

A hybrid of CNN and LSTM methods of XSS Attacks

Sruthy Varghese
Department of Computer Application
Amal Jyothi College of Engineering, Kanjirappally,
Indiasruthyvarghese2022b@mca.ajce.in

Lisha Varghese
Department of Computer Application
Amal Jyothi College of Engineering, Kanjirappally, India
lishavarghese@amaljyothi.ac.in

Abstract — Cross-site scripting is a common attack on online applications that is vulnerable at the application layer. The attacker injects the script and handles malicious scripts for trusted websites. There are various sorts of XSS scripts that can be used to target extremely open websites and online apps. The attacker could load or redirect the malicious web page. We then train and test word vectors using CNNs to create a model that can be used in a web application. The performance of the CNN algorithms in identifying XSS vulnerabilities in online applications and web pages was used to train machine learning algorithms. To detect the accuracy of the machine learning algorithm of the model, convolutional neural networks are used.

Keywords—*Python Programming, Xss Attack, and Machine Learning.*

I. INTRODUCTION

Cross-Site Scripting is the ability to embed HTML frames in a web page and then read information from one frame to another using arbitrary code thanks to the introduction of JavaScript in online applications. This method can be used to steal passwords, cookies, and other sensitive information. Even after the implementation of the "same origin policy," which prevented JavaScript from one website from accessing data from another, attackers continued to find methods around it. A successful attack could expose both the website host and the user viewing it to serious security risks. Modern attacks have the ability to inject arbitrary code and change user input fields.

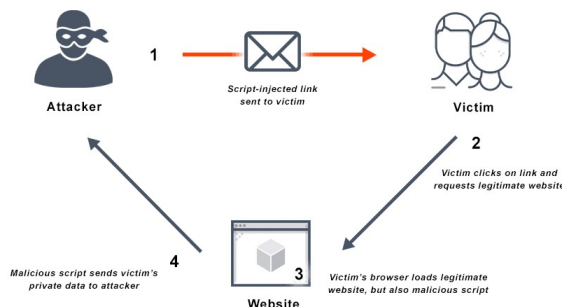


Fig: Architecture of XSS attacks

Reflected, stored, and DOM-based XSS are the three most common types of XSS.

A. Stored XSS

A perpetrator must first find a weakness in a web application and then inject a malicious script into their server in a stored XSS attack.

B. Reflected XSS

A reflected XSS attack occurs when a malicious script is reflected in a web application and the victim's browser. By clicking on a link that sends a request to a website with a vulnerability that allows malicious script execution, the script is triggered.

C. DOM XSS

The attack payload in a DOM-based XSS assault never leaves the browser.

When predicting whether a new URL matches to an XSS attack, Convolutional Neural Networks (CNN) has a deep learning system that reaches up to 97.4 percent accuracy. Cross-site scripting attacks in client-side web applications are prevented with great accuracy using deep learning.

II. LITERATURE REVIEW

Raed Waheed Kadhim, Methaq Talib Gaata[1] propose A method for detecting XSS attacks was proposed that combined a convolutional neural network (CNN) with long short term memories (LSTM), Word2vec was used to convert words into word vectors in XSS payloads after pre-processing the XSS Data Set with decoding, generalisation, and tokenization.

Bronjon Gogoi, Tasiruddin Ahmed, and Hemanta Kumar Saikia[2] Traditional methods of defence against XSS attacks include hardware and software-based web application firewalls, the bulk of which are rule and signature-based. Rule-based and signature-based web application firewalls can be bypassed by obfuscating the attack payloads. Machine learning is being used to detect XSS attacks in web apps and

websites.

Caio Lente, Roberto Hirata Jr., Daniel Macedo Bastista [3] deep learning system called 3C-LSTM that uses Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) techniques to predict if a new URL belongs to a benign location or an XSS assault with up to 99.4 percent accuracy. Before deciding on the ideal structure for a more efficient, yet equally accurate, system, 3C-LSTM proposes a variety of network configurations and validation techniques.

V S Stency and N Mohanasundaram [4] Methods for detecting XSS attacks based on various performance measures. In order to meet these conditions, this research examines a number of publications published in popular journals between 2019 and 2020. The simplicity of the algorithms, the kind to which they belong, and the performance indicators are all compared in the examined articles.

III. MOTIVATION

Cross-site scripting, one of the most popular web application attacks, is vulnerable to the application layer. On trusted websites, the attacker injects the script and manages malicious scripting. In a variety of ways, XSS scripting can be used to target extremely open websites. The attacker could load or redirect the malicious webpage. XSS attacks are common on well-known websites. It's tough to detect and fight against XSS assaults. ML tactics that exploit CNN of the algorithms in identifying XSS attacks in web apps and websites are used. Users in this system utilize a dataset to create a model. The model is trained in this manner.

IV. METHODOLOGY

The goal of this research is to find a Machine Learning model that can accurately detect XSS attacks using the provided dataset. The model should be able to identify XSS threats accurately. The image dataset is split into two parts: training and testing. To improve learning accuracy, the model should be trained with more data. This model's output can be used to attack classification. XSS attacks hinder CNN from using machine learning, Deep Learning, also referred to as CNN or Convolutional Neural Network is a type of artificial neural network that is used to recognize and categorize images and objects. Deep Learning recognizes items in an image using a CNN. Image processing, computer vision tasks such as localization and segmentation, and audio recognition in natural language processing are just a few of the activities and applications that CNNs are employed for.

The algorithms that require to be followed are:

Output: Normal or abnormal

Step1: Convolutional neural network one dimension(CNN1)

Step2: Convolutional neural network one dimension(CNN2)

Step3: Zero padding 1

Step4: Convolutional neural network one dimension(CNN3)

Step5: Zero padding 2

Step6: Concatenate (CNN1, CNN2, CNN3)

Step7: Dropout 0.5%

Step8: Fully Connected layer

Step9: Output

The structure is:

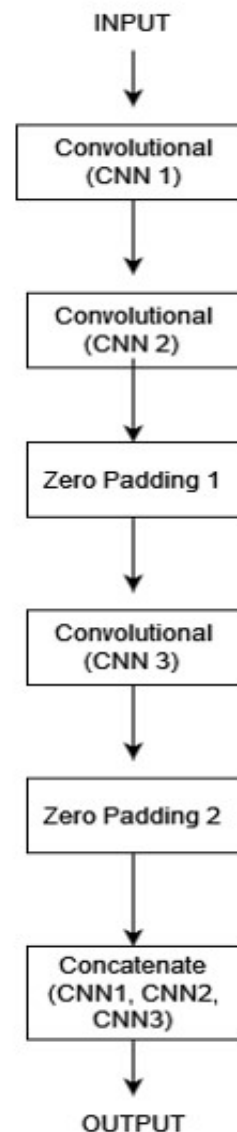


Figure2: CNN normal or abnormal.

There are many different sorts of assessment methods that

have been developed to assess the performance of new methods. The most popular sorts of evaluation methods used in this work are Accuracy, Recall, and Precision. Furthermore, the confusion matrix has four classes: True Positives, which represents the number of correctly classified XSS samples, False Positives, which represents the number of Non-XSS samples wrongly classified as XSS, True Negatives, which represents the number of correctly classified Non-XSS samples, and False Negatives, which represents the number of XSS samples wrongly classified as Non-XSS. The calculation is presented in this paper as follows:

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

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

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

V. BUILD MODEL

There is a various step taken part to detection of XSS attacks from that Convolutional Neural Network.

1. Import the packages that are necessary.

```
import numpy as np
import pandas as pd
import glob
import time
import pandas as pd
# from xml.dom import minidom

import os
import matplotlib.pyplot as plt
import keras
import cv2
```

2. Image Dataset Uploaded
3. Get Sentences data from data frame

```
# Get Sentences data from data frame
sentences=df['Sentence'].values
sentences[1]
```

4. Converted to ASCII
5. Send each sentence to be converted to ASCII

```
arr=np.zeros((len(sentences),100,100))

for i in range(len(sentences)):

    image=convert_to_ascii(sentences[i])

    x=np.asarray(image,dtype='float')
    image = cv2.resize(x, dsize=(100,100), interpolation=cv2.INTER_CUBIC)
    image/=128

    arr[i]=image
```

6. Reshape data for input to CNN

```
print("Input data shape : ", arr.shape)
```

Input data shape : (13686, 100, 100)

```
# Reshape data for input to CNN
data = arr.reshape(arr.shape[0], 100, 100, 1)
```

```
data.shape
```

(13686, 100, 100, 1)

7. Spilt into train/test data

```
# Split into train/test data
from sklearn.model_selection import train_test_split
trainX, testX, trainY, testY = train_test_split(data,y, test_size=0.2, random_state=42)
```

8. Import libraries for Making Model

```
# import libraries for Making Model
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense, Activation, Conv2D, MaxPooling2D, Flatten, Dropout, MaxPool2D, BatchNormalization
```

9. Stop when validation accuracy > 97
10. predict for test set

```
# predict for test set
pred=model.predict(testX)
```

11. Count True predicted and wrong predicted

```
# Count True predicted and wrong predicted
|
true=0
false=0

for i in range(len(pred)):
    if pred[i] == testY[i]:
        true+=1
    else:
        false+=1

print("correct predicted :: ", true)
print("false prediction :: ", false)
```

```
correct predicted :: 2682
false prediction :: 56
```

12. Number of attack and benign data in test set

```
# Number of attack and benign data in test set

attack=0
benign=0
for i in range(len(testY)):
    if testY[i]==1:
        attack+=1
    else:
        benign+=1

print("Attack data in test set :: ", attack)
print(" Benign data in test set :: ", benign)
```

```
Attack data in test set :: 1478
Benign data in test set :: 1260
```

13. Using confusion matrix

```
def accuracy_function(tp,tn,fp,fn):

    accuracy = (tp+tn) / (tp+tn+fp+fn)

    return accuracy
```

VI. RESULT

Deep learning and CNN can be used to build classifiers for XSS giving high accuracy (up to 97.75%), Recall (up to 96.50%) and precision (up to 99.88%) when applied to a large real world data set. ML approach combined with CNN can detect XSS attacks with a higher accuracy.

```
Accuracy : 0.9795471146822499
Precision : 0.9930651872399445
Recall : 0.9688768606224628
```

VII. CONCLUSION

Comparative examination of its output CNN, detect XSS assaults using the ML deep learning model. In terms of identifying XSS assaults, ML methods outperform Deep Learning with CNN. Deep learning is used to detect low-complexity, low-accuracy events. XSS attacks can be detected more accurately using a machine learning approach paired with CNN. This approach's future potential is that it can be integrated with other MCTS, GAN, and Deep learning in web applications.

VIII. REFERENCES

- [1] Raed Waheed Kadhim, Methaq Talib Gaata. "A hybrid of CNN and LSTM methods for securing webapplication against cross-site scripting attack", Indonesian Journal of Electrical Engineering and Computer Science Vol. 21, No. 2, February 2021.
- [2] Bronjon Gogoi, Tasiruddin Ahmed, and Hemanta Kumar Saikia. "Detection of XSS Attacks in Web Applications: A Machine Learning Approach", International Journal of Innovative Research in Computer Science & Technology (IJRCST) ISSN: 2347-5552, Volume-9, Issue-1, January 2021.
- [3] Caio Lente, Roberto Hirata Jr., Daniel Macedo Bastista. "An Improved Tool for Detection of XSS Attacks by Combining CNN with LSTM",
- [4] V. S Stency and N Mohanasundaram." A Study on XSS Attacks: *Intelligent Detection Methods*".
- [5] S.Shalini, S.Usha."Prevention of cross-site scripting attacks on web applications in the client side", Department of Computer and Communication, Sri Sairam Engineering College, Chennai- 44, Tamilnadu, India.
- [6] Jiss Varghese, Tressa Antony, and Liisha Varghese." An Enhanced Mechanism for Automated Removal of Cross-site Scripting Vulnerabilities using Input Validation", International Journal of Advance Research in Computer Science and Management Studies Volume 2, Issue 9, September 2014.
- [7] Open Web Application Security Project, "The ten most critical web application security vulnerabilities", 2007, www.owasp.org/index.php/OWASP_Top_Ten_Project
- [8] K. Fernandez and D. Pagkalos. Xssed.com - xss (cross-site scripting) information and vulnerable websites archive. [online], <http://xssed.com> (03/20/08).
- [9] Vishnu B A, Jevitha K P., " Prediction of Cross-Site Scripting Attack Using Machine Learning Algorithms", *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*. ACM, p.55, 2014.
- [10] Rathore S, Sharma P K, Park J H., " XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs", *Journal of Information Processing Systems*, vol. 13, no. 4, 2017.