International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

Enhanced Study of Deep Learning Algorithms for Web Vulnerability Scanner

Dr. Kalyan Devappa Bamane[1], Dr. Abhijit Janardan Patankar[2], Mrs.Priyanka Gupta[3], Dr.Ratnaraja Kumar Jambi[4], Ms. Nitisha Rajgure[5]

[1]Associate Professor
Department. of Information Technology
D Y Patil College of Engineering
Pune, India
kdbamane@dypcoeakurdi.ac.in

[2]Associate Professor
Department. of Information Technology
D Y Patil College of Engineering
Pune, India
ajpatankar@dypcoeakurdi.ac.in

[3]Assistant Professor
Department. of Information Technology
D Y Patil College of Engineering
Pune, India
pgupta@dypcoeakurdi.ac.in

[4]Associate Professor
Department of Computer Engineering
Genba Sopanrao Moze College of Engineering , Pune ,India
ratnaraj.jambi@gmail.com

[5]Assistant Professor
Department of Computer Engineering
Zeal Eductaion Society ,Pune ,India
nitisha.rajgure@zealeducation.com

Abstract - The detection of online vulnerabilities is the most important task for network security. In this paper, deep learning methodologies for dealing with tough or complicated challenges are investigated using convolutional neural networks, long-short-term memory, and generative adversarial networks.Experimental results demonstrate that deep learning approaches can significantly outperform standard methods when compared to them. In addition, we examine the various aspects that affect performance. This work can provide researchers with useful direction when designing network architecture and parameters for identifying web attacks.

Keywords-SQL Injection, Cross Site Scripting, Generative Adversarial Network, LSTM, CNN, Deep Learning.

I. INTRODUCTION

Web applications have become increasingly popular in recent years and are now seen as the primary platform for a wide range of business activities, including financial banking, e-commerce, infotainment, and administrative reforms. However, the rapid development of web technology and the widespread use of the internet have created new challenges for online security. The increased accessibility of online information and services has also led to an unprecedented increase in the number and sophistication of security threats targeting web applications.

Web application vulnerabilities and exploits can be found using deep learning, a potent machine learning technology. To learn the patterns of regular and anomalous online traffic, deep learning models can be trained on massive datasets of well-known threats and vulnerabilities. This enables them to recognise potential assaults and vulnerabilities in real time, even if signature-based intrusion detection systems (IDSs) are not yet aware of them. A number of web threats, including SQL injection, cross-site scripting, and denial-of-service attacks, have been shown to be successfully detected by deep learning. Insecure coding techniques and out-of-date software are examples of vulnerabilities in online applications that can be found using this technique. Although the application of deep learning for web attack detection is still in its infancy, it is a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

promising new technique that has the potential to greatly enhance the security of web applications.

II. RELATED WORK

However, this industry has seen the completion of other noteworthy investigations. Our study entails a methodical investigation of the application of Deep Learning for attack and vulnerability detection, primarily on Web applications. For instance, highlighted a number of mitigation strategies and provided a thorough analysis of the identification and mitigation of online vulnerabilities. However, they neglected Deep Learning techniques in favor of classic Machine Learning-based algorithms for web vulnerability detection.

Related studies likeexamined the use of machine learning or deep learning in the context of cybersecurity issues in general but did not particularly focus on web application security. Similar to how Refs. did not explore Deep Learning-based methods for identifying web vulnerabilities and did not follow the Systematic Literature Review (SLR) procedure, they concentrated on the classification of web vulnerabilities and countermeasures.

Deep Learning-based techniques for online vulnerability identification may not be directly addressed by an SLR on web services vulnerability and security, even if it were performed. The latest machine learning and deep learning-based methods for identifying XSS assaults, a particular kind of online vulnerability, are, on the other hand, highlighted in a recent survey.Therefore, although our study stands out as the systematic exploration of Deep Learning for vulnerability and attack detection on Web applications, it is important to acknowledge the existence of other partial surveys that have touched upon various aspects of web vulnerability detection and prevention, albeit with different emphases and scopes.

III. CLASSIFICATION OF DEEP LEARNING TECHNIQUES.

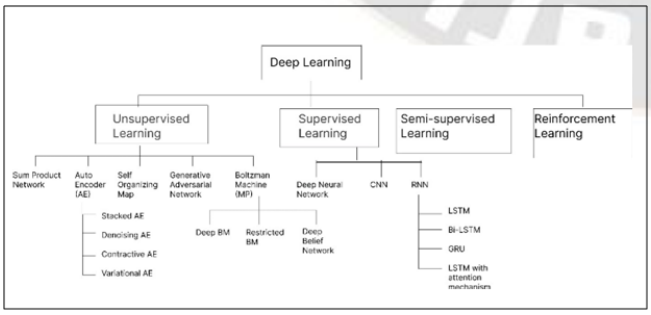

Fig 1. Classification of Deep Learning Techniques

In this work, we are especially interested in exploring CNN, LSTM Deep Learning, and Generative Adversarial Network algorithms for detecting Web vulnerabilities. In order to construct new and potent DL-based online vulnerabilities detection models, it is hoped that researchers and practitioners interested in applying deep learning for this purpose would find relevant information.

IV. DATASET

The selection of the dataset is a crucial factor in the developing of the effective detection models in deep learning technologies. Therefore, the purpose of this inquiry is to review and debate the dataset constraints that are frequently utilized in web vulnerability detection. We divided the research into two categories based on the dataset type: (i) public datasets with free and open access and (ii) private datasets with closed access. We discovered that some studies integrate multiple public datasets or even run experiments using both private and public databases simultaneously.

Private datasets: 37 studies used private datasets, which means they utilized proprietary or custom datasets specific to their research. Details about these datasets are not provided.

CSIC-2010: 29 studies used the CSIC-2010 dataset, which is an automatically generated dataset targeting an e- commerce web application. It contains 36,000 normal requests and over 25,000 anomalous requests (web attacks).

The ECML/PKDD 2007 dataset, which was produced from actual traffic data gathered during the ECML and PKDD conferences on machine learning, was utilized in three studies. It contains 15,110 malicious HTTP requests and 35,006 instances of normal traffic. To conceal parameter names and values, preprocessing was applied to the dataset.

KDD-Cup99: The KDD-Cup99 dataset, which is frequently used to create network intrusion detection models, was used in two research. There are three versions available: a testing set, a 10% training set, and the entire training set. There are 41 features in the dataset.

CICIDS-2017: The Canadian Institute for Cybersecurity's CICIDS-2017 dataset was used in six studies. It has 83 features, 2,830,540 different instances, and 15 class labels (one normal and 14 assault labels). However, the dataset is insufficient for evaluating web attack detection models because it only contains 2,180 instances of web attacks.

Six papers employed publicly accessible datasets that aren't often used in research. Xssed.com, Apache 2006/2017, and Http Params are some of these datasets. These datasets are not described in greater detail.

It's worth noting that while these datasets have been utilized

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

in previous studies, it's essential to consider the specific requirements and context of your own research when selecting a dataset for web vulnerability detection.

V. SUPERVISED AND UNSUPERVISED ALGORITHMS

A. Supervised Algorithms

Supervised learning algorithms are a type of machine learning algorithm that can be used to train a model to classify or predict an output based on a set of input data. In the context of web vulnerability detection, supervised learning algorithms can be used to train a model to identify web applications that are vulnerable to known attacks.

One of the most common supervised learning algorithms used for web vulnerability detection is the Convolutional Neural Network (CNN). CNNs can be used to extract features from web application source code. These features can then be used to train a model to identify web applications that are vulnerable to known attacks. For example, CNNs can be used to extract features such as the presence of known vulnerabilities, the use of insecure coding practices, and the presence of malicious code. Another common supervised learning algorithm used for web vulnerability detection is the Long Short-Term Memory (LSTM). LSTMs can be used to process sequences of web application requests. These sequences can then be used to train a model to identify web applications that are vulnerable to known attacks. For example, LSTMs can be used to process sequences of requests that are known to be associated with attacks.

1) Convolutional Neural Network (CNN) for Web Vulnerability Detection

A type of feedforward neural network is a convolutional neural network. Its artificial neurons have the ability to respond to a portion of the environment in the coverage, extract data features using convolutional functions, and separate high-dimensional features from low-dimensional features of the data. It performs superbly while processing enormous amounts of images. After being vectorized, the flow data can be transformed into an image data format that CNN can analyze with ease. The essential components of CNN are convolution and pooling processes. The spatial pattern and connection may be found using the convolution operation. Based on the shared-weights architecture, it can generate a large number of space invariant discriminative features with little computational effort. Using a pooling operation, which is a sub-sampling technique, you may recover the crucial information and reduce the computation required for following processes. A robust deep neural network may be constructed using a cascaded series of convolution and pooling processes.

Using a convolution kernel, or weight matrix (for two-dimensional avatars, this matrix may be a 3 3 or 5 5 matrix), each neuron in the convolution layer is associated to the local region of the feature surface of the layer above. Multiple feature surfaces, each of which is composed of numerous neurons, make up the convolution layer. [11] The first convolutional layer of the CNN recovers low-level information, and the convolutional layer of successive layers recovers higher-level data. The CNN's convolutional layer employs convolution operations to extract input features. Figure 2 shows a schematic representation of the convolutional layer and pooling layer structures of a one-dimensional CNN.

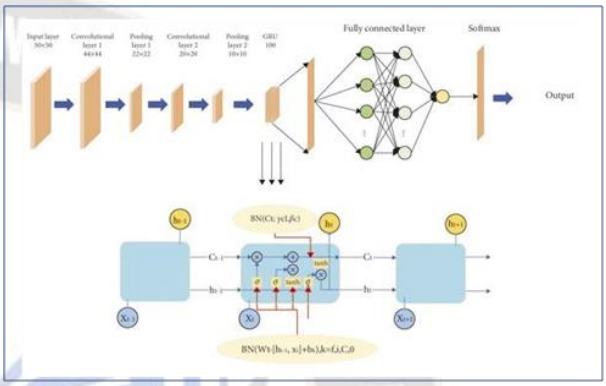

Fig 2. Convolutional Neural Network

The pooling layer comes after the convolutional layer and is similarly composed of a number of feature surfaces, each of which precisely matches a feature surface from the previous layer while maintaining the same overall number of feature surfaces. As shown in Figure 2, the convolutional layer is the input layer for the pooling layer. A feature surface in the convolutional layer uniquely correlates to a feature surface in the pooling layer, the local receptive fields of individual neurons do not overlap, and the neurons in the pooling layer are connected to the local receptive field of the input layer. The pooling layer aims to collect characteristics that are spatially invariant by reducing the resolution of feature surfaces.

The pooling layer, which serves as the secondary feature extraction stage, performs a pooling operation on the local receptive field for each neuron. One or more fully connected layers are merged in the CNN framework after multiple convolutional layers and pooling layers. Complete connection means that every neuron in the layer above is completely connected to every neuron in the layer below. The fully connected layer can incorporate the class-discriminative local information from the pooling layer or the convolutional layer.

The split of the convolution layer's neurons into each feature and the connections between each neuron and the pertinent local area of the higher layer, or the convolution layer's gland, are shown in Figure 3. The input layer feature controls local connections. Each neuron in the convolution layer had its local

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

connection weighted before being passed to a nonlinear function, like the RELU function, to determine its output value. Due to CNN's weight sharing, which also makes the network simpler to train, the model complexity for the same input and output feature may belowered.

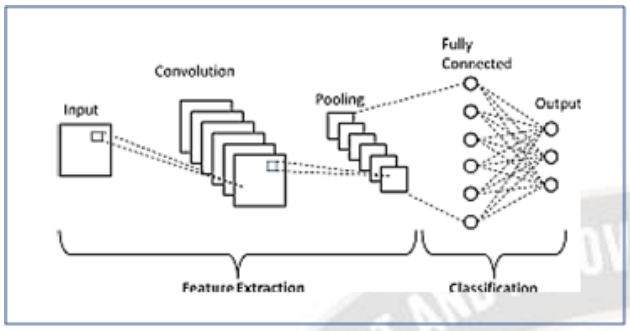

Fig 3. Pooling Layer in convolutional Neural Network

2) LONG SHORT TERM MEMORY

LSTMnetworks are used in contemporary methods of sequence learning. Despite being less prevalent, they are inherently appropriate for time series projections. Long-term dependencies can be learned using the RNN variant LSTM. Many other researchers have improved the LSTMs that Hochreiter and Schmidhuber first proposed. They are the most used RNN kind and perform well on a variety of issues.

Fig. 5(a) depicts a portion of the standard LSTM network, which consists of an input layer, an LSTM hidden recurrent layer, and an output layer. (2) A stacked LSTM network with an input layer, three LSTM hidden layers, and an output layer, as depicted in Fig. 5(b). Binary cross entropy serves as the loss function for both networks, Adam operates as the optimizer, and sigmoid serves as an activation function in the output layer. A traditional LSTM network is created by connecting a series of multiplication and addition unit sequences. There are 42 input neurons in the network, each with a unique input feature, and there are full connections between the input neurons, hidden layer memory blocks, and hidden layer LSTM neurons.

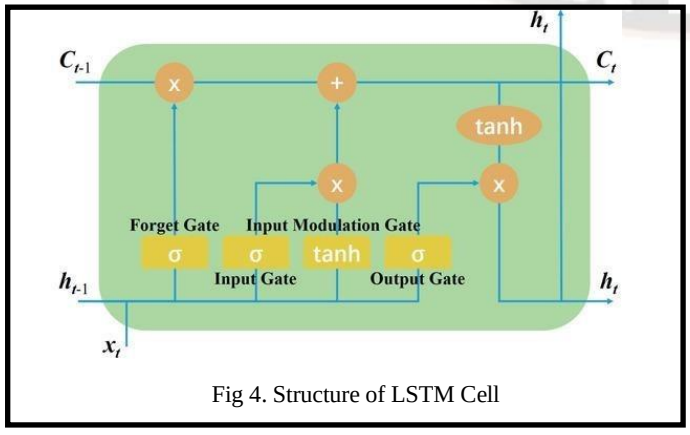

Fig 4. Structure of LSTM Cell

The complexity of the data determines how many memory blocks are needed in an LSTM hidden layer. All gates and output layer neurons are totally connected to each memory cell output in a memory block, and the optimal number of memory blocks can be determined by changing several parameters during LSTM model training, as indicated in Section 4.2.1. By sending information to a memory cell over a long period of time while continuously updating the state of the present memory cell and hidden state, these gates avoid the state of disappearing and exploding gradient concerns. The input and output gate controls the input and output flow to a memory cell at each timestep. The forget gate controls the prior state of each time step. The unidirectional flow of data between LSTM hidden layers and units in output layers, as well as into and out of LSTM hidden layers, is monitored by the LSTM network.

B. Unsupervised Algorithm

Unsupervised learning can be a valuable approach for web vulnerability detection, as it allows the system to automatically learn patterns and anomalies from the data without the need for labeled examples. While supervised learning techniques require labeled data (i.e., examples of both normal and vulnerable web requests), unsupervised learning algorithms can analyze the characteristics of web traffic and identify potential vulnerabilities based on deviations from normal patterns.

1) Generative Adversarial Network

Goodfellow et al. created GAN in 2014. It is one of the most potent and effective deep learning tools. A generative model is estimated using an adversarial technique by GAN. It is made up of the independent generator

(G) and discriminator (D) models. The data distribution p(g) across the actual data space x is estimated using the generative model G. The objective of G is to produce a new adversarial sample G(z) from the same distribution of x given an input noise variable p(z). The discriminator model D, on the other hand, returns the chance D(x) that the provided sample x comes from a real data collection rather than one created by G. While D wants to accomplish the opposite, G's ultimate objective is to increase the likelihood that D will anticipate created data as real. As a result, G and D engage in a two-player minmax game, coming up with an original solution in the process. The following is a definition of the value function V (G, D):

$$\min_G \max_D V(D, G) =$$

$$E_{x\sim pdata(x)}[\log D(x)] + E_{z\sim pz(z)}[\log(1 - D(G(z)))]$$

From the perspective of a CPS's security, GAN has applications. While some authors employed GAN to carry out assaults, others did so in order to fortify the system. Chhetri et

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
Article Received: 20 April 2023 Revised: 01 June 2023 Accepted: 24 June 2023

al. developed a conditional GAN-based model to observe important security criteria by looking at the connections between the cyber and physical domains in a CPS.

Yin et al. proposed a framework that builds on GAN and the Botnet detection model and improves the performance of the detection mechanism of a particularly potent attack while maintaining the fundamental components of the original detection model.

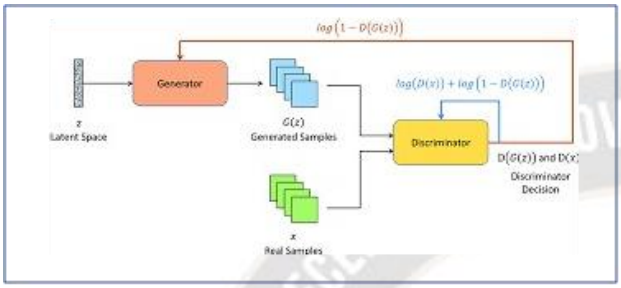

Fig 6. Generative Adversarial Network

VI. FRAMEWORKS AND PLATFORMS USED FOR DL-BASED WEB VULNERABILITY DETECTION

The main objectives of this research section are to give a general overview of the software and platforms frequently used in building DL-based online attack detection models. The frequency of use of the platforms and frameworks for building DL-based web vulnerability detection models is summarized in the table below. It first shows that the most widely used platforms and frameworks are Keras and TensorFlow.Then, a few research employ a variety of frameworks, including PyTorch, Theano, Scikit-learn, and MATLAB. 23 studies, however, lacked implementation information.

Sr. No	Frameworks	Percentage
1.	Tensorflow/ keras	34%
2.	Beautiful Soup	20%
3.	MATLAB	4%
4.	PyTorch	3%
5.	Scikit Learn	2%
6.	Theano	1%

Table 1. Frameworks and Platforms with sage Percentage

However, there are other platforms and tools used for DL-Based web vulnerability detection.

VII. VULNERABILITY DETECTION FOR DL-BASED WEB USING PERFORMANCE METRICS.

PRESENTING AND DEBATING THE PERFORMANCE MEASURES FREQUENTLY EMPLOYED TO ASSESS DL-BASED ONLINE ATTACK DETECTION MODELS IS THE GOAL OF THIS RESEARCH SECTION.

Making a classification model that can determine whether a web application is vulnerable or not (binary classification), whether it is vulnerable or not to a particular web attack (for example, vulnerable or not to SQLi attacks), or which web attack it is vulnerable to (multi-classification problem) is required when using DL models for web attack detection.

Accuracy: By comparing the proportion of examples that were properly classified to all instances, it assesses the general accuracy of the model's predictions.

$$\text{Accuracy} = \frac{TN+TP}{\text{TP + TN + FP + FN}}$$

Precision: The percentage of accurate positive predictions compared to all positive forecasts is measured. Focusing on the accuracy of positive predictions, precision evaluates the model's capacity to prevent false positives.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall, often referred to as sensitivity or the true positive rate, determines the proportion of correctly predicted positive outcomes to all of the actual positive cases. Recall emphasizes how well the model can spot positive occurrences, reducing false negatives.

$$\text{Recall} = \frac{TP}{TP+FN}$$

The area under the ROC curve (AUC-ROC) displays how effectively the model can discriminate between positive and negative situations by plotting the true positive rate against the false positive rate. A higher AUC-ROC score indicates better model performance.

False Positive Rate (FPR): This statistic shows how many projected false positives outweigh all actual negatives. The model's ability to avoid identifying benign events as attacks depends critically on FPR. $\text{FPR} = \frac{FP}{FP+TN}$

False Negative Rate (FNR): It measures the ratio of false negative predictions to the total number of actual positives. FNR helps evaluate the model's capability to detect actual attacks accurately.

$$\text{FNR} = \frac{FN}{FN+TP}$$

Since TP and TN don't apply to multi-classification issues, evaluating accuracy is straightforward, but computing metrics like precision, recall, FPR, F1-Score, and AUC is more difficult. Only when the challenge is reduced to a two-class problem (all classes vs. one class) and only when the metrics are generated for each class, can these metrics be computed. Only problems with three or more classes can make use of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

these measures. In multi-class circumstances, precision is typically the sole factor used.

VIII.PROCESSING STAGE FOR GANS, CNN AND LSTM

The three phases that make up the processing stage are tokenization, generalization, and decoding.

IX. DECODER

Attackers may utilize encoding methods instead of regular expressions for conventional filtration or validation, such as Hex encoding, URL encoding, Unicode encoding, HTML entity encoding, etc.

As a result, this study suggests a decoder for evaluating repeatedly and returning all input data encoding alternatives to their original form. words made of letters and digits In order to reduce the number of participles, it is necessary to streamline the quantity of numbers and hyperlinks by replacing numerals with "0" and hyperlinks to http: / u. For instance, an HTML: encoded XSS payload that was previously "script>a l e r t (1)" is now "script>alert (1) /script>" after the decoder's processing.

X. GENERALIZATION

The following procedures are used to generalize the decoded data in order to reduce the impact of duplicated and unrelated data: To start, we utilize 'http:/website' to modify the amount of URLs in the input data. The numbers in the data are then altered to "0." And the original string is replaced with "param string" as a function argument.

- Control and blank characters were also eliminated, along with other special features.

XI. TOKENIZATION

Based on the properties of the scripting language, a collection of regular expressions for customizing input data were constructed in this study. Table 1 displays the tokenization classification.

Classification	List
Start Label	<script>, <frame>, <img, <body>, etc
End Label	</script>, </frame>, </body>, etc
Windows Event	onerror=, onload=, onblur=, oncut=, etc
Function Name	alert(, prompt(, String.fromCharCode(, etc
Script URL	javascript:,vbscript:, etc
Others	>,), #, etc

Table 2. Classification of Tokenization

XII.COMPARISON

A comparative study of these three models was conducted. The study found that CNNs were the most effective at identifying known vulnerabilities, while LSTMs were better at identifying new vulnerabilities. GANs were not as effective as the other two models, but they were able to generate realistic-looking web pages that could be used to test the security of web applications.

The study also discovered that the amount of the training dataset had an impact on the models' performance.. CNNs performed better with larger datasets, while LSTMs performed better with smaller datasets. GANs were not as sensitive to the size of the training dataset.

XIII.LIMITATIONS

This study's main limitations are the absence of comparisons between certain studies. Additionally, we wish to emphasize that while we properly presented the experiment findings from the studies we examined, we did not follow their experimental design or apply their models.There could be some missing materials. Therefore, it is difficult to say that our search approach included all pertinent research.

XIV.CONCLUSION

Web applications may be vulnerable to a variety of security flaws. As a result, a number of solutions for detecting and stopping online assaults have been proposed. This research focuses largely on three commonly used DL-Based online vulnerability detection algorithms.The datasets used and the stages of processing for the deep learning-based online vulnerability detection are also described in this paper.

Based on the results of our investigation, we have identified several intriguing research directions for next work in the DL-based online assaults detection domain:

- Create common, open-source, real-world datasets: To address the current dataset’s problems, it is critical to create datasets for the detection of web attacks. In fact, most studies used proprietary datasets. Public datasets also don't take into account recently identified web assaults or the intricacy of real-world web applications. Additionally, it is more challenging to evaluate various approaches when different chunks of the same publicly accessible dataset are utilized for training and testing. Therefore, there is a need for standardized realistic public datasets in order for the research community to contribute to this field efficiently, to enable comparisons across research efforts simpler, and to make the presented models applicable in real-world online applications.
- Examine cutting-edge DL models for identifying online assaults: In various state-of-the-art DL models, the literature on DL-based online attack detection was sparse. Encoders-Decoders have previously been successfully utilized in a similar topic called Networks Intrusion Detection, making its application to the detection of online assaults appealing.
- Create a standard framework for comparing DL-based online attack detection models. Due to the wide range of performance metrics, datasets, and platforms employed in the evaluated

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7344
***Article Received**: 20 April 2023 **Revised:** 01 June 2023 **Accepted:** 24 June 2023*

studies, it is challenging, if not impossible, to compare research works. It is crucial to offer standardized datasets, performance measures, environments, as well as an open research methodology that enables comparing various approaches and assessing the models' applicability for actual online applications.

REFERENCES

[1] Alaoui, R.L.; Nfaoui, E.H. Deep Learning for Vulnerability and Attack Detection on Web Applications : A Systematic Literture Review. Future Internet 2022, 14, 118. https://doi.org/10.3390/fil4040118.

[2] Waleed Bin Shahid, Baber Aslam, Haider Abbas, Saad Bin Khalid, Hammad Afzal. An Enhanced deep Learning based framework for web attack detection, mitigation and attacker profiling. https://doi.org/10.1016/j.jnca.1021.103270.

[3] Laith Alzubaidi, Jinglan Zhang, Amjad j. Humaidi, Ayad Al-Dujaili, Ye Duan, Omaran Al-Shamma, J. Santamaria, Mohommaed A. Fadhel, Muthana Al-Amidie & Laith Farhan. Review of deep learning : concepts , CNN architectures challenges, applications, future directions. Article number: 53 (2021).

[4] Li Lei, Ming Chen, Chengwan He, Duojio Li. XSS Detection Technology Based on LSTM- Attention. 2020. DOI 10.1109/CRC51253.2020.9253484.

[5] R. Lamrani Alaoui and E.H. Nfaoui, "Cross Site Scripting Attack Detection Approach Based on LSTM Encoder- Decoder and Word Embeddings". Int J Intell Syst Appl Eng, vol. 11, no. 2, pp. 277-282, Feb. 2023.

[6] Laghrissi, F., Douzi, S., Douzi, K. et al. Intrusion detection systems using long short-term memory (LSTM). J Big Data **8**, 65 (2021). https://doi.org/10.1186/s40537-021-00448-4

[7] Supriya Shende, Samrat Thorat. Long Short-Term Memor (LSTM) Deep Learning Method for Intrusion Detection in network security. DOI: 10/17577/IJERTV9IS061016

[8] Shatha Fawaz Ghazal, Salameh A. Mjlae. Cybersecurity in Deep Learing Techniques: Detecting Network Attacks. Vol. 13

[9] Muhammad Shoib Akhtar, Tao Feng. Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. Symmetry 2022, 14(11), 2308; https://doi.org/10.3390/sym14112308.

[10] Aditi Sharma, Ankit Singh, Nikhil Sharma, IIa Kaushil, Bharat Bhushan. Security Countermeasures in Web Based Application. July 2019. DOI: 10.1109/ICICICTT46008.2019.8993141.

[11] Ran Wang, Guangquan Xu, Xiajio Zeng, Xiahong Li, Zhiyong Feng, TT-XSS: A novel taint tracking based dynamic detection framework for DOM Cross-site Scripting. Vol 118 , Part 1, Aug-2018. https://doi.org/10.1016/j.jpdc.2017.07.006.

[12] Sumaiya Tasneem, Kishor Datta Gupta, Arunav Roy. Generative Adversarial Networks (GAN) for Cyber Security: Challenges and Opportunities. Conference Paper · January 2023.

[13] Zilong Lin, Yong Shi, Zhi Xue. IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. Vol 13282. DOI: 10.1007/978-3-031-05981-0_7.

[14] Dr. Kalyan Bamane, Rasika Mahure, Sakshi Padamwar, Saniya Priya, Palak Tiwari, Mrunmai Kashyape. Techiniques for Detection and remediation of various web vulnerabilities.

[15] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, Miguel Alonso. G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System. 2020. DOI : 10.1109/COMPSAC48688.2020.0-218.

[16] J. Kim, J. Kim, H. L. Thi Thu and H. Kim, "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, Korea (South), 2016, pp. 1-5, doi: 10.1109/PlatCon.2016.7456805

Elastic- Pooling CNN. IEEE Access 2019, 7, 151475–151481.