iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67586

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Enhancing Network Security with Tree-Based Machine Learning: A Study on Ensemble Intrusion Detection System Models

Abhijeet Kumar[1], Pradeep Kandula[2]
School of computer Engineering, Kalinga Institute of Industrial, Technology, Bhubaneswar , India

Abstract: ***In the era of rapid technological advancement, cybersecurity has become a paramount concern, with the ever-growing threat of unauthorized access, data breaches, and malicious attacks on networks. In this paper, we proposed an advanced intrusion detection system (IDS) leveraging various machine learning ensemble learning techniques. Intrusion detection systems are critical for cybersecurity, identifying and mitigating unauthorized access and attacks on networks. By utilizing models such as Decision Tree, Random Forest, Gradient Boosting, and XGBoost, we aimed to enhance the accuracy and efficiency of detecting network intrusions.***

Index Terms: Intrusion detection system, Machine Learning, Ensemble Learning Decision Tree, random forest, Gradient Boosting, XGBoost, stacking, cyber security

I. INTRODUCTION

In today's digital era, the internet has become a fundamental tool for conducting a wide range of essential activities. Individuals and businesses alike rely on online platforms for crucial tasks such as bill payment, fund transfers, shopping, communication, and managing personal data [1]. This reliance on the internet has resulted in an unprecedented level of connectivity, where countless devices and users are interlinked across networks worldwide. However, with this connectivity comes a heightened vulnerability to cyber threats. Attacks on home and business networks have become increasingly common and are growing in both frequency and severity [2]. For many, the idea of being targeted by cybercriminals was once considered unlikely, but as networks become more interconnected, the risk of attack has become a reality that everyone must face.

When a cyberattack occurs, it is crucial to perform a thorough and organized analysis to understand its causes, identify potential vulnerabilities, and evaluate the extent of the damage. Cyberattacks can severely disrupt operations, compromise sensitive data, and result in substantial financial and reputational harm [3]. A swift and comprehensive investigation into an attack can help to minimize network downtime and prevent long-term damage, allowing businesses to restore critical systems and ensure they remain fully operational. With appropriate analysis and response measures in place, organizations can not only mitigate immediate risks but also reinforce their defenses against future threats.

Given the growing complexity of cyber threats, modern IDS approaches increasingly employ machine learning and artificial intelligence to identify unusual patterns in network activity. By training these systems on large datasets, machine learning algorithms can distinguish between normal and abnormal behavior, effectively identifying new and emerging threats.IDS technology is designed to monitor network traffic, identify suspicious activity, and alert administrators to potential threats4]. As a result, IDS systems are evolving to become more adaptive and capable of addressing diverse attack vectors.

By employing advanced machine learning techniques, researchers can efficiently identify and classify network intrusions before they escalate, reducing the risk of widespread security breaches [5]. These techniques find applications across diverse sectors, including cybersecurity, finance, healthcare, and critical infrastructure, underscoring their versatility and importance. By enhancing intrusion detection capabilities, machine learning-based methods contribute significantly to the proactive defense of sensitive information and essential digital systems [6].

With the rise in sophisticated threats, this report investigates the effectiveness of various tree-based machine learning models—specifically Decision Tree, Random Forest, Gradient Boosting, and XGBoost—in enhancing intrusion detection capabilities. These models are trained to recognize patterns of malicious network behavior, making them valuable tools for identifying potential threats accurately and swiftly. The study explores the potential of these ensemble learning models for automatic intrusion detection, aiming to develop a more resilient and efficient approach to protecting networks against a wide range of cyberattacks

The ultimate goal of this research is to identify and develop reliable machine learning techniques that can be effectively applied to enhance network security and improve the effectiveness of intrusion detection systems. In a world where cyber threats continue to grow in sophistication and frequency, traditional methods of network protection are often insufficient to address the wide variety of attack types that modern organizations face [7]. Intrusion detection systems play a critical role in network security by monitoring and analyzing network traffic for signs of malicious activity [8]. However, achieving timely, accurate detection remains a challenge due to the vast volume of network data and the complexity of detecting new or evolving threats.

We propose the practical implementation of Decision Tree, Random Forest, Gradient Boosting, and XGBoost—four widely recognized tree-based machine learning models—for intrusion detection. These models are well-suited for classification tasks, have been effectively applied to structured data, and demonstrate strong performance in identifying and differentiating patterns within network traffic. By leveraging their capabilities in feature extraction and classification, this research aims to enhance the accuracy and reliability of intrusion detection systems.

This paper makes the following contributions:

- Surveys the vulnerabilities and potential attacks in the network traffic patterns.
- Proposes an intelligent IDS for general networks by using the tree structure ML and ensemble learning methods.
- Presents a comprehensive framework to prepare network traffic data for the purpose of IDS development.
- Proposes an averaging feature selection method using tree structure ML models to improve the efficiency of the proposed IDS and to perform an analysis of network attributes and attacks for network monitoring uses.

This paper is organized as follows: Section II presents the system overview of the proposed IDS and its architecture. Section III discusses the proposed IDS framework in detail. Section IV provides the results and performance comparison. Finally, Section V concludes the paper

II. SYSTEM DESIGN

A. Problem statement

With the rapid increase in network traffic and cyber threats, organizations often struggle to identify malicious activities within their systems [9]. Traditional intrusion detection methods can be slow, require constant monitoring, and are often limited in identifying new or sophisticated attacks. This can lead to undetected intrusions, data breaches, and significant financial losses.

The goal of this research is to develop a machine learning-based Intrusion Detection System (IDS) that can autonomously analyze network traffic and accurately classify activities as normal or malicious. Using a labeled dataset, the model will be trained to identify various types of intrusions. This IDS will provide cybersecurity professionals with a user-friendly, reliable tool for early threat detection, enabling swift responses to secure networks and protect sensitive information.

B. IDS system overview and architecture

To provide protection for both the external communications, the proposed IDS is implemented in multiple locations within the AV system. To detect threats on the CAN bus and secure it, the IDS can be placed on the top of the CAN bus to process every transmitted message and ensure the nodes are not compromised [10]. Alternatively, the proposed IDS can be placed inside the gateway to secure the external communication networks [11]. The topology of IDS implementation is shown in Fig. 1

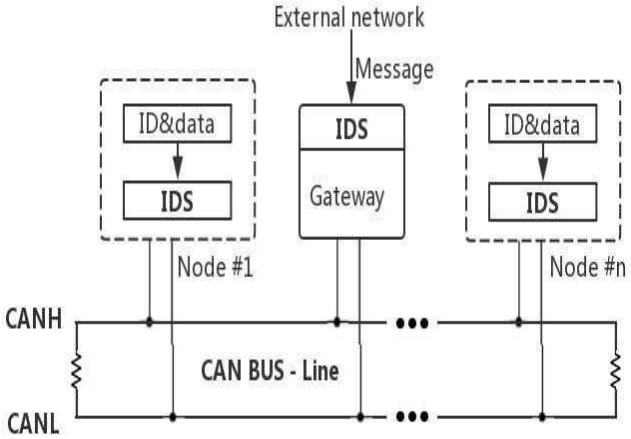


Fig. 1. The proposed IDS-protected AV architecture

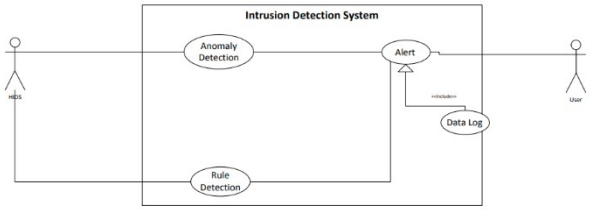


Fig. 2. The proposed IDS framework

This system will be installed in each of the host computers in a network system. It will detect any anomaly that is going into the host computers through the given rules.

1) Anomaly Detection: If Intrusion Detection System detects any abnormality in the network traffic, then it triggers the alert system.

2) Signature recognition Intrusion Detection System examines the traffic looking for well-known patterns of attack, which are saved in pattern database and triggers the alert system, if a match is found.

3) Alert System Whenever triggered by anomaly detection or signature recognition; it alerts the system Administrator.

To design the IDS, Fig. 2 Figure shows an example of deployment of host-based intrusion detection system in a network layout.

The process of the proposed model is as follows. Firstly, suffi- cient network traffic data is collected. Secondly, if the classes of the data set are imbalanced, oversampling is implemented to reduce its impact. At the next stage, feature selection based on averaging feature importance is done to reduce computational cost. After that, four base-models are built to be the input of the stacking ensemble model. At the end, the final model is built to classify the data.

III. PROPOSED IDS FRAMEWORK

A. Data pre-processing

We used Network Intrusion Detection, a publicly available dataset from Kaggle for intrusion detection, for our project. The data is supervised data in tabular form compromising more than 148517 rows and 43 columns created for an environment to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. For each TCP/Ip connection, 43 quantitive and qualitive features are obtained from normal and attack data (4 qualitative and 39 quantitative features).

We developed a model using different ensemble learning classification techniques to analyze network traffic and detect potential intrusions. Trained on a labeled dataset of normal and malicious activities, the model can accurately identify suspicious patterns, enabling effective real-time monitoring and enhancing network security The first step to develop an IDS is to collect sufficient amount of network traffic data under both the normal state and the abnormal state caused by different types of attacks. The data can be collected by the packet sniffers, but they should have suitable network attributes, or named network features, for the purpose of IDS development.

Regarding the external networks, since they belong to general networks and are prone to various regular network threats, the data with more network attributes should be collected to develop an effective IDS that can detect various types of cyber-attacks. Most of the regular network attributes such as packet length, data transfer rate, throughput, inter-arrival time, flags of TCP and their counts, segment size, and active/idle time should be considered [12]. However, the computational complexity of the proposed IDS may increase dramatically due to the high data dimensionality. Thus, further feature analysis should be done for the external network data.

The collected network data would be pre-processed after a few steps to be better suited for IDS development purpose. Firstly, the data can be encoded with label encoder because it has a certain threshold to help separate normal data and anomalies [13]. On the other hand, ML training is often more efficient with normalized data [14].

B. The proposed ML approaches

In the proposed system, to detect various cyber-attacks, developing the IDS can be considered as a Binary-classification problem, and machine learning algorithms are widely used to solve such classification problems [15] [16]. The selected ML algorithms are based on tree structure, including decision tree, random forest, Gradient Boosting, and XGBoost

Decision tree (DT) [17] is a common classification method based on divide and conquer strategy. A DT comprises decision nodes and leaf nodes, and they represent a decision test over one of the features, and the result class, respectively. Random forest (RF) [18] is an ensemble learning classifier based on the majority voting rule that the class with the highest votes by decision trees is selected to be the classification result.

Similarly, Gradient Boosting (ET) [19] is another ensemble model based on a collection of randomized decision trees generated by processing different subsets of data set. In contrast, XGBoost [20] is an ensemble learning algorithm designed for speed and performance improvement by using the gradient descent method to combine many decision trees.

For the purpose of model selection, the computational complexity of common supervised ML algorithms is calculated. Assuming the number of training instances is N, the number of features is P, and the number of trees is T, we have the following approximations. The complexity of DT is O(N2P) while the complexity of RF is O (N2 √ P T). In addition, GB and XGBoost have a similar complexity of O (NP T).

These algorithms were chosen for the additional reasons listed below: 1) Since the majority of tree structure machine learning models use ensemble learning, they frequently perform better than single models like KNN. 2) The suggested network data is a type of high-dimensional, non-linear data that they can handle. 3) The feature importance estimates are carried out as those models are being built, which is advantageous for feature selection.

To optimize the performance of various machine learning algorithms, hyperparameter tuning was performed using RandomizedSearchCV. This approach efficiently explores the hyperparameter space by randomly sampling a predefined number of configurations, allowing for faster optimization compared to exhaustive grid search [21]. For the Decision Tree Classifier, hyperparameters such as maximum tree depth (max_depthmax_depth), minimum samples required to split a node (min_samples_splitmin_samples_split), minimum leaf size (min_samples_leafmin_samples_leaf), and the number of features considered per split (max_featuresmax_features) were tuned.

Similarly, the Random Forest Classifier was optimized by adjusting the number of estimators (n_estimatorsn_estimators), tree depth, minimum samples per split, leaf size, and feature subset strategies (PP or log2Plog2 P). For both algorithms, 5-fold cross-validation was employed to ensure robust evaluation, and a computationally efficient subset of 10 hyperparameter configurations (n_iter=10n_iter=10) was sampled. These methods improved both predictive performance and generalization while preventing overfitting.

For ensemble methods like Gradient Boosting (GB) and XGBoost, additional hyperparameters were optimized to balance model complexity and training time. The learning rate (ηη), which controls the contribution of each tree, was tuned for both methods alongside subsampling rates (subsamplesubsample) to introduce randomness and prevent overfitting. GB utilized parameters like the maximum tree depth, minimum leaf size, and the number of boosting stages (n_estimatorsn_estimators) to improve its performance, with a subsample range between 0.5 and 0.8 to balance bias and variance. XGBoost introduced optimizations for sparse data and regularization, with hyperparameters such as the learning rate and the subsample fraction fine-tuned. For all models, the objective was to minimize classification error, evaluated using a binary classification loss function (log loss). The log loss function is given by:

Log Loss=−N1 i=1∑N (yi log (y^ i)+(1−yi) log (1−y^ i)) (1)

where yiyi is the true label, y^iy^i is the predicted probability, and NN is the total number of samples. These optimizations resulted in improved accuracy, as demonstrated by the evaluation of test set predictions using the best models selected from the hyperparameter search.

C. Validation Metrics

In this study, we evaluated the performance of the best-trained model using multiple validation metrics, including the confusion matrix and the Receiver Operating Characteristic (ROC) curve. The confusion matrix provides a detailed breakdown of the model's classification performance by showing the number of true positives, true negatives, false positives, and false negatives. A heatmap of the confusion matrix was plotted to visually represent these values, helping to understand the model's accuracy in predicting both the positive and negative classes. A qualified IDS should have a high DR to ensure most of the attacks can be detected and a low FAR to confirm the system does not misreport data for higher Recall [18]. Additionally, the ROC curve was plotted to assess the model's ability to distinguish between the two classes (attack vs. normal). The false positive rate (FPR) and true positive rate (TPR) were computed, and the area under the curve (AUC) was used to quantify the model's discriminatory power. A higher AUC indicates better performance in distinguishing between the classes. These evaluation techniques provide critical insights into the model's effectiveness in handling imbalanced datasets and its robustness in correctly identifying network intrusions.

IV. PERFORMANCE EVALUATION

A. Datasets Description

To evaluate the proposed IDS, and to build a comprehensive IDS that can also be effective in external communication networks, a standard IDS data set containing the most updated attack scenarios, named "NSL-KDD", is considered in this work.

To prepare better datasets for the IDS development, minor tasks including data combination, missing value removal and new label assignments were done for both datasets based on the methods proposed in [22]. The specifics of the improved datasets are shown in Table I.

TABLE I. DATA TYPE AND SIZE OF THE NSL-KDD DATASET

Class Label	Number of Instances
Normal	71463
Anamoly	77054

B. IDS performance analysis

The proposed system was implemented using Python 3.5 and the experiments were carried out on a machine with AMD Ryzen 7 5700U with Radeon Graphics with 8 cores and a CPU base speed of 1.80 GHz and a maximum boost speed of up to 4.3 GHz. Maximum available RAM is 16 GB and the processing time may vary device to device.

The results of testing different algorithms on NSL_KDD are shown in Table II. According to Table II, XGBoost outperforms the others with the highest accuracy (96.72%) and F1-Score (96.67%), showcasing its ability to achieve a balance between precision (94.83%) and recall (98.58%). Gradient Boosting also performs well, with slightly lower accuracy (96.34%) and F1-Score (96.29%) compared to XGBoost, but it has the highest recall of 98.28%. The Decision Tree and Random Forest models, while still effective with accuracies of 95.72% and 95.66% respectively, lag slightly behind in their precision and recall metrics. Overall, the results suggest that ensemble methods, particularly XGBoost, provide superior performance in detecting anomalies and reducing false positives, making them highly suitable for IDS applications.

TABLE II. PERFORMANCE EVALUATION OF IDS ON NSL-KDD DATASET

Model	Precision	Recall	F1-Score	Accuracy
Decision Tree	94.90%	96.39%	95.64%	95.72%
Random Forest	94.76%	96.33%	95.54%	95.66%
Gradient Boosting	94.36%	98.28%	96.29%	96.34%
XGBoost	94.83%	98.58%	96.67%	96.72%

V. CONCLUSION

In this work, we presented a robust ensemble learning approach for network intrusion detection, leveraging a variety of machine learning algorithms to enhance detection accuracy and reliability. Given the complexity of identifying various intrusion types in network traffic, our proposed method demonstrates substantial effectiveness in accurately classifying attacks across multiple classes.

We utilized a dataset comprising a comprehensive range of network traffic features to train our models, enabling the refinement of parameters for each algorithm and the creation of classifiers capable of accurately identifying different attack types.

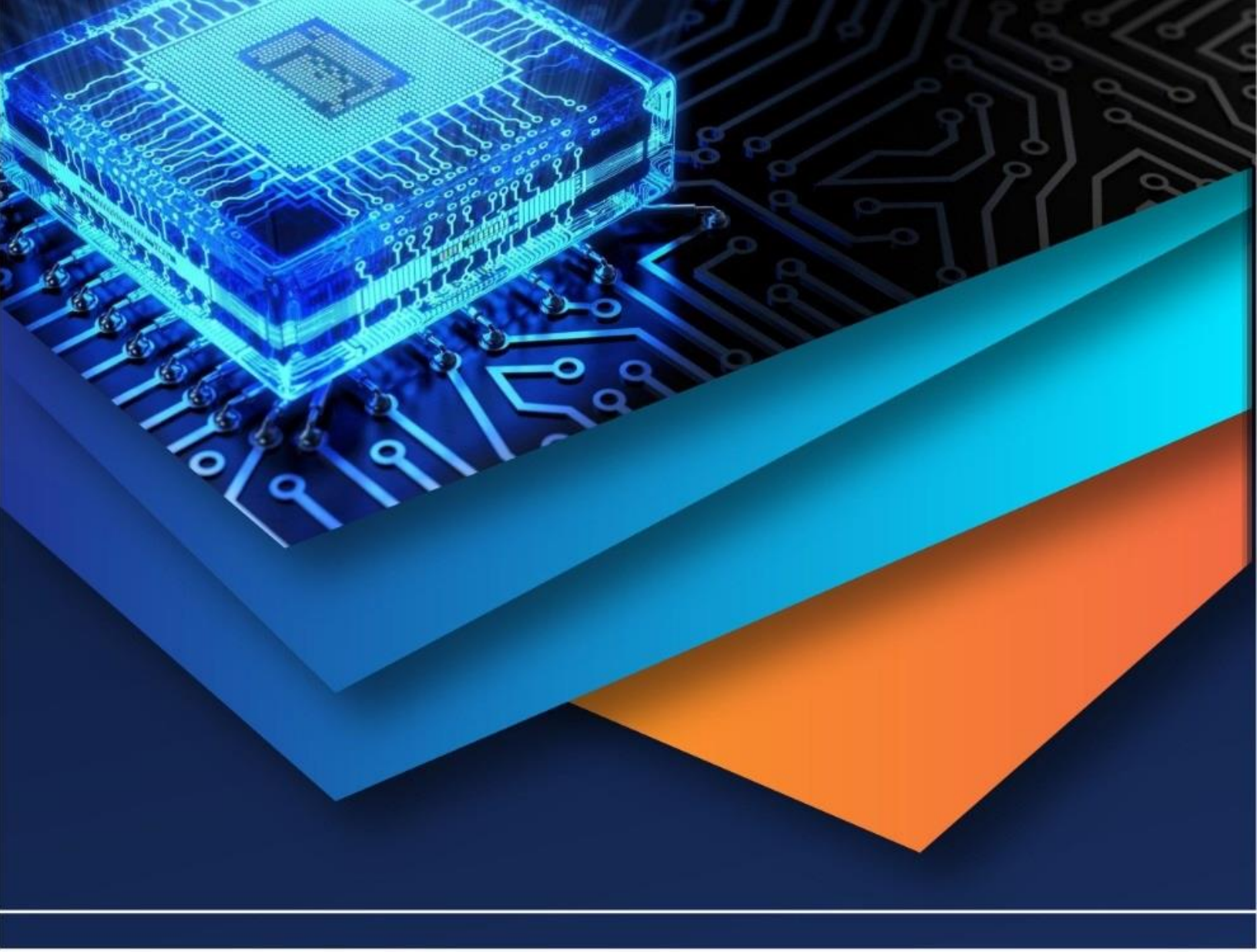
Among the algorithms tested, XGBoost emerged as the top performer, achieving a notable accuracy of 96.72% on unseen test data, underscoring its capability in distinguishing between normal and intrusive activities effectively.

Other models also showed promising results, with Gradient Boosting attaining an accuracy of 96.32%, Random Forest reaching 95.66%, and Decision Tree achieving 95.72% on the test set. These findings indicate that ensemble-based models are well-suited for handling complex and diverse data in intrusion detection.

Our research underscores the effectiveness of ensemble learning methods—particularly XGBoost, Gradient Boosting, Random Forest, and Decision Tree—as potent tools for network intrusion detection. The promising outcomes pave the way for further exploration in cybersecurity, with the ultimate goal of strengthening network defense mechanisms and contributing to the field of automated threat detection and response. For future work, we plan to expand the scope of our approach to encompass more complex network structures and larger datasets to improve detection across a broader spectrum of cyber threats.Additionally, we aim to incorporate real-time analysis capabilities to allow for proactive identification and mitigation of threats as they occur, further enhancing the practical utility of our intrusion detection system.

REFERENCES

[1] Ang, Tiing Leong, et al. "The rise of artificial intelligence: addressing the impact of large language models such as ChatGPT on scientific publications." Singapore medical journal 64.4 (2023): 219-221.
[2] Alwaisi, Z., Soderi, S., & De Nicola, R. (2023, October). Detection of energy consumption cyber-attacks on smart devices. In International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures (pp. 160-176). Cham: Springer Nature Switzerland.
[3] Jian, Jie, Siqi Chen, Xin Luo, Tien Lee, and Xiaoming Yu. "Organized Cyber-Racketeering: Exploring the Role of Internet Technology in Organized Cybercrime Syndicates Using a Grounded Theory Approach." IEEE Transactions on Engineering Management 69, no. 6 (2020): 3726-3738.
[4] Edwards, J. “Mastering Cybersecurity”.
[5] Umer, Muhammad Azmi, et al. "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations." International Journal of Critical Infrastructure Protection 38 (2022): 100516.
[6] Verma, Abhishek, and Virender Ranga. "Machine learning based intrusion detection systems for IoT applications." Wireless Personal Communications 111.4 (2020): 2287-2310.
[7] Dunnett, Kealan, et al. "A trusted, verifiable and differential cyber threat intelligence sharing framework using blockchain." 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2022.
[8] He, Ke, Dan Dongseong Kim, and Muhammad Rizwan Asghar. "Adversarial machine learning for network intrusion detection systems: A comprehensive survey." IEEE Communications Surveys & Tutorials 25.1 (2023): 538-566.
[9] Peppes, Nikolaos, et al. "Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0." Sensors 21.22 (2021): 7475.
[10] B. Groza and P. Murvay, “Efficient Intrusion Detection with Bloom Filtering in Controller Area Networks,” IEEE Trans. Inf. Forensics Security, vol. 14, no. 4, pp. 1037-1051, 2019.
[11] U. E. Larson, D. K. Nilsson and E. Jonsson,” An approach to specification-based attack detection for in-vehicle networks,” IEEE Intell. Veh. Symp. Proc, Eindhoven, pp. 220-225, 2008.
[12] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani,” Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” no. Cic, pp. 108-116, 2018.
[13] E. Seo, H. M. Song, and H. K. Kim,” GIDS: GAN based Intrusion Detection System for In-Vehicle Network,” 2018 16th Annu. Conf. Privacy, Secur. Trust, pp. 1-6, 2018.
[14] K. M. Ali Alheeti and K. Mc Donald-Maier,” Intelligent intrusion detection in external communication systems for autonomous vehicles,” Syst. Sci. Control Eng., vol. 6, no. 1, pp. 48-56, 2018.
[15] A. Moubayed, M. Injadat, A. Shami and H. Lutfiyya,” DNS TypoSquatting Domain Detection: A Data Analytics & Machine Learning Based Approach,” 2018 IEEE Glob. Commun. Conf., Abu Dhabi, United Arab Emirates, pp. 1-7, Dec. 2018.
[16] D. M. Manias, M. Jammal, H. Hawilo, A. Shami, et. al.,” Machine Learning for Performance-Aware Virtual Network Function Placement,” 2019 IEEE Glob. Commun. Conf., Waikolao, HI, USA, Dec. 2019.
[17] Y. Ping,” Hybrid fuzzy SVM model using CART and MARS for credit scoring,” 2009 Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2009, vol. 2, pp. 392-395, 2009.
[18] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami,” Bayesian Optimization with Machine Learning Algorithms Towards Anomaly Detection,” 2018 IEEE Glob. Commun. Conf., pp. 1-6, 2018.
[19] K. Arjunhjhan and C. N. Modi,” An enhanced intrusion detection framework for securing network layer of cloud computing,” ISEA Asia Secur. Priv. Conf. 2017, ISEASP 2017, pp. 1-10, 2017.
[20] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si,” A DataDriven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost,” in IEEE Access, vol. 6, pp. 21020-21031, 2018.
[21] Koskela, Antti, and Tejas D. Kulkarni. "Practical differentially private hyperparameter tuning with subsampling." Advances in Neural Information Processing Systems 36 (2024).
[22] R. Panigrahi and S. Borah,” A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems,” Int. J. Eng. Technol., vol. 7, no. 3.24, pp. 479-482, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY