Enhancing Web Application Security through Deep Learning Techniques and Web Application Firewalls for Detection of Web Attack

Tasdid Hasnain
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh.
tasdidhasnain@gmail.com

Md. Hasibul Islam
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh.
hasibul.syl@gmail.com

Md Shamihul Islam Khan Limon
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh
limon@metrouni.edu.bd

Md. Mushtaq Shahriyar Rafee
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh.
rafee@metrouni.edu.bd

Archi Arani Basak
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh.
archi@metrouni.edu.bd

Mahfujul Hasan
Dept. of Computer Science and Engineering
Metropolitan University
Sylhet, Bangladesh
mahfujul@metrouni.edu.bd

***Abstract*— New techniques and tactics create a possibility to develop a fresh perspective on improving the web application security by incorporating a CNN deep learning model into a Web Application Firewall (WAF). Designed for threat identification including DDoS attacks, SQL injection, and cross-site scripting (XSS) this method adopts the CIC DDoS and CSIC 2010 datasets for training. The DDoS detection model proved a very high accuracy of 99% and the XSS and SQL injection detection model had a detection rate of 97%. A layered CNN architecture where the first layer is designed specifically for DDoS detection, accurate to a 98.27%. Normal traffic as identified by this layer is then forwarded to the second layer responsible for detecting XSS and SQL injection which it does with a 96.08% accuracy. This multilayered approach considerably enhances identification of, and measures against, Web-based threats and strengthens Web application security.**

Keywords: WAF; CNN; XSS; SQL injection; web security.

I. Introduction

It's crucial to comprehend and implement reliable web security measures to reduce risks, safeguard sensitive information, uphold user trust, and maintain the reliability and accessibility of online resources. The importance of a security system increases as the number of internet users increases. A web application firewall (WAF) acts as a barrier between a web application and the client on the internet when it is deployed in front of a web application [4, 5].

A WAF is a type of reverse proxy that protects the web server from being exposed to the client by detecting anomalous traffic in the WAF, while a proxy server acts as an intermediary to protect a client machine's identity. A WAF is controlled by a set of rules known as policies and a pre-trained module to predict new incoming requests. By filtering harmful communications, these policies try to guard against application vulnerabilities. The usefulness of a WAF is derived in part from the speed and ease with which policy modifications may be deployed, allowing for a faster reaction to various attack vectors [5, 6]. Perspective 1 shows a simplified block diagram of a WAF (Web Application Firewall).

One of the common difficulties in various disciplines of computer science is protecting computers and networks from infiltration, theft, and disturbance [3]. The importance of a security system increases as the number of internet users increases. A web application firewall (WAF) acts as a barrier between a web application and the client on the internet when it is deployed in front of a web application [4, 7]. Many attempts have been made to build various security solutions, such as intrusion detection systems (IDS) and firewalls [6]. In most of these cases, network layer firewalls and IDS do not inspect HTTP packets in the application layer [4]. As a result, they are incapable of fully safeguarding web servers [8]. Web applications, especially in the cloud, are one of the most appealing targets for attackers looking to break into an organization's information infrastructure. Internal data leaks, financial losses, and website manipulation can all result from an organization's failure to implement web security. Traditional security solutions, such as network firewalls and intrusion detection systems (IDS), predominantly focus on network traffic monitoring and filtering at the network layer, leaving web servers vulnerable to sophisticated application layer attacks like SQL injection, cross-site scripting (XSS), and dynamic denial of service (DDoS) attacks [4, 8, 9]. Additionally, network layer firewalls and IDS typically do not inspect HTTP packets in the application layer, further limiting their ability to fully safeguard web servers, particularly in

cloud environments where web applications are highly targeted by attackers [8].

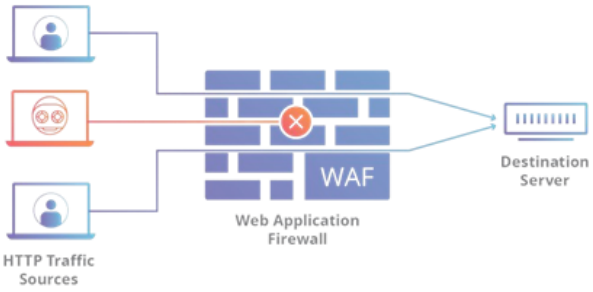


Figure 1. Working of a web application firewall.

We introduce a layered architecture of WAF. Generally, in the WAF, features from the incoming traffic are extracted and tested for different types of threat detection modules and signals. Based on the incoming traffic, a higher rate of detection is filtered in the first layer, and only the filtered traffic from the first layer is processed in the second layer. As the nature of the attack is different for different attacks, extracting the required parameter based on the nature of the attack and predicting new requests using a pre-trained model would increase the performance and accuracy of the WAF. The major contributions of this research are as follows:

- A WAF-layered architecture was proposed for DDoS, SQL injection, and XSS detection in the web-based service system.
- The proposed model's performance was evaluated and achieved 98.27% accuracy with DDoS detection and 96.08% accuracy with XSS/SQL injection detection.

The rest of this paper is organized as follows: Section 2 presents a background study with related work on web application firewall implementation practices. Section 3 provides the methodology of the current research, including system architecture, dataset preparation, and approach for analysis, whereas Section 4 presents the results and analyses. Section 5 concludes the paper[10].

II. Background and Related Work

This study borders on the world of web application security, specifically the pivotal importance of web application firewalls (WAFs) in strengthening the frontline against cyber attackers. Among the most widespread threats are distributed denial-of-service (DDoS) attacks, structured query language (SQL) injection, and cross-site scripting (XSS) vulnerabilities. In the modern digital era, where web applications are the primary vehicles of business activities, trade, and communications, it is essential to develop security and confidentiality measures on the basis of which to safeguard data and, subsequently, preserve the confidence of users. WAFs act as the first line, which starts with the analysis of its traffic, seeking and stopping possible web attacks that could lead to the breach of the confidentiality, integrity, and availability of web resources.

At the core of WAF functionality lies a sophisticated architecture comprising two main modules: the CM and the PAM, which are the core modules of the protocol analyzer. The CM (which stands for Central Management) functions as the core of the WAF, storing rule files and policies and the structures that articulate conditions for traffic filtration and processing. These regulations are specifically elaborated by utilizing the known scenarios of attacks, anomaly detection methods, and security best practices. All these are put together into a comprehensive defense protocol capable of fighting a whole variety of cyber threats. And on the same side, FIP exercises both watchdog functions of the IPS and strictly supervises packet governance, as well as making use of superpack feature extraction and a kind of deep analysis process to aim to distinguish normal from abnormal behavior. By bringing together the different layers of the network, the WAF operates as an entry point; only the traffic that's approved and appropriated sets sail to reach the backend web application, while all other malicious attempts are obstructed on the entry point.

This research paper persuades WAFs to adopt intelligent and farsighted neural networks that draw on machine learning tools, especially convolutional neural networks, as the most formidable defense weapons against constantly evolving cyber threats. Different from conventional signature-based recognition methods, CNN-based technology provides the advantage of learning deep and complex relations and patterns among web traffic data and can therefore outline the abnormalities behind certain actions that are malicious. Through the employment of large-scale datasets that span beyond simple attack scenarios to denser traffic patterns, CNN-based WAF models have the ability to pick out and distinguish complex attack vectors, adjust to newly discovered threats, and imitate such detection capabilities with less human oversight.

The effectiveness of the CNN-based WAF models depends on several key aspects, such as feature extraction, choice of parameters, and model optimization. The data extraction stage consists of identifying and pulling out meaningful characteristics from network packets, including such things as packet size and protocol kind, as well as get methods and payload content, which form an input matrix for deep neural network layers. The adjustment of parameters covers the fine-tuning of hyperparameters, such as the learning rate, batch size, and network architecture design, to determine the model performance and fix problems such as overfitting or underfitting. Model optimization runs on similar lines to that of other neural networks. The model is trained using either general datasets or even custom-designed surroundings for various environments. Techniques like cross-validation, data augmentation, and regularization are used to ensure generalization and robustness.

This research work will establish CNN-based WAF models in the real world by measuring their performance. The outcome will shed light on their effectiveness against cyber threats as well as web application security. By means of accurate research, validity checks, and reference to the benchmarks of

detection accuracy, false positive rates, and response time, this research aims at providing useful tips and guidance on how to use the Convolutional Neural Network (CNN) powered (WAF) in diverse organizational environments. Eventually, an increase in the reliability of web apps in the wake of cyber threats can be obtained through CNN-based intrusion prevention systems (WAFS), which in turn illuminate the ultimate goal of protecting digital assets, preserving user faith, and building a cyber-security ecosystem.

Related Work

Gustavo et al. [11] explored the deep learning techniques implemented in web application firewalls to classify the HTTP traffic. The author used a transformer encoder to analyze the classification of HTTP traffic. Using natural language processing, the authors trained the model by transferring the HTTP traffic to the feature vector.

Moradi et al. [4] used a stacked auto-encoder method in the deep belief network to detect bad HTTP requests. The authors used the n-gram feature extraction model to extract features for model development. Three different machine learning models have been used with the CSIC 2010 and ECML/PKDD 2007 dataset, and compared the performance of these models to verify which had better performance as a web application firewall in the detection of anomalies.

Pen et al. [12] presented the importance of an unsupervised method of machine learning over a supervised learning method for attack detection. The authors proposed an auto-encoder-based model for the detection of such attacks to analyze XSS and SQL injections.

Rajesh et al. [13] analyzed different features including UDP flood attacks, ICMP ping flood attacks, TCP SYN flood attacks, and land attacks to distinguish between normal and DDoS attack traffic. The authors also presented a comparative analysis of the different machine learning methods, including K-nearest neighbour, decision tree, random forest, and naive Bayes.

Lente et al. [14] proposed a new model called 3C-LSTM, which is a combination of LSTM and CNN, claiming it had better accuracy than other models.

The authors used the proposed model for XSS detection, trained by converting words to vectors. This work evaluated the model for different sizes of batch input, and proposed the best batch size for better results.

Keracan et al. [15] proposed using DA-SANA to detect attack traffic by considering the noise coefficient. The author used three datasets, CISC, PKDD, and a generated dataset to analyze the model to present the comparative results. In this work, the authors analyzed attacks including SQL injection, XSS, RCE, CSRF, XXE, and many more.

Liang et al. [16] worked on analyzing URL content and identifying whether a URL had an SQL injection and XSS payload or not. For this, they tokenized and vectorized the URL and used this information to train RNN, LSTM, and GRU machine learning modules. Tekerek et al. [17] used the CSIC2010 v2 dataset to train the CNN, and discussed the advantages of using CNN over ANN. The authors claimed that the proposed deep learning model had higher accuracy than other machine learning models.

To the best of our knowledge, there have been many studies carried out investigating TCP, UDP, SYN, and NTP flood types of DDoS attacks, but not specifically HTTP flood DDoS attacks. Hence, we investigated HTTP flood DDoS attacks and correlated two types of attacks, XSS and SQL injection, with one affecting the availability of service and the other affecting the confidentiality and integrity of the web services.

III. RESEARCH METHODOLOGY

3.1. System working Architecture

Web Application Firewall (WAF) is a system that identifies and inspects HTTP traffic between a web server and client. A particular dataset is used to train them for the identification of normal and malicious traffic. There is another dataset used by the WAF for the DDoS attacks, SQL injections and cross-site scripting attacks. A new HTTP request is taken and its attributes are preprocessed for the purpose of a prediction. If the traffic is classified as malicious, it is eliminated; the next one tests SQL injection and XSS. When the traffic is expected to be normal then the session goes through to the web server, else it is dropped. The first layer of the WAF eliminates high DDoS requests, which raises the accuracy and speed of the system. The proposed WAF consists of two modules: One of them is for detecting DDoS attacks in the first-layer while the second one is for SQL injection and XSS in the second layer. It could be better if the module is trained with separate datasets as the data and the kind of attacks are quite different.

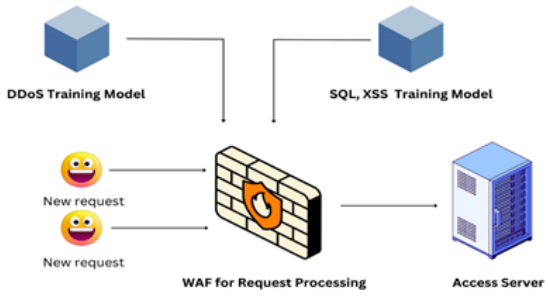


Figure 2. Model development framework.

3.2. Framework of the Proposed Model

In the training period for DDoS, XSS, and SQL injection attacks, two datasets were utilised. The first data set consisted of DDoS detection features they included: Flow ID Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, flow duration total forwarded packets, total backward packets, total length of forward packets, total length of backward packets, maximum forwarded packet length, minimum forwarded packet length, average forwarded packet length, standard deviation forwarded packet length, maximum backward packet length, minimum backward packet length,

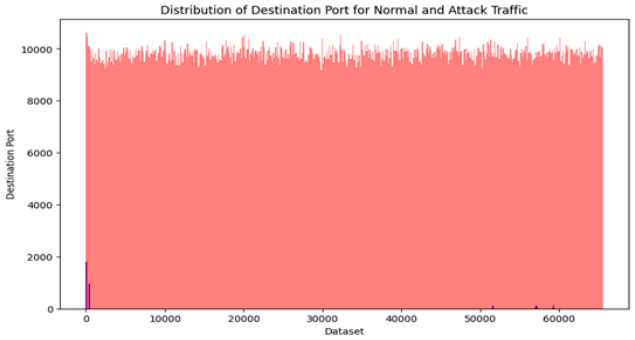
average backward packet length, The second dataset concerned SQL injection plus XSS detection exploring HTTP header and body for the analysis of attack intent. These datasets were employed for creating a training set of a CNN model that developed its ability to detect improper approaches by using certain patterns. Some of the steps of the module included pre-processing of the data to transform the data into standard formats, selected features and parameters and normalization of numerical forms. Some of the examples were representing the procedures GET and POST as 1 and 2 and converting the text flag values to binary form. Numerical parameters were normalized by the min-max normalization to put all ranges through the same transformation in order to simplify the sequential relations handling.

3.3. Dataset Collection

The standard datasets CIC DDoS using on DDoS attack detection and XSS/SQL injection attack detection using the CSIC 2010 dataset. The process of collecting data involves several important steps. Firstly, you'll need to locate and obtain access to the CIC DDoS dataset and the CSIC 2010 dataset from their respective sources, ensuring that we have permission to use them for our research. Once we have these datasets, we'll need to take the time to understand their contents, how they're organized, and their overall structure.

3..4 Dataset Representations for DdoS

Visualizations of the CIC dataset are presented in Figure 3. We considered almost 490 K data samples from the CIC2019 dataset, where 80% was the attack dataset and 20% was the normal dataset. During the analysis of 20 normalized features, the most distinguishable parameters that differentiated between normal and attack traffic were push flag, flow rate, port used, protocol used, and urgent flag.



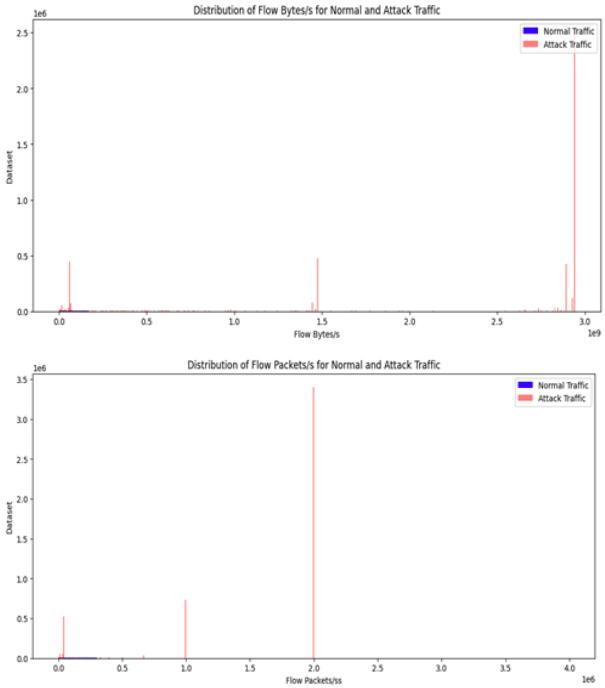


Figure 3. DDoS CIC 2019 dataset representation.

IV. Results and Analysis

4.1 2019 DDoS CIC Dataset

The analysis of CIC dataset,with source of 4,912,019 samples, shows that DDoS has relatively unpredictable flow rate in its traffic and TCP applications dominate its usage. Attack traffic contains more packets per second than normal traffic; attack packers are much greater in value. Of normal traffic most of them are inclined to select standard and usually not risky ports while attack traffic comes along with a pool of ports. Target segments of DDoS attacks are usually segments of lesser length. Furthermore, while the above chart shows a rise in push/urgent flags over February's total traffic, there is a significantly higher amount in the cyber-attack traffic.

4.2 XSS and SQL Injection Dataset

We analyzed the CISC 2010 normal and attack traffic dataset [9]. SQL Injection and XSS dataset presents data, which is entirely designed for the researcher and analyst to accomplish the studies of cross-site scripting (XSS) and SQL injection attacks. The dataset under consideration is usually composed of HTTP requests and responses. The dataset is generated automatically and contains 36,000 normal requests and more than 25,000 anomalous requests. The HTTP requests are labeled as normal or anomalous and the dataset includes attacks such as SQL injection, buffer overflow, information gathering, files disclosure, CRLF injection, XSS, server side include, parameter tampering and so on. This dataset has been successfully used for web detection in previous works [18, 19, 20, 21, 22, 23].

4.3 Train DDoS model with dataset

Data processing is a crucial part of training a machine-learning model. The first step involves replacing infinite values with NaN (not a number) to identify and solve possible outliers or anomalies that may result in erroneous analysis or poorly modeled outcomes. This standardization factor ensures a trustworthy and strong data set free from bias or errors, which could otherwise interfere with downstream analyses.

The second step is component selection from the target value for supervised learning, which is responsible for the independent career of predictors (features) and the targeted, resulting in more accurate modeling and predicting. Label encoding is applied to transform the target variable from categorical labels to numerical value representation, which is a critical preprocessing step in machine learning applications. The use of LabelEncoder for categorical features with their encoding to 'Flow ID', 'Source IP', 'Destination IP', and 'Timestamp' allows them to be used in machine learning models.

In the initial processing, a subset containing the 20 most important features relevant to network traffic analysis is pulled from the dataset. These features include identifying central network attributes like 'Flow ID', 'Source IP', 'Source Port', 'Destination IP', 'Destination Port', 'Protocol', 'Timestamp', 'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets', 'Total Length of Fwd Packets', 'Total Length of Bwd Packets', 'Fwd Packet Length Max', 'Fwd Packet Length Min', 'Fwd Packet Length Mean', 'Fwd Packet Length Std', 'Bwd Packet Length Max', 'Bwd Packet Length Min', 'Bwd Packet Length Mean' and 'Bwd Packet Length Std'.

Normalization of features is achieved using StandardScaler, which guarantees all items to have a mean of zero and a standard deviation of one, helping the vectorization process converge and perform efficiently for machine learning methods. The original feature data is also separately standardized and represented by 'X_train_20features_std' and 'X_test_20features_std'.

Using convolutional neural networks (CNN), approximately 20% of the dataset was reserved for the testing process, while the rest was allocated for the training process. The model architecture comprises three convolutional layers with dropout layers placed between them for reducing overfitting and two dense layers for classification. The create_cnn_model function is used to create a CNN model with an input shape and specific activation functions.

The model is trained on the given training data with early stopping based on the validation loss. Training history plots show the improvement of the model, making it visible. Finally, the model is evaluated on the testing set, and the results of test loss, accuracy, and prediction are printed.

Using CNN, the DDoS model gave the best outcomes, with a test accuracy of 99.99% using 20 epochs and a small test loss of 0.000582, indicating that the classifier identified the majority of test cases. Validation of the model with 30% of the training data during training ensured that the model could generalize unseen data, proving the robustness and reliability of the model in the successful detection of DDoS attacks.

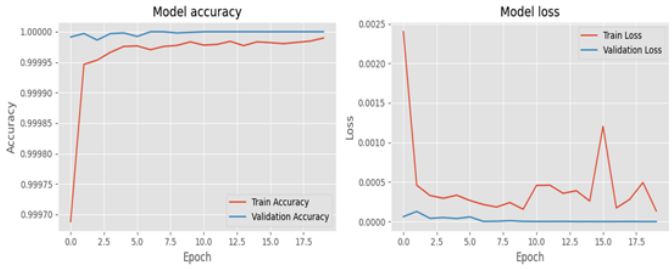


Figure 4: Training, test and loss accuracy of CNN model for DDoS detection.

The confusion matrix named ‘0’ for the normal data and ‘1’ for the malicious traffic consists of 659 correctly identified normal traffic and 981744 correctly identified malicious traffic. Normal traffic was not misclassified and, thus, the system knows how to discern normal traffic, but only as long as bad traffic conforms to the shape of normal traffic (false positives).

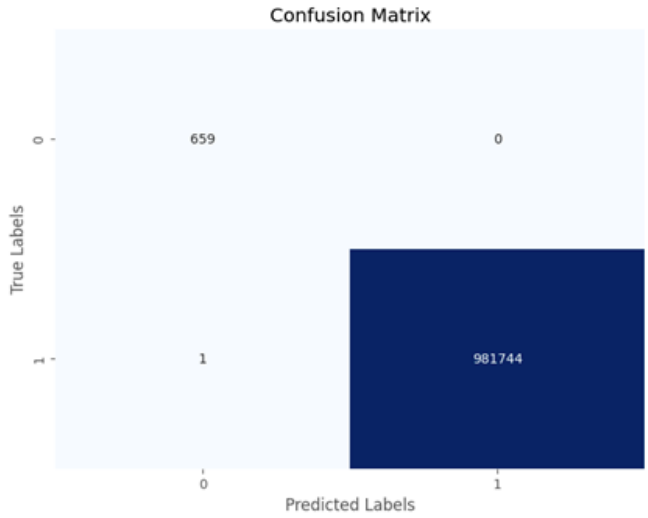


Figure 5. Confusion matrix for DDoS detection with CNN.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	659
1	1.00	1.00	1.00	981745
accuracy			1.00	982404
macro avg	1.00	1.00	1.00	982404
weighted avg	1.00	1.00	1.00	982404

Figure 6. Classification report for DDoS detection with CNN.

In the case of the classification tasks used in the paper, the paper used ‘K-fold cross-validation’ with a ‘CNN’ model, and

the folds were also further stratified in the study. This is achieved in such a way that with each fold the distribution of the input data classes of the drawn data set is preserved; The model is developed from the input data and labels data. As it has been mentioned, it is also necessary to note that the used CNN model is trained in the training fold, and the model performance is evaluated according to a validation subset. Finally, the proposed model can predict the accuracy and the loss with the test data as shown in the following equations. The average accuracy and loss query is computed on all the folds and they are stored for all of them. This means that on an average it has an average accuracy of 99% and on an average . Worst at −0020 loss and also at zero as well as at the lowest value in the folds.

4.4 Train XSS and SQL injection model with dataset

First of all, it should be noted that in the case of the dataset available, this one is subdivided into training dataset, validation dataset and test dataset. If the text data has to be ready for modeling then we join necessary features in it In order that the text data be ready for modeling we join the requisite features. These are converted to one shot with respect to the labels as also the text encodings are done with help of concatenation. ’ Third, the usage of the text data of them is normalized and brought into the same dimension for model use after tokenizing and padding of sequences. Concerning the architecture of the model used in work, here is CNN-GRU on regularization, which was distinguished in work to avoid a gift for overfitting. Some architectures out of the lot are embedding, convolutional, max-pooling, drop-out, GRU, and dense architecture. The training of the model is performed with categorical cross-entropy loss and also with Adam optimizer. When training, records are made involving the generality set accuracy and the validity set accuracy. And last but not the least when the training is done we have to check the percent accuracy of the validation and the test set and also observe the training as well as validation graph. We shall then demonstrate the use of the test set in the implementation of the model by mapping probabilities to class labels by employing equation 10 and evaluate the performance of the model by utilizing the confusion matrix together with precision, recall and F1-score. Last of all to get the final classification report we add up the summing up classification report that provides the model accuracy at the various classes. In the case of an epoch value of 50, CNN achieves an accuracy of 97.83% while using the 80:20 split. 0. 1822 lost actually a test accuracy that and in so doing arrives This is an indication that most of the samples in the test set were classified with a reasonable level of accuracy by the model with some level of loss value. From this perspective, it can be proved that the model was able to capture the observed data during conducting the research.

Figure 7: Training, test and accuracy of CNN model for XSS, SQL injection detection.

The confusion matrix has "normal" and "malicious." There were 777 instances with the correct normal traffic class and 917 instances with the correct malicious traffic class. However, the model misclassified the normal traffic 59 times as malicious, and it misclassified the malicious traffic 5 times as normal.

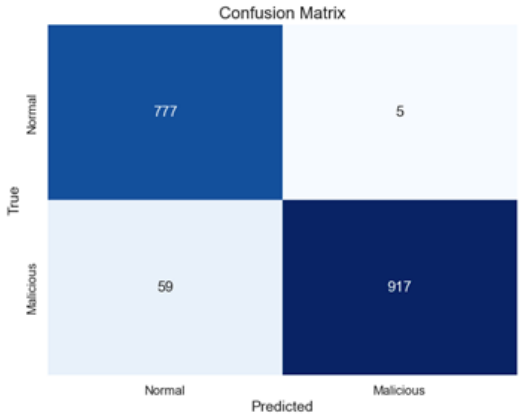


Figure 8: Confusion matrix of CNN model for XSS, SQL injection detection.

	precision	recall	f1-score	support
0	0.96	0.99	0.98	782
1	0.99	0.97	0.98	976
accuracy			0. 98	1758
macro avg	0.98	0. 98	0. 98	1758
weighted avg	0. 98	0. 98	0. 98	1758

Figure 9: Classification report of CNN model for XSS, SQL injection detection.

4.5 Testing the Combined Model with New Test Dataset

In our prediction system we use layer architecture. The first layer is DDoS attack detection using pre-train model and second layer is XSS, SQL injection attack detection using pre-train model. in this WAF traffic comes the first layer to check DDoS attacks if normal then pass into the second layer else drop this request. In the second layer it checks the XSS and SQL injection attack if normal then passes into the server else drop this request. Testing our combined model with the

new test dataset, we passed the DDoS traffic and applied it to our pre-trained model for prediction. It was able to correctly detect 98.27% of the traffic. Second layer it correctly detected 96.08% of HTTP packets with the payloads.

The prediction system first loads the pre-trained machine learning models and tokenizers needed for a botnet attack and SQL injection/XSS detection. After that, it runs the preprocessing of both types of attack datasets. With an instance of DDoS detection, the data is standardized using scaler, however, for SQL injection and XSS detection the textual features are tokenized and their sequences are padded up to a fixed length. After that, the detection involves two levels, the first layer being a would-be victim and the second one being a real victim. In the very initial layer, the DDoS detection model foretells whether the incoming examples are associated with DDoS attack or not. In a second degree, the SQL-injection and XSS respectively are also predicted by a model that detects whether the input instances consist of SQL-injection or XSS attacks. In the student layer, detailed analysis is performed on previous overall exam results such as the percentage of exams taken per session/class and the performance level and the consistency with respect to the number of lectures.

Conclusions

The implemented model, the CNN one, stood out as a powerful tool to detect the mentioned attacks (DDoS, XSS, and SQL injection) with a good accuracy. The start of the detection process aimed at spotting DDoS attacks brought the accuracy up to 98.27%. Later, the 2nd level was devoted to XSS and SQL injection tests, scoring 96.08%. Using thorough feature and parameter analysis to address detection of false positives in traffic filtration during the Web Application Firewall (WAF), we have secured the implementation of the WAF on our server. The advanced DDoS detection system allows the traffic of DDoS to be controlled as the initial layer, thus enhancing the speed in which this targeted traffic can get to the destination. Lastly, the performance report showed the limited interruption caused by the rendering of new filtering systems thus the perfect journey of the users. The paper above is concentrated on DDoS, SQL injection, and XSS attacks, but the wider area of web security may be considered including the cases of RCE and malware. Furthermore, evaluating various deep learning algorithms could help improve our web application firewall and make it much more secure.

References

[1] Understanding Web Application Firewalls (WAFs) and Their Role in Cybersecurity. Retrieved from Link

[2] What is a Web Application Firewall (WAF)? Retrieved from Link

[3] Krishnan, M.; Lim, Y.; Perumal, S.; Palanisamy, G. Detection and defending the XSS attack using novel hybrid stacking ensemble learning-based DNN approach. Digit. Commun. Netw. 2022, 2352–8648. [CrossRef]

[4] Moradi Vartouni, A.; Teshnehlab, M.; Sedighian Kashi, S. Leveraging deep neural networks for anomaly-based web application firewall. IET Inf. Secur. 2019, 13, 352–361. [CrossRef]

[5] Ito, M.; Iyatomi, H. Web application firewall using character-level convolutional neural network. In Proceedings of the 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), Penang, Malaysia, 9–10 March 2018; pp. 103–106.

[6] Hao, S.; Long, J.; Yang, Y. Bl-ids: Detecting web attacks using bi-lstm model based on deep learning. In Proceedings of the Security and Privacy in New Computing Environments: Second EAI International Conference, SPNCE 2019, Tianjin, China, 13–14 April 2019; pp. 551–563.

[7] Appelt, D.; Nguyen, C.D.; Panichella, A.; Briand, L.C. A machine-learning-driven evolutionary approach for testing web application firewalls. IEEE Trans. Reliab. 2018, 67, 733–757.

[8] Jaki´c, P.; Hajjaj, F.; Ibrahim, J.; Elsadai, A. The Overview of Intrusion Detection System Methods and Techniques. In Proceedings of the Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research; Singidunum University: Belgrade, Serbia, 2019; pp. 155–161.

[9] Moradi Vartouni, A.; Mehralian, S.; Teshnehlab, M.; Sedighian Kashi, S. Auto-Encoder LSTM Methods for Anomaly-Based Web Application Firewallall. Int. J. Inf. Commun. Technol. Res. 2019, 11, 49–56.

[10] Dawadi, B.R.; Adhikari, B.; Srivastava, D.K. Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. Sensors 2023, 23, 2073. https://doi.org/10.3390/s23042073

[11] Montes, N.; Betarte, G.; Martínez, R.; Pardo, A. Web Application Attacks Detection Using Deep Learning. In Proceedings of the Iberoamerican Congress on Pattern Recognition, Porto, Portugal, 10–13 May 2021; pp. 227–236.

[12] Pan, Y.; Sun, F.; Teng, Z.; White, J.; Schmidt, D.C.; Staples, J.; Krause, L. Detecting web attacks with end-to-end deep learning. J. Internet Serv. Appl. 2019, 10, 1–22. [CrossRef]

[13] Rajesh, S.; Clement, M.; SB, S.; SH, A.S.; Johnson, J. Real-Time DDoS Attack Detection Based on Machine Learning Algorithms. In Proceedings of the Yukthi 2021—The International Conference on Emerging Trends in Engineering—GEC Kozhikode, Kerala, India, 27 September 2021.

[14] Lente, C.; Hirata, R., Jr.; Batista, D.M. An Improved Tool for Detection of XSS Attacks by Combining CNN with LSTM. In Proceedings of the Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, Florianis, Brazil, 12–15 September 2021; pp. 1–8.

[15] Karacan, H.; Sevri, M. A Novel Data Augmentation Technique and Deep Learning Model for Web Application Security. IEEE Access 2021, 9, 150781–150797. [CrossRef]

[16] Liang, J.; Zhao, W.; Ye, W. Anomaly-based web attack detection: A deep learning approach. In Proceedings of the 2017 VI International Conference on Network, Communication and Computing, Kunming, China, 8–10 December 2017; pp. 80–85.

[17] Tekerek, A. A novel architecture for web-based attack detection using convolutional neural network. Comput. Secur. 2021, 100, 102096. [CrossRef] [9] Giménez, C.T.; Villegas, A.P.; Marañón, G.Á. HTTP Data Set CSIC 2010; Information Security Institute of CSIC (Spanish Research National Council): Madrid, Spain, 2010; Volume 64.

[18] A. Perez-Villegas, C. Torrano-Gimenez, G. Alvarez. Applying Markov Chains to Web IntrusionDetection. In Proc. of Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2010), pp. 361-366. Publicaciones urv. Tarragona (España), 7-10 Septiembre (2010).

[19] C. Torrano-Gimenez, A. Perez-Villegas, G. Alvarez. An anomaly-based approach for intrusion detection in web traffic. Journal of Information Assurance and Security, vol. 5, issue 4, pp. 446-454. ISSN 1554-1010 (2010).

[20] C. Torrano-Gimenez, A. Perez-Villegas, G. Alvarez, A Self-Learning Anomaly-Based Web Application Firewall. In Proc. of 2nd International Workshop in Computational Intelligence in Security for Information Systems (CISIS 09). Advances in Intelligent and Soft Computing, vol. 63, pp. 85-92, Springer-Verlag. A. Herrero, P. Gastaldo, R. Zunino, E. Corchado, editores. Burgos (España), 23-26 Septiembre (2009).

[21] C. Torrano-Gimenez, A. Perez-Villegas, G. Alvarez, An Anomaly-based Web Application Firewall. In Proc. of International Conference on Security and Cryptography (SECRYPT 2009), pp. 23-28. INSTICC

Press. E. Fernández-Medina, M. Malek, J. Hernando, editores. Milán (Italia), 7-10 Julio (2009).

[22] H. Nguyen, C. Torrano-Gimenez, G. Álvarez, S. Petrovic, K. Franke, Application of the Generic Feature Selection Measure in Detection of Web Attacks. In Proc. of International Workshop in Computational Intelligence in Security for Information Systems (CISIS 11), LNCS 6694, pp. 25–32. Editor Á. Herrero and E. Corchado, Springer-Verlag. Torremolinos, Málaga (España), Junio (2011).

[23] C. Torrano-Gimenez, H. Nguyen, G. Álvarez, S. Petrovic, K. Franke, Applying Feature Selection to Payload-Based Web Application Firewalls. In Proc. of International Workshop on Security and Communication Networks (IWSCN 11), pp. 75-81. Editor Patric Bours. Gjøvic (Noruega). ISBN: 978-82-91313-67-2. 18-20 Mayo (2011).