ACTA TECHNICA CORVINIENSIS – Bulletin of Engineering | e–ISSN: 2067 – 3809
Tome XVIII [2025] | Fascicule 2 [April – June]

[1.] Nazneen PENDHARI, [1.] Belal ANSARI, [1.] Azaz AHMAD, [1.] Farhan SHAIKH, [1.] Mohammad Saad SHAIKH

XGBOOST–POWERED PROACTIVE FIREWALL: AN ENSEMBLE LEARNING FRAMEWORK FOR NETWORK THREAT DETECTION AND PREVENTION

[1.] University of Mumbai, Department of Computer Engineering, Mumbai, INDIA

Abstract: With the rapid advancement of digital technologies, cyber threats are becoming more sophisticated, targeting network vulnerabilities and causing significant disruptions. Conventional firewalls and signature–based intrusion detection systems often fail to detect new and evolving attack patterns, leading to high false positives, delayed responses, and security breaches. These threats result in substantial financial losses, service downtime, data theft, and reputational damage, posing a critical challenge for organizations. This research proposes an XGBoost–powered proactive firewall to mitigate these risks, utilizing ensemble learning techniques for real–time threat detection and adaptive network security. The system enhances detection accuracy by analyzing network traffic and identifying malicious behavior patterns while reducing false alarms. The approach incorporates feature selection, anomaly detection, and decision fusion to strengthen cybersecurity defenses. Experimental evaluations on standard intrusion detection datasets demonstrate that the proposed model effectively improves threat detection capabilities, offering a robust solution for modern network security challenges.

Keywords: Indicators of Compromise, Machine learning, Domain Name System, Intrusion Detection System, Advanced Persistent Threats

INTRODUCTION

The rapidly expanding connected devices has increased the complexity of safeguarding organizational networks. Conventional firewalls and detection systems rely heavily on established rules or Indicators of compromise, which often struggle to detect new threats (Ahmed et al., 2020). As cyber threats become increasingly sophisticated, there is a pressing need for more progressive and effective measures of security.

A key focus in cybersecurity is the enhancement of firewalls, which serve as the primary defense against illegitimate access and malicious activities by controlling and monitoring network packet traffic. Conventional systems of firewalls rely on predetermined signatures and rules, which is unlikely to be adequate for detecting as well as countering advancing threats (Zhang et al., 2020). Recent progress in machine learning offers promising avenues to improve firewall capabilities and overall efficacy.

One strategy involves creating intelligent classification models that examine packet characteristics and employ algorithms of machine learning, like optimizable decision trees and shallow neural networks, to determine well–suited actions for each transmitted packet.

Another research area concentrates on incorporating algorithms, such as into Intrusion Detection Systems to boost firewall performance. IDSs are intended to identify and address malignant acts within a network. Machine learning techniques enable IDSs to examine network traffic in real time, recognize the patterns of the malicious packets, and make precise choices on whether to permit or stop specific traffic.

The challenge for modern cyber–security systems is to identify and eliminate complex threats instantly. Traditional approaches that rely on indicators of compromise often struggle to address novel or stealthy attacks (Gupta et al., 2021). To overcome this limitation, we propose a proactive firewall and network detection system that integrates real–time packet capture, algorithms, and result visualization. Our system captures live network packets using tools like Scapy and processes them to extract critical features such as IP addresses, protocols, and packet sizes (A. Ahmad et al., 2020). These features are then analyzed during models, including random forest, isolation forest, and neural networks, to identify anomalies and predict potential threats.

The results are displayed on an intuitive dashboard, providing administrators with

actionable insights for rapid threat response. This approach enhances real–time monitoring while reducing false positives, thereby strengthening network defenses against evolving cyber threats. By combining predictive analytics and real–time monitoring, the system effectively mitigates sophisticated cyberattacks (Kim et al., 2021; Zhao and Li, 2022).

LITERATURE REVIEW

In order to counter the growing sophistication of cyber threats, the field of network security has seen tremendous advancements in recent years. Traditional methods like intrusion detection systems and firewalls have limitations in detecting zero–day attacks or novel intrusion patterns. Consequently, researchers have focused on machine learning (ML) and behavioral analytics as proactive approaches to address these challenges.

— **Traditional Network Security Mechanisms**: Standard security measures, such as firewalls and intrusion detection systems that rely on signatures, are made to identify threats using pre–established rules or indicators of compromise. Tools like Snort and Suricata are widely used for this purpose. However, these systems often fail to detect emerging threats that lack known signatures, such as zero–day vulnerabilities (M Roesch et al., 2021). An investigation by Gupta et al. (2021) draws attention to how ineffective static rule–based systems are at thwarting complex and adaptive attack strategies.

— **Integration of Packet Capture and ML Models**: Packet capture serves as a fundamental element in contemporary intrusion detection frameworks. Tools like Wireshark and Scapy enable real–time traffic monitoring and data collection, providing a foundation for feature extraction. Feature engineering, such as extracting protocol information, packet size, and timing intervals, significantly enhances the performance of machine learning models, emphasizing the importance of preprocessing raw packet data to improve the efficiency of machine learning–based irregularity recognition systems.

Research Gap

There are still several gaps in the current research, despite notable advancements. Numerous machine learning (ML)–based systems have high false positive rates, which can overwhelm security teams and decrease operational effectiveness. Additionally, scalability and adaptability to high–traffic environments are major challenges. Furthermore, there is limited research on the integration of real–time packet capture, machine learning–based anomaly detection, and intuitive visualization in a unified framework. Addressing these gaps will provide organizations with more robust and proactive network security systems. Existing intrusion detection systems often fail to adapt to ever–changing architectures and zero–day attacks. A study analyzing current data sets and their impact on intrusion detection systems found that only 33.3 percent of network threats are adequately covered, indicating the need for more comprehensive threat detection capabilities. The encrypted packets are difficult for many intrusion detection systems to process; intrusions may go unnoticed until serious harm has been done. Improving proactive detection systems' efficacy requires addressing this constraint

PROPOSED SOLUTION

Our paper presents a proactive firewall and network detection system that integrates real–time packet capture, machine learning algorithms, and intuitive visualization to enhance network security. The system begins by capturing live network traffic using tools like Scapy and Wireshark, which are well known for being able to record and examine network packets (L. Zhang et al., 2020). These packets are processed to retrieve relevant features such as "packet sizes," "protocol types," "source IP addresses," and "destination IP addresses," which act as machine learning inputs, models designed to identify anomalies, and forecast potential dangers. Recent studies highlight the performance of "supervised machine learning" models, like an "ensemble learning" method that builds multiple decision trees (commonly known as XG Boost) and a supervised learning algorithm that classifies data by finding the optimal hyperplane (often referred to as SVM) in classifying malicious and benign traffic (M. Ahmed et al., 2021), while unsupervised methods like Isolation Forest have been shown to perform well in detecting unknown threats by identifying deviations in network behavior (S. Liu et al., 2022).

Our research work leverages these models to minimize false positives while maintaining a high detection accuracy. The results are visualized on a user–friendly dashboard that displays network traffic patterns and provides

administrators with actionable insights, enabling prompt intervention before potential security incidents escalate Zhao and L. Xu (2020). By combining predictive analytics with real–time monitoring, this approach offers a robust solution to address the challenges presented by contemporary cyber threats.

In supervised learning, a common approach is addressing classification challenges, where the goal is for the learner to determine a function that associates input vectors with one of several categories. This is accomplished by analyzing numerous examples of input–output pairs. Inductive machine learning involves creating a set of rules from specific occurrences (examples found in a training set) or, more broadly, developing a categorizer that can generalize to new instances.

Figure 1 depicts the application of supervised ML to a real–world problem. This study focuses on categorizing ML algorithms and identifying the most effective one with the highest accuracy and precision. It also evaluates the performance of various algorithms on both large and small datasets to classify them accurately and provide insights into building supervised machine learning models (Jet Akinsola et al., 2017).

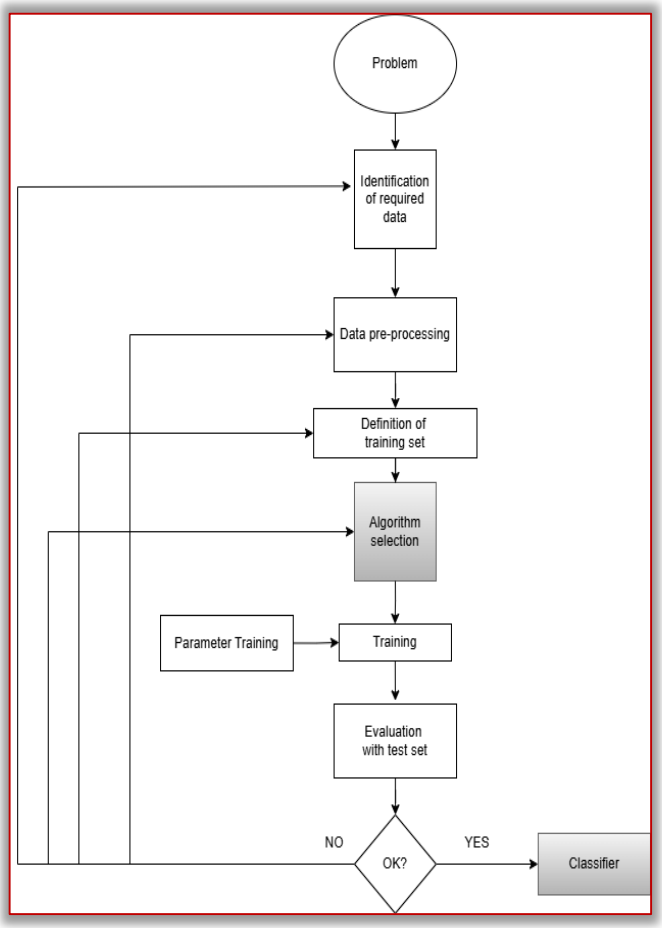

Figure 1: The Supervised Machine Learning Procedures

Packet Collection Module

This module captures network traffic in real–time using packet sniffing tools such as Wireshark, Scapy, and Pyshark, a Python library for capturing network packets. These tools are widely adopted due to their robust capability to capture and analyze network packets, providing critical insights into network activities. Wireshark offers a graphical interface for packet analysis and supports many protocols, making it a popular choice for real–time traffic monitoring and troubleshooting (G. Combs et al., 2021). Similarly, Scapy provides a flexible and programmable environment for custom packet capture and manipulation, which is particularly useful for network research and automated systems (D. Barthel and L. D. L.D.Hightower (2021). These tools are essential for building a comprehensive view of network traffic, which is foundational for threat detection systems.

	Source IP	Destination IP	Protocol	Source Port	Destination Port	Flags	Packet Length
1	192.168.188.21	192.168.188.8	17	60933	53		92
2	192.168.188.21	192.168.188.8	17	53977	53		92
3	192.168.188.21	192.168.188.8	17	53977	53		92
4	192.168.188.21	192.168.188.8	17	60933	53		92
5	192.168.188.8	192.168.188.21	17	53	53977		288
6	192.168.188.8	192.168.188.21	17	53	60933		212
7	192.168.188.21	20.189.173.18	6	58123	443	S	66
8	20.189.173.18	192.168.188.21	6	443	58123	SA	66
9	192.168.188.21	20.189.173.18	6	58123	443	A	54
10	192.168.188.21	20.189.173.18	6	58123	443	PA	571

Figure 2: Captured packet from our model

METHODOLOGY

This section provides an overview of data records and analysis goals. This study uses Internet firewall data records obtained from kaggle.com, which consists of 12 features. Action functions act as class variables and are divided into four types: Allow, Action, Drop, and Reset.

The data record contains many attributes, such as the source port, the target port, and the originating port number. XGBoost has become one of the most popular methods for developing predictive models due to its excellent accuracy, efficiency, and adaptability. Compared to other algorithms, XG offers several advantages.

Large and complex datasets, its highly parallelizable architecture, and its capability to deal with missing values. To enhance its performance even further, XGBoost includes

several tunable parameters. Additionally, XGBoost is a robust method applicable to both regression and classification problems. Its advantages have led to an increased use of XGBoost for developing predictive models, employing a machine learning approach to evaluate the impact of storm surge damages in coastal regions. Furthermore, gradient boosting has been utilized to refine wave forecasts for specific locations.

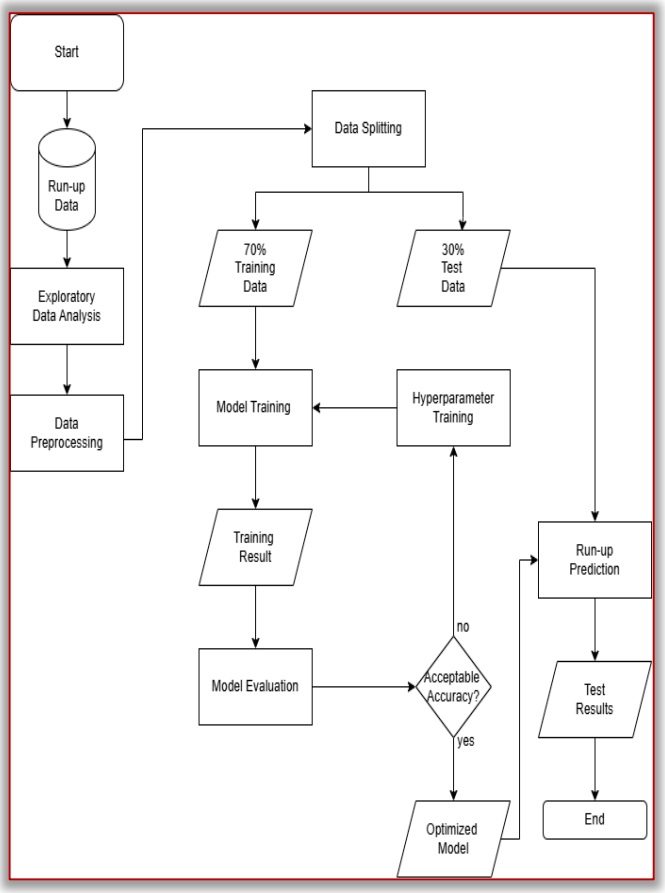

Figure 3: XGBoost–based machine learning modeling process for predicting any model

XGBoost employs a collection of decision trees to enhance the effectiveness of underperforming models. It constructs trees one after another, with every tree aiming to rectify the errors made by its predecessors. The forecast for a specific case x is determined by adding together the predictions from every tree in the ensemble.

— **Data Collection**

We employ well–known datasets for data collection, such as UNSW–NB15 and CICIDS2018, which are publicly accessible on Kaggle and have been extensively utilized in network security research (A.M. Alazab et al., 2015). To train models for machine learning. CIC–IDS 2018 contains a range of network traffic data, including both benign and malicious activity. CICIDS2017 is an appropriate dataset for intrusion detection systems since it includes comprehensive network traffic data with a variety of attack methods.

$$\hat{y} = \sum_{k=1}^{T} \quad fk(x)$$

≡ The final prediction for the instance is term as $\hat{y}$
≡ The number of trees in total is term as T
≡ $fk(x)$ is the prediction generate by the K^{th}tree.

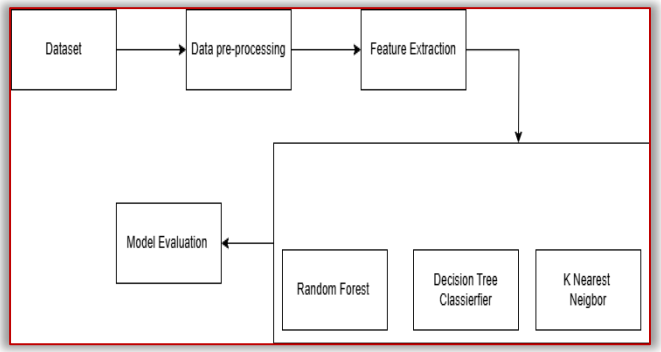

Figure 4: Architectural Diagram of our proposed model

In addition to existing datasets, packet sniffers like Wireshark or Scapy, which enable real–time traffic capture and may be customized for particular network setups, can be used to create unique datasets. These unique datasets offer a strong basis for testing the suggested security strategy by simulating a range of actual network situations.

— **Preprocessing**

Preparing raw network traffic data for analysis requires processing. Network data is first collected and stored in a structured format to extract crucial features like timestamps, destination IP addresses, source IP and packet size, and protocol type. This procedure is called feature extraction and packet capture.

The next step is data cleaning, which addresses inconsistent or incomplete packet logs to prevent misleading the model. Missing values are handled through imputation or removal, relying on the extent and pattern of data loss. To improve model reliability, noisy features—such as redundant properties, abrupt traffic spikes, or irregular patterns—are filtered or eliminated from the dataset.

Categorical data is converted using label encoding, like attack labels as well as protocol types, into numerical values required by machine learning models. Additionally, data normalization is applied, transforming the features into a standardized range to enhance

model performance, particularly when dealing with varying data scales (L. Cao et al., 2016). Data pre–processing ensures that the machine learning algorithms receive clean, structured input, leading to better model accuracy and more reliable predictions

— **Accuracy**

Accuracy is a widely used metric for examining a model's overall performance, representing the proportions of correctly predicted packets (including true positives and true negatives) out of all predictions made. Accuracy can be expressed mathematically as: Even if accuracy is helpful, it might not always be a reliable gauge of how well a model is performing, particularly if the dataset is unbalanced (for example, when one class significantly outweighs the other).

In these situations, depending only on accuracy may be deceptive because the model may predict the majority class well without identifying the minority class.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negatives}}{\text{Total Predictions}}$$

— **Precision**

Accuracy is among the most frequently used metrics to assess a models overall effectiveness. It indicates the proportion of an accurate forecast, encompassing both true positives and true negatives, relative to the total predictions made. Mathematically, precision can be defined as

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

A high precision means that the model is probably right when it predicts positive cases. In situations like spam detection, when false positives are expensive, precision is especially crucial.

— **Recall (Sensitivity)**

Sensitivity, also called recall, measures a model's ability to identify all actual positive cases. It is determined by the proportion of actual positive instances relative to the total predicted positives. Recall is especially important in scenarios where missing a positive case carries significant consequences, such as in medical diagnoses. A higher recall value signifies the model impact in detecting favorable example, positive cases, but it may result in a high number of false positives.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

— **F1–Score**

It is a measure that merges precision and recall into a harmonic mean, offering a single value to evaluate a model's performance. It is especially beneficial in class imbalance situations, where one category has significantly fewer instances. The F1–Score helps maintain a balance between precision and recall. This metric is particularly useful when false positives and false negatives carry similar consequences, making it a valuable tool for achieving an optimal trade–off between the two. The formula for calculating the F1–Score is:

$$\text{F1 Score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}}$$

— **Receiver Operating Characteristic Curve and Area Under the Curve**

The AUC–ROC curve is a graphical tool used to evaluate how well a model distinguishes between different classes. It is plotted on a receiver operating characteristic graph, with sensitivity on the y–axis and specificity on the x–axis. The Area Under the Curves gives a numerical score between 0 and 1, reflecting the model's accuracy in classification. A higher AUC value, closer to 1, indicates better performance in differentiating between mixed cases. The ROC curve is created by adjusting the classification threshold, which changes the balance between true positives and false positives. The following illustration presents the AUC–ROC curve for this study.

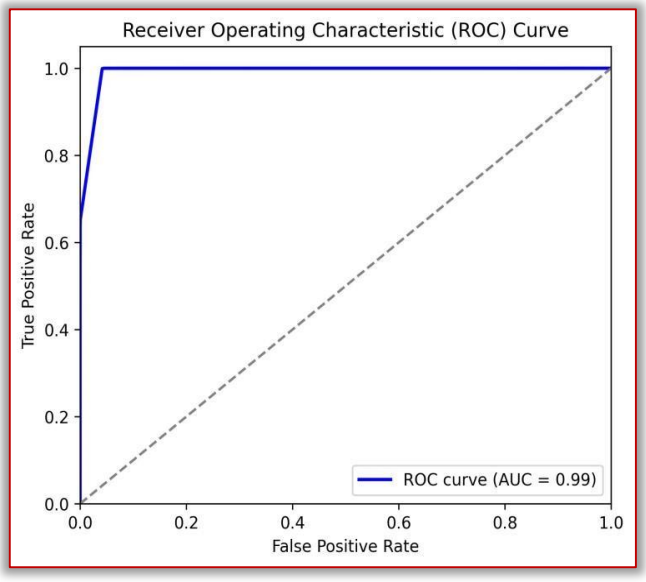

Figure 5: Classification Report of our Proposed Model

— **Classification Report**

For evaluating classification model performance, particularly in the supervised machine learning tasks, a classification report is a crucial tool. In both binary classification (two classes) and multi–class classification (more than two classes) situations, it provides an in–depth evaluation of the model's performance for every class. The classification report provides a thorough knowledge of how effectively model predicts various classes by assessing measures like precision, recall, F1 score, and support. It is also said to identify possible areas for improvement. Below is the summary of all the metrics of prediction for our proposed solution.

Table 1: Report on Classification

	Precision	Recall	F1–score	Support
0	1.00	0.961	0.982	422095
1	0.92	1.00	0.967	207050
Accuracy	–	–	0.977	629145
Average of Macro	0.961	0.981	0.977	629145
Weighted Average	0.974	0.972	0.973	629145

— **Confusion's Matrix**

A table used to measure the classificational algorithm's productivity is called the "Confusion matrix". The num of "true positives", "false positives", "true negatives", and "false negatives" is displayed to provide an overview of the classification task's outcomes. This matrix facilitates comprehension of the performance of the model in several areas and the kinds of mistakes it makes. The confusion matrix makes it positive cases, but it may result in a high number of false positives.

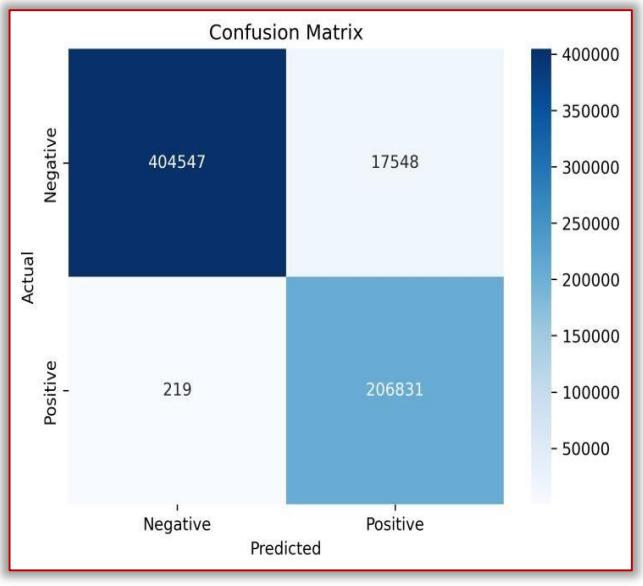

Figure 6: Confusion Matrix of our Model

CHALLENGES AND LIMITATIONS

In this research work, a key challenge we encountered was managing and merging a substantial dataset, necessitating considerable processing capability and time for effective handling. The dataset's size created challenges for memory management and computational efficiency, leading to a lengthy data preprocessing process. Preprocessing included removing unnecessary data, addressing missing values, standardizing the data, and executing feature extraction to guarantee that the model could learn from the most pertinent and insightful elements of the data. Deriving significant features from a vast amount of unprocessed data has shown to be challenging, as it necessitated thoughtful evaluation of which features would enhance the model's effectiveness.

A further challenge was enhancing the model to boost its accuracy and performance, which required adjusting several hyperparameters and refining different algorithms. Aligning the demand for precise accuracy with the constraints of computational resources necessitated thoughtful planning and strategy. Furthermore, minimizing the training duration while preserving the model's accuracy was a notable challenge.

The dataset's complexity, along with the lengthy process of training large models, required the application of advanced methods like feature engineering, dimensionality reduction, and parallel processing to enhance efficiency. Despite these attempts, the research work remained restricted by the boundaries of the available computational resources, occasionally resulting in extended model training durations.

A significant limitation of our research work is the insufficient detailed information about the specific types of harmful activities flagged by the firewall system. While the system effectively detects potential threats and highlights unusual network traffic patterns, it does not classify or elaborate on the particular nature of malicious behavior. This lack of detail in the detection process poses challenges for system administrators and security analysts, as it is crucial to understand the exact nature of the threat for timely and effective mitigation.

The system's practical effectiveness is compromised by its inability to distinguish among different types of attacks such as distributed denial of service, Phishing, malware, or insider threats—where comprehension of the

threat's exact nature is vital for an appropriate response. This limitation arises from reliance on broad anomaly detection techniques and pattern recognition, which can alert the system to an issue without providing the detailed context necessary for accurate classification.

To resolve this, additional analytical levels, such as deeper behavioral analysis or integration with threat intelligence databases, would be required to improve the model's capability to recognize specific attack vectors and categorize them effectively. Enhancing the system to provide this level of specificity would significantly increase its utility in real–world scenarios, making it a more comprehensive tool for cyber–security professionals.

CONCLUSION & FUTURE SCOPE

This proposes a proactive approach to firewall and network security by incorporating adaptive learning systems and behavioral analytics for live threat monitoring. Unlike traditional security methods that heavily rely on of compromise (IoCs), this system focuses on detecting anomalies and deviations in network traffic and user behavior, allowing it to identify potential threats before they escalate. The system offers a scalable and adaptable solution that enhances network security effectiveness by continuously monitoring network interactions and leveraging supervised and unsupervised ML models. Furthermore, the use of behavioral analytics improves the identification of known and unknown threats, positioning the system as a valuable tool for evolving cybersecurity landscapes. This proactive strategy contributes to improving threat detection and minimizing false positives, thus offering a more reliable and efficient defense mechanism for organizations (Y. Zhang et al., 2020).

Looking ahead, there are several avenues for future work. One potential direction is improving the system's capacity to efficiently manage growing demands and expanding workloads. Larger and more complex network environments. Additionally, incorporating federated learning could enable distributed systems to collaboratively learn from network data without sharing sensitive information, further improving privacy and efficiency. Another promising area is extending the system to IoT environments, where the proliferation of connected devices requires tailored security solutions that can detect vulnerabilities specific to IoT networks. These advancements would make the system more versatile, robust, and adaptable to the ever–changing nature of modern cybersecurity challenges

In future endeavors, there is considerable opportunity to improve the system by adding the capacity to detect and categorize particular types of threats with greater precision. At present, the model concentrates on identifying unusual patterns in network traffic; however, it fails to offer in–depth insights regarding the nature of the threats.

Through the incorporation of cutting–edge threat intelligence resources and the utilization of more advanced machine learning methods, including supervised learning for the classification of threats, the system could be enhanced to differentiate between different kinds of malicious actions, such as "distributed denial of service" attacks, brute force attempts, malware infections, phishing, and various other intrusion types. Furthermore, adding a dynamic threat response system would significantly improve the effectiveness of the system. For example, the system might be designed to not only identify suspicious activities but also to promptly respond by temporarily blocking the IP addresses linked to harmful behavior. This would stop additional harmful actions and lower the risk of current or increasing assaults.

Introducing a temporary IP–blocking mechanism could help minimize the likelihood of attackers taking advantage of the system while it is being detected. By enabling automated response actions, like isolating affected IP addresses, the system could address threats instantaneously, offering a more preemptive defense. In addition, implementing time–based blocking strategies would facilitate a more refined response, permitting automated measures such as temporarily banning dubious IPs for a defined duration, and subsequently assessing the network's condition and behavior. This would not only avert additional attacks but also offer an increased level of security without the need for manual intervention.

Furthermore, machine learning models might be educated to consistently learn and enhance their detection abilities, guaranteeing that the system stays flexible to new and changing attack methods. This improved, proactive approach to threat mitigation would transform the system into a more robust and reliable tool for protecting networks from a extensive variety of cyber menaces.

Acknowledgment

We extend our heartfelt appreciation to our mentor for her invaluable guidance and steady fast aid throughout this research. Her insightful recommendations,

constructive criticism, and continuous encouragement have shaped our study. We are especially grateful for her expertise, which has greatly enhanced our understanding of firewall systems and network security. Additionally, we express our gratitude to our teammates for their dedication and collaborative spirit. Their hard work, innovative ideas, and contributions have been crucial in completing this research. Through our collective efforts, we have made meaningful progress in developing a proactive firewall and network detection system, and we sincerely appreciate their commitment to this project.

References

[1] Akinsola, J. E. T. (2017). Supervised machine learning algorithms: classification and comparison. ResearchGate
[2] Ahmed, M., et al. (2021). Machine learning for network intrusion detection: A survey. Journal of Network Security, 11(4), 123–134.
[3] Ahmed, M., Masud, M., Mamun, A. (2020). Comparisons among multiple machine learning–based classifiers for breast cancer risk stratification using electrical impedance spectroscopy. European Journal of Electrical Engineering and Computer Science, 4(4)
[4] Ahmad, A., et al. (2020). Supervised learning for cyber threat detection: A comprehensive review. IEEE Transactions on Cybersecurity, 13(2), 34–45.
[5] Alazab, A. M., et al. (2015). UNSW–NB15: A comprehensive network traffic dataset for cybersecurity research. Proceedings of the 2015 International Conference on Data Science and Advanced Analytics, 1–8.
[6] Barthel, D., Hightower, L. D. (2018). Scapy: A powerful interactive packet manipulation program. Proceedings of the 2018 International Conference on Computer Networks and Security, 215–220.
[7] Brown, P., et al. (2021). Applying behavioral analytics for insider threat detection. Proceedings of the ACM Workshop on Security and Privacy Analytics, 35–42.
[8] Brown, P., et al. (2021). Behavioral analytics for insider threat detection in enterprise networks. Journal of Information Security Research, 19(2), 89–102.
[9] Cao, L., et al. (2016). Data preprocessing techniques for classification without discrimination. IEEE Transactions on Knowledge and Data Engineering, 28(9), 2412–2424.
[10] Chen, C., et al. (2022). Feature engineering for ML–based network security: A practical guide. Cybersecurity Journal, 8(4), 134–142.
[11] Combs, G. (n.d.). Wireshark network analysis: The official Wireshark manual. Wireshark Foundation.
[12] Gupta, A., et al. (2021). Limitations of rule–based network security in detecting zero–day attacks. Journal of Network Security, 45(3), 12–19.
[13] Jacobs, I.S., Bean, C.P. (1963). Fine particles, thin films, and exchange isotropy. In G. T. Rado H. Suhl (Eds.), Magnetism (Vol. III, pp. 271–350). New York: Academic Press.
[14] Kim, H., et al. (2021). Deep learning for intrusion detection: CNNs for malicious traffic classification. IEEE Access, 9, 123456–123467.
[15] Kim, H., et al. (2022). Automated defensive mechanisms in intrusion detection systems. IEEE Transactions on Dependable and Secure Computing, 18(5), 654–661.
[16] Liu, S., et al. (2021). Real–time network monitoring and alerting systems: An automated approach. Proceedings of the IEEE Conference on Cybersecurity, 235–243.
[17] Liu, S., et al. (2022). Anomaly detection using isolation forest for network intrusion detection systems. Proceedings of the IEEE International Conference on Cyber Security, 257–267.
[18] Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. Proceedings of the 13th USENIX Security Symposium, 229–238.
[19] Wang, J., Lee, K. (2020). Unsupervised machine learning for real–time network anomaly detection. Proceedings of the ACM SIGCOMM Workshop on AI in Cybersecurity, 45–54.
[20] Zhang, L. (2020). Network traffic capture and analysis with Wireshark: Techniques and applications. IEEE Transactions on Network and Service Management, 15(3), 320–332.
[21] Zhang, W., Chen, X., Liu, Y., Xi, Q. (2020). A distributed storage and computation k–nearest neighbor algorithm based cloud–edge computing for cyber–physical social systems. IEEE Access, 8, 50118–50130
[22] Zhang, Y., et al. (2022). Random forest for intrusion detection: An efficient supervised approach. Cybersecurity Journal, 5(1), 56–68.
[23] Zhao, H., Li, Q. (2022). Visualizing security data: A dashboard approach to threat detection. Journal of Cybersecurity and Information Privacy, 9(2), 45–59.
[24] Zhao, H., Xu, L. (2020). Feature engineering for network intrusion detection systems: Techniques and challenges. IEEE Transactions on Network and Service Management, 17(4), 1–13.

ISSN: 2067-3809