Tuniki Pravalika, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science,
Engineering and Technology
An Open Access Journal

Machine Learning-Based Insider Threat Detection For Enhanced Organizational Security

[1]Tuniki Pravalika,[2]Panuganti Lavanya

[1]Assistant Professor, [2]M.Tech Student, Department of CSE,

Megha Institute of Engineering and Technology for Womens, Edulabad (Village),

Ghatkesar (Mandal), Medchal District, Telangana

Abstract—The potential compromise of sensitive information and company assets by employees is a major concern for any firm. Robust ML algorithms that can handle complicated and biassed data are necessary for the threat detection process. Some of the ML models that are tested in this study using the well-known CERT dataset include Logistic Regression, Decision Trees, Random Forest, SVM, KNN, Naïve Bayes, Adaboost, and XGBoost. Approaches like SMOTE, which deal with problems brought on by data imbalance, emphasise the need of a balanced dataset. A 97.5% success rate in detecting insider threats was achieved using Random Forest and Adaboost, according to the data. This study lays the groundwork for more trustworthy organisational security measures by improving approaches for identifying insider threats and offering a systematic evaluation of model performance. A few of the terms that come up include SMOTE, CERT, insider threat detection, and machine learning.

Keywords: Insider Threat Detection, Machine Learning, CERT Dataset, SMOTE, Random Forest, AdaBoost, Cybersecurity, Anomaly Detection.

I. INTRODUCTION

Even today, insider threats are among the most challenging and potentially catastrophic hazards that companies encounter. Insider risks originate from trusted individuals inside the organisation who have access to confidential information or resources, as opposed to external threats that originate from beyond the company's boundaries. Data breaches, monetary loss, and damage to reputation are just some of the potential outcomes of insider assaults.

It may be challenging to detect insider activities, particularly when they imitate real user behaviours, due to their subtlety. Old habits and conventions rely on by traditional cybersecurity solutions allow them to overlook internal threats. This makes it harder for them to notice potentially harmful suspicious behaviour. The good news is that ML methods have just surfaced as an option, enabling the identification of subtle patterns of activity that might point to insider threats. The CERT dataset and other insider threat datasets show a very skewed proportion of good and bad behaviour. This disparity increases the risk that ML systems may develop biases against minority groups, rendering them unable to identify insider threats that are really associated with such groups. Logistic Regression, Decision Tree, Random Forest, SVM, KNN, Naïve Bayes, Adaboost, and XGboost are some of the machine learning methods that are used to the CERT insider threat dataset in this work. The objective is to learn their process for handling biassed data and identify any potential insider risks.

Login timings, file access patterns, and suspicious user activity are important behavioural aspects that help detect potential threats. Detailed descriptions of the algorithms used in this enquiry may be found below. First Section: Logistic Regression A statistical model that, given certain attributes, may be taught to make binary predictions.

One way LR might help find insider threats is by monitoring user behaviour, including as login and file access habits. When working with large datasets, LR shines, because it provides a simple framework for making probabilistic decisions. B. Tree of Decisions By continually partitioning the dataset based on feature values, this technique generates a tree-like model of decisions and their outcomes.

It has a tendency to overfit when presented with complex data, such as insider threat situations, yet it is great at identifying significant patterns of conduct.

Decision Tree (C): A method for ensemble learning that uses a combination of decision trees to improve prediction accuracy and prevent overfitting. RF's noise-resistance makes it a good fit for high-dimensional data, which covers a wide range of insider activity traits. Cluster D. Support Vector Machines They are excellent classifiers, therefore choose the optimum hyperplane that links the various data kinds.

While support vector machines (SVMs) are capable of handling complex, nonlinear user behaviour patterns, they may need to rectify imbalanced data in order to identify insider threats. Nonparametric feature space data point categorisation based on the principal class of adjacent points is known as K-Nearest Neighbours (KNN) and was developed by W. E. KNN.

By comparing user activity patterns to dataset objects, the KNN algorithm might potentially identify insider risks. A Naïve Bayes's Assuming feature independence, this probabilistic classifier applies Bayes' theorem. Contrary to appearances, NB is quite effective when dealing with high-dimensional data. For insider threat detection. We have Advaboy G. as our author. Assuming feature independence, this probabilistic classifier applies Bayes' theorem.

Contrary to appearances, NB is quite effective when dealing with high-dimensional data. For insider threat detection. Experienced physician XGboost Assuming feature independence, this probabilistic classifier applies Bayes' theorem. Contrary to appearances, NB is quite effective when dealing with high-dimensional data. For insider threat detection.

Here is the paper's structure: Section III lays out the proposed technique after Section II provides a concise review of the relevant literature. In Section IV, we provide the results of the experiments. Lastly, recommendations for future studies are provided in Section V.

II. LITERATURE SURVEY

With more and more people using the internet, insider threats are becoming more important, as pointed out by Sandra G et al. [1]. They did point out that despite many security measures, some insiders still manage to utilise their access for personal benefit, often undetected. In order to help academics create better threat detection systems, the article examined several detection methods, datasets, and performance metrics. The objective of analysing prediction algorithms in [2] was to detect insider threats in company networks. Out of 531 publications that used ML and DL, only 59 underwent a thorough review. Of this group, 58.8% offered original models, 23.5% discussed evaluation and execution, and 17.6% offered zero metrics per se.

Network traffic analysis routinely employs RNNs and SVMs to detect potential threats, as seen in the study's results. In order to evaluate ML models' ability to detect insider dangers, the authors of [3] laid forth a method. Logistic Regression, XGBoost, SVM, Random Forest, KNN, and MLP were some of the approaches that were used to analyse the CERT dataset. To determine which model was superior, we compared them using crucial performance criteria. Across all assessed granularities, XGBoost exposed more insider risks than its rivals. To identify insider risks, the authors presented a hybridMLDL method in [4]. Researchers found small insider risk behaviours using a model that used deep neural networks and patterns that were built using features.

The model outperformed state-of-the-art methods by assessing user behaviours and system information from several companies. This study sought to identify insider risks by analysing supervised, unsupervised, and reinforced machine learning [5]. Even with the right guidance, an unsupervised system nevertheless performed poorly in the evaluation. Machine learning

has the potential to help identify insider threats, but the study indicated that it works best when combined with other security measures, such organisational and physical safeguards. As stated in [6], dealing with insider threats in cybersecurity is becoming more complex. They stressed the requirement of targeted detection processes and technologies due to the difficulty of identifying insider threats. The paper included a thorough review of where insider threat detection using machine learning algorithms stands at the moment, showcasing several methodologies while also discussing their limitations and difficulties.

As stated in [7], the authors advocated for the use of an ML-based approach to enhance business information security via the detection of insider threats. The article began by defining insider threats and then went on to evaluate the dangers they pose before finally outlining the results of previous studies. Following the execution of the method's empirical testing, the study summed up by describing its advantages, disadvantages, and possible future research paths. The authors of [8] brought attention to the requirement of customised detection methods and insider threats in cybersecurity. They proposed a machine learning-based approach after a literature review revealed biases and an absence of real-world data. Research recommendations for threat detection followed a discussion of methodology and evaluation metrics. The authors of [9] suggested a new ML model to handle insider concerns using the XGBoost algorithm.

They proved that XGBoost may potentially outperform previous detection algorithms when hyperparameter tuned. The research also included the development of a web app using Flask to aid in the detection of insider threats using user input. To evaluate user actions and control insider threats, the writers of [10] used feature sequences and a tree structure. They were able to locate questionable users and feature sequences with the use of this and Copula Based Outlier Detection (COPOD). Using the CERT-IT dataset, they compared their method to Isolation Forest to see how well it detected suspicious activity. To differentiate between harmful and harmless insider threats, [11] used user behaviour analysis. They reduced the number of false positives while increasing the accuracy of detections by using deep learning. Discovering insider threats required the development of a new Random Forest algorithm, RF-RWFF [12]. It reduces computing cost while outperforming classic Random Forest in terms of accuracy using Fuzzy Membership Functions and a Randomised Weighted Majority Algorithm. This approach may help find insider threats more effectively. To address issues like data imbalance and changes in behaviour, the authors of [13] introduced ML-based insider threat detection. With a very low false positive rate of 0.78%, the system attained an impressive accuracy rate of 85%.

III. RESEARCH METHODOLOGY

The suggested technique is shown in Figure 1. Gathering data and becoming ready are the first steps in detecting insider threats. We included a vast amount of data that included environmental and behavioural factors, including positive and negative actions. In order to be ready to train the model, we encoded the categorical variables and checked for missing values. To rectify the disparity in class representation, synthetic samples were generated for the minority class using the Synthetic Minority Over-sampling Technique (SMOTE). Incorporating data from both sets of people could help machine learning models understand patterns without being skewed by oversampling. Due to the rarity but often-impactful nature of insider assaults, SMOTE's distribution equalisation improved the models' capacity to identify them.

When choosing our model, we looked at how well various machine learning algorithms performed and how easily they could be understood. Classes including XGBoost, Decision Trees, Logistic Regression, Random Forest, Naive Bayes, and K-Nearest Neighbours (KNN) were among those used. The interpretability of Logistic Regression and the

resistance to overfitting in Random Forest are two of the key features that were taken into account while making the decision. The process also placed a premium on hyperparameter optimisation. Each model's hyperparameters were fine-tuned using grid search and crossvalidation. For context, Random Forest used one hundred trees, Decision Trees divided data according to the Gini impurity criteria, and KNN settled on k=5. In contrast to XGBoost, whose performance was fine-tuned using parameters like max depth, Adaboost's performance was enhanced by careful tuning of its learning rate. With an emphasis on ensemble approaches—which have shown to be very successful in complicated prediction problems—this optimisation strategy sought to optimise the performance of each algorithm.

We divided the dataset in half to train and test the model: 80% for training and 20% for testing. Following training on the training set, many measures were used to assess each model's performance, including as F1-score, recall, accuracy, and precision. By comparing and contrasting the models, we can see how well they do at anticipating insider threats and where they fall short. The unique properties of ensemble methods, such as Random Forest, Adaboost, and XGBoost, make them very successful at detecting insider threats.

By averaging the predictions of many decision trees, Random Forest reduces the likelihood of overfitting and generates a resilient model that performs well when given fresh data. To enhance performance in challenging scenarios, Adaboost dynamically adjusts the weights of misclassified samples, with a focus on difficult-to-classify cases. When it comes to efficiency and speed, XGBoost is unrivalled. Employing regularisation techniques inside a gradient boosting framework, it improves accuracy and prevents overfitting.

IV. EXPERIMENTS AND RESULTS

Collecting Data and Creating Sets The insider threat detection community relies on the CERT 5.2 dataset, which was obtained from GitHub and is utilised extensively in this work. This dataset illustrates a wide range of human actions and system events in an organisation using 830 attributes per record. It has 693,649 records in total. Tools such as'starttime' and 'endtime' track the beginning and ending times of events, while markers such as 'day', 'week', 'isweekday', and 'isweekend' dictate whether events occur within or outside of typical business hours. Project, user, role, business unit, functional unit, department, and team are some of the phrases used to assist put the activities in perspective within the larger organisational and project-specific frameworks.

The conduct of users across various time periods makes up a large chunk of the data gathered. For example, although 'n_allact' shows the total number of activities, 'n_workhourallact' and 'n_afterhourallact' slice it even further to show actions that happen during work hours and those that happen after hours, respectively. In order to better understand the kind and level of intensity of these activities, the counts are further broken down using principal components ('pc0' through 'pc3'). Features like 'n_logon' keep track of login habits; it divides the overall count of logons into two groups: those that occur during business hours and those that occur outside of work. A large chunk of the collection consists of human behaviours involving computers and device interfaces, especially those using USB. Features such as 'n_usb', 'usb_mean_usb_dur', and 'usb_mean_file_tree_len' assess the complexity of the file structure during each connection, the average duration of each USB connection, and the number of connections overall.

There is a clearer picture of possible insider risks when we look at both work-related and non-work-related activities. Managing files is also a crucial part of gathering data. An all-encompassing picture of file interactions may be obtained from the 'n_file,' 'file_mean_file_len,' 'file_mean_file_depth,' and 'file_mean_file_nwords' properties. Information such as disc access, file activity, data transfers to and from USB devices, and file formats including "Compf," "phof,"

"DocF," "Txtf," and "Exef" are all part of this area. The data gathering system may identify any deviations from the norm by differentiating between file activities performed during and outside of work hours.

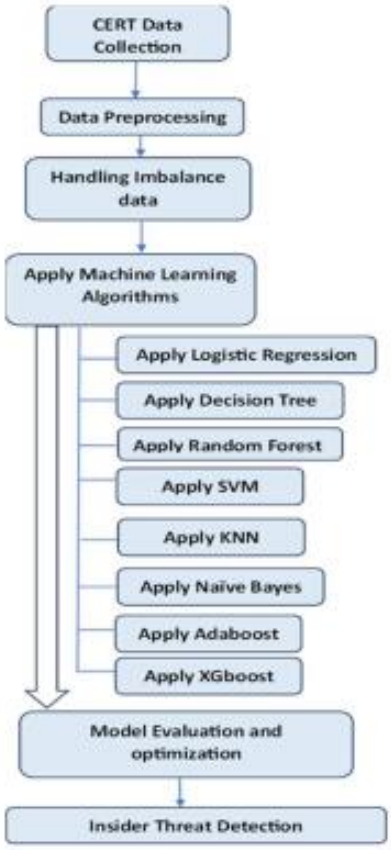

Fig. 1. Proposed method

The "insider" labelled variable is the one to aim for in this dataset since it indicates if a certain event presents an insider threat. With so many options, insider threat detection systems that rely on machine learning can do their jobs much better. After the data was collected, the dataset was examined for any outliers or missing values, but none were found. Additional ML-based analysis might be built using this data. Dealing with Class Inequality Issue B.

The great majority of samples are not insider threats; just 1,307 out of 693,649 samples are. This large disparity shows that dealing with biassed data becomes very difficult when insider risks are underrepresented. Due to its bias in training on the majority class, the model may fail miserably when tasked with identifying insider threats or other abnormalities in an unbalanced dataset. Through the use of SMOTE, data for minority classes was generated, and the model's capacity to learn from rare occurrences of insider threats was artificially amplified. The method improved insider threat identification, an important but seldom task. Thirdly, we combined the capabilities of many ML classifiers to employ ML models for insider threat prediction. For effective binary categorisation, the Logistic Regression default parameters were used.

Using the Gini impurity criterion, decision trees ensured that their outcomes were straightforward. We might use Random Forest with one hundred trees (n_estimators=100) to get more accurate predictions. We chose KNN with k=5 since it is quite good at detecting local patterns. Naive Bayes (NB) uses a Gaussian distribution to describe the feature probabilities, which guarantees effective performance with high-dimensional data. Its 50 estimators and 1.0 learning rate setup made boosting weak learners a simple using Adaboost. Last but not least, XGBoost with a maximum depth of 6 and a learning rate of 0.1 could readily handle complex decision boundaries. When these models are combined, they provide a comprehensive system that can detect insider threats.

Table I. Accuracy Comparison

Model	Accuracy	Precision	Recall	F1 score
Logistic Regression	58%	57%	57%	58%
Decision Tree	97%	98%	97%	97%
Random Forest	97.5%	97%	96%	97%
KNN	90%	90%	89%	90%
SVM	86%	86%	85%	85%
Naïve Bayes	75%	76%	74%	74%
Adaboost	97.5%	97%	97%	97%
XGboost	97%	97%	96%	96%

To help the models understand which patterns of behaviour pose an insider threat and which do not, the SMOTE approach might be useful; this strategy handles data imbalance. There are 80% training

samples and 20% testing samples. You may see measures of ML models' efficacy in Table I.

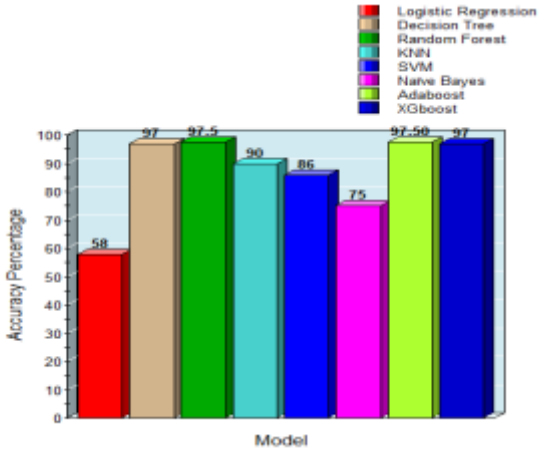

Fig. 2. Accuracy comparison of ML models

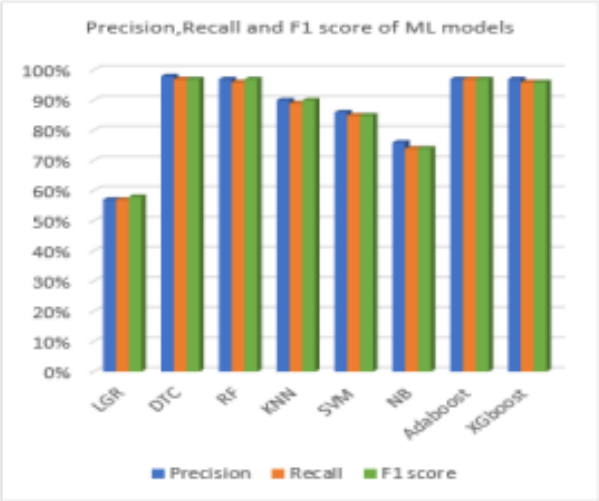

Fig. 3. Precision, Recall and F1 score comparison of ML models

Figure 2 shows the results of several ML models when testing for the presence of insider threats. The top three performers in terms of accuracy were Random Forest (95.2%), Adaboost (95.5%), and Rankine (95.7%). The 95% accuracy rate that XGBoost achieved was also remarkable. Compared to KNN's 90% and SVM's 86%, the accuracy of simpler models like Logistic Regression was much lower at 58%. The 75% performance attained by Naïve Bayes is commendable. Figure 3 displays the F1 score, recall, and accuracy of each model in identifying insider threats. All three models—Adaboost, Random Forest (RF), and Decision Tree (DTC)—perform well, with F1 scores of 97%, demonstrating parity between recall and accuracy. Equally impressive was XGBoost's 96% F1 score. The F1 score of 90% achieved by KNN is quite respectable, whereas SVM achieved just 85%.

With 74% and 58% of the F1 scores, respectively, Naïve Bayes (NB) and Logistic Regression (LGR) performed worse than the ensemble approaches. The comparison shows that ensemble methods, such as RF and Adaboost, provide better accuracy and recall. Identifying internal risks requires these abilities. Section I. Restrictions The present approach has several advantages, but it also has some serious drawbacks. In order to solve the issue of class imbalance, this solution used SMOTE. It is possible that other sampling methods might provide more precise results. The results may be much more remarkable if many machine learning models were used together.

V. CONCLUSION

Many businesses find it very difficult to identify possible threats from inside. This article shows that many ML methods work well for this task. We used the SMOTE approach to ensure that the models could understand both internal threats and non-threat behaviours, which helped with the class imbalance problem. Various approaches were used to assess the models' performance on the popular CERT dataset.

These methods included Logistic Regression, Decision Trees, Random Forest, SVM, KNN, Naïve Bayes, Adaboost, and XGBoost. With a success rate of 97.5%, ensemble methods, namely Random Forest and Adaboost, were the most effective strategies. This was a perfect example of their capacity to spot intricate patterns linked to insider threats. To further enhance the detection capabilities and interpretability of models in insider threat situations, future research

should integrate deep learning methods with expanded feature selection methodologies.

REFERENCES

1. S. G, S. Silas and E. B. Rajsingh, "A Pragmatic Enquiry to Learn Recent Trends in Insider Threat Detection Approaches," 2024 7th International Conference on Circuit Power and Computing Technologies, Kollam, India, 2024.
2. F. Femi-Oyewole, et al, "Survey on Predictive Algorithms to Detect Insider Threat on a Network Using Different Combination of Machine Learning Algorithms," International Conference SEB4SDG, OmuAran, Nigeria, 2024.
3. H. E. -E. Abdallah et al., "Performance Evaluation Framework for Insider Threat Detection Using Machine Learning," 2024 Intelligent Methods, Systems, and Applications, Giza, Egypt, 2024.
4. D. Sridevi et al, "Detecting Insider Threats in Cybersecurity Using Machine Learning and Deep Learning Techniques," International Conference on Communication, Security and Artificial Intelligence, Greater Noida, India, 2023.
5. R. Yousef et al, "A Machine Learning Framework & Development for Insider Cyber-crime Threats Detection," International Conference on Smart Applications, Communications and Networking, Istanbul, Turkiye, 2023.
6. N. Dixit et al, "Insider Threat Classification Using KNN MachineLearning Technique," International Conference on Contemporary Computing and Communications, Bangalore, India, 2023.
7. Y. Jing et al, "Analyze Organizational Internal Threats Using Machine Learning," International Conference on Applied Machine Learning , Dalian, China, 2023.
8. B. Nagabhushana Babu and M. Gunasekaran, "An Analysis of Insider Attack Detection Using Machine Learning Algorithms," 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications, Tumkur, Karnataka, India, 2022.
9. S. K. Mamidanna et al, "Detecting an Insider Threat and Analysis of XGBoost using Hyperparameter tuning," International Conference on Advances in Computing, Communication and Applied Informatics , Chennai, India, 2022.
10. X. Sun et al, "Insider Threat Detection Using An Unsupervised Learning Method: COPOD," 2021 International Conference on Communications, Information System and Computer Engineering, Beijing, China, 2021.
11. R. Nasir et al, "Behavioral Based Insider Threat Detection Using Deep Learning," in IEEE Access, vol. 9, pp. 143266-143274, 2021.
12. P. Varsha Suresh et al., "Insider Attack: Internal Cyber Attack Detection Using Machine Learning," 2021 12th International Conference on Computing Communication and Networking Technologies, Kharagpur, India, 2021.
13. D. C. Le et al., "Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 30-44, March 2020.