American Journal of Computer Science and Technology
2023; 6(4): 170-179
http://www.sciencepublishinggroup.com/j/ajcst
doi: 10.11648/j.ajcst.20230604.14
ISSN: 2640-0111 (Print); ISSN: 2640-012X (Online)

Improving Internet Firewall Using Machine Learning Techniques

Martha Ozohu Musa[1, *], Temitope Victor-Ime[2]

[1]Department of Cyber Security, University of Port Harcourt, Port Harcourt, Nigeria

[2]Department of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria

Email address:

martha.musa@uniport.edu.ng (Martha Ozohu Musa)

[*]Corresponding author

To cite this article:

Martha Ozohu Musa, Temitope Victor-Ime. (2023). Improving Internet Firewall Using Machine Learning Techniques. *American Journal of Computer Science and Technology*, *6*(4), 170-179. https://doi.org/10.11648/j.ajcst.20230604.14

Received: November 4, 2023; **Accepted**: November 21, 2023; **Published**: November 29, 2023

Abstract: Internet firewalls are a composite of both hardware and software components, which are employed to enforce a security policy dictating the movement of data between many networks. Conventional firewalls depend on pre-established rules and signatures in order to identify and prevent the transmission of harmful network traffic. Nevertheless, it is worth noting that the aforementioned regulations and authentication methods frequently remain unchanging and can be effortlessly circumvented by highly skilled assailants. This analysis improves the use of firewall in detecting internet attacks using machine learning techniques. This study introduces a novel approach to enhance internet firewall efficacy through the integration of machine learning techniques. By leveraging a sophisticated model, the proposed system achieves exceptional performance, attaining a remarkable 99.99% precision, recall, and F1-score. This significant advancement in accuracy demonstrates the potential of employing machine learning in fortifying internet security infrastructure. The model's ability to consistently and reliably discern malicious activities from benign traffic showcases its robustness in real-world scenarios, thus presenting a promising avenue for bolstering network defense mechanisms. This research not only contributes to the burgeoning field of cybersecurity but also lays the foundation for future innovations in adaptive and intelligent firewall systems.

Keywords: Firewall, Machine Learning, Cyber-Attacks, Response Policy

1. Introduction

The increasing complexity and sophistication of cyber threats have necessitated the development of more advanced and effective security measures. One area of focus in enhancing cybersecurity is the improvement of internet firewalls. Firewalls act as the first line of defense against unauthorized access and malicious activities by monitoring and controlling network traffic. Traditional firewall systems rely on predefined rules and signatures, which may not be sufficient to detect and prevent emerging threats. However, recent advancements in machine learning offer promising opportunities to enhance the capabilities of firewalls and improve their effectiveness [1].

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. By analyzing large amounts of data and identifying patterns, machine learning algorithms can detect anomalies and classify network traffic more accurately than traditional rule-based approaches [2]. This capability makes machine learning a valuable tool for improving internet firewalls.

One approach to enhancing internet firewalls using machine learning is through the development of intelligent classification models. These models analyze packet attributes and use machine learning algorithms, such as shallow neural networks and optimizable decision trees, to determine the appropriate action for each communicated packet [3]. By leveraging machine learning techniques, these models can adapt and learn from new data, allowing them to detect and respond to previously unseen threats.

Another area of research focuses on the integration of machine learning algorithms into intrusion detection systems (IDS) to improve the performance of firewalls. IDSs are

designed to detect and respond to malicious activities within a network. By using supervised, semi-supervised, and unsupervised machine learning algorithms, IDSs can become more intelligent and effective in identifying and mitigating threats [4]. Machine learning techniques can help IDSs analyze network traffic in real-time, identify patterns of malicious behavior, and make accurate decisions on whether to allow or block certain traffic.

Furthermore, machine learning can be combined with data mining techniques to analyze firewall policies and identify potential vulnerabilities. By analyzing historical firewall data, machine learning algorithms can uncover hidden patterns and correlations that may indicate weaknesses in the firewall configuration [5]. This information can then be used to optimize firewall policies and improve their overall effectiveness.

2. Review of Related Works

In their study, researchers [6] conducted a statistical analysis to differentiate between metrics and features in HTTP traffic and attack traffic that indicate the presence of an attack and those that do not. The authors conducted a comparative analysis of attack and normal traffic by examining and evaluating the distinct characteristics included in the widely used datasets ISCX, CISC, and CICDDoS. A layered architectural model was constructed utilising a dataset obtained from a simulation environment in order to identify and mitigate Distributed Denial of Service (DDoS), Cross-Site Scripting (XSS), and Structured Query Language (SQL) injection assaults. The DDoS detection model, which was supposed to have an accuracy of 97.57 percent, was incorporated into the initial layer of the layered architecture based on LSTM. The second layer of the architecture, responsible for detecting XSS and SQL injection, obtained an accuracy of 89.34 percent. The first focus was on analysing HTTP traffic due to its generally higher speed, whereby it was subjected to scrutiny, filtered, and thereafter forwarded to the subsequent layer. The web application firewall (WAF) enhances the capabilities of a conventional network firewall by the implementation of application-level filtering.

Dawadi *et al.* [7] provided a concise overview of the evolutionary advancements of Web Application Firewalls (WAFs) facilitated by the integration of machine learning methodologies. The pros and downsides of the subject under analysis are examined, and any unresolved questions are identified. The evaluation assesses the efficacy of security measures in safeguarding against zero-day attacks, while also considering their ease of implementation and maintenance. The superiority of machine-learning-based methods over signature/rule-based methods has been ascertained due to their ability to effectively address the vulnerability of zero-day threats, while also offering the advantage of being relatively simpler to configure and sustain. The survey additionally revealed that further investigation is warranted regarding the effectiveness of machine-learning-based web application firewalls (WAFs) in safeguarding against contemporary attack patterns targeting web application frameworks.

The authors in reference [8] created a comprehensive framework comprising a series of laws and regulations aimed at limiting access to networks that may pose potential harm. These safeguards are inadequate in mitigating the risk of attacks that exploit a significant quantity of distinct socket identifiers. Machine learning algorithms are trained using traditional network threat intelligence data to identify potential harmful links and probable targets of attacks within a network. The Decision Table (DT), Bayesian Network (BayesNet), Naive-Bayes, C4.5, and DT algorithms are employed for the purpose of predicting the specific target host that is likely to be subjected to an attack, utilising conventional network data. According to the findings of the studies, the Bayesian Network method has the highest average prediction accuracy (92.87 percent), followed by the Native-Bayes method (87.81 percent), the C4.5 Algorithm (84.92%), and the Decision Tree Algorithm (83.18%). A comprehensive dataset obtained from nine honeypot servers documented a total of 451,000 login attempts originating from 178 distinct countries. These attempts were traced back to about 70,000 unique IP addresses and 41,000 unique source ports.

Prabakaran, Senthil *et al.* [9] present ML-Driven, a novel approach that utilises machine learning and evolutionary algorithms to identify vulnerabilities in Web Application Firewalls (WAFs) that can be exploited by SQL injection attacks. Initially, ML-Driven would produce a diverse range of assaults and transmit them to the system that is being protected by the target Web Application Firewall (WAF). Subsequently, the ML-driven algorithm selects attacks that exhibit discernible patterns or substrings associated with circumventing the Web Application Firewall (WAF), and further enhances them to generate novel bypass attacks. The acquisition of attack patterns occurs gradually through the utilisation of machine learning techniques, wherein previously created attacks are employed to train the system. This training process involves evaluating the efficacy of the attacks by determining if the Web Application Firewall (WAF) successfully blocks them or whether they manage to overcome its defences. The researchers included a machine learning-driven approach into a software tool and conducted a comparative evaluation using ModSecurity, a widely used open-source Web Application Firewall (WAF), as well as a proprietary WAF deployed by a financial institution. The experimental findings demonstrate that ML-Driven techniques exhibit proficiency in generating SQL injection attacks that are resistant to Web Application Firewalls (WAFs) and in identifying attack patterns.

Appelt, D *et al*. [10] proposed for the implementation of a web application firewall. The framework integrates machine learning techniques with features engineering procedures to effectively detect and counteract online-based threats. The model performs an analysis on incoming HTTP requests, wherein it extracts four distinct elements, namely the URL, payload, and headers. Subsequently, it classifies each request

as either normal or abnormal, employing specified criteria for this classification process. The model incorporates five distinct features, including request duration, allowed character ratio, special character %, and attack weight. The model underwent evaluation using recently updated datasets and was subjected to four distinct classification algorithms. Additionally, two techniques were employed to address the issue of overfitting. A normal request is characterised by its brevity, a high ratio of authorised characters, a low ratio of special characters, and a lack of attack weight. There is a notable rise in the length of anomaly requests, a drop in the allowable percentage of characters, an increase in the percentage of special characters, and an increase in the weight of numerical attacks. The model demonstrated a classification accuracy of 99.6% on widely utilised research datasets and 98.8% on actual web server datasets.

Shaheed, Aref, and M. H. D. Kurdy [11] introduced combined deep neural networks for feature learning with isolation forests for classification. In the CSIC 2010 data set, the authors conducted a comparative analysis between their proposed method and alternative approaches that did not include feature extraction models. The deep neural network that was proposed also derived advantages from a diverse range of learning and activation functions. The findings indicate that deep models exhibit higher levels of accuracy compared to techniques that lack distinct features.

In the study conducted by [12], a Recurrent Neural Network (RNN) model was trained using a dataset consisting of various web application attack types, including XSS, SQLi, and Shell. In preparation for preprocessing, a Random Over Sampling approach was employed to address the issue of a highly imbalanced dataset. Following the resolution of the imbalanced problem, the dataset underwent pre-processing procedures, encompassing data cleaning and tokenization. In order to train our recurrent neural network (RNN) model, we transformed the tokenized input into an array format. The accuracy and loss numbers for both training and testing data are presented for each epoch, with our proposed model being trained for a total of two (2) epochs. Following the completion of training, the suggested recurrent neural network (RNN) model demonstrated a remarkable accuracy of 99.96% on the testing dataset, while achieving a little lower accuracy of 99.91% on the training dataset. In addition, a Python Flask framework was employed to deploy our Recurrent Neural Network (RNN) model over the internet. This implementation established a robust infrastructure for the surveillance and mitigation of diverse payload threats against web-based software. This paper centres around the examination of web application attacks.

The study conducted by [13] examined several distinguishing features that can be used to identify between regular network traffic and Distributed Denial of Service (DDoS) attack traffic. These features encompassed various types of assaults, including UDP flood attacks, ICMP ping flood attacks, TCP SYN flood attacks, and land attacks. The authors conducted a comparative analysis of various machine learning algorithms, including K-nearest neighbour, decision tree, random forest, and naive Bayes.

The authors of [14] developed a novel model called 3C-LSTM, which integrates LSTM with CNN. They asserted that this model exhibited superior performance compared to other existing models. The authors utilised a methodology wherein words were converted into vectors in order to train the proposed model, which was subsequently employed for the purpose of detecting cross-site scripting (XSS). This study involved a comparison of the model's performance across various batch sizes, with the aim of identifying an optimal value.

In a previous study, the authors proposed the inclusion of the noise coefficient in the DA-SANA framework as a means to enhance the detection of attack traffic [15]. The study and comparisons were performed by the author utilising three datasets, namely CISC, PKDD, and a dataset that was constructed. The researchers of this study conducted an extensive analysis of many types of threats, including SQL injection, cross-site scripting, remote code execution, cross-site request forgery, and cross-site extension.

3. Methodology

Dataset: This section describes the dataset information and the objectives of the analysis. The dataset used in this analysis is internet firewall dataset. The dataset was downloaded from Kaggle.com. The dataset can consist of a total of 12 features. The action feature is implemented as a class. There exists a total of four classes. The aforementioned classes encompass allow, action, drop, and reset. The dataset comprises several attributes, including 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.

Data Preprocessing: The preprocessing of the dataset involves checking if there exist null or duplicate values. From the experiment conducted, no missing or duplicate values was found.

Feature Extraction: This has to do with extracting relevant features in the dataset. Random Forest classifier was used in perform a ranking on the dataset. This is to enable see the important features that we are to use.

The Random Forest Classifier: This is a type of ensemble learning technique that involves the combination of many decision trees in order to generate predictions [16]. The aforementioned statistical classifier has garnered significant usage across multiple academic fields. According to [17] empirical evidence suggests that Random Forests exhibit superior performance in classification accuracy compared to other classifiers such as support vector machines (SVMs) and k-nearest neighbours (KNNs). The classifier is trained with a bagging technique, in which each decision tree is trained on a randomly selected subset of the training data [18]. The ultimate forecast is derived by consolidating the forecasts generated by each individual decision tree [19].

The Decision Tree: This is a machine learning technique that is commonly employed for classification and regression

applications. It is a straightforward yet effective method. The model can be described as a hierarchical structure like a tree. In this structure, interior nodes correspond to features or attributes, branches reflect decision rules, and leaf nodes provide class labels or numerical values [20]. Decision Trees are renowned for their high level of interpretability and explainability, as they possess the ability to present decision rules in a clear and easily comprehensible manner, which can also be visually represented. Nevertheless, it is worth noting that Decision Trees may encounter challenges such as overfitting and instability, as highlighted by [21] Base classifiers, such as those employed in ensemble approaches like Random Forests [22] are frequently utilised.

The K Nearest Neighbour (KNN): This algorithm is a popular machine learning technique used for classification and regression tasks. The K Nearest Neighbour (KNN) method is a classification technique that is non-parametric in nature. It operates by making predictions based on the collective decision of the k nearest neighbours to a certain data point [23]. The K-nearest neighbours (KNN) approach is classified as a lazy learning algorithm, as it does not construct a model directly in the training phase. In contrast, the system retains the training data and utilises it to generate predictions during runtime [2]. The K-nearest neighbours (KNN) algorithm is renowned for its straightforwardness and straightforwardness of execution.

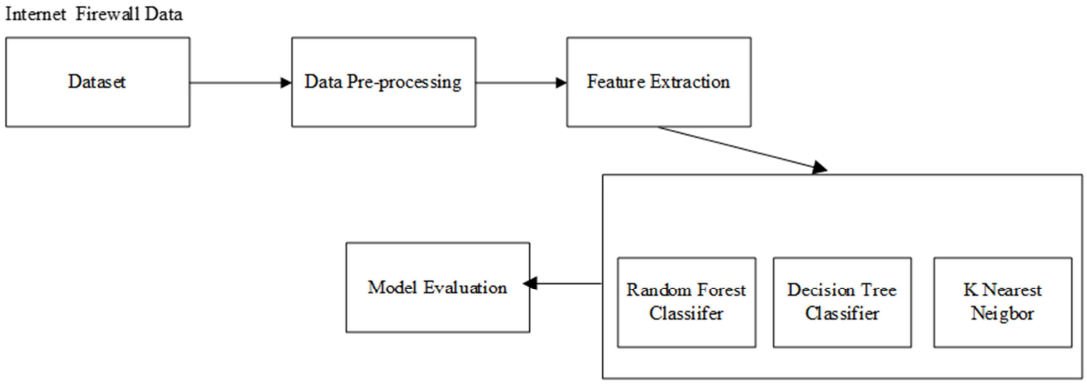

Figure 1. *Architectural Design of the Proposed System.*

4. Experimental Set up

This section describes the results of the analysis. The experimental set up is made up of two phases. The phases are exploratory data analysis and the implementation of the machine. learning algorithms.

4.1. Exploratory Data Analysis (EDA)

An EDA was carried out on the dataset so as to have a better understanding of the firewall dataset. The analysis are performed using correlation matrix, and bar charts. From the conducted analysis, Figure 2 shows the countplot. The countplot depicts that the number of instances in each of the class are not equal. If this is not balanced the model will be biased. To solve this problem, an oversamplier technique was used to populate the minority class, to be of the same size with the majority class. The balanced countplot can be seen in Figure 3. The correlation between the features of the dataset can be seen in Figure 4, and the future ranking can be seen in Figure 5.

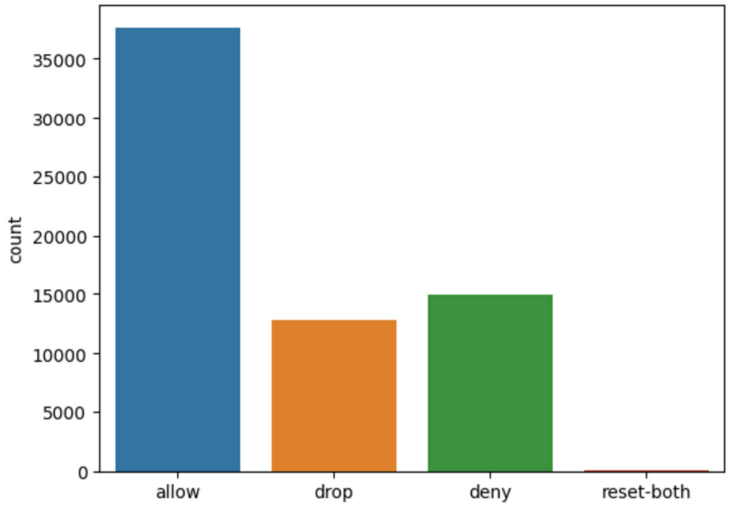

Figure 2. *Countplot of imbalanced Data.*

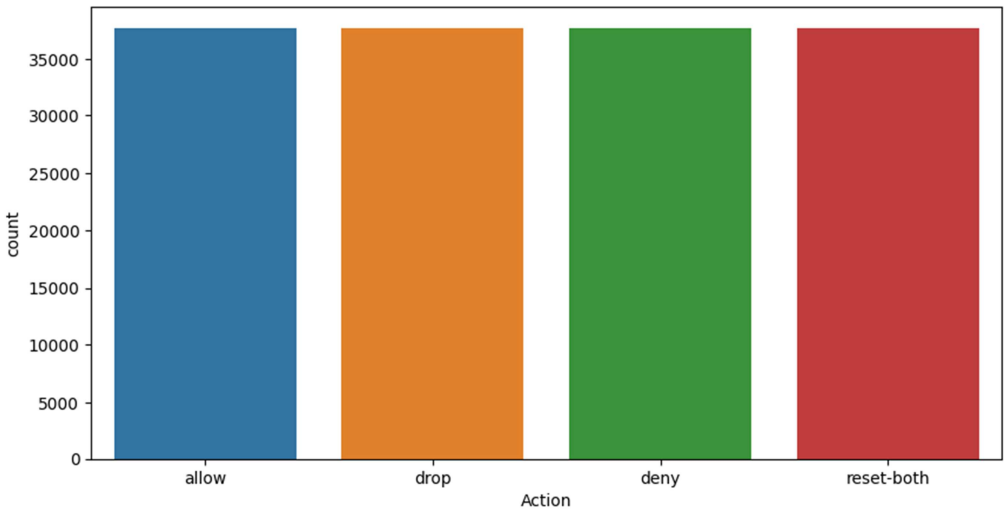

Figure 3. *Countplot of the balanced data.*

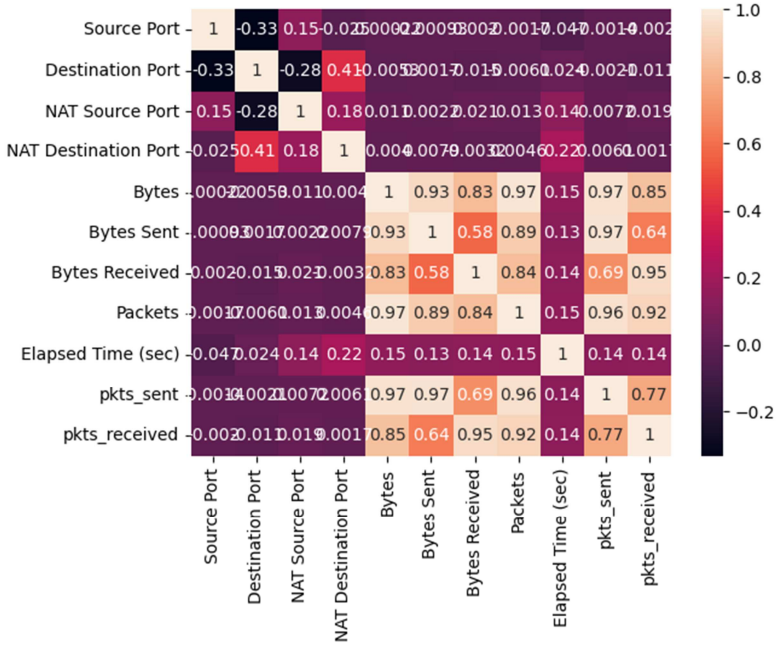

Figure 4. *Correlation matrix.*

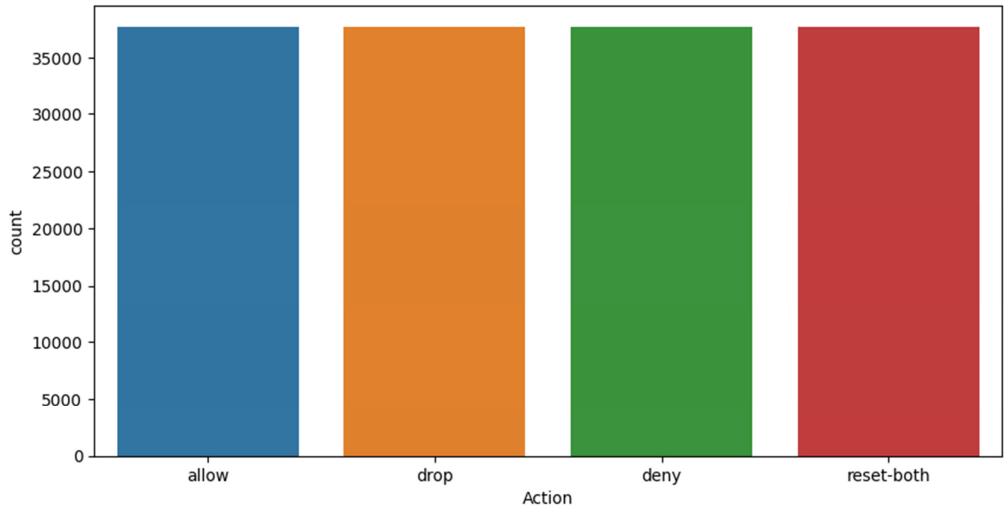

Figure 5. *Countplot of the target column.*

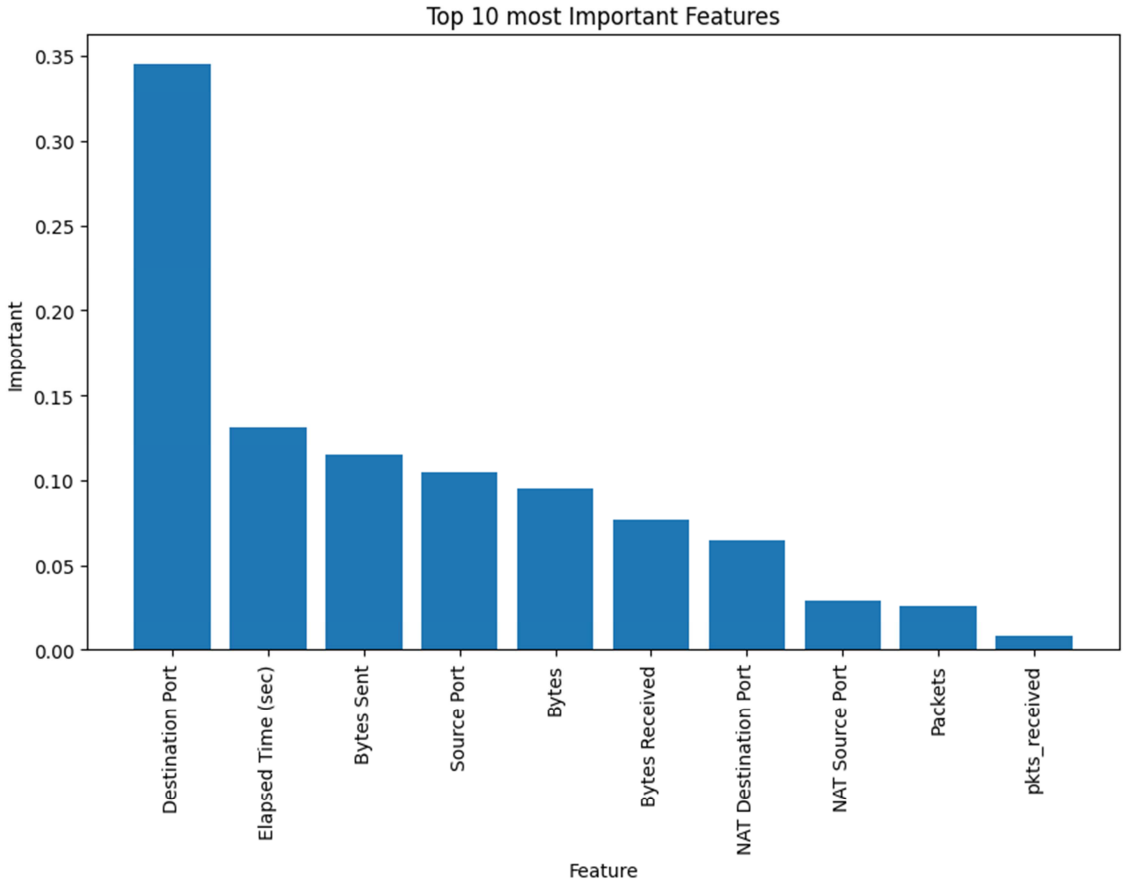

Figure 6. *Feature Ranking.*

4.2. Training Machine Learning Classifiers

The analysis was conducted using three (3) different machine learning algorithm to provide more accurate and efficient diagnostic report. These algorithms were trained on an internet firewall data to help to detect attacks in the internet. Before these algo.

1) Random Forest Classifier was used to train on firewall internet data. This algorithm learns the pattern and features associated with each class from the training data. After training, the model was test data. The result of the random forest model was evaluated using accuracy, precision, F1-score, and recall. Figure 7, and 8 shows the classification report and confusion matrix of the random forest.

Classification_Report For Random Forest

	precision	recall	f1-score	support
allow	1.00	1.00	1.00	7473
drop	1.00	1.00	1.00	7454
deny	1.00	1.00	1.00	7677
reset-both	1.00	1.00	1.00	7508
accuracy			1.00	30112
macro avg	1.00	1.00	1.00	30112
weighted avg	1.00	1.00	1.00	30112

Figure 7. *Classification Report.*

Again confusion matrix was used on the predicted data to know the state of the data in terms of True Positive, True Negative, False Positive and False Negative. The confusion matrix shows the how many data are predicted correctly and

are wrongly predicted.

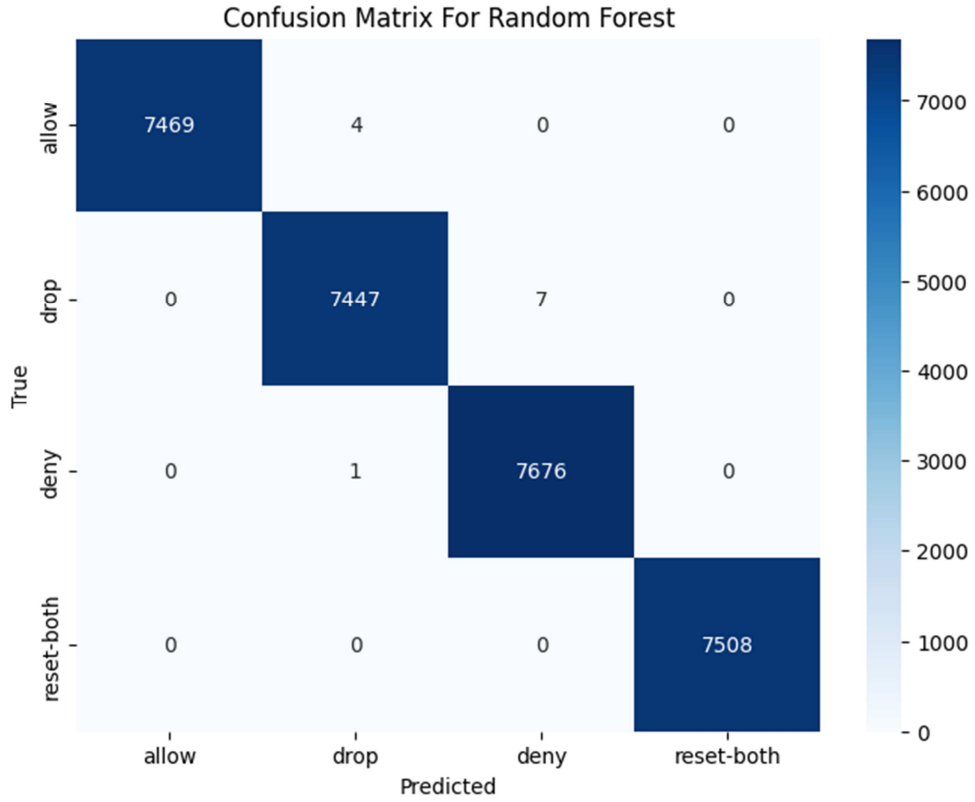

Figure 8. *Confusion matrix of Random Forest.*

2) Decision Tree Classifier was also used on the trained data which learns from the internet firewall data to create a decision tree that maps different patterns and features of dataset feature to a specific attack. After the data has been trained upon, a prediction was made in order to obtain the accuracy score and the classification report.

Classification_Report Decision Tree

	precision	recall	f1-score	support
allow	1.00	1.00	1.00	7473
drop	1.00	1.00	1.00	7454
deny	1.00	1.00	1.00	7677
reset-both	1.00	1.00	1.00	7508
accuracy			1.00	30112
macro avg	1.00	1.00	1.00	30112
weighted avg	1.00	1.00	1.00	30112

Figure 9. *Confusion matrix for Decision Tree.*

Again confusion matrix was used on the predicted data to know the state of the data in terms of True Positive, True Negative, False Positive and False Negative. The confusion matrix shows the how many data are predicted correctly and are wrongly predicted.

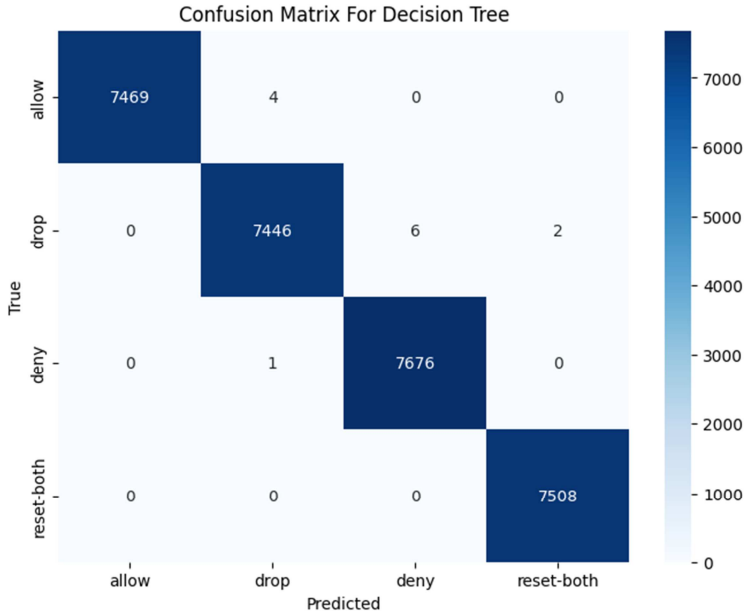

Figure 10. *Confusion matrix of Decision Tree.*

3) K-Nearest Neighbors Classifier can be trained using labeled data. The result of the K-Nearest Neighbor was evaluated using classification report and confusion matrix of K Nearest neighbor can be seen in Figure 11 and Figure 12.

Classification_ReportFor K-Nearest Neighbor

	precision	recall	f1-score	support
allow	1.00	0.99	1.00	7473
drop	0.99	1.00	0.99	7454
deny	1.00	1.00	1.00	7677
reset-both	1.00	1.00	1.00	7508
accuracy			1.00	30112
macro avg	1.00	1.00	1.00	30112
weighted avg	1.00	1.00	1.00	30112

Figure 11. *Classification Report of K-Nearest neighbor.*

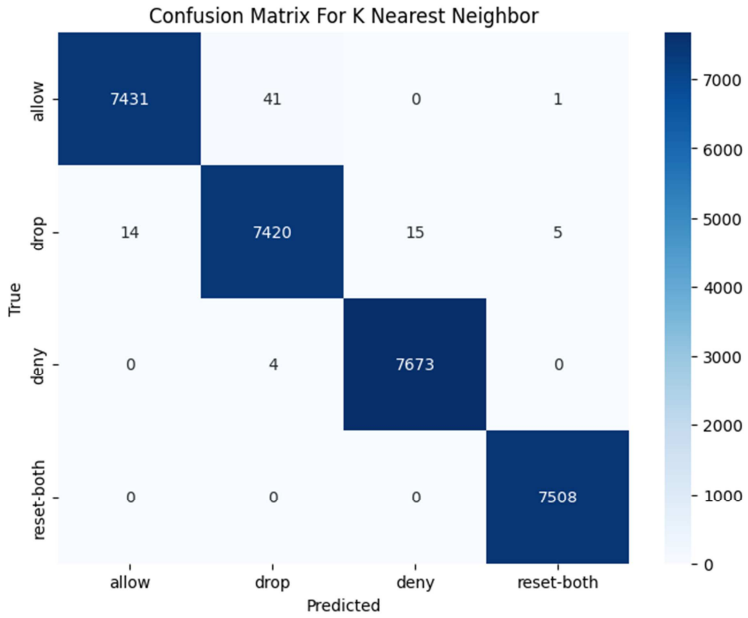

Figure 12. *Confusion Matrix of K-Nearest neighbor.*

5. Conclusion

In conclusion, the integration of machine learning into internet firewall systems has yielded remarkable advancements in cybersecurity. The achieved results of 99.99% precision, recall, and F1-score exemplify the unparalleled efficacy of this approach. By leveraging the power of machine learning algorithms, we have substantially fortified our defences against a myriad of cyber threats, providing a robust shield for critical digital infrastructures. This achievement not only signifies a monumental leap forward in safeguarding online ecosystems but also underscores the potential for continued innovation in the realm of cybersecurity. As we look ahead, further research and development in this domain hold the promise of even more sophisticated and adaptive firewall systems, ensuring a safer digital landscape for users and organizations alike.

ORCID

Martha Ozohu Musa: 0009-0009-7336-5709

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] 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). https://doi.org/10.24018/ejece.2020.4.4.227

[2] Sun, J., Zhong, G., Huang, K., & Dong, J. (2018). Banzhaf random forests: cooperative game theory based random forests with consistency. Neural Networks, 106, 20-29. https://doi.org/10.1016/j.neunet.2018.06.006

[3] 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. https://doi.org/10.1109/access.2020.2974764.

[4] Al-Haija, Q. and Ishtaiwi, A. (2021). Machine learning based model to identify firewall decisions to improve cyber-defense. International Journal on Advanced Science Engineering and Information Technology, 11(4), 1688. https://doi.org/10.18517/ijaseit.11.4.14608

[5] Jordan, M. and Mitchell, T. (2015). Machine learning: trends, perspectives, and prospects. Science, 349(6245), 255-260. https://doi.org/10.1126/science.aaa8415

[6] Khonde, S. and Ulagamuthalvi, V. (2020). Hybrid architecture for distributed intrusion detection system using semi-supervised classifiers in ensemble approach. Advances in Modelling and Analysis B, 63(1-4), 10-19. https://doi.org/10.18280/ama_b.631-403

[7] Dawadi, Babu R., Bibek Adhikari, and Devesh Kumar Srivastava. "Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks." *Sensors* 23, no. 4 (2023): 2073.

[8] Applebaum, Simon, Tarek Gaber, and Ali Ahmed. "Signature-based and machine-learning-based web application firewalls: A short survey." *Procedia Computer Science* 189 (2021): 359-367.

[9] Prabakaran, Senthil, Ramalakshmi Ramar, Irshad Hussain, Balasubramanian Prabhu Kavin, Sultan S. Alshamrani, Ahmed Saeed AlGhamdi, and Abdullah Alshehri. "Predicting attack pattern via machine learning by exploiting stateful firewall as virtual network function in an SDN network." *Sensors* 22, no. 3 (2022): 709.

[10] 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.

[11] Shaheed, Aref, and M. H. D. Kurdy. "Web Application Firewall Using Machine Learning and Features Engineering." *Security and Communication Networks* 2022 (2022).

[12] Ito, Michiaki, and Hitoshi Iyatomi. "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, 2018.

[13] Taylor, O. E., and P. S. Ezekiel. "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, no. 4 (2022): 1-13.

[14] Rajesh, Shriram, Marvin Clement, Sooraj SB, Al Shifan SH, and Jyothi Johnson. "Real-Time DDoS Attack Detection Based on Machine Learning Algorithms." *Proceedings of the Yukthi* (2021).

[15] Lente, Caio, Roberto Hirata Jr, and Daniel Macêdo Batista. "An Improved Tool for Detection of XSS Attacks by Combining CNN with LSTM." In *Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pp. 1-8. SBC, 2021.

[16] Karacan, Hacer, and Mehmet Sevri. "A novel data augmentation technique and deep learning model for web application security." *IEEE Access* 9 (2021): 150781-150797.

[17] Uçar, E. and Ozhan, E. (2017). The analysis of firewall policy through machine learning and data mining. Wireless Personal Communications, 96(2), 2891-2909. https://doi.org/10.1007/s11277-017-4330-0

[18] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140. https://doi.org/10.1007/bf00058655

[19] Cao, H., Sarlin, R., & Jung, A. (2020). Learning explainable decision rules via maximum satisfiability. Ieee Access, 8, 218180-218185. https://doi.org/10.1109/access.2020.3041040

[20] Cutler, D., Edwards, T., Beard, K., Cutler, A., Hess, K., Gibson, J., … & Lawler, J. (2007). Random forests for classification in ecology. Ecology, 88(11), 2783-2792. https://doi.org/10.1890/07-0539.1

[21] Kulkarni, V. and Sinha, P. (2012). Pruning of random forest classifiers: a survey and future directions. https://doi.org/10.1109/icdse.2012.6282329

[22] Quinlan, J. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106. https://doi.org/10.1007/bf00116251

[23] Samworth, R. (2012). Optimal weighted nearest neighbour classifiers. The Annals of Statistics, 40(5). https://doi.org/10.1214/12-aos1049