of E-Watcher against noisy data. (3) In this extension, we also propose a new metric called ***Impact Ratio (IR)*** to examine the effect of noise on the accuracy of classifiers specifically in the case of CERT threat test dataset. (4) Finally, in this extension we also present a rigorous comparison, between E-Watcher and the existing literature in this domain to highlight the significance and effectiveness of our approach.

The remaining sections of this paper are organized as follows: In Sect. 2, we delve into the related contributions from the cybersecurity community in this domain. Section 3 presents our E-Watcher framework and its main components. We discuss the evaluation of our framework and present a detailed evaluation in Sect. 4. Finally, in Sect. 5, we provide concluding remarks and explore potential future extensions of this work.

2 Background and related work

Insider threats can manifest as either intentional or unintentional actions. Intentional insiders engage in malicious activities driven by personal motives such as financial gain or a desire for revenge against the organization. Common motivating factors include financial difficulties, workplace grievances, or the allure of better opportunities elsewhere. Examples of intentional insider actions encompass stealing sensitive data, sabotaging organizational equipment, and causing disruptions. In some instances, intentional insiders may collaborate with external entities to amplify the harm inflicted upon the organization [11, 12].

According to the Ponemon Institute's 2022 report [6], 26% of insider threats are attributed to malicious insiders. Addressing these intentional threats incurs almost 50% higher costs compared to handling unintentional threats. This highlights the considerable impact and challenges posed by insiders with deliberate malicious intent.

Conversely, unintentional insiders are individuals who inadvertently pose a risk to the organization, lacking any malicious intent. Their actions unintentionally expose vulnerabilities to external threats or inadvertently disclose sensitive data to unauthorized entities [13].

Despite the implementation of employee training programs and the deployment of diverse security measures aimed at monitoring and mitigating security threats, insider threats persist as a persistent challenge in organizational environments [11]. Within the realm of research, substantial efforts have been devoted to minimizing response times, transitioning from minutes to seconds. This objective is geared towards swiftly containing the damage caused by both

intentional and unintentional attacks [14]. However, the practical application and demonstration of these solutions have not yet been fully realized, highlighting a gap between theoretical advancements and practical implementation in the ongoing battle against insider threats.

In recent years, the cybersecurity community has dedicated significant efforts towards developing diverse insider threat detection methods. These methods encompass a wide range of approaches, including similarity and distance metric-based identification and machine learning-based identification. In this section, we provide a concise overview of the key and relevant contributions made in this domain.

Recently, there has been a growing trend of utilizing Machine Learning (ML) based methods to enable timely detection of insider attacks. This is made possible by the availability of vast amounts of data that can now be collected and managed at the organizational level. ML offers the advantage of analyzing multi-variable data to identify patterns and gain insights into user behavior [15]. When an abnormal state or activity is detected, alerts are generated and reported to security analysts for further investigation and monitoring. This approach proves beneficial in identifying insider threats as it can effectively detect subtle behavioral changes that human analysts might not readily recognize.

Zhang et al. proposed an insider threat detection method with a self-supervised ensemble learning method and entity embedding [16]. TF-IDF (Term Frequency - Inverse Document Frequency) entity embedding is introduced to determine each entity's importance in every session. The authors use an ensemble learning strategy to address the over-fitting problem caused by legitimate and malicious data imbalance. The sessions dataset is sampled into multiple sub-datasets, which ensures each sub-dataset has enough malicious samples to train. The sub-datasets would then be trained with LSTM-based sub-detectors. To obtain the final malicious score, the malicious scores of the sub-datasets are averaged. The experimental results demonstrate that this method achieves a 99.2% AUC (Area Under the Curve) when detecting malicious sessions on CERT4.2 datasets [17]. However, its performance decreases to 95.3% on highly imbalanced datasets such as CERT6.2 [18]. This suggests that there is room for improvement and highlights the need for future work in addressing the challenges posed by imbalanced datasets.

Duc. Le. proposed a machine learning-based framework for user-centered insider threat detection [19]. The authors rely on extracting user activity data from the raw logs and preprocessing to extract numerical features with different temporal representations (e.g., daily or weekly timelines). Once the features are extracted, they utilize unsupervised machine learning for anomaly detection to assign anomaly scores to filter the activities/events. The anomaly scores are compared with a predefined threshold calculated by a user-selected percentage of data to decide whether the data sample is suspicious. Instead of testing on self-generated data, the authors rely on the CERT 4.2 dataset for evaluation. The authors' proposed method, which is mainly similar to other methods, achieved AUC scores of 90.7% and 90.9% when evaluated on weekly and daily data, respectively. However, they did not provide information regarding the effectiveness of their method on a monthly time frame.

Taher et al. proposed an ML-based model for detecting insider data leakage, aiming to address issues related to bias and data imbalance [20]. To achieve a balanced dataset, the authors employed a synthetic minority oversampling technique (SMOTE), which introduced artificial data points in the minority class. Categorical data was transformed into machine-readable data using label encoding and one-hot encoding. The study evaluated the performance of five different ML algorithms for anomaly detection in effectively identifying malicious sessions utilizing the CERT4.2 dataset. However, it is worth noting that no significant novelty was observed compared to previously presented approaches in the same domain.

Wei et al. introduced an innovative unsupervised anomaly detection based on cascaded autoencoders (CAEs) and joint optimization networks [21]. The authors utilized a Bidirectional Long Short-Term Memory (BiLSTM) feature extractor to extract features, which were then passed through a Corrective Auto-encoder and purification-based joint optimization scheme (CPJOS). CPJOS helped filter out normal and suspicious samples. To mitigate the high false positive rate resulting from data drop in CPJOS, a hypergraph correction model was employed. The experimental results demonstrated superior performance compared to current state-of-the-art techniques. However, it is important to note that the proposed method focuses solely on identifying the relationship between temporal actions and does not incorporate any contextual notion.

Improving data acquisition and processing is crucial to unleashing the full potential of Machine Learning and overcoming its limitations. Consequently, researchers have dedicated their efforts to developing data processing techniques as a means to address this challenge.

Jiang et al. proposed the utilization of big data analytics for user behavior analysis in order to detect insider threats over an extended period of time [22]. Their approach involved performing statistic-based feature extraction and preprocessing using the Spark analytical engine. The data was divided into six categories of features, aggregated based on time, and the XGBoost algorithm was employed for detection. To address any imbalances and biases in the dataset, the authors also employed the SMOTE algorithm. Although the experimental results showcased an impressive accuracy of 99.13% using the XGBoost algorithm on the CERT r6.2 dataset, the article

did not specifically emphasize or elaborate on the ability of big data analytics to deduce or infer information related to unseen scenarios or attacks.

In the literature, supervised machine learning methods have been put forth as solutions to address insider threat challenges. One approach proposed by Yuan et al. involves utilizing Deep Learning-based Neural Networks (DNNs) for detection [9]. In this method, the daily actions of each user are separated, and the temporal information is abstracted. Experimental evaluations conducted on the CERT 4.2 dataset indicate that this approach can achieve an AUC of 94.49% in the best-case scenario. However, it is important to note that the critical limitation of using supervised learning methods is their heavy reliance on mapping inputs to predefined outputs, as the data must be labeled for supervised methods to be leveraged. As a result, these methods may struggle to handle unknown scenarios.

Koutsouvelis et al. proposed a similar approach that utilizes supervised machine learning, specifically Convolution Neural Networks (CNNs), for insider threat detection [23]. In contrast to traditional methods that focus on tuning access control policies, the authors adopted a visual analysis approach. They convert employees' activities into images using activity data and compare them with pre-constructed and labeled images generated from normal activities. These newly constructed images are then inputted into a Google TensorFlow-based CNN algorithm to recognize and classify whether the activity pattern represented by the image is regular or malicious. The approach achieves a validation accuracy of 90.6%. However, similar to other supervised learning-based methods discussed earlier, this approach faces limitations. The lack of labeled activity data and labeling accuracy restricts its potential to identify unknown threats.

Rastogi et al. proposed using LSTM-based recurrent neural networks to detect insider threats [24]. Instead of directly utilizing the data generated by users' activities, the authors encode temporal activities as a sequence of events and assign them a key value. These activities over a specific period of time result in the generation of key-based sequences. Once the contextual pattern of normal key sequences is established, the authors compare newly generated instances with the baseline to identify potential threats. Although the approach achieves a prediction accuracy of 93%, it exhibits several limitations and necessitates a comprehensive performance evaluation. Encoding key-based sequences for many events in the network requires substantial computing resources and incurs significant overhead, making it impractical for real-time threat mitigation.

Rauf et al. introduced a unique approach inspired by nature to address the problem of insider threats by integrating concepts from DNA regulation [4]. Building upon prior work, the authors formalized and proposed the optimization of access control or security policies as a means to counter insider threats [3, 14]. Their proposed process begins with aggregating data from various logs based on timestamps. The authors then employ the encoding method known as One-hot encoding to construct activity vectors for each employee. To assess an employee's behavioral deviation, a distance-based metric called Behavioral Analytic Metric (BAM) is introduced to measure the variation of each newly generated activity vector from the user's historical profile. Furthermore, the authors propose that the data related to these behavioral profiles can be used to train machine learning models, and the learned parameters can be stored in an exporting format. These parameters can then be readily utilized by automated security policy tuning systems, eliminating the need for manual policy enforcement by human analysts or IT personnel. Experimental evaluations demonstrate that with optimized parameters, the authors achieved an accuracy of 98% on the CERT threat test dataset (Table 1).

While the research on insider threat detection is extensive, we cannot cover all the details within the confines of this document. Therefore, we recommend referring to a comprehensive and recently published literature review on insider threat detection for further information and in-depth analysis [2].

3 Proposed framework

In this section, we present the specifications of our proposed hybrid E-Watcher framework and discuss its components one by one in detail. Figure 1 presents a detailed description of our proposed framework. E-Watcher is based on three main modules, each performing a specific task.

- **Data Collection Module:** Responsible for time-interleaving-based aggregation and user-based segregation of data
- **Feature Engineering Module:** Responsible for contextual and user-based filtering, along with noise and bias removal
- **Hybrid Anomaly Checker:** Responsible for threshold and parametric-based classification of activity samples

The above-mentioned modules are the backbone of our initial E-Watcher design to mitigate threats using a hybrid approach. We will discuss these components one by one in detail.

3.1 Data collection module

The primary function of this module is to construct a knowledge base by collecting activity-related data, which will be utilized to identify and understand normal and abnormal patterns or behaviors. To achieve this goal, we propose collecting system and user-related logs from various sources, such as

Table 1 Related work review summary

Author	Method	Learning method	Dataset	Results	Type	Balancing
Zhang et al. [16]	LSTM-based sub-detectors	Self-supervised	CERT r4.2 /r6.2	99.2%/ 95.3% AUC	Anomaly detection	bagging algorithm-based ensemble Learning strategy
Le.D [19]	Unsupervised Ensemble	Unsupervised	CERT r4.2	90.7/90.9% AUC	Anomaly detection	–
Taher et al. [20]	Decision Tree + Random Forest	Supervised	CERT r4.2	100% AUC	Classification	SMOTE
Wei et al. [21]	Cascaded Autoencoders	Self-supervised	CERT r6.2	93.2% AUC	Anomaly detection	–
Jiang et al. [22]	XGBoost	Supervised	CERT r6.2	99.13% Acc	Anomaly detection	SMOTE
Yuan et al. [9]	LSTM-CNN	Supervised	CERT r4.2	94.49% AUC	Classification	–
Koutsouvelis et al. [23]	CNN	Supervised	CERT r6.2	90.6% Acc	Classification	–
Rastogi et al. [24]	RNN-LSTM	Self-supervised	CERT r5.2	93.29% Acc	Anomaly detection	–
Rauf et al. [3]	Random Forest	Unsupervised	CERT r4.2	98% Acc	Classification	–

network devices, system activities, and user activities. These logs can be obtained using basic shell commands like ***tail -f *.log*** or using a Security Information and Event Management (SIEM) tool like SPLUNK. By gathering this data, we establish a foundation for analyzing and learning from the activities within the system.

After collecting and storing all the logs in a sub-unit called log collection, our proposed approach combines these logs into a single file using a time-interleaving-based merging technique. This merging process allows us to consolidate data from multiple sources, including all the activities associated with different users. By consolidating the data into a

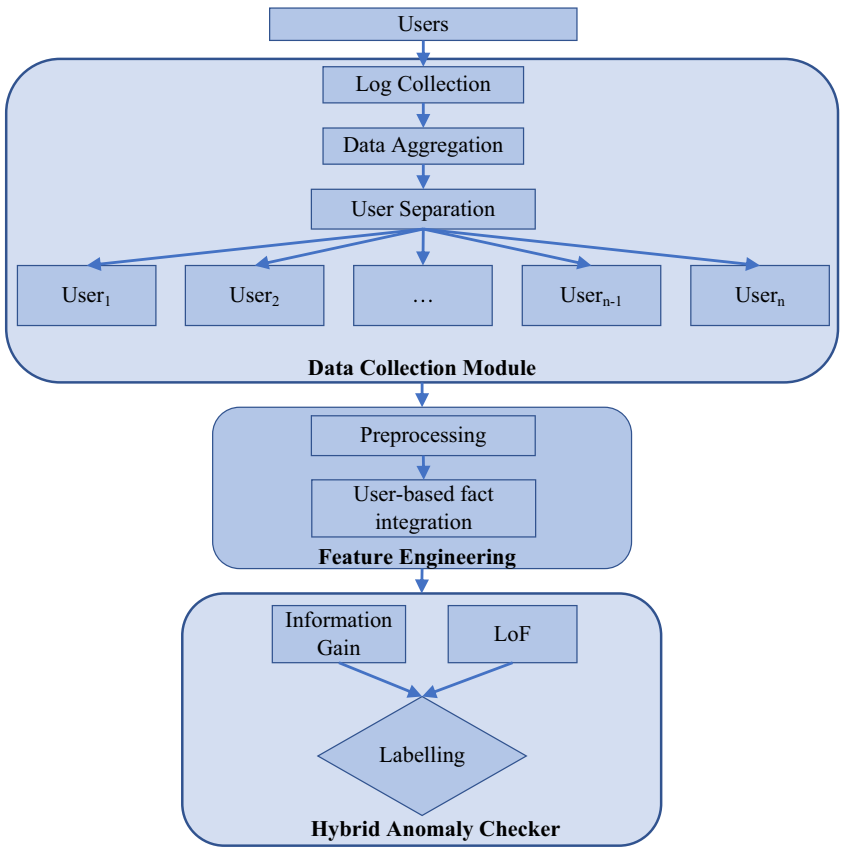


Fig. 1 Overview of the proposed framework

single file, we simplify the extraction of user-related information, eliminating the need to extract data from multiple log files. This approach is particularly advantageous when dealing with many employees, as it saves time and effort for analysts. Figure 2(a) provides a visual representation of an aggregated dataset that contains time-interleaved data samples of all the user-related activities over a specific period.

In the subsequent phase, our approach involves segregating the data of individual users based on specific requirements, i.e., only for the employees who are working on sensitive projects, so that their behavioral profiles need to be constructed for behavioral anomaly detection. Figure 2(b) illustrates a snapshot of a single user's data, which has been extracted from the aggregated logs containing data from all users.

3.2 Feature engineering module

The data obtained from the previous step is unstructured and cannot be directly processed by machine learning algorithms. In our approach, this unstructured data includes timestamps and variable values such as URLs. To make the data machine-readable and assign context to it, encoding is necessary. For example, timestamps must be transformed into meaningful date and/or time representations or periodic variables. Similarly, URLs need to be categorized as low or high risk to provide interpretation for the machine learning algorithm regarding the nature of the visited URLs.

During the preprocessing phase, we employ the One-HotEncoding technique to convert the unstructured user-related data into a structured format. Additionally, we apply variance threshold-based filters to identify and exclude attributes/variables that lack sufficient variance or do not contribute valuable information for machine learning methods. This step helps streamline the dataset and improve the efficiency of subsequent analysis. A snapshot of the structured data after this preprocessing phase is illustrated in Fig. 2(c).

3.3 Hybrid behavioral checker

The final and crucial component of the E-Watcher framework is the hybrid anomaly checker, which combines statistical methods and ML-based classification to determine whether an individual's behavior is normal or not. Since each individual has distinct work patterns and behaviors, their normal behavioral patterns cannot be easily measured by considering the entire dataset as a whole. Thus, in this step, we treat the individual users' datasets, which were segregated

	id	date	user	pc	device_activity	logon_activity	url	content
0	{J1...	01/02/20...	MOH0273	PC-6699	Connect	NaN	NaN	NaN
1	{N7...	01/02/20...	MOH0273	PC-6699	Disconnect	NaN	NaN	NaN
2	{U1...	01/02/20...	HPH0075	PC-2417	Connect	NaN	NaN	NaN
3	{H0...	01/02/20...	IIW0249	PC-0843	Connect	NaN	NaN	NaN
4	{L7...	01/02/20...	IIW0249	PC-0843	Disconnect	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
28434418	{J1...	05/16/20...	BRM0995	PC-0768	NaN	NaN	ht...	success...
28434419	{I5...	05/16/20...	BRM0995	PC-0768	NaN	NaN	ht...	below 1...

(a) Aggregation of Log Files

	id	date	user	pc	device_activity	logon_activity	url	content
0	{K9P7-J8WZ79GZ-1137FRBP}	2010-01-04 02:32:51	CSC0217	PC-3742	NaN	Logon	NaN	
1	{S5I3-G2FS33LA-7352MTZZ}	2010-01-04 02:37:02	CSC0217	PC-3742	NaN	Logoff	NaN	
2	{W7L2-H6ND52HZ-8917XDYV}	2010-01-04 07:24:00	CSC0217	PC-6377	NaN	Logon	NaN	
3	{W8V3-S5NA59BW-8974QJMB}	2010-01-04 07:28:30	CSC0217	PC-6377	NaN	NaN	http://mega...	is comple
4	{Z7A1-C8XY80LS-4273MQAI}	2010-01-04 07:43:42	CSC0217	PC-6377	NaN	NaN	http://sfga...	mats cont
...	...	...	...	...	...	...	...	

(b) Segregation of Individual Users' Activities

	pc	device_activity	logon_activity	url	content
0	0	0	1	0	0
1	0	0	0	0	0
2	1	0	1	0	0
3	1	0	0	0	0
4	1	0	0	0	0
...	...	...	...	...	...

(c) Resultant dataset after data preprocessing

Fig. 2 Data preprocessing and feature engineering

in the previous step, separately and individually. By analyzing each user's data in isolation, we can capture their unique behavioral patterns and establish personalized profiles. This approach enables a more accurate assessment of deviations from normal behavior for each user. We employ the entropy-based information gain metric with an added variable threshold (as the statistical method) and the unsupervised classification method Local Outlier Factor (LoF) as the ML-based method. To classify a sample as an anomaly, both methods must label a sample as abnormal after parametric and threshold tuning in order for it to qualify as an anomalous incident, as illustrated in Fig. 3.

The reason for using two different methods lies in their contextual design. LoF operates like a distance-based metric, where parameters such as the nearest neighbor count can influence whether a sample is grouped into one class or another. On the other hand, entropy-based analysis allows us to define and control filtering thresholds based on the user's profile, which can be adjusted and fine-tuned based on organizational needs. Unlike ML-based methods, it evaluates the relevance of an individual sample independently, regardless of its proximity to other samples. This flexibility empowers analysts to define their own statistical metrics based on their specific requirements.

3.3.1 Information gain based labeling

Information gain measures how much information a sample carries within the data context. Entropy quantifies how much information there is in a variable, specifically its relative probability distribution. A skewed distribution has a low entropy, whereas a distribution where events have equal probability has a larger entropy [25]. The information gain is computed using the Shannon entropy [26] as:

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

In the entropy-based analysis, each activity vector (corresponding to a row of the structured and preprocessed data) is assigned an entropy value. The scaled standard deviation, calculated as $\delta \times \sigma = \delta \times \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$, is then used as a threshold to determine the proximity of a sample to other samples. Here, $\delta \mapsto 1,3 \in \mathbb{R}$. This approach allows us to establish threshold-based criteria for labeling or classifying activity vectors. The rationale behind selecting this threshold is based on the statistical, empirical rule of 3-σ, which suggests that normal data samples should generally fall within the range of 3 times the standard deviation ($3\times\sigma$) from the mean of the distribution.

3.3.2 Unsupervised learning for outlier detection

The second method we utilize for identifying abnormal samples or activities is the Local Outlier Factor (LOF) algorithm as a machine learning approach. This algorithm assesses the distance between a data sample and its neighboring points, identifying samples that have significantly lower density compared to their neighbors [27]. During the classification process, the outlier factor and label are assigned to each data point. The Local Outlier Factor is computed using the following formula:

$$LOF_k(A) = \frac{\sum_B \epsilon N_k(A)^{lrd_k(B)}}{|N_k(A)| \cdot lrd_k(A)}$$

In this equation, $N_k(A)$ represents the number of nearest neighbors of point A, and $lrd_k(B)$ represents the inverse of the average reachability distance of point B from its neighbors. If the value of the outlier factor is significantly greater than 1, it indicates that the considered point has a much lower local reachability compared to its neighbors. In such cases, the label assigned to the data point would be "-1," indicating that it is an outlier or abnormal point. The LOF algorithm is known for its ability to detect outliers that may be overlooked by other algorithms. However, it should be noted that

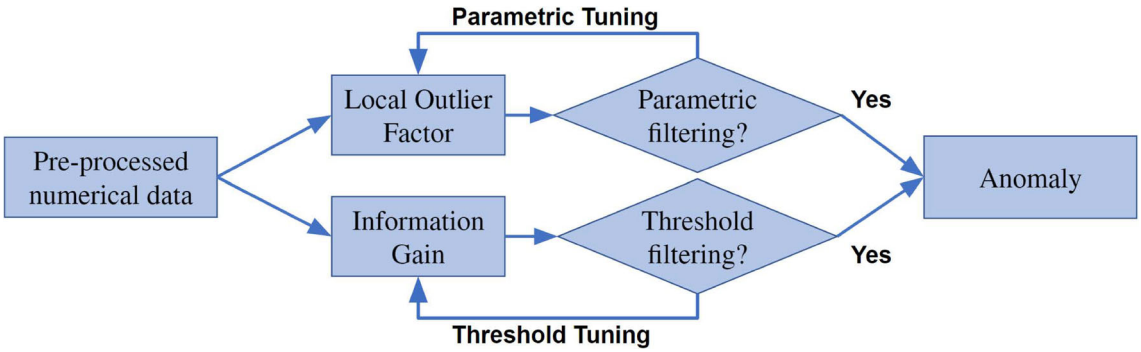


Fig. 3 Detailed overview of the proposed hybrid behavioral checker

the LOF algorithm can have a higher runtime when dealing with larger datasets [28].

3.3.3 Hybridization for better control and accuracy

In the last step, towards achieving the better of both worlds, we extract the shared results from Information Gain and Local Outlier Factor to find an intersection between the labeled samples. Only those samples are labeled as abnormal/anomalous if they appear abnormal in both results. We strongly believe that ensemble anomaly detection methods with different working principles improve the accuracy of anomaly detection because they have different working principles and can complement each other for better detection and control (Fig. 3).

4 Evaluation

In this section, we provide a detailed description of the experimental setup, including the datasets used and the performance metrics employed to evaluate the proposed E-Watcher framework. Additionally, we investigate the effect of threshold and parametric tuning on the accuracy of anomaly labeling, shedding light on the optimization process for achieving improved detection outcomes.

4.1 Experimental setup

We conducted our experiments using the CERT Threat Test dataset [1], which is commonly used in previous studies for evaluating insider threat detection approaches. The dataset consists of information from 1000 employees and covers five different types of insider threat scenarios. However, due to space limitations and the focus of this paper, we specifically focused on evaluating our framework using these scenarios which involve:

- ***Scenario 1:*** *A user who exhibits unusual behavior such as logging in after hours, using removable drives, uploading data to wikileaks.org, and subsequently leaving the organization.*
- ***Scenario 2:*** *The system administrator becomes disgruntled. He downloads a key logger and uses a thumb drive to transfer it to his supervisor's machine. The next day, he uses the collected key logs to log in as his supervisor and send out an alarming mass email, causing panic in the organization. He leaves the organization immediately.*

To evaluate these scenarios, our primary objective is to define criteria and effectively classify behavioral anomalies from normal activity vectors for a specific user. To achieve this, we selected the user with the highest frequency of data samples, namely DTAA-KEE0997 and CSC0217, to establish their profile. All the experiments were conducted using Python 3.8 and the Scikit-learn framework on a Macbook Pro M1 Max with 64 GB of memory for evaluation purposes.

4.2 Effect of threshold and parametric tuning on accuracy of labeling

In this section, we assess how altering the threshold and parametric values can affect classification accuracy. Towards this objective, we test different values of δ (for information gain), nearest neighbors parameter, and contamination values (in the case of LoF).

We vary the value of δ within the range of $\{2, 3\} \in \mathbb{R}$ and observe its impact on the classification. Figure 4(a) demonstrates that when $\delta = 2$, there is an increase in false positives. However, when we adjust the delta value to fall within the range of {2.5, 3}, Fig. 4(b) and (c) illustrate that we are able to accurately identify the behavioral anomaly cluster for scenario 1, which exhibits significantly higher entropy values compared to the other samples. In scenario 2, Fig. 5 illustrates that when the delta value varies within the range of {2, 3}, there are no alterations in clusters. When we adjust the delta value within the range of {2, 3}, Fig. 5(a), (b) and (c) illustrate that the behavioral anomaly cluster is accurately identified for scenario 2, where the clusters maintain distinct

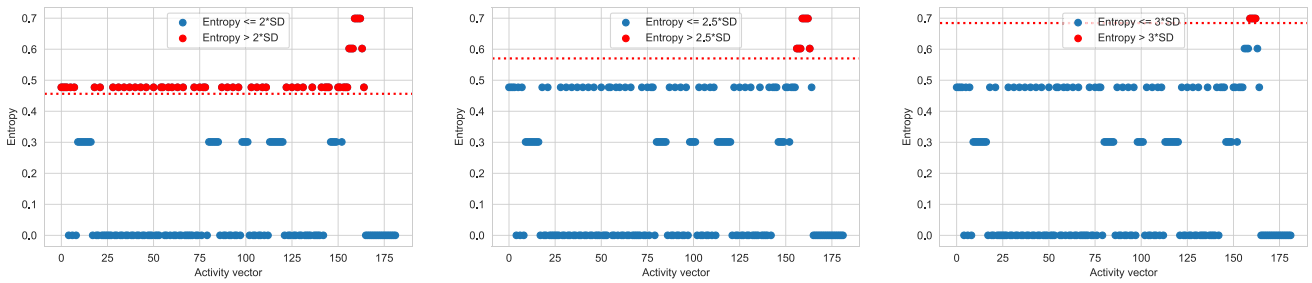


Fig. 4 Effect of the threshold value (δ) on the accuracy of classification: Scenario 1

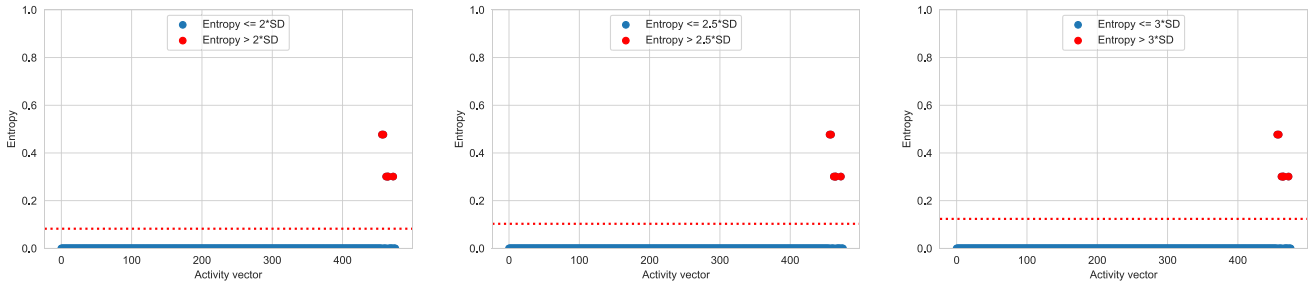


Fig. 5 Effect of the threshold value (δ) on the accuracy of classification: Scenario 2

labels with high sensitivity, resulting in a reduction of false negatives.

In the context of parametric tuning while using LoF (Local Outlier Factor), two critical parameters affecting classification are the number of nearest neighbors and the contamination value. The number of nearest neighbors specifies how many neighbors to identify for each sample, while the contamination parameter defines the proportion of outliers in the dataset (as an assumption for the algorithm).

In the first experiment, for scenarios 1 and 2, we vary the number of nearest neighbors and observe its impact on classification. Figures 6 and 8 depict the effect of changing the value of nearest neighbors on labeling accuracy and Fig. 8 shows the effect of changing nearest neighbors' value on the labeling accuracy (Fig. 7). In these figures, normal activity vectors are labeled as cluster "1," while abnormal activity vectors are labeled as cluster "-1." The results from Fig. 6 indicate that the classification accuracy remains consistent even with variations in the number of nearest neighbors (ranging from 5 to 15). However, Fig. 8 reveals that the classification accuracy changes with alterations in the number of nearest neighbor parameters. Although in this case (scenario 2), assuming the number of neighbors increases the specificity of the results, hence leading to a higher number of false positives by an additional 3%, the overall accuracy remains around 97%.

In the second experiment, for scenarios 1 and 2, we changed the contamination value from 0.01 (representing that 1% samples are anomalous) to 0.2 (20% samples are anomalous). Under different assumptions, the model should generate the same results if it has the tendency to capture anomalies. Figures 7 and 9 show that the change of contamination has a minute effect on the accuracy of scenario 2, but for scenario 1 we achieve 96.1% accuracy in the worst-case scenario and 99.7% accuracy in the best-case scenario. The false positive rate decreases, which results in an increase in accuracy as the contamination value is decreased. Therefore, it is better to keep the contamination value between 0.01 and 0.1 to avoid over-approximation. We also control this over-approximation and false positive rate by combining LoF with information gain, which eventually makes our proposed solution more tolerant to high noise in the data as well.

4.3 Effectiveness against noise

The accuracy of predictions in supervised learning is heavily dependent on the quality of the labels associated with the training data [29]. Accurate labeling ensures that the model learns meaningful patterns and relationships within the data. Conversely, if the labeled data used for training contains errors or noise, it can introduce confusion and result in the

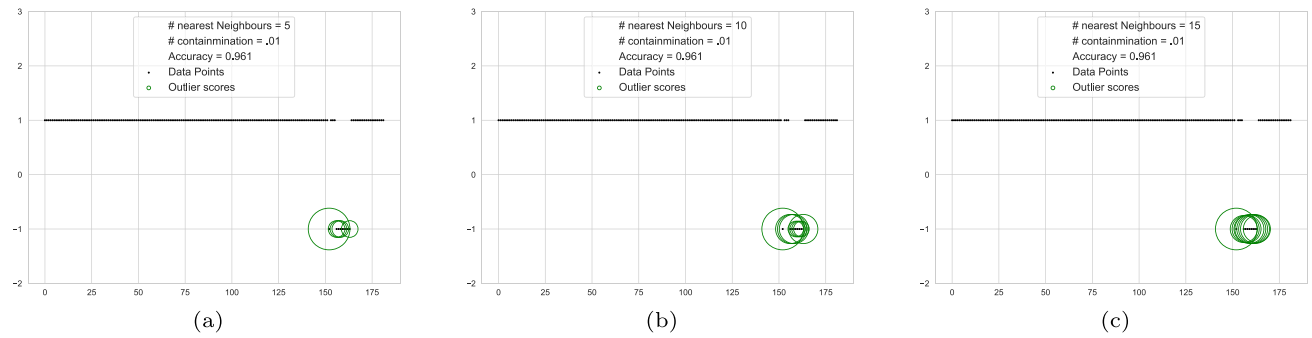


Fig. 6 Effect of number of nearest neighbors parameter on the accuracy of classification: Scenario 1

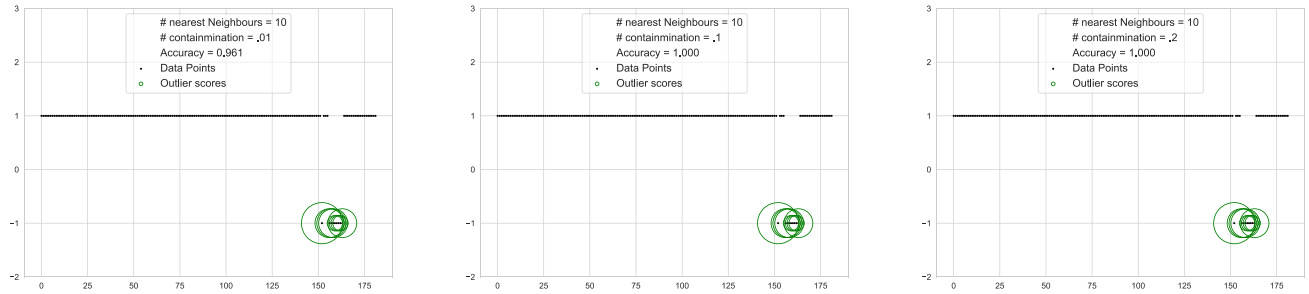
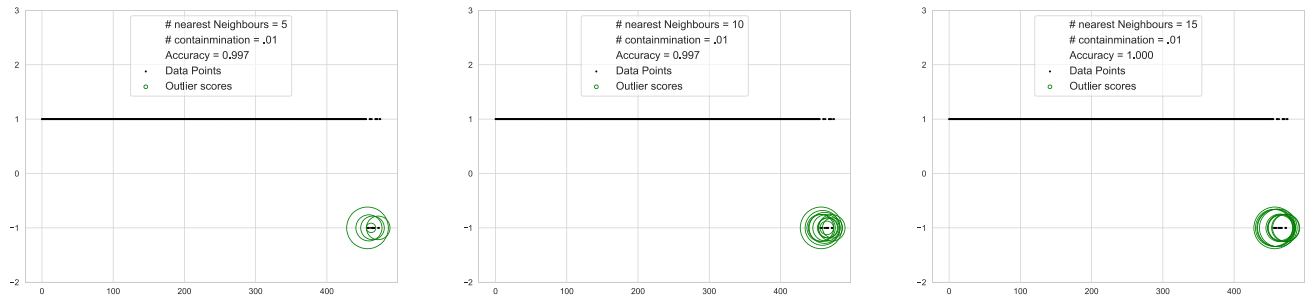
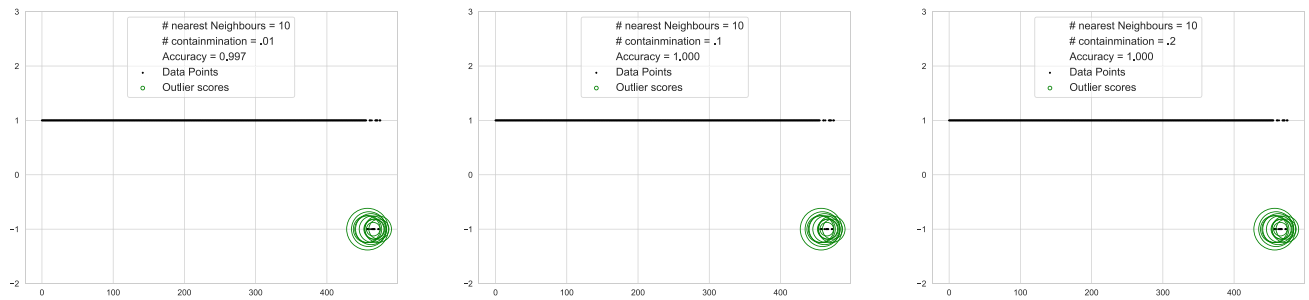


Fig. 7 Effect of data contamination on the accuracy of classification: Scenario 1



(a) (b) (c)

Fig. 8 Effect of number of nearest neighbors parameter on the accuracy of classification: Scenario 2



(a) (b) (c)

Fig. 9 Effect of data contamination on the accuracy of classification: Scenario 2

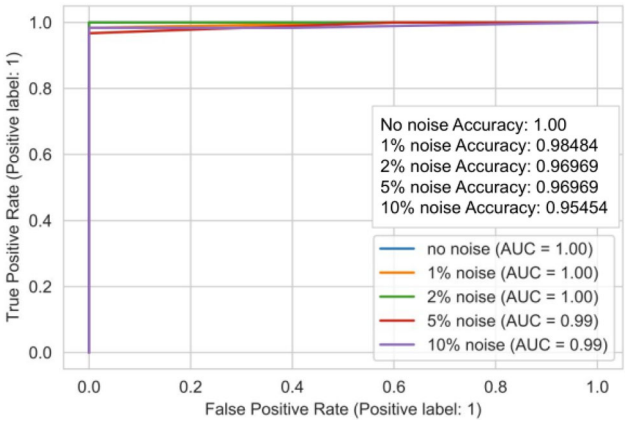


Fig. 10 Effectiveness of E-Watcher against noise: Roc curves for Scenario 1

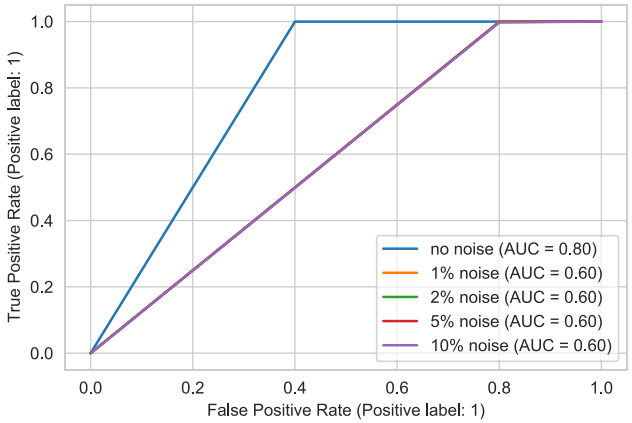


Fig. 11 Effectiveness of E-Watcher against noise: Roc curves for Scenario 2

derivation of incorrect labels [30]. This underscores the critical importance of meticulously curated and accurate labels in the training process to achieve reliable and effective supervised machine learning models.

To test the effectiveness of our proposed method against insiders, we add noise to the data and use a supervised machine learning method (Random Forest) on our labeled/clustered data to assess accuracy using RoC curves. We achieved a 95% accuracy in the worst-case under scenario 1, when 10% noise is added to the dataset, indicating a high level of resilience against noisy datasets/scenarios (c.f. Fig. 10).

For Scenario 2, the accuracy drops significantly with the addition of noise (c.f. Fig. 11). We attribute this result to under-fitting caused by the limited sample size, the presence of noise, and the imbalanced Impact Ratio (IR) between malicious samples and total samples in a scenario, which consequently leads to a large change in accuracy [31]. Impact ratio can be defined as:

$$\mathbf{IR} = \frac{\textbf{Number of Malicious Samples}}{\textbf{Total Number of Samples}}$$

For scenario 1, the value of $IR = 0.045$ (4.5%), whereas in scenario 2, the value of $IR = 0.01$ (1%), which leads to under-fitting. In conclusion, for our approach to work and avoid under-fitting, we conclude that the optimal value of IR should be above 3% ($IR \geq 0.03$).

4.4 Comparative analysis

In order to offer a thorough evaluation of our methodology, we conducted a comparative analysis specifically focused on studies involving the CERT r4.2 dataset. This deliberate selection was motivated by the fact that the chosen related work addresses the same scenario within the CERT r4.2 dataset, closely aligning with the primary focus of our study. This comparative analysis underscores the effectiveness of our proposed approach. The details of the comparison are discussed in Table 2. From the evaluation results, and comparison, we can confidently state, that our approach not only outperforms most of the methods, but also provides a resilient platform to deal with datasets containing high noise.

Table 2 Comparison with related work focused on CERT r4.2 dataset

Author	Accuracy	F1 Score	Recall	Precision
Gaval et al. (2015) [32]	N/A	N/A	76%	N/A
Aldairi et al. (2019) [33]	93%	N/A	92%	92%
Koutsouvelis et al. (2020) [23]	90.6%	N/A	N/A	N/A
Rastogi et al. (2020) [24]	93.29%	95%	N/A	N/A
Gayathri et al. (2020) [34]	N/A	N/A	99.32%	99.32%
Rauf et al. (2021) [3]	98%	N/A	N/A	N/A
Nicolaou et al. (2021) [35]	N/A	32.86%	82.85%	14.6%
Pantelidis et al. (2021) [36]	92%	96%	96%	94%
Le et al. (2021) [37]	99.73%	N/A	N/A	99.3%
Our method	98.48%	99.23%	98.48%	100%

5 Conclusion

In this paper, we introduced a novel hybrid threat detection framework called Employee-Watcher. This framework uses statistical methods for supervision over machine learning to enhance the accuracy of insider threat detection. Our detailed evaluation demonstrates that our approach outperforms relevant methods on a comparable dataset, highlighting the effectiveness of our hybrid framework in achieving improved accuracy. We also evaluate our approach against benign noise in the dataset, which leads to higher false positives, or false negatives. Our approach shows significant resilient against noisy dataset, given that $IR \geq 0.03$.

In future our main objective is to develop employee-specific trained models that can be easily exported and accessed on demand for threat intelligence. By implementing a client-server-based architecture, these analytical models can be deployed as an API (that is available around the clock), enabling remote clients to query the cloud-based threat detection module. This architecture will not only facilitate the practicality of deploying the threat intelligence API but also allow for the measurement and evaluation of its deployment effectiveness.

Author Contributions U.R and F.M developed the concepts of the paper, and equally contributed towards writing the main manuscript. Z.W conducted the implementation, and helped in writing as well.

Funding This research was funded by a National Centers of Academic Excellence in Cybersecurity grant (H98230-22-1-0256), which is part of the National Security Agency (NSA). Any findings, conclusions, or recommendations expressed in this research are those of author(s) and do not necessarily reflect the views of the sponsor.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Glasser J, Lindauer B (2013) Bridging the gap: a pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops, pp 98–104. https://doi.org/10.1109/SPW.2013.37
2. Rauf U, Mohsen F, Wei Z (2023) A taxonomic classification of insider threats: existing techniques, future directions & recommendations. J Cyber Secur Mobil. https://doi.org/10.13052/jcsm2245-1439.1225
3. Rauf U, Shehab M, Qamar N, Sameen S (2021) Formal approach to thwart against insider attacks: a bio-inspired auto-resilient policy regulation framework. Future Gener Comput Syst 117:412–425. https://doi.org/10.1016/j.future.2020.11.009
4. Rauf U, Shehab M, Qamar N, Sameen S (2019) Bio-inspired approach to thwart against insider threats: an access control policy regulation framework. In: Bio-inspired information and communication technologies. Springer, Cham, pp 39–57. https://doi.org/10.1007/978-3-030-24202-2_4
5. Verizon (2021) 2021 data breach investigations report. Tech Rep. https://www.verizon.com/business/resources/reports/2021/2021-data-breach-investigations-report.pdf
6. Ponemon Institute (2022) 2022 cost of insider threats global report. Tech Rep. https://www.proofpoint.com/us/resources/threat-reports/cost-of-insider-threats
7. Brdiczka O, Liu J, Price B, Shen J, Patil A, Chow R, Bart E, Ducheneaut N (2012) Proactive insider threat detection through graph learning and psychological context. In: Security and Privacy Workshops (SPW), 2012 IEEE Symposium On, pp 142–149. https://doi.org/10.1109/SPW.2012.29
8. Kim J, Park M, Kim H, Cho S, Kang P (2019) Insider threat detection based on user behavior modeling and anomaly detection algorithms. Appl Sci 9(19). https://doi.org/10.3390/app9194018
9. Yuan F, Cao Y, Shang Y, Liu Y, Tan J, Fang B (2018) Insider threat detection with deep neural network. In: Computational Science – ICCS 2018. Springer, Cham, pp 43–54. https://doi.org/10.1007/978-3-319-93698-7_4
10. Rauf U, Wei Z, Mohsen F (2023) Employee watcher: a machine learning-based hybrid insider threat detection framework. In: 2023 7th Cyber Security in Networking Conference (CSNet), pp 39–45. https://doi.org/10.1109/CSNet59123.2023.10339777
11. Cybersecurity Agency IS (2022) Insider threat mitigation guide. https://www.cisa.gov/insider-threat-mitigation
12. Cybersecurity Insiders (2020) 2020 insider threat report. Techn Rep. https://www.cybersecurity-insiders.com/portfolio/2020-insider-threat-report-gurucul/
13. Schoenherr JR, Lilja-Lolax K, Gioe D (2022) Multiple approach paths to insider threat (map-it): Intentional, ambivalent and unintentional insider threats. Counter-Insider Threat Research and Practice 1(1)
14. Rauf U (2020) Bio-inspired cyber security and threat analytics. PhD thesis, The University of North Carolina at Charlotte
15. Sarker IH (2021) Machine learning: algorithms, real-world applications and research directions. SN Comput Sci 2(160). https://doi.org/10.1007/s42979-021-00592-x
16. Chunrui Z, Shen W, Dechen Z, Tingyue Y, Tiangang W, Mingyong Y (2021) Detecting insider threat from behavioral logs based on ensemble and self-supervised learning. Secur Commun Netw 2021(4148441). https://doi.org/10.1155/2021/414844
17. Lindauer B (2020) Insider threat test dataset. Carnegie Mellon University, Pittsburgh, PA. https://doi.org/10.1184/R1/12841247.v1
18. CERT Threat Test Dataset (2016). https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

19. Le DC (2021) A machine learning based framework for user-centered insider threat detection. PhD thesis, Dalhousie University. https://dalspace.library.dal.ca/bitstream/handle/10222/80731/DucLe2021.pdf?sequence=1
20. Al-Shehari T, Alsowail RA (2021) An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. Entropy 23(10):1258. https://doi.org/10.3390/e23101258
21. Wei Y, Chow K-P, Yiu S-M (2021) Insider threat prediction based on unsupervised anomaly detection scheme for proactive forensic investigation. Forensic Sci Int Digit Investig 38:301126. https://doi.org/10.1016/j.fsidi.2021.301126
22. Jiang W, Tian Y, Liu W, Liu W (2018) An insider threat detection method based on user behavior analysis. In: 10th International conference on intelligent information processing (IIP). Intelligent Information Processing IX, vol AICT-538, Nanning, China, pp 421–429. https://doi.org/10.1007/978-3-030-00828-4_43. Part 10: Image Understanding
23. Koutsouvelis V, Shiaeles S, Ghita B, Bendiab G (2020) Detection of insider threats using artificial intelligence and visualisation. In: 2020 6th IEEE Conference on Network Softwarization (NetSoft), pp 437–443. https://doi.org/10.1109/NetSoft48620.2020.9165337
24. Ma Q, Rastogi N (2020) Dante: predicting insider threat using lstm on system logs. https://doi.org/10.1109/TrustCom50675.2020.00153
25. Kurniabudi, Stiawan D, Darmawijoyo, Bin Idris, MY, Bamhdi AM, Budiarto R (2020) Cicids-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access 8:132911–132921. https://doi.org/10.1109/ACCESS.2020.3009843
26. Vajapeyam S (2014) Understanding shannon's entropy metric for information. https://doi.org/10.48550/ARXIV.1405.2061
27. Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00, New York, NY, USA, pp 93–104. https://doi.org/10.1145/342009.335388
28. Campos GO, Zimek A, Sander J, Campello RJGB, Micenková B, Schubert E, Assent I, Houle ME (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min Knowl Disc 30(7):891–927. https://doi.org/10.1109/JIOT.2019.2958185
29. IBM Cloud Education (2020) What is Supervised Learning? https://www.ibm.com/cloud/learn/supervised-learning
30. Gupta S, Gupta A (2019) Dealing with noise problem in machine learning data-sets: a systematic review. Procedia Comput Sci 161:466–474. https://doi.org/10.1016/j.procs.2019.11.146
31. IBM Cloud Education (2021) What is Overfitting? https://www.ibm.com/cloud/learn/overfitting
32. Gavai G, Sricharan K, Gunning D, Hanley J, Singhal M, Rolleston R (2015) Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data. 6:47–63. https://doi.org/10.22667/JOWUA.2015.12.31.047
33. Aldairi M, Karimi L, Joshi J (2019) A trust aware unsupervised learning approach for insider threat detection. In: 2019 IEEE 20th International conference on information reuse and integration for data science (IRI), pp 89–98. https://doi.org/10.1109/IRI.2019.00027
34. Gayathri RG, Sajjanhar A, Xiang Y (2020) Image-based feature representation for insider threat classification. Appl Sci 10(14):4945. https://doi.org/10.3390/app10144945
35. Nicolaou A, Shiaeles S, Savage N (2020) Mitigating insider threats using bio-inspired models. Appl Sci 10. https://doi.org/10.3390/app10155046
36. Pantelidis E, Bendiab G, Shiaeles S, Kolokotronis N (2021) Insider threat detection using deep autoencoder and variational autoencoder neural networks. In: 2021 IEEE International conference on cyber security and resilience (CSR), pp 129–134. https://doi.org/10.1109/CSR51186.2021.9527925
37. Le DC, Zincir-Heywood N (2021) Exploring anomalous behaviour detection and classification for insider threat identification. Int J Netw Manag 31(4):2109. https://doi.org/10.1002/nem.2109