University of Ibadan
Journal of Science and Logics in ICT Research

Leveraging Artificial Intelligence for Detecting Insider Threats in Corporate Networks

[1]✉ Nnodi J. T. and [2]Obasi E. C. M.

[1]*nnodijt@fuotuoke.edu.ng*, [2]*obasiec@fuotuoke.edu.ng*,
[1]*https://orcid.org/0009-0006-3326-6149*, [2] https://orcid.org/0009-0001-1513-9887

Abstract

In the modern corporate environment, insider threats pose a significant risk to data integrity, financial stability, and overall cybersecurity. Unlike external attacks, insider threats originate from individuals within an organization like employees, contractors, or partners who possess legitimate access to critical systems. Traditional security measures often fail to identify these threats due to the complexity of distinguishing malicious behaviour from regular activities. Artificial Intelligence (AI) based systems, with their ability to analyse large datasets, detect subtle patterns, and adapt to evolving threat landscapes, offer a powerful approach to insider threat detection. This research involves the application of machine learning algorithms to identify deviations from normal users' activities in corporate networks. The methodology involves analysing user behaviours and access patterns, development and training a machine learning model for classifying user behaviours into normal or abnormal activity. The system helps to identify abnormal user activities and flags suspicious activities in real time, providing an early warning sign for potential breaches. The results demonstrate the effectiveness of machine learning in enhancing threat detection, reducing insider threats, and improving overall cyber security in corporate networks.

Keywords*: Machine learning, insider threat, anomaly detection, behavioural analysis, cyber security*.

1. Introduction

Insider threats represent a significant security risk within corporate networks, where malicious or unintentional actions by trusted insiders can lead to data breaches, intellectual property theft, and other serious incidents. Unlike external threats, which often involve unauthorized access attempts, insider threats are challenging to detect because they stem from individuals who already possess legitimate access to sensitive information. These threats can be subtle, involving deviations from routine behaviour rather than outright malicious actions, making detection often difficult with traditional rule-based or statistical approaches. Insider threat has become a widespread issue and a significant challenge in cybersecurity.

1.1 Characteristics of an Insider Threat

An external threat is typically motivated financially to steal data, extort money, and potentially sell stolen data on darknet markets [12]. While insider threats could share this motivation, it is more likely that an insider will have authorized access to an organisation's system, data or facilities. Insider threats can be malicious or negligent, where the individual intentionally misuses their access to harm the organization, or negligent, where the individual unintentionally exposes the organization to risk through careless actions.

It can be difficult for security controls to distinguish normal from harmful activity and malicious insiders often employ different tactics. Sectors like healthcare, finance, manufacturing, and government are at a very high risk of insider threats due to the sensitive nature of their data and operations or an external entity who hijacks legitimate user credentials through phishing scams or malware, thus gaining unauthorized access to cause data breaches, intellectual property theft, and other serious cyber incidences.

Nnodi J. T., and Obasi C.E.M (2025). Leveraging Artificial Intelligence for Detecting Insider Threats in Corporate Networks. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 13 No. 1, pp. 130 - 143

1.2 Techniques of Detecting Malicious Insiders

Many organizations implement different strategies to detect and mitigate malicious insider threats to avoid significant damage to their data and reputation. Some of these techniques include user behavioural analytics, data loss prevention, machine learning modeling techniques, threat hunting, kill chain detection, user feedback learning and so on. In recent years, machine learning (ML) has shown promise in enhancing insider threat detection by identifying unusual behaviour patterns that deviate from established norms. Machine learning models, particularly those capable of anomaly detection, can learn what constitutes "normal" user behaviour within a network and flag deviations that might indicate insider threats. By leveraging vast amounts of data generated within corporate environments, such as network traffic, user access logs, and behaviour patterns, machine learning models offer a scalable and adaptive approach to detecting subtle threats in real-time.

In this study, we used supervised and unsupervised machine learning approaches to develop and train a model to classify user activities as normal or abnormal by analysing user behaviour, and access patterns. Through extensive experimentation, data preprocessing, and feature engineering, we created a robust and improved detection system capable of identifying insider threats with very high accuracy. The trained model demonstrated a strong ability to detect abnormal behaviour by flagging deviations from baseline activity, providing early warning signs for potential insider threats. Our results indicate that the model effectively reduced false positives compared to traditional detection methods and other machine learning models while accurately identifying potentially harmful activities in real - time.

2. Related Works

Insider threat detection is a broadly investigated topic in which a few solutions have been proposed, particularly, diverse learning methods to facilitate early, more precise discovery. Over the past two decades, analysts have explored insider threat detection and prevention utilizing anomaly-based approaches. These techniques "learn" from normal data only to identify anomalous instances that deviate from expected instances; which has remained one of the most popular methods in the related works reviewed. Anomaly-based detection is based on one major presumption that an attacker's activities are different from a normal user's pattern of activities. Specifically, some of the common behaviours associated with insider threats include (i) the collection of huge datasets and (ii) uploading files that come from outside the organisation's website in Jang *et al.,* [13]. One vital shortcoming of this conventional approach to anomaly detection is that once the baseline has been fully modeled, anything outside this limit will be considered a potential danger; this causes an abundance of false positives [17].

In addition, classification-based insider threat detection represents an alternative research method; it "learns" from normal and anomalous data to determine the decision boundary differentiating normal from anomalous incidences. Nnodi *et al.* [20] developed a machine learning model for classifying users' quality of experience (QoE) on the web using key performance indicators (KPIs) extracted from Quality of Web Service (QWS) dataset. Bin and Altwaijry, [9] gave an up-to-date and comprehensive study of some recent approaches that address insider threat detection which incorporate: (i) machine learning (ML) and deep learning (DL) approaches (either anomaly-based) or (ii) classification-based approaches.

2.1 Traditional Insider Threat Detection Approaches

Behavioural analysis plays a significant role in improving the identification of malicious intent within AI-driven systems compared to conventional strategies. By leveraging advanced machine learning strategies, behavioural analytics can detect subtle anomalies and patterns that indicate malicious activities, which often pass unnoticed by conventional systems. This capability is particularly vital in the context of evolving cyber threats. Enhanced Insider threat Detection Capabilities have been illustrated in the following ways:

(i) Machine Learning Algorithms where AI systems utilize algorithms to analyse behavioural patterns, significantly improving the detection of anomalies that suggest potential security breaches [19].

(ii) Predictive Analytics where AI can forecast future attack vectors by recognizing patterns in

historical data, allowing for proactive defences [23],

Behavioral Profiles where Systems can create detailed profiles of normal behaviour, enabling the identification of deviations that may indicate malicious intent [7] and Reduction of False Positives to Improved Accuracy where AI-driven systems have demonstrated a 30% reduction in false positives, enhancing the reliability of threat detection [4].

Tao *et al.*, [30] proposed a compelling insider threat detection approach based on Back Propagation Neural Network (BPNN). A combination of Variational Autoencoder (VAE) that models normal user behaviour and Back Propagation Neural Network (BPNN) to identify abnormal user behaviour accurately. In their paper, Sharma *et al*., [26] propose user behaviour modeling for anomaly detection using Long Short Term (LSTM).

Yilmaz and Can, [35] explored the intersection of AI and insider threat detection by acknowledging organizations' challenges in identifying and preventing malicious activities by insiders. In this context, the limitations of traditional methods are recognized, and AI techniques, including user behaviour analytics, Natural Language Processing (NLP), Large Language Models (LLMs), and Graph-based approaches, are investigated as potential solutions to provide more effective detection mechanisms.

For this purpose, this paper addressed challenges such as the scarcity of insider threat datasets, privacy concerns, and the evolving nature of employee behavior but the user behaviour analysis for anomaly detection was carried out on a limited dataset.

2.2 Machine Learning Approaches to Insider Threat Detection

AI-powered techniques for detecting insider threats in corporate networks have advanced significantly, leveraging machine learning and deep learning methodologies. These approaches aim to improve detection accuracy whereas minimizing false positives, addressing the unique challenges posed by insider threats.

The most effective strategy identified in recent investigation is: Hybrid Machine Learning Models which include Support Vector Machine (SVM) and K-Nearest Neighbour (KNN). This hybrid model combining SVM and KNN achieved an impressive accuracy of 99%, with high precision and recall rates, making it a robust solution for insider threat detection [2]. Also, combining deep neural networks with feature-engineered patterns has captured subtle behavioural anomalies, resulting in a detection accuracy of 96.3% [28].

Görmez *et al*., [13] showed that Advanced Deep Learning Techniques like Long Short-Term Memory (LSTM) models have demonstrated superior performance in user and entity behaviour analysis, outperforming traditional models in accuracy and detection rates. Synthetic Data Generation (S-LSTM model) which integrates synthetic sample generation to address class imbalance, achieved a prediction accuracy of 99%, showcasing its effectiveness in identifying abnormal behaviours [8].

Other researchers have employed many different algorithms for the insider threat detection problem, such as deep neural networks [3], multi-fuzzy classifiers [27], hidden Markov method, one-class support vector machines, deep belief networks, linear regression and clustering algorithms [9].

2.2.1 Existing Machine Learning Models for Insider Threat Detection

A survey of existing machine learning models, such as anomaly detection algorithms, and their applications in cybersecurity has shown the effectiveness of machine learning algorithms for insider threat detection which has been explored through various innovative means. Notably, hybrid models that combine different algorithms have shown significant promise in enhancing detection accuracy and minimizing false positives [25].

Key Algorithms for Insider Threat Detection include

(a) Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) [2].
(b) Locality Outlier Factor (LOF) and Isolation Forest (iForest). Bidirectional Encoder Representations from Transformers (BERT), combined with an optimized AdaBoost classifier, yielded an accuracy of 97.58%, indicating the potential of natural language processing in threat detection [16].

2.2.2 Related Works on Machine Learning Approach to Insider Threat Detection in Corporate Networks

Katarina *et al.,* [16] proposed a combination of natural language processing with robust classification algorithms, specifically utilizing a modified red fox algorithm and bidirectional encoder representations from transformers, achieving high accuracy in detecting insider threats through email content analysis. Femi-Olowole *et al.,* [11] in their study, identifies Support Vector Machines (SVM) and Recurrent Neural Networks (RNNs) as the most effective machine learning algorithms for insider threat detection, highlighting their versatility and effectiveness in analysing network traffic.

Abhay *et al.,* [1] in their paper proposes the Locality Outlier Factor (LOF) Algorithm and Isolation Forest (IF) Algorithm as effective machine learning algorithms for insider threat detection. Junkai *et al*., [15] proposes a hybrid approach combining unsupervised outlier mining algorithms with supervised learning methods to enhance insider threat detection. This integration improves predictive power, achieving an accuracy of 86.12%, outperforming other anomaly detection methods by up to 12.5%.

Shanmugapriya *et al.,* [10] identifies Conventional Neural Networks (CNN) and Long Short-Term Memory (LSTM) as effective deep learning algorithms for insider threat detection, with CNN demonstrating superior performance over LSTM in terms of accuracy, f-score, precision, and recall when using SMOTE-based balanced data.

Usman *et al.,* [32] in their study did not specify particular machine-learning algorithms for insider threat detection. Instead, the study focuses on a hybrid framework that enhances prediction accuracy using statistical criteria and information gain metrics alongside machine learning-based classification. Talgan *et al.,* [29] proposed effective machine learning algorithms for insider threat detection, including Logistic Regression, Decision Tree, Random Forest, and Xgboost. These models demonstrated improved accuracy, recall, precision, and F1-Score compared to existing methods.

Obasi and Nlerum [21] worked on Intrusion Detection System for Structured Query Language Injection Attack in E-Commerce Database. Their system introduces a filter layer specifically designed to verify user inputs and mitigate known SQL injection threats, thereby enhancing the security of e-commerce platforms.

Obasi and Nlerum [22] developed a model for the Detection and Prevention of Backdoor Attacks using CNN with Federated Learning. The model was trained on a dataset that comprises of 9 classes of MNIST images, of which 8 classes of the dataset were of different classes of backdoor attacks and the class is of non-backdoor attack. The model achieved an accuracy of 99.99% for training and 99.98 for validation.

Yasin *et al* [34] in their study developed several deep learning models, including fully connected layers, convolutional neural networks, and long short-term memory (LSTM) networks. Among these, LSTM models demonstrated superior accuracy and performance for insider threat detection compared to other algorithms. But while these machine learning algorithms demonstrate high accuracy, challenges such as scalability and imbalanced datasets remain prevalent in insider threat detection.

2.3 Gaps in Existing Research

Detecting insider threats in real-time faces several challenges. Delays in processing large amounts of data can prevent quick responses. Extracting useful information from data fast enough for real-time detection is also difficult. Insiders can hide their actions by behaving like normal users, making it harder for the system to spot anything unusual. Detection systems often give too many false alarms or miss real threats, which makes them unreliable.

As companies grow, the huge amounts of data they produce make it harder for these systems to work efficiently. Many companies use both cloud and on-site systems, and getting the models to work smoothly across these setups is tricky. Adding these systems to existing security tools often requires extra work to make them fit. More advanced models may work better but are harder for humans to understand and act on.

The systems also need regular updates to keep up with changing user behaviour, which is time-consuming and hard to scale. These challenges show the need for solutions that are fast, easy to expand, and simple for security teams to use. In

our approach, a machine learning model was developed for insider threat detection using t-sne algorithm, iForest, Random Forest and SMOTE technique to handle dimensionality reduction, anomaly detection, classification and class imbalance respectively.

3.0 Methodology

This section describes the detailed methodology for this research, outlining the approach used to identify anomalous behaviour in user activity and classify potential threats using machine learning models. The methodology comprises several stages: data preprocessing, feature engineering, anomaly detection, classification with hyperparameter tuning, and evaluation. Each stage is elaborated in Figure 1.

3.1 Data Loading and Preprocessing

The first step involves collecting and preprocessing datasets containing user activities, such as logon/logoff events, device usage, and file access. Datasets are loaded from CSV files and undergo cleaning and formatting. Inconsistent entries were resolved by converting string data to uppercase and trimming whitespace. The user information is filtered to include relevant functional units and departments. Merging datasets based on user identifiers allows for a comprehensive and unified view of user activity.

3.1.1 Feature Engineering

Feature engineering transforms raw data into meaningful representations to enhance model performance. Key features extracted from logon/logoff events, device activity, and file usage patterns are:

1. **Logon/Logoff Features**: Minimum and maximum timestamps for user logon and logoff activities are converted to seconds to quantify activity durations.
2. **Device Activity Features**: The average hours of device connection and disconnection activities are computed for each user.
3. **File Usage Features**: Daily file access frequencies are aggregated to calculate the mean and maximum number of files accessed per day per user. The engineered features are merged to form a consolidated dataset, ensuring null values are handled appropriately. This enriched dataset serves as the input for subsequent anomaly detection and classification steps.

3.1.2 Anomaly Detection Using Isolation Forest

Isolation Forest, an unsupervised machine learning algorithm, was applied to detect anomalous user behaviour. The model assigns anomaly scores based on decision functions, identifying deviations from typical patterns. A contamination factor specifies the proportion of expected anomalies in the dataset. The algorithm calculates anomaly values, which are visualized through scatterplot and histogram shown in figures 3 and 4 respectively to inspect their distribution. This step provides insights into the extent and nature of anomalies within the user activity data.

3.1.3 Addressing Class Imbalance and Data Rescaling

Before classification, the dataset was prepared to address the imbalance in the number of normal and anomalous samples. Synthetic Minority Oversampling Technique (SMOTE) generated synthetic examples for the minority class, ensuring balanced training data. The feature set was then scaled using the StandardScaler to standardize feature distributions and improve classifier performance.

3.1.4 Classification with Hyperparameter Tuning

To classify user activities as normal or anomalous, four machine learning classifiers were trained:

1. **Random Forest**: An ensemble method that constructs multiple decision trees and averages their predictions.
2. **Logistic Regression**: A linear model that predicts probabilities for binary classification tasks.
3. **Support Vector Machine (SVM)**: A robust classifier that separates data using a hyperplane in high-dimensional space.
4. **K**-Nearest Neighbour (KNN): A non-parametric classifier which uses proximity to the available categories to classify individual data points.

Each classifier undergoes hyperparameter tuning using GridSearchCV to identify the optimal parameter configuration. Cross-validation

ensures that the models generalize well to unseen data. The classifiers' performance is evaluated on a holdout test set.

3.1.5 Model Evaluation and Visualization

The classifiers are assessed using several performance metrics such as:

1. **Accuracy**: The proportion of correctly classified instances.
2. **Precision, Recall, and F1-Score**: Metrics capturing the balance between true positives and false positives.
   AUC-ROC: The area under the receiver operating characteristic curve, measuring the model's ability to distinguish between classes. Confusion matrices, ROC curves, and bar plots are visualized to compare model performance across metrics. This step highlights each model's strengths and weaknesses.

3.1.6 Model Deployment and Saving

The trained models, along with the scaler, are saved as serialized files using joblib. This enables efficient deployment in real-world applications. The model was stored in a dedicated directory, ensuring reproducibility and accessibility for further research or integration.

Figure 1 shows the architecture of the proposed insider detection system. Various components of the architecture are explained thus:

3.2 Dataset

The lack of actual data is a major barrier for researchers studying the insider threat problem. These data involve log files that contain private user information. To be able to protect their users and assets, organisations frequently deny researchers access to real data. However, under specific regulations, an organisation may agree to give the researchers restricted access after anonymizing the private and confidential attributes of the data. This problem made it difficult to gather data domiciled in a particular organization. To solve the insider threat detection problem, it was, therefore, pertinent to use historical data available in online repositories for the study.

The CERT dataset was used and it has seen a significant increase in usage for insider threat detection systems over the past decade. The dataset contains over one million instances of user activities (such as device usage, log-on activity, users, and file access), with up to one hundred and twenty features. After the feature extraction process, it was reduced to the twelve most important features as shown in Figure 5.

Figure 2 shows a dataset containing 12 columns: user, on_min_ts, on_max_ts, off_min_ts, off_max_ts, device _ connect _ mean _ hour, mean_files_per_day, mean_files_per_day, max _files_per_day, anomaly_score, anomaly_value, and threat. The data includes 607 rows. Each row represents a user with associated activities:

User: This represents a user's unique identifier in the system. It is the primary feature that correlates other features across the different files (logon, device,user and file).

on_min_ts: This is the timestamp (in seconds) when the user logged in for the earliest session (log -on activity). It is converted to seconds from midnight for easier comparison.

on_max_ts: This is the timestamp (in seconds) when the user logs in for the latest session (log-on activity). Like on_min_ts, it is also converted to seconds from midnight.

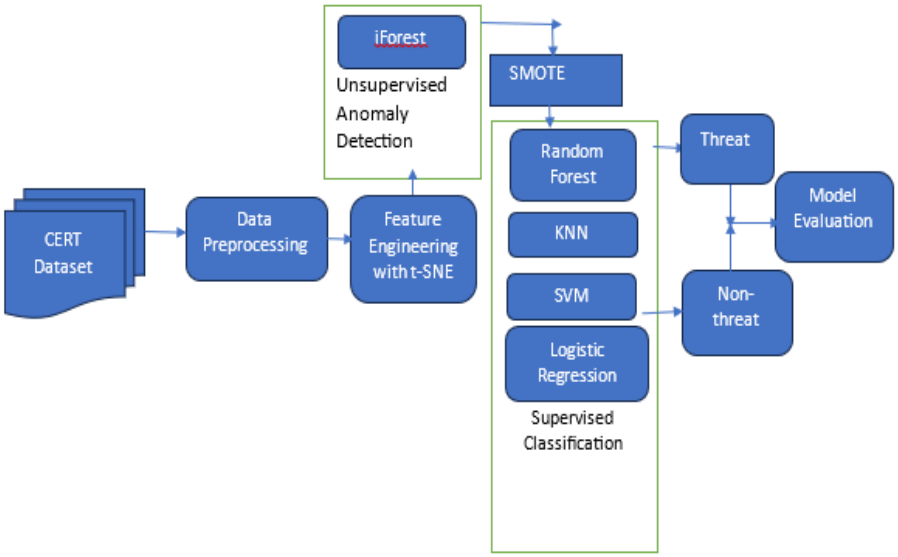

Figure 1: Architecture of the Proposed Insider Threat Detection System

Figure 2: Insider Threat Detection Dataset

off_min_ts: This is the timestamp (in seconds) when the user logs off for the earliest session (logoff activity). It is also converted to seconds from midnight.

off_max_ts: This is the timestamp (in seconds) when the user logged off for the latest session (logoff activity), similar to off_min_ts.

device_connect_mean_hour: This represents the average hour of the day when the user typically connects their device (based on the 'Connect' activity). This is useful for identifying patterns in device usage.

device_disconnect_mean_hour: This represents the average hour of the day when the user ordinarily disconnects their devices (based on the 'Disconnect' activity). This can assist in detecting unusual device usage patterns.

mean_files_per_day: This feature calculates the average number of files a user interacts with (uploads or accesses) daily, giving an overview of a user's file interaction habits.

max_files_per_day: This feature represents the maximum number of files a user interacts with on any given day. It provides insight into periods of high activity or unusual file usage.

anomaly_score: This is the score from the Isolation Forest model showing the level of deviation from normal activity for a given user. The higher the score, the more the behaviour deviates from the norm.

anomaly value: This is the decision function output of the Isolation Forest model. It reflects how much of an anomaly a particular data point

(user behaviour) is. Positive values indicate normal behaviour, and negative values suggest an anomaly.

Threat: This is the binary label assigned to indicate whether a user's behavior is considered a potential threat (1) or not (0). It is based on the anomaly_value (i.e., users with an anomaly_value above a certain threshold are flagged as threats).

These features are used in the supervised learning model (Random Forest) to predict whether a user's behaviour is normal or anomalous to detect potential security threats.

3.2.1 Data Preprocessing

Data collection and pre-processing are fundamental to detecting insider threats and also to perform other cybersecurity exercises. The data collected needed the foundation information necessary for feature extraction. Hence, we performed a feature engineering step in data pre-processing. Utilizing feature engineering, we gathered clean data for additional processing to determine normal business hours and website categories.

3.3. Feature Extraction

One of the main problems with insider threat detection is the extraction of features throughout the feature engineering process. There is no rule regarding the number of features derived from each log file, and it is variable in different studies. The dataset is seen as a relational dataset from which features may be manually derived from the relationships between the entities (files).

4. Results and Discussion

4.1. Results

4.1.1 Anomaly Detection Visualization

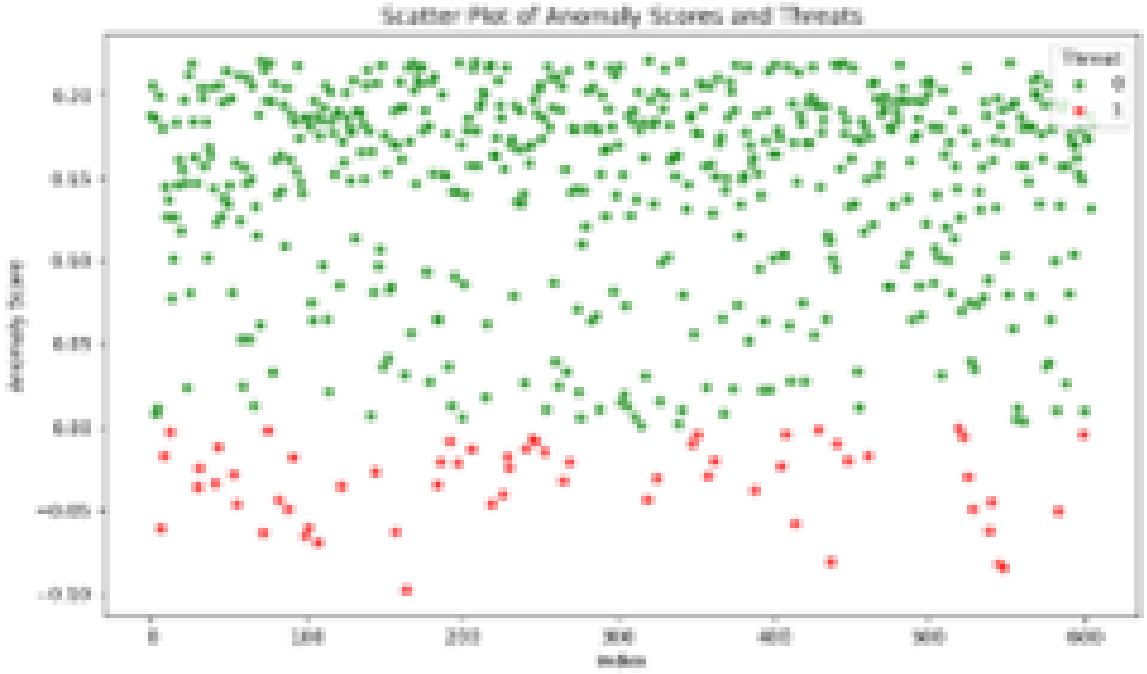

Figure 3: Anomaly Detection Visualization Plot

Figure 3 is a scatter plot that visualizes the relationship between anomaly scores assigned by the Isolation Forest algorithm and the actual threat labels (0 for non-threat, 1 for threat). Each point represents a user and its corresponding anomaly score. The Anomaly Score Distribution shows that the anomaly scores are distributed within a specific range, with most points clustered around a certain value. This indicates that the Isolation Forest has effectively identified a baseline of normal behavior. In threat identification, points colored red represent users labeled as threats while points colored green represent users labeled as non-threats.

Ideally, we would expect threat points to have higher anomaly scores with respect to non-threat points. In model performance, the plot suggests

that the Isolation Forest is moderately effective in identifying threats. There is some overlap between the two classes, indicating that the model might misclassified certain users. The performance of the model in separating threat and non-threat users was further assessed using metrics like precision, recall, and F1-score.

Figure 4 is a histogram that shows the distribution of anomaly scores generated by the Isolation Forest algorithm. It provides insights into the frequency of different anomaly scores, helping us identify potential outliers or anomalous behaviour. The distribution appears right-skewed, meaning that most of the data points have lower anomaly scores. This indicates that the majority of instances are considered normal. The anomaly scores range from approximately -0.25 to 0.05. Negative scores indicate more typical cases, while positive scores suggest anomalies.

The distribution has the highest around 0, suggesting that many data points are clustered around normal behaviour. The tail of the distribution extends towards higher anomaly scores, indicating the presence of a few outliers or anomalous instances. By setting a suitable threshold, we flagged instances that deviate significantly from the norm. The distribution graph provides insights into the model's performance and shows that it is a well-trained model that separates normal and anomalous data points. The choice of the anomaly score threshold is crucial. A higher threshold identifies fewer anomalies with higher confidence, while a lower threshold identifies more anomalies but may include false positives.

4.2 Isolation Forest Distribution of Anomaly Scores

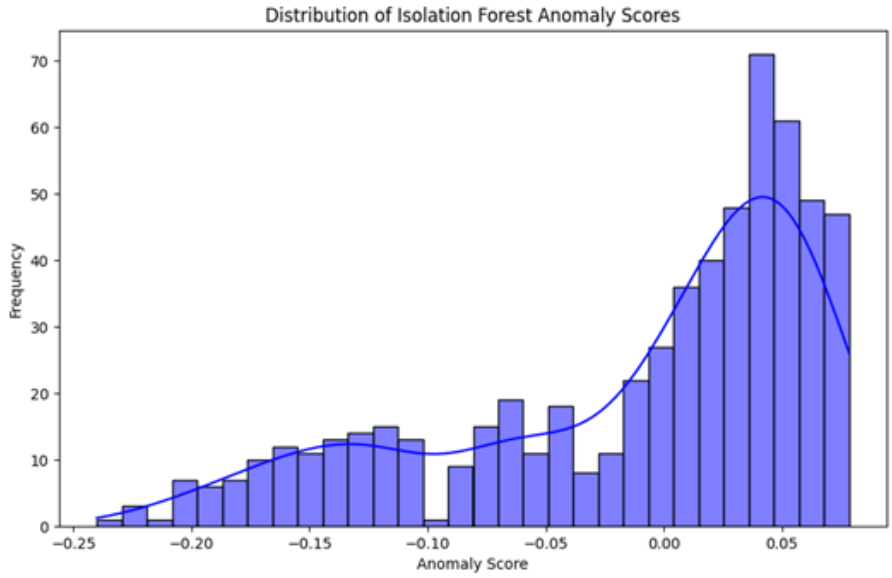

Figure 4: Isolation Forest Distribution of Anomaly Scores

4.3 Confusion matrix of the Random Forest Classifier

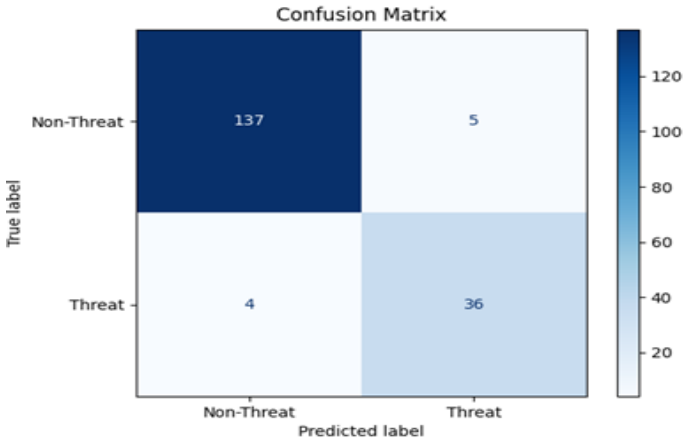

Figure 5: Confusion Matrix of the Random Forest Classifier

Figure 5 is the confusion matrix showing the classification performance of the model with respect to "the Threat" and " the non-Threat" categories.

True Positives (TP): 137 instances were correctly classified as "Threat."

True Negatives (TN): 36 instances were correctly classified as "non-Threat."

False Positives (FP): Only 5 instances were incorrectly classified as "Threat," when they were actually "non-Threat."
False Negatives (FN): Four instances of false negatives occurred, meaning that the model incorrectly classified four "Threat" cases as "non-threat."

Based on these results, the model exhibits a very high accuracy in detecting non-threat cases, with a low false positive rate. However, hyperparameter tuning improved the performance by reducing the relatively high number of false negatives.

Figure 6 is a Receiver Operating Characteristic (ROC) curve. ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

The Area Under the Curve (AUC) is a metric used to assess the overall performance of a classification model. It measures the ability of the model to distinguish between positive and negative classes. An AUC score of 0.97 in figure 6 indicates excellent performance which suggests that the model is highly accurate in distinguishing between positive and negative classes. The curve is closer to the top-left corner, which is ideal. This means that the model can correctly classify positive instances with high probability while minimizing false positives. The curve also shows the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at different threshold settings. By adjusting the threshold, we prioritized sensitivity or specificity based on the specific needs of the application. The high AUC score implies that the model is effective in detecting insider threats. This has significant implications for organizations as it can help them proactively identify and mitigate potential risks.

4.4 ROC Curve and AUC Score

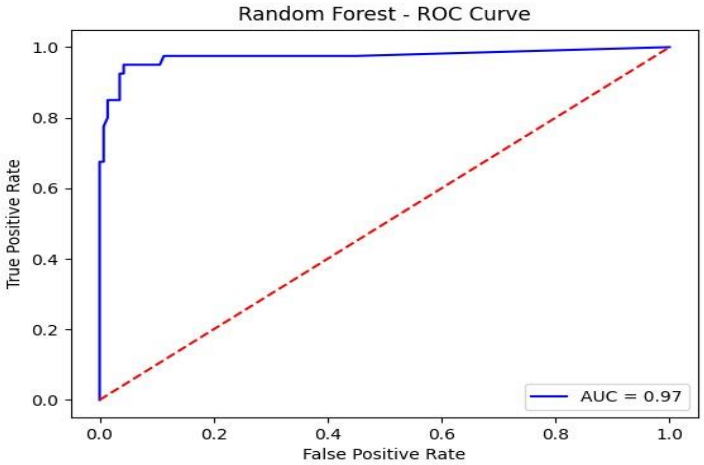

Figure 6: Random Forest ROC Curve

4.5: Insider Threat Detection System Results

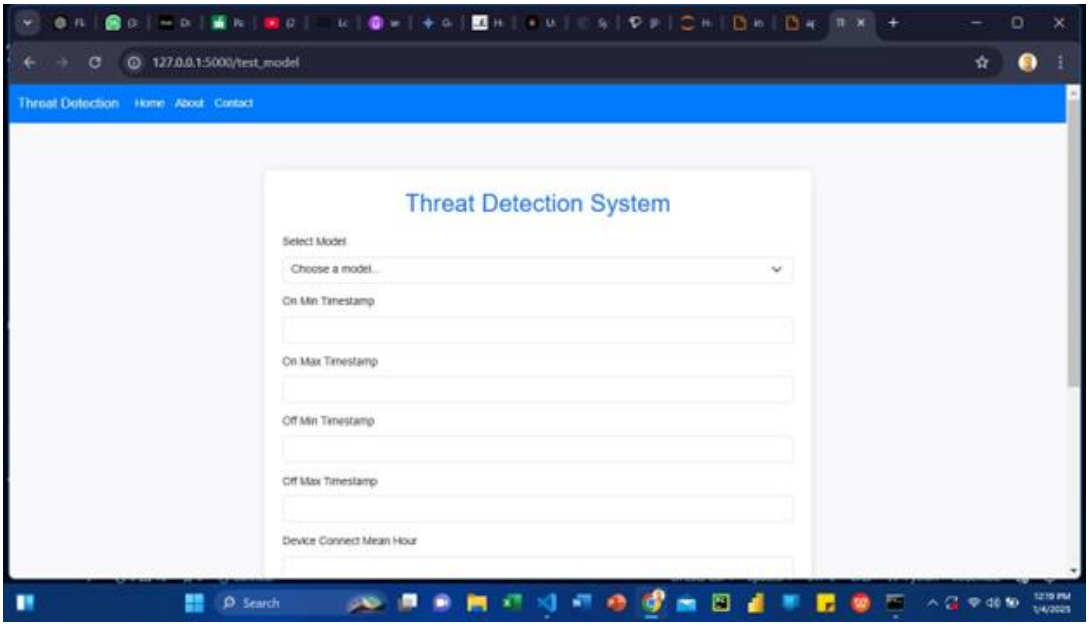

Figure 7: Insider Threat Detection System Input Page

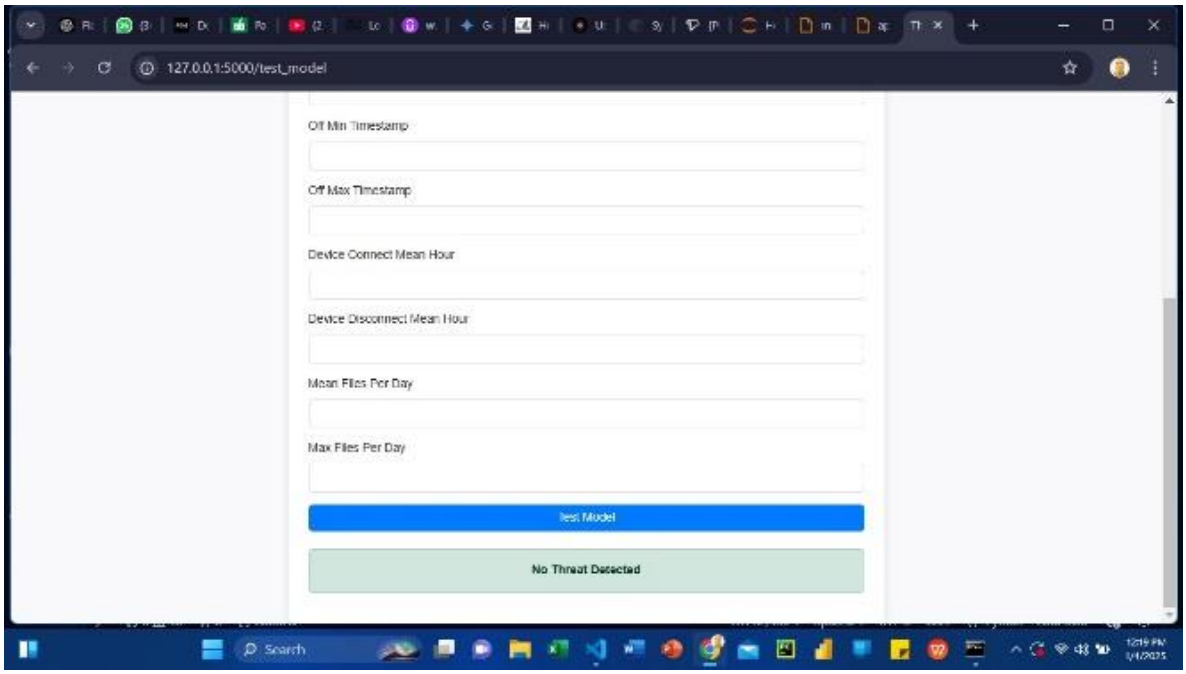

Figure 8: Model Testing page

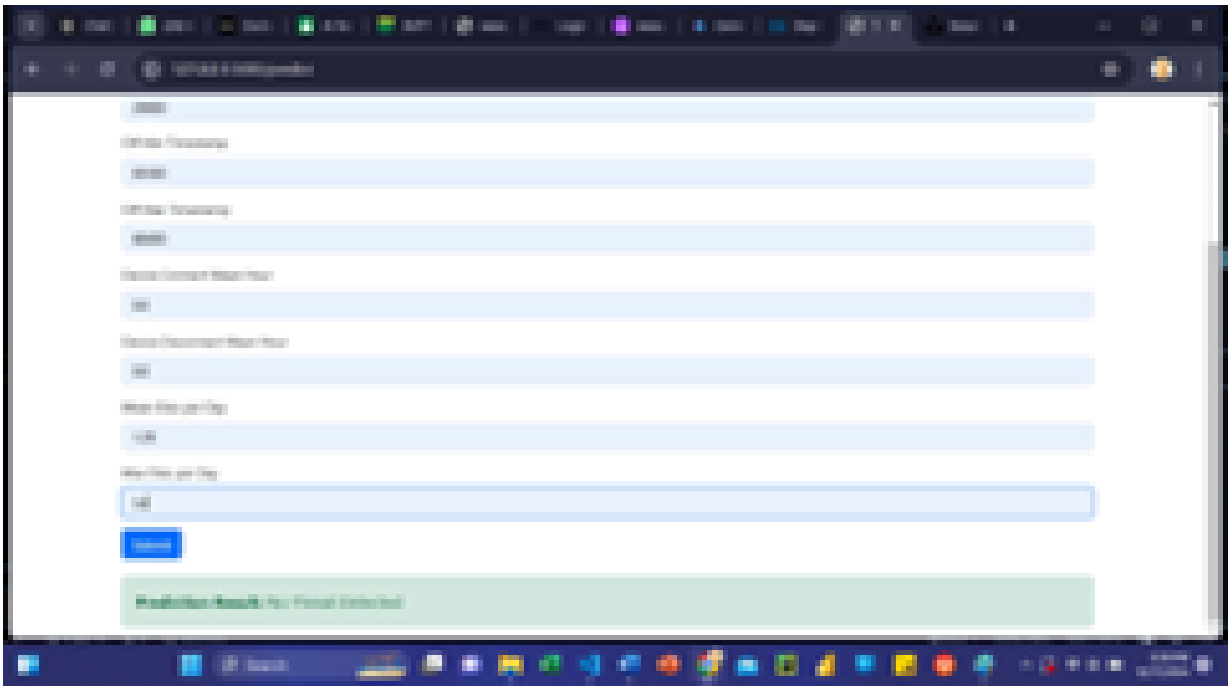

Figure 9: Prediction Result Page for Normal User

4.6 Model Performance Evaluation

Random Forest is an ensemble model based on decision trees and used for classification and regression tasks. It was used in this work to classify users activities into normal and anomalous activities. Metrics used to evaluate the result from the Random Forest classifier and their corresponding results are shown in Table 1.

Table 1: Performance Metrics for Random Forest

Metrics	Value
Precision	0.88
Recall	0.93
F1 Score	0.90
Accuracy	0.95

4.7 Evaluation Metric Comparison with other Algorithms

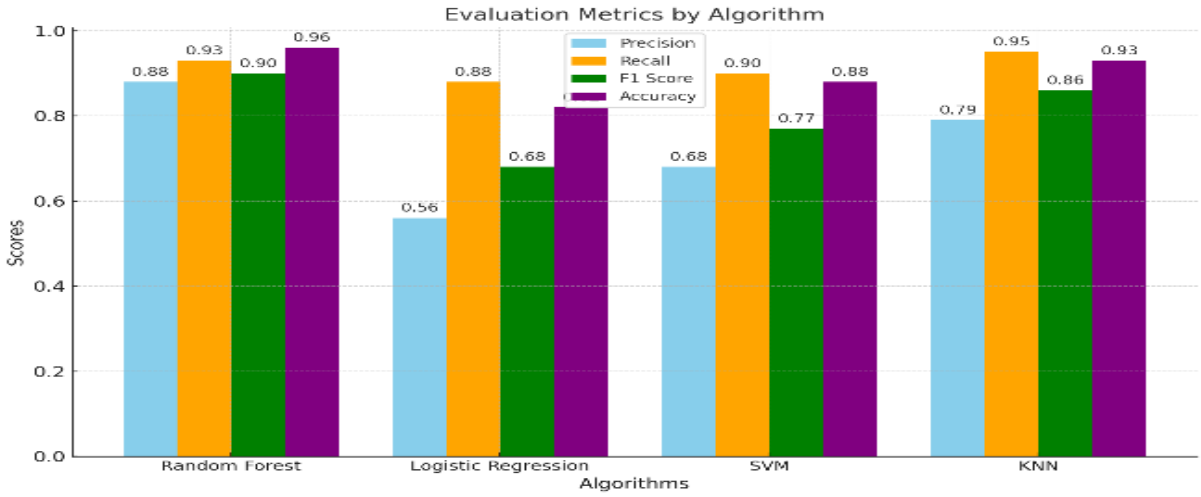

Figure 10: Plot of Evaluation Metrics by Algorithms

Figure 10 is the bar graph summarizing the evaluation metrics for each algorithm. It highlights the Precision, Recall, F1 Score, and Accuracy, allowing a quick comparison of the models' performance.

1. The graph shows that Random Forest outperforms the other algorithms across all metrics, achieving the highest Precision, Recall, F1 Score, and Accuracy.
2. K-Nearest Neighbors (KNN) is the second-best performer, with metrics close to those of Random Forest, particularly strong Recall and F1 Score.
3. Support Vector Machine (SVM) has moderate performance, with an F1 Score and Accuracy slightly lower than KNN, but still competitive.
4. Logistic Regression lags significantly, with the lowest Precision and F1 Score, indicating challenges in correctly predicting positive classes.
5. Random Forest's balanced Precision (0.88) and Recall (0.93) make it the most reliable for both minimizing false positives and false negatives.
6. Logistic Regression compensates for its low Precision with high Recall (0.88), making it suitable in scenarios prioritizing false negative reduction.
7. KNN shows strong Recall (0.95), which, combined with high Precision (0.79), results in a solid F1 Score (0.86).
8. SVM's Recall (0.90) is competitive but its lower Precision (0.68) reduces overall performance.
9. Random Forest and KNN's high Accuracy (0.96 and 0.93, respectively) confirm their reliability in this classification task.
10. Overall, Random Forest emerges as the best choice, followed by KNN, with Logistic Regression being the least effective for this application.

4.8 Results' Discussion

The overall model result shows a high accuracy of 96% with Random Forest after further hyper parameter tuning, which indicates strong overall performance. Slight trade-offs between precision and recall suggest that the model is biased towards certain classifications. To reduce this bias, we compared these metrics to other models or baseline performance, and the comparison shows that Random Forest produced the best result.

5. Conclusion

A real time insider threat detection system has been developed and implemented for real time detection and monitoring of user activities on corporate networks. Key contributions of this research include the development of a comprehensive data preprocessing pipeline, feature engineering, and the application of supervised and unsupervised machine learning techniques specifically tailored for insider threat detection. By deploying this system, organizations can strengthen their security posture against insider threats, allowing for proactive intervention before a potential breach

occurs. This research also contributes to cybersecurity by advancing machine learning applications in anomaly detection and highlighting effective methodologies for real-time, data-driven threat detection in corporate networks.

References

[1] Abhay, Mahajan. (2023). Hybrid Model using LOF and iForest Algorithms for Detection of Insider Threats.doi:10.36227/techrxiv.24522679.v1.

[2] Aghware, F. O., Ojugo, A. A., Adigwe, W., Odiakaose, C. C., Ojei, E. O., Ashioba, N. C., ... & Geteloma, V. O. (2024). Enhancing the random forest model via synthetic minority oversampling technique for credit-card fraud detection. *Journal of Computing Theories and Applications*, *1*(4), 407-420.

[3] Al-Mhiqani, M.N.; Ahmad, R.; Zainal Abidin, Z.; Yassin, W.; Hassan, A.; Abdulkareem, K.H.; Ali, N.S. & Yunos, Z. (2020). A Review of Insider Threat Detection: Classification, Machine Learning Techniques, Datasets, Open Challenges, and Recommendations. Appl. Sci. 2020, 10, 5208.

[4] Amirthayogam, G., Kumaran, N., Gopalakrishnan, S., Brito, K. A., RaviChand, S., & Choubey, S. B. (2024). Integrating behavioral analytics and intrusion detection systems to protect critical infrastructure and smart cities. *Babylonian Journal of Networking*, *2024*, 88-97.

[5] Anju, A., Nithya, Kalyani, M., Shalini, K., Haritha, Ravikumar., Saranya, P. & Krishnamurthy M. (2023). Detection of Insider Threats Using Deep Learning. doi: 10.1109/icpcsn58827.2023.00050

[6] Anupam, Mittal. & Urvashi G. (2023). Prediction and Detection of Insider Threat Detection using Emails: A Comparision. doi: 10.1109/ICEEICT56924.2023.10157297

[7] Asiri, M., Saxena, N., Gjomemo, R., & Burnap, P. (2023). Understanding indicators of compromise against cyber-attacks in industrial control systems: a security perspective. *ACM transactions on cyber-physical systems*, *7*(2), 1-33.

[8] Besnaci, S., Hafidi, M., & Lamia, M. (2023). Dealing with extremely unbalanced data and detecting insider threats with deep neural networks. In *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)* (pp. 1-6). IEEE.

[9] Bin S. B. & Altwaijry, N. (2023). Insider Threat Detection Using Machine Learning Approach. Appl.Sci., 13, 259. https://doi.org/ 10.3390/app13010259

[10] Shanmugapriya D., C., J., Dhanya., S., Asha., G., Padmavathi., D., N., P., Suthisini. (2024). 7. Cloud Insider Threat Detection using Deep Learning Models. doi: 10.23919/indiacom61295.2024.10498767

[11] Femi-Oyewole F., Victor C., Osamor., Daniel & Okunbor. (2024). 3. Survey on Predictive Algorithms to Detect Insider Threats on a Network Using Different Combination of Machine Learning Algorithms. doi: 10.1109/seb4sdg60871.2024.10630366

[12] Insider Threat Detection Techniques: Review of User Behavior Analytics Approach. Available from: https://www.researchgate.net/publication/384241231_Insider_Threat_Detection_Techniques_Review_of_User_Behavior_Analytics_Approach [accessed Nov 14, 2024].

[13] Jang, M., Ryu, Y., Kim, J.-S., & Cho, M. (2020). Against insider threats with hybrid anomaly detection with local-feature autoencoder and global statistics (LAGS). IEICE Transactions on Information and Systems, E103.D(4), 888–891. https://doi.org/10.1587/transinf.2019EDL8180

[14] Jang, M.; Ryu, Y.; Kim, J.S.; Cho, M. (2020). Against Insider Threats with Hybrid Anomaly Detection with Local-Feature Autoencoder and Global Statistics (LAGS). IEICE Trans. Inf. Syst, E103.D, 888–891. [CrossRef]

[15] Junkai, Yi., Yongbo & Tian. (2024). 10. Insider Threat Detection Model Enhancement Using Hybrid Algorithms between Unsupervised and Supervised Learning. Electronics, doi: 10.3390/electronics13050973

[16] Katarina, Kumpf., Mihajlo, Protic., Luka, Jovanovic., Miroslav, Cajic., Miodrag, Živković., Nebojša, Bačanin. (2024). 1. Insider Threat Detection Using Bidirectional Encoder Representations From Transformers and Optimized AdaBoost Classifier. doi: 10.1109/iccsc62074.2024.10616526

[17] Kim, T.Y. & Cho, S.B (2018). Web traffic anomaly detection using C-LSTM neural networks. Expert Syst. Appl. 2018, 106, 66–76. [CrossRef]

[18] Mohd, A, H., Mohd, Abdul, Rahim, Khan., Mohammed, Nasser, A. & Alshehri. (2022). Insider Threat Detection Based on NLP Word Embedding and Machine Learning. Intelligent

Automation and Soft Computing, doi: 10.32604/iasc.2022.021430

[19] Nassif A.B, M. A. Talib, Q. Nasir and F. M. Dakalbab, (2021). "Machine Learning for Anomaly Detection: A Systematic Review," in IEEE Access, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[20] Nnodi J.T., Asagba P. & Ugwu C. (2022). *User classification model for Quality of web experience.* I*nternational Journal of Scientific and Research Publications. 12 (4).*

[21] Obasi, E., & Nlerum, P. (2020). Intrusion Detection System for Structured Query Language Injection Attack in E-Commerce Database. International Journal of Scientific and Research Publications, 10(8), 446–453.https://doi.org/10.29322/IJSRP.10.08.2020.P10455.

[22] Obasi, E.C.M. & Nlerum, P.A. (2023). A Model for the Detection and Prevention of Backdoor Attacks using CNN with Federated Learning. University of Ibadan Journal of Science and Logics in ICT Research, 10(1), 9-21.

[23] Olusegun J. (2024). *Predictive Analytics in Cybersecurity: Using AI to Prevent Threats Before They Occur. Retrieved 0n 2nd January 2025 from*

https://www.researchgate.net/publication/387371545_Predictive_Analytics_in_Cybersecurity_Using_AI_to_Prevent_Threats_Before_They_Occur

[24] Rasheed, F., Yousef., Mahmoud, Jazzar., Amna, Eleyan., Tarek, Bejaoui. (2023). 13. A Machine Learning Framework & Development for Insider Cyber-crime Threats Detection. doi: 10.1109/smartnets58706.2023.10215718

[25] Sarhan, B., & Altwaijry, N. (2023). Insider Threat Detection Using Machine Learning Approach. Appl. Sci., 13, 259. https://doi.org/10.3390/app13010259

[26] Sharma, B., Pokharel, P., & Joshi, B. (2020). User behavior analytics for anomaly detection using LSTM autoencoder-insider threat detection. In *Proceedings of the 11th international conference on advances in information technology* (pp. 1-9).

[27] Singh, M.; Mehtre, B.M. & Sangeetha, S. (2020). Insider Threat Detection Based on User Behaviour Analysis. In Proceedings of the Machine Learning, Image Processing, Network Security and Data Sciences, Silchar, India, Bhattacharjee, A., Borgohain, S.K., Soni, B., Verma, G., Gao, X.Z., Eds.; Communications in Computer and Information Science; Springer: Singapore, pp. 559–574. [CrossRef]

[28] Sridevi, D., Kannagi, L., Vivekanandan, G., & Revathi, S. (2023). Detecting Insider Threats in Cybersecurity Using Machine Learning and Deep Learning Techniques. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)* (pp. 871-875). IEEE.

[29] Talgan, Kumar, Rao., Narayana, Darapaneni., Anwesh, Reddy, Paduri., Amarnath, G, S., Arun, Kumar. & Guruprasad, P. (2023). 8. Insider Threat Detection: Using Classification Models. doi: 10.1145/3607947.3608009

[30] Tao, X., Liu, R., Fu, L., Qiu, Q., Yu, Y., & Zhang, H. (2022). An Effective Insider Threat Detection Apporoach Based on BPNN. In *International Conference on Wireless Algorithms, Systems, and Applications* (pp. 231-243). Cham: Springer Nature Switzerland.

[31] Tej, Akash., Alaka, Jayan., Nimit, Gaur., Saddikuti, Vishnu, Vardhan, Reddy. & Manjith, B., C. (2023). Identifying Insider Cyber Threats Using Behaviour Modelling and Analysis. doi: 10.1109/ocit59427.2023.10431144

[32] Usman, Rauf., Zhiyuan, Wei., Fadi & Mohsen. (2023). Employee Watcher: A Machine Learning-based Hybrid Insider Threat Detection Framework. doi: 10.1109/csnet59123.2023.10339777

[33] Vasileios, Koutsouvelis., Stavros, Shiaeles., Bogdan, Ghita., Gueltoum & Bendiab. (2021). Detection of Insider Threats using Artificial Intelligence and Visualisation. arXiv: Cryptography and Security, doi: 10.1109/NETSOFT48620.2020.9165337

[34] Yasin, Görmez., Halil, Arslan., Yunus, Emre, Isik., Veysel & Gündüz. (2024). Developing Novel Deep Learning Models to Detect Insider Threats and Comparing the Models from Different Perspectives. Bilişim Teknolojileri Dergisi, doi: 10.17671/gazibtd.1386734

[35] Yilmaz, E., & Can, O. (2024). Unveiling Shadows: Harnessing Artificial Intelligence for Insider Threat Detection. Engineering, Technology & Applied Science Research, 14(2), 13341-13346.

[36] Zhiyuan W., Wei U., Usman R., & Fadi M. (2024). E-Watcher: insider threat monitoring and detection for enhanced security. doi:10.1007/s12243-024-01023-7