Journal of Global Research in Electronics and Communication
Volume 1, No. 4, April 2025
Available Online at: https://jgrec.info/index.php/jgrec/index
ISSN: 2321-3175
RESEARCH PAPER

Detecting Insider Threats Through Network Traffic Analysis Using Machine Learning for Cybersecurity

Mani Gopalsamy
Independent Researcher
manigopalsamy14@gmail.com

***Abstract*—Network traffic management alongside security becomes increasingly difficult because of expanding network quantities and advancing network complexities. Current traffic analysis practices, together with congestion management methods, require smart, automated solutions because they work in a reactive manner and consume extensive resources. This research utilizes CIC-Darknet2020 dataset during a study of machine learning technology which analyzes network traffic to detect insider threats. The study makes use of comprehensive preprocessing strategies that handle missing values and remove unnecessary features while converting IP addresses into octet-based features to support better feature processing. SMOTE functions as a technique to fix class imbalance by over-sampling minority classes, thereby maintaining balance between all traffic categories. The research adopts XGBoost as its main classification method while comparing against Random Forest and Naïve Bayes models for performance evaluation. XGBoost exhibits top classification abilities according to performance assessment, which includes a 90.12% accuracy rate and precision, recall scores and F1-score. XGBoost proves its ability to differentiate between regular and suspicious traffic patterns, which positions it as an effective tool for intrusion detection enhancement. This research brings new cybersecurity advancements by using a scalable data method for analyzing network traffic and keeping track of internal threats.**

Keywords*—Network Traffic analysis, Machine Learning, Extreme Gradient Boosting, Insider threat, cybersecurity.

I. INTRODUCTION

Modern information networks demand effective control of network traffic data analysis procedures. Network management necessities, along with automated traffic analysis, have grown because of substantial network usage increases. Potentially effective solutions emerged from machine learning-based methodologies, which have overcome these obstacles in the past few years. Current industry demands the implementation of network traffic evaluation and management systems for present-day digital environments. [1][2]. Network complexity, together with rapid network traffic growth, currently makes it difficult for network administrators to manage and analyze network traffic [3].

In the network, one can find highly confidential and valuable information regarding online banking transactions, business activities, and e-commerce data. The analysis of network traffic is crucial to ensuring appropriate information security, as shown in Figure 1. Network traffic analysis and tablebeing watched to make sure security lapses don't happen. Network traffic analysis is a crucial step in developing proactive congestion management strategies that effectively identify malicious and benign packets[4]. These strategies distribute network resources according to anticipated usage in an effort to prevent network congestion[5].

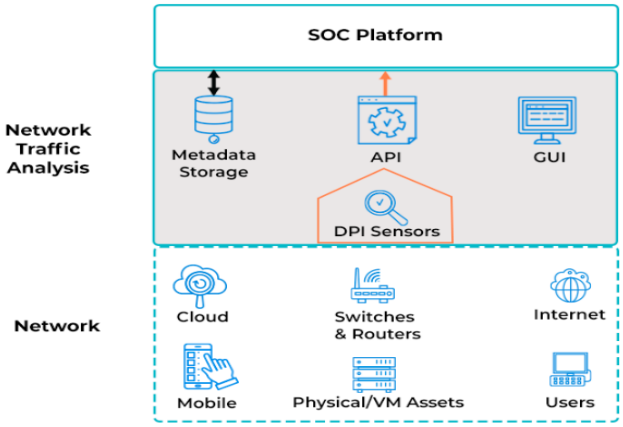

Fig. 1. Network Traffic Analysis

Insider threats are a serious issue in the world of cybersecurity [6][7]. A current or former employee, contract worker, or business associate with access to the company's network is considered an insider threat, system, or data, and purposefully take steps that would compromise the availability, confidentiality, or integrity of the data or information systems of the company[8]. It is thought to be the most dangerous danger to any company[8][9].

Increasing bandwidth, replacing servers with high-end servers, and upgrading network infrastructure are common solutions to congestion issues[10]. This method is costly, temporary, and non-scalable. Immediately following the upgrade, the congestion issue will temporarily ease before progressively worsening as users adjust their behavior. Deploying a scalable system for monitoring and analyzing network traffic is an alternate solution to this problem. This will help you comprehend the internet's fluctuations, traffic patterns, and general network stability [11].

The identification and integration of behavioral indications for One of the biggest risks is insider threat issues, aside from the technological and cyber arena. Sometimes, they overlook the human aspect of the problem while examining the data that is available to verify the behavior of insider threats [12][13]. Numerous researchers have contributed significantly throughout the years to the solution of these problems. The success of textual analysis using machine learning algorithms has been impressive.

A. Contribution of the Study

This study advances network security by offering a reliable machine learning-based method for identifying harmful traffic analysis.

- By leveraging the CIC-Darknet2020 dataset and implementing advanced preprocessing methods, including dealing with missing values, eliminating irrelevant features, and transforming IP addresses into octet-based features, the research enhances feature extraction for improved classification accuracy.
- The study utilizes SMOTE to solve class imbalance problems and establish balanced minority class distribution, which leads to improved model performance.
- The use of Extreme Gradient Boosting (XGBoost) and a comparison between Random Forest and Naïve Bayes demonstrates how well ensemble learning methods detect network breaches and insider threats.
- The research performs a systematic assessment of model performance through precision and recall tests alongside the accuracy and F1-score measures for comprehensive detection capability assessment.
- In the end, this research strengthens intrusion detection systems and proactive threat mitigation tactics by enhancing predictive analytics in network traffic categorization, which offers cybersecurity practitioners useful information.

B. Structure of paper

The remainder of the paper is structured in this manner. Give a review of the literature on insider threats using network traffic analysis in Section II. In Section III, methodologies and methodology are presented using Machine Learning models, and in Section IV, outcomes are analyzed and discussed. The study's conclusion and next steps are discussed in section V.

II. LITERATURE REVIEW

In this section, mention a few earlier research projects that used machine learning to examine network data and identify inside threats.

Pathan et al. (2023) analyze network traffic using machine learning techniques for SDN so that network flows can recognize traffic patterns. They show that intelligent pattern analysis for network traffic based on machine learning models produces better results than conventional techniques. Nowadays, a large number of traffic passes through the network. However, the network system is unreliable and has safety issues. Different attacking activities may arise on the network traffic. The SDN has the power to decouple its control plan from the data plan, which results in more effective observation in terms of traffic analysis [14].

Ganesh Parihar and Ghosh (2023) explore network traffic analysis using various technologies, focusing on dynamic analysis for malware study as an analysis environment. It investigates DNS and HTTP usage by malware, considers ethics and risk mitigation, and delves into machine learning-based IoT device identification. Addressing these topics contributes to network traffic analysis, its application, and its implications for network security and control. The study introduces a multi-stage meta-classifier methodology to enhance IoT device classification, discusses network traffic predictability, and highlights intrusion detection using packet sniffers for security [15].

Rajchel et al. (2020) explain a system that uses the categorization of temporal activity on a network to identify possible insider threats. For both government and non-governmental organizations, insider threats constitute an expensive and hazardous issue. The fundamental idea of network defense—keeping outsiders out and possible dangers in—This not apply to insider threats as insiders have a privileged degree of network access by nature. They demonstrate via several case studies that the method is successful in identifying behavioral anomalies that require further examination by a human analyst after testing the system on an operational network with more than 8,000 hosts [16].

Soykan and Boluk (2021) intended to use artificial neural networks and machine learning to identify whether network traffic is coming from the TOR network. They initially conducted data analysis on their dataset to learn more about it. In order to learn the dataset, numerical values were given categorical values. The data set is normalized after categorical data is converted to numerical data, with all attributes lying between -1 and 1. To ascertain if the upcoming traffic was TOR, previous data was analyzed using machine learning classification methods such as K Neighbor, Support Vector Machine, Random Decision Forest, Naive Bayes Classifiers, and Logistic Regression, among others [17].

Sudhakar and Kaliyamurthie (2022) discuss the use of machine learning algorithms in cybersecurity. They will be better protected against viruses, malware, and security lapses, thanks to machine learning. Cybersecurity will benefit from additional services offered by machine learning. With network traffic analytics, they may use each layer's depth to determine the assaults. Numerous machine learning techniques are employed as warning systems in cybersecurity. However, the precision provided by the machine learning system will surpass that of the human counterpart. It will show current assaults in real time, assisting cybersecurity professionals in averting dangers [18].

Dorrah et al. (2024) aim to create a highly accurate system for detecting harmful network activities by examining network traffic patterns, including the duration and frequency of connections, the amount of data transferred, and the involved endpoints. Numerous algorithms, including the research, employed Naive Bayes, Random Forest, Decision Trees, and Logistic Regression. Their goal is to have a big influence on the field of cybersecurity by presenting an effective machine learning-driven tool for accurate malware detection, thereby improving network security [19].

Table I provides the summary of related work on network traffic analysis using machine learning techniques with various key focused areas.

TABLE I. STUDIES INVOLVING INSIDER THREATS THROUGH NETWORK TRAFFIC ANALYSIS USING MACHINE LEARNING AND VARIOUS TECHNIQUES

Reference	Methodology	Dataset	Performance	Limitations & Future Work
Pathan et al. (2023)	Machine learning-based intelligent pattern analysis for SDN network traffic observation.	SDN traffic dataset	Superior outcomes compared to traditional methods	The network system remains unreliable with safety issues;

				further improvements in attack detection are needed
Ganesh, Parihar, and Ghosh (2023)	Dynamic analysis for malware detection, IoT device identification, intrusion detection using packet sniffers	DNS & HTTP traffic logs	Effective multi-stage meta-classifier for enhanced IoT classification	Ethical concerns and risk mitigation strategies need further exploration
Rajchel et al. (2020)	Characterization of temporal behavior for insider threat detection	Operational network data (8,000+ hosts)	Successfully detected behavioral anomalies for insider threat identification.	Requires human analysts for validation; potential for automation with AI integration
Soykan and Boluk (2021)	Machine learning and artificial neural networks for TOR network traffic identification	TOR network dataset	Effective classification using SVM, Random Forest, Naïve Bayes, KNN, and Logistic Regression	Dataset preprocessing is required; further improvements in model generalization.
Sudhakar and Kaliyamurthie (2022)	Machine learning approaches for cybersecurity and network traffic analytics.	Various cybersecurity datasets	High accuracy in real-time attack detection	Further enhancement of machine learning accuracy and scalability is needed
Dorrah et al. (2024)	Network traffic analysis for malware detection using ML algorithms	Network traffic logs	Significant improvement in malware detection using Logistic Regression, Random Forest, Decision Trees, and Naïve Bayes	Additional feature engineering and dataset diversity are needed for robustness

A. Research gaps

Despite significant advancements in using machine learning for identifying insider threats using network traffic analysis, several research gaps remain. Current approaches lack real-time detection capabilities, making them ineffective for proactive threat mitigation. Scalability is a challenge, as most models are tested on little datasets and might not function effectively in large, complex networks. Additionally, many studies focus on specific network architectures, limiting their generalizability. The lack of explainability in deep learning models makes it difficult for analysts to interpret results, while adversaries continue to develop evasion techniques that bypass detection. Furthermore, privacy concerns and regulatory compliance pose challenges in deploying these models in real-world environments. There is also a need for standardized benchmark datasets to ensure consistent evaluation across studies. Future research should focus on developing scalable, explainable, and adversarial robust ML models that integrate real-time analytics, privacy-preserving techniques, and hybrid threat intelligence frameworks to enhance insider threat detection.

III. METHODOLOGY

The research methodology consists of structured network traffic analysis followed by machine learning method application and performance assessment of classification outcomes. The procedure starts by obtaining network traffic data, which includes legitimate and malicious traffic examples. A preprocessing stage enables the data treatment of missing values through drop-and-remove and feature selection by discarding Flow ID and Timestamp while converting IP addresses into octet features to maintain network-related data. The dataset obtains 72 features for additional examination after data preprocessing procedures. To address the problem of class imbalance, the Synthetic Minority Over-Sampling Technique (SMOTE) makes sure that various traffic groups are distributed evenly. Splitting the dataset produced a training portion that amounts to 80% and a testing section that totals 20% to set a framework for model development and validation procedures. The Extreme Gradient Boosting (XGBoost) algorithm serves as the classification choice because it demonstrates efficient processing of structured data while handling imbalanced datasets effectively and producing high predictive accuracy results. The trained model receives its evaluation through existing key performance metrics that combine accuracy with Its categorization skills are assessed using the F1-score, recall, and accuracy. A final assessment of model outcomes will establish how well the model detects network traffic correctly. Figure 2 subsequent phases provide a thorough explanation based on the details given below:

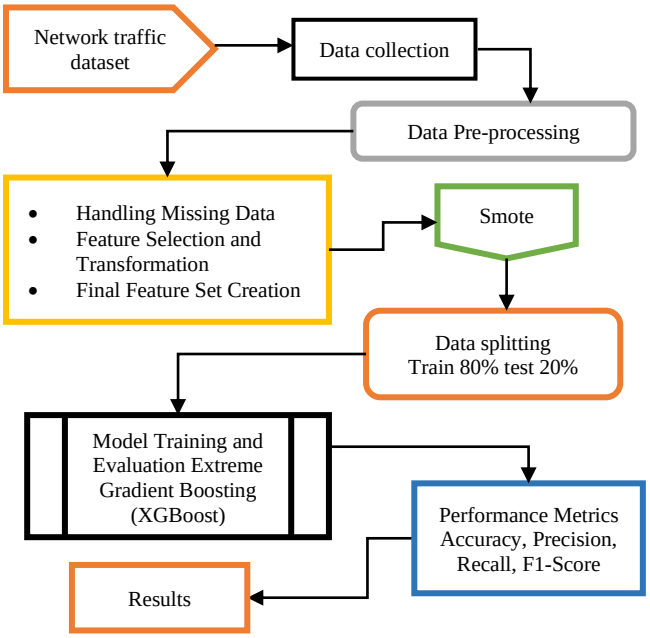

Fig. 2. Methodology flow Diagram for network traffic analysis

A. Data Collection

The dataset utilized in this study, CICDarknet2020, is a comprehensive collection of network traffic data specifically designed for analyzing Darknet communications and distinguishing them from normal internet activities. The dataset was generated to facilitate research in malware detection, intrusion detection, and network traffic classification, enabling the development of machine learning models that can detect possible dangers both before and after an assault. The dataset contains over 80 network traffic features, covering aspects such as flow characteristics, packet behavior, time-based attributes, and protocol information.

B. Data Pre-processing

There are missing data samples in the CIC-Darknet2020 dataset, notably feature values of "NaN." When they clean up their data, they eliminate samples that have these values. Comparatively speaking to the other traffic groups, there are far less Tor samples. Timestamp, Flow ID, Source IP, and Destination IP are the CIC Flow Meter flow labels—were

removed in earlier work utilizing this information. Additionally removed in their investigation are the Timestamp and FlowID. To maximize the amount of information that can be extracted the source and destination IP address octets are extracted from the CIC-Darknet2020 dataset and placed in distinct feature columns. When this IP information is kept, the classifiers perform better, based on initial experiments performed on the dataset both with and without certain IP octet characteristics. This preprocessing phase results in a total of 72 features in their dataset.

C. SMOTE

In order to overcome this imbalance in the classification job, they use SMOTE to oversample each minority class and examine the effects of mitigation. To create fresh samples, SMOTE interpolates feature values linearly. The oversampling values that they look at are 20%, 40%, 60%, 80% (partial SMOTE), 100% (full SMOTE), and 0% (no SMOTE). SMOTE is used to compare the class with the most samples to all classes with less samples than the oversampling threshold. Notably, lower SMOTE thresholds only provide an equal number of samples across oversampled classes, but 100% SMOTE produces an equal number of samples for each class.

D. Data Splitting

It is frequently required to separate the information in order to train and test the model. Two portions make up the dataset used in this study. Testing and training are two distinct stages. The dataset's division into 80% training and 20% test data enables the stratified option.

E. Classification Models

Some categorization models for Detecting Insider Threats Through Network Traffic Analysis are explained in this section. These models are used in investigations that compare things.

1) Extreme Gradient Boosting (XG-Boost)

The Gradient Boosting Machine (GBM), a hybrid of gradient descent and boosting, is the foundation of Extreme Gradient Boosting[20]. An ensemble-learning approach called "boosting" assigns a different weight to the distribution of training data for every iteration[21]. Weight is added for the incorrectly categorized sample in each boosting iteration, while weight is removed for the correctly classified sample. Consequently, it successfully modifies the training data distribution, as seen in Figure 3. GBM minimizes the following regularized objectives by using second order gradient statistics that shown in Equation (1).

$$\mathcal{L}\,\phi = \sum_i l(\hat{y}\,\mathrm{i}\,, \mathrm{yi}\,) + \sum_k \Omega(f_k) \qquad (1)$$

Where: - $\Omega(f) = \gamma^T + \frac{1}{2}\lambda\|w\|^2$

where Ω penalizes the model's complexity and l is a differentiable convex loss function that quantifies the difference between the target yi and the forecast $\hat{y}\,i$.

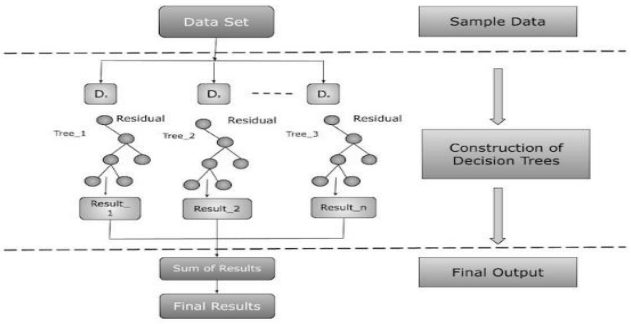

Fig. 3. The structure of XGBoost

2) Random forest

The Random Forest algorithm is a potent supervised learning technique for applications involving regression and classification. In order to improve accuracy and lessen overfitting, an ensemble approach builds several decision trees during training and aggregates their outputs. The algorithm follows a process called bootstrapping and bagging, in Figure 4, it trains individual decision trees on subsets of the training data that are randomly selected with replacement. Additionally, to introduce further randomness and improve generalization, Instead of using every feature that is available, each tree is trained on a randomly selected subset of characteristics. The prediction calculation for finished trees depends on majority voting in classification problems, while it uses prediction average for regression problems. The main strength of Random Forest is its capability to handle data missing issues while minimizing overfitting, which provides stronger performance than a single decision tree.

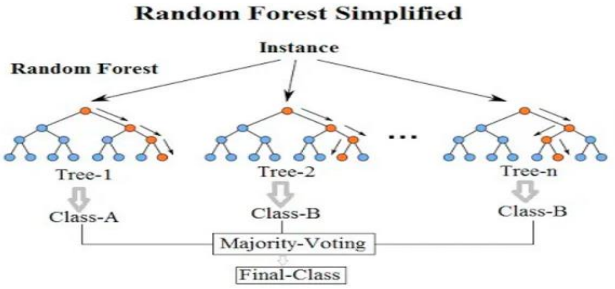

Fig. 4. The structure of random forest

3) Naive_Bayes

The Naïve Bayes (NB) is a probabilistic ML classifier that applies the ideas of Bayes Theorem, shown in Figure 5. It is simple but effective. Text categorization and spam filtering are two common uses for it, and network traffic analysis due to its efficiency and low computational cost. The initial naive Bayes classification's accuracy method may be significantly increased by using the training set's information to its fullest potential and overcoming the aforementioned drawbacks. Simultaneously, the enhanced traffic risk management, the naive Bayes classification method is employed to accurately forecast and categorize the driving risk of the driver and ultimately execute efficient risk management [22].

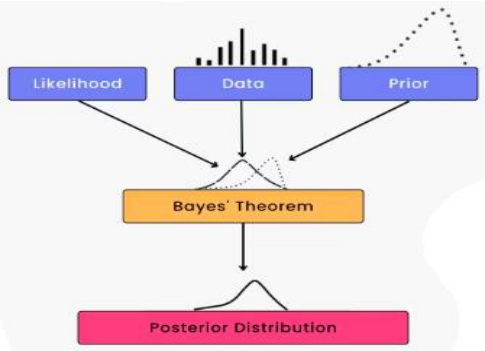

Fig. 5. The structure of Naïve Bayes

F. Performance Measures

To evaluate each model's effectiveness, four different performance criteria have been employed: recall, accuracy, precision, and F1-score. These parameters are given:

1) Confusion Matrix

One popular tool for analyzing the results is the confusion matrix, which is used to analyze academic achievement. Figure 6 displays the matrix's visual representation. The matrix, which is the combination of results from classifications, displays the result data in four primary ways. A result is considered true positive (TP) if the actual value of the classification equals the expected value. Similar in nature, true negative (TN) concepts are centered around zero. In contrast to a false negative (FN), which happens when the reverse is true, a false positive (FP) is when the expected result is 1, but the actual value is 0.

		Predicted values		
		Positive	Negative	Totals
Actual Values	Positive	TP	FN	P = (TP + FN) = Actual Total Positives
	Negative	FP	TN	N = (FP + TN) = Actual Total Negatives
	Totals	Predicted Total Positives	Predicted Total Negatives	

Fig. 6. Representation class of confusion matrix

2) Accuracy

The accuracy of the model is defined as the proportion of examples it correctly classifies and the overall class prediction error. This measure summarizes how well the model performs across classes. However, performance may be misrepresented by biased data. It is possible for a classifier that primarily predicts the majority class to incorrectly categorize occurrences of the minority class. It calculates as Equation 2.

$$Acuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (2)$$

3) Precision

The percentage of instances that are correctly assigned is known as precision to a class once all the data has been categorized into that class. In this instance, it shows the proportion of corona cases that actually are corona cases. It is determined for every class using the one-versus-all approach: It calculates as Equation 3.

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

4) Recall

The number of cases correctly categorized into a class is determined by sensitivity or recall. This context measures the proportion of properly represented instances by the classifier among all carriers of the illness. The one-vs-all method, similar to precision, is used to calculate recall. It calculates as Equation 4.

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

5) F1-score

The F1-score, sometimes referred to as the F-measure, is the weighted harmonic mean of recall and accuracy. This metric is best suited for usage when the dataset is very unbalanced. A more thorough evaluation is possible when a wider viewpoint is used. It calculates as Equation 5.

$$F1-score = 2 * \frac{precision*recall}{precision+recall} \quad (5)$$

IV. Results And Discussion

The outcomes of insider threat simulations using detection utilizing machine learning techniques in network traffic analysis are covered in this part. This section includes the classifier statistics, performance metrics, results, and dataset description. This shows the results of the dataset evaluation carried out for this study.

A. Experiment results

This section poses the findings of the Extreme Gradient Boosting (XGBoost) model applied to a large dataset for network traffic analysis using machine learning.

Fig. 7. Class prediction error of the XGBoost classifier

The XGBoost Classifier class prediction error curve is displayed in Figure 7. The actual class labels are shown on the x-axis, while the number of expected occurrences for each class is shown on the y-axis. Each bar corresponds to an actual class and is segmented into different colors, representing misclassified predictions across various classes. The legend on the right associate's colors with class labels (0-10). Some classes, such as class 2 and class 7, exhibit a high number of correct predictions (dominant bar segments), whereas other classes display more variation in misclassification. The chart visually highlights the extent of classification errors for each class, helping to assess the model's weaknesses in distinguishing certain categories.

TABLE II. Extreme Gradient Boosting Model Performance Matrices For Network Traffic Analysis

Model	Extreme Gradient Boosting
Accuracy (%)	90.12
Precision (%)	90.14
Recall (%)	75.00
F1-score (%)	89.90

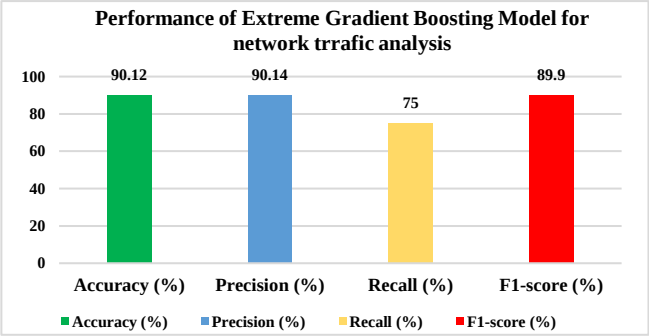

Fig. 8. Bar graph of parameters Performance of Extreme Gradient Boosting model for network traffic analysis

Table II and Figure 8 presents the performance metrics of the Extreme Gradient Boosting (XGBoost) model for network traffic analysis. The model's overall correctness in categorization was demonstrated by its 90.12% accuracy rate. The model is generally accurate when it predicts a positive class, as seen by its precision of 90.14%. But with a 75.00% recall, the model accurately detects 75% of real positive instances, suggesting some missed detections. Precision and recall are balanced According to the model's F1-score of 89.90%, recall may need to be enhanced for increased sensitivity even if it functions well overall.

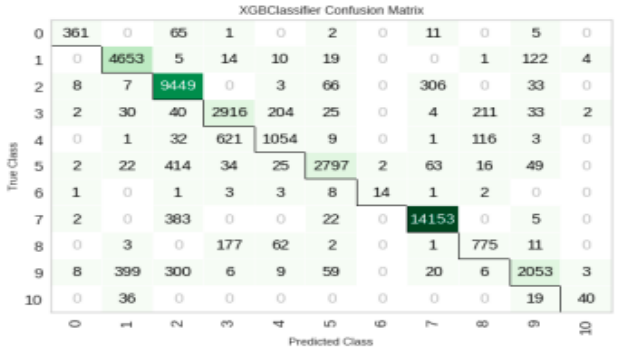

Fig. 9. Confusing Matrix of the XGBoost classifier

The classification performance of an XGBoost Classifier (XGBClassifier) is depicted in Figure 9 through a confusion matrix. With numerical values representing classification counts, the x-axis of the matrix displays anticipated classes, whereas the y-axis displays genuine classes. Instances that are correctly categorized, such 4653, 9449, and 14153, are shown by green-highlighted diagonal elements; misclassifications are indicated by off-diagonal values. The model performs well for dominant classes but shows some misclassification in lower-frequency categories. This visualization helps assess model accuracy and identify areas for improvement, such as class balancing or hyperparameter tuning.

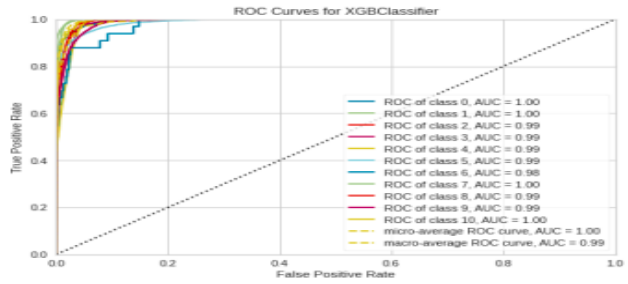

Fig. 10. ROC of the XGBoost classifier

Figure 10, displays ROC Curves for an XGBoost Classifier, showing model performance across multiple classes. The True Positive Rate (TPR) is shown on the y-axis, while the False Positive Rate (FPR) is shown on the x-axis. AUC values near 1.00 indicate great classification accuracy, and each colored line represents a distinct class. Strong overall performance is demonstrated by the macro-average AUC of 0.99 and the micro-average AUC of 1.00. The curves are concentrated in the upper-left corner, suggesting minimal false positives and excellent model efficiency.

B. Comparative analysis and Discussion

An examination of network traffic using several models in comparison. In Table III, many machine learning and deep learning models for detecting insider threats are compared and contrasted in terms of performance metrics.

TABLE III. COMPARISON BETWEEN VARIOUS MODEL FOR NETWORK TRAFFIC ANALYSIS

Models	XGBoost	Random forest [23]	Naïve Bayes [24]
Accuracy	90.12	87.2	85.97
precision	90.14	87.4	71.01
Recall	75	87.02	80.36
F1-Score	89.9	87.3	68.42

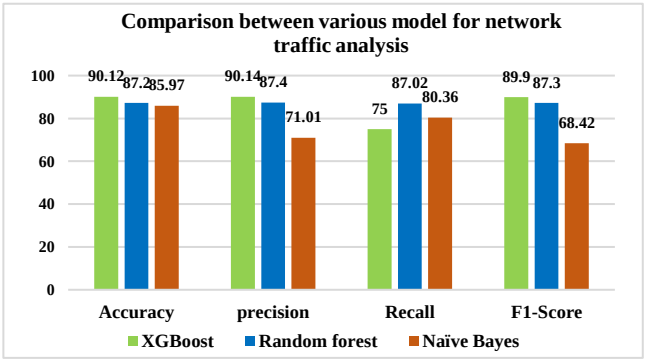

Fig. 11. Bar graph of the comparison various model for network traffic analysis

Table III and Figure 11 compare the XGBoost, Random Forest, and Naïve Bayes models' F1-score, recall, accuracy, and precision. With the best Accuracy (90.12%), Precision (90.14%), and F1-Score (89.9%), XGBoost performs the best, but its lower Recall (75%), suggests that it could overlook some favorable situations. Random Forest provides a balanced performance with Accuracy (87.2%), Precision (87.4%), Recall (87.02%), and F1-Score (87.3%), making it a well-rounded choice. In contrast, Naïve Bayes underperforms with lower Precision (71.01%) and F1-Score (68.42%), though its Recall (80.36%) is relatively better. Overall, XGBoost excels in classification accuracy, Random Forest maintains consistency across metrics, and Naïve Bayes struggles with precision and overall classification balance.

V. CONCLUSION AND FUTURE SCOPE

The assessment of several insider threat detection methods showed that insider threat detection is not a single issue but rather a collection of distinct computer science, psychology, and sociology study fields. After classifying the related fields based on the behavior type under analysis, they verified that their classification could identify all of the main insider threat classifications. Network traffic encryption is essential for data security and privacy, but it also makes it difficult to analyze and categorize the traffic for a variety of uses, including traffic

optimization, network management, and security monitoring. A robust machine learning method for network traffic analysis develops all the necessary components to identify both insider threats and malicious activities. The study reaches high network traffic pattern classification accuracy by applying CIC-Darknet2020 dataset alongside sophisticated data processing approaches and attribute manipulation methods. The SMOTE implementation effectively balances classes to improve the identification of minority attacks. Research compares ensemble learning approaches and demonstrates their success in cybersecurity through an analysis between XGBoost, Random Forest and Naïve Bayes. XGBoost delivers superior performance to its counterparts according to the experimental findings through strong achievement in measures like accuracy and precision while retaining high recall and F1-score which proves its competence for network intrusion detection tasks. These research findings allow the development of improved security measures for proactive threats alongside stronger network protection systems within business networks.

The general performance of the model can be enhanced by applying multiple methods that include parameter optimization and feature optimization with sequential traffic analysis using deep learning techniques such as XGBoost. Investigations of real-time deployment of this proposed model should happen in dynamic network environments to test its operational effectiveness. To boost insider threat detection capabilities the dataset should be enlarged by including recent threat patterns and implementing new behavioral indicators.

REFERENCES

[1] S. S. S. Neeli, "Critical Cybersecurity Strategies for Database Protection against Cyber Attacks," *J. Artif. Intell. Mach. Learn. Data Sci.*, vol. 1, no. 1, p. 5, 2023, [Online]. Available: https://urfjournals.org/open-access/critical-cybersecurity-strategies-for-database-protection-against-cyber-attacks.pdf%0A

[2] P. M. Rajendra Prasad Sola, Nihar Malali, *Cloud Database Security: Integrating Deep Learning and Machine Learning for Threat Detection and Prevention: 0*. 2025.

[3] R. Chauhan, "A Machine Learning-based Approach for Network Traffic Analysis and Management," *Turkish J. Comput. Math. Educ.*, 2020, doi: 10.17762/turcomat.v11i3.13603.

[4] A. Immadisetty, "Machine Learning for Real-Time Anomaly Detection," *Int. J. Multidiscip. Res.*, vol. 6, no. 6, 2022.

[5] M. Joshi and T. H. Hadi, "A Review of Network Traffic Analysis and Prediction Techniques," no. March 2020, 2015.

[6] B. Bin Sarhan and N. Altwaijry, "Insider Threat Detection Using Machine Learning Approach," *Appl. Sci.*, 2023, doi: 10.3390/app13010259.

[7] S. Duary, P. Choudhury, S. Mishra, V. Sharma, D. D. Rao, and A. Paul Aderemi, "Cybersecurity Threats Detection in Intelligent Networks using Predictive Analytics Approaches," in *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, IEEE, Feb. 2024, pp. 1–5. doi: 10.1109/ICIPTM59628.2024.10563348.

[8] M. I. Khan, A. Arif, and A. R. A. Khan, "AI-Driven Threat Detection: A Brief Overview of AI Techniques in Cybersecurity," *BIN Bull. Informatics*, vol. 2, no. 2, pp. 248–261, 2024.

[9] P. Piyush, N. S. Gill, P. Gulia, D. D. Rao, Y. Mandiga, and P. K. Pareek, "Systematic Analysis of threats, Machine Learning solutions and Challenges for Securing IoT environment," *J. Cybersecurity Inf. Manag.*, vol. 14, no. 2, pp. 367–382, 2024, doi: 10.54216/JCIM.140227.

[10] R. Tarafdar, "AI-POWERED CYBERSECURITY THREAT DETECTION IN CLOUD ENVIRONMENTS," *Int. J. Comput. Eng. Technol.*, vol. 16, no. 1, pp. 3858–3869, Feb. 2025, doi: 10.34218/IJCET_16_01_266.

[11] S. B. a Mohammed, "Network Traffic Analysis : A Case Study of ABU Network," vol. 4, no. 4, pp. 33–40, 2013, doi: 10.5120/2222-2863.

[12] K. Rajchandar, M. Ramesh, A. Tyagi, S. Prabhu, D. S. Babu, and A. Roniboss, "Edge Computing in Network-based Systems: Enhancing Latency-Sensitive Applications," in *2024 7th International Conference on Contemporary Computing and Informatics (IC3I)*, 2024, pp. 462–467. doi: 10.1109/IC3I61595.2024.10828607.

[13] A. K. MI Khan, A Arif, "The Most Recent Advances and Uses of AI in Cybersecurity," *BULLET J. Multidisiplin Ilmu*, vol. 3, no. 4, pp. 566–578, 2024.

[14] M. N. Pathan, F. I. Fahim, S. Akter, S. Ahamed, and A. Sultana, "An Intelligent Traffic Pattern Analysis and Optimization Using Machine Learning in SDN Enable Network Infrastructure," *2023 5th Int. Conf. Sustain. Technol. Ind. 5.0, STI 2023*, pp. 1–2, 2023, doi: 10.1109/STI59863.2023.10464623.

[15] N. Ganesh, A. S. Parihar, and G. Ghosh, "Analysing Network Traffic and Implementing Diverse Technologies to Examine Different Components of the Network," *3rd IEEE Int. Conf. ICT Bus. Ind. Gov. ICTBIG 2023*, pp. 1–2, 2023, doi: 10.1109/ICTBIG59752.2023.10456258.

[16] B. Rajchel, J. V. Monaco, G. Singh, A. Hu, J. Shingleton, and T. Anderson, "Temporal Behavior in Network Traffic as a Basis for Insider Threat Detection," in *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, 2020. doi: 10.1109/SSCI47803.2020.9308236.

[17] M. Soykan and P. S. Boluk, "Tor Network Detection by Using Machine Learning and Artificial Neural Network," in *2021 International Symposium on Networks, Computers and Communications, ISNCC 2021*, 2021. doi: 10.1109/ISNCC52172.2021.9615730.

[18] M. Sudhakar and K. P. Kaliyamurthie, "Machine Learning Algorithms and Approaches used in Cybersecurity," in *2022 IEEE 3rd Global Conference for Advancement in Technology, GCAT 2022*, 2022. doi: 10.1109/GCAT55367.2022.9971847.

[19] M. Dorrah *et al.*, "Advanced Network Malware Detection: Integrating PySpark and Machine Learning Techniques," *6th Int. Conf. Comput. Informatics, ICCI 2024*, pp. 131–136, 2024, doi: 10.1109/ICCI61671.2024.10485051.

[20] A. H. Anju, "Extreme Gradient Boosting using Squared Logistics Loss function," *Int. J. Sci. Dev. Res.*, vol. 2, no. 8, pp. 54–61, 2017.

[21] Y. H. Rajarshi Tarafdar, "Finding majority for integer elements," *J. Comput. Sci. Coll.*, vol. 33, no. 5, pp. 187–191, 2018.

[22] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *EURASIP J. Adv. Signal Process.*, 2021, doi: 10.1186/s13634-021-00742-6.

[23] Xiaoqi Jia, "A Comparative Study of Machine Learning-based Approach for Network Traffic Classification," *IEEE Xplore*, vol. 4, p. 128, 2024, doi: 10.17977/um018v4i22021p128-137.

[24] Xiaoqi Jia, "Development of Multistage Machine Learning Classifier using Decision Trees and Boosting Algorithms over Darknet Network Traffic," *IEEE Xplore*, 2024.

