International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8508
Article Received: *27 July 2023* ***Revised:*** *26 September 2023* ***Accepted:*** *15 October 2023*

Shielding against Web Application Attacks - Detection Techniques and Classification

[1]Sathya Priya.S, [2]Hariharan. B, [3]Nithisha. J
[1]Department of Computer Science and Engineering
SRM Institute of Science and Technology, Ramapuram
Chennai, India.
e-mail: sathyapriya80@gmail.com
[2]Department of Computer Science and Engineering
SRM Institute of Science and Technology, Ramapuram
Chennai, India.
e-mail: virathari007@gmail.com
[3]Department of Computer Science and Engineering
SRM Institute of Science and Technology, Ramapuram
Chennai, India.
e-mail: nithisha.j@gmail.com

Abstract: The field of IoT web applications is facing a range of security risks and system attacks due to the increasing complexity and size of home automation datasets. One of the primary concerns is the identification of Distributed Denial of Service (DDoS) attacks in home automation systems. Attackers can easily access various IoT web application assets by entering a home automation dataset or clicking a link, making them vulnerable to different types of web attacks. To address these challenges, the cloud has introduced the Edge of Things paradigm, which uses multiple concurrent deep models to enhance system stability and enable easy data revelation updates. Therefore, identifying malicious attacks is crucial for improving the reliability and security of IoT web applications. This paper uses a Machine Learning algorithm that can accurately identify web attacks using unique keywords. Smart home devices are classified into four classes based on their traffic predictability levels, and a neural system recognition model is proposed to classify these attacks with a high degree of accuracy, outperforming other classification models. The application of deep learning in identifying and classifying attacks has significant theoretical and scientific value for web security investigations. It also provides innovative ideas for intelligent security detection by classifying web visitors, making it possible to identify and prevent potential security threats.

Keywords-Web Application Attacks, Machine Learning, Web Application Attack Detection, Web Attacks Models.

I. INTRODUCTION

Web applications have become a crucial component of our daily lives as more personal data and applications migrate to the cloud. However, they are a prime target for assaults due to the enormous volumes of sensitive user data they store. DDoS assaults, when requests are purposefully engineered to overload the server, are among the most often exploited online application vulnerabilities. The biggest online application security vulnerabilities include SQL injection and Cross-Site Scripting.

There are two basic strategies for detecting such attacks: signature-based and anomaly-based methods. While anomaly-based approaches create typical request profiles to spot aberrant requests, signature-based approaches look for specific attack patterns in requests. As it typically has a lower false alarm rate and higher accuracy, the signature-based technique is more widely utilized. However, the rule-based method used by Web Application Firewalls (WAF) has limitations.

The effectiveness of WAFs' rule-based methods depends on how extensive their rule sets are, but are unable to detect assaults that are not in their signature collection. By substituting new keywords for the existing malicious requests that are repeatedly encoded themselves, attackers can easily get by WAFs. A large attack pattern set or lengthy queries also use a lot of computer power when comparing patterns.

In summary, while signature-based approaches are more commonly used due to their accuracy, the limitations of rule-based methods used by WAFs make them less effective in identifying and preventing attacks. Therefore, there is a need for new approaches and techniques to improve the security of web applications and prevent the exploitation of vulnerabilities.

A. *Web application attacks*

Web application attacks are malicious activities that target web applications to compromise their security and gain unauthorized access to sensitive data or systems. These types of attacks pose a significant threat to individuals and businesses that use web applications to store or process valuable information. Web application attacks exploit vulnerabilities in the application's code or infrastructure to gain access, steal sensitive data, or disrupt the application's normal operations. There are various forms of web application attacks, such as SQL injection, cross-site scripting, and distributed denial of service (DDoS) attacks. Since web applications store, large amounts of sensitive data, they are prime targets for attackers, making it essential for businesses to implement robust security measures to protect their web applications.

Different types of web application attacks use specific methods of attack and have unique characteristics.

B. Types of Web application assaults

Web application assaults are malicious attacks that target web applications and exploit vulnerabilities in their code or infrastructure. There are different types of web application assaults, and each type has its specific characteristics and methods of attack. The common types of web application assaults include cross-site scripting (XSS), SQL injection, cross-site request forgery (CSRF), distributed denial-of-service (DDoS), clickjacking, and file inclusion attacks.

Cross-site scripting (XSS) involves injecting malicious code into a web page viewed by other users to steal sensitive information or perform malicious actions on the user's behalf. SQL injection involves injecting SQL commands into a web application's database to access or modify the database, steal sensitive information, or perform other malicious actions. Cross-site request forgery (CSRF) tricks users into acting on a web application without their knowledge or consent, while DDoS floods a web application with traffic from multiple sources to overwhelm its servers and render it unavailable to legitimate users.

Clickjacking involves tricking users into clicking on a button or link disguised as something else to perform a malicious action, such as installing malware or stealing sensitive information. File inclusion attacks manipulate a web application's code to include a malicious file on the server, allowing the attacker to execute arbitrary code or perform other malicious actions. Web application assaults can have severe consequences, including the theft of sensitive data, financial loss, and damage to reputation.

C. Prevention of Web application assaults

Preventing web application assaults is essential in safeguarding the sensitive data stored within web applications. The consequences of such attacks can be severe, including damage to reputation, financial loss, and even legal action. Therefore, it is critical to take effective measures to prevent web application assaults.

Using a Web Application Firewall (WAF) is one of the best techniques to stop attacks on web applications. By examining the traffic between a web application and the internet, a WAF is a security solution created to filter out nefarious requests. It monitors incoming requests using a set of rules and denies those that match a known attack pattern. Organizations can considerably lower the risk of attacks like SQL injection and cross-site scripting (XSS) by utilizing a WAF.

Another effective prevention measure is to keep all software up to date, including the web application itself, the operating system, and any other components used in the web application stack. Keeping software updated ensures that known vulnerabilities are patched and reduces the risk of attackers exploiting them.

Implementing secure coding practices is another way to prevent web application assaults. Developers should be trained to write secure code that is free from common coding mistakes that could lead to vulnerabilities. This includes validating all user input, sanitizing data, and escaping special characters.

Organizations can also use vulnerability scanning tools to identify potential vulnerabilities in their web applications. These tools can scan the web application and identify vulnerabilities that attackers could exploit. Regular vulnerability scans can help organizations identify and fix vulnerabilities before attackers can exploit them.

Preventing web application assaults is critical to protecting sensitive data and preventing reputational damage. By implementing measures such as using a WAF, keeping software updated, implementing secure coding practices, and performing regular vulnerability scans, organizations can significantly reduce the risk of web application assaults [1].

D. Deep learning models

Deep learning models are complex neural networks that are capable of learning from vast amounts of data by using a multitude of parameters. They can extract intricate patterns and features from the input data and have shown remarkable performance in tasks like image and speech recognition, natural language processing, and robotics.

One such deep learning architecture is MRN, which employs multiple CNNs to extract features at different scales and resolutions for image classification tasks. The idea behind MRN is that it can capture both fine and coarse-grained features of an image, leading to improved accuracy.

Another algorithm used in this proposed system is LSTM, which has three gates (input, output, and forget). This selective memory can improve performance in language modeling, speech recognition, and sentiment analysis.

FastText, developed by Facebook's AI Research team, is a scalable and efficient text classification algorithm based on the concept of word embeddings. It represents each word as a vector of real numbers, and these embeddings are then aggregated to represent the document as a whole. FastText uses hierarchical softmax to predict the class of the document and can handle large and imbalanced datasets. It performs well in sentiment analysis, topic classification, and spam detection tasks.

The proposed system employs deep learning algorithms to analyze and classify data effectively. The quality and quantity of training data determine how effective these models are, therefore the system needs a substantial volume of high-quality data to get the best outcomes. Additionally, the system requires regular updates to better account for changes in data trends.

Deep learning models have revolutionized various fields by significantly enhancing the speed, accuracy, and efficiency of many processes. This has led to improved performance and increased productivity in different areas, including healthcare, finance, and marketing. Deep learning models can identify patterns and extract valuable insights from large datasets, enabling businesses to make informed decisions and optimize their operations. However, the accurate use of these models necessitates careful consideration of several criteria, such as model selection, data quality, and regular updates.

Section I deals with the introduction of web assaults, types, detection techniques, and deep learning models used in the proposed paper.

Section II contains all the Literature works and proposed models of the different authors

Section III deals with the proposed model and how web assault detection works and classifies different web attacks.

Section IV contains modules and implementation techniques used in this model to propose the web attack detection system.

Section V contains the outcomes and analysis of the suggested model.

Section VI consists of the conclusion of the proposed system and its advantages and future work on the system.

II. REVIEW OF LITERATURE

As cloud technologies and the Internet of Things continue to advance, a vast amount of data is being transmitted from various sensors and devices to cloud server farms for further analysis. While cloud-based administration and storage provide crucial services, they also present significant security risks, such as data misuse and congested cloud web servers [2]. Moreover, cloud IoT structures increase the likelihood of web server attacks as data centralization transmits information over a grant-based connection. In light of large-scale streaming learning, Zhihong Tian [12]/2020 suggests a web attack region structure that makes use of breaking URLs. The design is intended to detect cyberattacks and send alarming messages. From the perspective of the Edge of Things (EoT), the cloud addresses the aforementioned issues. Different basic models are used to manage the ampleness of the system and the solace of reviving.

Mohammad Hossein Amouei.[2]/2020 presented a Support Learning-Driven and Versatile Testing (Rodent), a mechanized black-box testing 8 methodology to find infusion weaknesses in WAFs. Specifically, they center around SQL injection and Cross-site scripting, which have been among the main ten weaknesses over the last ten years, yet Rodent can test other string-based code infusion assaults also. All the more explicitly, Rodent consequently extricates designs from assault tests of a far-reaching payload assortment, and groups comparative examples together [3]. Then, at that point, it uses a support learning procedure joined with a clever versatile hunt calculation to look through the bunches and effectively find practically all bypassing assault designs. Most of the time, the check structures that are used for online applications rely on common usernames and secret word-based passwords, which are easy to break into. Given the improved encryption method, progress is being made toward various complex client check plans eventually.

Rashidah F. Olanrewaju.[13]/2021 proposed a robust client authentication system for web applications that aims to provide a frictionless login experience for users. The system uses an electronic verification scheme during client-driven login events and follows a clear process for verifying the user's identity at the login interface. The capabilities of the profiler and authenticator determine which of the four login components the proposed system will be used. The profiler component builds a user profile using social data from the client, including login time, device location, browser, and information about online service access. The system processes this data and creates a client profile using a profiler that utilizes the authenticator capabilities. The study results show that the proposed system outperforms other authentication plans for real-time web-based applications. Compared to the current authentication plans for premium web applications, the proposed strategy reduces delay by about 10%, increases response time by 7%, and limits memory use by 11%. Overall, the proposed system is effective in providing a frictionless login experience for users, while ensuring the security and privacy of their personal information. This is obvious proof that security issues connected with web applications are, as of now, certified issues. In the examination of Chang and Choi, the makers inspected permission control and client confirmation and endeavored to research an immense issue and related investigation challenges.

Facial approval has turned into an ever-expanding number of renowned confidential contraptions. As a result of its convenience, it very well may be for the most part sent for web organization confirmation as soon as possible, by which people can without a doubt sign on to online records from different devices without recollecting long passwords. In any case, the increasing number of attacks on artificial intelligence, especially on deep neural networks (DNN), which are routinely used for facial affirmation, poses gigantic troubles for the successful completion of such web-based face affirmation. Even though there has been a focus on shielding some computer-based intelligence attacks, we have close to zero insight into a specific effort committed to the web organization's facial check setting.

Dalton Cole. [26] at first, shows another information-harming attack that doesn't have to have any data on the server side and just necessitates an unassuming bundle of vindictive photo mixtures to enable an attacker to copy the setback in the ongoing facial approval structures easily. Then, at that point, they propose a smartwatch approach called Defeat that utilizes significant learning techniques to distinguish such behaviors. Revathy, S.[4] proposed hybrid machine learning attack detection algorithm based on feature selection and feature discretization methods like Equal Width Discretization integrated with SSC-OCSVM algorithm to identify security attacks.

A framework based on stopping DNS attacks was proposed by Wen-Bin Hsieh [15] \ 2020. Programmers use a variety of tactics to launch digital assaults to take advantage of the Web, which has possibly emerged as the most important innovation on the planet. One of the most well-known social engineering attacks is phishing, which is frequently used to steal credit card numbers and login credentials from customers [5]. Even though the vehicle layer security test is used to check a website's trustworthiness, there are still flaws. A lot of research has been done on believed IP addresses, including IP whitelisting. A smart contract can be utilized to secure the URL and IP address of a permission site on the blockchain. This can be achieved by implementing a smart contract that performs a DNS query to prevent URL redirection attacks. The immutable nature of the blockchain can help to identify phishing sites, providing a long-lasting solution to this problem. The effectiveness of the proposed system can be validated through a comparison with existing related works, demonstrating its security benefits.

Saravana Balaji B. [14]/2021 proposed a SQL injection recognition framework. The SQL injection causes by far most users to rely upon different kinds of informational indexes be they used in any contraptions, or defenseless against computerized risk. SQL Injection should be perhaps the greatest risk that an informational index poses to assembling applications concerning the web [6]. The vulnerability of a database to SQL injection renders all of the client's information in the database susceptible to theft or misuse. Although recent SQL injection models have been able to detect patterns they have seen before or those they are programmed to identify, they are unable to classify new patterns. The success of the system will depend on its ability to recognize and identify all types of injection methods. The model will handle all component extraction and analysis, with the user only required to input the text. Currently, traditional username and password-based authentication systems used for electronic applications are easily compromised. To address this issue, complex user authentication schemes have been developed using advancedencryption systems.

Derya Erhan. [11] /2020 proposed an automated 11-layered user authentication system for web applications in 2020, which offers a frictionless experience for users during login events. According to the capabilities of the profiler and authenticator, this system uses an acceptable user authentication method that incorporates four different login structures to guarantee the uniqueness of user identities. The authenticator processes the profiler's collection of user social data—such as login time, device location, browser, and information about web interactions—to generate a user profile.

According to the study's findings, this suggested technique is superior to alternative authentication methods for real-time web-based applications. Comparing the proposed approach to distinct and current authentication schemes for premium online apps, the latter saves memory consumption by 11%, speeds up response time by 7%, and reduces latency by 10%. Liu Yan.[8] /2020 proposed an improved system for user authentication in web applications that offers a seamless experience. The proposed system employs an automatic authentication procedure that the user starts when they log in. The system keeps user IDs at the login interface unique and suggests an effective authentication procedure. Four alternative login formats are used during the authentication process, which depends on the capabilities of the profiler and authenticator [7]. To build a comprehensive user profile using a 12-profiler, the profiler analyses user social data including login time, device location, browser, and web interaction details. According to the study's findings, the suggested system performs better for real-time web-based applications than alternative authentication techniques [9]. The suggested solution decreases delay by about 10%, improves response time by 7%, and uses 11% less memory when compared to distinct and existing authentication schemes for premium web-based apps.

III PROPOSED MODEL

The centralized nature of IoT cloud environments can affect the use of distributed services, including the security of IoT applications. As new IoT applications emerge in the Edge of Things (EoT) paradigm, novel security models, controls, and options are needed to be presented at the edge of the cloud. This paper proposes a distributed framework for detecting web attacks, specifically DDoS attacks, using deep learning techniques. The proposed approach uses a deception model that leverages novel deep learning architectures such as convolutional neural networks (CNNs) and natural language processing (NLP) models to detect various types of code injection attacks. Additionally, the framework includes delegated protection components, which are systems designed to counter web attacks when they occur. The proposed framework utilizes AI algorithms to preprocess and compare datasets, and to identify vulnerabilities in applications that may lead to attacks [8][17].

The web application attack detection model shown in Figure 1 is a comprehensive approach to protecting users from cyber threats. The model operates by analyzing the URLs visited by users and processing the data to generate a feature representation. This feature representation is then used in training and testing with intermediate vector classification using various algorithms to identify potentially malicious connections [10]. Once the training and testing phase is complete, the model generation component is used to block any connections that are flagged as malicious, predict suspicious scores based on the type and severity of the attack, and exclude websites where attacks are not predicted [11]. This predictive capability is particularly important in the fight against cybercrime since it allows for the identification and mitigation of potential threats before they can cause significant harm.

By leveraging the power of machine learning algorithms, the web application attack detection model can analyze vast amounts of data in real-time, allowing for quick and effective threat detection [12][13]. Moreover, the model can be updated regularly to ensure that it remains effective against emerging threats. Overall, the web application attack detection model represents a significant step forward in the ongoing battle to protect users from cyber threats and to promote a safer and more secure online environment for all.

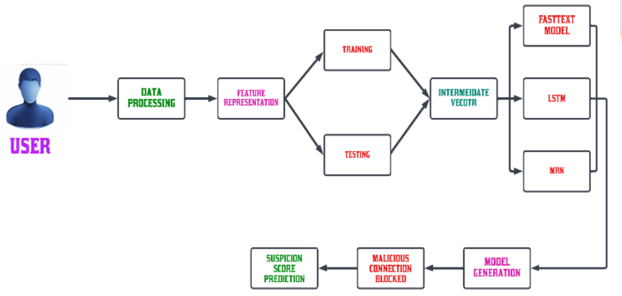

Figure 1. Web application Attack and detection of the attack

IV IMPLEMENTATION

The proposed model for web attack detection includes:

1. Feature selection
2. Multivariate correlation analysis
3. Decision-making module
4. Evaluation of attack detection

A. Feature selection module

This module extracts important elements from inbound traffic on the entrance network and redirects them to the internal network, where secure servers reside and are used to build traffic profiles over time. By focusing on relevant inbound traffic, and monitoring and analyzing the target network, we address the issue of identifying malicious activities. This also enables our detector to provide optimal protection, as the traffic profiles used by the detectors are tailored to the specific internal network with fewer connections.

B. Multivariate correlation analysis

Multivariate correlation analysis is a technique used to identify the relationship between two specific elements in each traffic record. This analysis uses the "Triangle Area Mapping" module to isolate these relationships. In the basic step, or by standardizing the traffic record in this step using the "Element Standardization" module, disruptions in network connections alter these relationships, which can be used as indicators to detect malicious activities. The analysis considers the removed relationships collectively, particularly the triangular areas stored in Triangle Area Maps, to override less important or standard elements. Essentially, this analysis helps to identify patterns and anomalies in traffic records to improve network security.

C. Decision-making module

The proposed model includes a Decision-making module that uses an anomaly-based detection mechanism to identify DoS attacks without prior knowledge of the attacks. This mechanism enhances the resilience of the detectors, making them difficult for attackers to evade. The module consists of two phases: the Training Phase and the Test Phase. During the Training Phase, profiles for different types of legitimate traffic records are generated and stored in a database. In the Test Phase, profiles for individual observed traffic records are generated and compared with the respective stored normal profiles in the Attack Detection module [14]. A threshold-based classifier is used by the Attack Detection module to distinguish between DoS attacks and genuine traffic. This method is more effective and challenging for attackers to get around since it avoids the time-consuming process of analyzing attacks and continually updating attack signatures. Attackers must produce attacks that correspond to the typical traffic profiles produced by a particular detection algorithm, which is a difficult task that calls for proficiency in the targeted algorithm [15][16].

D. Evaluation of attack detection

The evaluation process involves the utilization of 10% labeled data from the KDD Cup 99 dataset. This dataset comprises various types of legitimate traffic, including TCP, UDP, and ICMP traffic, as well as six different types of DoS attacks, such as Teardrop, Smurf, Pod, Neptune, Land, and Back attacks [17]. To streamline the evaluation process, records are initially filtered and then categorized into clusters based on their respective labels.

V RESULTS AND DISCUSSION

A. LSTM (Long Short-Term Memory)

A subset of recurrent neural networks called LSTM, or long short-term memory, are neural networks. Due to its effectiveness in processing sequential data, including speech recognition and natural language processing, it has gained popularity. Traditional RNNs struggle with the vanishing gradient problem, which makes it difficult for the network to learn long-term dependencies in sequential data. By utilizing a memory cell, LSTM networks can selectively forget or remember information over an extended length of time. The input gate, forget gate and output gate are the three gates that make up the memory cell and regulate the flow of information into and out of the cell. These gates enable the network to remember important information over extended periods while disregarding irrelevant information, allowing it to learn long-term dependencies in sequential data [15].

These gates use a combination of the sigmoid and tanh activation functions to selectively remember or forget information, making it easier for the network to capture long-term dependencies in the sequential data. This makes LSTM an ideal choice for various applications that involve processing sequential data.

LSTM models consist of several layers of these memory cells and gates. Each layer can be thought of as a separate processing step, where the inputs are processed sequentially and passed through the gates [18].

The architecture of LSTM has proven to be highly effective in modeling sequential data, and it has been mostly adopted in both academia and industry [19]. In recent years, various modifications and extensions of the basic LSTM architecture have been proposed, such as the Gated Recurrent Unit (GRU) and the Peephole LSTM, which have further improved the performance and capabilities of the model.

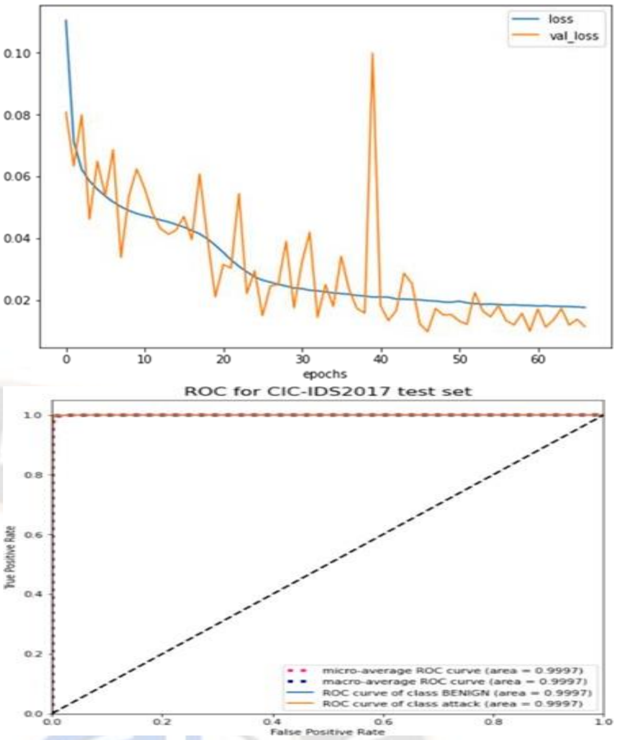

Figure 2. Graph for True and false positive rate using LSTM Algorithm

B. Random Forest Classifier

A machine-learning technique used for classification jobs is the Random Forest Classifier. It entails creating several decision trees, combining the output from each, and then making a final forecast. The decision tree algorithm builds a tree-like model that depicts a series of decisions and their associated results [20]. Random Forest takes this one step further by generating multiple decision trees and then combining their predictions to create a more reliable and accurate model.

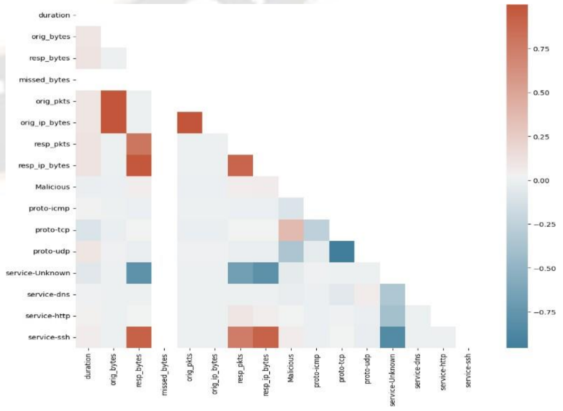

Figure 3. Graph using Random Forest Classifier

C. XGBoost Classifier

XGBoost (short for Extreme Gradient Boosting) is a commonly used machine learning algorithm that belongs

DOI: https://doi.org/10.17762/ijritcc.v11i10.8508
***Article Received:** 27 July 2023 **Revised:** 26 September 2023 **Accepted:** 15 October 2023*

to the boosting family of algorithms. It is an ensemble method that combines predictions from multiple individual models, typically decision trees, to generate a final prediction [21][22]. XGBoost is renowned for its scalability, speed, and accuracy and often outperforms other popular algorithms like random forests and gradient-boosting machines. The algorithm works by progressively adding decision trees to a model, with each tree trained on the errors or residuals of the previous trees [22]. Thus, each new tree aims to correct mistakes made by the previous trees, resulting in reduced bias and variance and improved overall performance.

Here are some of the key features of the XGBoost classifier:

Gradient boosting: XGBoost uses a gradient boosting technique to iteratively add decision trees to the model. Each new tree is trained on the residuals of the previous trees, which reduces the bias and variance of the model and leads to better performance.

Regularization: XGBoost offers several forms of regularization, including L1 and L2 regularization.

Feature importance ranking: XGBoost provides a feature importance ranking that can help identify which features are the most important for making predictions. This can be useful for feature selection and understanding the underlying patterns in the data.

Early stopping: XGBoost offers early stopping, which allows the algorithm to stop training once the performance on a validation set stops improving. This can help prevent overfitting and save time and computational resources.

Parallel processing: XGBoost supports parallel processing on a single machine or across multiple machines, which can significantly speed up training and improve scalability.

Handling missing values: XGBoost can handle missing values in the data and can automatically learn how to treat them based on the other available features.

Customizable loss functions: XGBoost allows users to define their custom loss functions, which can be useful for solving specific problems or optimizing for certain metrics.

It can be used in many applications and can deliver state-of-the-art performance.

Figure 4 represents the F Score graph generated using the XGBoost Classifier.

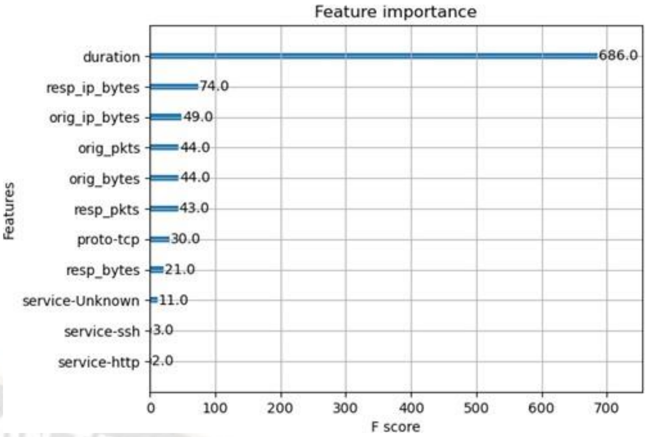

Figure 4. F Score Graph for XGBoost Classifier

D. MRN model

A type of neural network architecture that can handle several data kinds, including images, audio, and text, is the Multimodal Residual Network (MRN) model. It accomplishes this by fusing these various input modalities into one [24]. The ResNet model, which takes advantage of residual connections to train deep neural networks more, serves as the foundation for the MRN model. To help the network learn how to integrate the various modalities, residual connections between them are introduced in the MRN model [23]. This method has been effectively used for a variety of applications, including multimodal sentiment analysis, visual question answering, and image captioning. On several benchmark datasets for various tasks, the MRN model has proven to deliver state-of-the-art outcomes.

The MRN, LSTM, and CNN models are only a few of the deep-learning models used by the EDL-WADS system to identify web-based attacks [25]. To increase accuracy, true positive rate (TPR), and false positive rate (FPR), the system uses an ensemble classifier and thorough check, which aggregate the findings from the individual models. The EDL-WADS system is better at identifying web-based threats than the MRN paradigm.

This is demonstrated in Figure 5, where EDL-WADS surpasses each of the three independent models, proving its capacity to precisely blend the outputs of various deep-learning models.

Article Received: *27 July 2023* ***Revised:*** *26 September 2023* ***Accepted:*** *15 October 2023*

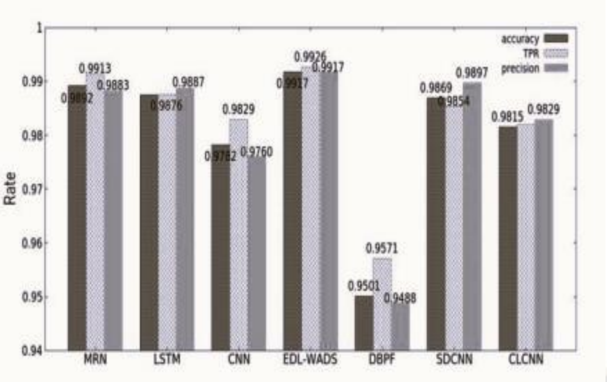

Figure 5. Comparison of EDL-WADS Accuracy, TPR, and Precision with other models.

The results indicate that MRN had the highest performance among the three models, meaning it had the highest accuracy or performed the best in terms of the specific metric being evaluated. In contrast, CNN had the worst performance among the three models.

VI CONCLUSION AND FUTURE DIRECTIONS

A new web attack detection system named EDL-WADS is specially designed for the Internet of Things. The system comprises four modules, which include a feature learning module, three deep learning models, an all-encompassing decision module, and a tweaking and refreshing module. The feature learning module creates representations of URL requests, while the second module uses three different deep learning models to produce a variety of representations of URL requests. The final decision is then made by the comprehensive decision module using a group classifier after these representations have been combined by the feature learning module. The fourth module is a tweaking and refreshing module that dynamically aligns and updates the three deep learning models. The system is evaluated using different datasets to assess its effectiveness in detecting web attacks. The suggested model also looks into ways to enhance EDL-WADS's capacity to identify new varieties of web attacks and boost the system's efficiency by examining various deep-learning models.

REFERENCES

[1] M. Lin, C. Chiu, Y. Lee, and H. Pao, “Malicious URL filtering—A big data application,” in Proc. IEEE Int. Conf. Big Data, 2013, pp. 589–596.

[2] D. Kar, S. Panigrahi, and S. Sundararajan, “SQLiDDS: SQL injection detection using query transformation and document similarity,” in Proc. Int. Conf. Distrib. Comput. Internet Technol., 2015, pp. 377–390.

[3] A. Le, A. Markopoulou, and M. Faloutsos, “PhishDef: URL names say it all,” in Proc. IEEE INFOCOM, 2011, pp. 191–195.

[4] Revathy, S. ., & Priya, S. S. . (2023). Enhancing the Efficiency of Attack Detection System Using Feature Selection and Feature Discretization Methods. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 156–160. https://doi.org/10.17762/ijritcc.v11i4s.6322

[5] P. Bisht, P. Madhusudan, and V. N. Venkatakrishnan, “Dynamic candidate evaluations for automatic prevention of SQL injection attacks,” ACM Trans. Inf. Syst. Secure., vol. 13, no. 2, pp. 398–404, 2010.

[6] C. Luo, S. Su, and Y. Sun, “A convolution-based system for malicious URL requests detection,” Comput. Mater. Continua, vol. 61, no. 3, pp. 399–411, 2019.

[7] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, “Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems,” IEEE Internet Things J., vol. 7, no. 7, pp. 6266–6278, Jul. 2020.

[8] Y. H. Hwang, “IoT security & privacy: Threats and challenges,” in Proc. 1st Acm Workshop on IoT Privacy Trust and Security, 2015, p. 1.

[9] A. Jamdagni, Z. Tan, and X. He, “RePIDS: A multi-tier real-time payload-based intrusion detection system,” Comput. Netw., vol. 57, no. 3, pp. 811–824, 2013.

[10] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, “A system for denial-of-service attack detection based on multivariate correlation analysis,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 447–456, Feb. 2014.

[11] Derya erhan (Member, IEEE), and Emin anari, “Hybrid DDoS Detection Framework Using Matching Pursuit Algorithm,” Comput. Secure., vol. 60, pp. 206–225, 2020.

[12] Zhihong Tian, Chaochao Luo, Jing Qiu, Xiaojiang Du, Mohsen Guizani, “A Distributed Deep Learning System for Web Attack Detection on Edge Devices,” 2020, arXiv:1702.08568.

[13] Rashidah F. Olanrewaju, Burhan Ul Islam Khan, Malik Arman Morshidi, Farhat Anwar, and Miss Laiha Binti Mat Kiah, “A Frictionless and Secure User Authentication in Web-Based Premium Applications.,” in Proc. IEEE 14th Int. Colloq. Signal Process. Its Appl., 2021, pp. 103–106.

[14] Jothi K R, Saravana Balaji B, Nishant Pandey, Pradyumn Beriwal, Abhinandan Amarajan “An Efficient SQL Injection Detection System Using Deep Learning,” in Proc. VI Int. Conf. Netw., Commun. Comput., 2021, pp. 80–85.

[15] Wen-Bin Hsieh, Jenq-Shiou Leu, and Jun-Ichi Takada, “Use Chains to Block DNS Attacks: A Trusty Blockchain-based Domain Name System.”, vol. 7, no. 6, pp.4682–4696, Jun. 2020.

[16] J. Ma, L. K. Saul, and S. Savage, “Beyond blacklists: Learning to detect malicious websites from suspicious URLs,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2009, pp. 1245–1254.

[17] Hariharan, B., Sathya Priya, S., “Recent Web Application Attacks’ Impacts, and Detection Techniques–A Detailed Survey” Lecture notes in Networks and Systems 2023, 673 LNNS, pp. 863–871

[18] F. Yong, P. Jiayi, L. Liang, and H. Cheng, “WOVSQLI: Detection of SQL injection behaviors using word vector and LSTM,” in Proc. 2nd Int. Conf. Cryptography, Secure. Privacy, 2018, pp. 170–174.

[19] B. Martin, M. Brown, A. Paller and D. Kirby, "CWE/SANS top 25 most dangerous software errors," The MITRE Corporation, 2011. Michele Bugliesi, Stefano Calzavara, Riccardo Focardi, Formal methods for web security, Journal of Logical and Algebraic Methods in Programming, Volume 87, 2017, Pages 110-126.

[20] Michele Bugliesi, Stefano Calzavara, Riccardo Focardi, "Formal methods for web security," Journal of Logical and Algebraic Methods in Programming, Volume 87, 2017, Pages 110-126.

[21] M. K. Gupta, M. C. Govil and G. Singh, "Static analysis approaches to detect SQL injection and cross-site scripting vulnerabilities in web applications: A survey," International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), Jaipur, 2014, pp. 1-5.

[22] N. Antunes and M. Vieira, "Detecting SQL Injection Vulnerabilities in Web Services," 2009 Fourth Latin-American Symposium on Dependable Computing, Joao Pessoa, 2009, pp. 17-24.

[23] Z. Ghanbari, Y. Rahmani, H. Ghaffarian and M. H. Ahmadzadegan, "Comparative approach to web application firewalls," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015, pp. 808-812.

[24] A. Alzahrani, A. Alqazzaz, Y. Zhu, H. Fu, and N. Almashfi, "Web Application Security Tools Analysis," 2017 IEEE 3rd International Conference on Big Data Security on Cloud (Big Data Security), IEEE International Conference on High Performance and Smart Computing (hpsc), and IEEE international conference on intelligent data and security (ids), Beijing, 2017, pp. 237-242.

[25] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufman, 2001.

[26] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," IEEE Trans. Ind. Informat., vol. 16, no. 4, pp. 2659–2666, Apr. 2020.

[27] I. Lee, S. Jeong, and S. Yeo, "A novel method for SQL injection attack detection based on removing SQL query attribute values," Math. Comput. Modelling, vol. 55, no. 1-2,pp. 58–68, 2012.