A PROJECT REPORT

ON

PHISHALERT: AN INTELLIGENT BROWSER EXTENSION FOR PHISHING DETECTION

SUBMITTED TO AN AUTONOMOUS INSTITUTE, AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE

SUBMITTED BY

KHATAVKAR ANIKET Roll No :48
GANDOLE ANIKET Roll No: 67
GUNAWAT PRUTHVIRAJSINH Roll No: 30

DEPARTMENT OF ARTIFICIAL INTELLI-GENCE

G H RAISONI COLLEGE OF ENGINEERING AND MANAGEMENT
WAGHOLI, PUNE 412207

2024-25

G H raisoni COLLEGE | Engineering and Management Pune

An Empowered Autonomous Institute, Affiliated to SPPU, Pune
NAAC Accredited "A+" Grade & NBA

CERTIFICATE

This is to certify that the project report entitles

PHISHALERT: AN INTELLIGENT BROWSER EXTENSION FOR PHISHING DETECTION

Submitted by

KHATAVKAR ANIKET	Roll No :48
GANDOLE ANIKET	Roll No: 67
GUNAWAT PRUTHVIRAJSINH	Roll No: 30

are a Bonafide students of this institute and the work has been carried out by them under the supervision of **Dr. R.Y. Sable** and it is approved for the partial fulfillment of the requirement of an autonomous institute, affiliated to Savitribai Phule Pune University, for the award of the degree of **Bachelor of Technology in Artificial Intelligence** in the academic year 2024-25**.**

Dr. R.Y.Sable
Guide and HOD

……
External Examiner

Dr. R. D. Khradkar
Director
GHRCEM, Pune

Date:
Place: Pune

ACKNOWLEDGMENT

It gives us great pleasure in presenting **PhishAlert: An Intelligent Browser Extension for Phishing Detection** as our B.Tech. project. Words have never seemed as inadequate as now when we are endeavoring to express our gratitude at the culmination of our B.Tech. Project to all those who have made it possible. Even the best efforts are waste, without the proper guidance and advice of our project guide **Dr. R.Y. Sable** for the consistent guidance, co-operation, inspiration, practical approach and constructive criticism, which provided us the much needed impetus to work hard & also thanks **Dr. R.Y. Sable** Head of Artificial Intelligence Department for their continuous support & valuable suggestions. We take this opportunity to thank our Campus Director **Dr. R. D. Kharadkar** for their whole hearted support, motivation & valuable suggestions. We would also like to thank **Dr. Vaishali Baviskar** our Project Coordinator for her valuable support in providing us with the required information.

We would also like to thank **Prithviraj Consultancy Services** for Sponsoring Our **project PhishAlert: An Intelligent Browser Extension for Phishing Detection.**

At the end, we would like to give special thanks to all staff members from **AI AnDepartment** of G H Raisoni College of Engineering and Management, Pune & our colleagues for their kind support & timely suggestions.

Mr. Khatavkar Aniket (A-48)
Mr. Gandole Aniket (A-67)
Mr. Gunawat Pruthvirajsinh (A-30)

ABSTRACT

Phishing attacks are a newly emerging problem in the field of computer security that involves misleading websites that require the user to enter their credentials. Often traditional detection technique fails and there arise the necessity of using enhanced techniques. This paper proposes a browser extension that utilizes the Random Forest Classifier in order to effectively filter phishing sites in real time. Justifiably, the extension performs comprehensive analytical functions of websites' URLs and contents to provide precise results of the threats. As they function within the browser environment it gives an instantaneous safeguard without having to complicate things regarding its use for the degree of technical understanding of its users. Therefore, this effort increases effectiveness of detections, reduces false positives and enhances internet browsing safety for everybody. This work contributes to the body of knowledge in cyber security as this system offers a way to block phishing exercises and protect users from fraud.

Keyword---- Phishing Detection, Browser Extension, Random Forest Classification, Cyber security, URL Analysis, Web Security, Online Threats, Phishing Attack Prevention

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	ILLUSTRATION
AI	Artificial Intelligence
CSV	Comma-Separated Values
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation

LIST OF APPENDIXS

CHAPTER 1

INTRODUCTION

1.1 Overview

With the rising interest in internet in day to day life, there has been a corresponding rise in cyber threats, with phishing being one of the most regular and dangerous. By duplicating reliable websites, phishing schemes betray users into disclosing their sensitive information, such as passwords or financial data, making them tough to detect using traditional security methods like blacklisting. As phishing strategies grow more sophisticated, more innovative methods are required to combat these threats. Unfortunately, traditional methods of detecting phishing attacks, including blacklisting known malicious sites and heuristic approaches that analyze patterns of behavior, often fall short in real-time applications. These traditional methods are not able to keep pace with the ever changing approaches employed by cybercriminals, rendering users vulnerable to various forms of online fraud and identity theft. This condition underscores a pressing need for an innovative solution that leverages cutting-edge technologies to effectively and efficiently identify phishing websites as they appear.

This research paper implements a smart browser extension that uses a Random Forest Classifier to identify phishing websites in real-time. By analyzing key features such as URL patterns and site content, the system can accurately identify harmful websites and deliver immediate protection without requiring user intervention. The extension is made to be user-friendly and efficient, confirming that it enriches cyber security while keeping false positives to a minimum. With its capacity to identify threats in real-time, the tool provides a simple yet powerful method to safeguard users. Furthermore, by adapting to evolving phishing techniques, this research contributes significantly to the ongoing fight against cybercrime, making online browsing safer and more secure for users across all experience levels.

1.2 Motivation

PhishAlert was developed to address the growing threat of phishing, which poses a significant risk in today's online-driven world. With increasing dependence on internet-based activities like banking, shopping, and communication, phishing attacks have become more sophisticated, causing financial losses and data breaches. Traditional detection methods, such as blacklists, are reactive and fail to adapt to evolving tactics used by cybercriminals. PhishAlert aims to fill this gap by leveraging the Random Forest algorithm to provide real-time detection and protection against phishing. Designed to be user-friendly and accessible,

it ensures enhanced cybersecurity without requiring technical expertise, fostering trust and safety in online interactions across various sectors.

1.3 Problem Definition and Objective:

ProblemDefinition:

Phishing attacks have evolved into sophisticated schemes that bypass traditional detection methods like blacklists and static rule sets, leaving users vulnerable to data theft, financial loss, and SQL injections. Current anti-phishing tools are inadequate in providing real-time protection against newly emerging threats, necessitating a more dynamic and intelligent solution.

Objective:

1. Develop a browser extension for real-time phishing detection using machine learning.
2. Implement the Random Forest algorithm to accurately classify websites as safe or phishing.
3. Provide instant alerts to users when phishing threats are detected.
4. Ensure the system is user-friendly, accessible to all users regardless of technical expertise.
5. Create a flexible, cross-platform solution that securely handles user data.
6. Continuously update the model with emerging phishing tactics to minimize false positives and negatives.

1.4 Scope and Limitations

Scope:

PhishAlert aims to provide robust, real-time detection of phishing websites, enhancing user security during online activities such as banking, shopping, and social networking. The system is designed to safeguard users from phishing attempts that seek to steal sensitive information like passwords and financial data. It functions as a browser extension, offering instant alerts and protection without requiring technical expertise. PhishAlert is intended for use across multiple platforms and browsers, ensuring broad accessibility and protection.

Limitations:

1. **Focus on Web-Based Phishing**: PhishAlert primarily targets website-based phishing and may not detect other forms, such as phishing through email or SMS.
2. **Data-Dependent Accuracy**: The system's effectiveness relies on the quality and diversity of training data, which may affect accuracy if the data is limited or biased.
3. **Performance Variability**: Real-time detection performance may be influenced by factors like internet speed and device capacity, potentially causing slower response times.
4. **False Positives and Negatives**: The system may occasionally misclassify legitimate sites as phishing or fail to detect actual phishing sites, impacting user trust.
5. **Ongoing Maintenance**: Continuous updates and improvements are required to keep pace with evolving phishing tactics and ensure the system remains effective.

1.5 Methodologies for Problem Solving

To effectively combat phishing in real-time web browsing, PhishAlert integrates the following methodologies:

1. **Machine Learning Techniques**: Utilizes the Random Forest algorithm to analyze URLs and HTML content, enabling accurate detection of phishing sites based on past data and adaptive learning.
2. **Feature Extraction**: Identifies critical attributes such as URL structure, metadata, and page content to distinguish between legitimate and phishing websites.
3. **Real-Time Monitoring**: Continuously monitors user activities, providing immediate alerts when potential phishing threats are detected.
4. **User Behavior Analysis**: Examines user browsing patterns to identify abnormal activities, enhancing detection accuracy by aligning with real user behaviors.

5. **Feedback Loop Mechanism**: Incorporates user feedback to refine the model, allowing continuous updates to address new phishing techniques.
6. **User-Friendly Interface**: Provides an intuitive interface with clear alerts, ensuring accessibility for users of all technical backgrounds to maximize usability and adoption

CHAPTER 2

LITERATURE SURVEY

Sr. No.	Paper Title and Year	Methodology	Dataset	Advantages	Disadvanatges	Details of Publication/ Problem Statement
1	Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation (2020)	Random Forest, SVM, Neural Network	Phishing site data (CSV)	High accuracy, multiple methods	Requires extensive training data	2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)
2	Intelligent Phishing Website Detection using Random Forest Classifier	Random Forest Classifier	Phishing site data (CSV)	Effective for large datasets	May require feature tuning	Effat University, College of Engineering, Jeddah, Saudi Arabia
3	Phishing Web Page Detection Using Optimised Machine Learning (2017)	Optimised Machine Learning	Phishing site data	Improved detection rates	Potential overfitting	2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)
4	Phishing Attacks and Protection Against Them [2021]	Social Engineering	-	1.Awareness Building	1.Complexity 2. Resource Intensive	Phishing attacks are major cybersecurity threats that steal personal data and install malware. Understanding these attacks and protective measures is essential for improving cybersecurity.

5	Detection of Phishing Websites using Machine Learning [2020]	Support Vector Machines (SVM)	Phishing Emails Dataset	1.High Accuracy 2.Real-Time Detection	1.Dependency on Blacklist 2.Potential False Positives	Phishing attacks are a significant financial and security threat. This research aims to efficiently detect phishing websites using a combination of blacklisting and semantic analysis techniques.
6	Phishing Web Page Detection Methods: URL and HTML Features Detection[2020]	Gaussian Naïve Bayes (GNB)	500 phishing web pages and 500 legitimate web pages	1.Faster Detection 2.Increased Accuracy	1.Evaluation Challenges 2.Complexity	Phishing is internet fraud using fake web pages to steal sensitive information. Current detection methods using machine learning often fail in real-world applications due to complexity and a narrow focus on detection accuracy over effectiveness and usability.
7	Malicious Web Content Detection Using Machine Leaning [2017]	Random Forest (RF) algorithm	Phishing Website Dataset	1.Real-time Protection 2.Feature Extraction	1.Limited to Chrome 2. Dependency on Training Data	The paper tackles the problem of naive web users encountering malicious content like phishing URLs and malware. It proposes a Google Chrome extension that uses machine learning to detect and categorize harmful content in real-time,

						protecting users from these threats.
8	Browser Extension based Hybrid Anti-Phishing Framework using Feature Selection[2020]	Random Forest	Phishing Websites Dataset	Low Computational Demand	1.Dependence on Pre-Defined Lists 2. Machine learning Constraints	Phishing involves creating fake web pages to steal sensitive information. Current detection methods using AI and ML often fail in real-world applications due to their complexity and focus on detection accuracy over effectiveness and usability.
9	Phishing Detection Using Machine Learning Techniques [2020]	1.K-Nearest Neighbors (KNN) 2.Artificial Neural Networks	Data have: 1.Having an IP address in the URL 2.URL length 3.Use of shortening services.	1.High Detection Accuracy 2. Real-Time Effectiveness	1.Dependence on Dataset 2. Complexity	Phishing, a persistent internet fraud, uses fake web pages resembling real ones to trick users into giving sensitive info. Current detection methods, focusing on accuracy, often fail due to complexity and usability issues.
10	Real-time phishing detection using deep learning methods by extensions [2021]	1.Support Vector Machine (SVM) 2.Convolutional Neural Network (CNN)	a mix of legitimate and phishing URLs	1.Real-Time Detection 2. Comprehensive Analysis	1.Complexity 2. Explainability	The problem is detecting phishing websites that mimic legitimate sites to steal user information. The aim is to develop an effective machine learning model to accurately identify

						phishing URLs and protect users.
11	A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods [2021]	1. Neural Network 2.Linear Model		1.Dataset Utilization 2.Feature Extraction	1.Complexity 2. False Positives	The paper tackles the challenge of detecting phishing websites that trick users into giving away sensitive information. Current detection methods using URL, HTML, CSS features, and machine learning often fail in real-world applications due to their complexity and focus on accuracy over usability and effectiveness.
12	Convolutional Neural Network with Character Embeddings for Malicious Web Request Detection [2021]	Convolutional-Neural Networks	PhishTank Dataset	1.Comprehensive Feature Set 2.Automation	1.Overfitting Risks 2.Scalability Issues	The paper addresses the issue of detecting phishing websites, which pose significant threats by misleading users into divulging sensitive information. Current detection methods often fail in practical scenarios due to their complexity and limited focus on usability and effectiveness, using features of URL, HTML, CSS, and machine learning models.

CHAPTER 3

METHODOLOGY

Here, we present the details concerning the 3.1 system architecture 3.2 available dataset and 3.3 measures of performance.

3.1 System Architecture

The system architecture of the proposed Intelligent Browser Extension for real–time phishing detection is described below. The architecture of the proposed approach can be broken down into multiple components, of which data collection, feature extraction, and a machine learning model make up the basics of the architecture. To distinguish between real and fake sites, the system takes into consideration several aspects to do with the site, including URL, HTML content and JS content. Integrated into the browser, it notifies users as soon possible when a phishing site is located and protects them automatically. Also, the architecture considers feedback loops that would maintain and enhance the detection that looks for novel forms of phishing attacks since the latter is a dynamic threat.

Figure 1 below displays the various parts and below each part is a description of that part.

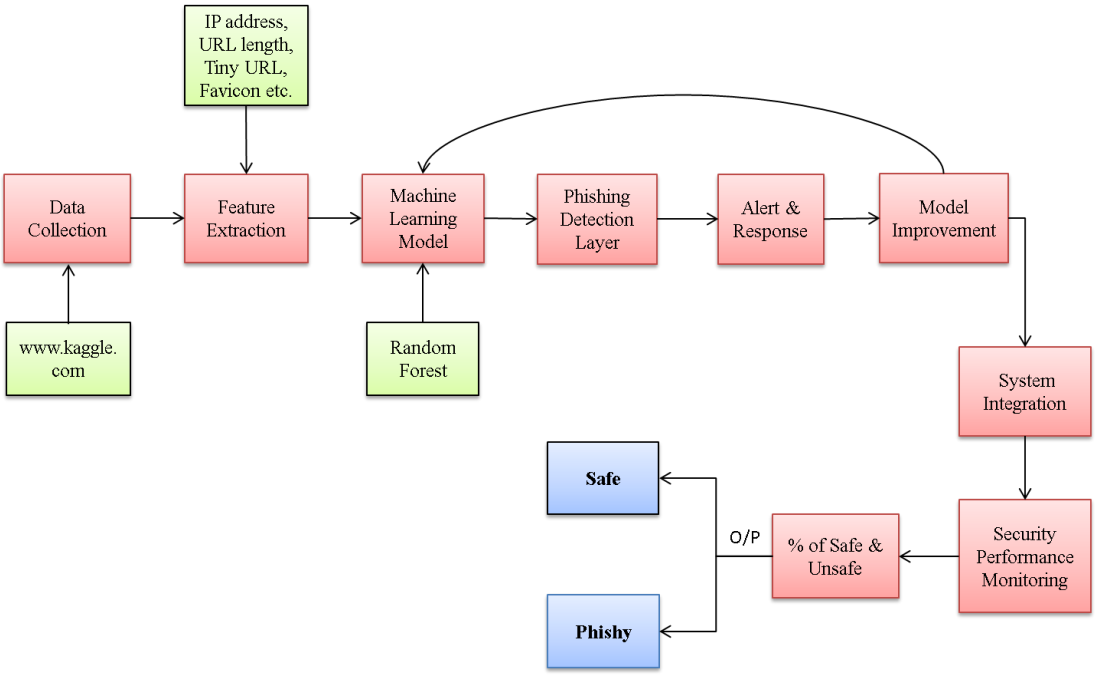


Fig 3.1: System Architecture.

1. **Data Collection Layer**

- **Input:** The system obtains data concerning URLs, HTML/CSS patterns of the Websites the user has accessed. content, and other metadata.
- **Sources:**
    - URL features (domain, length, IP addresses).
    - Website content (HTML, CSS structure, JavaScript).

2. Feature Extraction

- **Process:** Extracts relevant features from the collected data to identify phishing patterns.
    - **URL Features:** Length, domain age, presence of special characters, and IP addresses.
    - **HTML/CSS Features:** Anomalies in code structure, suspicious links, and forms.
    - **JavaScript:** Identification of suspicious scripts or redirection mechanisms.
- **Tools/Technologies:** Python, NumPy, and Scikit-learn for feature processing.

3. Machine Learning Model

- **Algorithm:** Random Forest Classifier.
- **Training Data:**
    - A labeled dataset containing legitimate and phishing websites.
    - Historical phishing data from CSV datasets (e.g., PhishTank, UCI repository).
- **Training Process:**
    - The model learns the distinguishing features between legitimate and phishing websites.

4. Phishing Detection Layer

- **Model Inference:**
    - The browser extension processes the current website through the trained Random Forest model. The system predicts whether the website is legitimate or a phishing attempt based on extracted features.
- **Real-Time Detection:**
    - The extension runs in real-time, immediately classifying a site as safe or phishing as soon as the page loads.
- **Tools:** JavaScript, HTML, jQuery, JSON for real-time detection within the browser environment.

5. Alert and Response System

- **User Interface (UI):** A simple, user-friendly browser extension interface that alerts users if a phishing site is detected. Visual indicators (e.g., warning pop-ups or icons) for phishing sites.
- **Actions:** The system blocks access to suspected phishing sites. Provides users with options (e.g., proceed with caution or return to safety).

6. Continuous Model Improvement

- **Feedback Loop:** The system gathers the response of the user to help enhance the model. Collaboration with cyber security experts ensure that there is a constant update and training of the model with the new data of the phishing.

7. Integration with Cyber security Networks

- **Data Sharing:** The extension can be complied with phishing databases or cyber-security networks to get the actual notifications concerning the new types of phishing.
- **Updates:** The system captures the new emerging phishing techniques and try to cat for it. from time to time in order to keep it relevant.

8. Security and system performance or utilization monitoring

- **Monitoring:** Records the pass and fail rate of the anti-phishing system as well as its ability to distinguish real phishing emails and resource utilization.
- **Optimization:** Facilitates minimized computational requirements making the system simple and easily executable in real time right within the browser.

3.2 Dataset Description

This Phishing dataset in this context is composed of several features which are meant to identify bad URLs. It has attributes like having IP Address, URL length and Shortening Service to determine the structural characteristic of URL. User Interaction characteristics are reflected in the features including on_mouseover, Right Click, and popUpWindow. The set also assesses security features such as SSLfinal_State, Domain_registration_length, HTTPS_token. In general, the dataset consists of diverse categorical variables that result in the determination of phishing or legitimate URLs for ML-based phishing detection. The details are described in the following subtopics.

Table 3.2: Overview of the Dataset

Attribute	Description	Values/Range
IP Address	Indicates whether the URL contains an IP address	{-1, 1}
URL Length	Length of the URL	{1, 0, -1}
Tiny URL	URL shortened using a service (like bit.ly)	{1, -1}
@ Symbol	Presence of '@' symbol in the URL	{1, -1}
Redirecting using //	Redirects using '//' in the URL	{-1, 1}
(-) Prefix/Suffix in domain	Indicates prefix or suffix usage in the domain	{1, -1}
No. of Sub Domains	Number of subdomains in the URL	{1, 0, -1}
HTTPS	Indicates the presence of HTTPS in the URL	{1, -1}
Favicon	Imported from an external source	{1, -1}
Using Non-Standard Port	Whether a non-standard port is used in the URL	{1, -1}
HTTPS in URL's domain part	Indicates if 'HTTPS' appears in the domain part of the URL	{-1, 1}
Request URL	Percentage of external objects in the URL	{1, -1}
URL of Anchor	Percentage of suspicious links in anchors	{1, 0, -1}
Links in script and link	Links embedded in script or link tags	{1, -1, 0}
Server Form Handler	Indicates if the form handler is a legitimate domain	{1, -1}
Submitting to mail	Form submission via email	{1, -1}
Using iFrame	Indicates the use of an iFrame	{1, -1}

3.3.Performance Metrics

To determine how accurate the phishing detection system is, the portion of the World Wide Web sites that the system is able to identify as safe is determined by the following formula. The specification equation developed entails thereby covering the number of websites and various parameters that have been applied for the classification.

$$\text{Percentage of Safe Websites} = \frac{\sum\left(\frac{Safe\ Website\ Count\ Per\ Parameter}{Total\ Website\ Count}\right)}{Total\ Number\ Of\ Parameters}\ X\ 100$$

Where:

- **Safe Website Count Per Parameter:** Number of websites classified as safe based on a specific parameter (e.g., URL structure, SSL certificate, etc.).
- **Total Website Count**: Number of websites analysed by the system during evaluation.
- **Total Number of Parameters:** Number of factors used in the classification.

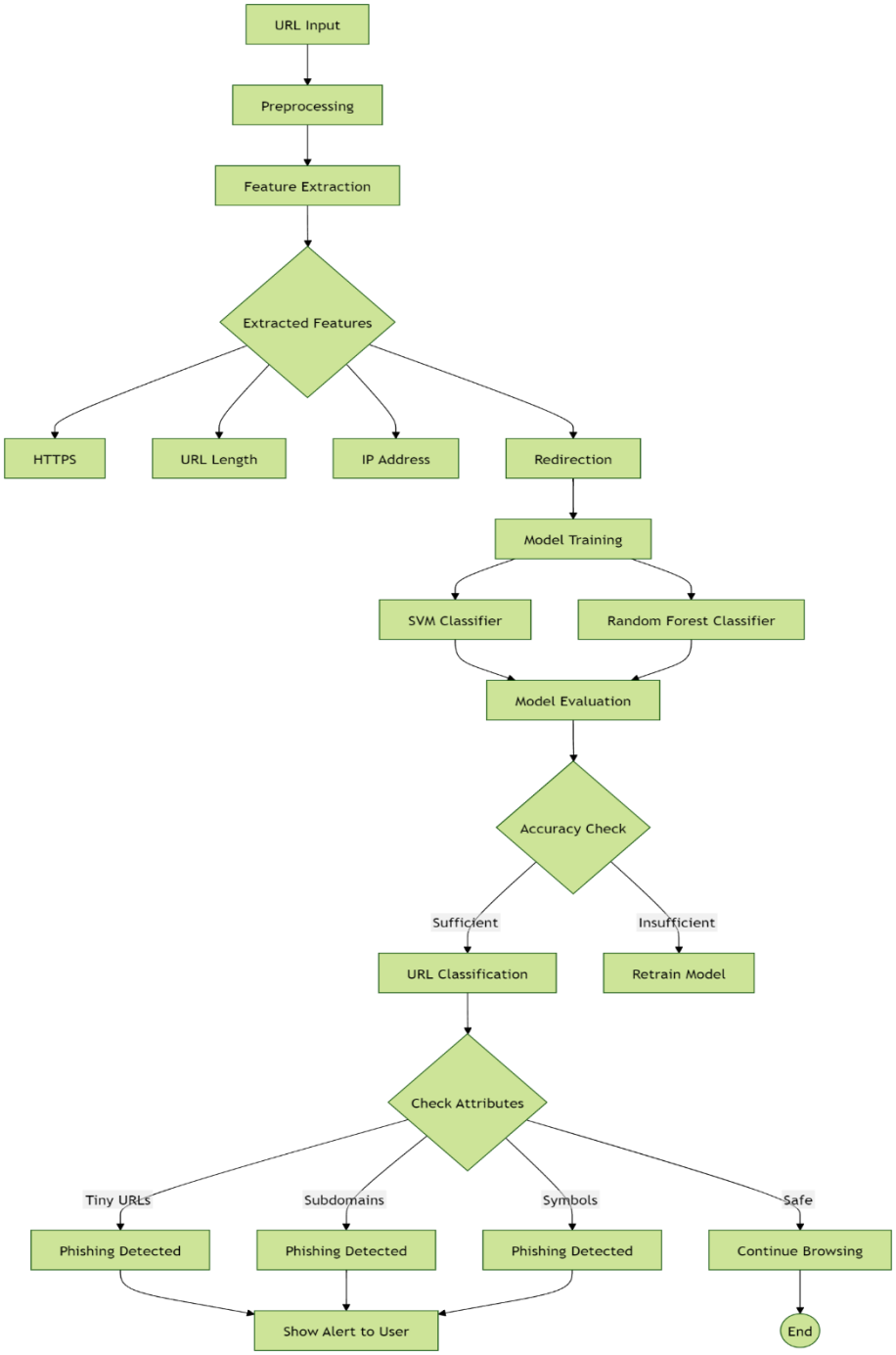
Table 3.3: Phishy vs. Safe Websites: A Percentage-Based Analysis

Website Name	Safe Website Count Per Parameter	Total Website Count	Safe Website Percentage (Per Parameter)	Classification
www.govsite1.gov	15	20	(15 / 20) * 100 = 75%	Safe
www.jobofferscam.net	3	20	(3 / 20) * 100 = 15%	Phishy
www.safebanking.com	17	20	(17 / 20) * 100 = 85%	Safe
www.phishingalerts.biz	5	20	(5 / 20) * 100 = 25%	Phishy
www.trustedgov.org	19	20	(19 / 20) * 100 = 95%	Safe
www.freegiveaway.co	4	20	(4 / 20) * 100 = 20%	Phishy
www.healthsafe.gov	18	20	(18 / 20) * 100 = 90%	Safe
www.reliablegov.info	17	20	(17 / 20) * 100 = 85%	Safe
www.clickbaitoffers.biz	4	20	(4 / 20) * 100 = 20%	Phishy
www.securecharity.org	16	20	(19 / 20) * 100 = 95%	Safe
www.suspiciouslogin.com	3	20	(3 / 20) * 100 = 15%	Phishy
www.phishydeals.com	2	20	(2 / 20) * 100 = 10%	Phishy
www.hdhub4u.wales	7	20	(7 / 20) * 100 = 35%	Phishy

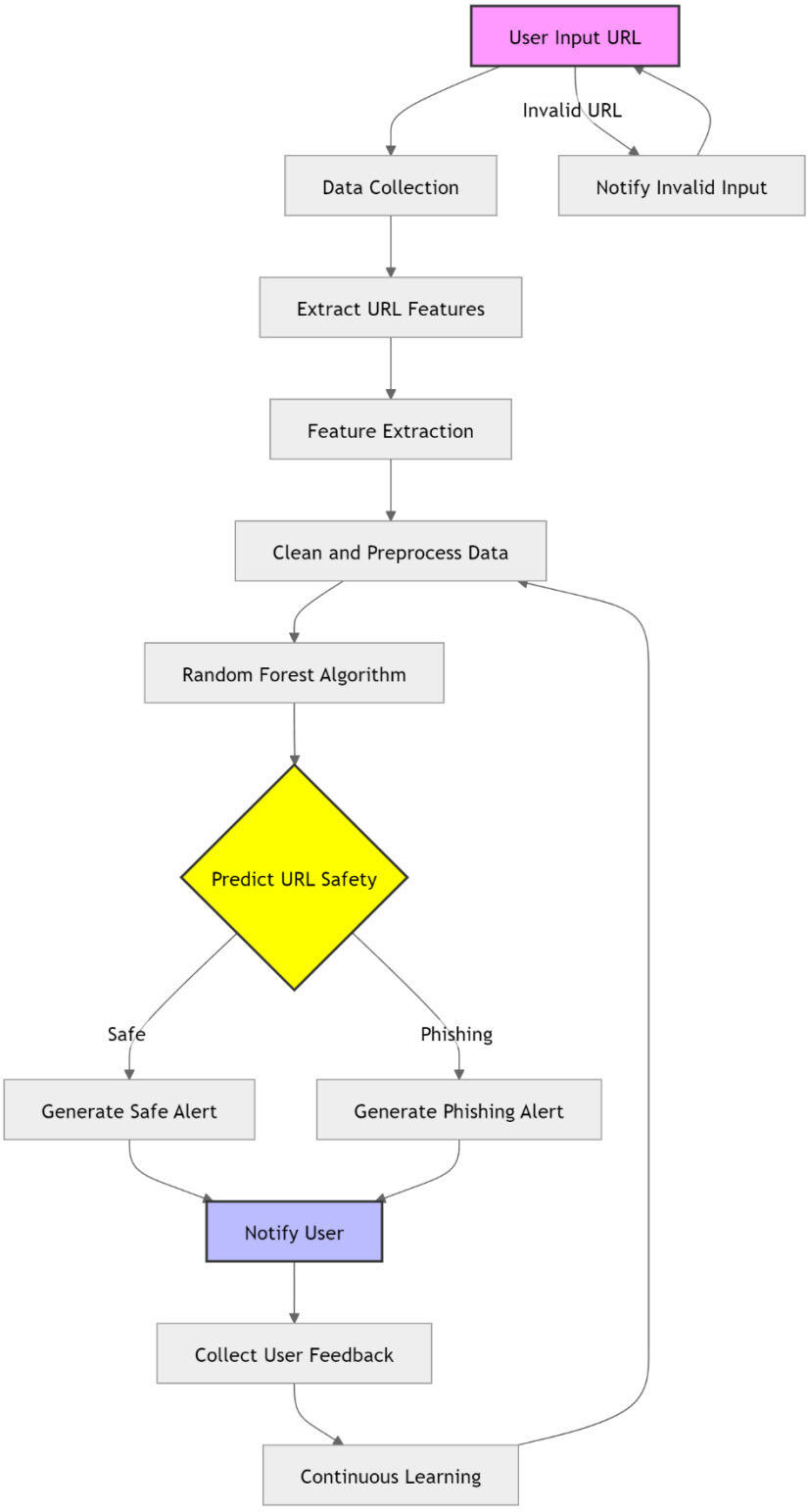
CHAPTER 4

SYSTEM DESIGN

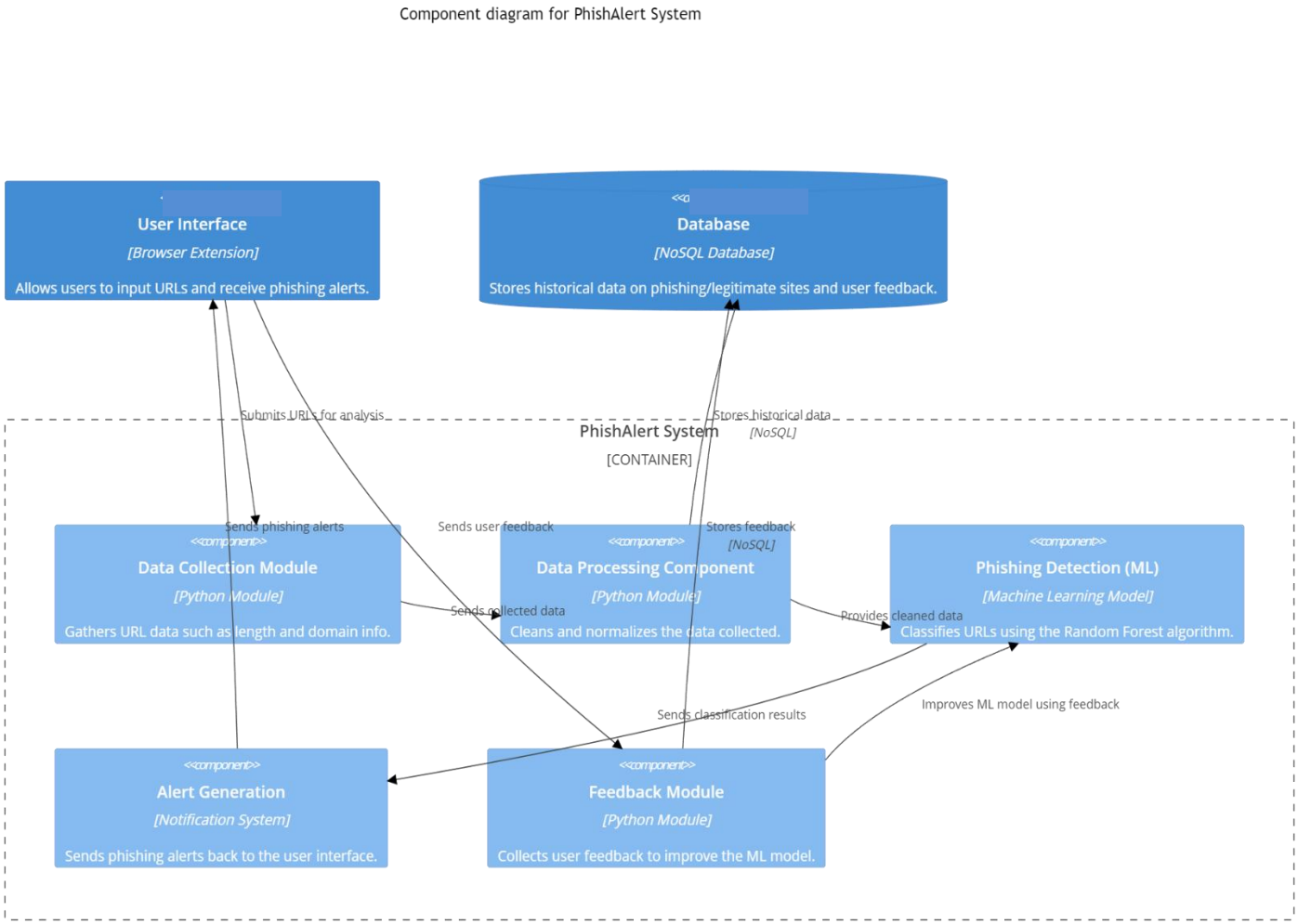
4.1 System Design Daigram:



4.2 Process Flow Diagram:



4.3 Component Diagram:



CHAPTER 5

5.1 Project Plan

- Presentation and Synopsis Submission
- Project Initialization
- Requirement Analysis
- Design Phase
- Implementation
- Testing and Validation
- Deployment
- Evaluation and Optimization
- Deployment and Optimization

5.1.1 Presentation and Synopsis Submission:

The project began with a comprehensive presentation of **PhishAlert: An Intelligent Browser Extension for Phishing Detection** to stakeholders. The initial proposal outlined an innovative approach to combating phishing through real-time detection using machine learning. The synopsis detailed the development of a browser extension utilizing the **Random Forest algorithm** to analyze website features such as URLs and metadata for accurate phishing identification. The presentation emphasized the system's ability to provide real-time alerts and continuous monitoring to protect users from phishing threats. Stakeholders were particularly interested in the feedback loop mechanism that incorporates user reports to refine detection accuracy and adapt to evolving phishing tactics. The focus on user-friendliness was also highlighted, showcasing how PhishAlert's interface is designed for users with varying technical expertise. Stakeholders appreciated its potential for wide application in personal, corporate, and educational settings, emphasizing its role in enhancing online security and trust.

5.1.2 Project Initialization:

Following approval, the project entered the initialization phase, focusing on establishing the core framework for development. **Visual Studio Code** was chosen as the primary development environment due to its robust support for Python applications, while **Google Colab** was utilized for model training, leveraging its GPU acceleration capabilities.

A structured project timeline was developed to guide each phase, from data collection to final deployment. This phase also included setting up the development environment by installing essential Python libraries, configuring API access for alert notifications, and implementing version control protocols for effective collaboration. Resource allocation was meticulously planned to ensure smooth and efficient progression through all development stages.

5.1.3 Requirement Analysis:

The requirement analysis phase involved a thorough assessment of the functional and technical needs of **PhishAlert: An Intelligent Browser Extension for Phishing Detection**. It was determined that the system needed to support various website input types, including dynamic URLs and metadata analysis. The analysis highlighted the need for a robust real-time alert system capable of delivering notifications directly through the browser interface. Security requirements were defined, leading to the implementation of user data protection measures to ensure privacy and secure handling of sensitive information. Performance requirements emphasized the importance of real-time detection capabilities and high accuracy in identifying phishing websites. This phase played a vital role in shaping the system's architecture, ensuring that all stakeholder needs—such as real-time monitoring, user-friendly design, and data security—were fully addressed, setting a strong foundation for the project’s development.

5.1.4 Design Phase:

During the design phase, a comprehensive system architecture was developed to support the core functionalities of PhishAlert. The system was built around a user-friendly browser extension interface, designed for seamless integration and interactive user experience. The architecture utilized the Random Forest algorithm as the core detection engine, supported by a robust feature extraction pipeline to ensure high accuracy in phishing detection. The alert system was designed with redundancy, providing real-time notifications via browser pop-ups and other integrated channels. Special focus was given to the interface design, ensuring intuitive navigation and clear presentation of detection results. The modular system design facilitated easy maintenance and allowed for future upgrades to incorporate emerging phishing techniques.

5.1.5 Implementation:

The implementation phase began with developing the core Random Forest model, trained on a dataset of phishing and legitimate websites. The PhishAlert browser extension was designed with a user-friendly interface for seamless interaction. Feature extraction algorithms were implemented to enhance detection accuracy by analyzing URL structures, HTML content, and metadata.The alert system was integrated to deliver real-time notifications directly through the browser, providing users with instant warnings about potential phishing threats. Security measures were incorporated, including data encryption and secure handling protocols to protect user information. Throughout the implementation, special attention was given to code optimization and system performance, ensuring efficient real-time detection while minimizing computational overhead.

5.1.6 Testing and Validation:

Comprehensive testing was conducted to ensure the reliability and accuracy of PhishAlert. The Random Forest model underwent validation to confirm accurate

phishing detection across diverse website structures and URL patterns. Feature extraction algorithms were tested to maintain consistent performance with different web inputs. The alert system was rigorously evaluated to ensure prompt, reliable delivery of real-time notifications. Performance testing assessed system response times and resource utilization under varying load conditions. Security features were thoroughly validated to guarantee secure data handling and access control. Additionally, user interface testing ensured intuitive navigation and clear presentation of phishing alerts, providing an optimal user experience for all users.

5.1.7 Deployment:

The deployment phase involved the systematic implementation of PhishAlert in its intended environment. The Streamlit application was deployed with all necessary dependencies and configurations, ensuring smooth functionality. Alert system credentials and APIs were properly configured to guarantee seamless notification delivery. The deployment process included comprehensive documentation of setup procedures and system requirements, ensuring ease of future maintenance. Environmental variables and security credentials were carefully managed to maintain system security. Additionally, monitoring tools were set up to track system performance and stability, ensuring the system operated efficiently in real-world conditions.

5.1.8 Evaluation and Optimization:

Following deployment, PhishAlert underwent continuous evaluation to assess its performance in real-world conditions. Performance metrics were collected and analyzed to identify areas for optimization, including monitoring phishing detection accuracy and fine-tuning the model based on real user interactions. User feedback was gathered to improve the interface and functionality, ensuring a better experience. System resource utilization was regularly tracked to ensure efficient operation. Based on this evaluation, several optimizations were implemented, including enhancements to the feature extraction pipeline and improvements to the alert system’s response times, boosting overall system performance and reliability.

	Day 15	Day 35	Day 50	Day 80	Day 100	Day 115	Day 130	Day 140	Day 150
Presentation and Synopsis Submission	Group Formation								
Project Initialization		10%							
Project Initialization			15%						
Requirement Analysis			25%						
Design Phase				50%					
Implementation						60%			
Testing and Validation							75%		
Deployment								90%	
Evaluation and Optimization								100%	

Table 4. Project Plan

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules:

The **PhishAlert** project consists of several key modules, each playing a crucial role in real-time phishing detection:

- **User Interface Module**: Provides a browser extension for users to input URLs, view phishing detection results, and submit feedback on detected phishing sites.
- **Data Collection Module**: Gathers URL data, including attributes like template, age, and SSL certificates, ensuring persistent data input.
- **Feature Extraction Module**: Analyses key URL characteristics, such as length and special symbols, and prepares them for the machine learning model.
- **Phishing Detection Module**: Uses the **Random Forest algorithm** and data mining techniques to classify URLs as safe or phishing.
- **Alert Generation Module**: Alerts users in real-time if a phishing attempt is detected.
- **User Feedback Module**: Collects user feedback on phishing detection accuracy to improve the model.
- **Data Storage Module**: Stores historical phishing data, user feedback, and legitimate websites for continuous model updates.

6.2 Tools and Technologies Used:

The development of **PhishAlert** utilized several key tools and technologies to ensure efficient real-time operation:

- **Python**: The primary programming language used to implement the **Random Forest** machine learning algorithm and data manipulation procedures.

- **Scikit-learn**: A Python library for building and deploying the **Random Forest** model to detect phishing sites.
- **HTML, CSS, JavaScript**: Technologies used to develop the browser extension's **GUI**, frontend integration, and user interfaces compatible with web browsers.
- **MySQL**: Databases for securely storing and retrieving phishing URLs, user feedback, and related data.
- **Jupyter Notebooks**: Used for experimenting with new models and testing system performance with datasets.
- **Amazon Web Services (AWS)**: Used for cloud computing to host the system and handle large-scale data processing and storage.

6.3 Algorithm Details:

6.3.1 Algorithm 1: Feature extraction Algorithm for URL.

Description: It is this particular algorithm that is used to identify useful features from URL input by users. These features are basic for winding up a differential indicator of whether the URL is a phishing one or not. These include; URL length, presence of Secure Sockets Layer certificate, age, and specific URL patterns extracted from the dataset.

Steps Involved:

1. Get the URL invoked by the user with the help of the browser extension.
2. Key characteristics include analyzing and extracting the length of URL, the registration period of the domain, as well as the-existent peculiarities or even certain characters.
3. Preprocess and sample the features into a machine learning for easy inputs.
4. After the extraction of necessary data send it to the Phishing Detection module for further processing.

6.3.2 Algorithm 2: Random Forest Phishing Detection Algorithm:

Description: Using the features described above this machine learning algorithm separates the URLs into the two classes: safe and phishing. The Random Forest decision tree model examines the input features and then just predicts results from learned data.

Steps Involved:

1. Using the extracted URL features, feed the data to the Random Forest model.
2. Then the features are passed through the decision trees to determine whether the URL is of a phishing site or not a phishing one.
3. The output of the classification result (safe/ phishing) is handed over to the Alert Generation module.
4. In case the site is categorized as phishing the user receives an instant notification from the system.
5. The model should be periodically trained with the new data on phishing and have feedback from users to be more accurate.

CHAPTER 7

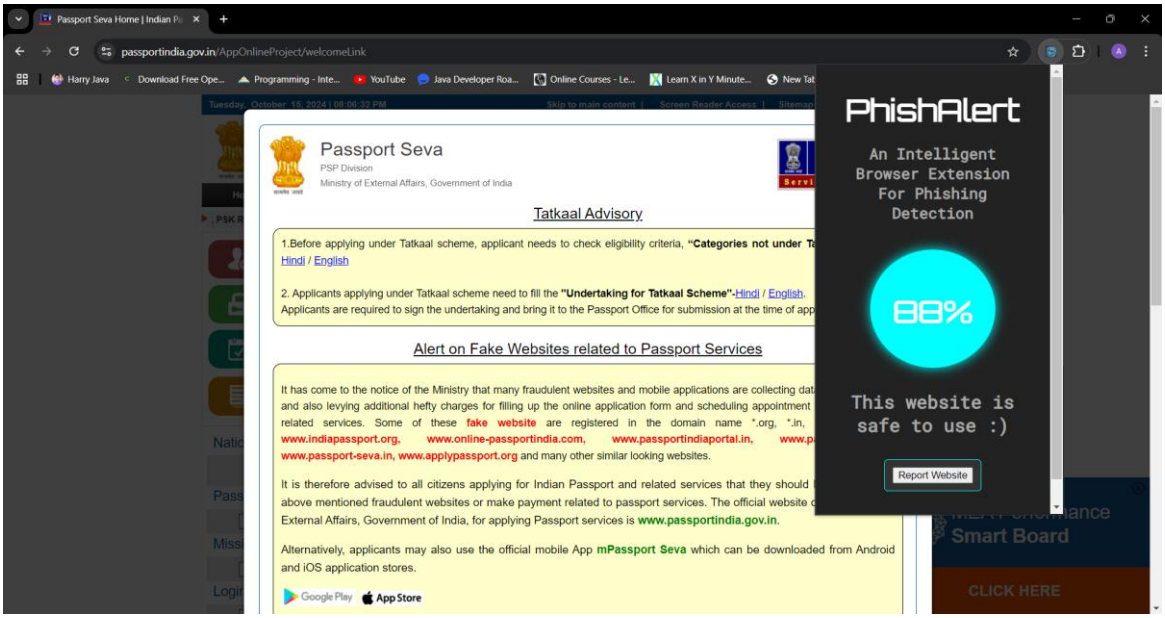
RESULTS

7.1 Model Comparison:

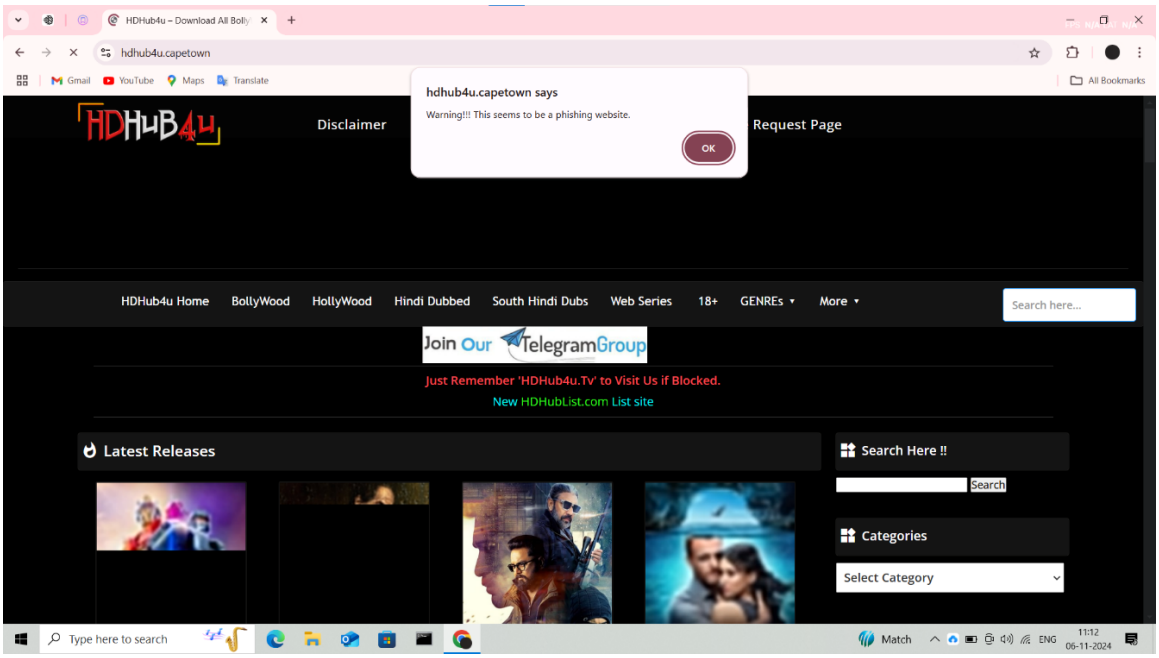
The proposed Smart Browser Extension fits the modern-day problem of inadequate phishing identification and more rigid internet safety. Using the Random Forest algorithm to analyze web properties, the extension informs users about potential phishing threats in time and help users to protect their personal information. Data from the testing show that the system effectively detects phishing websites to enhance the users' protection. Recursive enhancement possibilities may include moving from more basic, conventional machine learning algorithms to superior, cutting-edge models and polishing usability issues of the software to create a much stronger, more secure, and more simplistic interface to safeguard the consumer against increasingly sophisticated phishing strategies. The comparison of the proposed model's results with those of other models is shown below.

Model	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Remark
Model 1	Random Forest	93	91	90	90.5	High accuracy with balanced metrics
Model 2	Support Vector Machine (SVM)	88	85	86	85.5	Good for binary classification
Model 3	Decision Tree	85	82	84	83	Simpler model with quick execution
Model 4	Naive Bayes	80	78	79	78.5	Lower accuracy but fast classification

7.2.Screen Shots:-

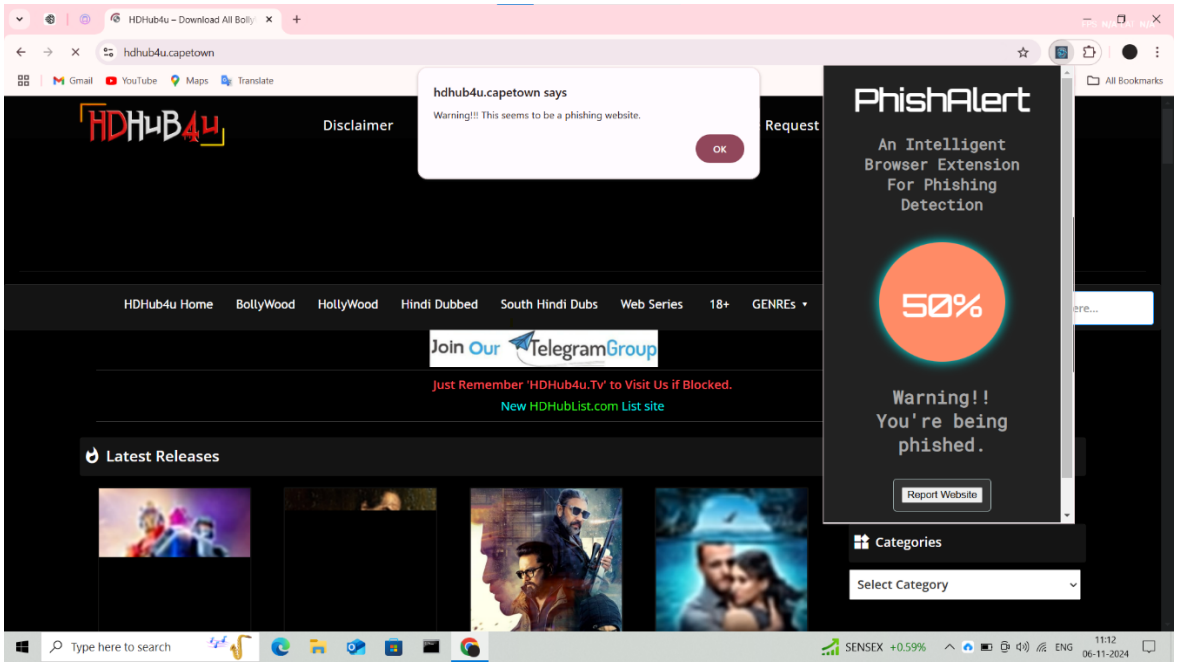


a) Safe website popup

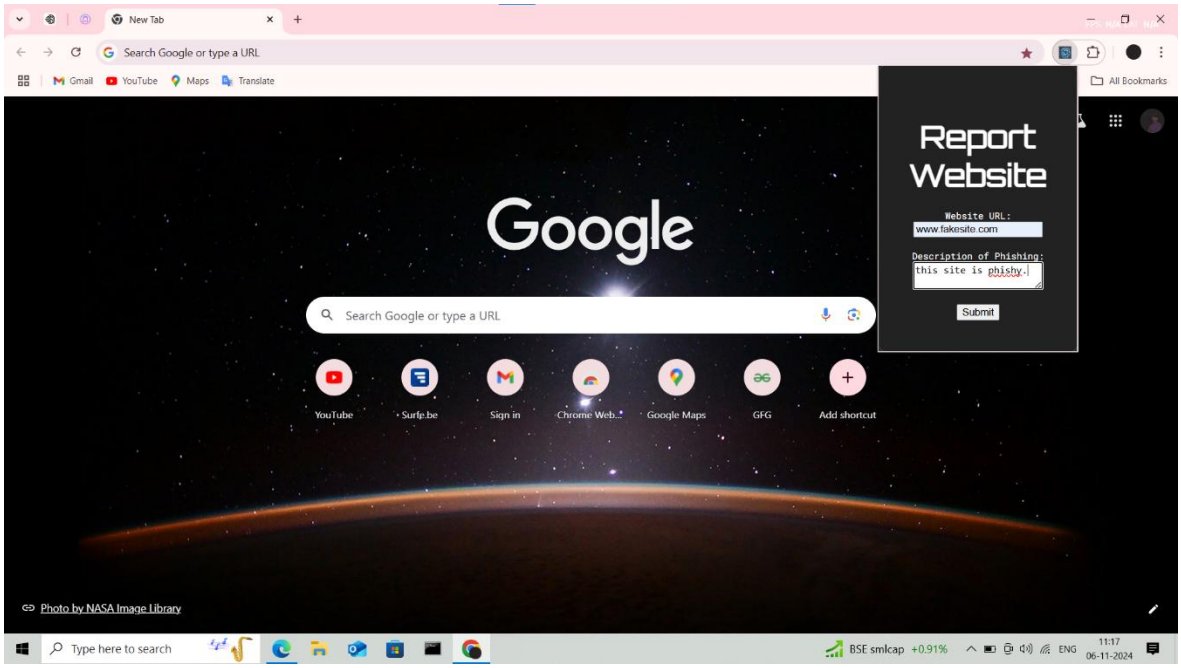


S

b) Phishing Detection Warning Interface



c) Extension Presence



d) Phishing Website Reporting Form

CHAPTER 8

CONCLUSION

8.1 Conclusion:

In conclusion, the first objective of this study is to establish that the Smart Browser Extension can accurately detect phishing domains in real time utilizing the Random Forest algorithm. Some other features of the extension include: Multiple suspicious URL detection, Alerts the user concerning potential phishing attempts, and provides an explanation for each detection made. The extension remains active, always scanning the web traffic and assigning each site as safe or unsafe depending on string and page properties. Through consultation of user feedback and thorough assessment of the system, it aims at improving the general online security, decrease cases of phishing, and safeguard data. Screenshots that follow demonstrate how simple the extension is and that it is capable of providing users with accurate phishing detections at the right time. As for comparison of the results of the proposed model with the results of other models, the following is provided in the paper.

8.2 Future Work

The future development of **PhishAlert** will focus on several key areas to enhance its capabilities:

- **Enhanced Detection Algorithms**: Integrating advanced algorithms, such as deep learning models, to improve phishing detection sophistication and handle more complex phishing attacks.
- **Expanded Browser Compatibility**: Developing versions of the extension for additional browsers (e.g., Safari, Edge) to broaden user base and availability.
- **Phishing Prevention Analytics**: Providing users with deeper insights into their browsing patterns and associated phishing risks.
- **User Engagement and Education**: Adding educational content on phishing schemes and prevention measures to raise awareness among users.

- **Gamification for User Feedback**: Introducing gamification elements (e.g., badges, rewards) to encourage users to provide feedback on false positives/negatives, improving model updates.
- **Performance Optimization**: Enhancing system speed and efficiency, particularly for data verification, while maintaining performance under heavy browser load.

8.3 Applications

PhishAlert has various practical applications in real-world scenarios:

- **Cybersecurity in Educational Settings**: Protects students and staff in schools and universities from phishing threats while using institutional systems, ensuring safe browsing.
- **Enterprise Security**: Safeguards employees in corporate environments, especially in sectors like finance, healthcare, and law, by securing their web interactions.
- **Personal Browsing Protection**: Provides individual users with real-time phishing alerts to prevent entering personal information on fake websites.
- **Support for Online Research**: Helps researchers and journalists avoid phishing scams when conducting surveys or gathering information online.
- **Cybersecurity Training**: Can be incorporated into cybersecurity awareness campaigns as an educational tool, helping users identify phishing threats and scams.

References

[1] Sindhu, S., Patil, S. P., Sreevalsan, A., & Rahman, F. (2020). Phishing detection using Random Forest, SVM, and Neural Networks with backpropagation. 2020 IEEE 19th International Conference on Trust, Security & Privacy in Computing and Communications. B.M.S. College of Engineering, Bengaluru, India.

[2] Stobbs, J., & Issac, B. (2017). Phishing web page detection using optimized machine learning techniques. 2017 International Conference on Electrical & Computing Technologies and Applications (ICECTA), Northumbria University, Newcastle-upon-Tyne, UK.

[3] Subasi, A., Molah, E., Almkallawi, F., & Chaudhery, T. J. "Intelligent Phishing Website Detection Using Random Forest Classifier." Conducted at the College of Engineering, Effat University, Jeddah, Saudi Arabia.

[4] Wu, J., et al. (2021). "Malicious Web Request Detection Using a Convolutional Neural Network with Character Embeddings.".

[5] Razaque, A. (2020). "Machine Learning-Based Phishing Website Detection." Computer Engineering & Telecommunications Department, International IT University, Almaty, Kazakhstan.

[6] Alkawaz, M. H. (2021). "Comprehensive Survey on Phishing Website Detection Using Machine Learning Methods." Faculty of Information Sciences, Engineering Management and Science University, Shah Alam, Selangor, Malaysia.

[7] Desai, A. (2017). "Detecting Malicious Web Content Using Machine Learning." Sardar Patel Institute of Technology.

[8] Linh, D. M., et al. (2021). "Real-Time Phishing Detection Through Deep Learning Techniques.

[9] Ivanov, M. A. (2021). "Phishing Attacks and Their Countermeasures." National Research Nuclear University MEPhI, Moscow, Russia.Maurya, S. (2020). "A Hybrid Anti-Phishing Framework Based on Browser Extensions and Feature Selection." Guru Gobind Singh Indraprastha University, New Delhi, India.

[10] Shahrivari, V. (2020). "Detecting Phishing Websites Using Machine Learning Techniques." Sharif University of Technology, Tehran, Iran.

[11] Mohammad, R. M., Thabtah, F., & McCluskey, L. "Predicting Phishing Websites Using a Self-Structuring Neural Network." Neural Computation and Applications, Vol. 25, No. 2, pp. 443–458, 2014.

[12] Gastellier-Prevost, S., Granadillo, G. G., & Laurent, M. (2011). "Decisive Heuristics for Differentiating Legitimate and Phishing Sites." Network and Information Systems Security (SAR-SSI), 2011 Conference.

[13] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 14, no. 2, p. 21, 2011.

[14] N. Sanglerdsinlapachai and A. Rungsawang, "Using domain top-page similarity feature in machine learning-based web phishing detection," *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*, 2010, pp. 187–190.

[15] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*, Elsevier, 2011.

[16] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Inf. Secur.*, vol. 8, no. 3, pp. 153–160, 2014.

[17] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7913–7921, 2010.

[18] M. He et al., "An efficient phishing webpage detector," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12018–12027, Sep. 2011.

[19] M. S. Arade, P. Bhaskar, and R. Kamat, "Antiphishing model with URL & image-based webpage matching," *Int. J. Comput. Sci. Technol. IJCST*, vol. 2, no. 2, pp. 282–286, 2011.

APPENDIX A

Copyright Certificate

FORM XIV
APPLICATION FOR REGISTERATION OF COPYRIGHT
[SEE RULE 70]

Diary Number:

To

The Registrar of Copyrights,
Copyright Office,
Department of Industrial Policy & Promotion,
Ministry of Commerce and Industry,
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110075
Email Address: copyright@nic.in
Telephone No.: (Office) 011-28032496, 08929474194

Sir,

In Accordance with Section 45 of the Copright Act, 1957 (14 of 1957), I hereby apply for registration of Copyright and request that enteries may be made in the Register of Copyrights as in the enclosed Statement of Particulars.

1. I also send herewith duly completed the Statement of further Particulars relating to the work. (for Literary/Dramatic, Musical, Atristic works only) **Literary/ Dramatic works**

2. In accordance with rule 16 of the Copyright Rules, 1958, I have sent by prepaid registered post copies of this letter and of the Statement of Particulars and Statement of Further Particulars to other parties concerned as shown below:

Name of Party	Address of Party	Date of Dispatch
DR.DEVIDAS S THOSAR	G H RAISONI COLLEGE OF ENGINEERING AND MANAGMENT PUNE-412207	

[See columns 7,11,12, and 13 of the Statement of Particulars and party referred in col.2 (e) of the Statement of Further Particulars.]

3. The prescribed fee has been paid, as per details below:

4. Communications on this subject may be addressed to:

DR.DEVIDAS S THOSAR
G H RAISONI COLLEGE OF ENGINEERING AND MANAGMENT PUNE-412207
8888805312

5. I hereby declare that to the best of my knowledge and belief, no person, other than to whom a notice has been sent as per paragraph 2 above any claim or interest or dispute to my copyright of this work or its use by me.

6. I hereby verify that the particulars given in this Form and the Statement of Particulars and Statment of Further Particulars are true to the best of my knowledge, belief and information and nothing has been concealed there from.

List of Enclosures:

1. 2 Copies of Work
2. DD/IPO of Rs.0 Per Work
3. Authorization from author/publisher
4. If the application is being filed through attorney , a specific Power of Attorney in original duly signed by the applicant and accepted by the attorney

Place:

Date: **12/11/2024**

For : ANIKET KHATAVKAR

11/13/24, 9:17 AM Copyright Office

Proprietor

APPENDIX B

PLAGIARISM REPORT

APPENDIX C

PAPER PUBLICATION SUMMARY

1) Paper Title: “A Smart Browser Extension for Live Phishing Detection Using Random Forest Algorithm “

2) Name of the Conference where paper Submitted: RBUCON.Computing’25

3) Paper Accepted/Rejected : In proceses

4) Date of Conference: 20th and 22th of February 2025

Phishalert: An Intelligent Browser Extension For Phishing Detection