RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2527

Insider Threat Detection Using Behavioural Analysis through Machine Learning and Deep Learning Techniques

Pennada Siva Satya Prasad [a, b, *], Sasmita Kumari Nayak [a], M. Vamsi Krishna [c]

[a] Department of Computer Science Engineering, Centurion University of Technology and Management, Bhubaneswar, Odisha, India.

[b] Aditya University, Surampalem, India

[c] Department of MCA, Aditya University, Surampalem, Andhra Pradesh, 533437, India.

* Corresponding Author Email: sivasatyaprasadp@gmail.com

DOI: https://doi.org/10.54392/irjmt2527

Received: 18-11-2024; Revised: 04-03-2025; Accepted: 11-03-2025; Published: 21-03-2025

Abstract: Insider threats pose a significant security challenge to organizational assets and sensitive information. This paper presents a novel approach to insider threat detection by categorizing features into several behavioral types, including Time-related, User-related, Project and Role-related, Activity-related, Logon-related, USB-related, File-related, and Email-related features. Using a comprehensive dataset of 830 features, this paper addresses the challenge of class imbalance through the Synthetic Minority Over-sampling Technique (SMOTE), which improves the balance and preserves data patterns. Dividing features into distinct behavioral categories enhances the precision of threat detection by focusing on specific patterns and anomalies related to different behaviors. The evaluation of machine learning classifiers demonstrates high accuracy across various feature types: Random Forest achieved 76.4% for Time-related, 96.4% for User-related, 85.3% for Project and Role-related, 91.2% for Activity-related, 65.3% for Logon-related, 81.4% for USB-related, 92.5% for File-related, and 99.8% for email-related features. Artificial Neural Networks (ANN) showed good performance with 72% for Time-related, 85% for User-related, 87.6% for Project and Role-related, 75% for Activity-related, 65.5% for Logon-related, 89.7% for USB-related, 86.5% for File-related, and 90% for email-related features. This work underscores the effectiveness of feature categorization and the SMOTE technique in enhancing classifier performance and provides valuable insights for improving organizational security against insider threats.

Keywords: Insider threat Detection, Behavioral Analysis, Machine Learning, CERT, ANN

1. Introduction

Threat assessment for insiders is the collection and analysis of information about a person who may pose a risk to the organization. Threat assessment is a technical process in which a team assesses the individual, determines the severity of the threat and the potential consequences it might have [1]. The rise in insider threats in the cybersecurity space over the past few years is a worrying trend that poses a serious threat to businesses all over the world. Insider threats come from those who have been granted permission to access the organization's data and systems, as opposed to external threats, which originate from outside the organizational boundaries. Because they function inside the secure boundaries of the company's infrastructure, insider threats are very subtle and challenging to identify due to their special quality.

While traditional security mechanisms, e.g. firewalls and intrusion detection systems, are designed to counter external threats, they are far less effective against insider threats, and they generally do not provide an admirable case against the insider attacker [2]. Moreover, insiders have intimate knowledge of your security policies and monitoring systems, which allows them to evade traditional controls without immediately triggering alarm.

Outside of simple data breaches, insider threats can have far-reaching and complex effects. Insider threats impact multiple aspects of corporate integrity, ranging from financial losses and intellectual property theft to reputational damage and legal consequences. In addition, the fact that insider attacks are covert frequently causes a lag in their discovery and reaction, worsening the effects. Proactive detection and mitigation solutions are crucial because insider threats are dynamic and have the potential to cause substantial harm.

According to reports filed in recent years, insider threats cause about 30 percent of data breaches, so organizations must take further precautionary steps.

Unlike external cyberattacks, in which threat actors try to breach security perimeters, insiders take advantage of their trusted access, making detection more complicate [3]. This requires sophisticated behavioral analytics and AI-driven detection mechanisms to best protect their assests [4].

To mitigate the constant threat posed by insider threats, organizations need to implement a comprehensive strategy that includes cutting-edge technologies, strong policies, and attentive oversight. Organizations can strengthen their security and lessen the possible impact of insider-driven security breaches by comprehending the special difficulties presented by insider threats and putting in place efficient countermeasures.

The detection of insider threat is considered by many as becoming increasingly more important as the number of data breaches originating from within an organization continues to rise [5]. Insider threats can wreak havoc on organizations leading to significant financial and operational damage that make up a major part of the security incidents. Unlike external threats that are evident in the way of a hacker trying to manipulate the system, insider threats come in the way of people with access to sensitive systems that grant them permission and therefore, the information needs to be secured within their perimeter [6]. Implementing Security Strategies Organizations need to implement proactive security strategies to protect sensitive data, intellectual property, and customer information from potential misuse by insiders [7].

Behaviorally, one of the biggest challenges of mitigating insider threats, is distinguishing between legitimate user behavior and possible security risks [8]. Although abnormal patterns can indicate malicious intent, they can also be the result of modifications to work habits, job roles, or one-off projects. That means having contextual awareness and adaptive learning models that continue to improve detection as behaviors change over time is critical [9, 10].

To address these challenges effectively, modern threat detection systems increasingly rely on advanced technologies like machine learning (ML) and deep learning (DL). These methods offer unparalleled capabilities in identifying subtle patterns and anomalies that traditional rule-based systems often fail to detect. By leveraging large-scale behavioral data, ML/DL models can analyze user activity, recognize deviations from normal behavior, and predict potential threats with greater precision. For example, DL models like neural networks excel at processing complex datasets, making them ideal for analyzing diverse insider behaviors, including login patterns, file access activities, and communication trends. Such advancements provide a significant edge in combating insider threats, as they allow organizations to stay proactive rather than reactive.

Additionally, as insider threats have grown increasingly sophisticated, data specifically designed for this purpose has emerged, including the CERT dataset, which encodes a wide array of behavioral dimensions. With these datasets, features can be consolidated into logical groups, e. g., time, USB, and email activity, increasing the granularity of the analysis. This not only identifies common insider threat behaviors but also highlights the need for addressing class imbalance through the use of sampling techniques like SMOTE, ADASYN in order to ensure that rare but important insider threat behavior is not missed. This systematic approach not only increases the accuracy of the detection algorithms but also helps to mitigate false positives, making it more appropriate and potentially useful for implementation in real-world threat detection systems.

Due to the multifaceted nature and variety of insider activities, grouping data into logical categories is important to improve analytical accuracy. This classification enables focused analysis of potentially malignant actions like usage of USB devices or email patterns that might indicate a threat. Moreover, complex ML and DL models require structured datasets for accurate anomaly detection. Approaches to handle class imbalance techniques ensure that rare yet crucial patterns of threat do not go unnoticed. All of these components work together to improve detection accuracy, minimize false positives, and increase the overall robustness of the system.

This paper introduces a new method for insider threat detection by analyzing user behavior using an extensive CERT dataset. This dataset covers a wide range of feature categories, such as time-related, user-related, project and role-related, activity-related, logon-related, USB-related, file-related, and email-related features. Proposed approach aims to offer organizations a reliable and efficient solution for identifying and addressing insider threats, leveraging the insights from this comprehensive dataset.

2. Related Works

Insider threats have become a growing concern in the cybersecurity domain, posing significant challenges to organizations. S. Song *et al.,* 2024 [11] investigated behavioural features like absolute time and covariance-based sequences for detecting insider threats. They proposed BRITD (Behavior Rhythm Insider Threat Detection), a DL-based model utilizing Stacked Bidirectional LSTM and Feedforward Neural Networks, which demonstrated high accuracy on the CMU CERT dataset. T. A. Al-Shehari *et al.,* 2024 [12] proposed DBLOF to balance the CERT r4.2 dataset. O. Nikiforova *et al.,* 2024 [13] analysed behavioural models with graph-based networks that improved the accuracy of detection over time. K. C. Roy *et al.,* 2024 [14] explored host data and psychological principles of risk-

taking and impulsiveness for insider threat detection. Cyber and psychological data from 35 students at a big U.S. institution was collected for 90 days. Impulsive, risk-taking users caused more system defects, resulting in (un)intentional insider attacks. A Graph Neural Network applied and achieved good results.

Mahine Learning is essential for insider threat identification. Y. Li *et al.,* 2023 [15] suggested a ML-based log anomaly detection methodology for university cluster system insider threats. It automatically discovers and parses log system data without annotation. By considering IP and role differences, the model learns behaviour patterns for each user type and identifies anomalous actions. D. Sridevi *et al.,* 2024 [16] proposed a hybrid approach using deep neural networks and feature-engineered patterns. It targeted unusual insider activity with high accuracy. It exceeded prior methods with 96.3% detection accuracy. The algorithm could detect minute patterns of risky conduct using classical ML and DL. Hybrid ML approaches were proposed by R. Kumar *et al.,* 2023 [17], where detection accuracies above 96% were achieved by incorporating feature-engineered patterns. Random Forest classifiers, another ML technique, have classified insider behaviour with micro-level accuracy as either "malicious" or "normal" in nature. A. Mittal *et al.,* 2023 [18] developed a similar psychological sentiment analysis technique (LDA+SMO) designed explicitly for negative feelings toward malicious insiders' detection. U. Rauf *et al.,* 2023 [19] introduced hybrid insider threat detection. It detected insider threats more accurately than previous approaches. Additionally, it successfully addressed prejudice and data imbalance. The ensemble technique proposed by A. Diop *et al.,* 2020 [20] for insider threat detection shown notable advantages in terms of scalability and improved classification AUC-score. Pennada Siva Satya Prasad *et al.*, 2024 [21] applied various class imbalance techniques for handling imbalance data. Later, several ML classifiers applied for insider threat detection and reported good results.

Models based on deep learning have been extensively investigated for insider threat detection. F. R. Alzaabi *et al.,* 2024 [22] highlighted insider threat detection and its cybersecurity implications. Modern deep learning and NLP methodologies were examined alongside older machine-learning methods. Experimental results on the CMU CERT dataset showed that enhanced methods work. The study recommended proactive, data-driven insider threat management and improved cybersecurity methods. M. Villarreal *et al.,* 2023 [23] suggested an anomaly detection framework using LSTM models to understand computer system event patterns and identify attack sequences even during protracted assaults. On a dataset of 39 million events, including a 4-day insider threat attack, the technique beat the old system with a 97.29% True Positive Rate and 0.38% False Positive Rate. LSTM demonstrated greater prediction accuracy in variable-length sequences than Hidden Markov Models, which are essential for sequence-analysis-based anomaly identification. J. Xiao *et al.,* 2023 [24] proposed MEWRGNN based on graph neural networks was proposed for contextual behaviour analysis. S. Singh *et al.*, 2023 [25] used Deep Neural Network models to categorize user behaviour as malicious or non-malicious, then identified the threat scenario if malicious. Each ensemble learner was trained with a balanced dataset to avoid bias, and CMU CERT insider threat data studies proved its efficacy. F. Meng *et al.,* 2021 [26] suggested GRU and multi-autoencoder insider threat detection approach. Using imbalanced insider threat data, multi-level filter learning recognized anomalous behaviour. After testing on the CERT dataset, the algorithm detected insider threats better than others. Hybrid methods combining deep neural networks (DNN) and other machine learning approaches have also shown superior performance. For instance, M. Singh *et al.,* 2021 [27] employed a bi-LSTM for feature extraction and a support vector machine as the classifier, achieving improved results compared to other methods. E. Pantelidis *et al.,* 2021 [28] explored Autoencoders and Variational Autoencoders, where the VAE performed better than the former on the CERT dataset.

Unsupervised learning techniques also useful for threat detection. The framework for anomaly detection introduced by D. C. Le et al., 2021 [29] achieved high detection rates on different datasets. J. Wang et al., 2023 [30] proposed deep clustering network to study multi-source behavioural events, and it optimized feature representations for insider threat detection. The technique was successful in identifying abnormal user activity in enterprise contexts. Historical data was also utilized by A. Anju *et al.,* 2023 [31] with unsupervised learning methods to extract behavioural features for identifying threats in computer networks and systems. These clustering and anomaly detection approaches demonstrated effectiveness in identifying unusual user behaviours indicative of potential insider threats.

Number of studies focused on human factors and organizational contexts. F. Whitelaw *et al.*, 2024 [32] investigated research on insider threat prevention in UK financial services, focusing on employee motivation and security habits. An integrated framework proposed by N. Kothari *et al.*, 2024 [33] included understanding technology controls, organizational processes and human factors. S. Eftimie *et al.,* 2020 [34] personality traits and natural language processing were employed to enable proactive detection of potential insider risks.

Emerging techniques and innovations in insider threat detection are required. R. Orizio *et al.*, 2020 [35] introduced AI-based constraint learning. On the CERT dataset, it provided human-interpretable feedback and identified threat limitations that were breached. A. Erola *et al.*, 2022 [36] described the tool's design, validation

process, and implementation in three business situations. It reported the detection system's six-month deployment in actual network infrastructure, lessons gained, challenges encountered, and possible limits. M. Alohaly *et al*., 2022 [37] tracked insiders in dynamic workplaces using ABAC and honey-based deceit. Testing showed that the approach could detect sensitive features and provide identical honey values with a good similarity measure of 0. 91.

3. Method

Figure 1 shows proposed method for insider threat detection. The CERT Insider Threat Dataset, a massive archive of 830 human behavior data, is used to identify insider threats. First, dataset is checked for missing values, categorical variable encoding. There are no missing values and categoical features in the dataset. Behavioral features are extracted using feature engineering techniques and categorized into Time-related, User-related, Project and Role-related, Activity-related, Logon-related, USB-related, File-related, and Email-related groups based on their nature and relevance. The Synthetic Minority Over-sampling Technique (SMOTE) generates synthetic samples for the minority class to balance the dataset's class imbalance, where the majority class outweighs the minority class. For analysis, Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, and Adaboost are chosen. These classifiers are assessed for insider threat detection using accuracy, precision, recall, and F1-score. Deep learning methods, such as Artificial Neural Networks (ANN), are used to assess and classify user behavior using classified datasets. To capture complicated data patterns and connections, ANN models need suitable structures and optimization methods. The proposed approach is verified by comparing model performance on the original and balanced datasets. Finally, experiment findings are examined to detect insider threat patterns, trends, and behavioral indications. These results aid in the development of proactive insider threat detection techniques. The methodology closes with a discussion of the study's consequences and suggestions for future research, emphasizing the need for thorough and proactive insider threat mitigation.

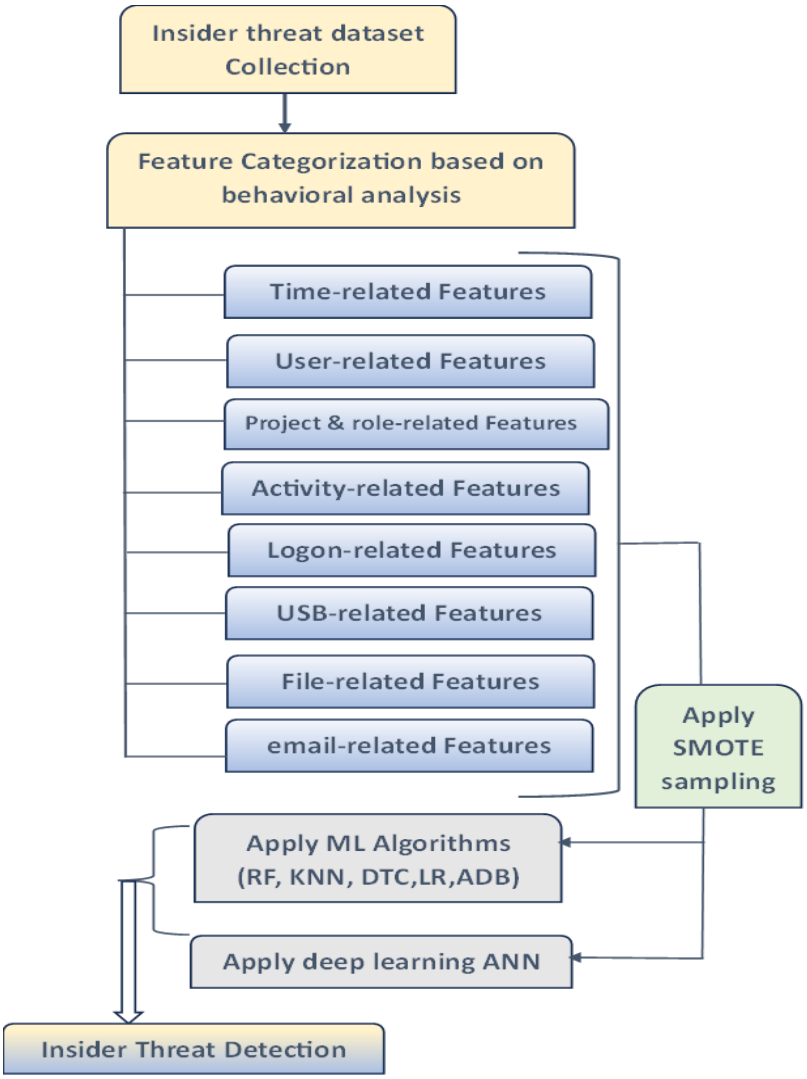

Figure 1. Proposed Method for Insider Threat Detection

3.1. CERT Dataset Collection

Data on insider threats was first gathered by CERT [38]. With 830 features, the dataset has 6,93649 entries. The dataset's "insider" variable is the target. Missing values and outliers are checked in the gathered dataset to ensure there are neither present.

3.2. Feature categorization based on behavioral analysis

During the Feature Engineering and Categorization phase, the raw dataset underwent feature categorization to organize the behavioral attributes into distinct groups based on their nature and relevance.

Table 1. Feature Categorization

Feature Type	Features	Number
Time-related	starttime, endtime, day, week, isweekday, isweekend	6
User-related	user, ITAdmin	2
Project and Role-related	project, role, b_unit, f_unit, dept, team	6
Activity-related	n_allact, allact_n-pc0, allact_n-pc1, allact_n-pc2, allact_n-pc3, n_workhourallact, workhourallact_n-pc0, workhourallact_n-pc1, workhourallact_n-pc2, workhourallact_n-pc3, n_afterhourallact, afterhourallact_n-pc0, afterhourallact_n-pc1, afterhourallact_n-pc2, afterhourallact_n-pc3	15
Logon-related	n_logon, logon_n-pc0, logon_n-pc1, logon_n-pc2, logon_n-pc3, n_workhourlogon, workhourlogon_n-pc0, workhourlogon_n-pc1, workhourlogon_n-pc2, workhourlogon_n-pc3, n_afterhourlogon, afterhourlogon_n-pc0, afterhourlogon_n-pc1, afterhourlogon_n-pc2, afterhourlogon_n-pc3	15
USB-Related	n_usb, usb_mean_usb_dur, usb_mean_file_tree_len, usb_n-pc0, usb_n-pc1, usb_n-pc2, usb_n-pc3, n_workhourusb, workhourusb_mean_usb_dur, workhourusb_mean_file_tree_len, workhourusb_n-pc0, workhourusb_n-pc1, workhourusb_n-pc2, workhourusb_n-pc3, n_afterhourusb, afterhourusb_mean_usb_dur, afterhourusb_mean_file_tree_len, afterhourusb_n-pc0, afterhourusb_n-pc1, afterhourusb_n-pc2, afterhourusb_n-pc3	21
File-Related	n_file, file_mean_file_len, file_mean_file_depth, file_mean_file_nwords, file_n-to_usb1, file_n-from_usb1, file_n-file_act1, file_n-file_act2, file_n-file_act3, file_n-file_act4, file_n-disk0, file_n-disk1, file_n-pc0, file_n-pc1, file_n-pc2, file_n-pc3, file_n_otherf, file_otherf_mean_file_len, file_otherf_mean_file_depth, file_otherf_mean_file_nwords, file_otherf_n-to_usb1, file_otherf_n-from_usb1, file_otherf_n-file_act1, file_otherf_n-file_act2, file_otherf_n-file_act3, file_otherf_n-file_act4, file_otherf_n-disk0, file_otherf_n-disk1, file_otherf_n-pc0, file_otherf_n-pc1, file_otherf_n-pc2, file_otherf_n-pc3	32
Email-Related	n_email, email_mean_n_des, email_mean_n_atts, email_mean_n_exdes, email_mean_n_bccdes, email_mean_email_size, email_mean_email_text_slen, email_mean_email_text_nwords, email_mean_e_att_other, email_mean_e_att_comp, email_mean_e_att_pho, email_mean_e_att_doc, email_mean_e_att_txt, email_mean_e_att_exe, email_mean_e_att_sother, email_mean_e_att_scomp	16

This process involved grouping the features into categories such as Time-related, User-related, Project and Role-related, Activity-related, Logon-related, USB-related, File-related, and Email-related features. This categorization facilitated a structured approach to analyzing various aspects of user behavior, enabling a more comprehensive understanding of potential indicators for insider threat detection. The features used in each category is shown in Table 1. The features are categorized into eight distinct groups(datasets) based on behavioral analysis, each capturing specific aspects of user interactions and activities within the organizational environment. Time-related features encompass temporal information, including starttime, endtime, day, week, isweekday, and isweekend, shedding light on the timing and frequency of user activities. User-related features focus on individual user attributes and administrative roles, such as user and ITAdmin, providing insights into user behaviors and permissions.

Project and Role-related features (6) delineate user roles, affiliations, and responsibilities within specific projects and organizational units, including project, role, b_unit, f_unit, dept, and team. Activity-related features (15) capture user activity metrics and patterns during regular and after-hours periods, facilitating the identification of abnormal engagement levels. Logon-related features (15) characterize user logon activities and access patterns, aiding in the detection of abnormal login behaviors.

USB-related features (21) focus on USB device interactions, providing insights into usage patterns and potential data exfiltration incidents. File-related features (32) encompass file-related metrics, such as file access, transfer, and manipulation activities. Lastly, Email-related features (16) capture email-related metrics and communication patterns, facilitating the analysis of email-based threats and information exchange within the organization. This categorization approach offers a view of user behaviors, enabling effective analysis and detection of insider threats.

3.3. Addressing Class Imbalance

Addressing class imbalance is essential so as to ensure that Machine learning approaches do not dispense with the focus on the majority class and thus actually learn to recognize instances of the minority class. A class imbalance problem occurs in the dataset employed in this study, in which the majority class (the insider threat class label 0) is highly dominant over the minority class (the insider threat class label 1). In order to address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied on all datasets. SMOTE generates synthetic samples from the feature space of the existing minority class instances. Thus, it ensures that in the resultant sample both classes are represented in a balanced manner to avoid skewing the model towards the majority class. Their method improves the model's accuracy in detecting instances of the minority class, leading to better overall performance on various behavioral-based insider threat detection datasets.

3.4. Machine Learning and Deep Learning Approaches

ML and DL based approaches are used in order to detect insider threats based on the user behavior. There are numerous algorithms used in ML. For Random Forest, an ensemble method, the mode of the classes predicted by individual trees is the final output. Logistic Regression: This is again a linear model wherein the dependent variable is binary and this model predicts the log-odds of a binary output given the predictor variables KNN is a non-parametric method which assigns a class to an instance by looking up its closest instances in the feature space and taking the class that the majority belong to. For classification, Decision Tree, a tree-like model, and Adaboost, an ensemble algorithm that combines multiple weak learning individuals into a strong classifier, are also used. Currently deep learning techniques Artificial Neural Networks (ANN)also useful. This involves feeding the information into a neural network (ANN) made up of interconnected neurons arranged across multiple layers that can analyze complex patterns from the data. Different architectures and optimization algorithms are used to create and train ANN models specific to the behavioral datasets. You are trained on the data until October 2023. The detailed evaluation of each algorithm is done through deep experimentation to check how well each algorithm performs in identifying insider threats based on behavioral features.

4. Results and Discussions

4.1. Applying Machine Learning Models

Five ML algorithms, including Random Forest, Logistic Regression, KNN, Decision Tree, and Adaboost are applied. Each algorithm is trained and evaluated on the categorized datasets to assess its performance in detecting insider threats. Table 2 shows the results of using different ML methods with time-related features for insider threat detection. Both Random Forest and Decision Tree models performed well with 80% precision, 76% recall, 76% F1-score, and 76.4% accuracy. Logistic Regression had lower precision at 25%, but a recall of 50%, and an F1-score of 33%, meaning it was less effective overall. K-Nearest Neighbors (KNN) did fairly well with 72% precision, 72% recall, 71% F1-score, and 71.4% accuracy. Adaboost had good performance with78% precision, 11% recall, 69% F1-score, and 70.8% accuracy. These results show that Random Forest and Decision Tree are the best methods for insider threat detection given a consistent performance for all metrics. On the other hand, Logistic

Regression's performance in precision and F1 scores underscores its inability to handle time-specific features efficiently, whereas KNN and Adaboost deliver average performance, but both methods demonstrate a significant opportunity to improve recall values and overall accuracy.

Table 3 shows the results with user-related features. For user-related features, Random Forest and Decision Tree models achieved excellent results with 96% precision, recall, F1-score, and accuracy. Logistic Regression had lower performance with 25% precision, 50% recall, a 33% F1-score, and 50% accuracy.

The K-Nearest Neighbors (KNN) classifier also performed very well with 96% precision, recall, F1-score, and 96.2% accuracy. Adaboost showed moderate performance with 74% precision, recall, F1-score, and 73.5% accuracy. It reveals that Random forest and Decision tree and KNN models are better than others for detecting insider threat using user related features. In comparison, Logistic Regression performs worst while Adaboost performs moderately well; this means that they are both less effective than other models.

Table 4 shows the results of ML models with project and role-related features. Random Forest and Decision Tree models exhibited high performance with 86% precision, 85% recall, 85% F1-score, and 85.3% accuracy. K-Nearest Neighbors (KNN) also performed well with 83% precision, recall, F1-score, and 82.7% accuracy. Logistic Regression showed lower performance with 59% precision, recall, and F1-score, and 60% accuracy. Adaboost had moderate results with 72% precision, recall, 71% F1-score, and 71.5% accuracy.

It reveals that Random forest and Decision tree and KNN models are better than others for detecting insider threat using user related features. In comparison, Logistic Regression performs worst while Adaboost performs moderately well.

Table 2. ML Techniques results with Time Related Features

Method	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random Forest	80	76	76	76.4
Decision Tree	80	76	76	76.4
Logistic Regression	25	50	33	50
KNN	72	72	71	71.4
Adaboost	78	71	69	70.8

Table 3. ML Techniques results with User Related Features

Method	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random Forest	96	96	96	96.4
Decision Tree	96	96	96	96
Logistic Regression	25	50	33	50
KNN	96	96	96	96.2
Adaboost	74	74	74	73.5

Table 4. ML Techniques results with Project and role-related Features

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random Forest	86	85	85	85.3
Decision Tree	86	85	85	85.3
Logistic Regression	59	59	59	60
KNN	83	83	83	82.7
Adaboost	72	72	71	71.5

Table 5 shows the results of ML models with activity related features. Logistic Regression exhibited lower precision (80%) and recall (70%) compared to the tree-based models, resulting in a lower F1-score of 67% and accuracy of 70%. KNN and Naïve Bayes classifiers showed comparable performance, with precision, recall, F1-score, and accuracy all around 65% to 67%. For activity related features, Random Forest and Decision Tree models achieved good results. On the other hand, Logistic Regression was unable to provide good results, KNN and Adaboost were average.

Table 6 shows the results of ML models with login-related features. For login-related features, both Random Forest and Decision Tree models showed solid performance with 71% precision, 65% recall, 63% F1-score, and 65.3% and 65% accuracy, respectively. Logistic Regression exhibited slightly lower precision at 64%, 51% recall, 57% F1-score, and 61% accuracy.

Table 5. ML Techniques results with activity related Features

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random Forest	91	91	91	91.2
Decision Tree	91	91	91	91
Logistic Regression	80	70	67	70
KNN	65	65	65	65
Adaboost	67	67	67	67

Table 6. ML Techniques results with login related Features

Model	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
Random Forest	71	65	63	65.3
Decision Tree	71	65	63	65
Logistic Regression	64	51	57	61
KNN	60	60	60	60
Adaboost	61	61	61	61

Table 7. ML Techniques results with USB related Features

Model	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
Random Forest	83	81	81	81.4
Decision Tree	83	81	81	81.4
Logistic Regression	79	78	78	78
KNN	79	78	78	78
Adaboost	75	74	75	75

Table 8. ML Techniques results with file related Features

Model	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Accuracy (%)**
Random Forest	92	91	91	92.5
Decision Tree	92	91	90	90.5
Logistic Regression	87	86	86	86.4
KNN	85	84	85	85
Adaboost	87	86	86	86.3

K-Nearest Neighbors (KNN) had consistent results with 60% precision, recall, F1-score, and accuracy. Adaboost classifier had similar performance with 61% recall and accuracy. For login related features, Random Forest and Decision Tree models demonstrated reasonable performance. Logistic Regression, KNN, and Adaboost showed limited performance.

Table 7 shows the results of ML models with USB-related features. For USB-related features, both Random Forest and Decision Tree models demonstrated consistent performance, achieving 83% precision, 81% recall, 81% F1-score, and 81.4% accuracy. Logistic Regression and KNN classifiers also showed similar results, with 79% precision, 78% recall, 78% F1-score, and 78% accuracy. Adaboost had slightly lower performance with 75% precision, 74% recall, 75% F1-score, and 75% accuracy. Overall, Random Forest and Decision Tree models exhibited strong and consistent performance with USB-related features.

Table 8 shows the results of ML models with file-related features. For file-related features, Random Forest and Decision Tree models displayed strong performance, achieving 92% precision, 91% recall, and F1-scores of 91% and 90%, with accuracies of 92.5% and 90.5%, respectively. Logistic Regression also performed well, with 87% precision, 86% recall, 86% F1-score, and 86.4% accuracy. K-Nearest Neighbors (KNN) showed solid results with 85% precision, 84% recall, 85% F1-score, and 85% accuracy. Adaboost had good performance with 87% precision, 86% recall, 86% F1-score, and 86.3% accuracy. Overall, Random Forest and Decision Tree models demonstrated excellent performance with file-related features.

Table 9 shows the results of ML models with email-related features. Random Forest demonstrated exceptional performance with 100% precision, 99.8% recall, 99% F1-score, and 99.8% accuracy. Decision Tree also performed excellently with 99% precision, recall, F1-score, and 98.8% accuracy. Logistic Regression had 85% precision, recall, F1-score, and 85% accuracy. KNN & Adaboost achieved measures between 82% to 85%. Random Forest achieved exceptional performance with near-perfect precision, recall, and accuracy, significantly outperforming other models. Decision Tree also delivered strong results, while Logistic Regression, KNN, and Adaboost showed moderate performance with lower precision and recall scores.

Table 9. ML Techniques results with email related Features

Model	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Random Forest	100	99.8	99	99.8
Decision Tree	99	99	99	98.8
Logistic Regression	85	85	85	85
KNN	83	80	82	82.5
Adaboost	84	84	84	84

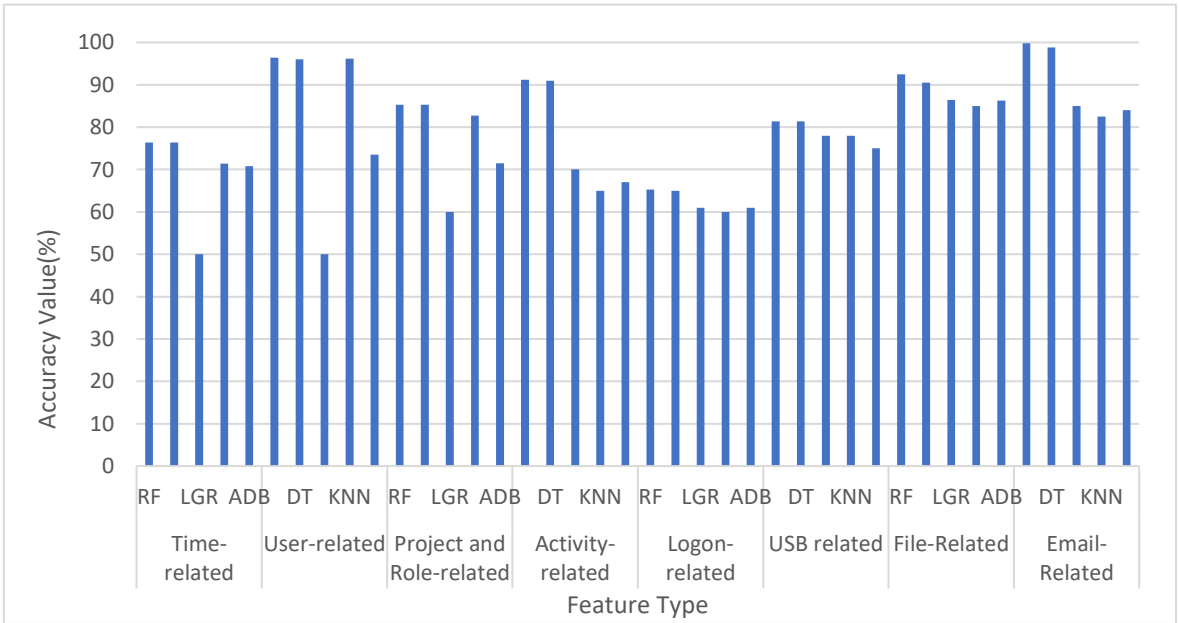

Figure 2. Accuracy comparison of ML methods with all feature datasets

Figure 2 illustrates the accuracy comparison among different feature categories. Analysis of model accuracies across feature types reveals varying performance. For time-related features, Random Forest (RF) and Decision Tree (DT) models achieved the highest accuracy of 76.4%, while Logistic Regression (LGR) exhibited a lower accuracy of 50%. In contrast, user-related features displayed consistent performance, with RF achieving the highest accuracy of 96.4%. Project and role-related features showed good accuracy, with RF and DT reaching 85.3%, whereas LGR had only 60% accuracy. Activity-related features presented a moderate challenge, with RF leading at 91.2% accuracy. Logon-related features followed a similar pattern, with RF achieving 65.3% accuracy. USB-related features showed balanced performance across models, with RF and DT models achieving 81.4% accuracy. For file-related features, RF achieved the highest accuracy of 92.5%, with DT following at 90.5%. Finally, email-related features exhibited strong performance across models, with RF achieving the highest accuracy of 99.8%

4.2. Applying Deep Learning ANN Model

A deep learning Artificial Neural Network (ANN) model was employed for insider threat detection. Table 10 shows the ANN model's accuracy across various feature types.

Table 10. ANN results with all types of features

Model	Accuracy (%)
Time-related	72
User-related	85
Project and Role-related	87.6
Activity-related	75
Login-related	65.5
USB-Related	89.7
File-Related	86.5
Email-Related	90

The model performed best with email-related features (90%) and USB-related features (89.7%). It also showed strong results with project and role-related features (87.6%) and file-related features (86.5%). Accuracy was lower for activity-related features (75%), time-related features (72%), and login-related features (65.5%). Figure 3 compares the accuracy of ANN and Random Forest (RF) models across various feature types for insider threat detection. The ANN model generally performs competitively or slightly better than the RF model. Specifically, ANN outperforms RF in user-related (85% vs. 96.4%), activity-related (75% vs. 91.2%), and file-related features (86.5% vs. 92.5%).

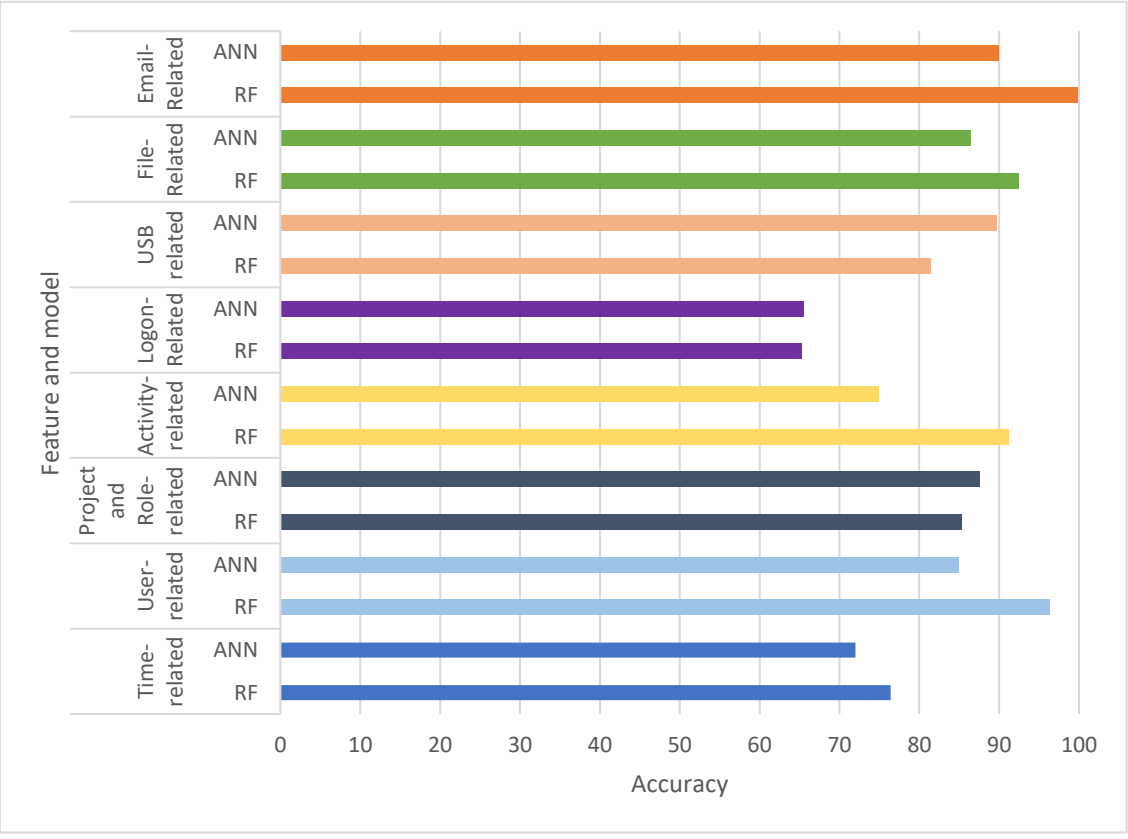

Figure 3. Comparison of ML and ANN models

However, RF achieves slightly higher accuracy for logon-related (65.3% vs. 65.5%) and USB-related features (81.4% vs. 89.7%). Both models excel in email-related features, with ANN achieving 90% accuracy and RF achieving 99.8%. This comparison highlights the strengths of DL in capturing complex patterns in certain features, leading to improved accuracy in insider threat detection.

4.3 Comparison with existing works

Table 11 present a performance comparison of the proposed method with existing works for insider threat detection. The proposed method shows exceptional performance, with the email-related features achieving 99.8% accuracy and USB-related features reaching 89.7%. Compared to existing methods, Neural Networks [24] achieve an accuracy of 98%, while SVM [26] performs slightly lower at 87.5%.

Proposed method delivers excellent improvement, especially for insider threat users, over existing techniques. The model showcases exceptional accuracy rates of 99.8% for the e-mail-related features and 89.7% for the USB-related features combining together numerous features. This highlights the model's capacity to discover complex behaviour and nuanced signs of insider attacks.

Table 11. Comparsion with existing work

Model	Accuracy
Neural Networks [21]	98%
SVM [25]	87.5%
Proposed method [email related features]	99.8%
Proposed method [usb related features]	89.7%

5. Conclusion

This paper highlights the user behavioral analysis for mitigating insider threats with in the organizations. To address the complexity of insider threat detection, behavioral features are divided into independent categories like Time-based, User-based, Project and Role-based, Activity-based categories, etc. To reduce class imbalance and improve machine learning model performance, the SMOTE was employed. Later, several ML models applied. Among ML models, Random Forest showed high accuracy with User-related (96.4%), Activity-related (91.2%), and Email-related features (99.8%). Deep learning models such as Artificial Neural Networks (ANN) also done well with accuracy of 87.6% for Project and Role Related features, 89.7% for USB Related ,90% for Email Related features. These results indicate that while Random Forest provides reliable performance across most feature types, ANN excels in identifying complex patterns. The results demonstrated the importance of integrating ML and DL techniques to enhance insider threat detection systems.

References

[1] Assessing insider threats: CISA (no date) Cybersecurity and Infrastructure Security Agency CISA. Available at: https://www.cisa.gov/topics/physical-security/insider-threat-mitigation/assessing-insider-threats

[2] S. Yuan, X. Wu, Deep learning for insider threat detection: Review, challenges and opportunities. Computers & Security, 104, (2021) 102221. https://doi.org/10.1016/j.cose.2021.102221

[3] M. F. Arroyabe, C.F.A. Arranz, I.F. De Arroyabe, J.C.F. de Arroyabe, Revealing the realities of cybercrime in small and medium enterprises: Understanding fear and taxonomic perspectives. Computers & Security, 141, (2024) 103826. https://doi.org/10.1016/j.cose.2024.103826

[4] Z. Wei, U. Rauf, F. Mohsen, E-Watcher: insider threat monitoring and detection for enhanced security. Annals of Telecommunications, 79(11), (2024) 819–831. https://doi.org/10.1007/s12243-024-01023-7

[5] T.O. Oladimeji, C.K. Ayo, S.E. Adewumi, Review on Insider Threat Detection Techniques. Journal of Physics: Conference Series, IOP Publishing, 1299(1), (2019) 012046. https://doi.org/10.1038/s41598-024-77240-w

[6] D. Mladenovic, M. Antonijevic, L. Jovanovic, V. Simic, M. Zivkovic, N. Bacanin, T. Zivkovic, J. Perisic, Sentiment classification for insider threat identification using metaheuristic optimized machine learning classifiers. Scientific Reports, 14(1), (2024) 25731. https://doi.org/10.1038/s41598-024-77240-w

[7] B. Bin Sarhan, N. Altwaijry, Insider Threat Detection Using Machine Learning Approach. Applied Sciences, 13(1), (2022) 259. https://doi.org/10.3390/app13010259

[8] K. Fei, J. Zhou, Y. Zhou, X. Gu, H. Fan, B. Li, W. Wang, Y. Chen, LaAeb: A comprehensive log-text analysis based approach for insider threat detection. Computers & Security, 148, (2025) 104126. https://doi.org/10.1016/j.cose.2024.104126

[9] M. Vanitha, M. Navya Patel, K. Madhumitha, J. Sathvika, Enhancing Insider Threat Detection in Cloud Environments Through Ensemble Learning. International Journal of Communication Networks and Information Security (IJCNIS), 16(5), (2024) 638–647. https://www.ijcnis.org/index.php/ijcnis/article/view/7870

[10] S. Zeadally, B. Yu, D.H. Jeong, L. Liang, Detecting insider threats: Solutions and trends. Information security journal: A global perspective, 21(4), (2012) 183-192. https://doi.org/10.1080/19393555.2011.654318

[11] S. Song, N. Gao, Y. Zhang, C. Ma, BRITD: behavior rhythm insider threat detection with time awareness and user adaptation. Cybersecurity, 7(1), (2024). https://doi.org/10.1186/s42400-023-00190-9

[12] T. Al-Shehari, D. Rosaci, M. Al-Razgan, T. Alfakih, M. Kadrie, H. Afzal, R. Nawaz, Enhancing Insider Threat Detection in Imbalanced Cybersecurity Settings Using the Density-Based Local Outlier Factor Algorithm. IEEE Access, 12, (2024) 34820 – 34834. https://doi.org/10.1109/ACCESS.2024.3373694

[13] O. Nikiforova, A. Romanovs, V. Zabiniako, J. Kornienko, Detecting and Identifying Insider Threats Based on Advanced Clustering Methods. IEEE Access, 12, (2024) 30242-30253. https://doi.org/10.1109/ACCESS.2024.3365424

[14] K.C. Roy, G. Chen, GraphCH: A Deep Framework for Assessing Cyber-Human Aspects in Insider Threat Detection. IEEE Transactions on Dependable and Secure Computing, 21(5), (2024) 4495-4509. https://doi.org/10.1109/TDSC.2024.3353929

[15] Y. Li, Y. Su, (2023) The Insider Threat Detection Method of University Website Clusters Based on Machine Learning. 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, Chengdu, China. https://doi.org/10.1109/ICAIBD57115.2023.10206282

[16] D. Sridevi, L. Kannagi, G. Vivekanandan, S. Revathi, (2023) Detecting Insider Threats in Cybersecurity Using Machine Learning and Deep Learning Techniques. In 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI), IEEE, India. https://doi.org/10.1109/ICCSAI59793.2023.10421133

[17] R. Kumar, (2023) Thee Machine Learning Analysis of Data Granularity for Insider Threat Detection. 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India. https://doi.org/10.1109/GCAT59970.2023.10353269

[18] A. Mittal, U. Garg, (2023) Prediction and Detection of Insider Threat Detection using Emails: A Comparision. Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), IEEE, Trichirappalli, India. https://doi.org/10.1109/ICEEICT56924.2023.10157297

[19] U. Rauf, Z. Wei, F. Mohsen, (2023) Employee Watcher: A Machine Learning-based Hybrid Insider Threat Detection Framework. 7th Cyber Security in Networking Conference (CSNet), Canada. https://doi.org/10.1109/CSNet59123.2023.10339777

[20] A. Diop, N. Emad, T. Winter, A Parallel and Scalable Framework for Insider Threat Detection. (2020) IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC), Pune, India. https://doi.org/10.1109/HiPC50609.2020.00024

[21] P.S.S. Prasad, S.K. Nayak, M.V. Krishna, Enhanced Insider Threat Detection Through Machine Learning Approach With Imbalanced Data Resolution. Journal of Theoretical and Applied Information Technology, 102(3), (2024).

[22] F.R. Alzaabi, A. Mehmood, A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods. IEEE Access, 12, (2024) 30907-30927. https://doi.org/10.1109/ACCESS.2024.3369906

[23] M. Villarreal-Vasquez, G. Modelo-Howard, S. Dube, B. Bhargava, Hunting for Insider Threats Using LSTM-Based Anomaly Detection. IEEE Transactions on Dependable and Secure Computing, 20(1), (2023) 451-462. https://doi.org/10.1109/TDSC.2021.3135639

[24] J. Xiao, L. Yang, F. Zhong, X. Wang, H. Chen, D. Li, Robust Anomaly-Based Insider Threat Detection Using Graph Neural Network. IEEE Transactions on Network and Service Management, 20(3), (2023) 3717-3733. https://doi.org/10.1109/TNSM.2022.3222635

[25] S. Singh, P. Chattopadhyay, (2023) Hierarchical Classification Using Ensemble of Feed-Forward Networks for Insider Threat Detection from Activity Logs. IEEE 20th India Council International Conference (INDICON), Hyderabad, India. https://doi.org/10.1109/ICPCSN58827.2023.00050

[26] F. Meng, P. Lu, J. Li, T. Hu, M. Yin, F. Lou, (2021) GRU and Multi-autoencoder based Insider Threat Detection for Cyber Security. IEEE Sixth International Conference on Data Science in Cyberspace (DSC), Shenzhen, China. https://doi.org/10.1109/DSC53577.2021.00035

[27] M. Singh, B. Mehtre, S. Sangeetha, (2021) User Behaviour based Insider Threat Detection in Critical Infrastructures. International Conference on Secure Cyber Computing and Communications (ICSCCC), IEEE, Jalandhar, India. https://doi.org/10.1109/ICSCCC51823.2021.9478137

[28] E. Pantelidis, G. Bendiab, S. Shiaeles, N. Kolokotronis, (2021) Insider Threat Detection

using Deep Autoencoder and Variational Autoencoder Neural Networks. IEEE International Conference on Cyber Security and Resilience (CSR), IEEE, Greece. https://doi.org/10.1109/CSR51186.2021.9527925

[29] D.C. Le, N. Zincir-Heywood, Anomaly detection for insider threats using unsupervised ensembles. IEEE Transactions on Network and Service Management, 18(2), (2021) 1152-1164. https://doi.org/10.1109/TNSM.2021.3071928

[30] J. Wang, Q. Sun, C. Zhou, Insider Threat Detection Based on Deep Clustering of Multi-Source Behavioral Events. Applied Sciences, 13(24), (2023) 13021. https://doi.org/10.3390/app132413021

[31] A. Anju, K. Shalini, H. Ravikumar, P. Saranya, M. Krishnamurthy, (2023) Detection of Insider Threats Using Deep Learning. In 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), IEEE, India. https://doi.org/10.1109/ICPCSN58827.2023.00050

[32] F. Whitelaw, J. Riley, N. Elmrabit, A Review of the Insider Threat, a Practitioner Perspective Within the U.K. Financial Services. IEEE Access, 12, (2024) 34752-34768. https://doi.org/10.1109/ACCESS.2024.3373265

[33] N. Kothari, C. Bhardwaj, S. Mishra, S. K. Satapathy, S.B. Cho, P. K. Mallick, (2024) Towards Insider Threat Resilience: A Proposed Mitigation Model. 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India. https://doi.org/10.1109/ESIC60604.2024.10481615

[34] S. Eftimie, R. Moinescu, C. Răcuciu, (2020) Insider Threat Detection Using Natural Language Processing and Personality Profiles. 13th International Conference on Communications (COMM), Bucharest, Romania. https://doi.org/10.1109/COMM48946.2020.9141964

[35] R. Orizio, S. Vuppala, S. Basagiannis, G. Provan, (2020) Towards an Explainable Approach for Insider Threat Detection: Constraint Network Learning. International Conference on Intelligent Data Science Technologies and Applications (IDSTA), Spain. https://doi.org/10.1109/IDSTA50958.2020.9264049

[36] A. Erola, I. Agrafiotis, M. Goldsmith, S. Creese, Insider-threat detection: Lessons from deploying the CITD tool in three multinational organisations. Journal of Information Security and Applications, 67. (2022) 103167. https://doi.org/10.1016/j.jisa.2022.103167

[37] M. Alohaly, O. Balogun, D. Takabi, Integrating cyber deception into attribute-based access control (ABAC) for insider threat detection. IEEE Access, 10, (2022)108965-108978. https://doi.org/10.1109/ACCESS.2022.3213645

[38] Tree, M.B. Directory tree. Available at: https://web.cs.dal.ca/~lcd/data/CERTr5.2/

Authors Contribution Statement

Pennada Siva Satya Prasad: Conceptualization, Methodology, Writing, Software, Validation, Result Analysis. Sasmita Kumari Nayak: Conceptualization, Validation, Visualization, Supervision, Review and Editing. M. Vamsi Krishna: Methodology, Data Collection, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License