International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 8, pp. 2886-2904
doi: 10.25147/ijcsr.2017.001.1.193
https://stepacademic.net

Short Paper

Firewall Defense and Response Policy towards Resisting Attacks on Network Logs

Onate E. Taylor
Department of Computer Science, Rivers State University, Nigeria
taylor.onate@ust.edu.ng
(corresponding author)

Promise S. Ezekiel
Department of Computer Science, Rivers State University, Nigeria
Ezekielpromise27@gmail.com

Date received: September 27, 2023
Date received in revised form: May 17, 2024; May 26, 2024
Date accepted: May 27, 2024

Recommended citation:

> Taylor, O. E., & Ezekiel, P. S. (2024). Firewall defense and response policy towards resisting attacks on network logs. *International Journal of Computing Sciences Research*, *8*, 2886-2904. https://doi.org/10.25147/ijcsr.2017.001.1.193

Abstract

Background – In an era marked by escalating cyber threats, safeguarding network infrastructure and preserving the integrity of network logs have become paramount concerns for organizations worldwide.
Objective: This paper proposes a robust Firewall Defense and Response Policy leveraging a state-of-the-art Gradient Boost Classifier to achieve exceptional accuracy in detecting cyber threats.

Methods – The proposed methodology combines advanced machine learning techniques with an in-depth analysis of network logs. The model was trained on a comprehensive dataset, downloaded from Kaggle.com, comprising 65,533 instances of diverse attack vectors. This training enables the model to discern subtle patterns indicative of cyber threats.

Results – The Gradient Boost Classifier achieved an accuracy of 99.99% in identifying and thwarting malicious intrusion attempts. The Response Policy integrates an adaptive approach, dynamically adjusting countermeasures based on the severity and nature of detected anomalies.

Conclusion – Through extensive experimentation and validation, the proposed approach demonstrates superior performance in detecting and mitigating a wide spectrum of attacks, including sophisticated and evasive tactics. This paper contributes a highly effective and resilient framework for bolstering network security, empowering organizations to fortify their defenses against evolving cyber threats and safeguard the integrity of their network logs.

Recommendation – Organizations should implement the proposed Firewall Defense and Response Policy across various environments, regularly update the training dataset with new attack vectors, and periodically re-evaluate the model to maintain its effectiveness. Integrating this policy with existing security systems, training personnel, and promoting awareness about cyber threats will optimize its implementation. Continued research into advanced machine learning techniques will further enhance the system's accuracy and resilience.

Keywords – Firewalls, Network Logs, Cyber Attacks, Gradient Boosting Technique

INTRODUCTION

Security breaches, theft, and network disruption are all problems that computer scientists from all backgrounds face. As the number of people who rely on the internet grows, so does the significance of a reliable security system. When placed in front of a web application, a WAF prevents unauthorized users from accessing the application over the internet. In contrast to a proxy server, which hides the true identity of its user, the web application firewall (WAF) keeps the web server hidden from the client by blocking suspicious requests. A WAF is managed by policies and a trained module that can anticipate incoming requests. These policies are an effort to reduce the risk of application vulnerabilities by blocking potentially dangerous communications (Ito & Iyatomi, 2018).

Security solutions like firewalls and intrusion detection systems (IDS) have been the subject of numerous attempts. Most firewalls and intrusion detection systems (IDS) operating at the network layer do not perform application layer inspection of HTTP packets. Therefore, they can't guarantee the safety of web servers in any way. One of the most tempting entry points into a company's IT infrastructure is through its web applications, especially those hosted in the cloud. A lack of web security measures can expose an organization to internal data leaks, financial losses, and website manipulation. Attacks like SQL injections, XSS, and DDoS can all be prevented with the help of a web application firewall (WAF). To protect websites, WAF employs IDS techniques at the application layer (Moradi et al., 2019).

Most modern WAFs rely on signature-based detection methods. The needs of the modern environment for filtering out attacks over any kind of network are not being met by traditional firewall systems. Hackers are getting more intelligent and inventive in their attempts to break into systems. Therefore, there are situations in which a signature-based security system is not the best option. Unfortunately, zero-day attacks are beyond the capabilities of the signature-based system. Any Distributed Denial of Service (DDoS) attack on a web service system would have a direct impact on the service, the business, and the economy as a whole. Training the system results in a deep learning-based web application firewall that can identify novel attack vectors, tactics, and behaviors (Krishnan *et al.,* 2022).

The security model used by a WAF is typically dependent on the policies it employs. There are two major types of security models: negative and positive. If a WAF is using a negative security model, all traffic is allowed through unless it specifically violates one of the rules. The negative security model simply lets through any traffic that doesn't conform to the rules. In most cases, a signature-based method is used to implement the negative security model. Using pattern-matching techniques, the signature-based approach can identify malicious incoming data and prevent it from entering the system (Clincy & Shahriar, 2018).

In contrast, a WAF that uses a positive security model will let traffic through if it meets certain criteria. All non-compliant traffic is immediately terminated. Both positive and negative security models have their merits and drawbacks. Since hackers can always find new ways to circumvent security measures, the negative security model, for instance, may not be enough. In contrast, the positive security model may be inflexible where the functionality or content presented by the application is evolving, requiring extensive planning and implementation based on a thorough knowledge of the application (Thang, 2020)

LITERATURE REVIEW

Following the study, Ito and Iyatomi (2018) performed a statistical analysis to distinguish attack-indicating parameters and features from non-attack-indicating ones in HTTP traffic and attack traffic. The authors compared attack and normal traffic by analyzing and contrasting the various features of the standard datasets ISCX, CISC, and CICDDoS. Using a dataset gathered from the simulation environment, a layered architecture model was developed to detect DDoS, XSS, and SQL injection attacks. The DDoS detection model, designed with an accuracy of 97.57 percent, was implemented in the first layer of the LSTM-based layered architecture, while the XSS and SQL injection layer, implemented in the second layer, achieved an accuracy of 89.34 percent. Since HTTP traffic is typically faster, it was examined first, filtered out, and then sent on to the next layer. The web application firewall (WAF) supplements the functionality of a traditional network firewall by performing application-level filtering.

Applebaum et al. (2021) gave a quick summary of how WAFs have evolved thanks to the application of machine learning techniques. Their benefits and drawbacks are analyzed, and unanswered questions are pointed out. It evaluates which ones can protect against zero-day attacks and are simple to set up and keep current. It was discovered that machine-learning-based approaches have advantages over signature/rule-based approaches because they can mitigate the risk of zero-day attacks and are typically less complicated to set up and maintain. The survey also found that there is room for more research into the efficacy of machine-learning-based WAFs in protecting against modern attack patterns aimed at web application frameworks.

Prabakaran et al. (2022) developed a set of rules and regulations to restrict access to potentially harmful networks. Such measures are insufficient to prevent attacks using a large number of unique socket identifiers. To detect potentially malicious links and likely attack targets in a network, traditional network threat intelligence data is used to train machine learning algorithms. Decision Table (DT), Bayesian Network (BayesNet), Naive-Bayes, C4.5, and DT algorithms are used to forecast the target host that will be attacked based on traditional network data. According to the results of the experiments, the Bayesian Network algorithm has the highest average prediction accuracy (92.87 percent), followed by the Native-Bayes Algorithm (87.81 percent), the C4.5 Algorithm (84.92%), and the Decision Tree Algorithm (83.18%). Over four hundred and fifty thousand (451,000) login attempts were recorded from 178 different countries across more than 70,000 unique IP addresses and 41,000 unique source ports in a massive dataset collected from nine honeypot servers.

Appelt et al. (2018) introduce ML-Driven, a machine learning and evolutionary algorithm-based method for discovering vulnerabilities in WAFs that SQL injection attacks can exploit automatically. At first, ML-Driven will generate an assortment of attacks and send them to the system that the target WAF is guarding. Then, ML-Driven picks attacks that display patterns (substrings) linked to getting around the WAF and refines them to produce new bypass attacks. Attack patterns are learned incrementally using machine learning from previously generated attacks based on testing results, i.e. whether or not the WAF blocks or is bypassed by the attack. The authors integrated ML-Driven into a tool and tested it against ModSecurity, a popular open-source WAF, and a proprietary WAF safeguarding a bank. Their experimental results show that ML-Driven is proficient in creating SQL injection attacks that are immune to WAFs and in spotting patterns of attack.

Shaheed and Kurdy (2022) presented a framework for a web application firewall that uses machine learning and features engineering methodologies to identify and mitigate online-based attacks. The model analyzes incoming HTTP requests, extracting four features: URL, payload, and headers, and classifying each request as normal or anomalous based on predetermined criteria. The model uses five features: request length, allowed character ratio, special character percentage, and attack weight. The model was assessed using updated datasets and four classification algorithms, with two methods used to mitigate overfitting. A typical request has a short length, high allowed character ratio, low special character ratio, and zero attack weight. Anomaly requests have a significant increase in request length, decreased allowed character percentage, increased special character percentage, and increased numerical attack weight. The model achieved a classification accuracy of 99.6% on commonly used research datasets and 98.8% on real web server datasets.

Ito and Iyatomi (2018) proposed an approach utilizing deep neural networks for feature learning and isolation forests for classification. On the CSIC 2010 data set, the authors compared their method to others that did not use feature extraction models. The proposed deep neural network also benefited from a variety of learning and activation functions. The results demonstrate the superior accuracy of deep models over feature-less methods.

Ito and Iyatomi (2018) proposed an efficient machine learning approach to solve these issues. Their proposed approach uses a character-level convolutional neural network (CLCNN) with very large global max-pooling for extracting the feature of HTTP requests and identifying it into normal or malicious requests. The authors evaluated their system on HTTP DATASET CSIC 2010 dataset and achieved 98.8% accuracy under 10-fold cross-validation and the average processing time per request was 2.35ms.

Taylor and Ezekiel (2022) trained a Recurrent Neural Network (RNN) model on a dataset that comprises different types of web application attack types (XSS, SQLi, and Shell). To get ready for preprocessing, a highly imbalanced dataset was fixed using a Random Over Sampling strategy. The dataset underwent pre-processing, which included data cleaning and tokenization after the imbalanced problem was fixed. To train our RNN model, we took the tokenized data and converted it into an array. Each epoch displays the accuracy and loss values obtained by the model for both training and testing data, and our proposed model was trained over two (2) epochs. After training, the proposed RNN model achieved a 99.96% accuracy on testing data and a 99.91% accuracy on training data. We also used a Python flask to publish our RNN model online, creating a secure framework for monitoring and protecting against a variety of payload attacks on web-based software. Attacks against web applications are the focus of this paper.

Rajesh et al. (2021) analyzed a variety of characteristics to differentiate between normal and DDoS attack traffic, such as UDP flood attacks, ICMP ping flood attacks, TCP SYN flood attacks, and land attacks. K-nearest neighbor, decision tree, random forest, and naive Bayes were just some of the machine learning techniques the authors compared and contrasted.

Ito and Iyatomi (2018) presented a comprehensive approach to cyber security evaluation and threat detection through the integration of penetration testing, web mining, text mining, and machine learning techniques. By applying penetration testing, vulnerabilities for popular cyberattacks are identified, leading to the formulation of security suggestions. Web mining techniques are then utilized to understand visitor behavior and assess cyber security risks. An intelligent host-based intrusion detection system (HIDS) is developed using text mining, with a focus on constructing a dataset of malicious URLs and employing the DOC2VEC model for feature representation. Machine learning algorithms, particularly the multilayer perceptron, are applied to enhance detection accuracy for SQL injection (SQLi), cross-site scripting (XSS), and directory traversal attacks. The multilayer perceptron achieves the highest accuracy of 90.67%. Furthermore, a new security intelligent system (SIS-ID) is introduced to detect both malicious URLs and distributed denial of service (DDOS) attacks, achieving high accuracies of 98.52% for malicious URLs and 77.04% for DDOS attacks. These accuracies are based on the voting and stacking models, respectively, derived from machine learning techniques and optimization methods. Validation of SIS-ID is conducted using hardware-based real-time simulation, demonstrating effectiveness in mitigating denial of service attacks.

Manjunatha and Kempanna (2022) present an Ensemble Method for classifying Structure Query Language (SQL) injection vulnerabilities. It utilizes a benchmark dataset consisting of 33,758 records that include different types of SQL and XML injection attacks. Following artifact removal during preprocessing, Natural Language Processing techniques are utilized for feature engineering, encompassing the extraction of six distinct types of features: TF-IDF, Word-to-Vector, SkipGram, Count Vectorizer, Glove, and Continuous Bag of Words. Imbalanced data is mitigated via sampling strategies, while the optimal features are chosen by four validation procedures: Significant Test, PCA (Principal Component Analysis), Variance Threshold, and Sbest. The Ensemble Model comprises two stages: Stage 2 identifies vulnerability in subdomains and domains using the URLs provided by the user. On the other hand, Stage 1 utilizes nine distinct machine learning models, namely Multinomial, Gaussian, Bernoulli Naive Bayes, Logistic Regression, Decision Tree, Random Forest, AdaBoost, and SVC with poly, rbf, and linear kernel. These algorithms, which have been trained using other vectors like Google News and GloVe, are designed to identify new queries and determine whether they include any vulnerabilities. The proposed ensemble technique achieves a 99% accuracy.

Kaur et al. (2023) provided an extensive examination of modern machine learning and neural network methods used to identify cross-site scripting (XSS) attacks. The study explores a range of methods such as deep neural networks, decision trees, and web-log-based models, while also pinpointing important areas of research that are essential for improving detection models. The paper discusses the challenges faced in establishing these algorithms and presents potential directions to improve XSS attack detection methodologies, providing valuable insights for future research efforts.

Shahrivar (2022) investigates the utilization of machine learning methods to identify vulnerability scanning attacks in web applications. Real-world data from tCell, a web application firewall, is employed for this purpose. The security vulnerabilities were mitigated by training twenty-four models, which achieved high precision and recall rates ranging from 91% to 96% and 85% to 93%, respectively. These models were able to identify and respond to the automated attacks effectively. Nevertheless, the models displayed insufficient calibration, leading to forecasts that lacked confidence. However, the results demonstrate progress compared to previous online strategies that relied on thresholds. They define a performance standard and emphasize the need for additional research and development to enhance and adjust the models accurately.

Bhardwaj et al. (2022) focused on employing machine learning to tackle different security issues such as intrusion detection, cross-site scripting (XSS), SQL injection (SQLI), and phishing. The study utilizes various machine learning techniques to identify certain types of attacks. Convolutional Neural Network (CNN) is used for detecting Cross-Site Scripting (XSS) attacks, Logistic Regression is employed for identifying SQL Injection (SQLI) attacks, and Support Vector Machine (SVM) is utilized for detecting phishing attacks. Additionally, Decision Tree Classifier (DTC), Bernoulli Naive Bayes (BNB), and K-Nearest Neighbors (KNN) are employed for intrusion detection. The results indicate high levels of accuracy, with the Convolutional Neural Network (CNN) earning a rate of 98.59% for detecting Cross-Site Scripting (XSS) attacks. Logistic Regression yielded a rate of 92.85% for detecting SQL Injection (SQLI) attacks, while Support Vector Machines (SVM) reached 85.62% accuracy for detecting phishing attempts. Decision Tree Classifier (DTC), Bernoulli Naive Bayes (BNB), and K-Nearest Neighbors (KNN) achieved accuracy rates of 99.47%, 90.67%, and 99.16% correspondingly for intrusion detection.

Hassan et al. (2021) focus on the significant threat of SQL injection (SQLi) in online applications that rely on databases. SQLi can result in unauthorized access to sensitive data and administrative privileges. Therefore, it is crucial to develop effective detection systems to minimize financial losses for enterprises. The paper presents a new approach for detecting using deep learning, which utilizes correlation and chi-squared methods for ranking features and a feed-forward network for selecting features and detecting them. After conducting thorough testing on more than 1,850 datasets, the proposed method has shown exceptional efficiency, with an accuracy rate of 98.04%. This surpasses the performance of existing machine learning solutions.

Sharma et al. (2020) examined the effectiveness of machine learning methods in identifying web-based attacks, with a focus on the difficulty of dealing with false positives and false negatives. The project aims to improve the accuracy of detection by refining the process of extracting features from the CSIC 2010 HTTP dataset, which simulates e-commerce web traffic. The experimental results demonstrate enhanced performance across a range of machine learning algorithms when utilizing the suggested refined feature set. The evaluation metrics, including Precision, Recall, Accuracy, and F-measure, demonstrate the J48 decision tree algorithm's superiority in obtaining a high True Positive rate, Precision, and Recall. This suggests that the algorithm can reliably detect and prevent web-based attacks.

Oudah et al. (2022) investigated the utilization of machine learning in identifying SQL injection attacks, utilizing four separate machine learning models for this objective. This study examines the influence of data preparation and feature extraction on the accuracy of detection. It uses a training dataset created from real user requests and a set of both harmless and harmful SQL queries. The comparative examination of the models indicates that the Support Vector Model achieves the best accuracy of .997, closely followed by Extreme Gradient Boosting at .995. It is worth mentioning that Naïve Bayes with N-gram level feature extraction is the quickest model, taking only 6 milliseconds for classifier training.

Montes et al. (2021) Investigated the utilization of deep learning methods to improve the effectiveness of web application firewalls (WAFs) in identifying and thwarting cyber-attacks. Historically, Web Application Firewalls (WAFs) have been employed to scrutinize HTTP requests exchanged between clients and servers to detect and prevent possible security risks. The topic is tackled by the authors using a one-class supervised learning framework. They employ a deep language model with a transformer encoder architecture, which is renowned for its self-attention processes. By utilizing pre-trained language models for transfer learning, they extract features from HTTP requests and turn them into feature vectors for one-class categorization. In addition, they provide a performance indicator to automatically determine an operating point in the one-class model. The experimental findings show that the performance is better than standard rule-based MODSECURITY settings, like vanilla OWASP CRS, without requiring a security professional to manually define features.

METHODOLOGY

Incoming Network Packets: The dataset comprises attacks that were carried out on network logs. There are 12 features in total. The action feature is used as a class. There are 4 classes in total, which include allow, action, drop, and reset classes. The features of the dataset are Source Port, Destination Port, NAT Source Port, NAT Destination Port, Action, Bytes, Bytes Sent, Bytes Received, Packets, Elapsed Time (sec), pkts_sent, pkts_received (Figure 1).

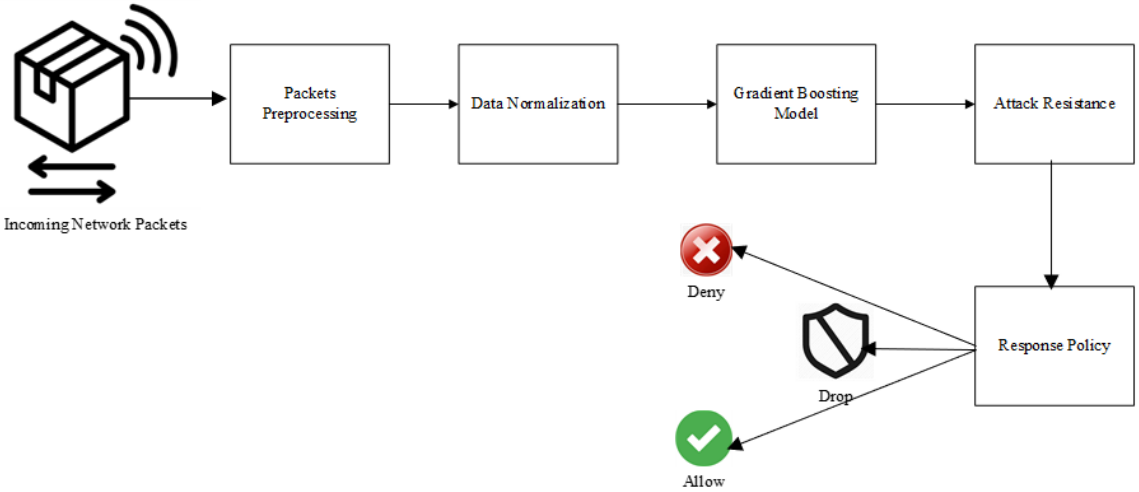


Figure 1. Architectural Design

Packets Preprocessing: In the preprocessing of the dataset, applied Min-Max scaler for feature scaling. This technique transforms the features in such a way that they all fall within a specified range, typically [0, 1]. This is achieved by subtracting the minimum value of each feature and then dividing it by the difference between the maximum and minimum values. Specifically, the features in the dataset include Source Port, Destination Port, NAT Source Port, NAT Destination Port, Action, Bytes, Bytes Sent, Bytes Received, Packets, Elapsed Time (sec), pkts_sent, and pkts_received. Each of these features will undergo the Min-Max scaling process independently. This means that for each feature, the minimum value will be set to 0, and the maximum value will be set to 1 after applying the scaling formula. The mathematical expression of MinMaxScaler can be seen in Equation 1

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad \textit{Equation 1}$$

where:

x is the original value of the feature.
min(*x*) is the minimum value of the feature in the dataset.
max(*x*) is the maximum value of the feature in the dataset.
x′ is the scaled value of the feature.

Gradient Boosting Model: Gradient Boosting is an ensemble learning technique used for both regression and classification problems. It builds an additive model in a forward stage-wise manner. In the case of classification tasks like detecting attacks on network logs, it is known as a Gradient gradient-boosting classifier. The mathematical expression can be seen in Equation 2.

$$F(x) = \sum_{m=1}^{M} y_m h_m(x) \qquad \textit{Equation 2}$$

Where:

$F(x)$ represents the final ensemble prediction.
M is the total number of base learners (individual decision trees in the case of gradient boosting).
γ_m is the weight (shrinkage) assigned to the m-th base learner.
$h_m(x)$ is the m-th base learner's prediction for input x.

In the context of detecting attacks on network logs, each $h_{m(}x)$ could represent the prediction of an individual decision tree that is focused on classifying logs as either normal or attack, and γ_m would be the weight assigned to each tree's prediction. The final prediction $F(x)$ is obtained by aggregating the predictions of all the individual trees.

Attack Resistant: Gradient boosting, represents a formidable defense against firewall breaches in network security. By harnessing the power of ensemble learning, this approach combines the strengths of multiple weak learners, creating a robust, adaptive shield against sophisticated intrusion attempts. The utilization of gradient boosting enables the model to learn and adapt to evolving attack patterns, continuously refining its ability to discern genuine network activity from malicious intent.

Response Policy: This has to do with the action taken when the network packets come into the network system. The response policy can be either allowed, denied, or dropped.

RESULTS

An experiment was set up on Google Colab. The experimental phase consists of the exploratory data analysis phase and the implementation of a machine learning model for firewall defense and their response policy towards resisting attacks on network logs.

Presentations of Results

Some tools such as pandas, seaborn, and matplotlib library were used in analyzing the firewall dataset. The analysis phase gave us a proper insight into the dataset before training the ML model for firewall defense on network logs. First, the dataset features were checked if correlated. Then, the seaborn library was used in performing a correlation between the dataset features. The correlated matrix is shown in Figure 2. Second, a bar chart is plotted to check if the number of classes (different types of response policies) have the same number of instances. The bar chart is shown in Figure 3 and reveals that the number of instances of each of the different types of response policies is not equal. That simply makes the dataset imbalance, this simply means that if the data imbalance is not solved, the ML classifier will produce a high rate of false positives and negatives. To solve the data imbalance problem, random over-sampling was performed using an over-sampling technique called RandomOverSampler. The result of the sampling is used to populate the dataset, making all the classes have an equal number of instances. The populated data is shown in Figure 4.

Finally, important features from the dataset were extracted using the Random Forest Classifier (RF). The RF classifier was used in ranking the features of the dataset. Table 1 shows the extracted features (the most important features), and Figure 5, shows the visualized plot of the important features.

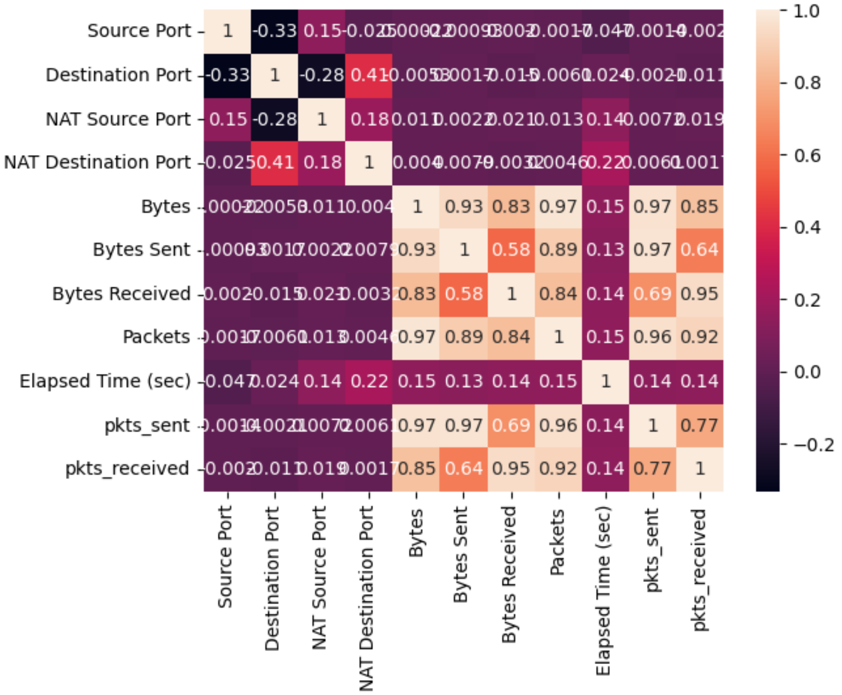


Figure 2. Correlated Matrix

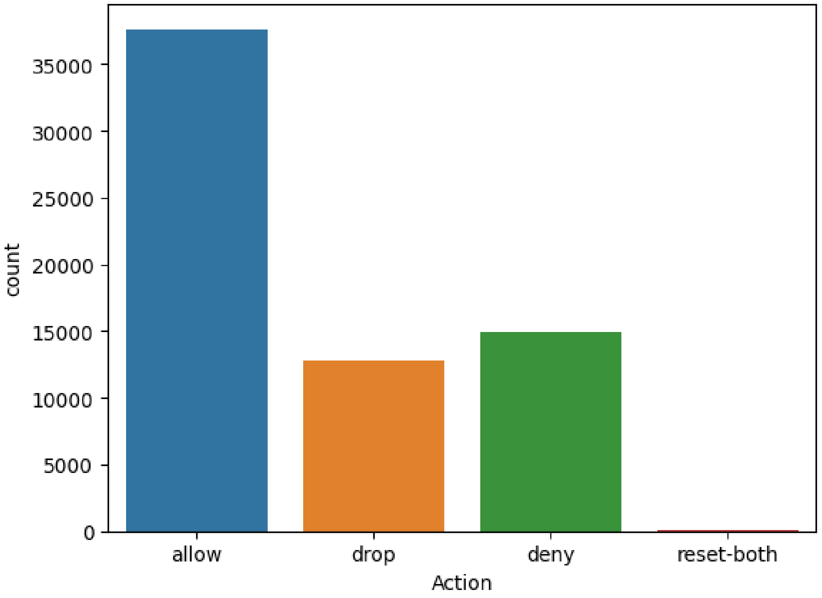


Figure 3. Bar chart of the imbalanced classes.

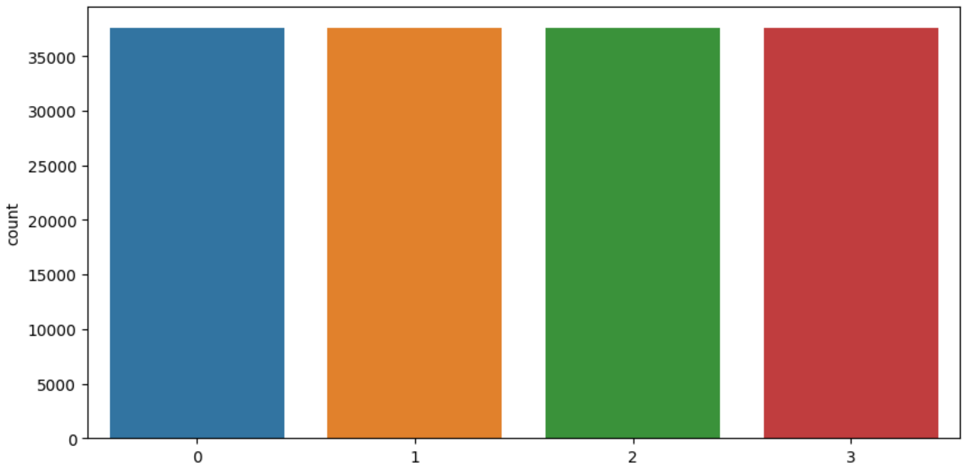
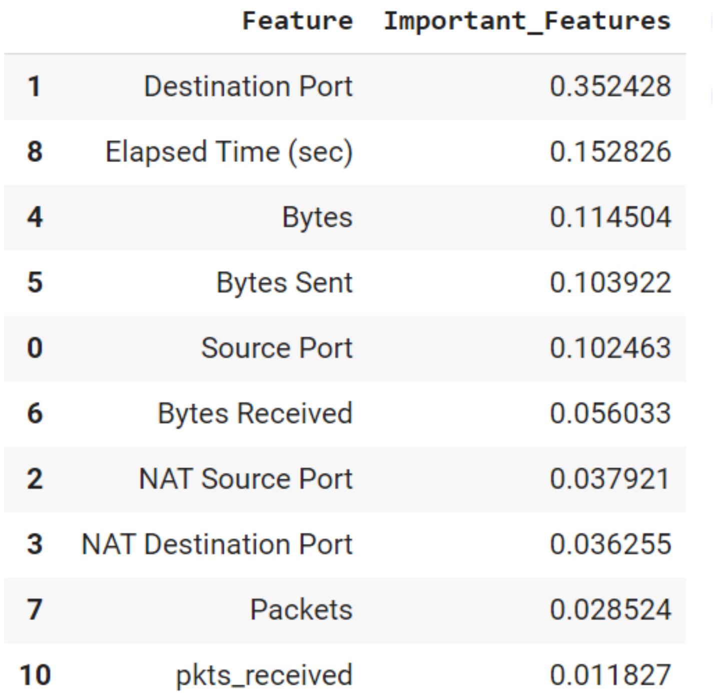


Figure 4. Bar chat of the balanced classes.

Table 1. Extracted Features



	Feature	Important_Features
1	Destination Port	0.352428
8	Elapsed Time (sec)	0.152826
4	Bytes	0.114504
5	Bytes Sent	0.103922
0	Source Port	0.102463
6	Bytes Received	0.056033
2	NAT Source Port	0.037921
3	NAT Destination Port	0.036255
7	Packets	0.028524
10	pkts_received	0.011827

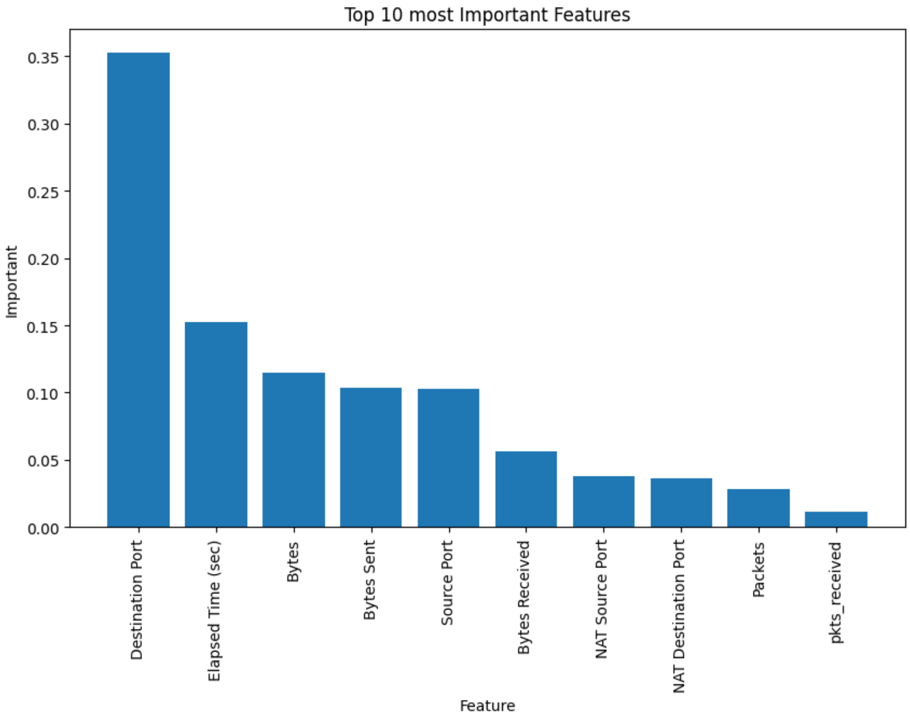
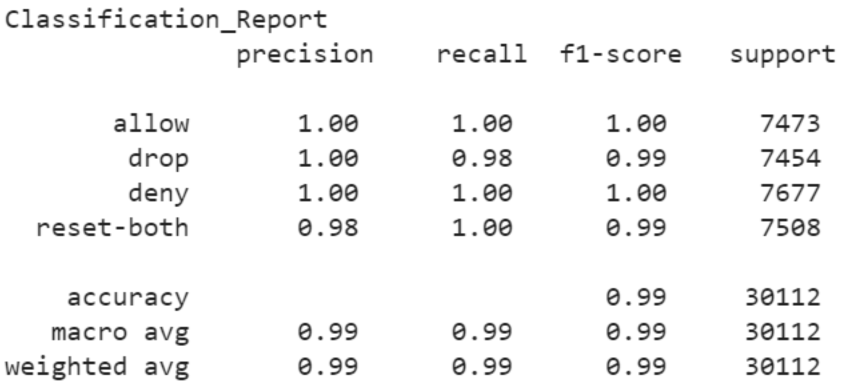


Figure 5. Bar Chart of Top 10 Important Features

Model Training with Gradient Boost Classifier (GBC)

The GBC model on the firewall dataset. We fine-tuned the hyperparameters. The fine-tuned parameters are n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42. The GBC model was used to predict an unseen dataset to detect the various types of DDOS attacks. The result of the GBC model was also evaluated using matrix evaluation (Classification matrix, and Confusion matrix). The result of the GBC model is shown in Figure 6 and Figure 7. The GBC model had an accuracy of 99% on the test data.



```
Classification_Report
              precision    recall  f1-score   support

       allow       1.00      1.00      1.00      7473
        drop       1.00      0.98      0.99      7454
        deny       1.00      1.00      1.00      7677
  reset-both       0.98      1.00      0.99      7508

    accuracy                           0.99     30112
   macro avg       0.99      0.99      0.99     30112
weighted avg       0.99      0.99      0.99     30112
```

Figure 6. Classification Report GBC model.

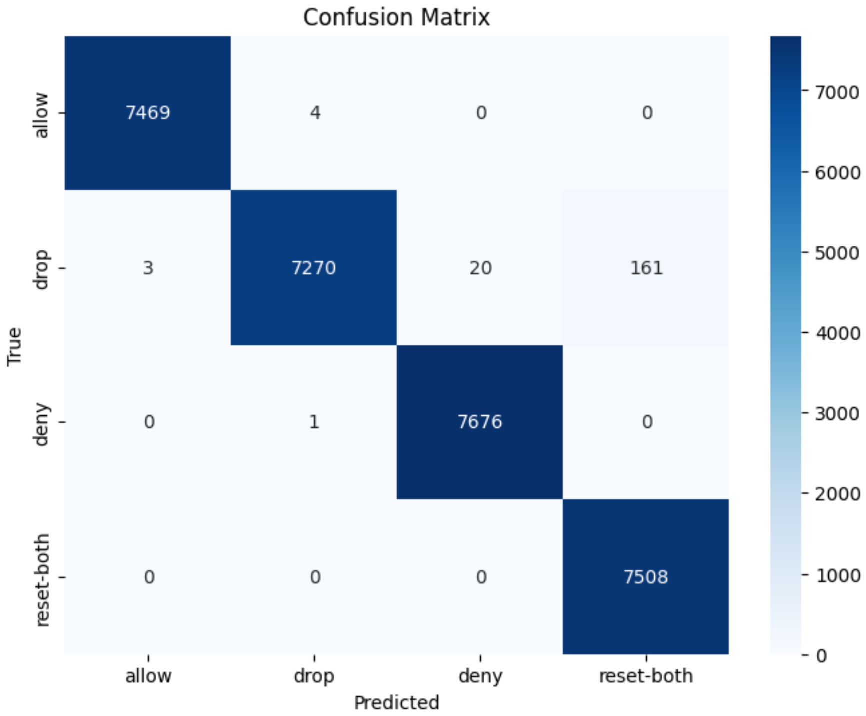


Figure 7. Confusion Matrix of the GBC Model.

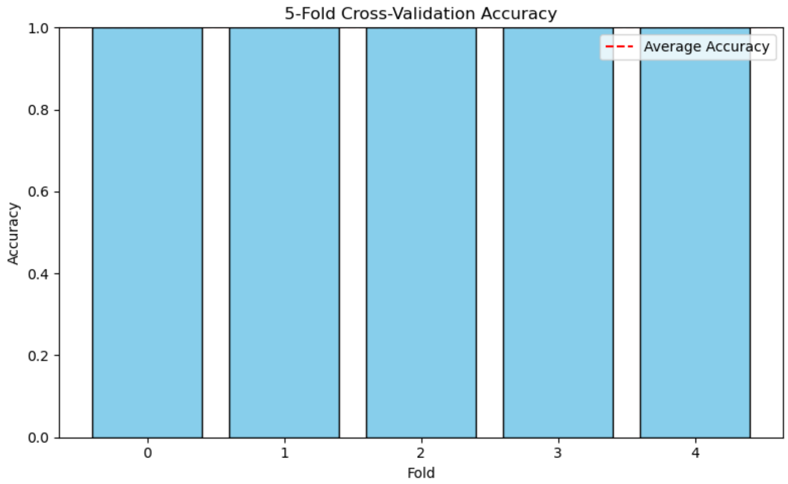


Figure 8. Five-Fold Cross Validation Test

DISCUSSION

The analysis of the firewall dataset yielded significant findings on its attributes and class distribution, highlighting an imbalance that may potentially affect the efficacy of machine learning classifiers. By utilizing methods like correlation analysis and oversampling, these problems were resolved, guaranteeing a well-balanced dataset for effective model training. The assessment of feature importance enabled the discovery of relevant features, which then guided subsequent modeling efforts. The Gradient Boost Classifier (GBC) exhibited outstanding performance, achieving a 99% accuracy rate on the test data, as confirmed by the evaluations of the Classification Report and Confusion Matrix. The results highlight the effectiveness of the model in accurately identifying different response policies and probable Distributed Denial of Service (DDoS) assaults. The model's reliability and generalization capabilities were further proven by cross-validation, which validates its appropriateness for real-world cybersecurity applications. Ongoing improvement and verification of the model's performance will be crucial for its implementation and efficacy in protecting against network threats.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

High Accuracy in Threat Detection: The proposed Firewall Defense and Response Policy, leveraging a Gradient Boost Classifier, achieves a remarkable accuracy of 99.99% in detecting cyber threats. This high level of precision underscores the effectiveness of using advanced machine-learning techniques for network security.

Comprehensive Analysis of Network Logs: The integration of in-depth network log analysis with machine learning allows the model to identify subtle patterns indicative of malicious activities. This enhances the capability to detect both common and sophisticated intrusion attempts.

Adaptive Response Policy: The dynamic adjustment of countermeasures based on the severity and nature of detected anomalies ensures a proactive and flexible defense strategy. This adaptability is crucial in responding to the ever-evolving landscape of cyber threats.

Robust and Resilient Framework: The proposed methodology demonstrates superior performance in mitigating a wide range of attacks, including evasive tactics. This robustness and resilience make it a valuable framework for enhancing network security.

Recommendations

Implementation in Diverse Environments: Organizations should consider implementing the proposed Firewall Defense and Response Policy in various operational environments to benefit from its high accuracy and adaptability in detecting and mitigating cyber threats.

Continuous Dataset Expansion: Regularly updating and expanding the training dataset with new and diverse attack vectors will help maintain the model's effectiveness in identifying emerging threats.

Periodic Model Re-evaluation: Regularly re-evaluating and fine-tuning the machine learning model is essential to ensure it remains effective against the latest cyber threat tactics and strategies.

Integration with Existing Security Systems: Integrating this defense policy with existing security infrastructure can provide a comprehensive security solution, enhancing overall protection and response capabilities.

User Training and Awareness: Training network administrators and security personnel on the functionalities and benefits of the proposed system can optimize its implementation and efficacy. Awareness programs about the latest cyber threats can also complement technical defenses.

Research and Development: Ongoing research into advanced machine learning techniques and their application in cybersecurity should be encouraged to further improve the accuracy and resilience of threat detection systems.

ACKNOWLEDGEMENT

None to mention.

FUNDING

The author did not receive funding from any institutions.

DECLARATIONS

Conflict of Interest

The researcher declares no conflict of interest in this study.

Informed Consent

Informed consent was obtained from all participants involved in this study, ensuring they were fully aware of the research objectives, procedures, potential risks, and benefits. Participants were allowed to ask questions and withdraw at any time without penalty.

Ethics Approval

This study received ethics approval from the Institutional Review Board (IRB) of Rivers State University, ensuring that all research activities were conducted by ethical standards and guidelines for human subject research.

REFERENCES

Appelt, D., Nguyen, C. D., Panichella, A., & Briand, L. C. (2018). A machine-learning-driven evolutionary approach for testing web application firewalls. *IEEE Transactions on Reliability, 67*(3), 733-757.

Applebaum, S., Gaber, T., & Ahmed, A. (2021). Signature-based and machine-learning-based web application firewalls: A short survey. *Procedia Computer Science, 189*, 359-367.

Bhardwaj, A., Chandok, S. S., Bagnawar, A., Mishra, S., & Uplaonkar, D. (2022). Detection of cyber attacks: XSS, sqli, phishing attacks, and detecting intrusion using machine learning algorithms. In *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)* (pp. 1-6). IEEE.

Clincy, V., & Shahriar, H. (2018). Web application firewall: Network security models and configuration. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 835-836). IEEE.

Hassan, M. M., Ahmad, R. B., & Ghosh, T. (2021). SQL injection vulnerability detection using deep learning: a feature-based approach. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 9*(3), 702-718.

Ito, M., & Iyatomi, H. (2018). Web application firewall using character-level convolutional neural network. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 103-106). IEEE.

Kaur, J., Garg, U., & Bathla, G. (2023). Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review. *Artificial Intelligence Review, 56*(11), 12725-12769.

Krishnan, M., Lim, Y., Perumal, S., & Palanisamy, G. (2022). Detection and defending the XSS attack using novel hybrid stacking ensemble learning-based DNN approach. *Digital Communications and Networks, in press*. Retrieved from https://www.sciencedirect.com/science/article/pii/S2352864822001997

Manjunatha, K. M., & Kempanna, M. (2022). Count vectorizer model-based web application vulnerability detection using artificial intelligence approach. *Journal of Discrete Mathematical Sciences and Cryptography, 25*(7), 2039-2048.

Montes, N., Betarte, G., Martínez, R., & Pardo, A. (2021). Web application attacks detection using deep learning. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 25th Iberoamerican Congress, CIARP 2021*, Porto, Portugal, May 10–13, 2021, Revised Selected Papers 25 (pp. 227-236). Springer International Publishing.

Moradi A., Teshnehlab, M., & Sedighian Kashi, S. (2019). Leveraging deep neural networks for anomaly-based web application firewall. *IET Information Security, 13*(4), 352-361.

Oudah, M. A., Marhusin, M. F., & Narzullaev, A. (2022). SQL injection detection using machine learning with different TF-IDF feature extraction approaches. In *International Conference on Information Systems and Intelligent Applications* (pp. 707-720).

Prabakaran, S., Ramar, R., Hussain, I., Kavin, B. P., Alshamrani, S. S., AlGhamdi, A. S., & Alshehri, A. (2022). Predicting attack pattern via machine learning by exploiting stateful firewall as virtual network function in an SDN network. *Sensors, 22*(3), 709.

Rajesh, Shriram and Clement, Marvin and S. B., Sooraj and S. H., Al Shifan and Johnson, Jyothi, Real-Time DDoS Attack Detection Based on Machine Learning Algorithms (September 27, 2021). Proceedings of the Yukthi 2021- The International Conference on Emerging Trends in Engineering – GEC Kozhikode, Kerala, India, Available at SSRN: https://ssrn.com/abstract=3974241 or http://dx.doi.org/10.2139/ssrn.3974241

Shaheed, A., & Kurdy, M. B. (2022). Web application firewall using machine learning and features engineering. *Security and Communication Networks*, *2022*(1), 5280158.

Shahrivar, P. (2022). *Detection of Vulnerability Scanning Attacks using Machine Learning: Application Layer Intrusion Detection and Prevention by Combining Machine Learning and AppSensor Concepts*. Retrieved from https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1714133&dswid=-5695

Sharma, S., Zavarsky, P., & Butakov, S. (2020). Machine learning-based intrusion detection system for web-based attacks. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)* (pp. 227-230). IEEE.

Taylor, O. E., & Ezekiel, P. S. (2022). A Robust System for Detecting and Preventing Payloads Attacks on Web-Applications Using Recurrent Neural Network (RNN). *European Journal of Computer Science and Information Technology, 10*(4), 1-13.

Thang, N. M. (2020). Improving efficiency of web application firewall to detect code injection attacks with random forest method and analysis attributes http request. *Programming and Computer Software, 46*, 351-361.

Author's Biography

Onate E. Taylor obtained his B.Sc, M.Sc, and Ph.D degrees all in Computer Science from the Rivers State University of Science and Technology, University of Ibadan, and University of Port Harcourt, Nigeria respectively. He is currently an Associate Professor in the Department of Computer Science, Rivers State University, Port Harcourt, Nigeria. He is a chartered member of the Computer Professionals (Registration Council) of Nigeria and Nigeria Computer Society. His research focuses on machine intelligent systems, context-aware systems, and pervasive systems. He has over sixty academic publications and more than fifteen years of teaching and research experience.

Promise S. Ezekiel is an AI developer with a BSc and MSc in Computer Science from Rivers State University, I have over twenty publications online, showcasing my expertise in machine learning, deep learning, and computer vision. Specializing in cybersecurity, smart systems, and blockchain technology, my research aims to advance the fields through innovative solutions that enhance security, efficiency, and transparency. My passion lies in pushing the boundaries of technology to create a smarter, more secure, and sustainable future for all.