INTERNSHIP REPORT

SCIENTIFIC ANALYSIS GROUP

DEFENCE RESEARCH AND DEVELOPMENT ORGANISATION

Finding Vulnerabilities Using AI

Under the guidance of Shri Amrendra Kumar

By: Manjusha Iyer Date: 29th July 2024

Acknowledgement

I am deeply grateful to many people for their support and guidance during my summer internship at the DRDO's SAG Lab.

First and foremost, I would like to express my sincere thanks to my supervisor, Amrendra Kumar sir, for his invaluable mentorship, insightful feedback, and constant encouragement throughout this project. His expertise and dedication were instrumental in shaping my understanding and approach to the work.

I am also immensely thankful to Aditya sir for his guidance and assistance with the web application. His technical knowledge and patient explanations were crucial in overcoming numerous challenges and ensuring the successful completion of this project.

Additionally, I would like to extend my heartfelt gratitude to my parents for their unwavering support and belief in my abilities. Their encouragement provided me with the motivation to persevere and excel in my endeavors.

Lastly, I am grateful to my friends, whose companionship and support made this journey enjoyable and memorable. Their well-wishes and encouragement were a source of strength throughout the internship.

This experience has been incredibly enriching, and I owe its success to the support and guidance of all the aforementioned individuals.

Thank you all.

Manjusha Iyer

Index

About DRDO

DRDO is the R&D wing of Ministry of Defence, Govt of India, with a vision to empower India with cutting-edge defence technologies and a mission to achieve self-reliance in critical defence technologies and systems, while equipping our armed forces with state-of-the-art weapon systems and equipment in accordance with requirements laid down by the three Services. DRDO's pursuit of self-reliance and successful indigenous development and production of strategic systems and platforms such as Agni and Prithvi series of missiles; light combat aircraft, Tejas; multi-barrel rocket launcher, Pinaka; air defence system, etc. have given quantum jump to India's military might, generating effective deterrence and providing crucial leverage.

"Balasya Mulam Vigyanam"—the source of strength is science-drives the nation in peace and war. DRDO has firm determination to make the nation strong and self-reliant in terms of science and technology, especially in the field of military technologies.

DRDO was formed in 1958 from the amalgamation of the then already functioning Technical Development Establishment (TDEs) of the Indian Army and the Directorate of Technical Development & Production (DTDP) with the Defence Science Organisation (DSO). DRDO was then a small organisation with 10 establishments or laboratories. Over the years, it has grown multi-directionally in terms of the variety of subject disciplines, number of laboratories, achievements and stature.

Today, DRDO is a network of around 41 laboratories and 05 DRDO Young Scientist Laboratories (DYSLs) which are deeply engaged in developing defence technologies covering various disciplines, like aeronautics, armaments, electronics, combat vehicles, engineering systems, instrumentation, missiles, advanced computing and simulation, special materials, naval systems, life sciences, training, information systems and agriculture. Several major projects for the development of missiles, armaments, light combat aircrafts, radars, electronic warfare systems etc are on hand and significant achievements have already been made in several such technologies.

Vision

Empowering the nation with state-of-the-art indigenous Defence technologies and systems.

Mission

Design, develop and lead to production state-of-the-art sensors, weapon systems, platforms and allied equipment for our Defence Services.

Provide technological solutions to the Services to optimise combat effectiveness and to promote well-being of the troops.

Develop infrastructure and committed quality manpower and build strong indigenous technology base.

About SAG

The Scientific Analysis Group (SAG) is a premier laboratory under the Defence Research and Development Organisation (DRDO), Ministry of Defence, Government of India. SAG is dedicated to advancing research and development in the field of cryptology and information security, playing a crucial role in ensuring the security of military communications and information systems. It focuses mainly on fields like Cryptology, Information Security, Cyber Security and Secure Communication Systems. SAG specializes in the study and development of cryptographic algorithms and protocols. This includes work on both encryption and decryption techniques, ensuring that sensitive military information remains secure from unauthorized access. The lab works on advanced cryptographic methods, including symmetric and asymmetric encryption, hashing algorithms, and digital signatures. In the realm of information security, SAG develops robust systems and frameworks to protect data integrity, confidentiality, and availability. This involves creating secure communication channels, developing intrusion detection systems, and implementing secure protocols for data transmission and storage. SAG also focuses on cyber security measures to protect critical defense infrastructure from cyber threats. This includes developing advanced techniques for threat detection, analysis, and mitigation. The lab works on creating resilient systems that can withstand sophisticated cyber-attacks, ensuring the integrity of India's defense networks. Developing secure communication systems is another vital area of SAG's work. This involves creating systems that enable secure voice, data, and video communications, ensuring that information transmitted over military networks is protected against interception and eavesdropping. SAG boasts advanced research facilities equipped with the latest technology for cryptographic and security research. The lab has a team of highly skilled scientists and engineers who are experts in various fields of cryptology, information security, and cyber security. The lab also collaborates with other DRDO labs, academic institutions, and international research organizations to stay at the forefront of technological advancements.

Vision

To make Scientific Analysis Group the finest Cryptology and Information Security Laboratory in the World.

Mission

To develop tools and techniques based on contemporary Mathematics, Computer Science and Electronics & Communication for Analysis of Security and IT products.

To build generalized as well as domain/system specific analytical algorithms and tools.

To establish state-of-art facilities for electronic probing, communication signal & protocol analysis.

Finding Vulnerabilities Using AI

Abstract: *This project aims to enhance cybersecurity by leveraging artificial intelligence to detect vulnerabilities in web applications. To train ML models, two datasets containing code snippets in different languages with labels for specific vulnerabilities were sourced and merged. Then, different models were trained and tested to understand which of these models performed best on the given datasets and their accuracy was measured. Finally, a web app was created and the trained ML models were made to predict the number and type of vulnerable line of codes. The project's findings demonstrate the efficacy of AI in vulnerability detection, providing a robust approach to securing web applications.*

1 PROJECT UNDERTAKEN:

The primary objective of this project was to develop a robust system capable of identifying vulnerabilities in web applications using artificial intelligence. The project was divided into several phases:

1.1 PHASE 1: TRAINING THE ML MODELS

Training AI to correctly recognize vulnerabilities in code had two main phases:

1. Dataset Sourcing to Train ML Model
2. Model Training

1.2 PHASE 2: TESTING THE ML MODELS

Testing the trained models to check their accuracy and thoroughly analyze their performance

1.3 PHASE 3: APPLYING TRAINED ML MODELS TO FIND VULNERABILITIES

Creating a web application called Offline Forum and using the trained models to find vulnerabilities.

2 INTRODUCTION:

The advent of digital technologies has revolutionized the way we live and work, but it has also introduced significant security challenges. Cybersecurity is essential to protect digital assets and ensure the integrity, confidentiality, and availability of information.

2.1 CYBERSECURITY

Cybersecurity encompasses a wide range of practices, tools, and concepts aimed at protecting computers, networks, and data from unauthorized access, attacks, and damage. Its primary goal is to safeguard sensitive information, ensuring its confidentiality, integrity, and availability. Cybersecurity involves various domains including network security, application security, information security, and operational security.

A critical component of cybersecurity for web applications is the Open Web Application Security Project (OWASP). It is a global nonprofit organization dedicated to improving the security of software. The OWASP Top Ten is a list of vulnerabilities in web applications:

1. **Injection**: Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query. This can lead to unauthorized access to sensitive data.

2. **Broken Authentication**: Weaknesses in authentication mechanisms can allow attackers to compromise user accounts, passwords, or session tokens.

3. **Sensitive Data Exposure**: Failure to properly protect sensitive data (such as credit card details, passwords, or personal information) can lead to exposure and theft.

4. **XML External Entities (XXE):** Improper processing of XML input can allow attackers to access internal files, execute remote code, or perform denial-of-service attacks.

5. **Broken Access Control**: Inadequate restrictions on authenticated users can allow unauthorized access to functionalities or data, such as viewing other users' data or modifying URLs to access restricted pages.

6. **Security Misconfiguration**: Poorly configured security settings, default configurations, or unnecessary features left enabled can lead to vulnerabilities that attackers can exploit.

7. **Cross-Site Scripting (XSS)**: XSS flaws occur when an application includes untrusted data in a web page without proper validation or escaping. This allows attackers to execute malicious scripts in the context of the victim's browser.

8. **Insecure Deserialization**: Improper handling of serialized objects can lead to remote code execution, tampering with data, or denial-of-service attacks.

9. **Using Components with Known Vulnerabilities:** Outdated or vulnerable components, libraries, or frameworks integrated into an application can expose it to known exploits.

10. **Insufficient Logging & Monitoring**: Inadequate logging and monitoring can hinder an organization's ability to detect and respond to security incidents, allowing attackers to maintain persistence or escalate attacks undetected.

Out of these 10 vulnerabilities, the project focuses on two of the most common type of web app vulnerabilities, cross site scripting or XSS and SQL injection.

2.2 Machine Learning and Artificial Intelligence

Machine learning (ML) models have become increasingly important in cybersecurity for their ability to analyze vast amounts of data and identify patterns that may indicate potential security threats. ML models can be broadly categorized into supervised, unsupervised, and reinforcement learning models:

1. **Supervised Learning**: This type of ML involves training a model on a labeled dataset, meaning that each training example is paired with an output label. Common algorithms include logistic regression, support vector machines (SVM), and neural networks. Supervised learning is widely used in cybersecurity for tasks such as spam detection, intrusion detection, and malware classification.

2. **Unsupervised Learning**: Unlike supervised learning, unsupervised learning deals with unlabeled data. The goal is to infer the natural structure present within a set of data points. Clustering and association are common tasks, with algorithms such as k-means, hierarchical clustering, and association rules being frequently used. In cybersecurity, unsupervised learning can help in anomaly detection and identifying unusual patterns that may signify a security threat.

3. **Reinforcement Learning:** This type of ML involves training models to make sequences of decisions by rewarding desirable behaviors and punishing undesirable ones. It is used in cybersecurity for dynamic decision-making tasks such as automated threat response and adaptive defense strategies.

Artificial intelligence (AI) in cybersecurity is not limited to ML. AI encompasses various techniques, including expert systems, neural networks, and natural language processing (NLP), which can enhance the capability to predict, detect, and respond to cyber threats. AI can analyze patterns in network traffic to identify anomalies, predict potential attack vectors based on historical data, and automate responses to detected threats.

2.3 Integrating AI/ML into Cybersecurity

The integration of AI and ML in cybersecurity offers several advantages. It enables the automation of threat detection and response, reducing the need for manual intervention and allowing for quicker mitigation of potential threats. AI systems can process and analyze data at scales and speeds far beyond human capabilities, making them highly effective in identifying complex and subtle attack patterns. Additionally, AI can continuously learn and adapt to new threats, improving its effectiveness over time.

3 METHOD:

The methodology of this project is structured into four main phases: data collection, data preprocessing, model selection and training, and application of the AI model to detect vulnerabilities. Each phase is detailed below:

3.1 PHASE 1: DATA COLLECTION AND PREPROCESSING

This phase consisted of two steps:

3.1.1 Data Collection

The first step involved gathering datasets that contained code snippets from HTML, JavaScript, and MySQL. These datasets included labels indicating whether each snippet was vulnerable and, if so, the type of vulnerability. Two different datasets were identified:

1- Cross-Site Scripting (XSS) Dataset for Deep Learning
(https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning)

The Cross-Site Scripting (XSS) Dataset for Deep Learning, available on Kaggle, is a valuable resource designed to facilitate the training and evaluation of machine learning models for detecting XSS vulnerabilities. Created by Syed Saqlain Hussain, this dataset contains numerous code snippets labeled to indicate the presence or absence of XSS vulnerabilities. The dataset is particularly useful for researchers and practitioners working on cybersecurity applications that involve machine learning and deep learning techniques.

The XSS dataset comprises thousands of code snippets written in various programming languages, predominantly HTML and JavaScript. Each code snippet is annotated with a label indicating whether it is vulnerable to XSS attacks.

The primary goal of this dataset is to enable the development of models that can accurately identify XSS vulnerabilities in code. The dataset is structured in a way that makes it straightforward to use for machine learning tasks. It typically includes the following columns:

- Code Snippet: This column contains the actual piece of code that is to be analyzed for XSS vulnerabilities. The code snippets vary in length and complexity, encompassing a wide range of potential XSS attack vectors.

- Label: This column indicates whether the corresponding code snippet is vulnerable to XSS attacks. The labels are binary, with '1' representing a vulnerable snippet and '0' representing a non-vulnerable snippet.

2- SQL Injection (SQLi) and Cross-Site Scripting (XSS) Dataset
(https://www.kaggle.com/datasets/alextrinity/sqli-xss-dataset)

The SQL Injection (SQLi) and Cross-Site Scripting (XSS) Dataset, available on Kaggle and created by AlexTrinity, is a comprehensive dataset designed to aid in the detection and prevention of two of the most prevalent web application security vulnerabilities: SQL Injection and Cross-Site Scripting.

This dataset provides a valuable resource for training and evaluating machine learning models aimed at identifying these vulnerabilities in web applications. The SQLi and XSS dataset comprises a wide array of code snippets labeled to indicate whether they are vulnerable to SQL Injection or XSS attacks. This labeling is crucial for supervised learning tasks in machine learning, where models are trained to distinguish between vulnerable and non-vulnerable code.

The dataset is structured with the following key components:

- Code Snippet: Contains the actual piece of code that needs to be analyzed for vulnerabilities. These snippets are written in various web programming languages, primarily focusing on those prone to SQLi and XSS vulnerabilities.

- Vulnerability Type: This column indicates the type of vulnerability present in the code snippet. The possible values include 'SQLi' for SQL Injection vulnerabilities and 'XSS' for Cross-Site Scripting vulnerabilities.

- Label: A binary or multi-class label indicating the presence (and sometimes the specific type) of a vulnerability. For instance, '0' might represent no vulnerability, '1' might indicate a SQLi vulnerability, and '2' might indicate an XSS vulnerability.

3.1.2 Data Preprocessing

Preprocessing the datasets was a crucial step to ensure that the data was clean and consistent for model training. The following steps were undertaken:

1. ***Merging Datasets:***

The two sourced datasets were combined into a single dataset. This involved aligning the structure of both datasets and ensuring that the combined dataset had consistent labeling.

2. ***Label Standardization:***

The original datasets had different labeling systems for vulnerabilities. A unified labeling system was established, with:

- 0 representing no vulnerability,
- 1 indicating SQL injection,
- 2 denoting Cross-Site Scripting (XSS).

3. ***Handling Missing Values:***

Any missing or incomplete data entries were addressed. Incomplete records were either filled with appropriate values or removed from the dataset to maintain data quality.

4. ***Encoding:***

Non-numeric data were encoded into numeric formats suitable for model training. This included converting categorical variables into numeric codes.

3.2 Phase 2: Training and Testing ML models

This phase consisted of the following steps:

3.2.1 Choosing ML Models

Logistic Regression is a linear model ideal for binary and multiclass classification. It is simple, interpretable, and computationally efficient, making it a reliable baseline model. However, it assumes a linear relationship between features and the target, which might not capture complex patterns. Scaling data and increasing iterations can address convergence issues.

Random Forest builds multiple decision trees and averages them to enhance accuracy and stability. It is robust against overfitting and can capture complex, non-linear patterns. However, it is less interpretable than linear models and can be slow to train with many trees.

XGBoost (Extreme Gradient Boosting) is a powerful ensemble method known for its high performance and ability to handle missing data. It consistently achieves top accuracy in competitions and provides feature importance insights. Despite its strength, XGBoost is computationally intensive and complex to tune.

3.2.2 Training ML Models

The models were trained on the same labeled dataset made, and the total training time was around 5 minutes for the Logistic Regression model, almost 15 minutes for XGBoost model and over 25 minutes for Random Forest Classifier, which is expected since the dataset contains 4990 total records.

3.2.3 Testing ML Models

The accuracy and other factors such as confusion matrix was created and recorded for each model and is discussed further in this report.

3.3 Phase 3: Web Application

To test the models' ability to correctly predict vulnerabilities in real world applications, a web application called Offline Forum was created.

3.3.1 Introduction

Offline Forum is designed for users in an offline network environment. It mimics the functionality of online forums like Quora but operates locally, without requiring an internet connection. Users can log in, post content, and view others' posts within the local network. The backend is built with Node.js and MySQL for data storage, while the frontend uses HTML, CSS, and JavaScript. It's structured to manage user authentication, post creation, and retrieval within the constraints of an offline environment.

3.3.2 Code

1. ***Architecture***

The application follows a client-server architecture:

- Client Side: Uses HTML, CSS, and JavaScript for frontend interactions.

- Server Side: Implemented in Node.js with Express.js framework for backend operations.

2. ***Components***

The web app consists two main components:

3.3.2.1 Frontend Components:

- `index.html`: Main landing page.

- `login.html`, `register.html`: User authentication pages.

- `dashboard.html`: User interface for managing posts and activities.

- `styles.css`: Stylesheet for consistent UI design.

3.3.2.2 Backend Components:

- `server.js`: Main backend server file handling HTTP requests and responses.

- `db.js`: Configuration file for MySQL database connection.

- Authentication using bcrypt for secure password hashing.

3.3.3 Features

3.3.3.1 User Authentication

- Registration: Allows new users to create accounts securely.

- Login: Authenticates users based on stored credentials using bcrypt hashing.

3.3.3.2 Post Management

- Posting: Users can create and upload content/posts.

- Viewing Posts: Displays recent activities and manages posts.

- Data Storage: Uses MySQL for storing user information and posts.

3.3.4 Technologies Used

3.3.4.1 Frontend

- HTML5, CSS3, and JavaScript: For interactive and responsive frontend design.

- AJAX: Enables asynchronous data retrieval and updates.

3.3.4.2 Backend

- Node.js: Runtime environment for server-side JavaScript execution.
- Express.js: Web application framework for Node.js.
- MySQL: Relational database management system for data storage.

3.3.5 Deployment and Hosting

3.3.5.1 Local Deployment:

Hosted locally using XAMPP for development and testing.

Thus, the application was built keeping in mind that it was going to be used to test for vulnerabilities using AI.

3.4 PHASE 4: PREDICTING VULNERABILITIES

Finally, the trained models were applied on the web application to detect vulnerabilities. The code base of the entire web app was converted into one CSV file containing two columns. The Sentence column had each line from all the HTML and JS files the web app contained. The Label column was left empty and thus, the models were made to predict the label of each line.

Each line of the predicted CSV was assigned a label by the model, based on the legend:

- 0 representing no vulnerability
- 1 indicating SQL injection,
- 2 denoting Cross-Site Scripting (XSS)

Thus, the ML models were used to detect vulnerabilities in the real world web application.

4 CODE

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Import the stopwords module

from nltk.corpus import stopwords

import warnings

warnings.filterwarnings('ignore')

# Define the path to CSV file

csv_file_path = 'E:\MANJUSHA\Internships\DRDO\KaggleDatas\Submission\TrainingDataset.csv'

# Read the CSV file using pandas

data = pd.read_csv(csv_file_path, encoding='latin1')

# Extract sentences and labels

sentences = data['Sentence'].values

labels = data['Label'].values

# Split the data into training and testing sets

sentences_train, sentences_test, labels_train, labels_test = train_test_split(sentences,
labels, test_size=0.2, random_state=42)

vectorizer = CountVectorizer(min_df = 2, max_df = 0.8, stop_words =
stopwords.words('english'))

sentences = vectorizer.fit_transform(sentences.astype('U')).toarray()

# Split the data into training and testing sets

sentences_train, sentences_test, labels_train, labels_test = train_test_split(sentences,
labels, test_size=0.2, random_state=42)

print(sentences_train.shape)
```

```
#LOGISTIC REGRESSION

lr_clf = LogisticRegression()

lr_clf.fit(sentences_train, labels_train)

labels_pred_lr = lr_clf.predict(sentences_test)

# Evaluate the model

accuracy_lr = accuracy_score(labels_test, labels_pred_lr)

report_lr = classification_report(labels_test, labels_pred_lr)

confusion_matrix_lr = confusion_matrix(labels_test, labels_pred_lr)

print(f"Logistic Regression Accuracy: {accuracy_lr:.2f}")

print("Classification Report:")

print(report_lr)

print("Logistic Regression Confusion Matrix:")

print(confusion_matrix_lr)

# RANDOM FORREST CLASSIFCATION

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)

rf_model.fit(sentences_train, labels_train)

labels_pred_rf = rf_model.predict(sentences_test)

# Evaluate the model

accuracy_rf = accuracy_score(labels_test, labels_pred_rf)

report_rf = classification_report(labels_test, labels_pred_rf)

confusion_matrix_rf = confusion_matrix(labels_test, labels_pred_rf)

print(f"Random Forest Accuracy: {accuracy_rf:.2f}")

print("Random Forest Classification Report:")

print(report_rf)

print("Random Forest Confusion Matrix:")

print(confusion_matrix_rf)

# ADA BOOST MODEL

ada_model = AdaBoostClassifier(n_estimators=100)

ada_model.fit(sentences_train, labels_train)

labels_pred_ada = ada_model.predict(sentences_test)

# Evaluate the model

accuracy_ada = accuracy_score(labels_test, labels_pred_ada)
```

```
report_ada = classification_report(labels_test, labels_pred_ada)
confusion_matrix_ada = confusion_matrix(labels_test, labels_pred_ada)
print(f"ADA Boost Accuracy: {accuracy_ada:.2f}")
print("Classification Report:")
print(report_ada)
print("ADA Boost Confusion Matrix:")
print(confusion_matrix_ada)
# STEP PREDICTION OF LABELS FOR WEB APPLICATION
# Read the new CSV file
new_csv_file_path = E:\MANJUSHA\Internships\DRDO\Kaggle\Submission\Offline_forum_test.csv'
new_data_lr = pd.read_csv(new_csv_file_path)
new_data_rf = pd.read_csv(new_csv_file_path)
new_data_ada = pd.read_csv(new_csv_file_path)
# Predict the labels
new_sentences = new_data_lr['Sentence'].values
# Vectorize the new sentences using the already fitted vectorizer
sentences_new = vectorizer.transform(new_sentences)
# Predict the labels using the trained model
predicted_labels_lr = lr_clf.predict(sentences_new)
predicted_labels_rf = rf_model.predict(sentences_new)
predicted_labels_ada = ada_model.predict(sentences_new)
# Add the predicted labels to the new data DataFrame
new_data_lr['Label'] = predicted_labels_lr
new_data_rf['Label'] = predicted_labels_rf
new_data_ada['Label'] = predicted_labels_ada
# Save the predictions to new files, model wise
o_csv_file_lr = 'E:\MANJUSHA\Internships\DRDO\KaggleDatas\Submission\logisticR_Pred.csv'
new_data_lr.to_csv(o_csv_file_lr, index=False)
o_csv_file_rf = 'E:\MANJUSHA\Internships\DRDO\KaggleDatas\Submission\RForrest_Pred.csv'
new_data_rf.to_csv(o_csv_file_rf, index=False)
o_csv_file_ada = 'E:\MANJUSHA\Internships\DRDO\KaggleDatas\Submission\ADABoost_Pred.csv'
new_data_ada.to_csv(o_csv_file_ada, index=False)
```

5 Final Prediction Results

Sentence	Label
const bcrypt = require('bcryptjs');	0
const express = require('express');	0
const bodyParser = require('body-parser');	0
const cookieParser = require('cookie-parser');	0
const db = require('./db');	0
const app = express();	0
app.use(express.static('docs', {	0
extensions: ['js']	0
}));	0
const PORT = 3000;	0
app.use(bodyParser.json());	0
app.use(bodyParser.urlencoded({ extended: true }));	0
app.use(cookieParser());	0
app.use(express.static('docs'));	0
app.post('/register', (req, res) => {	0
const { username, password } = req.body;	0
const hashedPassword = bcrypt.hashSync(password, 10);	0
const sql = 'INSERT INTO users (username, password) VALUES (?, ?)';	0
db.query(sql, [username, hashedPassword], (err, result) => {	0
if (err) throw err;	0
res.redirect('/login.html');	0
});	0
});	0
app.post('/login', (req, res) => {	0
const { username, password } = req.body;	0
u1=http://cyber-unity.com&m1=%3Cscript%3Eale<br/>rt(document.cookie)%3C/script%3E&m2=%3Ch1%3ECyber-Unity%3C/h1%3E&m3=by%20s3aL&m4=%3Ch1%3ESystem%20By<br/>%20XSSED!%3C/h1%3E	2
db.query(sql, [username], (err, results) => {	0
if (err) throw err;	0
if (results.length > 0 && bcrypt.compareSync(password, results[0].password)) {	0
res.cookie('user_id', results[0].id);	0
res.redirect('/dashboard.html');	0
} else {	0
res.send('Invalid credentials');	0
}	0
});	0
});	0

app.post('/post', (req, res) => {	0
const { title, content, allowedUsers } = req.body;	0
const userId = req.cookies.user_id;	0
const allowedUsersString = Array.isArray(allowedUsers) ? allowedUsers.join(',') : allowedUsers;	0
const sql = 'INSERT INTO posts (user_id, title, content, allowed_users) VALUES (?, ?, ?, ?)';	0
db.query(sql, [userId, title, content, allowedUsersString], (err, result) => {	0
if (err) throw err;	0
res.redirect('/dashboard.html');	0
msg=%3Cscript%3Ealert(%22by%20s3aL%20||Cyber-U<br/>nity.Com%22)%3C/script%3E	2
});	0
app.get('/posts', (req, res) => {	0
const userId = req.cookies.user_id;	0
select (case when (7645 = 5921) then 7645 else 7645* (select 7645 from information_schema.character_sets) end) #	1
db.query(sql, [userId], (err, results) => {	0
if (err) {	0
console.error('Error fetching posts:', err);	0
return res.status(500).json({ error: 'Error fetching posts' });	0
}	0
res.json(results);	0
select count (*) from generate_series (1,5000000) and ("jzmg" = "jzmg	1
});	0
app.listen(PORT, () => {	0
console.log(`Server running on http://localhost:${PORT}`);	0
});	0
const mysql = require('mysql');	0
const connection = mysql.createConnection({	0
host: '127.0.0.1',	0
user: 'root',	0
password: 'SQL_Entry2003',	0
database: 'offline_forum'	0
});	0
connection.connect((err) => {	0
if (err) throw err;	0
console.log('Connected to MySQL');	0
});	0
module.exports = connection;	0
<!DOCTYPE html>	0
<html lang="en">	0
<head>	0
<meta charset="UTF-8">	0
<meta name="viewport" content="width=device-width, initial-scale=1.0">	0
<title>Register</title>	0
<link rel="stylesheet" href="styles.css">	0

</head>	0
<body>	0
<h1>Register</h1>	0
<form action="/register" method="POST">	0
<input type="text" name="username" placeholder="Username" required>	0
<input type="password" name="password" placeholder="Password" required>	0
<button type="submit">Register</button>	0
</form>	0
<a href="index.html">Home</a>	0
</body>	0
</html>	0
<!DOCTYPE html>	0
<html lang="en">	0
<head>	0
<meta charset="UTF-8">	0
<meta name="viewport" content="width=device-width, initial-scale=1.0">	0
<title>Login</title>	0
<link rel="stylesheet" href="styles.css">	0
</head>	0
<body>	0
<h1>Login</h1>	0
<form action="/login" method="POST">	0
<input type="text" name="username" placeholder="Username" required style="margin: 10px;">	0
<input type="password" name="password" placeholder="Password" required style="margin: 10px;">	0
<button type="submit">Login</button>	0
</form>	0
<a href="index.html">Home</a>	0
</body>	0
</html>	0
<!DOCTYPE html>	0
<html>	0
<head>	0
<title>Offline Forum</title>	0
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />	0
<link type="text/css" rel="stylesheet" href="styles.css" media="screen,projection" />	0
</head>	0
<body>	0
<div id="container">	0
<div id="sitetitle">	0
<h1><a href="#">Offline Forum</a></h1>	0
<h2>Discussion Platform</h2>	0
</div>	0
<div id="menu">	0

```
<a href="index.html">Home</a>                                                                    0
<a href="login.html">Login</a>                                                                   0
<a href="register.html">Register</a>                                                             0
</div>                                                                                           0
<div id="content">                                                                               0
<div id="left">                                                                                  0
<div class="entry">                                                                              0
<h2>Welcome to Offline Forum</h2>                                                                0
<p>This is an offline discussion forum platform where you can post content and choose which
users can access your posts. Please log in or register to get started.</p>                       0
</div>                                                                                           0
</div>                                                                                           0
</div>                                                                                           0
<div id="footer">                                                                                0
<span>Offline Forum</span><br />                                                                 0
&copy; SAG, DRDO 2024                                                                            0
</div>                                                                                           0
</div>                                                                                           0
</body>                                                                                          0
</html>                                                                                          0
<!DOCTYPE html>                                                                                  0
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">                                        0
<head>                                                                                           0
<title>Dashboard - Offline Forum</title>                                                         0
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />                       0
<link type="text/css" rel="stylesheet" href="styles.css" media="screen,projection" />            0
</head>                                                                                          0
<body>                                                                                           0
<div id="container">                                                                             0
<div id="sitetitle">                                                                             0
<h1>Offline Forum</h1>                                                                           0
<h2>Discussion Platform</h2>                                                                     0
</div>                                                                                           0
<div id="menu">                                                                                  0
<a href="index.html">Log Out</a>                                                                 0
<a href="dashboard.html">Refresh Dashboard</a>                                                   0
</div>                                                                                           0
<div id="content">                                                                               0
<div id="left">                                                                                  0
<div class="entry">                                                                              0
<h2>Your Dashboard</h2>                                                                          0
<p>Welcome to your dashboard. Here you can manage your posts and view recent activities</p>      0
</div>                                                                                           0
<div class="entry">                                                                              0
```

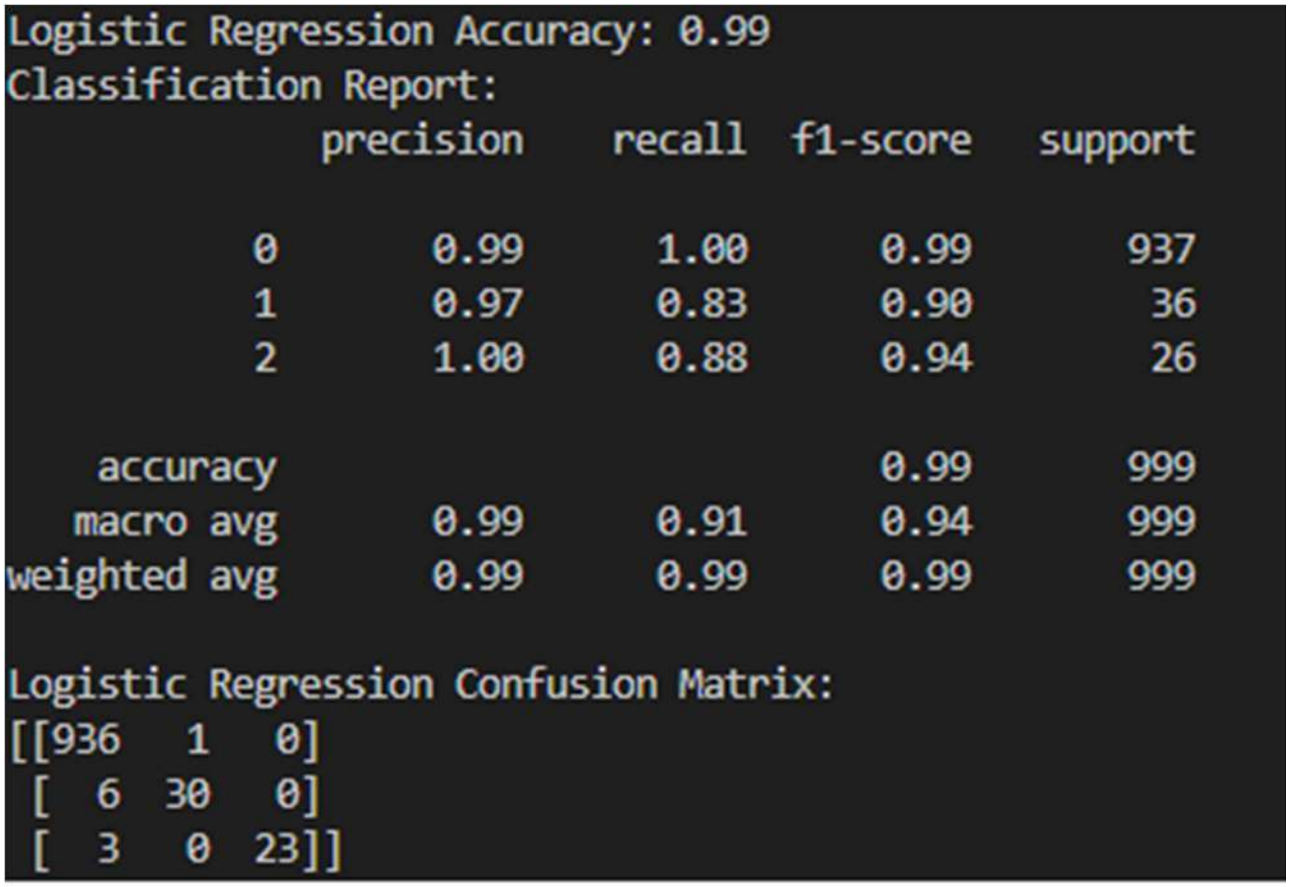
```
<h3>Publish a Post</h3>                                                                    0
<form id="publish-post-form" action="/post" method="POST">                                 0
<label for="post-title">Title:</label>                                                     0
<input type="text" id="post-title" name="title" required>                                  0
<label for="post-content" style="margin-top: 10px;">Content:</label>                       0
<textarea id="post-content" name="content" required></textarea>                            0
<label for="allowed-users" style="margin: 5px;">Allowed Users (comma-separated list):</label> 0
<input type="text" id="allowed-users" name="allowedUsers" required>                        0
<div class="button-container">                                                             0
<button type="submit">Publish</button>                                                     0
</div>                                                                                     0
</form>                                                                                    0
</div>                                                                                     0
<div class="entry">                                                                        0
<h2>Your Posts</h2>                                                                        0
<div id="user-posts" action="/posts" method="POST">                                        0
</div>                                                                                     0
</div>                                                                                     0
</div>                                                                                     0
<div id="footer">                                                                          0
<span>Offline Forum</span><br />                                                           0
&copy; 2024 SAG, DRDO | <a href="index.html">Logout</a> |                                  0
</div>                                                                                     0
</div>                                                                                     0
<script>                                                                                   0
fetch('/posts')                                                                            0
.then(response => response.json())                                                        0
.then(data => {                                                                            0
console.log('Fetched posts:', data); // Log fetched data                                   0
const userPosts = document.getElementById('user-posts');                                   0
userPosts.innerHTML = ''; // Clear previous posts                                          0
data.forEach(post => {                                                                     0
const postElement = document.createElement('div');                                         0
postElement.textContent = `${post.title}: ${post.content}`;                                0
userPosts.appendChild(postElement);                                                        0
});                                                                                        0
})                                                                                         0
.catch(error => {                                                                          0
console.error('Error fetching posts:', error);                                             0
});                                                                                        0
</script>                                                                                  0
</body>                                                                                    0
</html>                                                                                    0
```

6 Results and Analysis

The results for the three models are as follows:

6.1 Logistic Regression

The code output for the training and testing of the logistic regression model is given below:



```
Logistic Regression Accuracy: 0.99
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       937
           1       0.97      0.83      0.90        36
           2       1.00      0.88      0.94        26

    accuracy                           0.99       999
   macro avg       0.99      0.91      0.94       999
weighted avg       0.99      0.99      0.99       999

Logistic Regression Confusion Matrix:
[[936   1   0]
 [  6  30   0]
 [  3   0  23]]
```

6.1.1 Accuracy and Classification Report:

The Logistic Regression model achieved an accuracy of 0.99 (99%).

The classification report shows precision, recall, and f1-score for each class (0, 1, 2).

The support column indicates the number of instances for each class in the test set.

The model performs very well across all classes with high precision, recall, and f1-scores.

6.1.2 Confusion Matrix:

The confusion matrix shows the number of true positives, false positives, false negatives, and true negatives for each class.

True Positives (TP): No of records correctly predicted as original class 0

False Positives (FP): No. of records predicted as class 0 but actually class 1 or class 2

False Negatives (FN): No. of records predicted as class 1 or class 2 but actually class 0

True Negatives (TN): Total instances - (TP + FP + FN)

Note: The total number of instances used for prediction = 999

6.1.2.1 Class 0

- True Positives (TP): 936
- False Positives (FP): 9
- False Negatives (FN): 1
- True Negatives (TN): 53
- Proportions:
  - TP (94%), FN (5%) : Recognition Metrics
  - FP (1%), TN (0%) : Misclassification Metrics

6.1.2.2 Class 1

- True Positives (TP): 30
- False Positives (FP): 1
- False Negatives (FN): 6
- True Negatives (TN): 962
- Proportions:
  - TP (3%), FN (1%) : Recognition Metrics
  - FP (0%), TN (96%) : Misclassification Metrics

6.1.2.3 Class 2

- True Positives (TP): 23

- False Positives (FP): 0

- False Negatives (FN): 3

- True Negatives (TN): 973

- Proportions:

  - TP (2.3%), FN (0.3%): Recognition Metrics

  - FP (0%), TN (97.4%): Misclassification Metrics

6.1.2.4 Conclusion

- The model performs well across all classes with high true positive rates and low false positive and false negative rates.

- Class 0 has the highest recall and precision, indicating very few misclassifications. Also Class 0 has the highest recognition metrics of 99%.

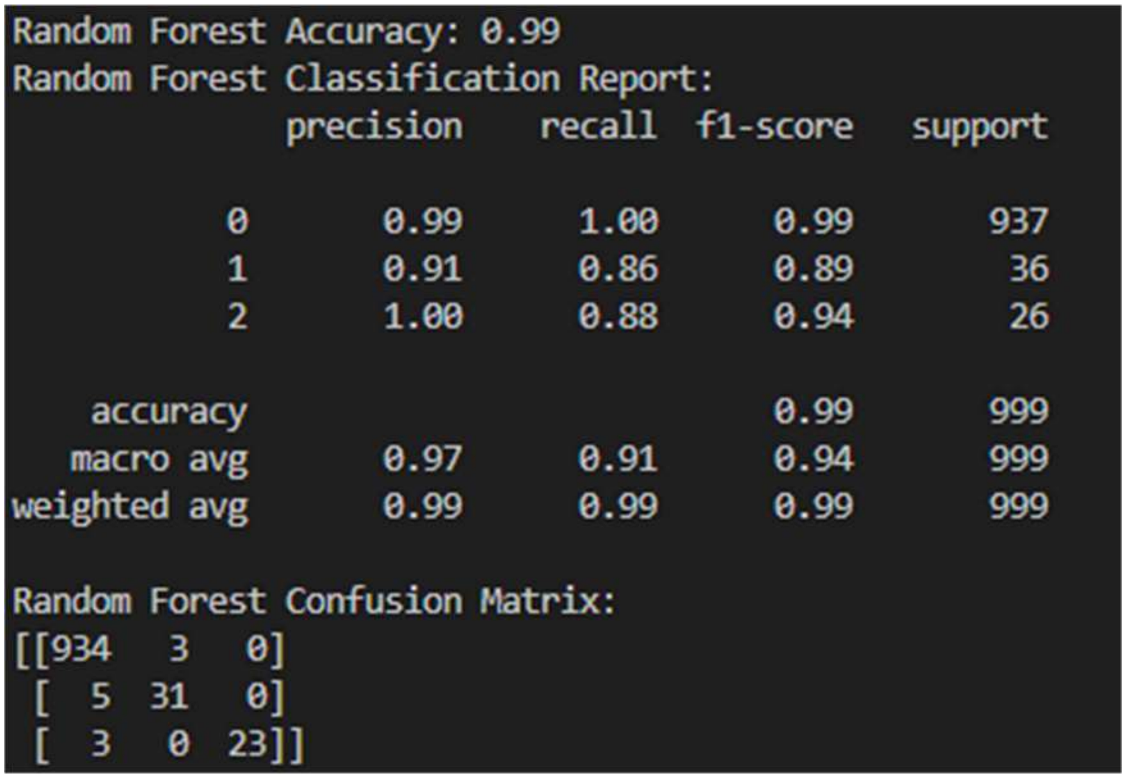
- Class 1 has slightly lower recall, indicating some instances are misclassified as other classes, but it still has high precision.

- Class 2 also shows high performance with perfect precision but slightly lower recall. Both recognition metrics and misclassification metrics scores demonstrate effective distinction between Class 2 and other classes

These metrics indicate a strong overall performance of the Logistic Regression model in this text classification task.

6.2 RANDOM FOREST

The code output for the training and testing of the Random Forest model is given below:



```
Random Forest Accuracy: 0.99
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       937
           1       0.91      0.86      0.89        36
           2       1.00      0.88      0.94        26

    accuracy                           0.99       999
   macro avg       0.97      0.91      0.94       999
weighted avg       0.99      0.99      0.99       999

Random Forest Confusion Matrix:
[[934   3   0]
 [  5  31   0]
 [  3   0  23]]
```

6.2.1 Accuracy and Classification Report:

The Random Forest model achieved an accuracy of 0.99 (99%).

The classification report shows precision, recall, and f1-score for each class (0, 1, 2).

The support column indicates the number of instances for each class in the test set.

The model performs very well across all classes with high precision, recall, and f1-scores.

6.2.2 Confusion Matrix:

The confusion matrix shows the number of true positives, false positives, false negatives, and true negatives for each class.

6.2.2.1 Class 0

- True Positives (TP): 934

- False Negatives (FN): 3

- False Positives (FP): 8

- True Negatives (TN): 54

- Proportions:

  - TP (93.49%) FN (0.3%): Recognition Metrics

  - FP (0.8%), TN (5.41%) : Misclassification Metrics

6.2.2.2 Class 1

- True Positives (TP): 31

- False Negatives (FN): 5

- False Positives (FP): 3

- True Negatives (TN): 960

- Proportions:

  - TP (3.1%), FN (0.5%): Recognition Metrics

  - FP (0.3%), TN (96.1%): Misclassification Metrics

6.2.2.3 Class 2

- True Positives (TP): 23

- False Negatives (FN): 3

- False Positives (FP): 0

- True Negatives (TN): 973

- Proportions:

  - TP (2.3%), FN (0.3%): Recognition Metrics

  - FP (0.0%), TN (97.4%): Misclassification Metrics

6.2.2.4 Conclusion

The Random Forest model shows a high level of accuracy and strong performance across all classes.

- - Class 0 has a true positive rate of 93.49% and a false positive rate of 0.8%, indicating robust classification capability for this class. Also it has high precision and recall, indicating a good performance with minimal misclassifications.

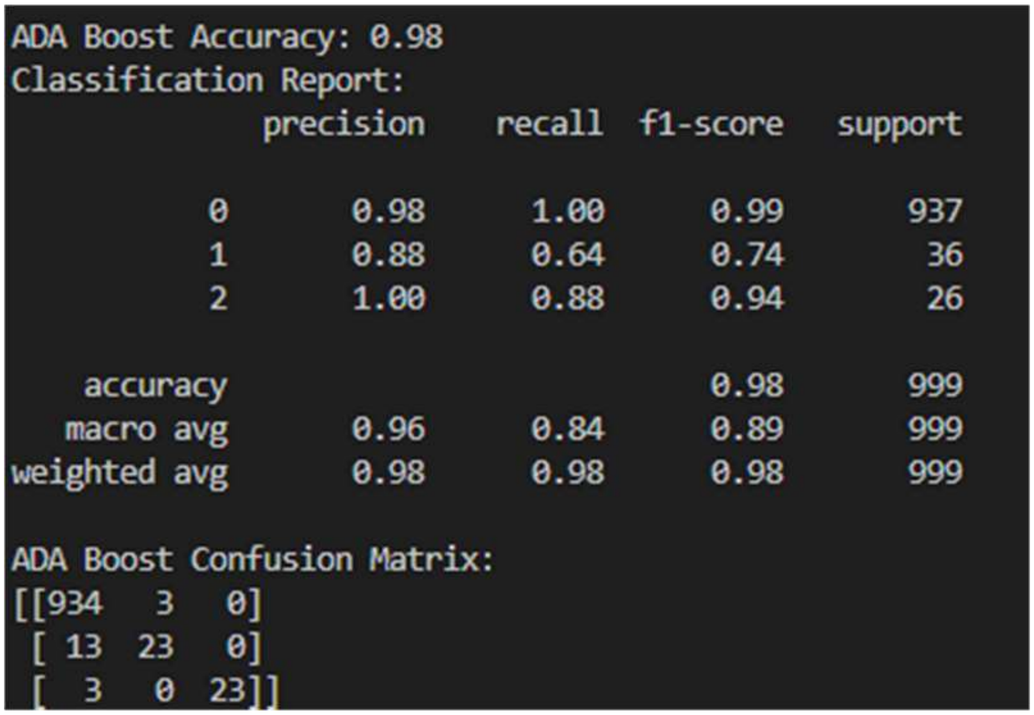
- - Class 1 has slightly lower recall compared to precision, indicating some instances are missed.

- Class 2 has high performance with perfect precision and high recall. Also it shows the highest true negative rate at 97.4%, demonstrating effective distinction between Class 2 and other classes.

These metrics affirm the Random Forrest model's reliability and effectiveness in handling multi-class classification tasks.

6.3 ADA BOOST MODEL

The code output for the training and testing of the   ADA Boost model is given below:



```
ADA Boost Accuracy: 0.98
Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       937
           1       0.88      0.64      0.74        36
           2       1.00      0.88      0.94        26

    accuracy                           0.98       999
   macro avg       0.96      0.84      0.89       999
weighted avg       0.98      0.98      0.98       999

ADA Boost Confusion Matrix:
[[934   3   0]
 [ 13  23   0]
 [  3   0  23]]
```

6.3.1 Accuracy and Classification Report:

The ADA Boost model achieved an accuracy of 0.98 (98%).

The classification report shows precision, recall, and f1-score for each class (0, 1, 2).

The support column indicates the number of instances for each class in the test set.

The model performs very well across all classes with high precision, recall, and f1-scores.

Confusion Matrix:

The confusion matrix shows the number of true positives, false positives, false negatives, and true negatives for each class.

6.3.1.1 Class 0

- True Positives (TP): 934

- False Negatives (FN): 3

- False Positives (FP): 16

- True Negatives (TN): 46

- Proportions:

  - TP (93.49%), FN (0.30%) : Recognition Metrics

  - FP (1.61%), TN (4.60%) : Misclassification Metrics

6.3.1.2 Class 1

- True Positives (TP): 23

- False Negatives (FN): 13

- False Positives (FP): 3

- True Negatives (TN): 960

- Proportions:

  - TP (2.30%), FN (1.30%) : Recognition Metrics

  - FP (0.30%), TN (96.10%) : Misclassification Metrics

6.3.1.3 Class 2

- True Positives (TP): 23

- False Negatives (FN): 3

- False Positives (FP): 0

- True Negatives (TN): 973

- Proportions:

  - TP (2.3%), FN (0.3%) : Recognition Metrics

  - FP (0.0%), TN (97.4%) : Misclassification Metrics

6.3.1.4 Conclusion

The ADA Boost model shows a high level of accuracy and strong performance across all classes.

- - Class 0 has a true positive rate of 93.49% and the lowest false positive rate of 1.61%, indicating robust classification capability for this class.

- - Class 1 has slightly lower recall compared to precision, indicating some instances are missed.

- - Class 2 shows the highest true negative rate at 97.4%, demonstrating effective distinction between Class 2 and other classes.

These metrics affirm the ADA Boost model's reliability and effectiveness in handling multi-class classification tasks.

7 Conclusion and Future Scope

7.1 Conclusion

This project underscores the transformative potential of artificial intelligence (AI) and machine learning (ML) in enhancing cybersecurity, particularly in the context of web application security. By leveraging the power of AI/ML models, we can significantly improve the detection of common vulnerabilities, such as cross-site scripting (XSS) and SQL injection, which are often exploited by attackers to compromise web applications. The integration of AI/ML into cybersecurity practices offers a proactive approach to identifying and mitigating threats, moving beyond traditional, reactive security measures.

The evaluation of these models highlighted the strengths and limitations of different algorithms in the context of cybersecurity. While accuracy was the primary metric, the consideration of precision, recall, and F1-score provided a comprehensive understanding of the models' performance. The best-performing models demonstrated a high level of accuracy in identifying vulnerabilities, showcasing the potential of AI/ML in automating the detection process.

The development of the Offline Forum web application served as a practical demonstration of the project's findings. By integrating the trained models into a user-friendly platform, the application allowed users to submit code snippets for analysis and receive detailed reports on detected vulnerabilities. This real-world application illustrated how AI/ML can be seamlessly incorporated into cybersecurity tools to enhance their effectiveness and usability.

The successful implementation of AI/ML models in this project highlights several key benefits. Firstly, it significantly reduces the time and effort required for manual vulnerability assessment, allowing security professionals to focus on more complex and strategic tasks. Secondly, the continuous learning capabilities of AI/ML models enable them to adapt to new and emerging threats, ensuring that the detection mechanisms remain up-to-date. Lastly, the scalability of AI/ML solutions makes them suitable for large-scale applications, providing robust security for extensive web infrastructures.

7.2 Future Scope

Looking forward, there are several key areas for further research and development that can enhance the effectiveness and robustness of AI/ML models in cybersecurity. These include exploring advanced machine learning techniques, addressing the causes of vulnerabilities, and integrating real-time data from manual testing software. The following points outline the future scope of this project:

7.2.1 Exploring Reinforcement Learning Models

- To further improve the accuracy and reliability of vulnerability detection, exploring the use of reinforcement learning models is a promising avenue. Reinforcement learning, which focuses on training models through reward-based mechanisms, can potentially enhance the model's ability to detect vulnerabilities with greater precision. One critical aspect to focus on is minimizing false positive rates, ensuring they remain consistently below 0.3. This threshold is essential for maintaining the usability and trustworthiness of the AI/ML models in practical applications.

7.2.2 Understanding and Addressing Vulnerability Causes

- Beyond merely detecting vulnerabilities, it is crucial to delve deeper into understanding the underlying reasons why certain code snippets are vulnerable. By analyzing the specific patterns and structures that lead to vulnerabilities, the AI/ML models can provide more insightful feedback. This feedback can include detailed explanations of why the code is deemed vulnerable and actionable recommendations on how to fix these issues. Such an approach not only helps in immediate remediation but also educates developers, enabling them to write more secure code in the future.

7.2.3 Integrating Real-Time Inputs from Manual Testing Software

- To enhance the practicality and real-time applicability of the AI/ML models, integrating inputs from manual testing software is a strategic direction. Manual testing tools often provide real-time data and insights during the testing phase, which can be invaluable for the AI/ML models. By incorporating this real-time data, the models can be continually updated and refined, ensuring they remain effective in identifying and mitigating vulnerabilities as they are discovered. This integration will create a dynamic, real-time defense mechanism, combining the strengths of automated AI/ML detection with the nuanced insights of manual testing.

By focusing on these areas, future research can significantly advance the capabilities of AI/ML models in cybersecurity, making them more accurate, insightful, and responsive to real-time threats. These efforts will contribute to building more robust and secure web applications, better equipped to withstand the evolving landscape of cyber threats.

8 REFERENCES

1. **WEB APPLICATION DEVELOPMENT:**

- Crockford, D. (2008). JavaScript: The Good Parts. O'Reilly Media.

- Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley.

- Derry, B., & Buchele, A. (2014). Node.js, MongoDB, and AngularJS Web Development. Addison-Wesley.

2. **VULNERABILITY DETECTION AND CYBERSECURITY**:

- OWASP Foundation. (2023). OWASP Top Ten. Retrieved from https://owasp.org/www-project-top-ten/

- Antunes, J., & Vieira, M. (2009). Evaluating and improving web vulnerability scanners. In 2009 Fourth International Conference on Software Testing, Verification and Validation (pp. 386-395). IEEE.

- Doupe, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 111-131). Springer, Berlin, Heidelberg.

3. **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING:**

- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

4. **MACHINE LEARNING FRAMEWORKS AND LIBRARIES:**

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) (pp. 265-283).

5. Cybersecurity Datasets:

- Syed Saqlain Hussain. (n.d.). Cross-Site Scripting (XSS) Dataset for Deep Learning. Kaggle. Retrieved from https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning/code

- Prince Roy. (n.d.). XSS Detection by Machine Learning. Kaggle. Retrieved from https://www.kaggle.com/code/princeroy15/xss-detection-by-machine-learning

- AlexTrinity. (n.d.). SQL Injection (SQLi) and Cross-Site Scripting (XSS) Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/alextrinity/sqli-xss-dataset