IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

WEB ATTACK DETECTION ANALYSIS SYSTEM USING CLASSIFIER AND DEEP LEARNING

Sanjaysithon.s
Computer Science Engineering, Madha Engineering College,Tamilnadu ,India.
Computer Science Engineering, Madha Engineering College,Tamilnadu ,India.

Abstract. Web application assaults are an inexorably significant area in data security and computerized criminology. It has been seen that assailants are cultivating the ability to sidestep security controls and send off an enormous number of refined assaults. A few endeavors have been made to address these assaults using an extensive variety of innovations, and one of the most noteworthy difficulties is compellingly answering new and obscure assaults. Web application attacks are on the rise, and reviews show they are one of the best explanations behind data breaches. As these attacks become more common, organizations must understand what they are up against, how to mitigate risks, and how to protect against them. This survey intends to investigate various types of web application assaults, an examination of the most successful types of web assaults, and various investigations based on web application assaults and discovery methods. It was expected to add to this growing field of concentration by exploring more sharp and viable strategies for web application assault identification and be helpful for concentrations in view of web assaults.

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

1 Introduction

Web applications expect a huge job in people's everyday schedules, especially when people start to transfer their applications and their data to the cloud. Web applications are engaging assault targets because of their shared trait and the way that they store a lot of confidential client data. In this regard, it is critical to protect web applications from interference. Among many shortcomings, Distributed Denial of Service (DDoS) - related shortcomings, which are exploited by sending purposefully arranged requests, involve the greatest part. According to injections, for instance, Structured Query Language (SQL) injection and Cross-site scripting are recorded as the first and third most fundamental web application security bets, separately [1]. All around, there are two methods for managing and recognizing the attacks referred to beforehand.

The first is the imprint-based method, which is to look for unequivocal attack plans in requests; the second is the anomaly-based strategy, which is to spread out conventional sales profiles with the objective that unusual sales can be isolated from normal ones. The imprint-based system is embraced more fiercely than the peculiarity-based procedure because ordinarily the imprint-based one has a lower duplicity rate and achieves higher precision.

The Web Application Firewall (WAF) contains a massive set of rules that can distinguish between SQL Injection and cross-site scripting. Whatever the case may be, the standard-based strategy is risky at this point. It is, first and foremost, basically as extraordinary as the level of the standard set, and that suggests it is unequipped for recognizing attacks that are not in its imprint dataset [2]. Furthermore, bypassing WAF should be possible by subscribing to existing poisonous requests or encoding themselves on various events [3]. Thirdly, a very enormous assault design set or demand with long lengths consumes heaps of processing assets to complete the example examination [4]. The Web is a critical piece of a huge package of strategic policies your affiliation partakes in each day. It is the repository of information and the home of cloud-based motorized limits. It holds the data that clients purposely give through happy *association frameworks, shopping cases, login fields, and request and submit structures.*

In any case, as unfathomable and good as these endeavors have all the earmarks of being, they are altogether feeble against web application assaults from cyber criminals [5]. Understanding how web applications work and focusing on their most commonly exploited flaws can benefit you and your security by joining forces and executing strategies. It will limit the possible results that your business and clients will experience in the event of an information

1.1 Web-based Attacks

Web applications truly do raise various security concerns arising from inappropriate coding. Serious flaws or weaknesses allow criminals to gain immediate and unrestricted access to data sets and manipulate sensitive information; this is known as a web application attack. A considerable number of these data sets contain significant data (for example, individual information and monetary subtleties), making them regular targets of assaults. Even though such destructive incidents (frequently committed by alleged content children) as destroying corporate sites are still common, aggressors now prefer accessing the sensitive information residing on the data set server because of the enormous changes in selling the consequences of data breaks. In the system portrayed above, it is not difficult to perceive how a lawbreaker can rapidly get to the information dwelling on the data set through a portion of imagination and, with karma, carelessness, or human blunder, prompting weaknesses in the web applications [7].

As expressed, sites rely on data sets to convey the expected data to guests. If your web applications are not secure, for example, helpless against at least one of the various types of hacking methods, then your entire set of sensitive data is at risk of a web application assault. SQL Injection attacks, which target information bases directly, are still the most well-known and dangerous type of vulnerability [8]. Different aggressors might infuse malevolent code, utilizing the client contribution of weak web applications to deceive clients and divert them towards phishing locales. This is called as "cross-site scripting (XSS) attack" and might be utilized even though web servers and information base engines contain no weaknesses themselves. It is frequently used in conjunction with other attack vectors, such as social engineering attacks. There are numerous different kinds of normal goings-on, like registry crossing, and neighborhood record incorporation, and that's only the tip of the iceberg.

Ongoing exploration shows that 75% of digital assaults are finished at the web application .

1.2 Web Applications Working

Web applications manage their business by first examining a substance-enlightening assortment and conveying a web record as exhibited by the client's nuances.

The data is familiar so it is open with all activities, which run each fulfilled and make the record both justifiable and dynamic [10].

Web applications guessing that essentially, no work ought to be available on the client's end can be bought by affiliations second or re-attempted to meet a business' interesting decisions.

1.3 Online Attacks

Precisely when crooks exploit weaknesses in coding to get enough close to a server or enlightening file, such modernized mutilation gamblers are known as application-layer assaults. Clients acknowledge that the delicate individual data they uncover on your site will be shielded privately [11].

Impedance as online assaults can recommend that their Visa, Government retirement partner, or clinical data could become public, prompting possibly grave results [12].

Web applications are especially vulnerable to hacking. Since these applications should be uninhibitedly open, they can't be protected behind firewalls or got from taking a chance with Secure Sockets Layer (SSL).

A gigantic number of these endeavors approach, either straightforwardly or by implication, to especially certain client information.

Designers make it their business to glance through deficiencies so this data can be taken or rerouted. Endeavoring to obstruct web application assaults ought to be a fundamental necessity for your IT security pack.

1.4 Types of Web Assaults

Yet the techniques of cybercriminals are ceaselessly creating, their secret attack strategies remain decently consistent. Coming up next are presumably the most notable:

- Cross-site scripting (XSS)

That integrates an assailant moving a piece of noxious substance code onto your site that can then be utilized to take information or perform different sorts of insidiousness. However, this system is all around unsophisticated, it remains incredibly average and can genuinely sting [13].

- SQL Injection (SQLI)

This happens when an Attacker submits destructive code into an information structure. Expecting your framework's dismissal to clean this data, it very well may be submitted into the enlightening assortment, progressing, killing, or uncovering information to the aggressor [14].

- Way crossing

In this way coming about given ill-advised security of information that has been inputted, these webserver assaults consolidate embedding plans into the webserver natural pecking order that permits fomenters to gain

client agitators gain client authorizations, instructive assortments, game-plan records, and different types of data put away on hard drives [15].

• Close by Record Thought

This sensibly magnificent assault strategy integrates obliging the web application to execute a record tracked down somewhere else on the framework [16].

• Distributed denial of service (DDoS) attacks

Such harmful events happen when an attacker attacks the server with requests. A significant part of the time, developers use an association of compromised computers or bots to mount this aggression. Such exercises cripple your server and hold real visitors back from getting to your organization [17].

Though fomenters don't generally mull over these strategies, they regularly use them to "possess" your modernized systems, leaving you frail against other malware and violations.

S.NO	EVENT	ATTACK	IMPACTS
1.	Kaseya Ransomware Attack	Malware attack	This assault impacted their clients and many of their companies.
2.	Cisco vulnerability	SQL Injection Attack	The flaw allowed aggressors to gain shell access to the frameworks to which the permit chief was dispatched.
3.	Amazon DDoS Attack	DDoS Attack	The organization experienced one of the biggest DDoS assaults ever.
4.	British Airways Attack	Cross-Site Scripting (XSS)	They prevailed with regards to performing Credit card skimming on 380,000 booking exchanges before the break was found.
5.	Twitter Celebrities Attack	Social engineering	Many notable leaders' and Celebrity's records were hacked

1.5 Protecting against the Site Assaults

An affiliation's capacity to utilize online assets to catch and store client information appreciates many advantages, yet it additionally makes the way for undermining aggressors. Luckily, there are strategies you can use to give assessment and affirmation to your site and its mysterious servers and educational records. They solidify the going with:

• Computerized weakness filtering and security testing. These endeavors assist you with finding, isolating, and diminishing weaknesses, as frequently as conceivable before affirmed assaults happen. Setting resources into these preventive measures is a financially sharp strategy for decreasing the probability that weaknesses will change into electronic fiascoes [18].

• Web Application Firewalls (WAFs). These work on the application layer and use rules and information about known break philosophies to limit authorization to applications [19]. Since they can get to all layers and shows, WAFs can make genuine progress guards concerning safeguarding assets from assault.

• Secure Development Testing (SDT). This bearing is anticipated by all security partners, including analyzers, planners, modelers, and bosses. It gives data about the freshest assault vectors. It helps the gathering in fanning out a model and empowering a rational, unique strategy for overseeing and forestalling site assaults and limiting the results of breaks that can't be halted. The negation, control, and equilibrium of web application assaults are typical work [20]. Mounting a multi-pronged guard containing improvement, mechanized adventures, and human strength will permit you to screen, independently, perceive, and kill dangers of different sorts rapidly and really.

TABLE I Contains the rundown of occasions in light of Web Application assaults and their significant effects of them are addressed in the table.

2 Related Study

Following the breakthrough in artificial reasoning innovation, scientists in the field of organizational security have widely embraced profound learning. A ton of exploration work has been done centered around web assault identification given profound learning. It appears that security location innovation based on human consciousness is rapidly becoming a critical course strategy. Strategies for web assault location in light of profound learning are driven by extensive information examination [21]. Along these lines, profound learning models can investigate inputs by extricating these helpful elements and gain examples from these highlights through iterative preparation. Based on deep learning procedures, web assault recognition strategies improve location execution indefinitely. As of now, the commitments of existing related works are for the most

part reflected in two Points: First, is the technique applied to break down Uniform Resource Locators (URL) demands and change them into vectors, and the other is the profound learning model used to find out and identify web assaults [22]. We sum up these three sorts of techniques for URL investigation below.

1) Measurable qualities in light of coordinating and counting typical words or accentuations from crude traffic are most generally used to address URL demands, for example, the length of URL demands, the strange expression of accentuations in the solicitations, the kinds of odd words, and the number of boundaries.

2) Addressing URL demands in light of a customary semantic and syntactic examination of crude information has become a well-known technique in the field of web assault discovery. Highlights removed from semantic and syntactic examination contain the profundity of the linguistic structure tree, the number of roots in the grammar tree, the number of leaf hubs in the punctuation tree, and so on.

3) The strategy for examining URL demands and changing them into vectors naturally shows its predominant capacity for addressing URL demands precisely. It has turned into the best-in-class technique in the field of web assault recognition.

3 Survey Of Exısting Work

In Existing Framework, the web application assaults might include security misconfigurations, broken confirmation and meeting the board, or different issues. The absolute most perilous and predominant web application assaults, nonetheless, exploit weaknesses related to ill-advised approval or sifting of untrusted inputs, bringing about the infusion of vindictive content or space-explicit language code. Assailants appear to track down better approaches to acquaint vindictive code with applications utilizing different dialects and strategies. In the meantime, during the last 10 years, there have been various components intended to identify more sorts of such goes after IoT web applications [23]. In like manner, interruption recognition frameworks, like grunt and WAF are utilized to safeguard against web attacks, yet they are right now feeble because most WAFs depend upon ordinary verbalization-based channels delivered utilizing acknowledged assault engravings, and they require a lot of master planning. Critical learning has been executed in many areas with clear accomplishments [24].

For Instance, critical learning can be utilized in altered interpretation machines to develop endurance. Meanwhile, huge learning has been applied to sort out confirmation in an evaluation by its capacity to separate and self-learn. Perceiving web assaults from regular clients and aggressors inside a DDOS attack is an attempt, and there are four basic issues.

In TABLE II Characterization of different works based on Web attack detection and prevention system are represented in the table format below.

Derya Erhan et al. [11]/2020 proposed a hybrid DDoS discovery structure model, which employs the word reference delivered from the organization's traffic boundaries via the K SVD calculation.

Zhihong Tian et al. [12]/2020 proposed a distributed learning framework model. They propose a web assault discovery framework that exploits breaking down URLs utilizing Natural language processing (NLP) and Convolutional neural networks (CNN). This framework is intended to recognize web assaults and send alerts. Numerous simultaneous profound models are utilized to improve the framework's security and the accommodation of refreshing and are approved by the Hypertext Transfer Protocol (HTTP) Dataset, the FWAF, and the HttpParams Dataset.

Rashidah F. Olanrewaju et al. [13]/2021 recommended a framework called the Frictionless and Secure Client Confirmation Framework, they proposed a protected client approval part for web applications with a frictionless encounter utilizing the Solid internet-based exchange calculation. A mechanized approval plot is arranged considering client-led login occasions. The uniqueness of the client character is endorsed in the proposed structure at the login interface, trailed by a recommendation for a fitting client affirmation process. The insistence association is carried out in four distinct login parts, the capacities of which are determined by the profiler and authenticator.

Jothi K. R. et al. [14] /2021 proposed a SQL Injection Location Framework model for detecting SQL Injection by detecting patterns
in data. The upside of this structure is that it will need to recognize all kinds of infusion methods. All the component extraction and choice will be done by the real model. The dataset utilized in this model is Lib-infusion.

In the last review, Wen-Container Hsieh et al. [15]/2022 proposed a framework named "Blockchain-based DNS Framework." They proposed a clever component for checking sites utilizing blockchain innovation.

This instrument won't add any heap to clients and gives sealed capabilities given the attributes of the blockchain, and this component is safer than different models that utilize the Raft consensus algorithm

4. Deep Learning

.

In EDL-WADS, this section discusses the key module for detecting web attacks .According to the feature vectors provided in the model of feature learning, we utilized three deep learning models for classification, they are the MRN model, LSTM model, and CNN model, respectively. Particularly, there are two
main reasons for we used three deep learning models instead of two or more deep learning models. First, other models will be
clearly affected if one model is compromised when there are two models. Second, more deep learning models will lead to more
cost of computing source and time.

1.MRN LAYER:

MRN is a new structure of a computing
unit, which has been improved on the bias of Residual Network (Res Net), proposed in the previous work The structure of MRN is illustrated in and the equation of the unit is described as follows:
H (**x**) = pool (αF1 (**x**) + βF2 (**x**) + γF3 (**x**)) (1)where α, β, and γ are to be optimized with all parameters of
the model in the training phase. In MRN, F2(**x**) and F3(**x**) are designed to analyse URL requests in a semantic way, and they are able to extract useful semantic features, F1(**x**) is a fast-track that retains all statistic information which includes
information dropped by F2(**x**) and F3(**x**). The procedure of MRN is explained. We utilized a two-channel matrix for inputting into the MRN model as researchers do for pictures in the field of the computer version. As shown in , the URL requests are represented by the matrix in two channels, one is composed of CBOW vectors and the other is composed
of TF-IDF vectors, so that EDL-WADS can effectively take advantages of the MRN units .semantic vectors and statistic vectors generated in the feature learning module. The part of feature extraction is composed of four parallel MRN layers that are referred as the structure of the "Inception." By using multiple MRN layers stacked, different scales of semantic and statistic features are increased while the depth of the model is still shallow. Through the concatenation and the flatten layer, all features from MRN layers will be concatenated and flattened, then sent to the classification module. The classifier is composed of three dense layers: full connected layer, batch normalization layer, and sigmoid layer. There is a dropout layer after each dense layer, which is omitted in .We provide all parameters.FIG :1

2)LSTM:

LSTM is the most widely used deep learning model in research on NLP. The URL requests are essentiallytexts. Therefore, it is common that we took the detection task as a text classification and designed a LSTM model for the task. The structure of the LSTM model is shown in and the list of parameters are provided in . For the LSTM model, the semantic and statistic vectors generated in
method one of feature representation are utilized as input. We PARAMETERS OF CNN MODEL concatenate two types of k dimensional vectors and used the
combined 2k-dimension vectors as the input of the LSTM model. Specially, the LSTM model consists of LSTM layers, an MLP module and an output layer. The core of the LSTM Model is the LSTM layers, which is utilized to extract features from input vectors. The MLP model is used to map the output of LSTM layers and classify them. Finally, a sigmoid layer is designed to normalize the classification probability and make the final decision.
FIG :2

3 CNN MODEL:

In EDL-WADS, we designed a CNN model that uses a feature representation method based on the embedding layer. The structure of the CNN model is illustrated in with parameters provided in The input of this CNN model is a sequential vector of normalized URL requests ,which is processed by the approach described in method one of the feature representation. The embedding layer is utilized to
convert these input words into vectors and propagate them to the CNN layers. Similarly, we used the same structure as the
MRN model to stack the CNN layers, and the three different CNN neural networks can increase the scale of the features. The
concatenation and flatten layer will concatenate these features and propagate them into the classification model composed of
two dense layers: a batch normalization layer and a sigmoid layer.FIG :3

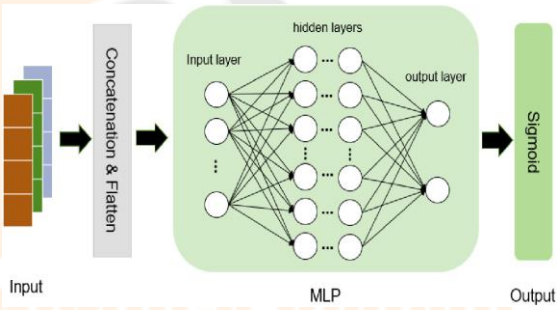


FIG 2: LSTM model

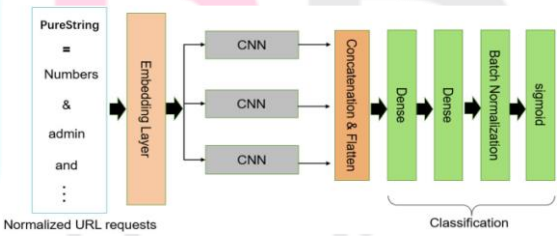


FIG 3:CNN MODEL

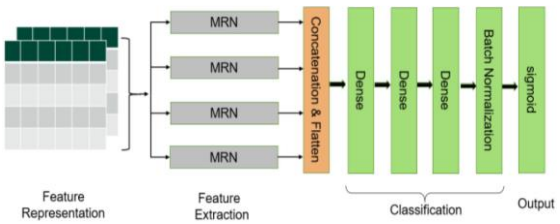
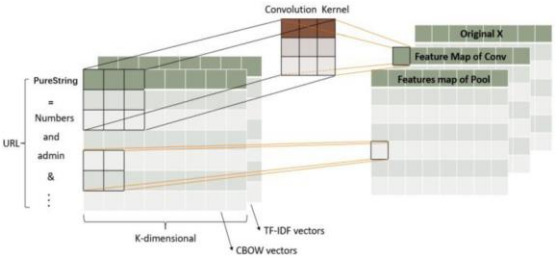


FIG 1: MRR MODEL



Process of MRN

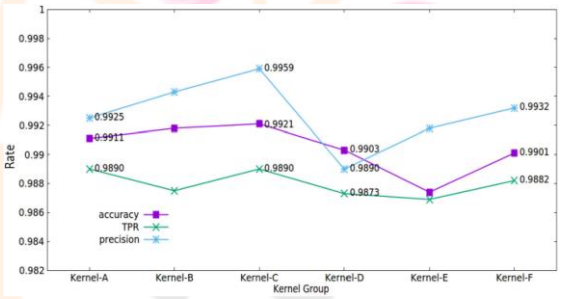
3)comprehensive decision:

The comprehensive decision module is designed to combinethose parallel results from multiple deep learning models and obtain the final decision for detection. Three deep learning models are used for classification, and each of them outputs an intermediate vector after its computing. To get the best predictive performance, we perform a comprehensive check and use an ensemble classifier. The comprehensive check is to calculate a vector Vr that denotes the reliability of results of every deep learning model, as described in Algorithm 1. First, we get Vm that represents the average of immediate vectors. Second, for each immediate vector Vi, a Euclidean distance between it and Vm is calculated. Finally, we obtain a reliable vector Vr, according to the Euclidean distance, for every immediate vector Vi. The Euclidean distance shows the immediate vectors' reliability according to the normal fluctuation range of results of every model. Specifically, if the Euclidean distance is less than threshold ε , the immediate vector is considered as reliable and the value is then set to 1, or the immediate vector is considered as unreliable. Consequently, its value is set to 0.In EDL-WADS, we used an MLP model as an ensemble classifier to combine all intermediate vectors and make the final
decision. The structure of the ensemble classifier is depicted in The inputs of the model are vectors calculated using immediate vector Vi and reliability vector Vr. The concatenation and flatten layer will merge these vectors into one and propagate
it to the MLP model. The MLP model and sigmoid layer

4) Fine-Tuning and updates

Because of the complexity of real-world network environment and the diversity of web attacks, deep learning models in the intrusion detection system (IDS) needs regular updates. As is shown in , in order to improve the robustness and reliability of EDL-WADS, we integrate in it a feedback mechanism to fine-tune and update the system. In the fine-tuning and updates module, all raw URL requests, normalized data, and detection results are recorded in a database to facilitate further analysis by the security experts. Moreover, EDL-WADS is designed to take advantage of experts' analysis to fine-tune deep learning models

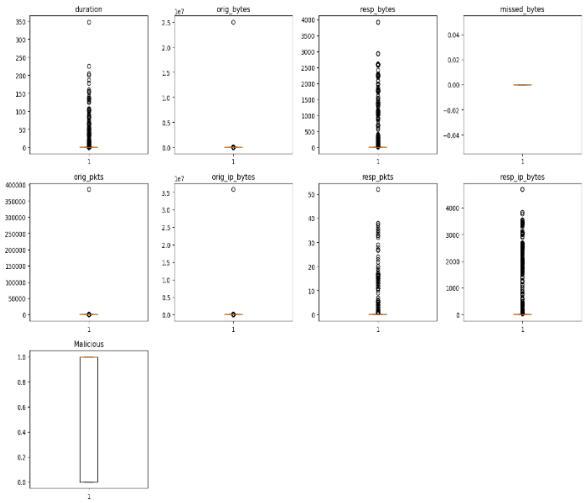
Algorithm: Comprehensive Check.
Input: Intermediate vectors from MRN model V_1, Intermediate vectors from LSTM model V_2, Intermediate vectors from CNN model V_3, The average of immediate vectors V_m,Thresholds ε.
Output**:** A vector V_r represnets the reliability of all intermediate vectors.
1: for each V_i, i ∈ {1, 2, 3} do
2: sum = 0
3: for $V_i[k]$, k ∈ dim(V_i) do
4: dif =V_i [k]- V_m[i]
5: sum = sum+dif2
6: end for
7: if sum ≤ ε then
8: V_r[i] = 1
9: else
10: V_r[i] = 0
11: end if
12: end for
in the training phase and update these models incrementally in order to discover new web attacks. When one of the three models is being fine-tuned and updated, the remaining two other models continue to work. This ensures the fine-tuning and update on one model makes very little negative impact on the overall detection making. Most importantly, in terms of the reliability our proposed system is fault tolerant, namely, when one deep
learning model is under attack (e.g., attacks described , two other deep learning models are still active and making
decisions jointly with very little performance degradation



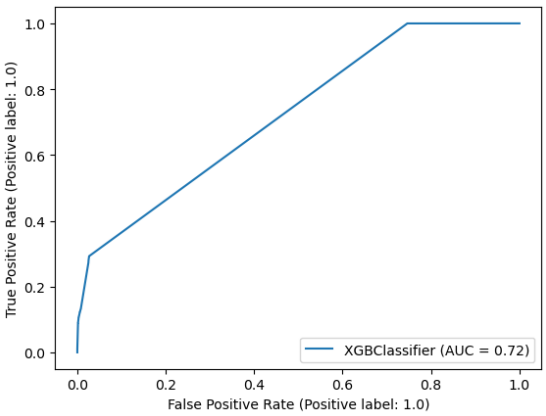
Experimental results for kernels in MRN.

5 Research Gaps

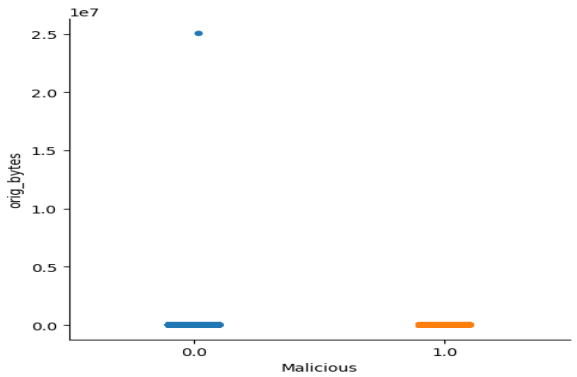
Various assaults show various marks in their URLs, and in this manner, they highlight that determination is difficult.Fig :-1
The beneficial execution of frictionless confirmation procedures in electronic applications is not focused on specific works.
Existing overviews are more application-explicit and do not focus on the whole scope of safety and security in distributed computing frameworks and web administration organizations.Fig :-2
The current EDL-Web Attacks Detection system framework can only recognize SQLI and Cross-site scripting (XSS), not different kinds of attacks .



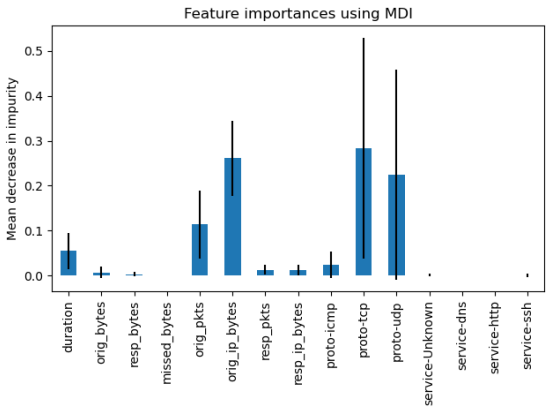
Suspension for score



B. False postive rate



malicous



A. Feature importances using mdi

5 *Conclusion And Future Directions*

It is extremely difficult to classify all types of web application assaults and the most common types of assaults in the ongoing framework. Numerous examiners are carrying out new calculations with less misleading positive rates that identify a wide range of web assaults, yet it requires numerous years to design the best framework for forestalling web-based assaults.

In the impending future, attackers are getting more careful and will like to produce blended traffic to confound the current safeguard systems. Thus, the necessity is to concoct a fiery arrangement that ought to be sufficiently nonexclusive to recognize all Web application assaults and forestall them.

The ongoing EDL-WADS framework can recognize SQL injection and cross-site scripting, but not different kinds of assaults. Future investigations ought to be centered around the identification of different kinds of assaults, and feature selection can be centered more around those reviews.

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, pp. 589–596 (2013)
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., pp. 377–390 (2015)
3. A. Le, A. Markopoulou, and M. Faloutsos.: PhishDef: URL names say it all, in Proc. IEEE INFOCOM, pp. 191–195 (2011)
4. 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 (2020)
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 (2020)
8. Y. H. Hwang.: IoT security & privacy: Threats and challenges, in Proc. 1st Acm Workshop on IoT Privacy Trust and Security (2015)
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, (2014)
11. Derya erhan (Member, IEEE), and Emin anari.: Hybrid DDoS Detection Frame work Using Matching Pursuit Algorithm, Comput. Secure., vol. 60, pp. 206–225 (2020)
12. Zhihong Tian, Chaochao Luo, Jing Qiu, Xiaojiang Du, Mohsen Guizani.: A Dis tributed Deep Learning System for Web Attack Detection on Edge Devices, arXiv:1702.08568 (2020)
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., pp. 103–106 (2021)
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 (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, pp. 1245–1254 (2009)
17. 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)
18. F. Yong, P. Jiayi, L. Liang, and H. Cheng.: WOVSQLI: Detection of SQL inject tion behaviors using word vector and LSTM, in Proc. 2nd Int. Conf. Cryptography, Secure. Privacy, pp. 170–174 (2018)

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