IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

Artificial Intelligence (AI)- Based Threat Detection Models for Privileged App Abuse in Modern Systems

Maunik K. Shah
Independent Researcher
IEEE Senior Member
maunik.shah27989@gmail.com

Abstract: *The insider threat is becoming one of the most talked-about and pressing problems in cybersecurity. This occurrence highlights the need for specialized detection systems, methodologies, and tools to enable fast and accurate identification of malevolent insiders. This paper presents a strong detection system to use the CERT r5.2 dataset, which has been preprocessed with features that improve relevance through statistical aggregation, encoding, z-score normalization, and PCA-based dimensionality reduction. To increase identification of infrequent harmful behaviors, oversampling and class-weighted learning are used to overcome class imbalance. A convolutional neural network (CNN) trained in one dimension is used for automatic feature extraction. Experimental findings demonstrate near-perfect performance under all conditions, with an AUC value up to 1.000 and F1-score, recall, accuracy, and precision all above 99.9%. Results from comparisons with both conventional ML and other DL methods show that the suggested model is superior. This research provides a scalable solution to the problem of proactive insider threat detection and establishes that the framework is effective, resilient, and applicable in real-world security systems. A dependable solution for proactive insider threat identification, the framework is scalable, interpretable, and applicable to real-world security systems*

Keywords: Cybersecurity, Privileged App Abuse, insider Threat Detection, Android Security, Malware Detection, Permission Analysis, API Call Monitoring, Machine Learning

I. INTRODUCTION

Modern digital ecosystems rely on privileged access apps to power enhanced system features on mobile and desktop platforms. On the other hand, there are major security concerns with this increased access since attackers are finding new ways to use privileged apps to circumvent security measures and take over the system[1]. Because of this, privileged app misuse has become a major issue in cybersecurity, especially with the increasing complexity, frequency, and scope of cyberattacks[2][3]. Due to their reliance on static rules and signature-driven detection, modern security systems often fail to identify complex assaults that use permission abuse, dynamic behaviors, and obfuscation tactics[4][5]. This issue has been made all the more visible in applications of modern operating systems, such as Android, which heavily rely on API calls and permissions to interact with system services[6]. Maliciously inserted applications and even rogue applications often utilize these rights to carry out harmful activities masquerading as legitimate applications. Absence of systematic and comprehensive mapping of specified permissions to runtime API behaviour contributes further to large scale insider threat detection[7]. Despite the long-established research on detection methods that rely on permissions and APIs, the current set of solutions remains largely ineffective against advanced malware versions, which actively avoid both static and dynamic analysis, by code obfuscation and behaviour concealment[8].

To address these drawbacks, AI has become a significant topic of interest in cybersecurity studies. ML and DL permit the efficient processing of the large volumes of unstructured data by systems. AI-based models can be left to learn

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, January 2026 Impact Factor: 7.67

complex behavioural patterns independently[9]. This ability allows them to detect hidden and unnoticed cyber-attacks. These models achieve better detection accuracy by incorporating hybrid models such as permission utilisation, API call sequences, behavioural graphs, and execution traces[10]. Therefore, AI-based models of insider threat detection offer a proactive protection measure against the abuse of privileged applications[11]. The rising sophistication of detecting insider threats in environments with high rates of privileged access and complex user behaviours is the reason behind the suggested effort. The issues with highly skewed data and with the insidious attacks that resemble normal behaviour are the areas where the conventional detection systems are ineffective. The proposed solution to these issues is that to make more use of deep learning to identify outliers and know trends of behaviour based on information obtained on the basis of multiple sources on user actions. A dependable, scalable, and proactive solution for insider threat detection in current organisational systems is being looked for by the presented technique. This solution aims to identify and capture behavioural and temporal variation and enhance detection of rare threat situations. Here are provide some key contribution of paper;

- Uses statistical aggregation, encoding, normalization, and PCA to reduce redundancy and improve model learning efficiency.
- PCA and EDA highlight the most influential behavioural features, enhancing interpretability and guiding CNN input design.
- Applies oversampling and class-weighted learning to reliably detect rare malicious activities, improving recall.
- Automatically extracts temporal and behavioral features, outperforming traditional and deep learning baselines.
- Confusion matrices, ROC curves, and score distributions support the model's ability to distinguish between normal and abnormal events and aid in the decision-making process for architecture.
- Shows better accuracy, precision, recall and F1-score and AUC, which proves its strength and generalization.

The originality of this study is that it is able to determine common insider threats using multi-source user activity data to determine specific behavioural and temporal patterns. It is the only method that appropriately recognizes rare yet significant harmful activity by handling class imbalance. Through combination of deep learning and robust feature representation, the framework can reliably and generally distinguish between normal and abnormal behaviours. It has practical architecture and can be easily scaled and applicable in realistic security systems. It is a massive advancement compared with earlier methods, which fail to capture trivial or infrequent insider threats. Consequently, the cybersecurity of an organization is enhanced.

A. Paper Organization

The rest of this work is organised as follows. Current techniques for identifying internal threats are reviewed and evaluated in Section II. Section III outlines the general approach of the suggested system. Section IV presents and analyses the outcomes of the threat problem-solving ML models. Lastly, we wrap off with the conclusion found in Section V.

II. LITERATURE REVIEW

This section offers a current, thorough analysis of modern methods that use a variety of methodologies to identify insider threats.

Sadegh-Zadeh and Tajdini (2025) suggested method detects anomalies and potential cyber danger locations by combining geolocation computation with passive DNS information and K-means clustering. Moreover, compared to traditional DNS monitoring systems, our entropy-based anomaly detection system generated less false positives and more reliably detected risky DNS activity (92.3% accuracy). The geographical study reveals that 82 percent of cyberthreats originate in 15 high-entropy areas, and this is comparable to reports of cybersecurity incidents across the globe. The proposed predictive strategy significantly contributes to real-time threat awareness and response functions as it increases cyber threat detection[12].

Lokuliyana *et al.* (2025) explores the prospects of ML in strengthening IoT security through suspicious behaviour detection, incorporating behavioural biometrics into cloud-based dashboard security, and detecting botnet threats early.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

Researchers used open-source datasets to evaluate several ML algorithms, such as XGBoost, DT, Logistic Regression, and KNN. The Decision Tree model's superiority in handling complex security threats was proved by its excellent anomaly detection accuracy rate of 0.73. On the other hand, the XGBoost model showed high performance in identifying TCP SYN flood assaults with an accuracy rate of 92%[13].

Yi and Tian (2024) offers a way to improve supervised insider threat detection by using unsupervised outlier scores. Our technique outperforms existing top-notch anomalous detection methods in terms of performance and predictive capacity, thanks to our innovative methodology. They found an 86.12% accuracy with just 20% of the computational budget, which is an improvement of up to 12.5% over other anomaly detection algorithms using the same budget [14].

Mehmood *et al.* (2023) applies methods for ML to the problem of insider attack categorisation. The CERT dataset is used to create a customised dataset. The dataset is subjected to the analysis of four ML algorithms: LightGBM, XGBoost, Adaboost, and RF. In general, LightGBM was the most effective. With an accuracy of 97%, LightGBM outperforms all of the other algorithms suggested. RF comes in at 86%, AdaBoost at 88%, and XGBoost at 88.27% complete the list[15].

Grace and Sughasiny (2022) suggested concept is a small-footprint monitoring system that analyses app activity to find malice. The coded model keeps tabs on the app's actions by comparing them to the log file. These log files record permissions and app activity as it happens in real time. This model outperforms state-of-the-art methods like MO Droid, Crow Droid, and dl-Droid with an acquired accuracy of 95%. Consequently, the malicious program may be detected instantaneously using the suggested methodology by evaluating the policy-based permissions[16].

Akbar *et al.* (2022) using a variety of MLmodels to classify the apps as safe or harmful. In comparison to the current methods, the suggested approach achieved higher malware detection accuracies, namely 89.7 percent with SVM, 89.96% with RF, 86.25% with Rotation Forest, and 89.52% with NB models. In addition, the assessment metrics including Fmeasure, acc, sensitivity, and prec were improved, and the suggested strategy optimised around 77% of the feature set compared to current methods[17].

Research Gap

Despite the fact that machine learning algorithms have shown promising results in detecting cyber threats, anomalies, insider threats, and malware in recent research, there are still some gaps. Most of the previous work relies on static models like K-means, Decision Trees, or gradient-boosting to achieve high accuracy with single-domain datasets like DNS logs, IoT traffic, mobile applications, or CERT insider data. Research on integrated, cross-domain frameworks that include system-level, behavioural, and geographic data is lacking when it comes to thorough threat detection. An intelligent and hybrid detection system is necessary since important problems including real-time adaptation, generalisation to changing attack patterns, explain ability, scalability, and unified risk prioritisation have not been adequately solved.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

III. METHODOLOGY

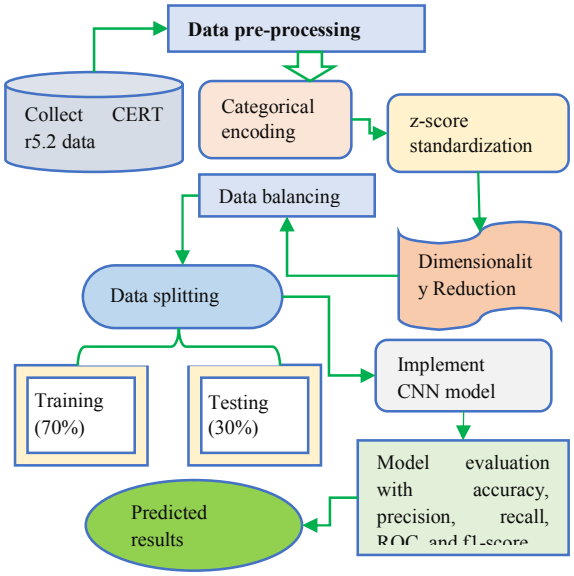

Fig. 1. Propose Flowchart for Threat Detection

This study makes use of the CERT r5.2 insider threat dataset, which spans 18 months and includes multi-source user activity records from 2000 workers. Figure 1 shows the implementation pipeline. The user's actions are collected once a week, numerically encoded, normalised with z-score transformation, and dimensionality reduced with principal component analysis (PCA). Data balancing strategies are used to correct class imbalance when the dataset is split 70:30 across the training and testing sets. Train a one-dimensional convolutional neural network (CNN) to recognise abnormal insider actions and learn behavioural patterns; measure its performance with AUROC, F1-score, recall, and accuracy.

All implementation steps of proposed framework are explained in below:

A. Data Gathering

This work makes use of the CERT insider threat dataset[1], which is open to the public and is used for studying, creating, and testing methods to reduce insider threats. The latest version of the dataset, CERT r5.2, mimics a company with 2000 workers over a whole year. Information about the organisation and its users, as well as logs for the following user actions (log on/off, email, web, file, and thumb drive connect), make up CERT r5.2's data collecting phase (III-A). The visual insights of datasets are given in below:

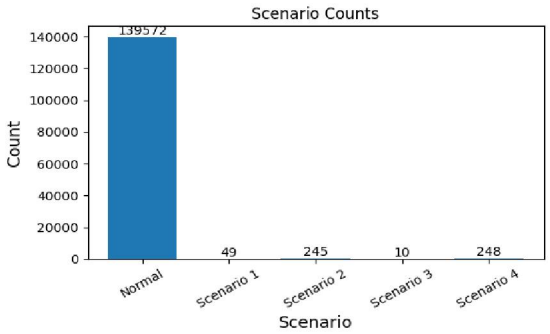

Fig. 2. Count plot for Scenarios Distributions

[1] https://www.kaggle.com/datasets/mrajaxnp/cert-insider-threat-detection-research?utm_source=chatgpt.com

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

The distribution of scenario is shown in Figure 2, which clearly shows a significant class imbalance. There are 139,572 occurrences in the "Normal" category, whereas the occurrences of the anomalous situations (Scenario 1–4) range from 10 to 248. This disparity emphasises the necessity for strong anomaly detection methods by demonstrating the difficulty of identifying uncommon threat occurrences in a dataset that is mainly normal.

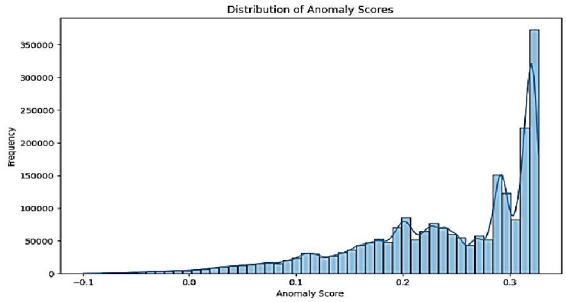

Fig. 3. Anomaly Score distributions

Figure 3 shows the overall distribution of anomaly scores, which is right-skewed and mostly concentrated in the upper half. A huge number of events with high anomaly likelihood are indicated by the frequency peak around a score of 0.32. The scoring mechanism's ability to distinguish between typical and unusual behaviour is demonstrated by this distribution, which also indicates the existence of clear outliers.

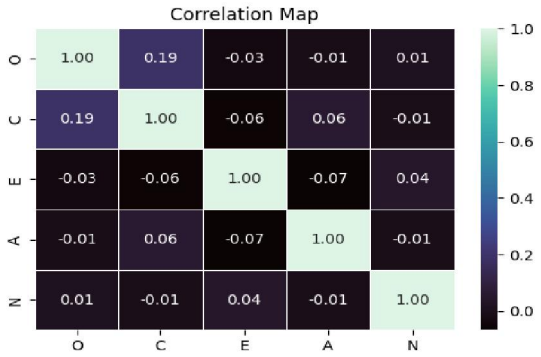

Fig. 4. Correlation Heatmap of Features

Figure 4 shows the relationships among each of the five features. From -1 to 1, the PCC shows the relationship among the two features in each cell. Element diagonals exhibit complete self-correlation (1.00), but off-diagonal values, which are often close to zero, imply poor inter-feature connections. It is worth mentioning that the connection between O and C is the highest at 0.19, while the correlation between E and A is the strongest at -0.07. The varying shades of green graphically highlight the strength of these correlations, going from dark (low correlation) to bright green (high correlation).

B. Data pre-processing

The pre-processing pipeline starts by consolidating multi-source user activity data (logon, device, email and file) and LDAP user data and psychometric characteristics by the week per-user. Categorical data are coded with numbers and behavioural clues are statistically summarized. Insider threat labels are added, and z -scores are used to add features. Lastly, PCA of dimensional reduction. These steps are explained in below:

C. Categorical Encoding

Data encoding transforms heterogeneous insider threat records and user data into numerical data by encoding attributes such as department or role, summarizing behavioural indicators (logins, USB use, file access, emails, HTTP requests) using a statistical representation and combining psychometric characteristics and threat labels into one coherent

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

structured feature set that is prepared to undergo normalization and transformation.

D. Z-score normalization

The input dataset was normalised such that all feature values were within the same range. The z-score normalises features so that their values follow the normal distribution with a mean of 0 and a StandardDeviation of 1 unit. Equation (1) represents the z-score.

$$z(x) = (x[:, i] - \mu_i)/\sigma_i \quad (1)$$

where μ_i= mean of the ith feature, σ_i= Standard Deviation of the ith feature.

E. PCA for Dimensionality Reduction

Principal Component Analysis (PCA) helps find the key characteristics and further decrease the data's dimensionality. Figure 5 shows the feature weight ratios for the fifteen most important characteristics that made up the first principal component (PC1). The ability to identify unusual actions relies heavily on these characteristics. By gathering complex behavioural patterns that distinguish between genuine users and malicious behaviour, this strategy enhances the model's ability to detect insider threats.

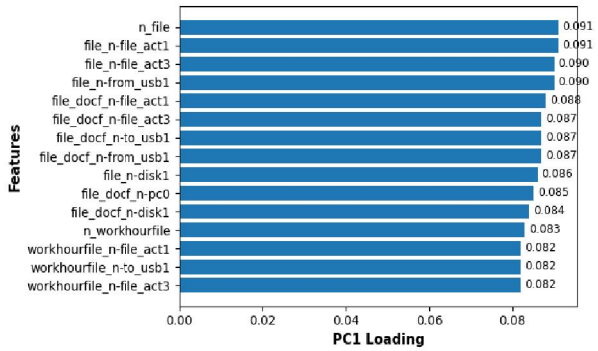

Fig. 5. Top 15 features contributing to principal component

Figure 5 shows the top 15 features which were used in creating PC1, the first principal component in PCA. The loading values of the features are shown by the bars, which show the relative effect of each feature on PC1. With loading values around 0.09, the most significant features are n_file, file_n-file_act1, and file_n-file_act3. These features strongly contribute to the variance collected by PC1. There are also significant factors that indicate patterns in how people use devices and manage files. The main component can be better understood with the help of this visualisation, which highlights the most important behavioural indications.

F. Data Balancing

This research paper uses the CERT v5.2 dataset, which has more than 139,000 normal records but few hundreds of malicious cases in all the scenarios, and this may skew the model towards normal behavior prediction. In order to cope with this, heuristic methods like oversampling minor classes (e.g. by replicating or even synthetically creating malicious samples), under sampling major classes (cutting back on normal samples) or class-weighted loss functions are used. These techniques ensure the CNN is balanced in its training signals, enabling it to identify rare insider threats more effectively without compromising overall accuracy and recall.

G. Proposed CNN Model

CNNs are the building blocks of deep learning. A pooling layer and one or more convolutional layers stacked on top of each other make up a convolutional neural network's architecture. A fully linked layer and a classification layer are both incorporated in these cases. The suggested CNN model is built to efficiently train discriminative features from (50,6)-shaped sequential input data, where each sample has 50 time steps and 6 feature channels. Figure 6 shows the architecture's initial step, a Conv1D that uses 64 filters to extract local temporal patterns from the input and generates a

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

feature map with dimensions (49,64). Then, to improve computational performance and save the most important characteristics, a MaxPooling1D layer is used, reducing the temporal dimension to (24,64). After that, a fully connected dense layer consisting of fifty neurones is given the flattened output in order to learn more complex representations. A single-neuron output layer then generates the prediction score which facilitates the process of efficient binary classification. The CNN model is suitable in identifying complex patterns associated with the harmful behaviour in modern security applications due to its hierarchical structure that allows it to automatically process and enrich both the low-level and high-level information.

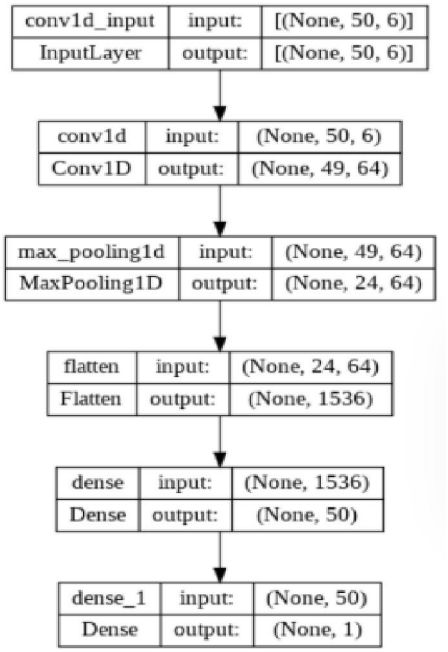

Fig. 6.CNN Model Training Architecture

H. Evaluation Parameters

A comparison of the model effectiveness using various classification metrics, which include: recall, accuracy, precision, F1score and AUC. The counts are TP, TN, FP and FN. The metrics are specified as Equation (2) to (5)

$$Accuracy = \frac{TP+TN}{TP+Fp+TN+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1-Score = \frac{2(Precision*Recall)}{Precision+Recall} \quad (5)$$

Accuracy is a statistic that measures whether a model is effective in predicting the outcomes; precision is a statistic that measures how many suspect wallets a model can identify without FP. Recall or sensitivity measures the efficiency of the model in reducing false negatives by detecting actual questionable cases. F1-Score balances precision and recall; hence, it is suitable in the case of imbalanced datasets. AUROC measures the effectiveness of classifications at every threshold, which provides an overall picture of model discrimination.

IV. RESULTS AND DISCUSSION

The tests were performed in the Lenovo legion pro Core i9-13900HX PC with Windows 10 and a memory of 32 GB RAM and a processor frequency of 3.90 GHz. To increase the processing speed, we utilized the NVIDIA GeForce RTX 4070 GPU. Many Python libraries, such as Pandas, Numpy, TensorFlow, and Keras were used. The proposed CNN-

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

based threat detection model is highly effective and it is demonstrated in Table I. The model has a high accuracy of 99.97, meaning that it can categorize data rather well. A high F1-score suggests a balanced relationship between recall and precision; the high recall shows that the model can detect nearly all cases of threats. Collectively, these findings demonstrate the CNN model's usefulness in threat identification.

Table 1: Performance results of proposed model for threat detection

Measures	CNN (%)
Accuracy score	99.97
Precision score	99.98
Recall score	99.99
F1 − Score	99.97

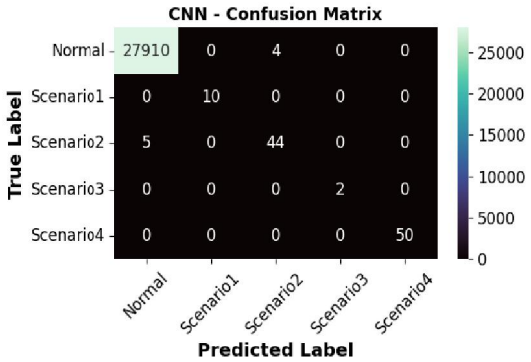

Fig. 7. Plot Confusion Matrix Of CNN Model

Figure 7 illustrated the confusion of the CNN model, which showed its classification performance in the Normal class, and four different scenarios. The majority of the samples are correctly identified along the diagonal proving a high degree of predictive precision. The Normal class has very small misclassifications, and the majority of them are to Scenario2. The scenario classes are all classified with near-perfect or perfect accuracy with low confusion. In general, the confusion matrix confirms the validity and reliability of the CNN model in distinguishing between normal and scenario conditions.

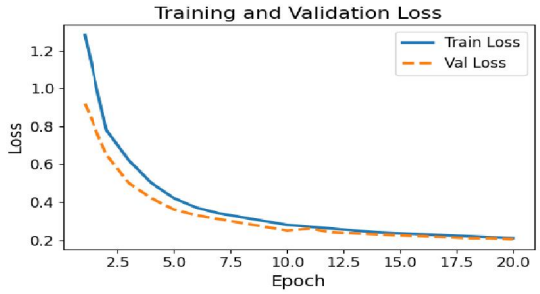

Fig. 8. Plot Accuracy Graph Of CNN Model

Figure 8 showed the training and validation performances of CNN model with different epochs, and a pattern exists towards reducing loss as the algorithm learns. Validation loss is very near to training loss, which indicates strong generalisation and no overfitting. In the early epochs, loss drops dramatically, but in later epochs, it levels out. In general, the figure shows that the CNN model learns well and maintains a consistent performance level as training progresses.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

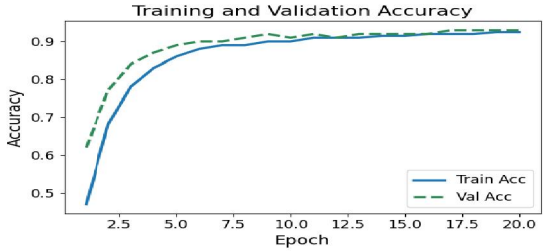

Fig. 9. Plot loss Graph Of CNN Model

Figure 9 displayed the accuracy of the CNN model throughout 20 epochs, both during training and validation. As a measure of successful learning, the training accuracy rises in the first few epochs before levelling off at around 90%. Good generalisability is indicated by the fact that the validation accuracy curve follows the training curve with small gap. In general, the model shows no signs of overfitting and consistent convergence.

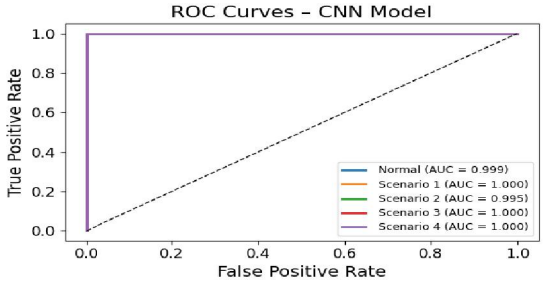

Fig. 10. Plot ROC Curve of CNN Model

As seen in Figure 10, the suggested CNN model performed well across all five classes. An extremely low FPR and a high TPR are shown by the ROCcurves' proximity to the top-left corner of the figure. The CNN model's excellent discriminative power is demonstrated by the achieved AUC values of about 0.999 for the Normal class, 1.000 for Scenario1, 0.995 for Scenario2, and 1.000 for Scenario3 and Scenario4. These excellent AUC values demonstrate that the model can reliably and robustly detect threats in current security systems, differentiating among benign and malicious circumstances.

A. Comparison and Discussion

Table II makes the comparison of a variety of models on CERT data to identify threats, showing that there is quite a significant difference in performances. Conventional Logistic Regression (LR) demonstrates a slight accuracy (70%) and sophisticated ensemble techniques such as XGBoost and AdaBoost demonstrate high balance in terms of metrics. The models of deep learning differ: LSTM shows lower recall (48.91) in comparison with Autoencoder (AE) with high recall (96). Competitive scores are observed with a bit lower scores in generative AI. The top performer is CNN which has almost perfect results on all the measures of its better ability to detect the threat in this dataset.

Table 2: COMPARATIVE ANALYSIS ON CERT DATA FOR THREAT DETECTION

Model	Accuracy	Precision	Recall	F1 Score
LR[18]	70	80	70	67
XGBoost[19]	90.89	91.77	89.10	90.41
LSTM[20]	93.14	66.26	48.91	56.28
AdaBoost[21]	94	98	94	95
Gen AI [22]	89.1	86.3	85.4	85.8
AE[23]	92	94	96	96
CNN	99.97	99.98	99.99	99.97

The key strength of the study is the combination of a useful pre-processing pipeline and a very precise CNN-based insider threats detection model. The proposed solution includes solving the issue of class imbalance, using PCA to

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, January 2026 Impact Factor: 7.67

optimize features, and deep learning to provide almost perfect results in all measures of evaluation. The high generalization rate, low false classification, and high performance in comparison to the known models prove the high strength and practical implementation of the proposed structure in real-world insider threat detection.

V. CONCLUSION AND FUTURE WORK

As the cybersecurity environment gets more difficult, insider threat detection has emerged as an important study topic. Conventional ML and DL models, although effective for recognising common threats, fail to detect malevolent insiders. The goal of this project was to create an effective insider threat detection system that would reliably identify aberrant behaviours using a CNN-based deep learning model. The proposed solution has a high detection rate, with an accuracy of 99.97 and a AUC of 1.000 on scenario classes, thus performing well in detecting rare insider threats. It surpassed other traditional models such as Logistic regression (70% accuracy) and LSTM (48.91% recall). The CNN model was robust and generalized, with equal training and validation curves and a minimum of misclassifications. However, the study is limited by the fact that the CNN architecture and the simulated dataset used were insufficient to reflect the complexity of insider threats in the real world. Future research ought to investigate hybrid or explainable AI approaches, test the model in real organisational data, and apply adaptive or federated learning methods to scale by making use of the approach, enhance its interpretability and practical application in dynamic security contexts.

REFERENCES

[1] A. R. Bilipelli, "Forecasting the Evolution of Cyber Attacks in FinTech Using Transformer-Based Time Series Models," *Int. J. Res. Anal. Rev.*, vol. 10, no. 3, pp. 383–389, 2023.

[2] V. Prajapati, "Enhancing Threat Intelligence and Cyber Defense through Big Data Analytics: A Review Study," *J. Glob. Res. Math. Arch.*, vol. 12, no. 4, 2025.

[3] S. K. Chintagunta and S. Amrale, "Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation," *Tech. Int. J. Eng. Res.*, vol. 9, no. 10, pp. 49–55, 2022, doi: 10.56975/tijer.v9i10.159996.

[4] N. K. Prajapati, "Federated Learning for Privacy-Preserving Cybersecurity: A Review on Secure Threat Detection," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 4, pp. 520–528, Apr. 2025, doi: 10.48175/IJARSCT-25168.

[5] S. Narang and G. K. V, "Next-Generation Cloud Security: A Review of the Constraints and Strategies in Serverless Computing," *Int. J. Res. Anal. Rev.*, vol. 12, no. 3, 2025, doi: 10.56975/ijrar.v12i3.319048.

[6] H. Liang, J. Chen, Y. Leng, X. Guo, and J. Luo, "Overview of Mobile Security Detection Technologies in Internet Traffic," in *ACM International Conference Proceeding Series*, 2024. doi: 10.1145/3689236.3691493.

[7] S. Narang and A. Gogineni, "Zero-Trust Security in Intrusion Detection Networks: An AI-Powered Threat Detection in Cloud Environment," *Int. J. Sci. Res. Mod. Technol.*, vol. 4, no. 5, pp. 60–70, Jun. 2025, doi: 10.38124/ijsrmt.v4i5.542.

[8] G. Sarraf and V. Pal, "Autonomous Threat Detection and Response in Cloud Security: A Comprehensive Survey of AI-Driven Strategies," *Int. J. Emerg. Res. Eng. Technol.*, vol. 6, no. 4, 2025, doi: 10.63282/3050-922X.IJERET-V6I4P114.

[9] R. Patel, "Automated Threat Detection and Risk Mitigation for ICS (Industrial Control Systems) Employing Deep Learning in Cybersecurity Defence," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 06, pp. 584–591, 2023, doi: 10.14741/ijcet/v.13.6.11.

[10] V. Shewale, "Demystifying the MITRE ATT&CK Framework: A Practical Guide to Threat Modeling," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 3, pp. 182–186, May 2025, doi: 10.32996/jcsts.2025.7.3.20.

[11] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, IEEE, 2024, pp. 1–6. doi: 10.1109/ACET61898.2024.10730334.

[12] S.-A. Sadegh-Zadeh and M. Tajdini, "An unsupervised machine learning approach for cyber threat detection

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, January 2026
Impact Factor: 7.67

using geographic profiling and Domain Name System data," *Decis. Anal. J.*, vol. 15, p. 100576, Jun. 2025, doi: 10.1016/j.dajour.2025.100576.

[13] S. Lokuliyana, A. G. A. Kalupahanage, H. M. S. D. Herath, D. Siriwardana, D. N. Bulathsinhala, and H. M. T. M. Herath, "Enhancing IoT Resilience: Machine Learning Techniques for Autonomous Anomaly Detection and Threat Mitigation," *Procedia Comput. Sci.*, vol. 254, pp. 68–77, 2025, doi: 10.1016/j.procs.2025.02.065.

[14] J. Yi and Y. Tian, "Insider Threat Detection Model Enhancement Using Hybrid Algorithms between Unsupervised and Supervised Learning," *Electronics*, vol. 13, no. 5, p. 973, Mar. 2024, doi: 10.3390/electronics13050973.

[15] M. Mehmood, R. Amin, M. M. A. Muslam, J. Xie, and H. Aldabbas, "Privilege Escalation Attack Detection and Mitigation in Cloud Using Machine Learning," *IEEE Access*, vol. 11, pp. 46561–46576, 2023, doi: 10.1109/ACCESS.2023.3273895.

[16] M. Grace and M. Sughasiny, "Behaviour analysis of inter-app communication using a lightweight monitoring app for malware detection," *Expert Syst. Appl.*, 2022, doi: 10.1016/j.eswa.2022.118404.

[17] F. Akbar, M. Hussain, R. Mumtaz, Q. Riaz, A. W. A. Wahab, and K.-H. Jung, "Permissions-Based Detection of Android Malware Using Machine Learning," *Symmetry (Basel).*, vol. 14, no. 4, p. 718, Apr. 2022, doi: 10.3390/sym14040718.

[18] S. S. P. Pennada, S. K. Nayak, and M. V. Krishna, "Insider Threat Detection Using Behavioural Analysis through Machine Learning and Deep Learning Techniques," *Int. Res. J. Multidiscip. Technovation*, vol. 7, no. 2, pp. 74–86, Mar. 2025, doi: 10.54392/irjmt2527.

[19] W. Jiang, Y. Tian, W. Liu, and W. Liu, "An Insider Threat Detection Method Based on User Behavior Analysis," in *IFIP Advances in Information and Communication Technology*, 2018, pp. 421–429. doi: 10.1007/978-3-030-00828-4_43.

[20] D. Bhandari and K. Pudashine, "Insider Threat Detection using LSTM," *J. Sci. Technol.*, vol. 3, no. 1, pp. 57–65, Dec. 2023, doi: 10.3126/jost.v3i1.69066.

[21] B. Bin Sarhan and N. Altwaijry, "Insider Threat Detection Using Machine Learning Approach," *Appl. Sci.*, vol. 13, no. 1, p. 259, Dec. 2022, doi: 10.3390/app13010259.

[22] P. S. S. Prasad, S. K. Nayak, and M. V. Krishna, "Hybrid Machine Learning for Enhanced Insider Threat Detection Using Generative Latent Features," *Int. J. Eng. Trends Technol.*, vol. 73, no. 6, Jun. 2025, doi: 10.14445/22315381/IJETT-V73I6P110.

[23] E. Pantelidis, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Insider detection using deep autoencoder and variational autoencoder neural networks," *arXiv Prepr. arXiv2109.02568*, 2021.