EasyChair Preprint

№ 10029

Web Attack Detection Using Deep Learning

B Logesh, Perasani Bhargav, Appikonda Jeevan Kumar,
Genji Yaswanth, Chintala Uday Kiran and Jyoti Godara

May 9, 2023

Web Attack Detection Using Deep-Learning

Logesh B, Perasani Bhargav, Appikonda Jeevan Kumar, Genji Yaswanth,
Chintala Uday Kiran, *Jyoti Godara
School of Computer Science and Engineering
Lovely Professional University, Phagwara, Punjab
logeshbala88@gmail.com, pbhargav0510@gmail.com, jk7680979612@gmail.com
genjiyaswanth16@gmail.com, Ch.udaykiran2001@gmail.com, *jyotipoonia6@gmail.com

***Abstract*—The number of unsecured internet programs has recently increased significantly. A detection mechanism that uses a triangle module operator and deep learning methods is proposed to battle web attacks like SQL injection attacks. To handle these types of attacks on data-based websites such as SQL injection or other similar ones, we advocate Data Collection, Data preprocessing, and Model Training through deep learning algorithms followed by Evaluation techniques. For us to protect against SQL injections from happening at all times parallelly while running the website online, therefore it improves information security if we conduct penetration testing tests along with source code vulnerability tests alongside configuration verification beforehand making sure our Web System passes every vulnerability test before going live. When dealing with Cross-Site Scripting (XSS), verifying input values become crucial so only allow-listed inputs are accepted whereas converting the variable output into encoded versions before showing it back onto your page helps prevent XSS-type malicious activities take place inadvertently taking over control off-hand which may lead toward damage beyond imagination.**

***Keywords*—Deep learning algorithms, SQL injection attack, Cross-Site Scripting, Neural Network**

I.INTRODUCTION

The detection of web attacks is an essential part of day-to-day living since it safeguards individuals and businesses from a wide range of cyber threats XSS and SQL Injection are the most common web Vulnerability attacks to gain the sensitive information which is stored online. The internet has become an indispensable component of our lives. Identifying and preventing web assaults helps to safeguard our data. SQL injection [4,22], Cross-Site Scripting [1,2] and various sorts of malware are all examples of online attacks. XSS is a kind of attack in which malicious code is inserted into a website and then executed on the user's browser. This enables the attacker to obtain important information such as login passwords or personal data, giving them more power over the situation as represented in Figure 1. Altering the input that is sent to the database of a web application in order to gain unauthorized access to the data or modify it is an example of an attack known as SQL injection in Figure 2. These attacks might result in a range of unfavorable results, such as the loss of money, the theft of identity, and damage to reputation. By deploying online attack detection systems and keeping up to date with the most recent security standards, both individuals and companies may help to mitigate the threat posed by these assaults and protect themselves from the resulting harm. Artificial Neural Networks (ANN) have been used in the past to detect and prevent these types of attacks. ANN algorithms work by learning patterns in the data and using this knowledge to predict the likelihood of an attack. To detect SQL injection and XSS attacks, ANN algorithms can be trained on a dataset of known attack patterns, such as SQL injection strings, and in XSS Attacks such as javascript and event handlers. The algorithm can use this to identify and analyze incoming requests and flag any incoming requests that contain similar patterns.

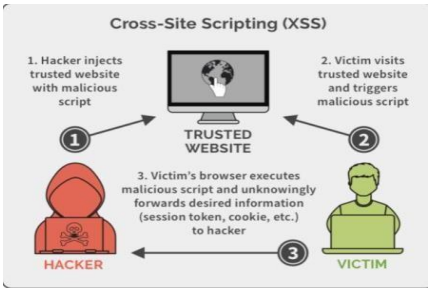

Figure 1: Cross Site Scripting Attack

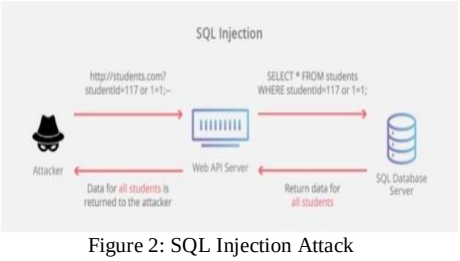

Figure 2: SQL Injection Attack

The main objectives of our project to provide an overview of SQL Injection and Cross-Site Scripting (XSS) vulnerabilities in web applications and to detect both SQL Injection and XSS attacks using the deep learning model and ANN algorithm. Neural Networks are effective in detecting SQL Injection and XSS attacks in web applications because they can handle non-linear relationships, high-dimensional data, and complex patterns, and can be trained using both supervised and unsupervised learning techniques. Our proposed system detects whether the query is normal or SQL injected query or XSS affected.

II.LITERATURE REVIEW

The author in [1] proposed a system to resolve Cross Site Scripting which provides data security and also improves accuracy and reliability. The logistic regression model, SVM model, and decision tree model were used for training to detect the attacks by using a machine learning platform. As a result, the logistic regression model yields an accuracy of 92.85% for SQLI and the SVM approach yields an accuracy of 85.62%.

The author in [2] proposed an intrusion detection system (IDS) that protects legitimate users from malicious activity by processing traffic collected in a backbone network in real time, providing near-optimal detection performance and low false positive rates. In fact, extensive experimental tests conducted to validate and evaluate the system confirm that with appropriate parameter settings, it can achieve a detection rate of approximately 90% with an accuracy of 0.871.

The author in [3] proposed a WAF that is used to determine the HTTP traffic and to construct features from HTTP data. They also constructed features from HTTPS logs by using an n-gram character-based model. SAE is used for feature extraction and to detect anomalies. Anomaly detection is achieved with the isolation forest method. As a result, the various performance with different structures of SAE which has better performance.

The author in [4] proposed session-based pattern detection, which is a combination of different methods. A proprietary session ID cookie is used to store a session ID, and web applications are dynamically based on the page controller and front controller software design patterns. Two methods are used, the first is to create a special one and the second is to request a URL converter. To detect attacks more reliably used higher-order Markov models. Therefore, the proposed hybrid method is more reliable for attack detection.

The author in [5] proposed to develop a supervised machine learning system to classify network traffic as malicious or benign. To determine the best model based on the detection success rate, a supervised learning algorithm and a feature selection method are combined. Through this study, it was found that ANN-based machine learning with wrapper feature selection outperforms SVMs in network traffic classification.

The author in [6] presented a review of several deep learning techniques proposed in the literature for detecting network attacks. For the identification of malware and intrusions, deep learning approaches like recurrent neural networks, convolutional neural networks, and deep belief networks have been used. It also identifies additional areas of research, such as the development of techniques to prevent attacks before they cause damage, the problem of imbalanced data where it is difficult to train models, and other areas where the effectiveness of intrusion detection systems could be improved. It describes deep learning techniques to detect cyberattacks.

The author in [7] presented a review of deep learning techniques used in anomaly-based intrusion detection systems (IDSs). The authors highlighted the benefits of using deep learning techniques for IDS and provides a taxonomy of these techniques. And the taxonomy is divided into three categories as supervised, unsupervised and semi-supervised learning. Within each category, various deep learning techniques are described such as neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders. The detailed analysis shows techniques such as supervised learning techniques are highly accurate but require large labeled datasets, while unsupervised learning techniques can work with unlabeled data but may produce false positives.

The author in [8] proposed a method for detecting network intrusions using deep learning techniques. The process entailed utilizing a deep neural network to examine network traffic and categorize it as either benign or malevolent. The authors collected the traffic data from various network sources and extracted good features using deep learning techniques. Self-taught learning, SoftMax regression, and deep learning techniques were used for developing NIDS, and the NSL-KDD dataset was used for the evaluation of

anomaly detection accuracy. Based on the testing data, STL-based NIDS showed higher performance and accuracy, and it is observed that to develop efficient detection systems, deep-learning approaches are best.

The author in [9] proposed a method for detecting false data injection attacks (FDIA) in industrial control systems. FDIA is a type of cyber-attack that can compromise the integrity of data in control systems, leading to incorrect decisions and potentially dangerous consequences. The proposed method combines wavelet transform and deep neural networks to detect FDIA in real-time. Feature extraction from the data had done by using the wavelet transform which is then fed into deep neural networks for classification. The authors compared the proposed method with other existing methods on a data set simulating industrial control systems. As a result, the proposed method overcame other methods in terms of accuracy, false positive rate and detection time.

The author in [10] proposed a deep neural network-based approach for detecting network anomalies. It involves the use of convolutional neural networks, auto encoders and long short-term memory techniques to extract features from network traffic and to train the deep neural networks. The deep learning models were trained and evaluated on NSLKDD datasets. As a result, the method achieved 85% and 89% accuracy with DCNN and LSTM models states that deep neural networks show its efficiency in detecting network anomalies.

III.METHODOLOGY

We have used two modules for this project they are “Deep learning model” and the “Website module”.

“Deep learning model” are powerful for solving problems in machine learning and it is to provide better performance in detecting attacks.

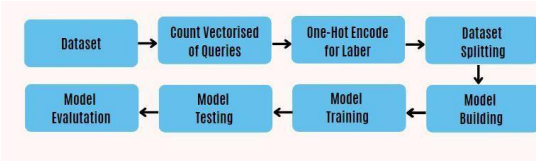

Figure 3: Flow Diagram of Deep-Learning Model

A. Data Collection:

Data will be collected from a web application that receives the user queries and stored in the form of a CSV file. The queries are labelled as 0,1 and 2, where 0 is for normal queries, 1 is for SQL Injection and 2 is for XSS. Sample dataset with their label has mentioned in Figure 4.

Query	Label
select * from users where id = 1 or 1#"" (union select 1,version ( ) -- 1"	1
SELECT * FROM memory,0	0
"Backpropagation :"	0
<nobr id=x tabindex=1 onfocus=alert(1)></nobr>	2

Figure 4: Dataset Sample

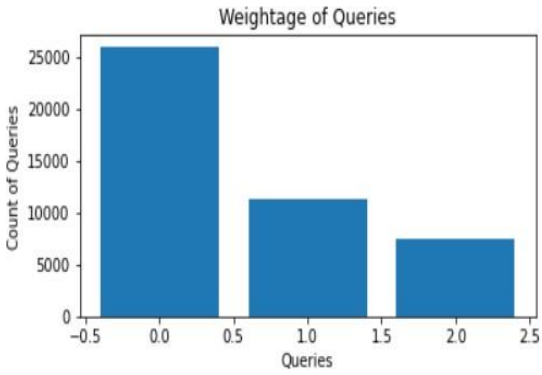

Figure 5: Weightage of Queries

B. Data Preprocessing:

The collected dataset is pre-processed to convert the information into a format that can be used for training the model. Countvectorizer and One-Hot Encoding are used where each row represents a sentence and each column represents a distinct word in the 'sentences', the Countvectorizer transforms the input sentences into a form of a matrix of token counts. One-hot encoding is to represent categorical data in a numerical format and it works by creating a binary vector for each category, where each element in the vector represents a unique category.

C. Model Building:

The process of designing and constructing a neural network model, the framework which is used here is sequential model from keras. Here three activation function is used they are ReLU, Tanh and SoftMax activation functions. ReLU uses an input layer with 20 nodes which is the first layer and it receives the input data and also hidden layer with 1024 nodes for output from the second hidden layer. Tanh uses a hidden layer with 10 nodes where this layer presented in between the

input and output layer and this is to extract and amplify the most relevant features from the input data. SoftMax uses a hidden layer with 3 nodes it takes a vector of values and applies functions to each value. The model is compiled by using binary cross-entropy loss and Adam optimizer.

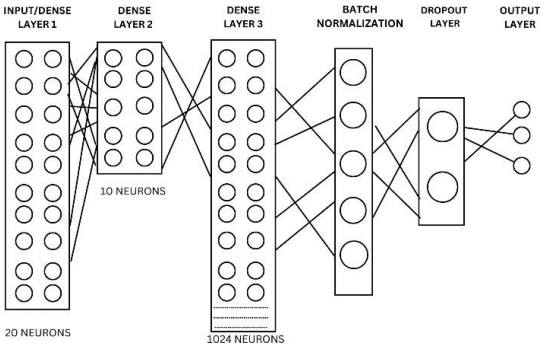

Figure 6: Layers Used in Model Building According to the Number of Neurons

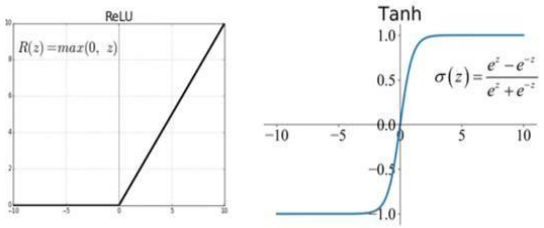

Figure 7: ReLu and Tanh Activation Functions.

D. Training and Testing:

Neural network is designed to identify and classify the different types of queries based upon their patterns in the input features. During the training process the network learns to recognize pattern where patterns could be based on various aspects like length, structure and so on. Test set is to validate the performance and prevent overfitting.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 20)                225760

 dense_1 (Dense)             (None, 10)                210

 dense_2 (Dense)             (None, 1024)              11264

 batch_normalization (BatchN  (None, 1024)             4096
 ormalization)

 dropout (Dropout)           (None, 1024)              0

 dense_3 (Dense)             (None, 3)                 3075

=================================================================
Total params: 244,405
Trainable params: 242,357
Non-trainable params: 2,048
_________________________________________________________________
```

Figure 8: Training of Dataset

E. Evaluation:

When the model has been trained, it is tested using a test dataset to see how well it predicts the proper labels for each.

Accuracy = (TP + TN) / (TP+TN+PF+FN)
Precision Score = TP / (FP + TP) Recall Score = TP / (FN + TP),
F1 Score = 2* ((Precision Score * Recall Score)/ (Precision Score + Recall Score))
Where TP is True Positive, FP is False positive, FN is False Negative and Support is also used where the number of actual occurrences of the class in the specified dataset.

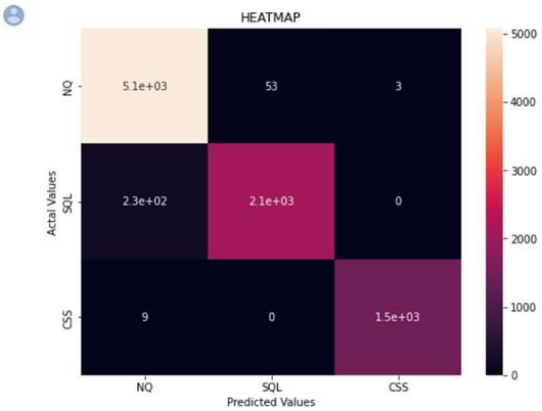

Figure 9: Graphical Representation using Heatmap

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.95	0.98	0.97	5133
1	0.96	0.90	0.93	2308
2	1.00	1.00	1.00	1471
micro avg	0.96	0.96	0.96	8912
macro avg	0.97	0.96	0.97	8912
weighted avg	0.96	0.96	0.96	8912
samples avg	0.96	0.96	0.96	8912

Figure 10: Average occurs during Evaluation

"Website Module" comprises of a set of files including three PHP files, HTML and CSS files, images, and a SQL database. Where HTML & CSS are used to make the form accessible for users which provides user interface to display the forms and queries. Bootstrap is used for input form. PHP is used here to handle the server-side logic and database interactions.

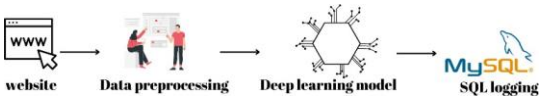

Figure 11: Website Module

Index.php is the main page where users can enter a query and it sends a GET request to itself by passing the query text as a parameter. Getquery.php fetches the logged queries from the database and also establish a connection to database and executes the SQL SELECT query. Insertlog.php logs a new query database which receives the query text and type as GET parameter and connects to the database to run an INSERT query to provide values. Where php are used to allow users to log, fetch queries from the database and also it outputs a success or error message to the database.

The python library named Beautiful Soup which is used for parsing HTML and XML documents. It is used handy for "scraping" data out of HTML, and that's its purpose here - to allow the Python code to get the query data.

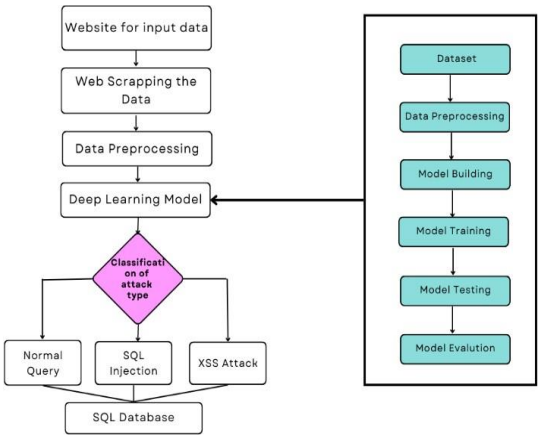

Figure 12: System Architecture

IV.IMPLEMENTATION & RESULT

The Deep Learning Model process is done in python with the help of Google colab. Here a few steps for the implementation.

1)User inputs text into a form on the website hosted on a web host server.

2)The form sends a request to the server using PHP, which includes the user input as a parameter.

3)On the server, Python running on Google Colab receives the request and uses the requests module to retrieve the user input from the web host server.

4)The retrieved user input is processed using Beautiful Soup to extract any relevant features for the machine learning model.

5)The extracted features are preprocessed using count vectorization to convert the text into a numerical format that can be input into the machine learning model.

6)The preprocessed features are passed through the machine learning model to classify the type of attack (e.g., normal query, SQL injection, XSS attack).

7)The classification result is logged in an SQL database on the web host server, along with any other relevant information about the request (e.g., user IP address, timestamp, etc.).

8)The server sends a response back to the website with the classification result, which can be displayed to the user.

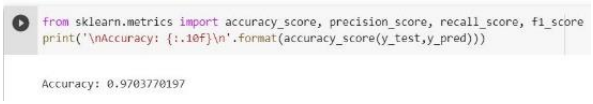

Figure 13: Accuracy

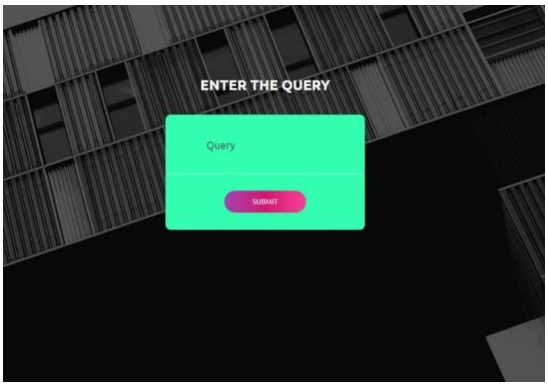

Figure 14: Input Query

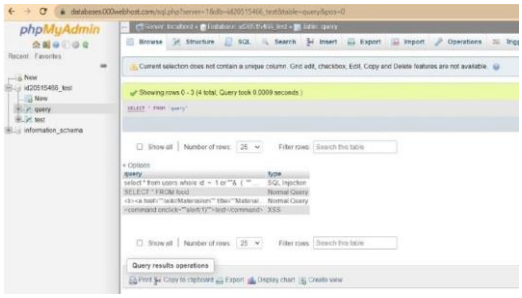

Figure 15: Fetching Queries from the Database

V.CONCLUSION

Web attack detection using deep learning is a promising approach that can improve the accuracy and efficiency of web security systems. Deep learning models can learn complex patterns and features that may not be easily detectable through traditional rule-based methods, making them effective in detecting new and evolving attack types. But it is important to note that deep learning models can be computationally expensive and require a large amount of training data to achieve high performance. We predicted that models that we are going perform to have high accuracy and low false positive rates, indicating their potential for real-world deployment. This project highlights the importance of data preprocessing, feature engineering, and model selection in building an effective neural network model. The potential benefits of deep learning for web attack detection make it an area of active research and development. Future work in this area could focus on developing more efficient deep learning architectures, improving the quality and diversity of training datasets, and integrating deep learning models with other security mechanisms to create more robust and effective web security systems.

VI.REFERENCES

[1] Bhardwaj, A., Chandok, S. S., Bagnawar, A., Mishra, S., & Uplaonkar, D. (2022, September). Detection of Cyber Attacks: XSS, SQLI, Phishing Attacks and Detecting Intrusion Using Machine Learning Algorithms. In 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT) (pp. 1-6). IEEE.

[2] Callegari, C., Giordano, S., & Pagano, M. (2017, January). Entropy-based network anomaly detection. In 2017 International Conference on Computing, Networking and Communications (ICNC) (pp. 334-340). IEEE.

[3] artouni, A. M., Kashi, S. S., & Teshnehlab, M. (2018, February). An anomaly detection method to detect web attacks using stacked auto-encoder. In 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS) (pp. 131-134). IEEE.

[4] Dik, D., Polyakova, E., Chelovechkova, A., & Moskvin, V. (2019, October). Web attacks detection based on patterns of sessions. In 2019 International Multi- Conference on Industrial Engineering and Modern Technologies (FarEastCon) (pp. 1- 5). IEEE.

[5] Taher, K. A., Jisan, B. M. Y., & Rahman, M. M. (2019, January). Network intrusion detection using supervised machine learning technique with feature selection. In 2019 International conference on robotics, electrical and signal processing techniques (ICREST) (pp. 643-646).

[6] Wu, Y., Wei, D., & Feng, J. (2020). Network attacks detection methods based on deep learning techniques: a survey. Security and Communication Networks, 2020, 1- 17.

[7] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge- Based Systems, 189, 105124.

[8] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio- inspired Information and Communications Technologies (formerly BIONETICS) (pp. 21-26).

[9] James, J. Q., Hou, Y., & Li, V. O. (2018). Online false data injection attack detection with wavelet transform and deep neural networks. IEEE Transactions on Industrial Informatics, 14(7), 3271-3280.

[10] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. IEEE access, 6, 48231-48246.