GLOBAL ACADEMY OF TECHNOLOGY

Ideal Homes Township, Rajarajeshwari Nagar,
Bengaluru – 560 098

Final Project Presentation

DETECTION OF WEB ATTACKS BASED ON LOGS USING DEEP LEARNING

Domain Name:
Cyber security and deep learning

Group Number:
G-3

Presented By:

Architha.J	1GA17IS001
B.U.Kavya	1GA17IS005
Sri Harsha.A.	1GA17IS038
Varsha Venkatesh	1GA17IS045

Guided By:
Prof.Deepthi.V.S

Introduction

- Today’s interconnected world makes everyone more susceptible to cyber-attacks.
- Many authorized institutions like the financial institutions, military, government agencies etc. have confidential data that's stored on computers and transmitted to networks.
- With growing cyber-attacks, it has become necessary to safe guard this sensitive data and private information.
- As cybercriminals are getting more advanced, there is a requirement to know their change in target, how are that affecting organizations, and their methods used in targeting.
- Web applications do raise variety of security issues from improper cryptography. Serious vulnerabilities permit criminals to gain direct access to databases so as to steal sensitive data – this is often known as web attack.

INTRODUCTION

- Few security issues with web applications are XSS attacks, SQL injection, Local file inclusion, Path traversal and many more. Among these, SQLi and XSS are the two major web application attacks.
- Effects of these attacks include: Authentication bypass, information disclosure, compromised data integrity, user’s session hijacking, and access cookies.
- Objectives of proposed framework:

1. Detecting the presence of an attack on the website.
2. Classification of SQL injection attacks.
3. Classification of Cross-site scripting (XSS) attacks.
4. Prevention of SQL injection and Cross-site Scripting attacks.

LITERATURE SURVEY

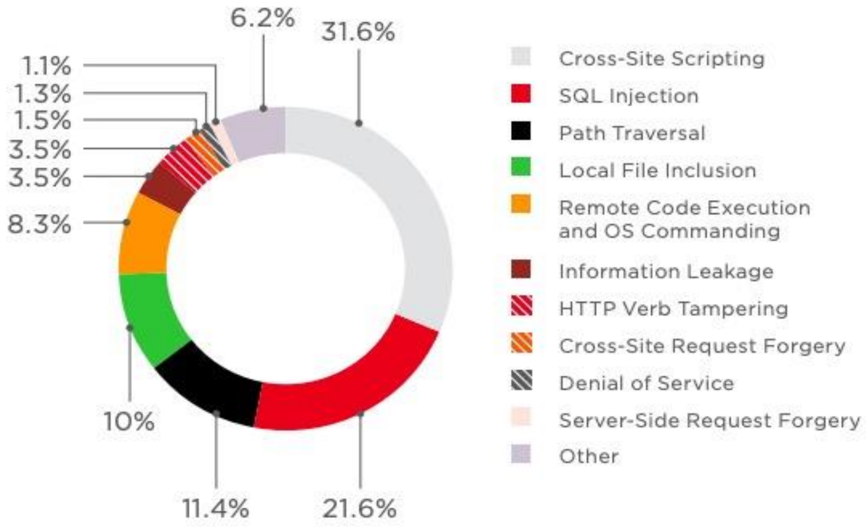
Title of the paper and year	Methodology	Merits	Demerits
SQL Injection Detection Using Machine Learning Year: (2019, San Jose State University)	Naïve Bayes Classifier and Gradient Boosting Classifier from ensemble machine learning approaches to classify and detect SQL Injection attacks	Identifies union based, error based and Boolean SQL injections Good accuracy	Gradient Boosting approach is computationally expensive than simple classifiers in terms of memory and computation. More memory is needed to store multiple trees.
Detection of SQL Injection Attacks: A Machine Learning Approach Year: (2019, ICECTA)	a machine learning based heuristic algorithm to prevent the SQL injection attack.	system is also scalable is a sense that any enhancement can be easily implemented with minor modification.	more non-injected SQL statements need to be considered in the dataset
Log based Dynamic Intrusion Detection of Web Applications Year(2019,Indian Institute of Technology, Kanpur)	A new machine learning based IDS which feeds on the log files of the web applications to detect attacks like SQLi , XSS, CRLF etc. (LR, SVM, gradient boosting , decision trees,KNN,random forest)	proposed work gives a good average detection accuracy of 99.02%	This can be extended to all the other areas like ‘Accept-Encoding,’ ‘Cookie,’ ‘Referrer’, etc.
SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN Year: (2019,School of Information and Communication Engineering)	This paper proposes elastic pooling CNN technique to detect SQL injections.	Based on the irregular matching characteristics, it can identify new attacks and is harder to bypass.	Regular technology has high recognition accuracy and speed, but it cannot identify new attacks. It is inevitable that new bypassing methods will emerge to avoid rules such as URL multiple encoding.

Title of the paper and year	Methodology	Merits	Demerits
Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query Year: (2018, IEEE)	The proposed workflow is to Compare the static and the dynamic query, And based on the result it is detected as SQL attack or not	Combining existing SQL injection detection mechanisms to develop more strong mechanism to make web application more robust	work on efficient methods for detecting the SQL injection attack and methods to prevent it. A less time must be consumed to detect the SQL injections.
Research on SQL injection detection technology based on SVM Year:(2018, MATEC Web of Conferences)	SVM algorithm word2vec method is selected to process the text data of HTTP requests	The proposed method effectively solves SQL injection mutation and overcomes the defects of the existing rule matching methods	Large numbers of dataset required Word2vec consumes large memory
Detecting Cross-Site Scripting Attacks using Machine Learning Year: (2018, City, University of London Institutional Repository)	This paper detects using KNN, RF, SVM algorithms and compares them to provide better accuracy.	Structural and behavioral functions of XSS are checked and therefore accuracy increased	Future works: Neural network classifier
Detecting Web Attacks Using Multi-Stage Log Analysis Year:(2017, San Jose State University, USA)	This work proposes a multi-stage log analysis architecture, which combines both pattern matching and supervised machine learning methods.	The trained model had good accuracy with less probability of false alarm	The techniques used are not compactable with the updated technologies.

Title of the paper and year	Methodology	Merits	Demerits
CROSS SITE SCRIPTING (XSS) attack detection using intrusion detection system Year: (2017,ICICCS)	Here they are introducing snort rule that can detect XSS Attack efficiently and create an alert entry for it in snort alert file.	After successful implementation of rule, XSS attack was able to be detected through Alert File containing details of the attack and attack was also logged in the form of dumps.	Few false-positive were generated, not able to see the rejected packets on Iptables output.
XSS Classifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs Year: (2017,Journal of information processing systems)	The proposed methods is usage of AD Trees and decorate algorithms to detect the attack.	The features selected are URL, HTML tags. In addition, The Social Networking services (SNSs) features are added to provided to enhance the accuracy.	Enhancement in feature set and usage of Advance learning algorithm such as deep learning.

Problem statement

- In today’s technical environment, cyber attacks have increased dramatically over the last decade, exposing sensitive personal and business information.
- The threat of attacks on web applications to extract data or to distribute malicious code persists.
- Few major security issues of web applications are XSS, SQL injection.
- Effects :Alter or modify the normal application behavior, authentication bypass, information disclosure, compromised data integrity, user’s session hijacking, access cookies.



System Requirements

- Hardware:

Processor: Intel Core i5 , RAM: 4GB or more , Hard disk: 50 GB or more ,

Display: Monitor(1024*768 pixels) or higher resolution monitor with 32 bit color settings.

- Software:

Operating System: Kali Linux, Windows

- Languages:

Programming language: Back - End: PYTHON

Programming language: Front - End: HTML,CSS,PHP

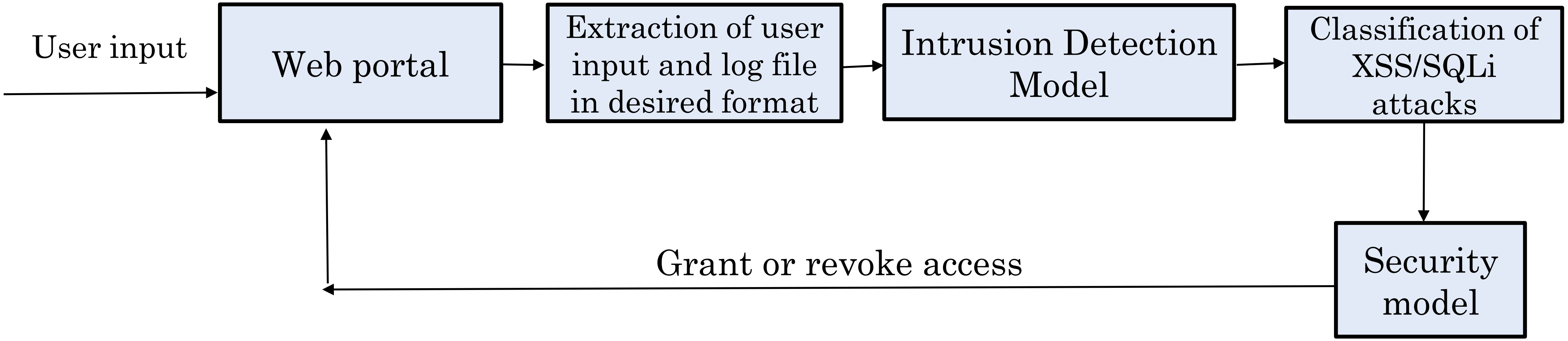
- Tools:

Development Environment: SUBLIME/ ANY OTHER PYTHON IDE

Application Server: XAMPP

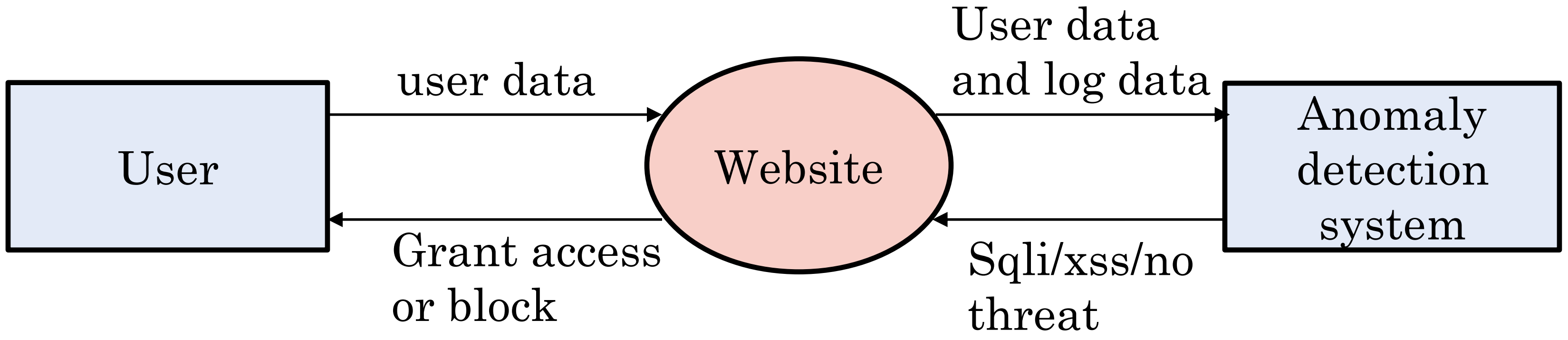
Database: MYSQL

Proposed System

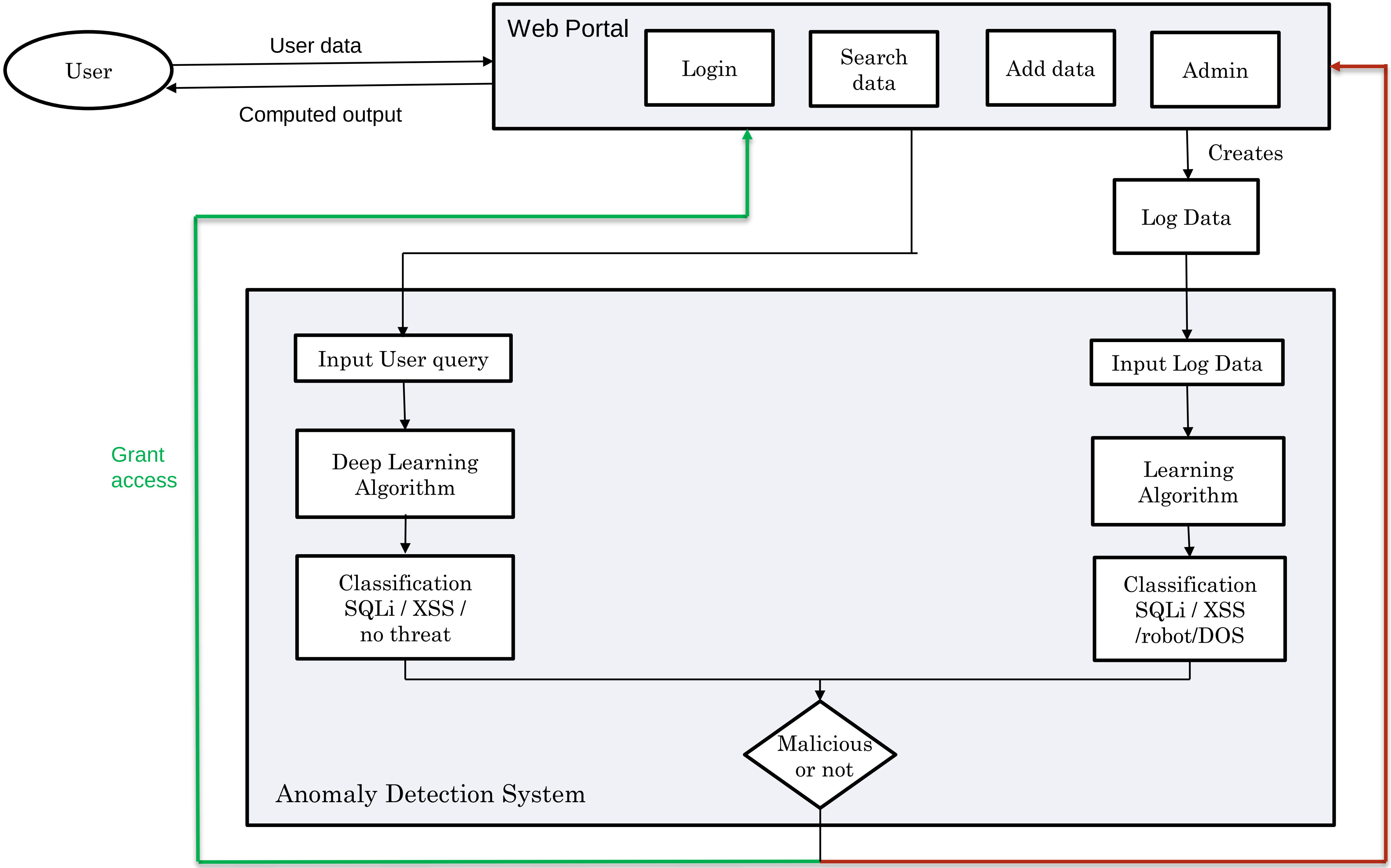

User input
Web portal
Extraction of user input and log file in desired format
Intrusion Detection Model
Classification of XSS/SQLi attacks
Security model
Grant or revoke access

DATA FLOW DIAGRAM

Level -0



Level -1



User data
User
Computed output
Web Portal
Login
Search data
Add data
Admin
Creates
Log Data
Input User query
Deep Learning Algorithm
Classification SQLi / XSS / no threat
Input Log Data
Learning Algorithm
Classification SQLi / XSS /robot/DOS
Malicious or not
Anomaly Detection System
Grant access
Revoke access

Gantt Chart

Task Name	Oct 2nd week	Oct 2nd week-4th week	Nov 1st week-4th week	Dec 1st week- 4th week	Jan 1st week-4th week	March1st week-4th week	April 1st week-4th week	May 1st week – june 4th week	July 2nd week
Identification of problem statement	■								
Approval of title and project synopsys		■							
Literature survey			■	■					
Design					■				
Coding and Implementation						■	■	■	
Testing and debugging							■	■	
Reporting and documentation									■

Detailed Design

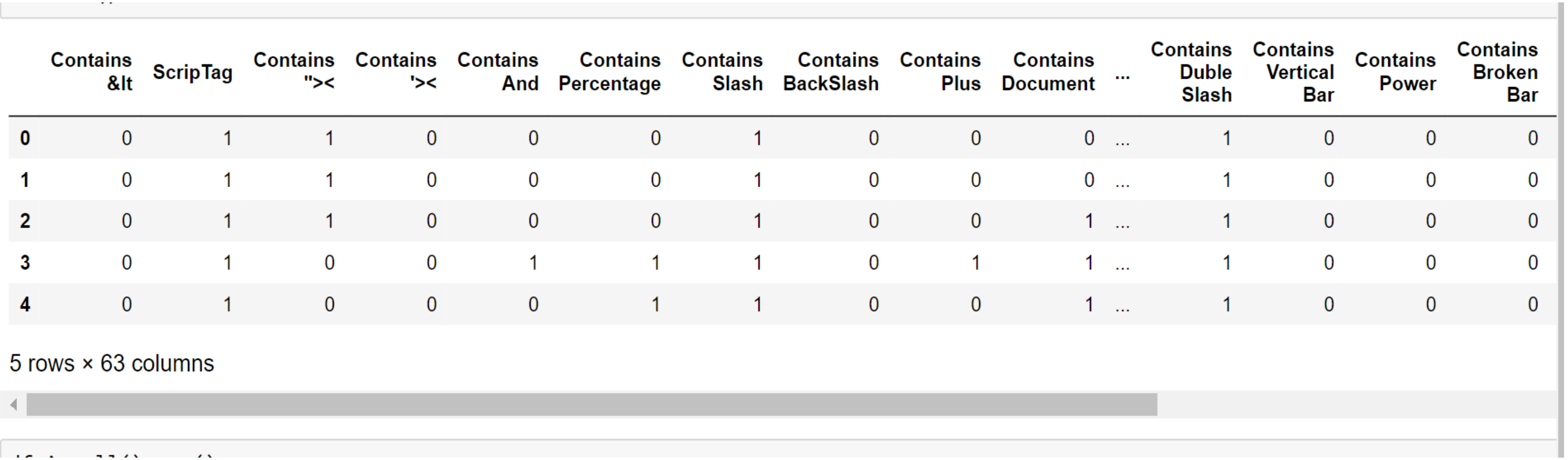
- Module - 1
  - Module Name: Classification of query string
  - Description: Analysis of query string and classifying it into sqli, xss or a normal input.
  - Methodology:

❖ Sub module 1: Detection of SQLi

➢ A dataset was prepared using the kali-Linux tool “sqlmap”. Sqlmap allows injecting malicious queries into our website, since some queries cannot be included by the tool these are included manually.

➢ The dataset is preprocessed through CountVectorizer available in sklearn.feature_extraction.text. The CountVectorizer provides a simple way to both tokenize a collection of query statements and build a vocabulary of known words. This encoded vector is a sparse matrix that is provided as input to the multilayer neural network for training and testing.

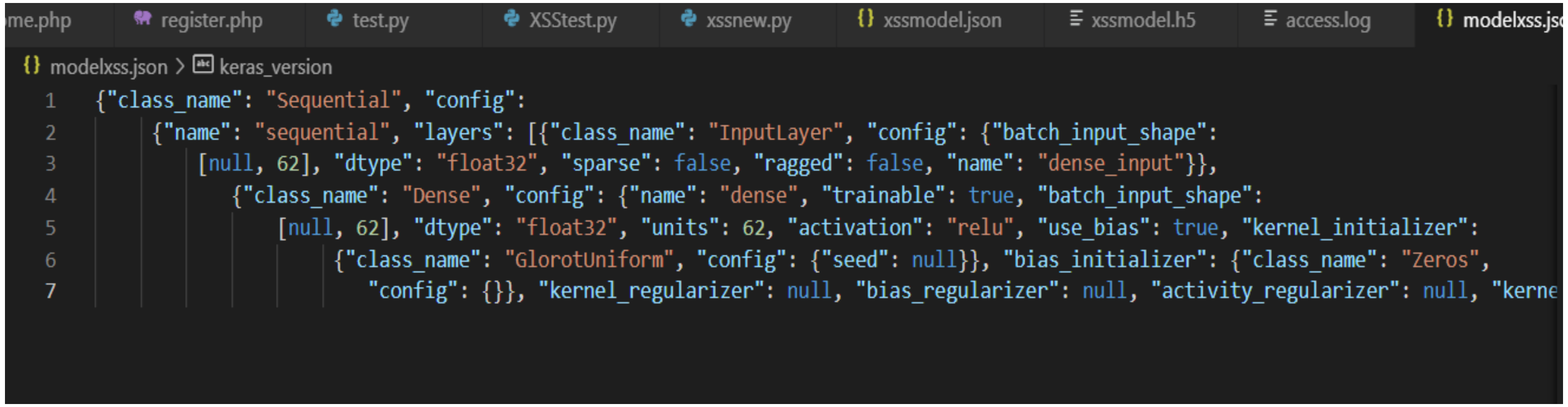
- This model is saved into a JSON file and weights are stored in the hdf5 file.
- The model is loaded to predict the preprocessed test instance into a binary classification. Based on the result provided to the front end, web portal can grant or deny access to the user.
- Sub module 2: Detection of XSS
- A dataset was prepared using the kali-Linux tool “owasp”. Since the dataset contains Chinese characters such as œ,□š, ^, Ÿ, it is necessary to remove them, this can be done by converting it into ASCII characters for the preprocessing of data.
- These values are stored into numeric array are passed into convoluted neural network model for training and testing. For Further use, the model is saved into a JSON file and weights are stored in the hdf5 file and is loaded to predict the preprocessed test instance into a binary classification.
- Another model, with 63 feature set was built to detect xss attacks. The features are selected manually and passed through a multi-layer perceptron. The model is saved and loaded when the script is called.
- Input: Query string
- Output: Binary classification(Malicious or not)



	Contains &lt	ScripTag	Contains "><	Contains '><	Contains And	Contains Percentage	Contains Slash	Contains BackSlash	Contains Plus	Contains Document	...	Contains Duble Slash	Contains Vertical Bar	Contains Power	Contains Broken Bar
0	0	1	1	0	0	0	1	0	0	0	...	1	0	0	0
1	0	1	1	0	0	0	1	0	0	0	...	1	0	0	0
2	0	1	1	0	0	0	1	0	0	1	...	1	0	0	0
3	0	1	0	0	1	1	1	0	1	1	...	1	0	0	0
4	0	1	0	0	0	1	1	0	0	1	...	1	0	0	0

5 rows × 63 columns

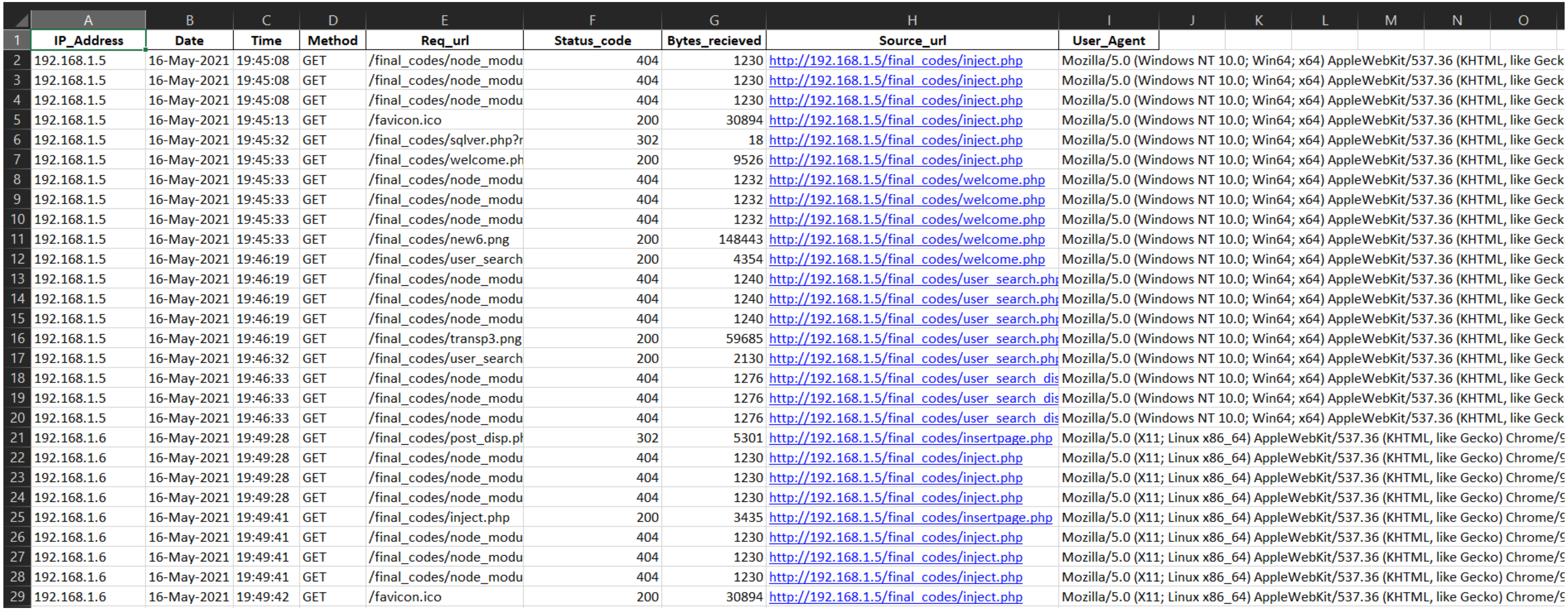
Feature set



```
{"class_name": "Sequential", "config":
    {"name": "sequential", "layers": [{"class_name": "InputLayer", "config": {"batch_input_shape":
        [null, 62], "dtype": "float32", "sparse": false, "ragged": false, "name": "dense_input"}},
            {"class_name": "Dense", "config": {"name": "dense", "trainable": true, "batch_input_shape":
                [null, 62], "dtype": "float32", "units": 62, "activation": "relu", "use_bias": true, "kernel_initializer":
                    {"class_name": "GlorotUniform", "config": {"seed": null}}, "bias_initializer": {"class_name": "Zeros",
                        "config": {}}, "kernel_regularizer": null, "bias_regularizer": null, "activity_regularizer": null, "kerne
```

.json format

- Module - 2
  - Module Name: Classification using log data
  - Description: Analysis of log data from the website and classifying it into SQLi, XSSS, DOS or a normal input.
  - Methodology:
  - All the requests sent to a server will be stored in access.log file. Using Apache log parser, the log data is split into required format. The following fields: IP address, Date, Time, Method, Requested URL, Status code, number of Bytes received, Source URL and User agent are extracted from the log file and stored in pandas dataframe. Based on the values of different fields and patterns in log data, the log analysis is done to classify the type of request into DOS, Human, Robot, SQLi and XSS request. Based on certain features such as number of requests, source URL and other features, the requests can be classified into DDOS, robot, SQLi, XSS.
  - Input: Log file data
  - Output: Classification of log data into DOS, Robot, SQLi, XSS.



	IP_Address	Date	Time	Method	Req_url	Status_code	Bytes_recieved	Source_url	User_Agent
2	192.168.1.5	16-May-2021	19:45:08	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
3	192.168.1.5	16-May-2021	19:45:08	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
4	192.168.1.5	16-May-2021	19:45:08	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
5	192.168.1.5	16-May-2021	19:45:13	GET	/favicon.ico	200	30894	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
6	192.168.1.5	16-May-2021	19:45:32	GET	/final_codes/sqlver.php?r	302	18	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
7	192.168.1.5	16-May-2021	19:45:33	GET	/final_codes/welcome.ph	200	9526	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
8	192.168.1.5	16-May-2021	19:45:33	GET	/final_codes/node_modu	404	1232	http://192.168.1.5/final_codes/welcome.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
9	192.168.1.5	16-May-2021	19:45:33	GET	/final_codes/node_modu	404	1232	http://192.168.1.5/final_codes/welcome.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
10	192.168.1.5	16-May-2021	19:45:33	GET	/final_codes/node_modu	404	1232	http://192.168.1.5/final_codes/welcome.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
11	192.168.1.5	16-May-2021	19:45:33	GET	/final_codes/new6.png	200	148443	http://192.168.1.5/final_codes/welcome.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
12	192.168.1.5	16-May-2021	19:46:19	GET	/final_codes/user_search	200	4354	http://192.168.1.5/final_codes/welcome.php	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
13	192.168.1.5	16-May-2021	19:46:19	GET	/final_codes/node_modu	404	1240	http://192.168.1.5/final_codes/user_search.ph	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
14	192.168.1.5	16-May-2021	19:46:19	GET	/final_codes/node_modu	404	1240	http://192.168.1.5/final_codes/user_search.ph	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
15	192.168.1.5	16-May-2021	19:46:19	GET	/final_codes/node_modu	404	1240	http://192.168.1.5/final_codes/user_search.ph	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
16	192.168.1.5	16-May-2021	19:46:19	GET	/final_codes/transp3.png	200	59685	http://192.168.1.5/final_codes/user_search.ph	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
17	192.168.1.5	16-May-2021	19:46:32	GET	/final_codes/user_search	200	2130	http://192.168.1.5/final_codes/user_search.ph	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
18	192.168.1.5	16-May-2021	19:46:33	GET	/final_codes/node_modu	404	1276	http://192.168.1.5/final_codes/user_search_dis	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
19	192.168.1.5	16-May-2021	19:46:33	GET	/final_codes/node_modu	404	1276	http://192.168.1.5/final_codes/user_search_dis	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
20	192.168.1.5	16-May-2021	19:46:33	GET	/final_codes/node_modu	404	1276	http://192.168.1.5/final_codes/user_search_dis	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
21	192.168.1.6	16-May-2021	19:49:28	GET	/final_codes/post_disp.ph	302	5301	http://192.168.1.5/final_codes/insertpage.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
22	192.168.1.6	16-May-2021	19:49:28	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
23	192.168.1.6	16-May-2021	19:49:28	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
24	192.168.1.6	16-May-2021	19:49:28	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
25	192.168.1.6	16-May-2021	19:49:41	GET	/final_codes/inject.php	200	3435	http://192.168.1.5/final_codes/insertpage.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
26	192.168.1.6	16-May-2021	19:49:41	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
27	192.168.1.6	16-May-2021	19:49:41	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
28	192.168.1.6	16-May-2021	19:49:41	GET	/final_codes/node_modu	404	1230	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
29	192.168.1.6	16-May-2021	19:49:42	GET	/favicon.ico	200	30894	http://192.168.1.5/final_codes/inject.php	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9

Log data in data.xlsx

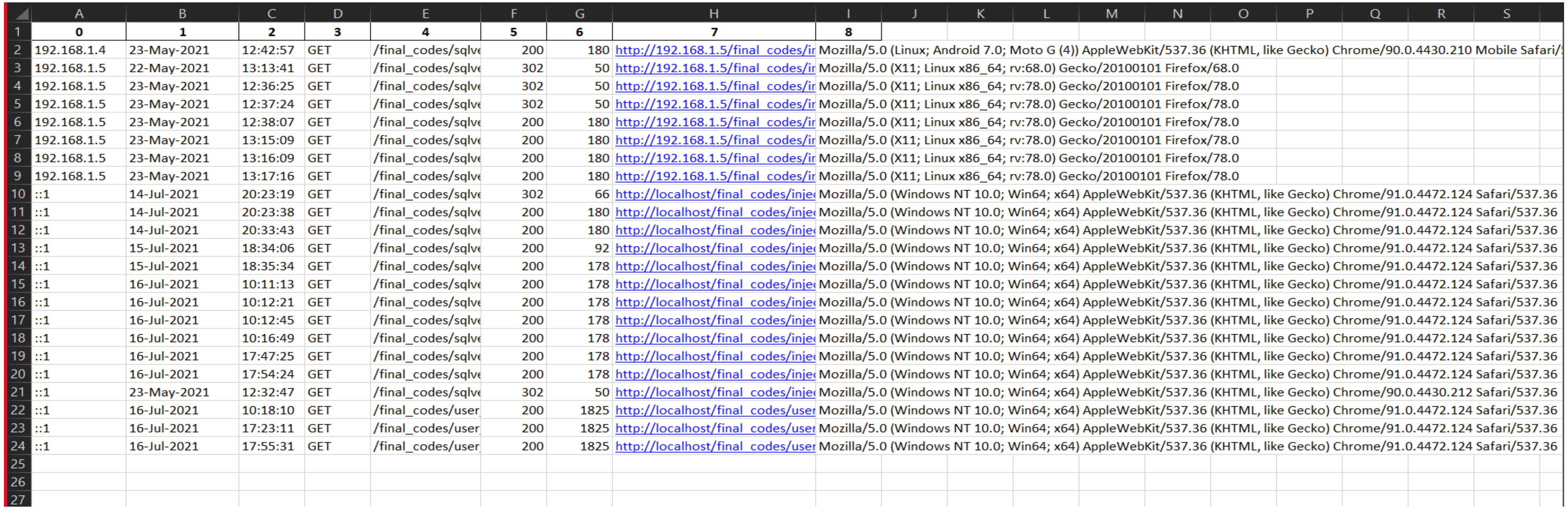
IMPLEMENTATION

- Detection of SQLi using Multilayer Neural network
  - A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). This sequential model can be constructed through Keras which is a open source software library. The layers of the neural network can be constructed through “layers” module in Keras.
  - The model consist of 1 input layer , 2 hidden layers and 1 output layer. The output from hidden layers are batch normalized and passed through dropout layer which removes certain neurons from a neural network at each training step to avoid overfitting of the model.
  - The input layer consists of 20 neurons and ‘relu’ activation function is used. The first hidden layer consists of 10 neurons and ‘tanh’ activation function is used. The second hidden layer consists of 1024 hidden neurons with ‘relu’ activation function. The output layer consist of one neuron with sigmoid activation function to classify as Malicious or not.

- Detection of XSS using convoluted neural network
  - A convolutional neural network is a special kind of feedforward neural network with fewer weights than a fully-connected network. TensorFlow is an open-source library of software for dataflow and differential programing for various tasks. Keras is a open source software library. A Convoluted Neural Network can be created by using Tensorflow and keras.
  - A 11 layer network is constructed which consist of 1 input layer, 5 hidden layers with 3 maxpooling and 1 flatten layer, and 1 output layer.
  - The input layer consists of 64 neurons and relu activation function. The first hidden layer has 128 neurons and second & third hidden layer consists of 256 with relu activation function. The fourth and fifth hidden layer consists of 128 and 64 hidden neurons with ‘relu’ activation function. The output layer consist of one neuron with sigmoid activation function to classify as Malicious or not.

- Detection of XSS using multilayer perceptron
  - The features are manually extracted through a script and these extracted features are provided to a multilayer perceptron.
  - The model consist of 1 input layer , 2 hidden layers and 1 output layer.
  - The model uses binary cross entropy as the loss function with adam optimizer.
  - Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.
  - Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

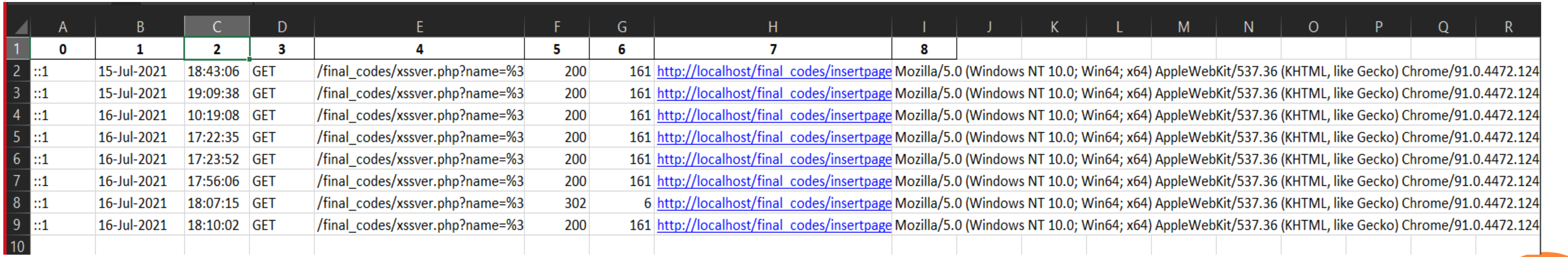
- Log Analysis
  - For identification of DOS: The Source URL, User agent will be blank and status code is 400 and number of requests from a particular IP at a particular instance of time will be more than the usual value.
  - For identification of robot: The source URL will be blank. The user agent may be:
    1. sqlmap, owasp or any such automation tools.
    2. Browsers like Chromium, Firefox, etc.
  - If the user agent is an automation tool the request can be classified directly into Robot. Else, The source URL will be blank and number of request will be high for both type of User agent. All request apart from robot and DOS is considered to be requested by humans.
  - For identification of SQLi attack: The URL contains the data entered by the user which is decoded and passed to the deep learning algorithm to classify into malicious and benign.
  - For identification of XSS attack: The URL contains the data entered by the user which is decoded and passed to the deep learning algorithm to classify into malicious and benign.



0	1	2	3	4	5	6	7	8
192.168.1.4	23-May-2021	12:42:57	GET	/final_codes/sqlve	200	180	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (Linux; Android 7.0; Moto G (4)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.210 Mobile Safari/
192.168.1.5	22-May-2021	13:13:41	GET	/final_codes/sqlve	302	50	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
192.168.1.5	23-May-2021	12:36:25	GET	/final_codes/sqlve	302	50	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
192.168.1.5	23-May-2021	12:37:24	GET	/final_codes/sqlve	302	50	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
192.168.1.5	23-May-2021	12:38:07	GET	/final_codes/sqlve	200	180	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
192.168.1.5	23-May-2021	13:15:09	GET	/final_codes/sqlve	200	180	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
192.168.1.5	23-May-2021	13:16:09	GET	/final_codes/sqlve	200	180	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
192.168.1.5	23-May-2021	13:17:16	GET	/final_codes/sqlve	200	180	http://192.168.1.5/final_codes/ir	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
::1	14-Jul-2021	20:23:19	GET	/final_codes/sqlve	302	66	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	14-Jul-2021	20:23:38	GET	/final_codes/sqlve	200	180	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	14-Jul-2021	20:33:43	GET	/final_codes/sqlve	200	180	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	15-Jul-2021	18:34:06	GET	/final_codes/sqlve	200	92	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	15-Jul-2021	18:35:34	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	10:11:13	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	10:12:21	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	10:12:45	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	10:16:49	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	17:47:25	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	17:54:24	GET	/final_codes/sqlve	200	178	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	23-May-2021	12:32:47	GET	/final_codes/sqlve	302	50	http://localhost/final_codes/inje	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
::1	16-Jul-2021	10:18:10	GET	/final_codes/user	200	1825	http://localhost/final_codes/user	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	17:23:11	GET	/final_codes/user	200	1825	http://localhost/final_codes/user	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
::1	16-Jul-2021	17:55:31	GET	/final_codes/user	200	1825	http://localhost/final_codes/user	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36

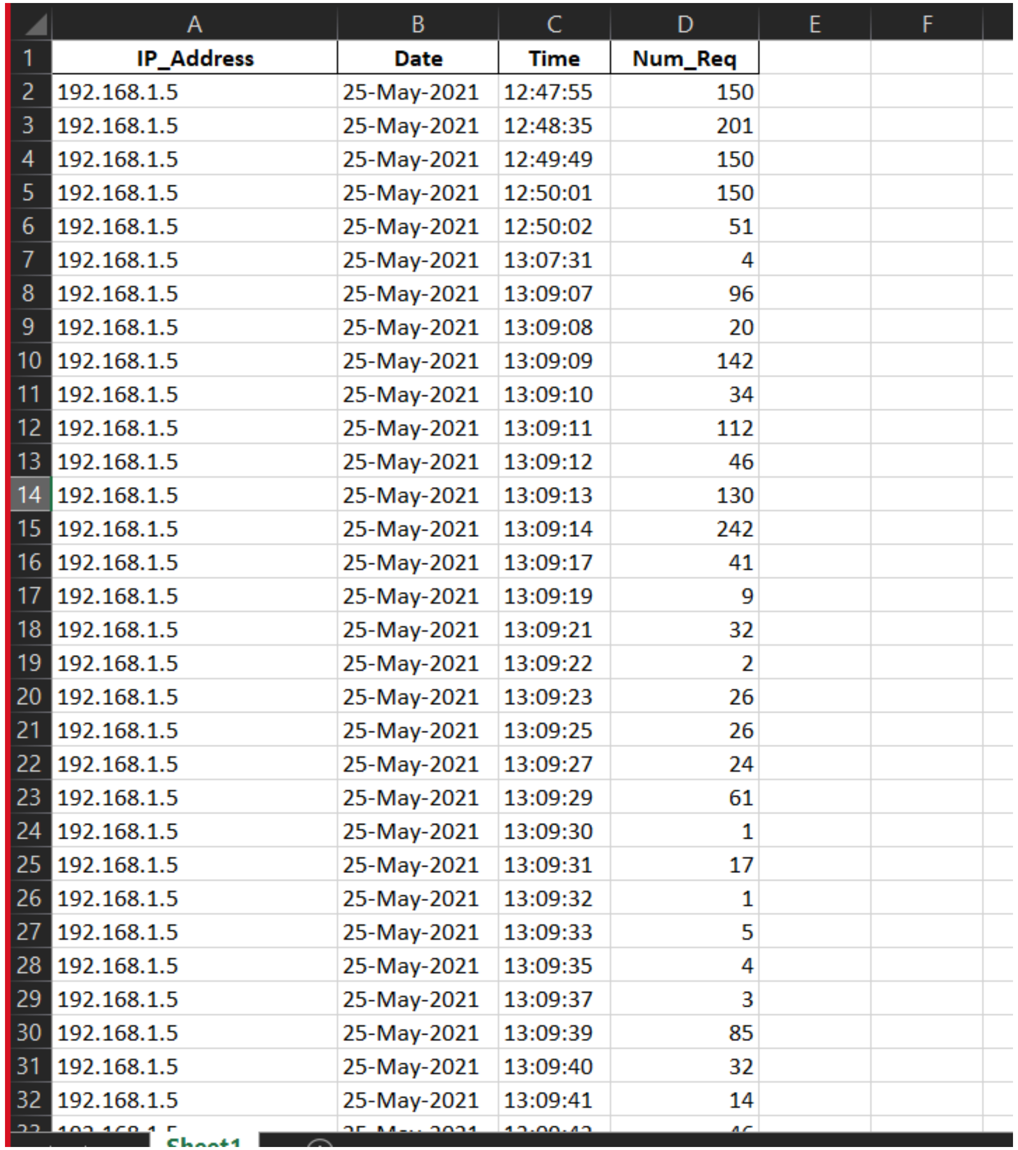
Sqli attack requests

17ISP85



0	1	2	3	4	5	6	7	8
::1	15-Jul-2021	18:43:06	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	15-Jul-2021	19:09:38	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	10:19:08	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	17:22:35	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	17:23:52	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	17:56:06	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	18:07:15	GET	/final_codes/xssver.php?name=%3	302	6	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124
::1	16-Jul-2021	18:10:02	GET	/final_codes/xssver.php?name=%3	200	161	http://localhost/final_codes/insertpage	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124

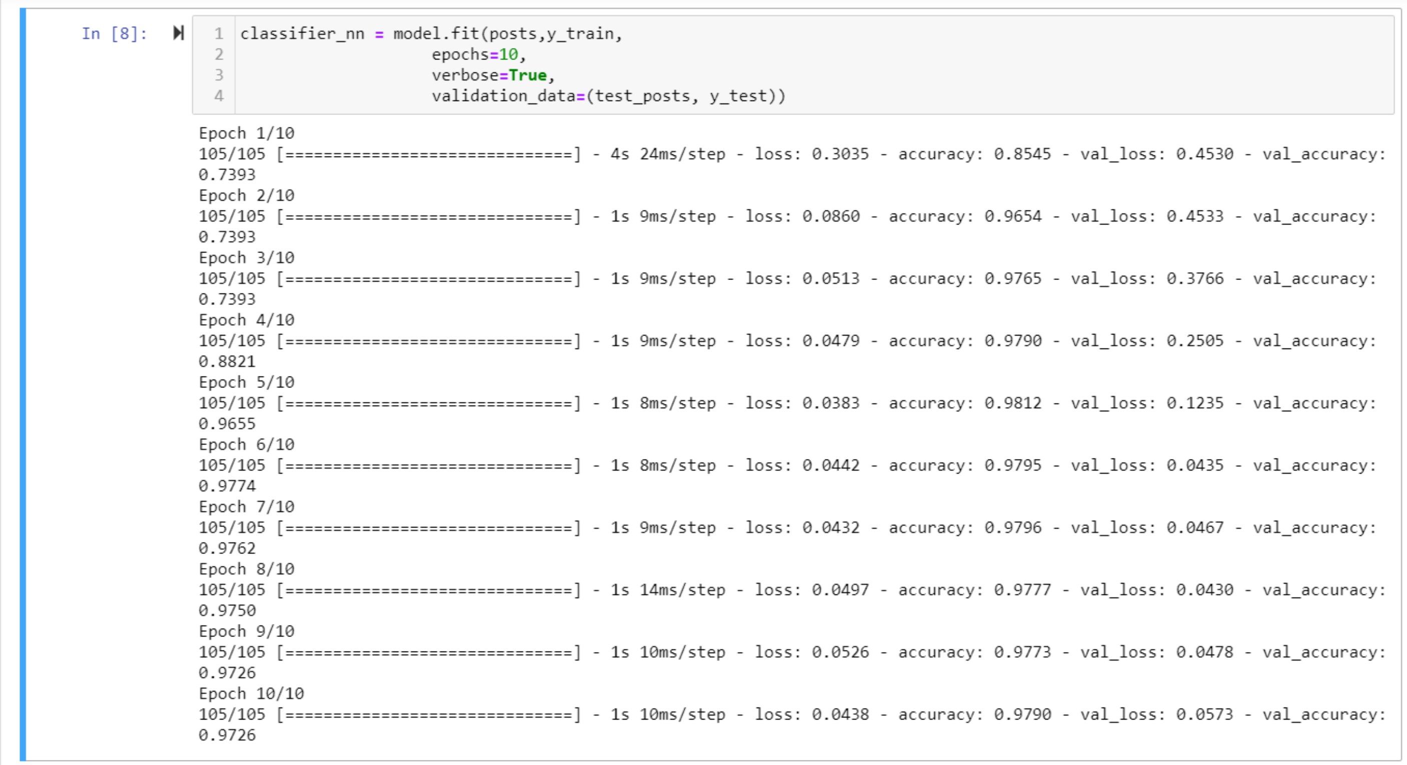
Xss attack
requests



IP_Address	Date	Time	Num_Req
192.168.1.5	25-May-2021	12:47:55	150
192.168.1.5	25-May-2021	12:48:35	201
192.168.1.5	25-May-2021	12:49:49	150
192.168.1.5	25-May-2021	12:50:01	150
192.168.1.5	25-May-2021	12:50:02	51
192.168.1.5	25-May-2021	13:07:31	4
192.168.1.5	25-May-2021	13:09:07	96
192.168.1.5	25-May-2021	13:09:08	20
192.168.1.5	25-May-2021	13:09:09	142
192.168.1.5	25-May-2021	13:09:10	34
192.168.1.5	25-May-2021	13:09:11	112
192.168.1.5	25-May-2021	13:09:12	46
192.168.1.5	25-May-2021	13:09:13	130
192.168.1.5	25-May-2021	13:09:14	242
192.168.1.5	25-May-2021	13:09:17	41
192.168.1.5	25-May-2021	13:09:19	9
192.168.1.5	25-May-2021	13:09:21	32
192.168.1.5	25-May-2021	13:09:22	2
192.168.1.5	25-May-2021	13:09:23	26
192.168.1.5	25-May-2021	13:09:25	26
192.168.1.5	25-May-2021	13:09:27	24
192.168.1.5	25-May-2021	13:09:29	61
192.168.1.5	25-May-2021	13:09:30	1
192.168.1.5	25-May-2021	13:09:31	17
192.168.1.5	25-May-2021	13:09:32	1
192.168.1.5	25-May-2021	13:09:33	5
192.168.1.5	25-May-2021	13:09:35	4
192.168.1.5	25-May-2021	13:09:37	3
192.168.1.5	25-May-2021	13:09:39	85
192.168.1.5	25-May-2021	13:09:40	32
192.168.1.5	25-May-2021	13:09:41	14
192.168.1.5	25-May-2021	13:09:43	46

Dos attack
requests

Experimental Results

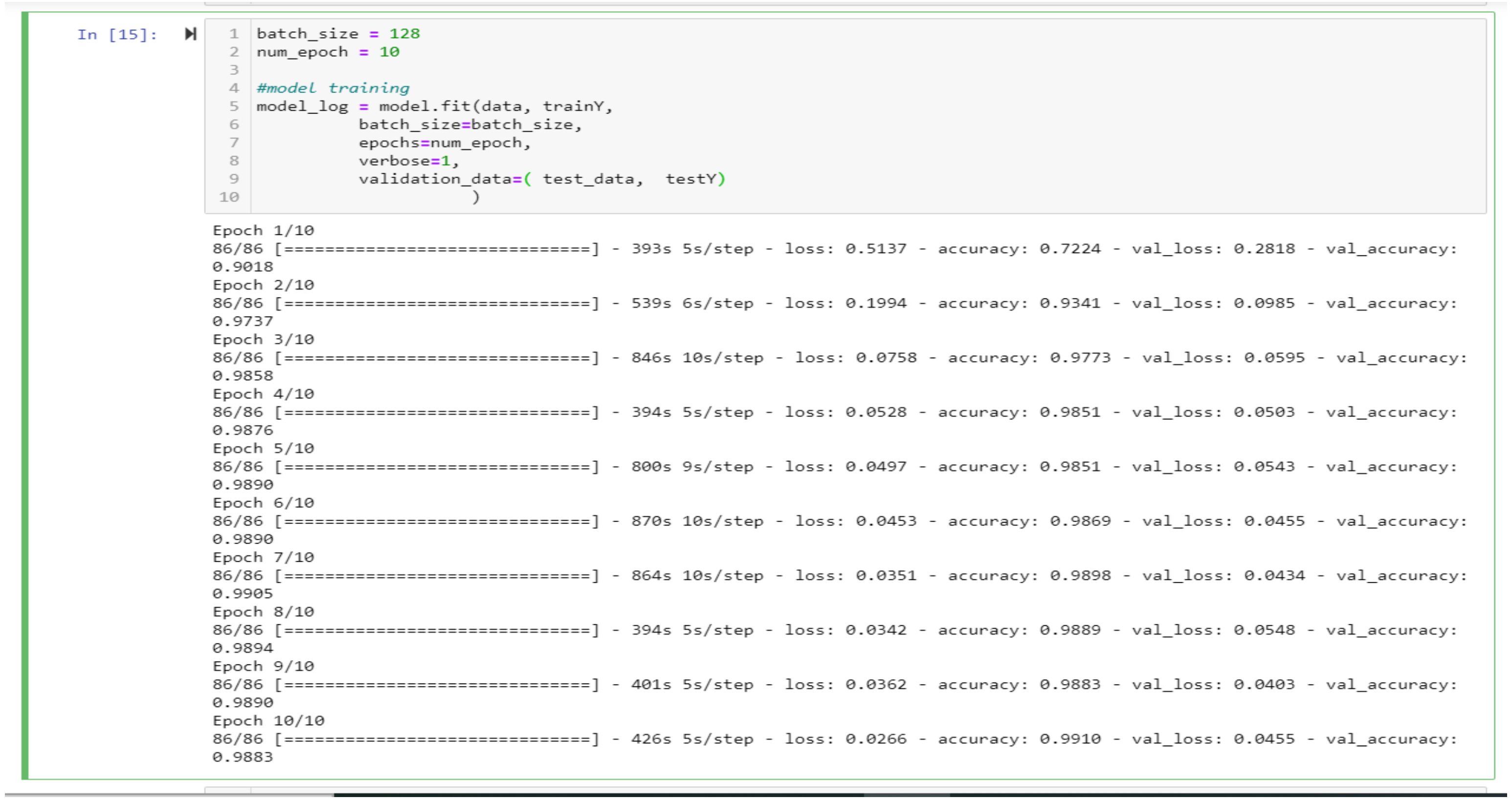


```
classifier_nn = model.fit(posts,y_train,
                    epochs=10,
                    verbose=True,
                    validation_data=(test_posts, y_test))
```

```
Epoch 1/10
105/105 [==============================] - 4s 24ms/step - loss: 0.3035 - accuracy: 0.8545 - val_loss: 0.4530 - val_accuracy:
0.7393
Epoch 2/10
105/105 [==============================] - 1s 9ms/step - loss: 0.0860 - accuracy: 0.9654 - val_loss: 0.4533 - val_accuracy:
0.7393
Epoch 3/10
105/105 [==============================] - 1s 9ms/step - loss: 0.0513 - accuracy: 0.9765 - val_loss: 0.3766 - val_accuracy:
0.7393
Epoch 4/10
105/105 [==============================] - 1s 9ms/step - loss: 0.0479 - accuracy: 0.9790 - val_loss: 0.2505 - val_accuracy:
0.8821
Epoch 5/10
105/105 [==============================] - 1s 8ms/step - loss: 0.0383 - accuracy: 0.9812 - val_loss: 0.1235 - val_accuracy:
0.9655
Epoch 6/10
105/105 [==============================] - 1s 8ms/step - loss: 0.0442 - accuracy: 0.9795 - val_loss: 0.0435 - val_accuracy:
0.9774
Epoch 7/10
105/105 [==============================] - 1s 9ms/step - loss: 0.0432 - accuracy: 0.9796 - val_loss: 0.0467 - val_accuracy:
0.9762
Epoch 8/10
105/105 [==============================] - 1s 14ms/step - loss: 0.0497 - accuracy: 0.9777 - val_loss: 0.0430 - val_accuracy:
0.9750
Epoch 9/10
105/105 [==============================] - 1s 10ms/step - loss: 0.0526 - accuracy: 0.9773 - val_loss: 0.0478 - val_accuracy:
0.9726
Epoch 10/10
105/105 [==============================] - 1s 10ms/step - loss: 0.0438 - accuracy: 0.9790 - val_loss: 0.0573 - val_accuracy:
0.9726
```

Multilayer Neural network

Accuracy	0.9726190476190476
Precision	1.0
Recall	0.8949771689497716
Confusion Matrix	[[621, 23], [0, 196]]



In [15]:

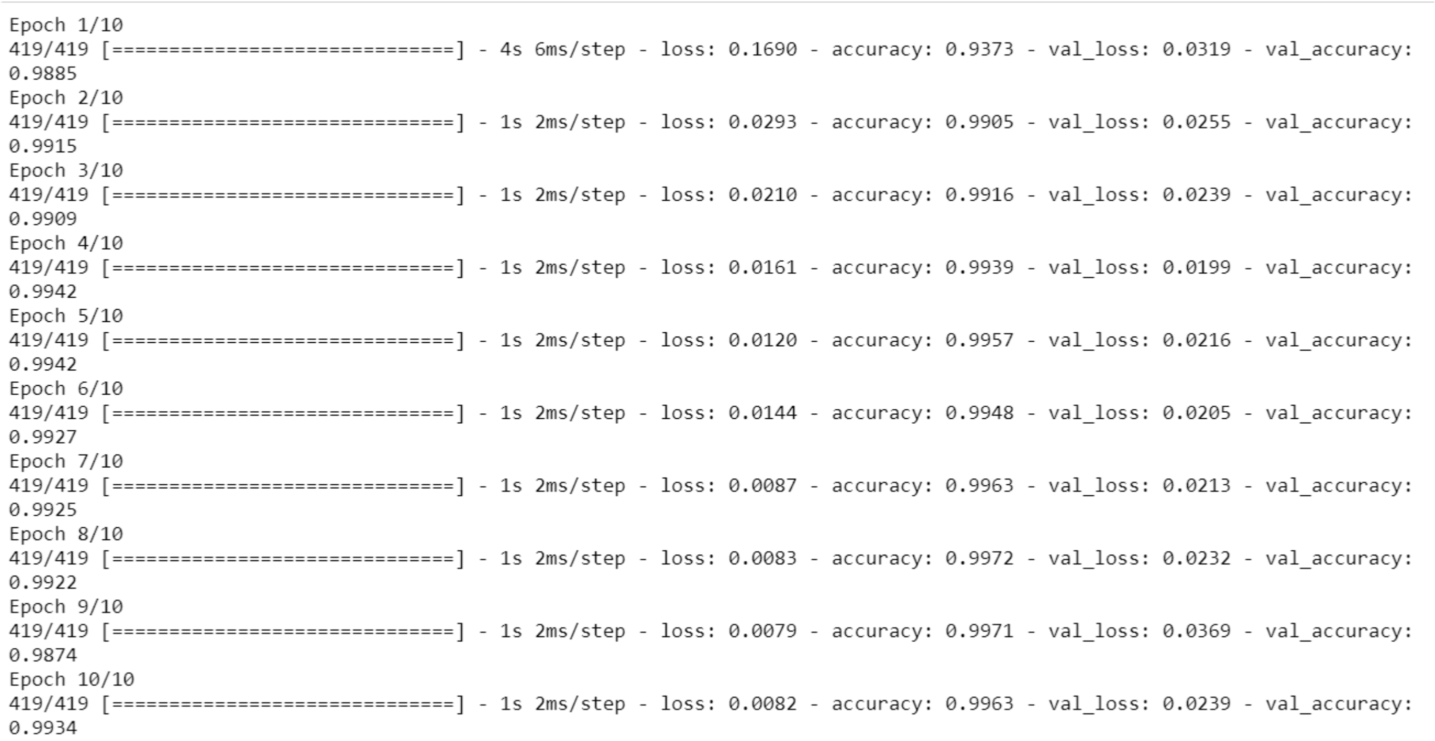
```
batch_size = 128
num_epoch = 10

#model training
model_log = model.fit(data, trainY,
          batch_size=batch_size,
          epochs=num_epoch,
          verbose=1,
          validation_data=( test_data,  testY)
                     )
```

```
Epoch 1/10
86/86 [==============================] - 393s 5s/step - loss: 0.5137 - accuracy: 0.7224 - val_loss: 0.2818 - val_accuracy:
0.9018
Epoch 2/10
86/86 [==============================] - 539s 6s/step - loss: 0.1994 - accuracy: 0.9341 - val_loss: 0.0985 - val_accuracy:
0.9737
Epoch 3/10
86/86 [==============================] - 846s 10s/step - loss: 0.0758 - accuracy: 0.9773 - val_loss: 0.0595 - val_accuracy:
0.9858
Epoch 4/10
86/86 [==============================] - 394s 5s/step - loss: 0.0528 - accuracy: 0.9851 - val_loss: 0.0503 - val_accuracy:
0.9876
Epoch 5/10
86/86 [==============================] - 800s 9s/step - loss: 0.0497 - accuracy: 0.9851 - val_loss: 0.0543 - val_accuracy:
0.9890
Epoch 6/10
86/86 [==============================] - 870s 10s/step - loss: 0.0453 - accuracy: 0.9869 - val_loss: 0.0455 - val_accuracy:
0.9890
Epoch 7/10
86/86 [==============================] - 864s 10s/step - loss: 0.0351 - accuracy: 0.9898 - val_loss: 0.0434 - val_accuracy:
0.9905
Epoch 8/10
86/86 [==============================] - 394s 5s/step - loss: 0.0342 - accuracy: 0.9889 - val_loss: 0.0548 - val_accuracy:
0.9894
Epoch 9/10
86/86 [==============================] - 401s 5s/step - loss: 0.0362 - accuracy: 0.9883 - val_loss: 0.0403 - val_accuracy:
0.9890
Epoch 10/10
86/86 [==============================] - 426s 5s/step - loss: 0.0266 - accuracy: 0.9910 - val_loss: 0.0455 - val_accuracy:
0.9883
```

Convoluted Neural Network

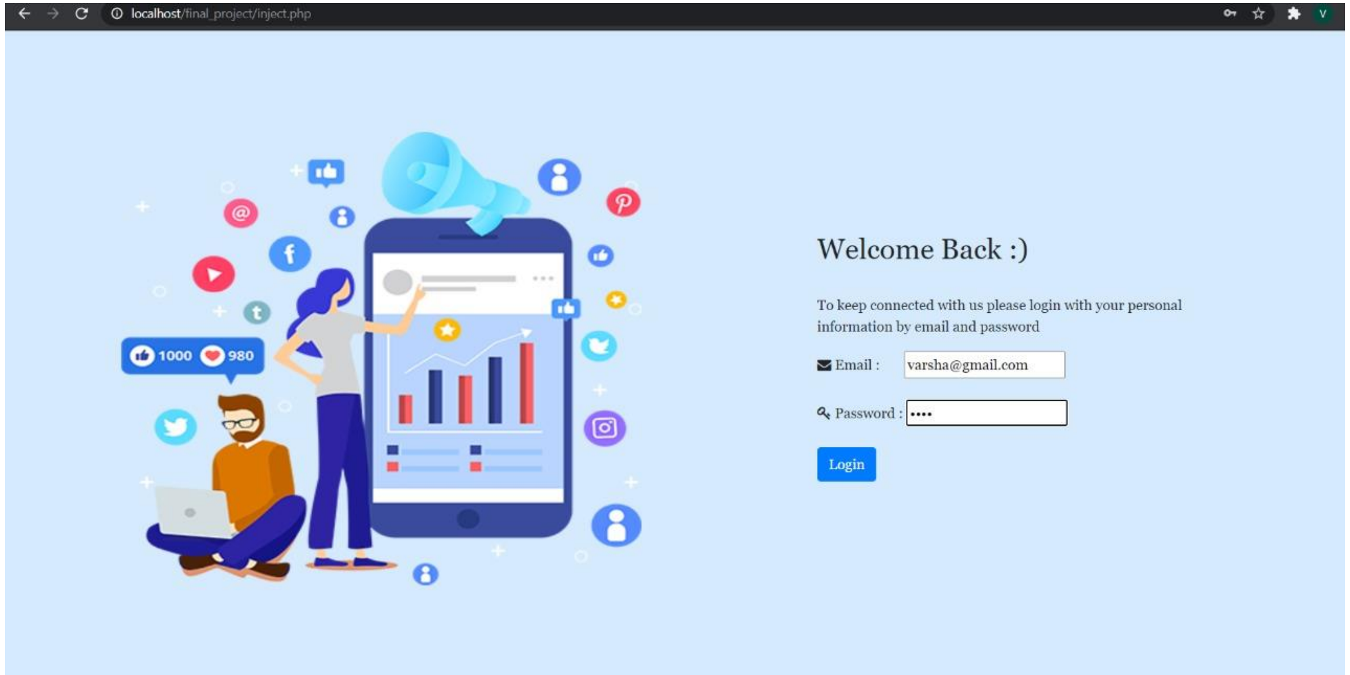
Accuracy	0.9883387874360847
Precision	0.992537313432835 8
Recall	0.9891818796484111
Confusion Matrix	[[1248, 16], [11, 1463]]



```
Epoch 1/10
419/419 [==============================] - 4s 6ms/step - loss: 0.1690 - accuracy: 0.9373 - val_loss: 0.0319 - val_accuracy:
0.9885
Epoch 2/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0293 - accuracy: 0.9905 - val_loss: 0.0255 - val_accuracy:
0.9915
Epoch 3/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0210 - accuracy: 0.9916 - val_loss: 0.0239 - val_accuracy:
0.9909
Epoch 4/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0161 - accuracy: 0.9939 - val_loss: 0.0199 - val_accuracy:
0.9942
Epoch 5/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0120 - accuracy: 0.9957 - val_loss: 0.0216 - val_accuracy:
0.9942
Epoch 6/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0144 - accuracy: 0.9948 - val_loss: 0.0205 - val_accuracy:
0.9927
Epoch 7/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0087 - accuracy: 0.9963 - val_loss: 0.0213 - val_accuracy:
0.9925
Epoch 8/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0083 - accuracy: 0.9972 - val_loss: 0.0232 - val_accuracy:
0.9922
Epoch 9/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0079 - accuracy: 0.9971 - val_loss: 0.0369 - val_accuracy:
0.9874
Epoch 10/10
419/419 [==============================] - 1s 2ms/step - loss: 0.0082 - accuracy: 0.9963 - val_loss: 0.0239 - val_accuracy:
0.9934
```

Multilayer Perceptron model for XSS

Accuracy :	0.9877763290918599
Precision :	0.9824561403508771
Recall :	0.9936628643852978
Confusion matrix:	[[4131, 10], [28, 1568]]



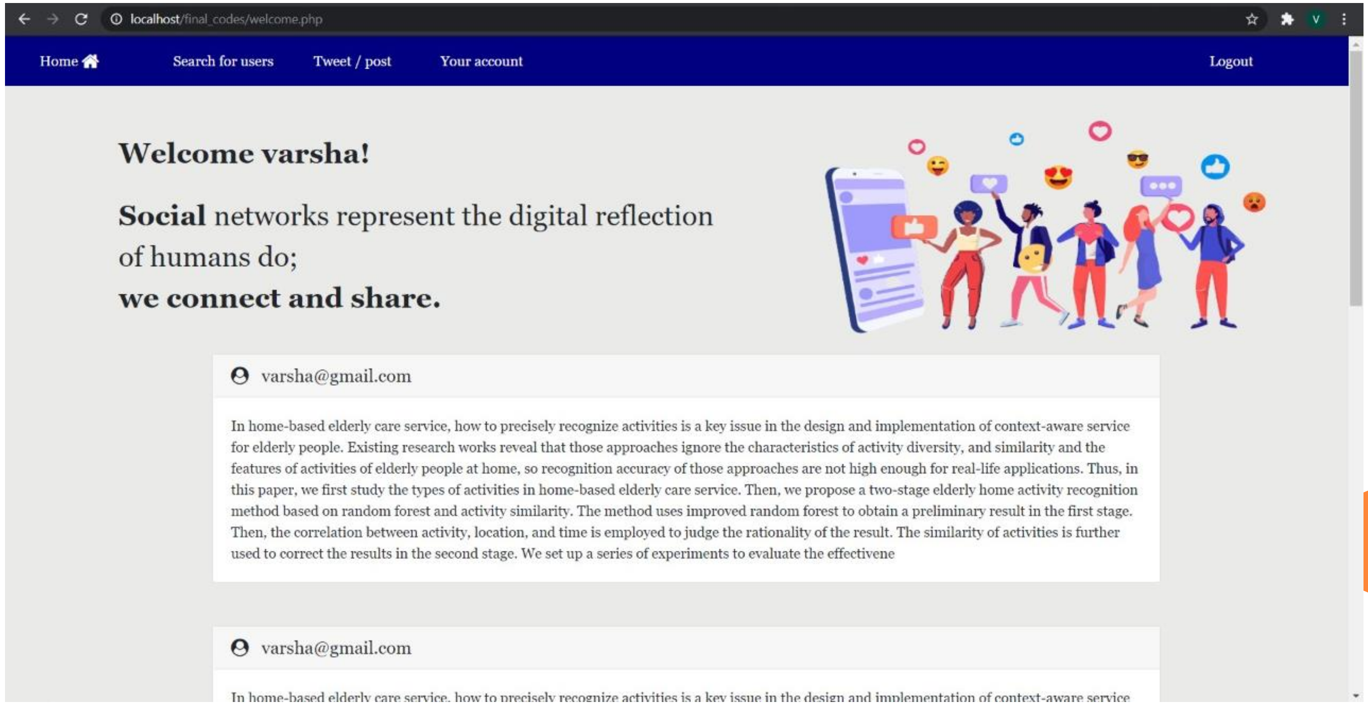
Welcome Back :)

To keep connected with us please login with your personal information by email and password

Email : varsha@gmail.com

Password : ••••

Login



Welcome varsha!

Social networks represent the digital reflection of humans do;

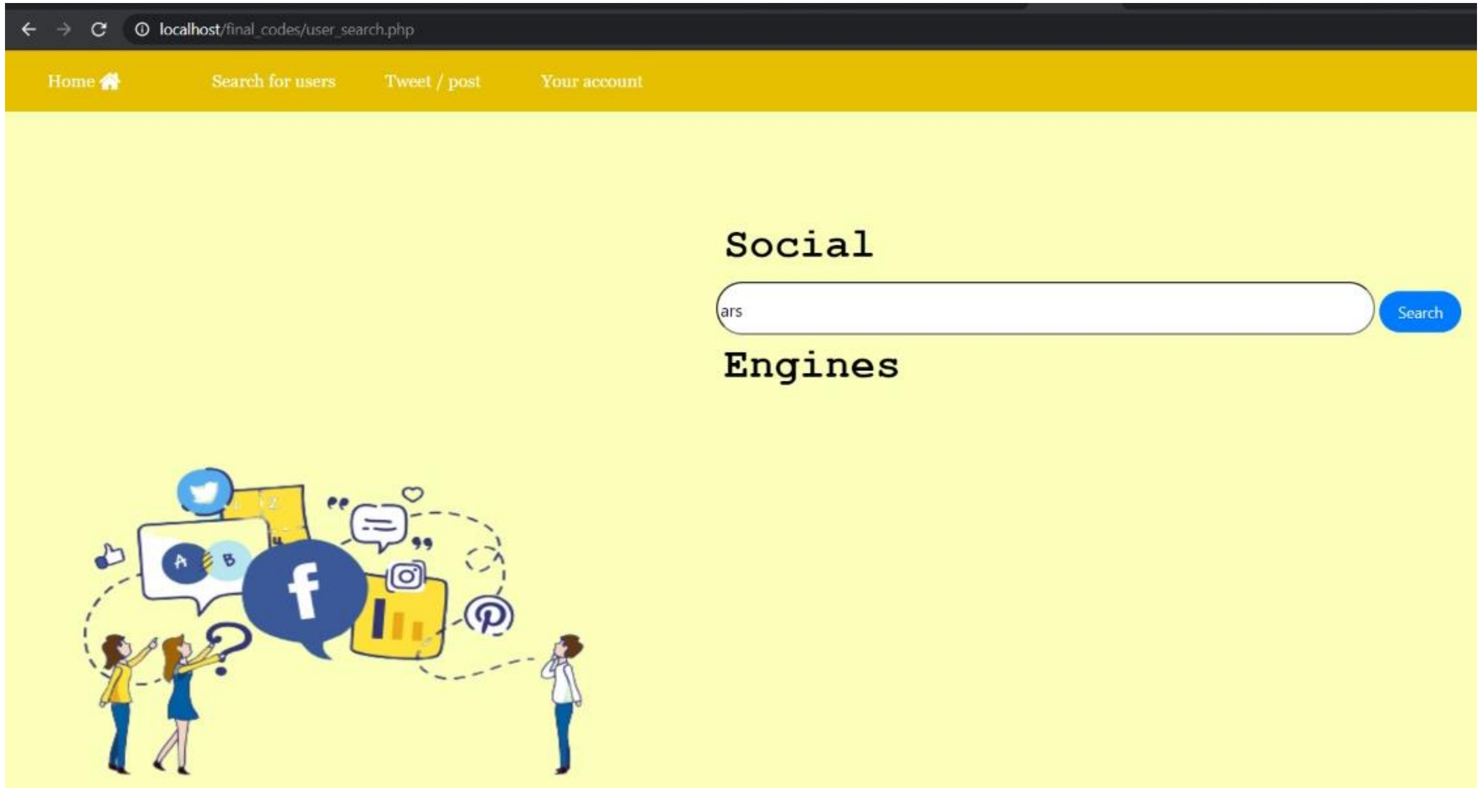
we connect and share.

varsha@gmail.com

In home-based elderly care service, how to precisely recognize activities is a key issue in the design and implementation of context-aware service for elderly people. Existing research works reveal that those approaches ignore the characteristics of activity diversity, and similarity and the features of activities of elderly people at home, so recognition accuracy of those approaches are not high enough for real-life applications. Thus, in this paper, we first study the types of activities in home-based elderly care service. Then, we propose a two-stage elderly home activity recognition method based on random forest and activity similarity. The method uses improved random forest to obtain a preliminary result in the first stage. Then, the correlation between activity, location, and time is employed to judge the rationality of the result. The similarity of activities is further used to correct the results in the second stage. We set up a series of experiments to evaluate the effectivene

varsha@gmail.com

In home-based elderly care service, how to precisely recognize activities is a key issue in the design and implementation of context-aware service

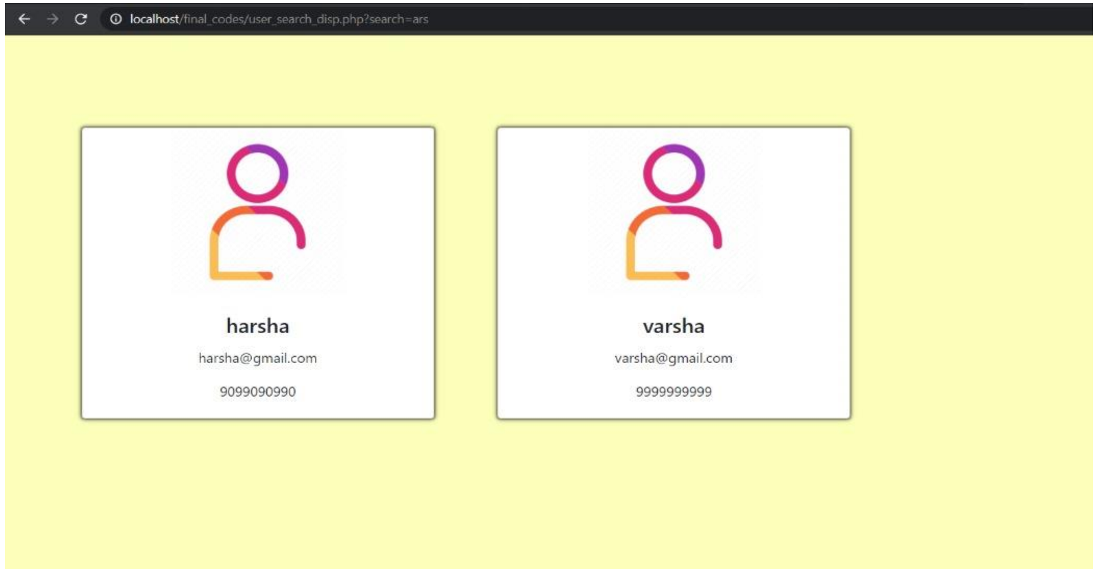


localhost/final_codes/user_search.php

Social

ars

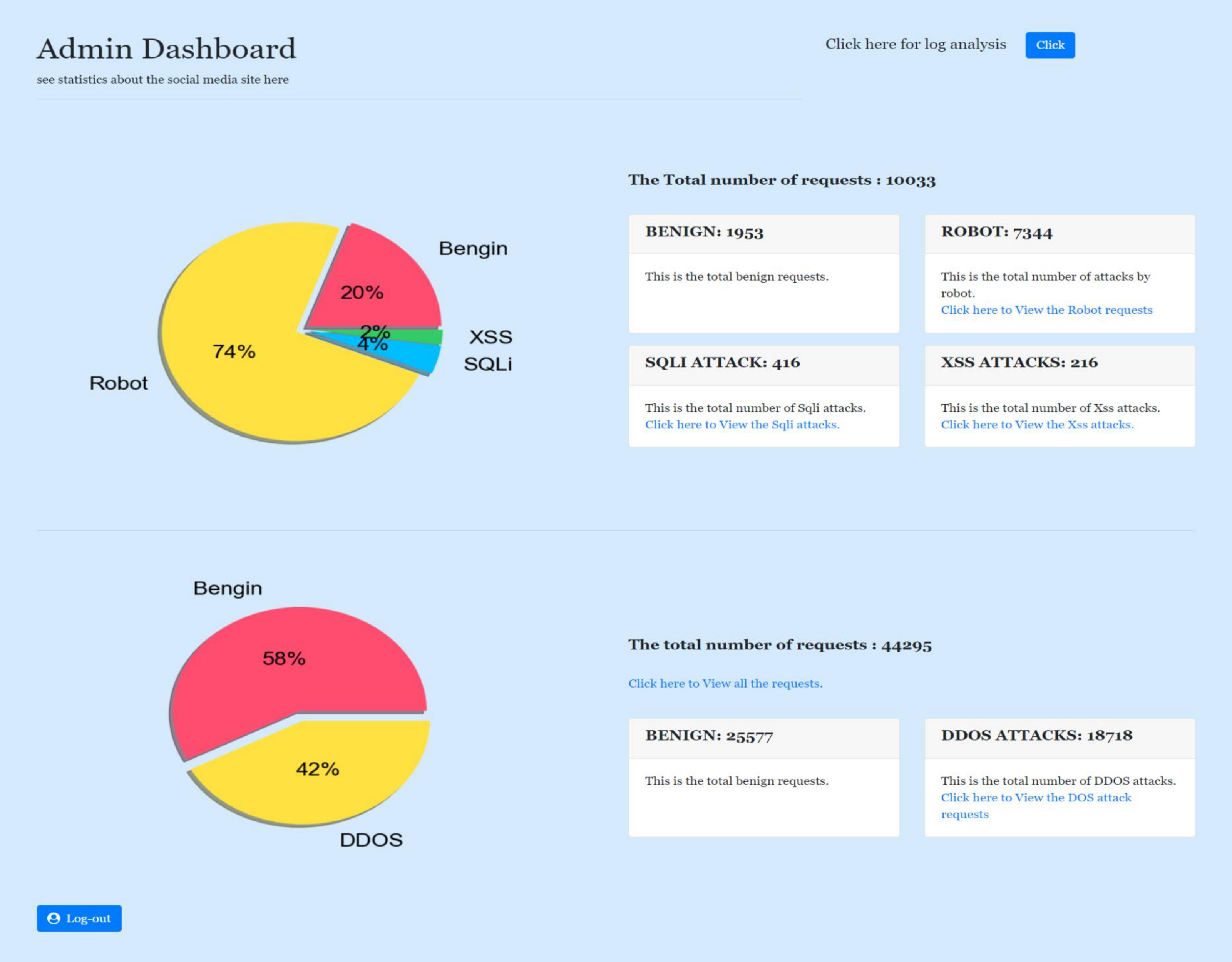
Engines



localhost/final_codes/user_search_disp.php?search=ars

harsha
harsha@gmail.com
9099090990

varsha
varsha@gmail.com
9999999999



Admin Dashboard

see statistics about the social media site here

Click here for log analysis Click

The Total number of requests : 10033

BENIGN: 1953

This is the total benign requests.

ROBOT: 7344

This is the total number of attacks by robot.
Click here to View the Robot requests

SQLI ATTACK: 416

This is the total number of Sqli attacks.
Click here to View the Sqli attacks.

XSS ATTACKS: 216

This is the total number of Xss attacks.
Click here to View the Xss attacks.

The total number of requests : 44295

Click here to View all the requests.

BENIGN: 25577

This is the total benign requests.

DDOS ATTACKS: 18718

This is the total number of DDOS attacks.
Click here to View the DOS attack requests

Log-out

Log Analysis Results

Conclusion

- Website on social media theme is developed in php script to analyze and demonstrate the SQLi and XSS attack in a real time environment.
- The Multilayer neural network that is developed to detect SQLi yields an accuracy of 97.26%.
- The Convoluted neural network that is developed to detect XSS yields an accuracy of 98.83%.
- The Multilayer perceptron is developed to detect XSS with accuracy of 98.77%.
- Log analysis module classifies DOS, XSS, Sqli, Robot attacks accurately.
- A survey paper on detection of SQLi and XSS attack is published in Journal of Emerging Technologies and Innovative Research.
- A paper on our implementation will be published soon.

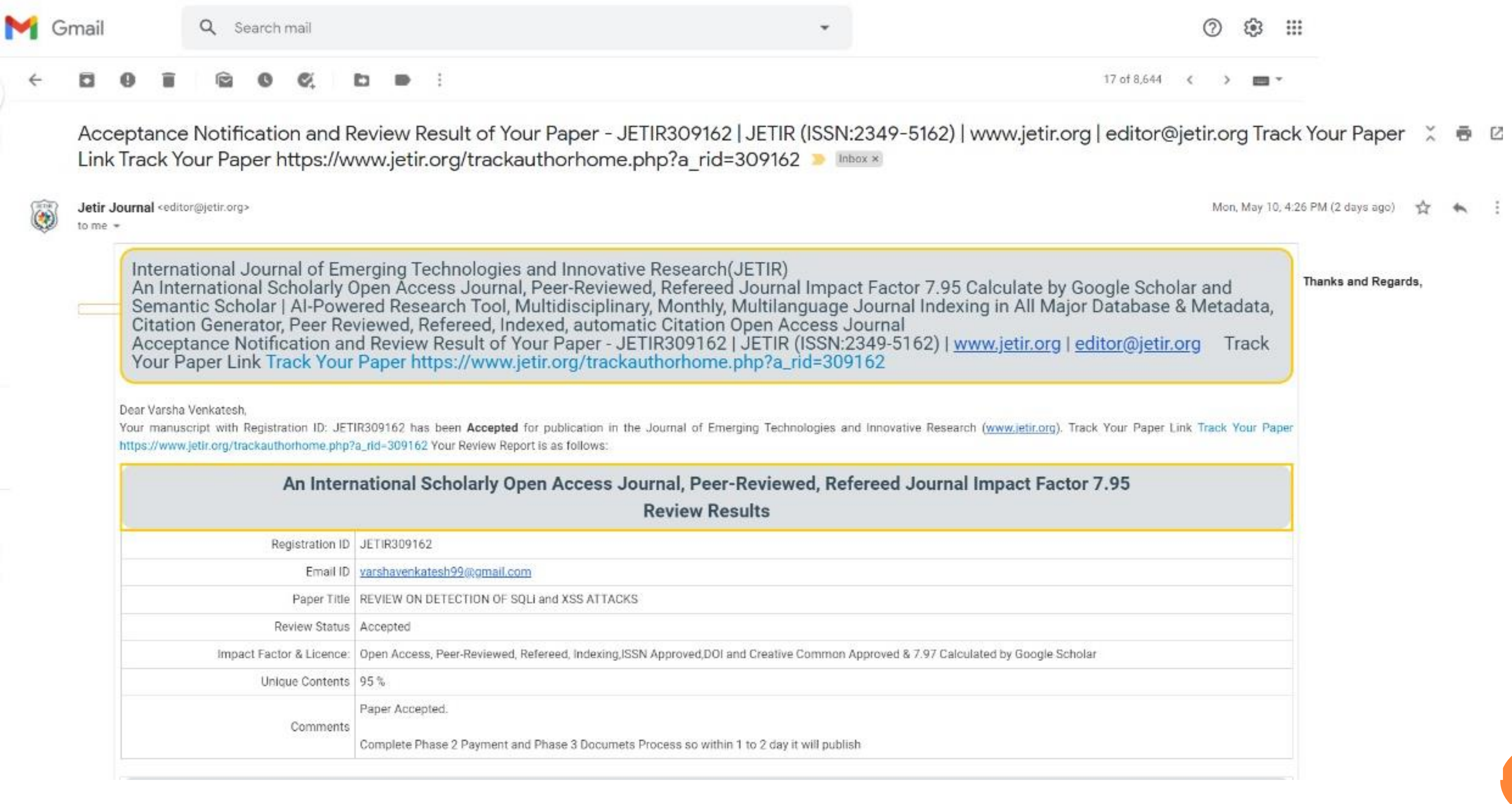
References

- [1] Sonali Mishra, "SQL Injection Detection Using Machine Learning", San José State University, 2019,USA.
- [2] Musaab Hasan, Zayed Balbahaith, and Mohammed Tarique, "Detection of SQL Injection Attacks: A Machine Learning Approach", International Conference on Electrical and Computing Technologies and Applications (ICECTA) , 2019.
- [3] Harsh Bhagwani,"Log based Dynamic Intrusion Detection of Web Applications", Indian Institute of Technology, Kanpur, 2019.
- [4] Xin Xie, Chunhui Ren, Yusheng Fu , Jie Xu , and Jinhong Guo "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN", 2019 IEEE.
- [5] Rajashree A. Katole, Dr. Swati S. Sherekar, Dr. Vilas M. Thakare, "Detection of SQL Injection Attacks by Removing the Parameter Values of SQL Query", Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018) and IEEE , 2018.

- [6] Zhuang Chen, Min Guo, Lin zhou, “Research on SQL injection detection technology based on SVM”, MATEC Web of Conferences, 2018.
- [7] Fawaz A. Mereani and Jacob M. Howe, “Detecting Cross-Site Scripting Attacks using Machine Learning”, City, University of London Institutional Repository, 2018.
- [8] Melody Moh*, Santhosh Pininti, Sindhusha Doddapaneni, and Teng-Sheng Moh, “Detecting Web Attacks Using Multi-Stage Log Analysis Year”, San Jose State University, 2017, USA.
- [9] Kunal Gupta, Rajni Ranjan Singh, Manish Dixit, “CROSS SITE SCRIPTING (XSS) attack detection using intrusion detection system”, International Conference on Intelligent Computing and Control Systems, 2017.
- [10] Shailendra Rathore, Pradip Kumar Sharma, Jong Hyuk Park, “XSSClassifier: An Efficient XSS Attack Detection Approach Based on Machine Learning Classifier on SNSs”, Journal of information processing systems, 2017.

PROOF OF ACCEPTANCE

- Journal Name: Journal of Emerging Technologies and Innovative Research
- Year of Consideration: 2021
- Impact Factor: 7.95
- Publication Frequency: -
- Paper Title: REVIEW ON DETECTION OF SQLi AND XSS ATTACKS
- Published Date with DOI number: 2021/05/12



Acceptance Notification and Review Result of Your Paper - JETIR309162 | JETIR (ISSN:2349-5162) | www.jetir.org | editor@jetir.org Track Your Paper Link Track Your Paper https://www.jetir.org/trackauthorhome.php?a_rid=309162

Jetir Journal <editor@jetir.org>
to me

Mon, May 10, 4:26 PM (2 days ago)

International Journal of Emerging Technologies and Innovative Research(JETIR)
An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor 7.95 Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator, Peer Reviewed, Refereed, Indexed, automatic Citation Open Access Journal
Acceptance Notification and Review Result of Your Paper - JETIR309162 | JETIR (ISSN:2349-5162) | www.jetir.org | editor@jetir.org Track Your Paper Link Track Your Paper https://www.jetir.org/trackauthorhome.php?a_rid=309162

Thanks and Regards,

Dear Varsha Venkatesh,
Your manuscript with Registration ID: JETIR309162 has been **Accepted** for publication in the Journal of Emerging Technologies and Innovative Research (www.jetir.org). Track Your Paper Link Track Your Paper https://www.jetir.org/trackauthorhome.php?a_rid=309162 Your Review Report is as follows:

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor 7.95
Review Results

Registration ID	JETIR309162
Email ID	varshavenkatesh99@gmail.com
Paper Title	REVIEW ON DETECTION OF SQLi and XSS ATTACKS
Review Status	Accepted
Impact Factor & Licence:	Open Access, Peer-Reviewed, Refereed, Indexing,ISSN Approved,DOI and Creative Common Approved & 7.97 Calculated by Google Scholar
Unique Contents	95 %
Comments	Paper Accepted. Complete Phase 2 Payment and Phase 3 Documets Process so within 1 to 2 day it will publish

ANY QUESTIONS

THANK YOU