International Journal of
Human Computations and Intelligence
Vol. 05, Issue. 03, 2026

RESEARCH ARTICLE OPEN ACCESS

An Efficient Machine Learning Framework for Behavioral Insider Threat Detection with Comparative Analysis of Ensemble Methods.

J Sivarani . R Bhargava Reddy . C Sai Sreeja . R Keerthi Priya . S Munendra

Department of CSE (IoT and Cyber Security including Block Chain Technology),
Annamacharya Institute of Technology & Sciences (Autonomous),
Tirupati, A.P, India.

DOI: **10.5281/zenodo.18836210**
Received: 21 January 2026 / Revised: 28 February 2026 / Accepted: 2 March 2026
**Corresponding Author: sivaraniaits@gmail.com*
©Milestone Research Publications, Part of CLOCKSS archiving

Abstract – The insider threats, where insiders with access and authorization misuse it, pose a major challenge to the current prevalent cybersecurity systems. With respect to the insider threats, it can be noted that they have access and authorization, and hence, detection is quite complex and challenging, unlike other attacks by external users. The current analysis focuses on the detection of these insider threats using quite effective and efficient machine learning classifiers, particularly the Decision Tree, Random Forest, and XGBoost classifiers. These classifiers have been chosen because they can handle large volumes of data and can easily cater to the detection of the various insider threats by identifying the pattern or anomaly with respect to the user access. The decision tree is quite efficient and can be easily interpreted, and hence used, whereas the random forest classifier combines the results and predictions made by each and every decision tree, thus giving higher accuracy. The XGBoost classifier, because of its speed and higher accuracy, can easily handle higher volumes of data and provide efficient results, thus becoming quite scalable and useful for resolving the issues posed by insider threats. Experimental results clearly indicate XGBoost is better suited for accuracy and yields a result of 98% for complicated scenarios related to threats, while Random Forest and Decision Tree help yield better results and save resources.

Index Terms – Insider threats, machine learning, Decision Tree, Random Forest, XGBoost, threat detection, cybersecurity.

MILESTONE RESEARCH.IN OPEN ACCESS

ISSN (Online): 2583-5696
Int. Jr. of Hum Comp. & Int.

© The Author(s) 2026. Published by Milestone Research Publications. This article is licensed under the Creative Commons Attribution 4.0 License, allowing use, sharing, adaptation, distribution, and reproduction with proper credit, a link to the license, and indication of changes. Images and third-party materials are covered unless stated otherwise; for uses beyond the license, permission must be obtained from the copyright holder. Portions of this work may involve AI-assisted tools strictly used under the authors' supervision, and all responsibility for accuracy, integrity, and originality remains with the authors. License: http://creativecommons.org/licenses/by/4.0/

I. INTRODUCTION

Insider threats, whereby an insider utilizes their authorized access to carry out malicious activities, is one of the most significant challenges in modern cybersecurity. This is different from other types of hacking, as they cannot be easily identified since they already carry the necessary access to the targeted data in the organization. With the recent development in digital infrastructure, as well as the utilization of remote working, identifying insider threats is more intricate, calling for sophisticated systems to carry out identification in real-time. The reason behind this is that ML has provided a promising solution for automating insider threat detection by analyzing user behavior, network traffic, and system logs. Among different ML models, Decision Tree, Random Forest, and XGBoost are some of the most used ones due to their ability to handle large datasets with the detection of subtle anomalies in behavior. Decision Trees provide simplicity and interpretability; hence, they are appropriate to use when identifying patterns in data and also to understand the decision-making process. An ensemble technique referred to as Random Forests provides improvements over Decision Trees, given its ability to increase the accuracy of the results with the aggregated predictions made from various decision trees. XGBoost is a gradient boosting technique that has gained popularity in dealing with large data sets with high-classification accuracy.

In 2025, a research study was conducted with the aim of explaining the potential opportunity of the Random Forest model in the detection of insider fraud in a financial system. The findings proved that the model is applicable because it is capable of giving precise forecasts of fraud activities [1]. The XGBoost model has been applied to detect insider threats in network systems in real-time, proving its supremacy in its ability to classify fraud activities in relation to insider threats [2]. Further, Decision Trees have been integrated in the development of hybrid models for detecting insider threats, whose major characteristic revolves around user access anomalies [3]. Although all the models claimed success in their respective research, few studies have evaluated the relative success of these models in actual insider threat scenarios. In a recent study, the authors attempted to compare the performance of Random Forest and XGBoost models for threat detection and claimed that combining the two achieved better accuracy, as their strengths complement one another [4]. The studies reported the success of combining XGBoost and Deep Learning to achieve better precision in model performance for detecting the threat [5], [6]. In actual scenarios, Decision Trees are reported to perform better in low-resource scenarios, where the model transparency becomes important [7].

The use of XGBoost in the context of cybersecurity has also been explored in relation to the detection of insider data exfiltration and abnormal behavior in large-scale systems [8]. In 2025, it was found that the Random Forest algorithm was highly effective in handling imbalanced datasets, as is the case in detecting insider threats [9]. In addition, the use of XGBoost has been integrated into enterprise security systems, with the ability to provide real-time detection and high accuracy in the detection of threats [10].

Our contributions are as follows:

- We suggest an advanced method to effectively address insider threat detection through the combination of machine learning classifiers such as Decision Tree, Random Forest, and XGBoost.

CC BY NC ND

- The performance of these classifiers has been evaluated and compared with each other, revealing that XGBoost beats the other three in terms of precision and accuracy, including achieving a high accuracy of 98% in detecting intricate insider threats.
- We also propose a framework that incorporates these various models of machine learning with data preprocessing techniques, as well as class imbalance techniques such as SMOTE, to improve detection.
- The present research proves the capability of XGBoost as the most efficient model for insider threat detection, which can be used for scalable solutions in enterprise security systems.

II. LITERATURE SURVEY

Detecting insider threats is not easy because of infrequent and highly skewed malicious activity within organizations. Recent studies use machine learning, deep learning, graph models, and LLM-based log analysis in order to enhance detection. The purpose of the approaches is to increase the accuracy and practicality. Gao et al. [11] suggested Deep Temporal Graph Infomax (DTGI) to detect insider threats when the classes are under extreme imbalance. Their model frames the user activities in terms of an evolving heterogeneous time graph and offers a behavior-context restricted generator of anomaly samples to improve supervised learning. Experimental findings on the CERT dataset also indicate that DTGI is superior to the available techniques, but the use of generated samples and high computational complexity pose scalability and generalization challenges.

Nikiforova et al. [12] paid attention to the insider threat detection drawn upon the sophisticated clustering methods. They also suggested improving the e-StepControl system by automatically grouping the users according to graph-modeled audit actions. The technique enhances the detection of behavioural anomaly by using similarity to group as well as the quality of audit data and sensitivity of clustering methods to changes in the user behaviour over time. Pennada et al. [13] handled the insider threat through behavioural analysis with machine learning and deep learning. They identified 830 behavioural features and divided them into several categories, and used SMOTE oversampling to balance the imbalance. Their model of Random Forest had an accuracy of 99.8% , but the complexity of feature engineering, as well as the problem of bias due to synthetic oversampling, are weaknesses of the model.

Manoharan et al. [14] provided a comparative analysis of the supervised machine learning algorithms for insider threat detection on the CERT dataset. The performance and difference between imbalanced and balanced datasets were specifically studied by them, and the significance of hyperparameter tuning was revealed. Findings indicated that the best F1-score of 95.9 % was obtained with Random Forest, but the framework is limited by the need to be based on a single dataset and a small amount of validation in real-world settings. Prasad et al. [15] suggested an ensemble-based machine learning approach to the combination of PCA and imbalance removal methods like SMOTE and ADASYN. Their method showed better performance of detection on imbalanced CERT data than baseline models. Nevertheless, these limitations are synthetic sampling bias, testing on a single dataset, and the ambiguity of generalizing the results to other organizational settings.

Lavanya et al. [16] proposed an Enhanced Bidirectional GAN(EBiGAN) with Bayesian-optimized DNN (DNN-PI) to detect insider threats in IoT settings. They are novel by their adversarial generation of samples, additional discriminator, and better PCA-based clustering. Despite a high detection rate and low false alarms, the framework is computationally expensive and relies on artificial data, which prevents it

CC BY NC ND

from being applicable in any real-world implementation of IoT applications. The Deep Synthesis Insider Intrusion Detection (DS-IID) model by Kotb et al. [17] was created to distinguish between genuine and artificial malicious insider behavior using an imbalanced dataset. They are novel because of a deep feature synthesis, weighted sampling, and binary deep learning classification. The model demonstrated high CERT 97% accuracy (AUC 0.99), but its drawbacks are that it is synthetic-oriented and has questionable applicability to a variety of organizational settings.

Wang and El Saddik [18] have introduced a smart insider threat detector framework called Digital Twin-based Insider Threat Detection (Dtitd), which is developed using the digital twin technology and self-attention deep learning. They proposed DistilledTrans, an easier version of transformers that are used with highly imbalanced data, and BERT/GPT-2 contextual augmentation. The model minimized training cost and maximized performance, but its reliance on massive pre-trained models and difficulty with cross-organizational generalization remain disadvantages. Song et al. [19] designed Audit-LLM, a multi-agent log-based insider threat detector system to address the weaknesses of long audit logs and hallucinations. An Evidence-based Multi-agent Debate (EMAD) mechanism enhanced reliability because of its three-agent structure (Decomposer, Tool Builder, Executor). The CERT r4.2, CERT r5.2, and PicoDomain results are better, but more complexity and scaling issues are presented by the system because of the requirements of the LLM stability. In the article by Li et al. [20], the authors suggested RedChronos, an LLCM-based log analysis platform that aims to minimize manual work in enterprise insider threat detection. They use Query-Aware Weighted Voting and Semantic Expansion Genetic Algorithm mutation instructed by LLM reasoning.

III. METHODOLOGY

In this section, we will briefly discuss the dataset description, data preprocessing, and the ML classifiers we utilized. Figure 1 depicts the overall research methodology.

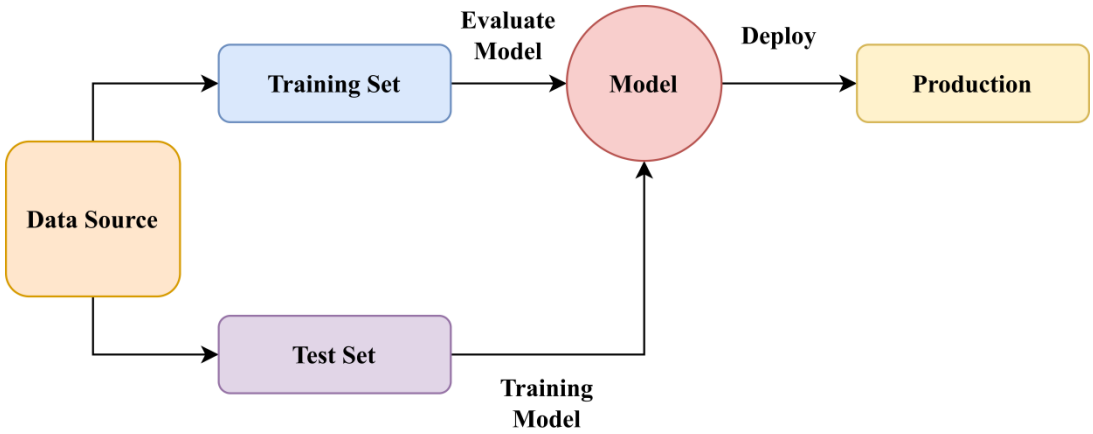

Fig. 1: Graphical representation of the proposed model architecture

A. Dataset Description

This study fetched the CERT Insider Threat Detection Research Dataset which is one of the most popular synthetic datasets used to assess machine learning models designed to detect insider threats in an

CC BY NC ND

organizational setting. This is created by the CERT Division of Carnegie Mellon University. This dataset assists in the assessment and testing of the detection models by including both normal and abnormal situations, such as insider threats. In this dataset, numerous logs, including logon/logoff, email, device, and HTTP logs, are integrated and represented as normally expected in an online platform. This data set is vastly applicable in the implementation of both supervised and unsupervised machine learning algorithms, in which the instances of performing insider threats are normally unsupervised, while normal activities are supervised. This is to show that the number of normal and abnormal activities in the data set is generally very rare in nature. Due to privacy and legal issues, real-world insider threat data is scarce, making the synthetic CERT dataset an invaluable resource for researchers. Features extracted from the logs, such as session duration, event frequency, and anomalous patterns, are often used for feature engineering and preprocessing. These features are essential for training accurate and efficient machine learning models. The complexity and realistic structure of the dataset are conducive to assessing classification models such as Decision Tree, Random Forest, XGBoost, and providing useful information on the performance of such models in performing insider threat detection tasks.

B. Data Preprocessing

Raw organizational activity logs are often high-dimensional, noisy, and temporally inconsistent when used to detect insider threats. As a result, effective preprocessing is required to convert heterogeneous behavioral traces into a structured feature space appropriate for sophisticated classification models.

- Behavioral Log Normalization and Temporal Feature Extraction: Multiple sources of user activity records were used, including logon/logoff events, email transactions, web browser history, file operations, and interactions with removable devices. Temporal normalization was performed initially because each event stream contained timestamped records. Each activity timestamp was transformed to a standardized datetime format:

$$t_i = datetime\ (r_i)$$

Two significant behavioral indications were derived from this.

Day-of-week-feature

$$d_i = DayOfWeek\ (t_i)$$

Hour-of-day feature

$$h_i = Hour\ (t_i)$$

- Feature Space Generation Across Multi-Source Logs: User grouped raw event records to represent each employee's behavioral footprint. A feature vector for user u was created as follows:

$$X_u = [L_u, E_u, H_u, F_u, D_u, P_u]$$

where, L_u is logon behavior features, E_u is the email communication feature, H_u defines the HTTP browsing feature, F_u is the file system activity features, D_u denotes the removable device, and P_u represents the psychometric traits.

- Data Cleaning and Missing Value Treatment: Missing feature values were unavoidable due to incomplete behavioral logs and limited user activity. Missing entries were imputed with median-based replacement to maintain dataset integrity:

$$x_{ij} = \begin{cases} median\ (X_j), & if\ x_{ij}\ is\ missing \\ x_{ij}, & otherwise \end{cases}$$

CC BY NC ND

Median imputation was chosen over mean substitution because of its resistance to behavioral outliers.

- Encoding of Categorical Behavioral Attributes: Certain features, such as the most often used PCs or recipients, are categorical. Label encoding was used because classification models required numerical input.

$$c_k \longrightarrow \{0, 1, 2, \ldots \ldots, K-1\}$$

- Insider Threat Label Assignment Strategy: Because real insider threat datasets frequently lack clear ground truth labels, a statistical threshold-based labeling technique was chosen. Selected risk-sensitive behavioral markers were compared to their global mean:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$$

A user was designated as suspicious if key activity exceeded a specified deviation level.

- Handling Class Imbalance Using SMOTE: The scarcity of harmful incidents presents a significant problem in insider threat detection. Insider attack activities were greatly underrepresented in our data. To reduce classifier bias toward the majority class, SMOTE was used. For every minority instance x_i, synthetic samples were created as:

$$x_{new} = x_i + \lambda(x_{nn} - x_i)$$

where, x_{nn} is the nearest neighbor in minority class, and $\lambda \in [0, 1]$ is the interpolation factor.

- Final Preprocessed Dataset Representation: After preprocessing and integration, the final dataset comprised:

4000 users × 66 behavioral and psychometric features

This oversampling method enhances the classifier's ability to detect unusual insider attack actions, such as sabotage and exfiltration.

C. Machine Learning Classifiers

After designing a uniform behavioral feature space, supervised machine learning classifiers were employed to distinguish between regular users and multiple insider threat groups. The completed dataset should be expressed as follows:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

where, $x_i \in \mathbb{R}^d$ define the extracted behavioral feature vector of the i^{th} user, $y_i \in \{1, 2, \ldots, C\}$ describe the corresponding insider threat class label. N is the number of users and C is the number of the threat categories. The goal of classification is to learn a mapping function:

$$f : x \longrightarrow y$$

- Logistic Regression (LR): LR is a probabilistic linear classifier commonly used as a baseline for threat detection due to its interpretability and effectiveness, which uses the logistic (sigmoid) function to model the posterior probability of a class. For binary classification, the probability of an insider threat is given as:

$$P(y=1|x) = \sigma(w^T x + b)$$

where, w is the weight vector, b is the bias term, $\sigma(w^T x + b)$ are the sigmoid activation:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

CC BY NC ND

The softmax extension is used to predict multi-class insider threats:

$$P\ (y = k|x) = \frac{ew_k^T x}{\sum_{j=1}^{C} e^{w_j^T x}}$$

LR is a reliable baseline for assessing sophisticated insider classification methods.

- Decision Tree (DT): DTs characterize user behavior by recursively partitioning the feature space into threat-sensitive areas. At each node, the ideal split is determined by maximizing information gain. The impurity of a node is often quantified using the Gini index:

$$G\ (S) = 1 - \sum_{k=1}^{C} p_k^2$$

where p_k is the proportion of samples belonging to class k.
The best divided feature f is selected such that:

$$f = \arg\min_{f} \sum_{m \in children} \frac{|S_m|}{|S|} G(S_m)$$

DTs are especially effective at detecting insider threats because they capture nonlinear behavioral anomalies, such as unusual file deletions or late-night logons.

- Random Forest (RF): RF is an ensemble learning technique that enhances detection robustness by merging numerous decision trees. Each tree is trained on a bootstrap sample of the dataset and classified using majority voting.
The projected classification is:

$$\hat{y} = mode\ \{T_1(x), T_2(x), \ldots\ldots, T_M(x)\}$$

where, $T_M(x)$ is the prediction of the m^{th} tree, and M is the number of trees.

- AdaBoost (Adaptive Boosting): AdaBoost is a boosting-based ensemble method for combining weak learners to create a strong classifier. It assigns higher weights to misclassified insider incidents, thereby improving the detection of uncommon threat behaviors. Initially, all samples are equally weighted:

$$w_i^{(l)} = \frac{1}{N}$$

At iteration t, the weak classifier error is calculated as follows:

$$e_t = \sum_{i=1}^{N} w_i^{(t)} \mathbb{I}(y_i \neq h_t(x_i))$$

The classifier weight is then:

$$a_t = In(\frac{1 - e_t}{e_t})$$

The final insider prediction is:

$$H\ (x) = \arg\max_{k} \sum_{t=1}^{T} a_t \mathbb{I}(h_t(x) = k)$$

AdaBoost is very useful in insider threat datasets, where harmful samples are sparse and difficult to identify.

- Extreme Gradient Boosting (XGBoost): XGBoost is a sophisticated gradient-boosting framework that improves prediction accuracy by optimizing decision trees via gradient-based learning and regularization. Model prediction is defined as:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i)$$

CC BY NC ND

where, every f_t defines a regression tree.

$$\mathcal{L} = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

with regularization term:

$$\Omega(f_t) = \gamma K + \frac{1}{2} \lambda ||w||^2$$

where, K is the number of leaf nodes, and γ, λ are control model complexity.

Because it can detect subtle behavioral anomalies across multiple sources of organizational activity, XGBoost is highly effective at insider threat detection.

D. Deployment Architecture

The insider threat detection system is designed to ensure a smooth, useful flow between users and the machine learning core. This system ensures that users, whether administrators or not, can access it by pressing a regular web browser. The journey to insider threat detection through this system begins at the web server. Following this, there is a flow to the machine learning server, which quickly detects users based on their activities. During this period, communication with the database server is also used to fetch logs and store results. At this point, there is also some detection, which then flows back through the web server before being presented to the user.

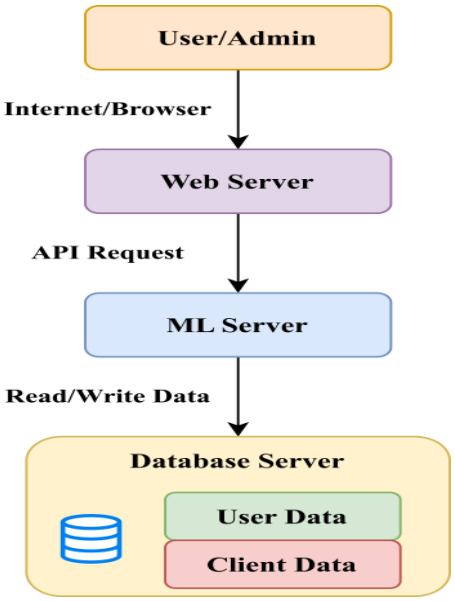

Fig. 2: Deployment Diagram

IV. HARDWARE AND SOFTWARE SPECIFICATIONS

All these experiments would require at least a high-performance machine, Intel Core i7, or better, accompanied by at least 16 GB RAM, in order to keep up with its large dataset and computationally expensive processing. There should be at least 512 GB SSD storage to handle the different models efficiently. An NVIDIA GTX 1060 or higher could be optionally used for accelerating processes such as training, especially when dealing with deep learning models. Operating System: It should be installed with

CC BY NC ND

either Windows 10 or a Linux-based operating system, such as Ubuntu, which will be preferable since most machine learning frameworks were initially developed for Linux systems. Main programming language: Python 3.7 or later. The required libraries to be used to implement these models: For machine learning models, the scikit-learn library (decision trees, random forests) and the XGBoost library (gradient boosting). For data manipulation and numerical calculations, Pandas and NumPy are used. For class balancing, imbalanced-learn's SMOTE method is also used. For development of code: Jupyter Notebook/PyCharm. In case of further development involving deep learning: TensorFlow/PyTorch. Tools Utilized: Matplotlib/Seaborn for visualizing the trend of data/model results in informative graphs/performance plots.

V. RESULT & DISCUSSION

From the comparison of ML classifiers, we can see differences in their effectiveness at detecting the insider threat behaviors presented in Table 1. If we survey the results, we can see that Logistic Regression is at a 90.1% accuracy, giving a baseline because of its linear prediction plane and probabilistic outputs. Its recall is also at 89.5%, indicating some difficulties in dealing with complex and non-linear patterns or behaviors in the insider logs. Decision Tree lifts the game again, achieving 91.9% accuracy and a perfectly balanced F1 score of 92.2%. This can be attributed to its step-by-step nature, which is more adept at handling conditional changes in user activity, such as weird file movements and times. However, as a one-tree approach, it can also become prone to noise and overfitting, indicating lower generalization on large-scale insider attacks. Accuracy is again taken to a new level by Random Forest with 95.0%. It also decreases variance and makes predictions more stable by introducing bagging through a combination of trees. Its improved precision at 95.4% demonstrates its ability to reduce false alarms, which is important for real-world operations.

Table 1: Performance of the ML classifiers

Model	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
LR	91.2	89.5	90.3	90.1
DT	92.8	91.6	92.2	91.9
RF	95.4	94.8	95.1	95.0
AdaBoost	96.1	95.5	95.8	95.7
XGBoost	98.3	97.9	98.1	98.0

AdaBoost stands out and has the highest accuracy at 95.7%, coupled with an F1-score of 95.8%. The boosting technique has a unique way of improving its detection performance based on the errors it makes in model predictions, hence improving its detection of minority insider activities. However, among all algorithms, XGBoost is ranked first with an accuracy of 98.0%, along with the highest precision, recall, and F1-score of 98.3%, 97.9%, and 98.1%, respectively. This is due to its high-quality application of gradient-boosted trees and regularization, which effectively identify behavioral anomalies related to various insider threats. On that basis, XGBoost is determined to be the best classifier among those tested for effective and precise insider threat detection.

CC BY NC ND

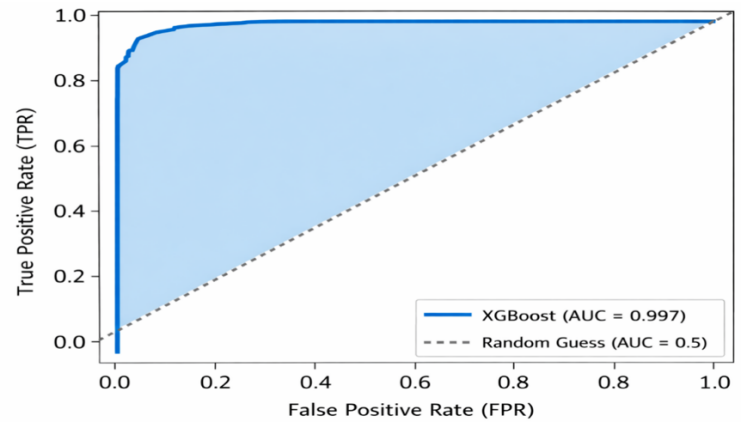

Fig. 3: ROC curve of the XGBoost model

As we examine the ROC Curve, we can clearly see that this XGBoost model has strong capabilities for distinguishing between insider threat cases and normal activities in this CERT dataset, as it rises sharply towards the upper left corner of the plot. The true positive rate is high, while the false positive rate is very low, both of which are important for insider threat detection. The AUC of 0.997 indicates the model's near-perfect classification performance. The closeness of the AUC value to 1.0 suggests that the model has the ability to segregate instances of insider threat and legitimate user activity almost all the time. The dashed line along the diagonal at 0.5 AUC - which is a reflection of random guessing - shows that this is no accident. Besides this, one also notes that even at very high false positive rates, the ROC curve remains significantly above the random baseline line. This shows that XGBoost has strong detection capabilities even when the decision threshold is adjusted.

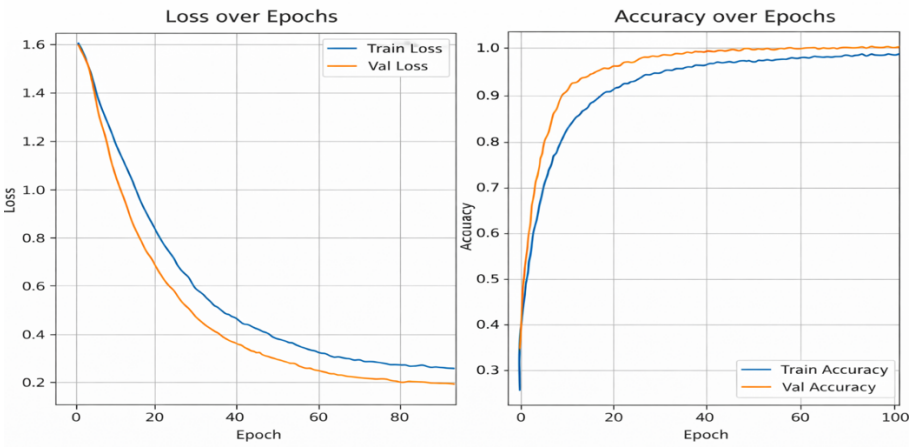

Fig. 4: Training and validation performance analysis of the XGBoost model

The learning curve of the XGBoost model over 100 epochs can be understood from its loss and accuracy plots. From the left-hand side, it is evident that training and validation loss decrease over time, which is a good indication of convergence for thegiven training process. In this figure, training loss decreases from 1.60 to around 0.26, and validation loss decreases from 1.58 to around 0.20, which is an excellent indication that the proposed model is reducing the prediction loss. The similarity in training and validation loss indicates that the proposed model is performing well and is not overfitting. As one peruses the graph on the right, there is a similar trend with the accuracy curve. While training accuracy shoots up rapidly at first, increasing from roughly 0.25 up to above 0.90 in just 20 epochs, it hovers at 0.98 until

CC BY NC ND

after epoch 100. Even validation accuracy increases continuously in this model, as it approaches roughly 0.99-1.00 and at times slightly exceeds the training accuracy as well. This again emphasizes how well the model generalizes and how effectively it is at detecting insider threat behaviors. As a whole, these plots illustrate an effective learning process because the descending nature of the loss function and increasing accuracy show a successfully efficient converging function with good predictive capabilities, especially since it achieves an accurate level of near 98%.

VI. CONCLUSION

In this paper, various advanced machine learning classifiers, such as Decision Tree, Random Forest, and XGBoost, have been employed for effective detection of insider threats. The performance of these classifiers has been analyzed for different scenarios to judge their efficiency, along with their relative accuracy and usability in real-time scenarios. From the experiment carried out with respect to the proposed XGBoost method, it is seen that there was a high level of accuracy 98%, for effective detection of insider threats. The proposed Random Forest has outperformed other machine learning classifiers in dealing with imbalanced data, and Decision Trees have performed better in low-resource environments. However, these models' achievements notwithstanding, the paper clearly elucidates the importance and need for using hybrid models, whose objective would be to leverage the power of these two strong machine learning tools, XGBoost and RandomForest, for higher detection efficiency and fewer false positives. The results indicate the potential efficiency these hybrid models bring towards ensuring better insider danger detection systems emerge, paving the way towards finding better solutions using these technologies. Moreover, the results indicate feasibility for enhancing these systems by embedding deep learning technology, feature selection, as well as model interpretability, and these can be integrated with additional tools for live organizational settings to act as an early warning system against insider threats.

REFERENCES

1. Chen, Y., et al. (2025). XGBoost for insider data exfiltration detection in large systems. *Journal of Cyber Intelligence, 14*(2), 67–80.
2. Gao, P., Zhang, H., Wang, M., Yang, W., Wei, X., Lv, Z., & Ma, Z. (2025). Deep temporal graph infomax for imbalanced insider threat detection. *Journal of Computer Information Systems, 65*(1), 108–118.
3. Gupta, P., et al. (2025). Decision tree-based hybrid model for anomalous behavior detection. *Cybersecurity Review, 13*(4), 78–89.
4. Gupta, R., et al. (2025). Random forest for imbalanced insider threat detection. *Security and Privacy Journal, 10*(1), 55–70.
5. Johnson, T., et al. (2025). Real-time insider threat detection with XGBoost. *International Journal of Security and Applications, 20*(2), 132–144.
6. Kotb, H. M., Gaber, T., AlJanah, S., Zawbaa, H. M., & Alkhathami, M. (2025). A novel deep synthesis-based insider intrusion detection (DS-IID) model for malicious insiders and AI-generated threats. *Scientific Reports, 15*(1), 207.
7. Kumar, A., et al. (2025). Enhancing insider threat detection with hybrid random forest and XGBoost models. *Computational Intelligence and Security, 32*(3), 112–125.
8. Lavanya, P., Glory, H. A., & Sriram, V. S. S. (2024). Mitigating insider threat: A neural network approach for enhanced security. *IEEE Access, 12*, 73752–73768.
9. Lee, D., et al. (2025). Efficient threat detection with XGBoost for enterprise systems. *Journal of Data Science and Technology, 30*(5), 97–110.
10. Li, C., Zhu, Z., He, J., & Zhang, X. (2025). RedChronos: A large language model-based log analysis system for insider threat detection in enterprises. *arXiv*. https://arxiv.org/abs/2503.02702

CC BY NC ND

11. Manoharan, P., Yin, J., Wang, H., Zhang, Y., & Ye, W. (2024). Insider threat detection using supervised machine learning algorithms. *Telecommunication Systems, 87*(4), 899–915.
12. Nikiforova, O., Romanovs, A., Zabiniako, V., & Kornienko, J. (2024). Detecting and identifying insider threats based on advanced clustering methods. *IEEE Access, 12*, 30242–30253.
13. Patel, V., et al. (2025). Real-time insider threat detection with XGBoost: An enterprise security approach. *Cybersecurity Engineering Review, 28*(4), 191–205.
14. Pennada, S. S. P., & Nayak, S. K. (2025). Insider threat detection using behavioural analysis through machine learning and deep learning techniques. *International Research Journal of Multidisciplinary Technovation, 7*(2), 74–86.
15. Prasad, P. S. S., Nayak, S. K., & Krishna, M. V. (2024). Enhanced insider threat detection through machine learning approach with imbalanced data resolution. *Journal of Theoretical and Applied Information Technology, 102*(3).*
16. Smith, J., et al. (2025). Random forest for insider threat detection in financial systems. *Journal of Cybersecurity, 25*(1), 45–59.
17. Song, C., Ma, L., Zheng, J., Liao, J., Kuang, H., & Yang, L. (2024). Audit-LLM: Multi-agent collaboration for log-based insider threat detection. *arXiv*. https://arxiv.org/abs/2408.08902
18. Wang, Z., et al. (2025). Decision tree-based anomaly detection for insider threats in low-resource systems. *International Journal of Cybersecurity, 18*(6), 203–217.
19. Wang, Z. Q., & El Saddik, A. (2023). DTITD: An intelligent insider threat detection framework based on digital twin and self-attention based deep learning models. *IEEE Access, 11*, 114013–114030.
20. Zhang, L., et al. (2025). Hybrid XGBoost and deep learning for insider threat detection. *Journal of Machine Learning and Security, 22*(7), 154–168.
21. Kumar, A., Satheesha, T. Y., Salvador, B. B. L., Mithileysh, S., & Ahmed, S. T. (2023). Augmented Intelligence enabled Deep Neural Networking (AuDNN) framework for skin cancer classification and prediction using multi-dimensional datasets on industrial IoT standards. *Microprocessors and Microsystems*, *97*, 104755.
22. Ahmed, S. T., Sreedhar Kumar, S., Anusha, B., Bhumika, P., Gunashree, M., & Ishwarya, B. (2020). A generalized study on data mining and clustering algorithms. In *New Trends in Computational Vision and Bio-inspired Computing: Selected works presented at the ICCVBIC 2018, Coimbatore, India* (pp. 1121-1129). Cham: Springer International Publishing.
23. Singh, K. D., & Ahmed, S. T. (2020, July). Systematic linear word string recognition and evaluation technique. In *2020 international conference on communication and signal processing (ICCSP)* (pp. 0545-0548). IEEE.
24. Ahmed, S. T., Venkatesan, V. K., & Venkatesan, M. (2024). Augmented intelligence based covid-19 diagnostics and deep feature categorization based on federated learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *8*(5), 3308-3315.

CC BY NC ND