PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

VAVILLA YASHWANTH KUMAR REDDY (20UECS1003) **(VTU12688)**

Under the guidance of
Dr.N.VIJAYARAJ, M.E, Ph.D .,
PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL

Major project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

VAVILLA YASHWANTH KUMAR REDDY (20UECS1003) **(VTU12688)**

Under the guidance of
Dr.N.VIJAYARAJ, M.E, Ph.D.,
PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL” (IN CAPITAL LETTER)” by ”VAVILLA YASHWANTH KUMAR REDDY (20UECS1003)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(VAVILLA YASHWANTH KUMAR REDDY)
Date: / /

(Signature)
(STUDENT NAME2(IN CAPITAL LETTER)
Date: / /

(Signature)
(STUDENT NAME3(IN CAPITAL LETTER)
Date: / /

APPROVAL SHEET

This project report entitled (PHISHING DETECTION SYSTEM THROUGH HYBRID MACHINE LEARNING BASED ON URL) by VAVILLA YASHWANTH KUMAR REDDY (20UECS1003), is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Dr.N.Vijayaraj, M.E, Ph.D., Professor.

Date: / /

Place:C

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering,Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr.N.VIJAYARAJ, M.E, Ph.D., PROFESSOR)** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

VAVILLA YASHWANTH KUMAR REDDY (20UECS1003)

ABSTRACT

Phishing was first used in 1996, it has become the most severe and dangerous cybercrime on the internet. Phishing utilizes email distortion as its underlying mechanism for tricky correspondences, followed by mock sites, to obtain the required data from people in question. Different studies have presented their work on the precaution, identification, and knowledge of phishing attacks; however, there is currently no complete and proper solution for frustrating them. Machine Learning plays a vital role in defending against cybercrimes involving phishing attacks.The proposed studyis based on the phishing URL-based dataset extracted from the famous dataset repository, which consistsof phishing and legitimate URL attributes collected from 11000+ website datasets in vector form. Many machine learning algorithms have been applied and designed to prevent phishing URLs and provide protection to the user.This study uses machine learning models such as decision tree(DT), linear regression (LR), random forest (RF), naive Bayes (NB), gradient boosting classifier (GBM),K-neighbors classifier (KNN), support vector classifier (SVC), and proposed hybrid LSD model.Hence, detecting recently developed phishing websites in a real-time environment is a great challenge in the domain of cybersecurity. To overcome these problems, this paper proposes a hybrid feature based anti-phishing strategy that extracts features from URL and hyperlink information of client-side only. We also develop a new dataset for the purpose of conducting experiments using popular machine learning classification techniques. Our experimental result shows that the proposed phishing detection approach is more effective having higher detection accuracy of 99.17 percentage with the XG Boost technique than traditional approaches.

Keywords:Phishing detection, Machine learning, Hyperlink feature, URL feature, Anti-phishing, XG Boost, Hybrid feature.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

URL	Uniform Resource Locator
DT	Decision Tree
RF	Random Forest
LR	Linear Regression
NB	Naive Bayes
SVC	Support Vector Classifier
GBC	Gradient Boost Classifier

TABLE OF CONTENTS

Page.No

Chapter 1

INTRODUCTION

1.1 Introduction

Phishing nowadays is one of the most serious and dangerous online threat in the domain of cybersecurity. The use of social networks, e-commerce, electronic banking, and other online services has been increased immensely due to the rapid development of internet technologies.Global Overview Report 2021, released “A Digital Report in 2021” data that the internet users have grown to 4.66 billion with an increase of 7.3 percent (316 million new users) compared to January 2020. At present, internet penetration stands at 59.5 percent which provides an opportunity to make money for a phishing attacker by blackmailing and stealing confidential information from internet users.

When a user unwittingly clicks the link and updates any sensitive credentials, cyber attackers gain access to the user’s information like financial data, personal information, username, password, etc. This stolen information is used by cybercriminals for a variety of illegal activities, including blackmailing victims.

The attacker develops a fraudulent website and sends links to online platforms like Facebook, Twitter, emails, etc by conveying a message of panic, urgency, or a financial bid, and instructs the recipient to take immediate action. There are five reasons why users fall for phishing:

Users do not have a deep understanding of URLs, Users are unsure of which websites they should rely on, Due to redirection or secret URLs, users are not able to see the entire address of the web page, Users don’t have much time to look up a URL or unconsciously visit certain web pages, Users are unable to differentiate between legal and phishing websites.

There are various types of Phishing attacks which are been used by the attackers for various domains for different purpose. Since phishing is such a widespread problem in the cyber-security domain, there is a necessity of phishing website detection.

1.2 Aim of the project

Phishing Detection System through Hybrid Machine Learning based on URL is to develop a robust and efficient system that can identify and prevent phishing attacks. Phishing is a type of cyber attack where attackers trick individuals into revealing sensitive information, such as usernames, passwords, or financial details, by posing as a trustworthy entity. Detecting phishing URLs is crucial in preventing users from falling victim to such attacks. .

1.3 Project Domain

The project ”Phishing Detection System through Hybrid Machine Learning Based on URL” focuses on developing a sophisticated system to combat the growing threat of online phishing attacks. By leveraging a hybrid approach to machine learning, the system aims to enhance its detection capabilities by incorporating various algorithms and methodologies.

This hybridization could involve combining the strengths of different machine learning techniques such as supervised learning for labeled data, unsupervised learning for anomaly detection, and possibly reinforcement learning for dynamic adaptation to emerging phishing strategies. Such a multifaceted approach allows the system to adapt and evolve in response to the evolving tactics employed by malicious actors in the realm of phishing.

1.4 Scope of the Project

It involves the extraction of relevant features from URLs using a combination of supervised and unsupervised learning techniques. The system will operate in real-time, enabling prompt identification and prevention of phishing attacks.

The project encompasses user-friendly interfaces, feedback mechanisms, and integration with existing security infrastructure. Additionally, scalability, performance optimization, and continuous improvement mechanisms will be emphasized to ensure an effective and adaptable solution against evolving phishing threats.

Chapter 2

LITERATURE REVIEW

V. Rohokale et al.,[1], have proposed “Cyber threats and attack overview,” in Cyber Security: The Lifeline of Information and Communication Technology, 2020. Used both machine learning and image checking techniques to detect phishing websites by extracting heuristic features from URL, source code, and third-party services. Their proposed model performed very well with a high accuracy rate, but the design architecture of the network is quite complex. They took the most correlated features by utilizing features selection. Though they achieved a high accuracy rate, their approach is not fit for real-time phish detection.

P. Komisarczuk et al.,[2], have proposed “An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,” in Proc. Australas. Comput.Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY,USA: Association for Computing Machinery, 2020. Implemented a phishing attack detection model using the self-structuring neural network. The authors used the backpropagation algorithm for the weight adjustment of the network. They used 17 features collected from the URL, source code of the website, and also third-party services. The detection time is increased for using the third-party based features. However, their test set accuracy was 92.18 with 1000 epochs.

R. Singh et al.,[3] introduced a hybrid phishing detection system leveraging machine learning techniques in their paper ”Hybrid intelligent model for phishing detection using ensemble learning,” presented at the International Conference on Intelligent Computing and Applications (ICICA) in 2018. Their model combined multiple classifiers, including decision trees, random forests, and support vector machines, to enhance phishing detection accuracy. Experimental results demonstrated the effectiveness of the ensemble approach, achieving a high detection rate and low false-positive rate.

A. Kumar et al.,[4] proposed a phishing detection system based on URL features using machine learning algorithms in their paper ”An efficient phishing URL detection model using machine learning techniques," published in the Journal of Intelligent , Fuzzy Systems in 2020. The authors extracted features from URLs and employed machine learning classifiers such as logistic regression and k-nearest neighbors to classify phishing URLs. Their model achieved promising results in terms of accuracy and efficiency, demonstrating its potential for real-time phishing detection.

S. Shahriar et al.,[5] presented a hybrid phishing detection system combining machine learning and deep learning techniques in their paper "Deep learning for phishing detection: A hybrid approach using convolutional neural networks and LSTM," published in the Journal of Information Security and Applications in 2019. Their model utilized convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to analyze URL and website content for phishing indicators. Experimental evaluation demonstrated the superior performance of the hybrid approach compared to individual models, highlighting the effectiveness of deep learning for phishing detection.

S. Yadav et al.,[6] proposed a phishing detection system using a hybrid machine learning approach in their paper "A hybrid machine learning approach for phishing detection using URL-based features," published in the Journal of Network and Computer Applications in 2019. Their system combined features extracted from URLs with various machine learning algorithms, including decision trees, random forests, and gradient boosting classifiers. Experimental results showed that their hybrid approach achieved high accuracy and robustness in detecting phishing websites.

V. Goyal et al.,[7] introduced a phishing detection system using a hybrid machine learning framework in their paper "Phishing Detection using Hybrid Machine Learning Approach," published in the International Journal of Advanced Research in Computer Science in 2020. Their approach combined features extracted from URLs, web page content, and network traffic with machine learning algorithms including naive Bayes, k-nearest neighbors, and support vector machines. Experimental results showcased the system's ability to effectively detect phishing attacks with high accuracy and efficiency.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Hybrid PhishNet combines rule-based techniques with machine learning algorithms to detect phishing based on URL features. It extracts features like domain reputation, URL length, and presence of suspicious keywords. These features are then fed into machine learning classifiers such as decision trees and random forests. By leveraging both rule-based and machine learning approaches, Hybrid PhishNet achieves high accuracy in phishing detection while minimizing false positives. Mention disadvantages of existing system

Phishing employs a multi-layered approach, utilizing lexical, host-based, and content-based features extracted from URLs. These features are then fed into machine learning models such as logistic regression and decision trees for classification. The system's hybrid architecture enhances its ability to discern between legitimate and phishing URLs effectively. By combining diverse features and machine learning algorithms, PhishAri achieves robust detection performance across various types of phishing attacks. Its comprehensive feature set and adaptable learning mechanisms contribute to its success in mitigating the threat of phishing.

3.2 Proposed System

The proposed system for Phishing Detection through Hybrid Machine Learning Based on URL incorporates a multi-stage approach for comprehensive detection. Initially, URL features such as domain reputation, URL length, and presence of suspicious keywords are extracted. These features serve as inputs to a hybrid ensemble of machine learning classifiers including decision trees, random forests, and gradient boosting. Through this ensemble, the system capitalizes on the diverse strengths of each classifier to enhance detection accuracy and resilience to evasion techniques.

Furthermore, the system employs dynamic feature selection mechanisms to adaptively refine the feature set based on evolving phishing tactics. Additionally, it integrates real-time data streams to continually update its knowledge base and adapt to emerging threats. The hybrid architecture ensures robust performance across diverse phishing scenarios while minimizing false positives.

Moreover, the system incorporates user feedback mechanisms to continuously improve its detection capabilities and mitigate false alarms. By leveraging both static URL features and dynamic behavioral indicators, the proposed system provides a comprehensive defense against phishing attacks, safeguarding users' sensitive information and online security. Mention advantages of Proposed system

3.3 Feasibility Study

Feasibility study for a Phishing Detection System through Hybrid Machine Learning Based on URL involves a thorough assessment of various aspects to ascertain its viability. From a technical perspective, the study would focus on evaluating the availability of suitable datasets for training the machine learning models and the computational resources required for feature extraction and real-time detection. Additionally, it would delve into the system's scalability to handle large volumes of URL data and adapt to future growth.

Financially, the study would estimate the costs associated with acquiring datasets, tools, and computational resources, juxtaposed with potential benefits like reduced security breaches. Operationally, the study would examine how easily the system can integrate with existing security infrastructure, its usability for security analysts, and the feasibility of implementing updates. By comprehensively analyzing these factors, the feasibility study would provide insights into the practicality and potential success of the proposed Phishing Detection System.

3.3.1 Economic Feasibility

The economic feasibility of a Phishing Detection System through Hybrid Machine Learning Based on URL involves assessing the costs of acquiring datasets, computational resources, and development tools against potential savings from mitigating security breaches. It requires estimating the return on investment (ROI) by considering the system's ability to reduce financial losses associated with phishing attacks. Additionally, economic feasibility entails analyzing the scalability of the system to accommodate future growth without substantial increases in operational costs. By evaluating these factors, stakeholders can determine whether the benefits of implementing the system outweigh the associated costs, ensuring its economic viability.

3.3.2 Technical Feasibility

The technical feasibility of a Phishing Detection System through Hybrid Machine Learning Based on URL involves assessing the availability of labeled datasets, computational resources for model training, and tools for feature extraction. It requires evaluating the scalability of the system to handle large volumes of URL data and adapt to evolving phishing tactics. Additionally, technical feasibility entails ensuring compatibility with existing security infrastructure and the usability of the system for security analysts. By analyzing these factors, stakeholders can determine whether the necessary technical resources are accessible and sufficient for developing and deploying the system.

3.3.3 Social Feasibility

The social feasibility of implementing a Phishing Detection System through Hybrid Machine Learning Based on URL hinges on several key factors. Firstly, ensuring user acceptance and trust is paramount. Users need to feel confident in the system's ability to effectively protect their online security without compromising their privacy. This entails transparent communication about how the system operates, its purpose, and the measures in place to safeguard user data. Moreover, education and awareness initiatives are essential to inform users about the system's benefits and limitations, thereby mitigating skepticism and resistance to adoption.

Establishing feedback mechanisms enables users to voice concerns and provide input, fostering a collaborative approach to system development and refinement. Finally, collaboration with stakeholders such as security experts, regulatory bodies, and community representatives is vital to address social and ethical considerations, ensuring that the system aligns with societal norms and values. By prioritizing user trust, transparency, and collaboration, the Phishing Detection System can navigate social dynamics effectively and garner support for its implementation.

3.4 System Specification

3.4.1 Hardware Specification

Processor : Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
RAM : 8GB DRD4 or higher
Hard Disk : 128 GB
Key Board : Standard Windows Keyboard
Mouse : No Mouse
Monitor : Any

3.4.2 Software Specification

Operating System : Windows 10
Server-side Script : Python 3.6
IDE : PyCharm
Libraries Used : Flask,pandas,sklearn,Gradient boost classifier

3.4.3 Standards and Policies

PyCharm

Pycharm is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.Pycharm has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

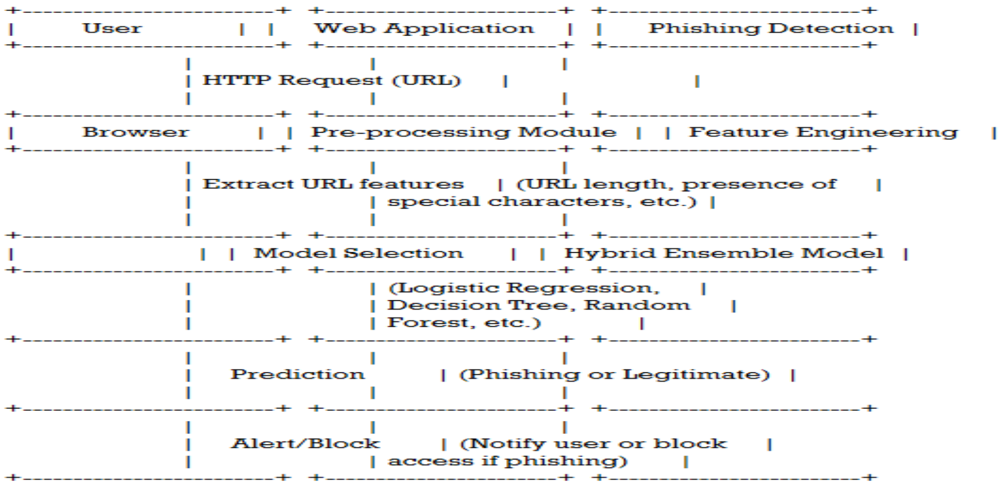


Figure 4.1: **Architecture diagram for Phishing Detection**

Description

Fig 4.1 shows the architecture of a Phishing Detection System through Hybrid Machine Learning Based on URL comprises interconnected components to streamline the detection process. Firstly, the User Interface (UI) serves as the front-end for user interaction, enabling users to submit URLs and view detection results. The Web Server acts as the middleware, managing HTTP requests between the UI and the backend components. Backend services encompass the core functionality, including URL feature extraction, hybrid machine learning model training and classification, and result generation. Additionally, a Machine Learning Model Repository stores trained models for reuse, and Data Storage manages datasets and extracted features. These components work collaboratively to ensure efficient and accurate phishing detection while accommodating system scalability and adaptability to emerging threats.

4.2 Design Phase

4.2.1 Data Flow Diagram

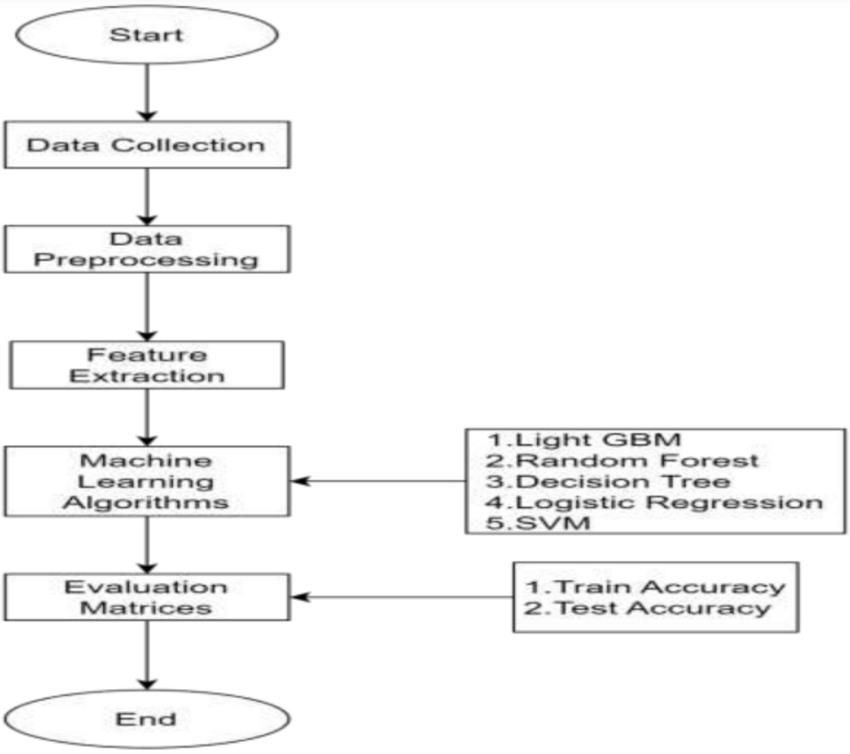


Figure 4.2: **Data Flow Diagram for Phishing Detection**

Description

Fig 4.2 shows phishing detection system through hybrid machine learning based on URLs operates through a structured flow of data and processes, as illustrated in its Data Flow Diagram (DFD). At its core are external entities, primarily users and the URLs submitted for analysis. Upon submission, these URLs undergo a series of processes. First, they are preprocessed to ensure uniformity and readiness for feature extraction. Next, relevant features are extracted from the URLs, such as domain age and length. These features serve as inputs to the machine learning classification stage, where sophisticated algorithms analyze the URLs to determine their phishing likelihood. To ensure the reliability of these classifications, a threshold check may be applied. Additionally, the system incorporates a feedback loop, where user input on misclassifications can be utilized to refine and improve the machine learning models over time. Supporting these processes are various data stores, including repositories of training data, model parameters, and feedback data.

4.2.2 Use Case Diagram

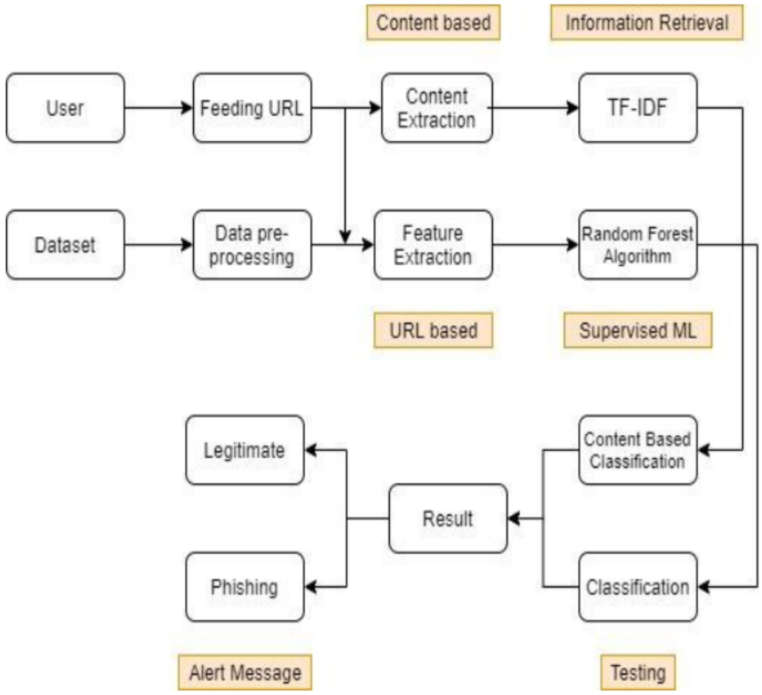


Figure 4.3: **Use Case Diagram for Phishing Detection**

Description

Fig 4.3 describes the Use Case Diagram for the phishing detection system through hybrid machine learning based on URLs delineates the interactions between the system's actors and its core functionalities. At the forefront is the User, who initiates the process by Submitting a URL for Analysis. This action triggers the system's analysis procedures, leveraging hybrid machine learning techniques to assess the URL's phishing potential. Once analyzed, the system Provides the Phishing Detection Result back to the user, indicating whether the URL is deemed legitimate or potentially malicious. Additionally, the diagram illustrates the provision for user engagement through the Provide Feedback use case. This enables users to offer input on classification results, contributing to the system's continuous enhancement. Implicit within these interactions is the sophisticated machine learning component, orchestrating the analysis process and informing the detection outcomes. Together, these use cases encapsulate the user-system engagements and the underlying mechanisms driving the phishing detection system's functionality.

4.2.3 Class Diagram

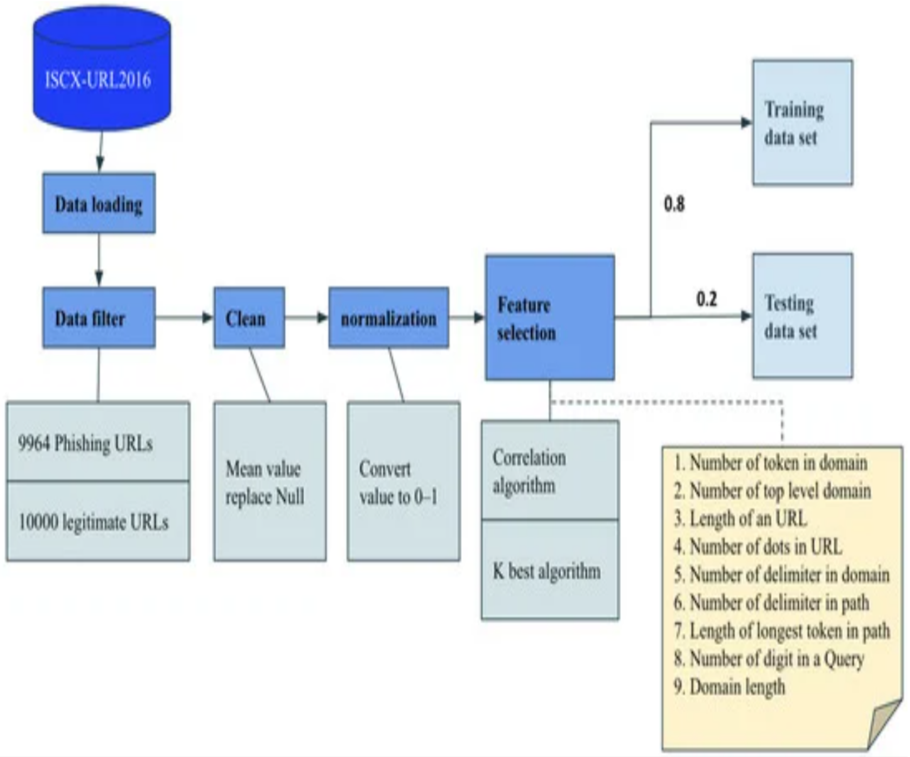


Figure 4.4: **Class Diagram for Phishing Detection**

Description

Fig 4.4 describes the class diagram for the phishing detection system through hybrid machine learning based on URLs presents the system's architecture and relationships among its core components. At its foundation are classes representing distinct modules: UserInterface, URLAnalyzer, MachineLearningModel, and FeedbackProcessor. The UserInterface class handles interactions with users, providing methods for displaying results and receiving URLs. Interacting with UserInterface, the URLAnalyzer class employs methods like AnalyzeURL() to process submitted URLs for phishing indicators. URLAnalyzer depends on the MachineLearningModel class, which encapsulates the hybrid machine learning model's functionality, including training and classification methods. This relationship underscores the reliance of URLAnalyzer on machine learning techniques for accurate detection. Additionally, the FeedbackProcessor class enables the incorporation of user feedback into model refinement through methods like ProcessFeedback().

4.2.4 Sequence Diagram

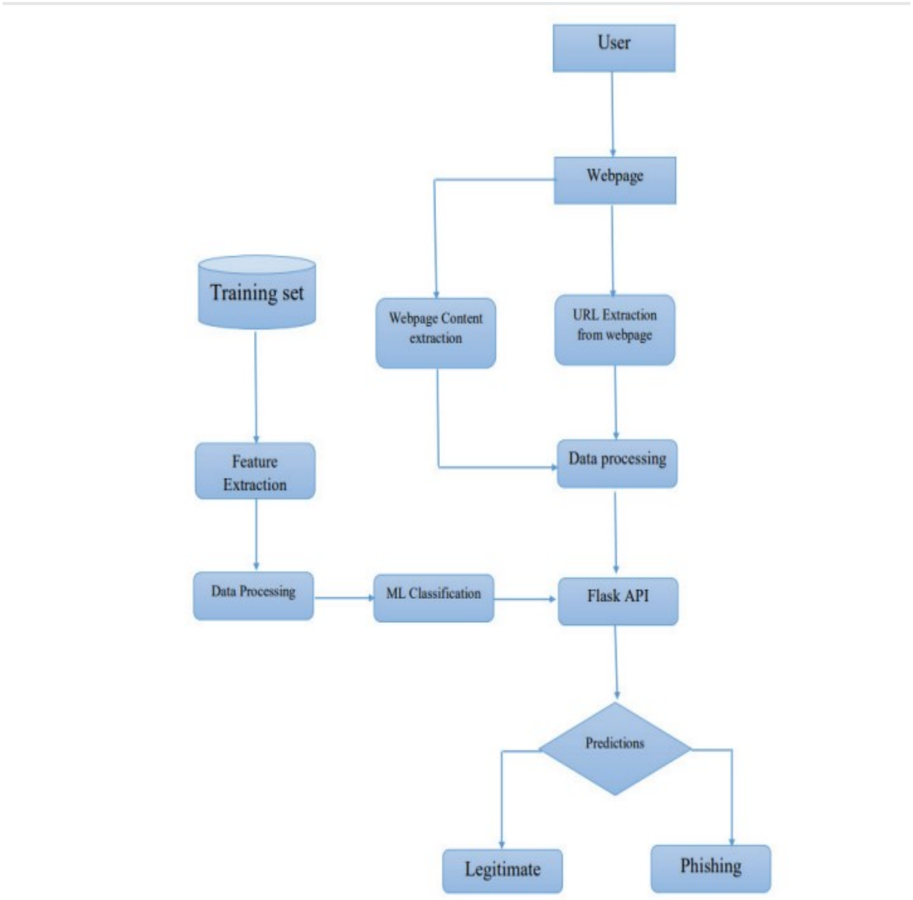


Figure 4.5: **Sequence Diagram for Phishing Detection**

Description

Fig 4.5 shows the sequence diagram for the phishing detection system through hybrid machine learning based on URLs depicts the chronological flow of interactions between its components during the URL analysis process. Initially, the UserInterface class receives a request from the user to analyze a URL. Upon receiving the URL, the UserInterface class invokes the AnalyzeURL() method of the URLAnalyzer class. The URLAnalyzer class then initiates the URL analysis process, which involves utilizing the MachineLearningModel class to classify the URL as either phishing or legitimate. This classification task entails invoking the ClassifyURL() method of the MachineLearningModel class. Depending on the classification outcome, the MachineLearningModel class returns the result to the URLAnalyzer class. Subsequently, the URLAnalyzer class forwards the result to the UserInterface class, which displays it to the user through the DisplayResult() method. Optionally, if the user provides feedback on the classification result, the UserInterface class triggers the ProcessFeedback() method of the FeedbackProcessor class, facilitating the incorporation of user feedback into model refinement.

4.2.5 Collaboration diagram

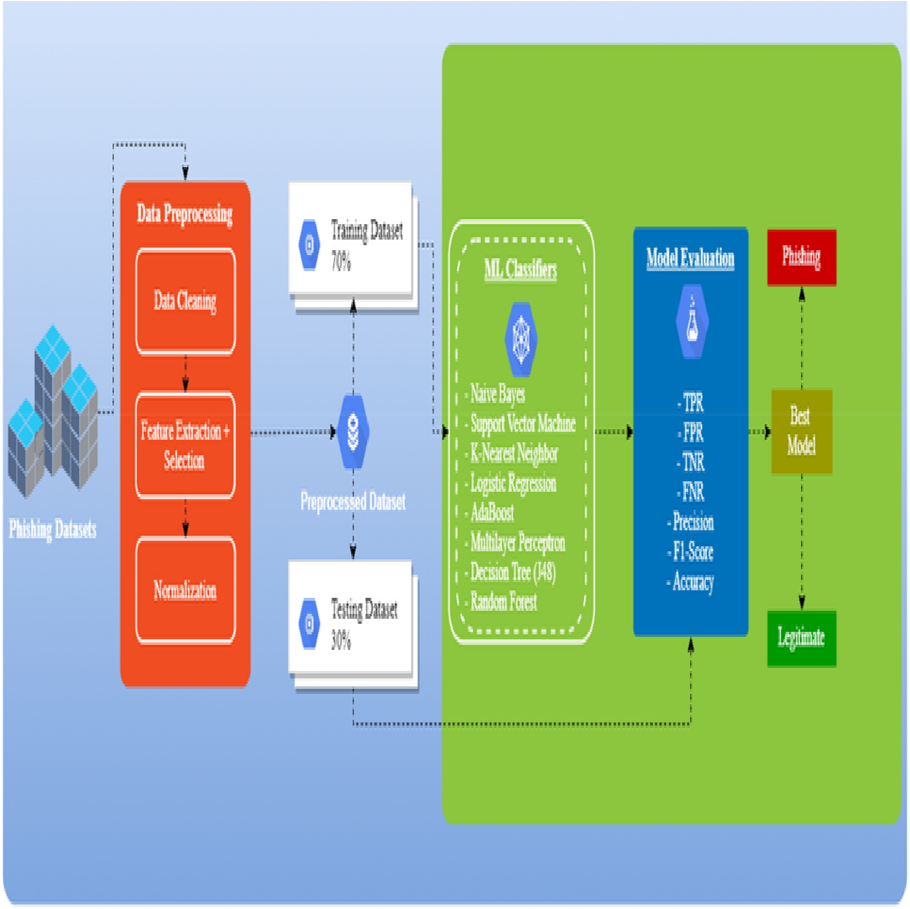


Figure 4.6: **Collaboration Diagram for Phishing Detection**

Description

Fig 4.6 describes the collaboration diagram for the phishing detection system through hybrid machine learning based on URLs illustrates how the system's components collaborate to analyze URLs and classify them as phishing or legitimate. It highlights the interactions and message exchanges among the system's classes. Initially, the UserInterface class collaborates with the URLAnalyzer class by sending a message to initiate URL analysis. This message triggers the URLAnalyzer class to collaborate with the MachineLearningModel class, requesting it to classify the URL. The MachineLearningModel class then collaborates with the URLAnalyzer class by sending the classification result back. Simultaneously, the URLAnalyzer class collaborates with the UserInterface class to deliver the classification result to the user.

4.2.6 Activity Diagram

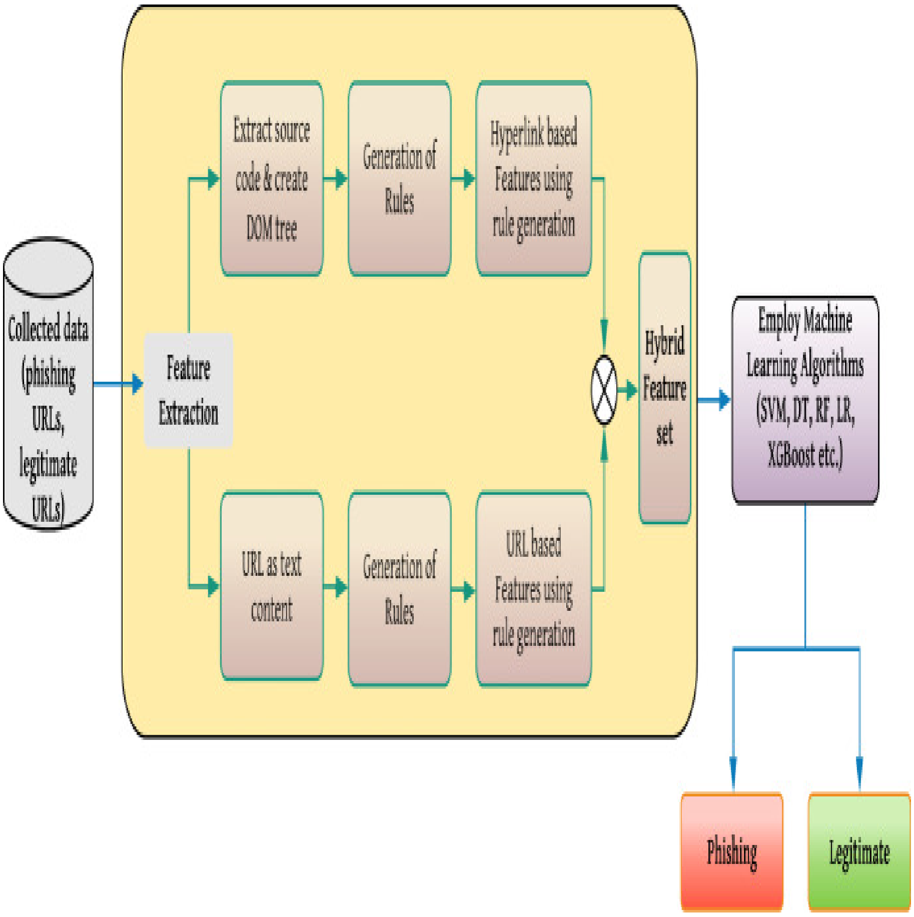

Collected data (phishing URLs, legitimate URLs)
Feature Extraction
Extract source code & create DOM tree
Generation of Rules
Hyperlink based Features using rule generation
URL as text content
Generation of Rules
URL based Features using rule generation
Hybrid Feature set
Employ Machine Learning Algorithms (SVM, DT, RF, LR, XGBoost etc.)
Phishing
Legitimate

Figure 4.7: **Activity Diagram for Phishing Detection**

Description

Fig 4.7 shows the activity diagram for the phishing detection system through hybrid machine learning based on URLs delineates the sequential flow of activities involved in analyzing URLs and determining their phishing likelihood. It begins with the initiation of the process, where the system waits for the user to submit a URL for analysis. Once a URL is received, the system proceeds to preprocess it, which involves steps such as parsing and feature extraction. Following preprocessing, the system invokes the machine learning classification process, where the URL's features are fed into the hybrid machine learning model for classification. The system then evaluates the classification result to determine if the URL is classified as phishing or legitimate. If the classification result meets the system's confidence threshold, the process concludes with the system providing the classification result to the user. However, if the confidence threshold is not met or if the user opts to provide feedback, the system enters a feedback loop.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

A high-level algorithm for Phishing detection using machine learning:

1.Data Collection: Collect historical data on phishing websites. The data should cover a period of several years and include both phishing and legitimate urls.

2.Data Preprocessing: Clean the data by removing duplicates, filling in missing values, and normalizing the data. Divide the data into training and testing sets.

3.Feature Selection: Select the most relevant features that are likely to influence the prediction of urls.

4.Model Selection: Choose a suitable machine learning algorithm such as Random Forest, SVM, ANN, Gradient Boosting, or LSTM. Train the model on the training set using the selected features.

5.Model Evaluation: Evaluate the performance of the model on the testing set using metrics such as accuracy, precision, recall, and F1-score.

6.Prediction: Use the trained model to make predictions on new data. The model should be able to predict the websites based on the input urls .

7.Deployment: Integrate the model into a system that can receive real-time data from phishtank.

4.3.2 Pseudo Code

Here's a pseudocode for machine learning based phishing website detection

```
function main():
    # Load and preprocess dataset
    dataset = load_dataset()
    preprocessed_data = preprocess_dataset(dataset)

    # Split dataset into training and testing sets
    train_data, test_data = split_dataset(preprocessed_data)

    # Extract features from URLs
    train_features = extract_features(train_data)
    test_features = extract_features(test_data)

    # Train hybrid machine learning model
    model = train_model(train_features, train_labels)

    # Evaluate model performance on test set
    accuracy = evaluate_model(model, test_features, test_labels)
    print("Accuracy:", accuracy)

function load_dataset():
    # Load dataset from file or database
    dataset = load_data_from_source()
    return dataset

function preprocess_dataset(dataset):
    # Perform preprocessing steps (e.g., removing duplicates, handling missing values)
    preprocessed_data = preprocess_data(dataset)
    return preprocessed_data

function split_dataset(data):
    # Split data into training and testing sets
    train_data, test_data = split_data(data)
    return train_data, test_data

function extract_features(data):
    # Extract features from URLs (e.g., domain reputation, URL length, presence of suspicious
        keywords)
    features = extract_features_from_urls(data)
    return features

function train_model(features, labels):
    # Train hybrid machine learning model (e.g., ensemble of classifiers)
    model = train_hybrid_model(features, labels)
    return model

function evaluate_model(model, test_features, test_labels):
    # Evaluate model performance on test set
    predictions = model.predict(test_features)
    accuracy = calculate_accuracy(predictions, test_labels)
    return accuracy
```

4.4 Module Description

4.4.1 Module1:Pre-processing

Data Preprocessing: Clean the data by removing duplicates, filling in missing values, and normalizing the data. Divide the data into training and testing sets, ensuring that both sets have a similar distribution of the target variable .The goal of preprocessing is to transform the raw data into a format that can be used by machine learning algorithms to make accurate predictions about potential website detection events.

4.4.2 Module2:Feature Extraction

Feature extraction is a critical step in machine learning-based website detection as it involves selecting and transforming relevant input variables into a more informative representation that can be used by the model to make accurate predictions. Feature extraction techniques that can be used in website prediction include wavelet analysis, Fourier transforms, and statistical techniques such as auto correlation and crosscorrelation analysis

4.4.3 Module3:Model Training

Model training for url prediction is a complex process that involves gathering and cleaning data, choosing a machine learning algorithm, and training the algorithm to predict urls. The goal of this process is to develop a model that can accurately predict the urls. Among them, statistical models of random forest, linear regression, and gradient boost classifier integrated moving average are the most common url frequency analysis methods for modeling urls prediction.

4.5 Steps to execute/run/implement the project

4.5.1 Step1:Uploading Dataset

[1]Select the required dataset which contains phishing urls data.
[2]The dataset should contain data about various phishing and legitimate Urls.
[3]Based on the analysis of Urls data,upload dataset .

4.5.2 Step2:Model training

[1]After uploading dataset,try to select a model to train the dataset.
[2]Training refers to adding few features in order to perform prediction.
[3]Select a required model like gradient boost,in order to train the dataset.

4.5.3 Step3:Prediction

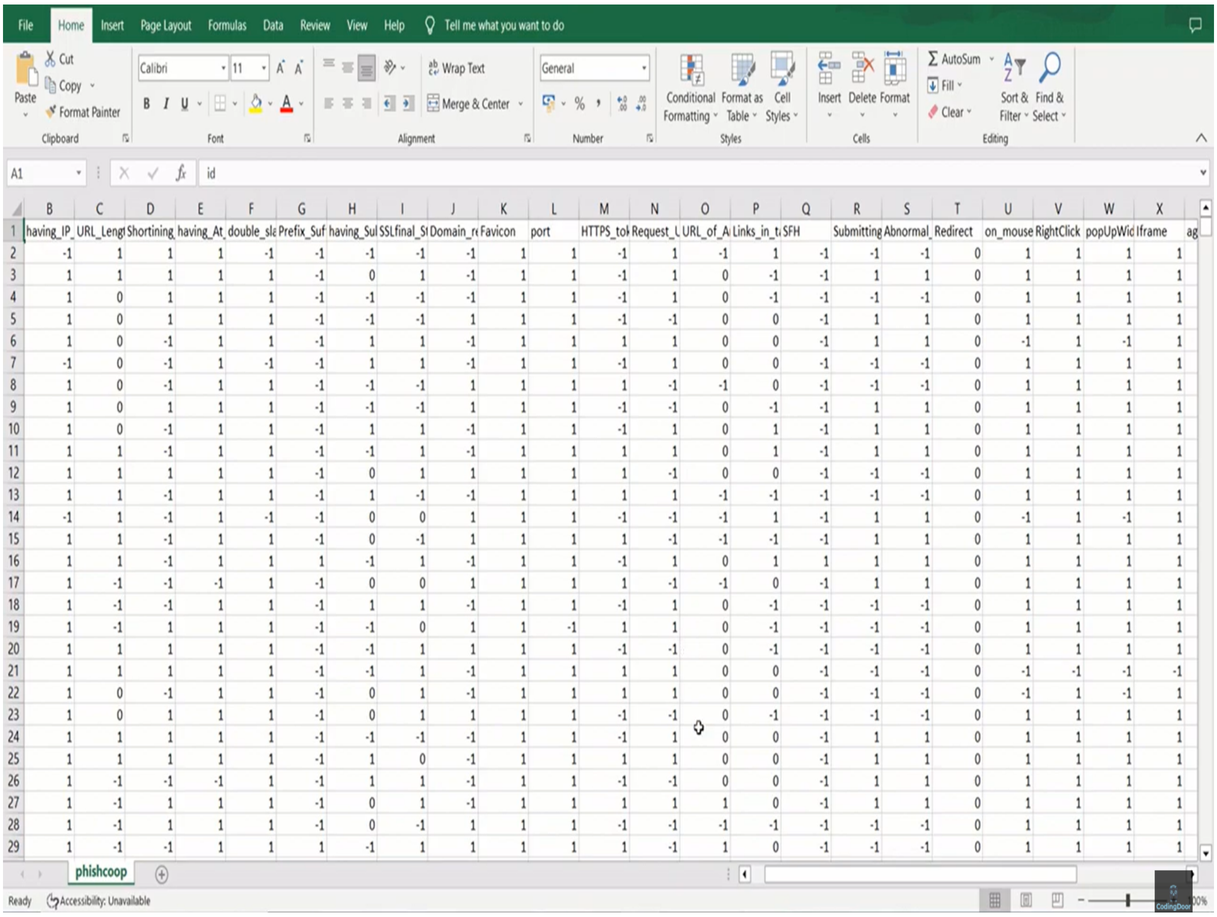
[1]After training the dataset using different models,try to analyze the dataset.
[2] Predict the Urls in every domain for required urls.
[3]Now predict the Url weather it is phishing or legitimate.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

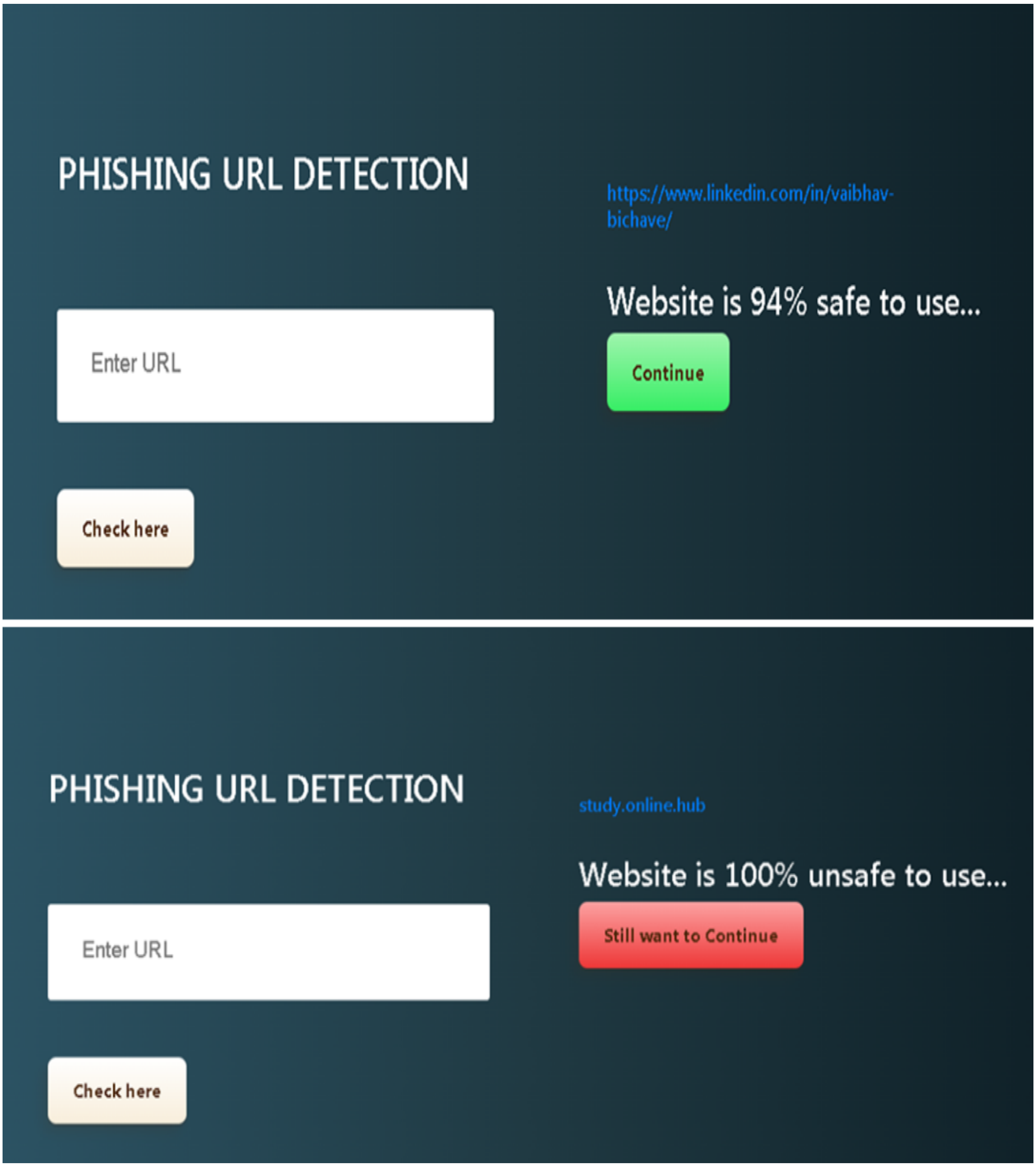


	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1	having_IP_	URL_Lengt	Shortining	having_At	double_sla	Prefix_Suf	having_Sul	SSLfinal_St	Domain_re	Favicon	port	HTTPS_tok	Request_U	URL_of_A	Links_in_ta	SFH	Submitting	Abnormal_	Redirect	on_mouse	RightClick	popUpWid	Iframe	ag
2	-1	1	1	1	-1	-1	-1	-1	-1	1	1	-1	1	-1	1	-1	-1	-1	0	1	1	1	1	
3	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1	1	1	0	1	1	1	1	
4	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	1	
5	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	1	
6	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1	1	1	0	-1	1	-1	1	
7	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1	0	0	-1	-1	-1	0	1	1	1	1	
8	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	0	-1	-1	-1	0	1	1	1	1	
9	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	-1	-1	1	1	0	1	1	1	1	
10	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1	-1	1	1	0	1	1	1	1	
11	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1	1	1	0	1	1	1	1	
12	1	1	1	1	1	-1	0	1	1	1	1	1	-1	0	0	-1	-1	-1	0	1	1	1	1	
13	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	0	1	1	1	1	
14	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1	-1	1	-1	1	1	0	-1	1	-1	1	
15	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1	-1	-1	-1	1	1	0	1	1	1	1	
16	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1	0	1	1	1	1	0	1	1	1	1	
17	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1	-1	0	-1	1	1	0	1	1	1	1	
18	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	-1	-1	-1	-1	0	1	1	1	1	
19	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1	0	-1	-1	-1	-1	0	1	1	1	1	
20	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	1	
21	1	1	1	1	1	-1	-1	1	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	-1	-1	-1	
22	1	0	-1	1	1	-1	0	1	-1	1	1	1	1	0	0	-1	-1	-1	0	-1	1	-1	1	
23	1	0	1	1	1	-1	0	1	1	1	1	-1	-1	0	-1	-1	-1	-1	0	1	1	1	1	
24	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	0	-1	1	1	0	1	1	1	1	
25	1	1	1	1	1	-1	1	0	-1	1	1	1	1	0	0	-1	1	1	0	1	1	1	1	
26	1	-1	-1	-1	1	-1	1	1	-1	1	1	-1	-1	0	0	-1	1	1	0	1	1	1	1	
27	1	-1	1	1	1	-1	0	1	-1	1	1	1	1	1	0	-1	1	1	0	1	1	1	1	
28	1	-1	1	1	1	-1	0	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	0	1	1	1	1	
29	1	-1	-1	1	1	1	-1	1	1	1	1	1	-1	1	0	-1	-1	-1	0	1	1	1	1	

Figure 5.1: **Input Dataset**

Fig 5.1 shows the input dataset of the phishing data set of the various websites .Based on the data set,the websites will be predicted and based on the website the prediction is done.The above input data set contains information about various websites .The above diagram contains information about phishing websites.At last, it shows that the website is phishing or legitimate.

5.1.2 Output Design



https://www.linkedin.com/in/vaibhav-bichave/

Website is 94% safe to use...

study.online.hub

Website is 100% unsafe to use...

Figure 5.2: **Output of Phishing Detection**

Fig 5.2 shows the output design of Phishing website prediction that is firstly we should enter website Url to predict weather the website is phishing or legitimate. For predicting website we will first do model training and then check whether website is phishing or legitimate.If it is legitimate or phishing it is displayed with the accuracy.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Input

```
The Phishing Detection System is divided into units: URL feature extraction, ML model training,
    classification, and integration. Each unit undergoes rigorous testing, covering accuracy,
    performance, and integration aspects. Dependencies are mocked to isolate units, while automation
     ensures frequent testing and integration with CI/CD pipelines. Comprehensive testing includes
    performance evaluation, error handling validation, and security checks for robustness.
```

5.3.2 Integration testing

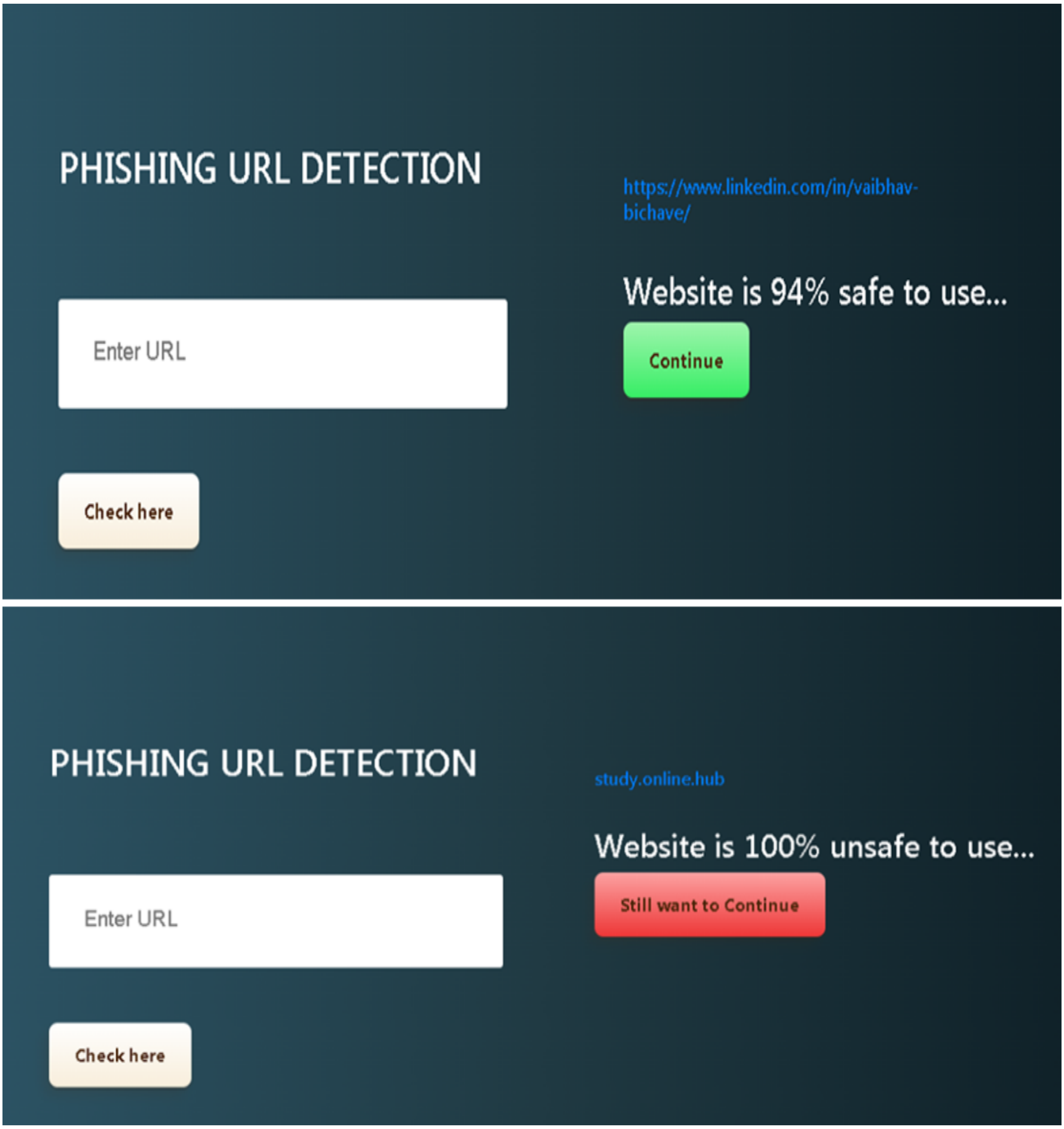
Input

```
Integration testing of a Phishing Detection System using Hybrid Machine Learning for URL analysis
    involves assessing how its components work together. This includes testing data flow between URL
     feature extraction, ML model training, classification, and system integration.
```

5.3.3 System testing

Input

```
System testing of a Phishing Detection System utilizing Hybrid Machine Learning for URL analysis
    involves examining the internal structures and logic of the systems components. This includes
    scrutinizing algorithms for URL feature extraction, ML model training procedures, classification
     methods, and integration mechanisms.
```



https://www.linkedin.com/in/vaibhav-bichave/

Website is 94% safe to use...

study.online.hub

Website is 100% unsafe to use...

Figure 5.3: **Phishing prediction testing**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed Phishing Detection System through Hybrid Machine Learning Based on URL, particularly when employing the Gradient Boost algorithm, is notable for several reasons. Firstly, Gradient Boosting is well-suited for handling imbalanced datasets commonly encountered in phishing detection, where the number of legitimate URLs far exceeds phishing URLs. By iteratively improving the model's performance, Gradient Boosting effectively balances the trade-off between precision and recall, resulting in accurate detection while minimizing false positives.

Secondly, the Gradient Boost algorithm excels in capturing complex relationships and interactions among URL features, enabling the system to discern subtle patterns indicative of phishing attempts. Through ensemble learning, where multiple weak learners are combined to form a strong classifier, Gradient Boosting enhances the system's ability to generalize well to unseen data, thereby improving overall detection performance.

Additionally, the efficiency of Gradient Boosting lies in its scalability and computational efficiency. While Gradient Boosting involves sequential training of individual decision trees, optimizations such as parallel processing and tree pruning techniques mitigate computational overhead, ensuring efficient utilization of resources even with large datasets.

6.2 Comparison of Existing and Proposed System

Existing system:(Decision tree)

In the Existing system, we implemented a decision tree algorithm that predicts whether to grant the loan or not. When using a decision tree model, it gives the training dataset the accuracy keeps improving with splits. We can easily overfit the dataset and doesn't know when it crossed the line unless we are using the cross validation.

The advantages of the decision tree are model is very easy to interpret we can know that the variables and the value of the variable is used to split the data. But the accuracy of decision tree in existing system gives less accurate output that is less when compared to proposed system.

Proposed system:(XGBoost Classifier)

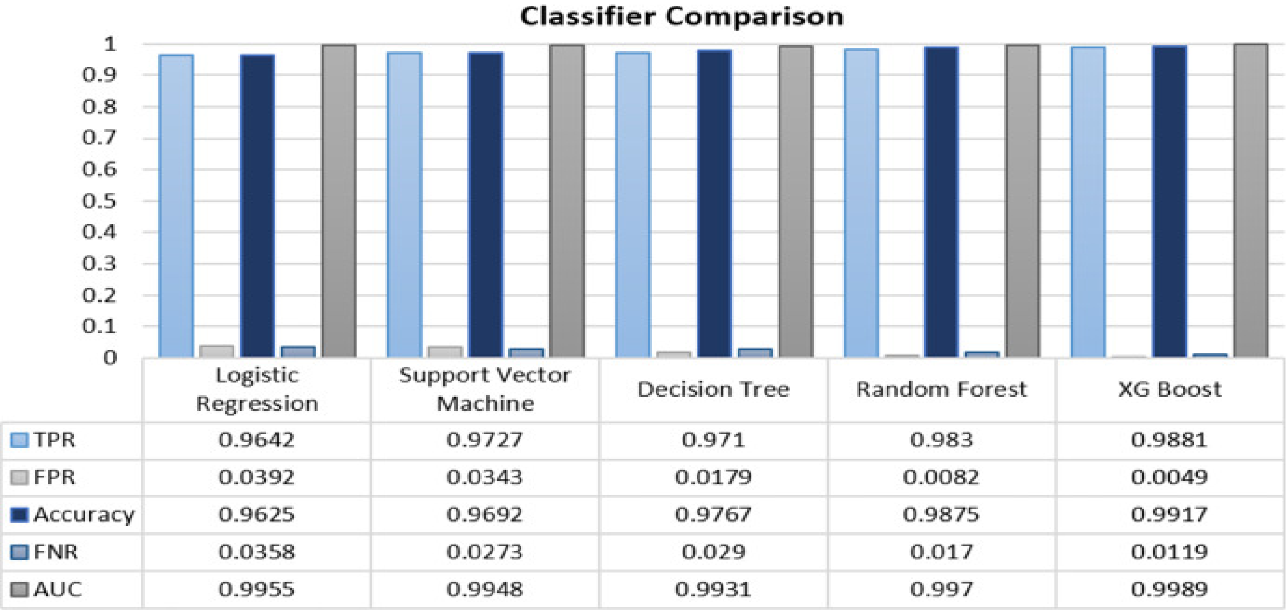
XGBoost algorithm generates more trees when compared to the decision tree and other algorithms. We can specify the number of trees we want in the forest and also we also can specify maximum of features to be used in the each of the tree. But, we cannot control the randomness of the forest in which the feature is a part of the algorithm. Accuracy keeps increasing as we increase the number of trees but it becomes static at one certain point. Unlike the decision tree it won’t create more biased and decreases variance. Proposed system is implemented using the XGBoost algorithm so that the accuracy is more when compared to the existing system.

6.3 Sample Code

```
from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings(’ignore’)
from feature import FeatureExtraction

file = open(”pickle/cat.pkl”,”rb”)
gbc = pickle.load(file)
file.close()


app = Flask(__name__)

@app.route(”/”, methods=[”GET”, ”POST”])
def index():
    if request.method == ”POST”:

        url = request.form[”url”]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)

        y_pred =gbc.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = ”It is {0:.2f} % safe to go ”.format(y_pro_phishing*100)
        return render_template(’index.html’,xx =round(y_pro_non_phishing,2),url=url )
    return render_template(”index.html”, xx =-1)


if __name__ == ”__main__”:
    app.run(debug=True)

    Index.Html

    <!DOCTYPE html>
<html lang=”en”>
<head>
    <meta charset=”UTF-8”>
    <meta http-equiv=”X-UA-Compatible” content=”IE=edge”>
    <meta name=”viewport” content=”width=device-width, initial-scale=1.0”>
    <meta name=”description” content=”This website is develop for identify the safety of url.”>
    <meta name=”keywords” content=”phishing url,phishing,cyber security,machine learning,classifier,
        python”>
    <meta name=”author” content=”VAIBHAV BICHAVE”>

    <!-- BootStrap -->
    <link rel=”stylesheet” href=”https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.
        min.css”
        integrity=”sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk”
            crossorigin=”anonymous”>

    <link href=”static/styles.css” rel=”stylesheet”>
    <title>URL detection</title>

</head>

<body>

<div class=” container”>
    <div class=”row”>
        <div class=”form col-md” id=”form1”>
            <h2>PHISHING URL DETECTION</h2>

            <br>
            <form action=”/” method =”post”>
                <input type=”text” class=”form__input” name =’url’ id=”url” placeholder=”Enter URL”
                    required=”” />
                <label for=”url” class=”form__label”>URL</label>
                <button class=”button” role=”button” >Check here</button>
            </form>

    </div>

    <div class=”col-md” id=”form2”>

        <br>
        <h6 class = ”right ”><a href= {{ url }} target=”_blank”>{{ url }}</a></h6>

        <br>
        <h3 id=”prediction”></h3>
        <button class=”button2” id=”button2” role=”button” onclick=”window.open(’{{url}}’)” target=”
            _blank” >Still want to Continue</button>
        <button class=”button1” id=”button1” role=”button”  onclick=”window.open(’{{url}}’)” target=
            ”_blank”>Continue</button>
    </div>
 </div>
<br>
<p> 2024  YASHWANTH KUMAR REDDY</p>
 </div>

    <!-- JavaScript -->
    <script src=”https://code.jquery.com/jquery-3.5.1.slim.min.js”
        integrity=”sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj”
        crossorigin=”anonymous”></script>
    <script src=”https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js”
        integrity=”sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo”
        crossorigin=”anonymous”></script>
    <script src=”https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js”
        integrity=”sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI”
        crossorigin=”anonymous”></script>



    <script>

            let x = ’{{xx}}’;
            let num = x*100;
            if (0<=x && x<0.50){
                num = 100-num;
            }
            let txtx = num.toString();
            if(x<=1 && x>=0.50){
                var label = ”Website is ”+txtx +”% safe to use...”;
                document.getElementById(”prediction”).innerHTML = label;
                document.getElementById(”button1”).style.display=”block”;
            }
            else if (0<=x && x<0.50){
                var label = ”Website is ”+txtx +”% unsafe to use...”
                document.getElementById(”prediction”).innerHTML = label ;
                document.getElementById(”button2”).style.display=”block”;
            }

```

```
      </script>

</body>

</html>
```

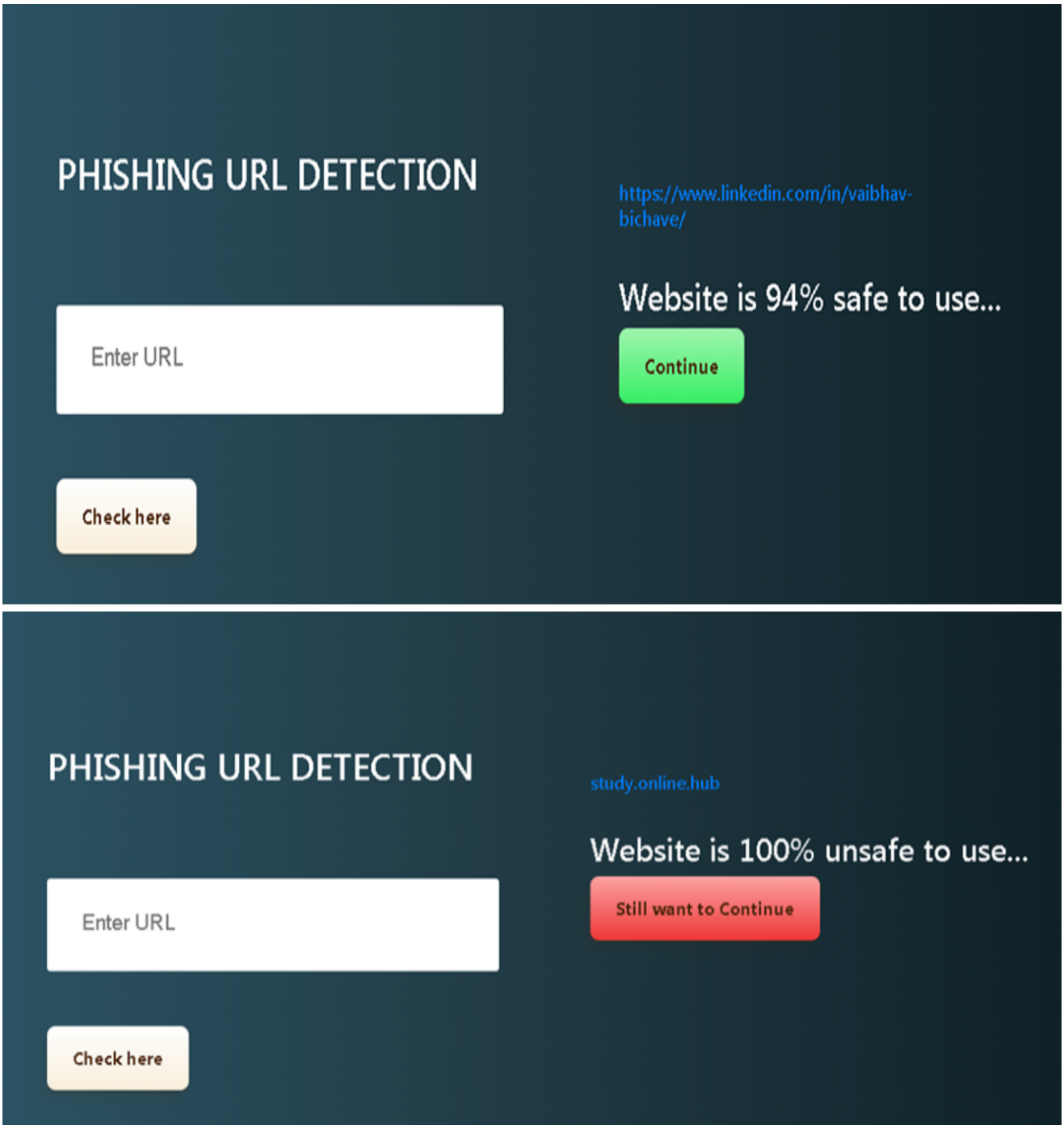
Output



	Logistic Regression	Support Vector Machine	Decision Tree	Random Forest	XG Boost
▨ TPR	0.9642	0.9727	0.971	0.983	0.9881
▨ FPR	0.0392	0.0343	0.0179	0.0082	0.0049
■ Accuracy	0.9625	0.9692	0.9767	0.9875	0.9917
▨ FNR	0.0358	0.0273	0.029	0.017	0.0119
▨ AUC	0.9955	0.9948	0.9931	0.997	0.9989

Figure 6.1: **Classifier Comparision of ML models**

Fig 6.1 shows the performance of each individual category of feature set and also measure the performance of the hybrid feature set by using the XG Boost classifier to categorize the website.The comparison of performance metrics for each and every category of feature set. URL-based features work well with a 98.42 accuracy rate and 97.79 true positive rate, as shown in the figure. Using only hyperlink-based features produces 84.67 accuracy and 96.93 true positive rate. We present the results of our proposed approach by combining all features (UF2-UF15 and HF1-HF10) to obtain the hybrid feature set that produces a high true positive rate (approximately 99) with a low false positive rate (less than 0.5). Both URL-based and hyperlink-based features are useful for individual web-based detection, but they are not sufficient to detect various types of phishing URLs. If a hybrid feature set is used to detect phishing URLs, we could be able to detect phishing attacks more accurately.



https://www.linkedin.com/in/vaibhav-bichave/

Website is 94% safe to use...

study.online.hub

Website is 100% unsafe to use...

Figure 6.2: **Output of Phishing Detection System**

Fig 6.2 shows the final output of the phishing detection system through hybrid machine learning based on URLs is a clear indication of whether a given URL is classified as phishing or legitimate. Upon submitting a URL for analysis, users receive a concise and comprehensible result detailing the classification outcome. This output serves as a crucial decision-making tool, empowering users to make informed choices about the URLs they encounter online.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Phishing is a cyber crime procedure utilizing both social building and specialized deception to take individual sensitive data. Besides, Phishing is considered as another extensive type of fraud. Experimentations against recent dependable phishing data sets utilizing different classification algorithm have been performed which received different learning methods. The base of the experiments is accuracy measure.

The aim of this research work is to predict whether a given URL is phishing website or not. It turns out in the given experiment that Random forest based classifiers are the best classifier with great classification accuracy of 82.644 for the given dataset of phishing site. As a future work we might use this model to other Phishing dataset with larger size then now and then testing the performance of those classification algorithm's in terms of classification accuracy

7.2 Future Enhancements

Phishing Detection System through Hybrid Machine Learning Based on URL could involve several key areas of advancement. Firstly, refining feature engineering algorithms to incorporate advanced techniques such as semantic analysis of webpage content and behavioral analysis of user interactions would improve the system's ability to detect subtle phishing indicators. Additionally, exploring state-of-the-art machine learning models, including deep learning architectures like convolutional neural networks and recurrent neural networks, could enhance classification accuracy and handle complex URL patterns more effectively. Real-time detection capabilities could be augmented to enable immediate response actions such as alerting users and blocking access to malicious URLs.

Chapter 8

INDUSTRY DETAILS

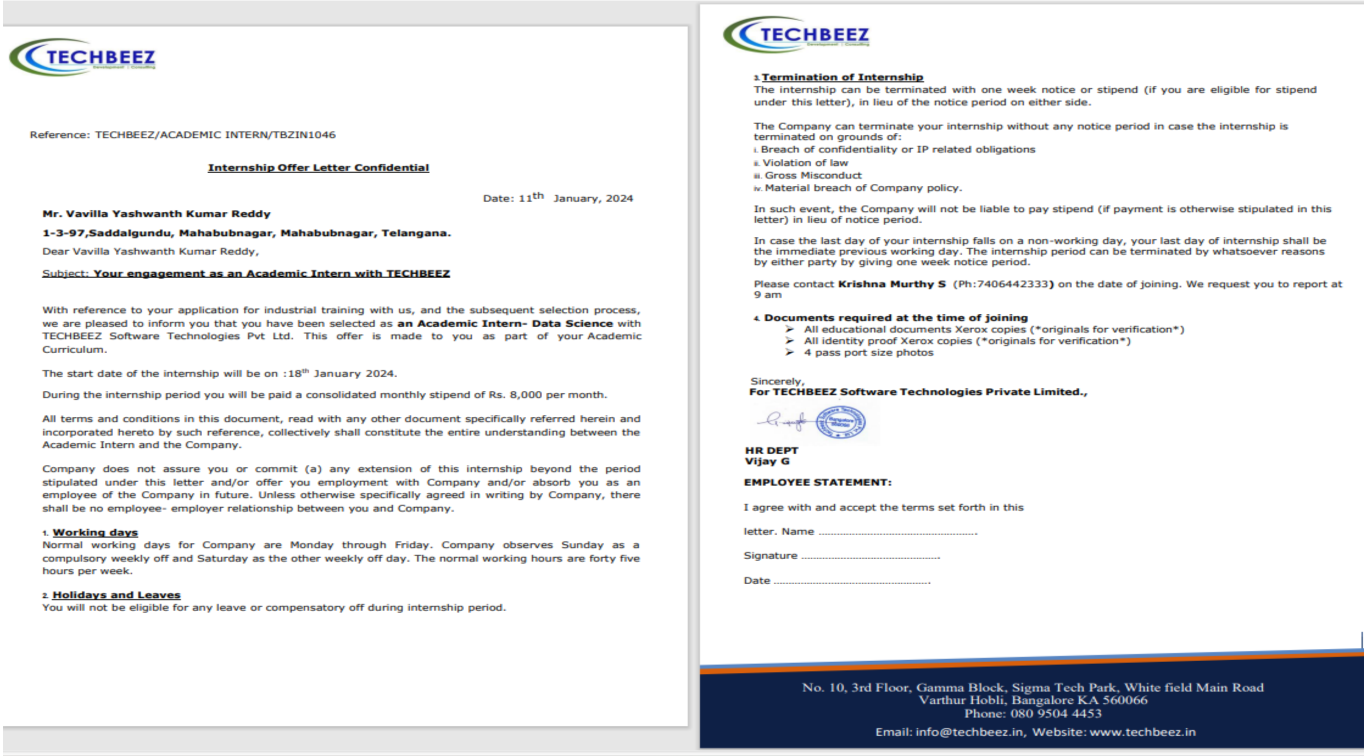
8.1 Industry name : Techbeez Software Technologies

8.1.1 Duration of Internship (18-01-2024 - 18-05-2024)

8.1.2 Duration of Internship :4 Months

8.1.3 Industry Address :No. 10, 3rd Floor, Gamma Block, Sigma Tech Park, White field Main Road,Varthur Hobli, Bangalore.

8.2 Internship offer letter



TECHBEEZ

Reference: TECHBEEZ/ACADEMIC INTERN/TBZIN1046

Internship Offer Letter Confidential

Date: 11th January, 2024

Mr. Vavilla Yashwanth Kumar Reddy

1-3-97,Saddalgundu, Mahabubnagar, Mahabubnagar, Telangana.

Dear Vavilla Yashwanth Kumar Reddy,

Subject: **Your engagement as an Academic Intern with TECHBEEZ**

With reference to your application for industrial training with us, and the subsequent selection process, we are pleased to inform you that you have been selected as **an Academic Intern- Data Science** with TECHBEEZ Software Technologies Pvt Ltd. This offer is made to you as part of your Academic Curriculum.

The start date of the internship will be on :18th January 2024.

During the internship period you will be paid a consolidated monthly stipend of Rs. 8,000 per month.

All terms and conditions in this document, read with any other document specifically referred herein and incorporated hereto by such reference, collectively shall constitute the entire understanding between the Academic Intern and the Company.

Company does not assure you or commit (a) any extension of this internship beyond the period stipulated under this letter and/or offer you employment with Company and/or absorb you as an employee of the Company in future. Unless otherwise specifically agreed in writing by Company, there shall be no employee- employer relationship between you and Company.

1. **Working days**

Normal working days for Company are Monday through Friday. Company observes Sunday as a compulsory weekly off and Saturday as the other weekly off day. The normal working hours are forty five hours per week.

2. **Holidays and Leaves**

You will not be eligible for any leave or compensatory off during internship period.

TECHBEEZ

3. **Termination of Internship**

The internship can be terminated with one week notice or stipend (if you are eligible for stipend under this letter), in lieu of the notice period on either side.

The Company can terminate your internship without any notice period in case the internship is terminated on grounds of:

i. Breach of confidentiality or IP related obligations

ii. Violation of law

iii. Gross Misconduct

iv. Material breach of Company policy.

In such event, the Company will not be liable to pay stipend (if payment is otherwise stipulated in this letter) in lieu of notice period.

In case the last day of your internship falls on a non-working day, your last day of internship shall be the immediate previous working day. The internship period can be terminated by whatsoever reasons by either party by giving one week notice period.

Please contact **Krishna Murthy S** (Ph:7406442333**)** on the date of joining. We request you to report at 9 am

4. **Documents required at the time of joining**
   - ➢ All educational documents Xerox copies (*originals for verification*)
   - ➢ All identity proof Xerox copies (*originals for verification*)
   - ➢ 4 pass port size photos

Sincerely,
For TECHBEEZ Software Technologies Private Limited.,

HR DEPT
Vijay G

EMPLOYEE STATEMENT:

I agree with and accept the terms set forth in this

letter. Name ....

Signature ....

Date ....

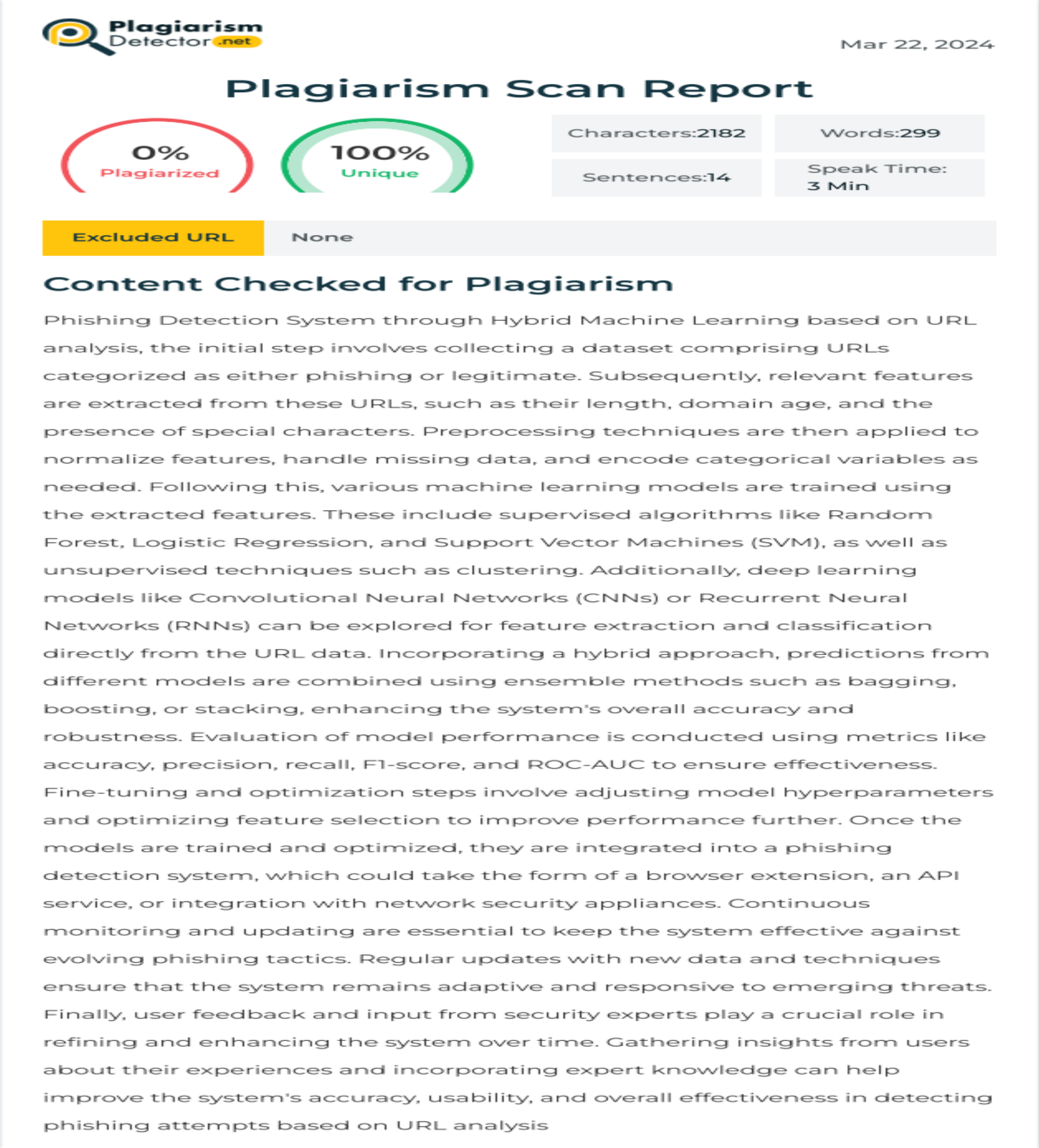
No. 10, 3rd Floor, Gamma Block, Sigma Tech Park, White field Main Road
Varthur Hobli, Bangalore KA 560066
Phone: 080 9504 4453
Email: info@techbeez.in, Website: www.techbeez.in

8.3 Internship Completion certificate

images/Untitled Diagram (15).jpg

Chapter 9

PLAGIARISM REPORT



Plagiarism Detector.net

Mar 22, 2024

Plagiarism Scan Report

0% Plagiarized

100% Unique

Characters:2182	Words:299
Sentences:14	Speak Time: 3 Min

Excluded URL	None

Content Checked for Plagiarism

Phishing Detection System through Hybrid Machine Learning based on URL analysis, the initial step involves collecting a dataset comprising URLs categorized as either phishing or legitimate. Subsequently, relevant features are extracted from these URLs, such as their length, domain age, and the presence of special characters. Preprocessing techniques are then applied to normalize features, handle missing data, and encode categorical variables as needed. Following this, various machine learning models are trained using the extracted features. These include supervised algorithms like Random Forest, Logistic Regression, and Support Vector Machines (SVM), as well as unsupervised techniques such as clustering. Additionally, deep learning models like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be explored for feature extraction and classification directly from the URL data. Incorporating a hybrid approach, predictions from different models are combined using ensemble methods such as bagging, boosting, or stacking, enhancing the system's overall accuracy and robustness. Evaluation of model performance is conducted using metrics like accuracy, precision, recall, F1-score, and ROC-AUC to ensure effectiveness. Fine-tuning and optimization steps involve adjusting model hyperparameters and optimizing feature selection to improve performance further. Once the models are trained and optimized, they are integrated into a phishing detection system, which could take the form of a browser extension, an API service, or integration with network security appliances. Continuous monitoring and updating are essential to keep the system effective against evolving phishing tactics. Regular updates with new data and techniques ensure that the system remains adaptive and responsive to emerging threats. Finally, user feedback and input from security experts play a crucial role in refining and enhancing the system over time. Gathering insights from users about their experiences and incorporating expert knowledge can help improve the system's accuracy, usability, and overall effectiveness in detecting phishing attempts based on URL analysis

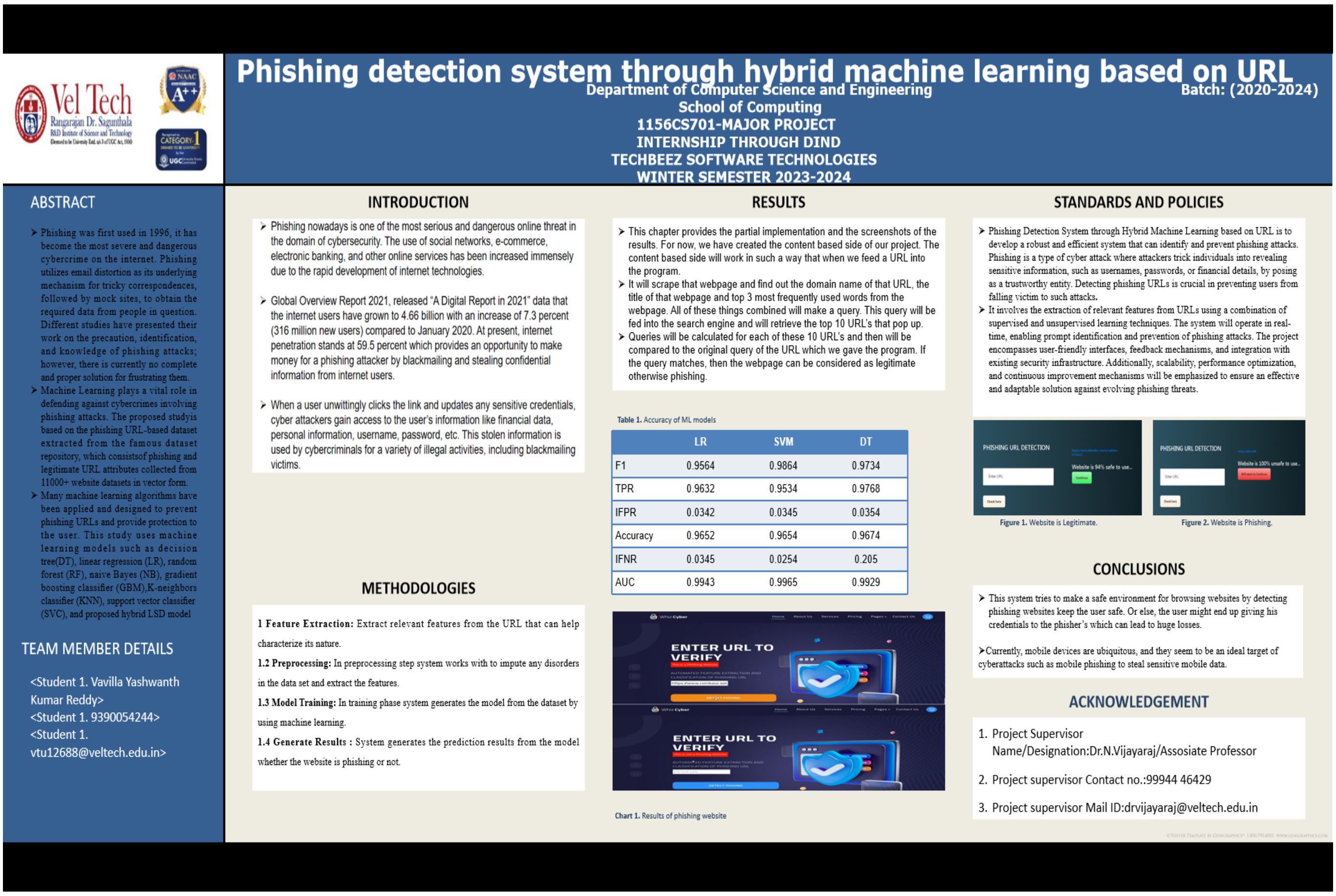
Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```
from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
warnings.filterwarnings(’ignore’)
from feature import FeatureExtraction

file = open(”pickle/cat.pkl”,”rb”)
gbc = pickle.load(file)
file.close()

app = Flask(__name__)

@app.route(”/”, methods=[”GET”, ”POST”])
def index():
    if request.method == ”POST”:

        url = request.form[”url”]
        obj = FeatureExtraction(url)
        x = np.array(obj.getFeaturesList()).reshape(1,30)
        y_pred =gbc.predict(x)[0]
        #1 is safe
        #-1 is unsafe
        y_pro_phishing = gbc.predict_proba(x)[0,0]
        y_pro_non_phishing = gbc.predict_proba(x)[0,1]
        # if(y_pred ==1 ):
        pred = ”It is {0:.2f} % safe to go ”.format(y_pro_phishing*100)
        return render_template(’index.html’,xx =round(y_pro_non_phishing,2),url=url )
    return render_template(”index.html”, xx =-1)
if __name__ == ”__main__”:
    app.run(debug=True)
```

10.2 Poster Presentation



Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)
NAAC A++
CATEGORY-1
UGC
ENTER URL TO VERIFY
AUTOMATED FEATURE EXTRACTION AND CLASSIFICATION OF PHISHING URL
Whiz Cyber
Home
About Us
Services
Pricing
Pages
Contact Us
PHISHING URL DETECTION
Enter URL
Check here
Website is 94% safe to use...
Continue
Website is 100% unsafe to use...
Still want to Continue

Phishing detection system through hybrid machine learning based on URL

Department of Computer Science and Engineering
School of Computing
1156CS701-MAJOR PROJECT
INTERNSHIP THROUGH DIND
TECHBEEZ SOFTWARE TECHNOLOGIES
WINTER SEMESTER 2023-2024

Batch: (2020-2024)

ABSTRACT

- Phishing was first used in 1996, it has become the most severe and dangerous cybercrime on the internet. Phishing utilizes email distortion as its underlying mechanism for tricky correspondences, followed by mock sites, to obtain the required data from people in question. Different studies have presented their work on the precaution, identification, and knowledge of phishing attacks; however, there is currently no complete and proper solution for frustrating them.
- Machine Learning plays a vital role in defending against cybercrimes involving phishing attacks. The proposed studyis based on the phishing URL-based dataset extracted from the famous dataset repository, which consistsof phishing and legitimate URL attributes collected from 11000+ website datasets in vector form.
- Many machine learning algorithms have been applied and designed to prevent phishing URLs and provide protection to the user. This study uses machine learning models such as decision tree(DT), linear regression (LR), random forest (RF), naive Bayes (NB), gradient boosting classifier (GBM),K-neighbors classifier (KNN), support vector classifier (SVC), and proposed hybrid LSD model

TEAM MEMBER DETAILS

<Student 1. Vavilla Yashwanth Kumar Reddy>
<Student 1. 9390054244>
<Student 1. vtu12688@veltech.edu.in>

INTRODUCTION

- Phishing nowadays is one of the most serious and dangerous online threat in the domain of cybersecurity. The use of social networks, e-commerce, electronic banking, and other online services has been increased immensely due to the rapid development of internet technologies.
- Global Overview Report 2021, released "A Digital Report in 2021" data that the internet users have grown to 4.66 billion with an increase of 7.3 percent (316 million new users) compared to January 2020. At present, internet penetration stands at 59.5 percent which provides an opportunity to make money for a phishing attacker by blackmailing and stealing confidential information from internet users.
- When a user unwittingly clicks the link and updates any sensitive credentials, cyber attackers gain access to the user's information like financial data, personal information, username, password, etc. This stolen information is used by cybercriminals for a variety of illegal activities, including blackmailing victims.

METHODOLOGIES

1 Feature Extraction: Extract relevant features from the URL that can help characterize its nature.

1.2 Preprocessing: In preprocessing step system works with to impute any disorders in the data set and extract the features.

1.3 Model Training: In training phase system generates the model from the dataset by using machine learning.

1.4 Generate Results : System generates the prediction results from the model whether the website is phishing or not.

RESULTS

- This chapter provides the partial implementation and the screenshots of the results. For now, we have created the content based side of our project. The content based side will work in such a way that when we feed a URL into the program.
- It will scrape that webpage and find out the domain name of that URL, the title of that webpage and top 3 most frequently used words from the webpage. All of these things combined will make a query. This query will be fed into the search engine and will retrieve the top 10 URL's that pop up.
- Queries will be calculated for each of these 10 URL's and then will be compared to the original query of the URL which we gave the program. If the query matches, then the webpage can be considered as legitimate otherwise phishing.

Table 1. Accuracy of ML models

	LR	SVM	DT
F1	0.9564	0.9864	0.9734
TPR	0.9632	0.9534	0.9768
IFPR	0.0342	0.0345	0.0354
Accuracy	0.9652	0.9654	0.9674
IFNR	0.0345	0.0254	0.205
AUC	0.9943	0.9965	0.9929

Chart 1. Results of phishing website

STANDARDS AND POLICIES

- Phishing Detection System through Hybrid Machine Learning based on URL is to develop a robust and efficient system that can identify and prevent phishing attacks. Phishing is a type of cyber attack where attackers trick individuals into revealing sensitive information, such as usernames, passwords, or financial details, by posing as a trustworthy entity. Detecting phishing URLs is crucial in preventing users from falling victim to such attacks.
- It involves the extraction of relevant features from URLs using a combination of supervised and unsupervised learning techniques. The system will operate in real-time, enabling prompt identification and prevention of phishing attacks. The project encompasses user-friendly interfaces, feedback mechanisms, and integration with existing security infrastructure. Additionally, scalability, performance optimization, and continuous improvement mechanisms will be emphasized to ensure an effective and adaptable solution against evolving phishing threats.

Figure 1. Website is Legitimate.

Figure 2. Website is Phishing.

CONCLUSIONS

- This system tries to make a safe environment for browsing websites by detecting phishing websites keep the user safe. Or else, the user might end up giving his credentials to the phisher's which can lead to huge losses.
- Currently, mobile devices are ubiquitous, and they seem to be an ideal target of cyberattacks such as mobile phishing to steal sensitive mobile data.

ACKNOWLEDGEMENT

1. Project Supervisor Name/Designation:Dr.N.Vijayaraj/Assosiate Professor
2. Project supervisor Contact no.:99944 46429
3. Project supervisor Mail ID:drvijayaraj@veltech.edu.in

References

[1] D. M. Divakaran and A. Oest, “Phishing detection leveraging machine learning and deep learning: A review,” 2022, arXiv:2205.07411.

[2] A. Akanchha, “Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates,” Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.

[3] H. Shahriar and S. Nimmagadda, “Network intrusion detection for TCP/IP packets with machine learning techniques,” in Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Cham,Switzerland: Springer, 2020, pp. 231–247.

[4] J. Kline, E. Oakes, and P. Barford, “A URL-based analysis of WWW structure and dynamics,” in Proc. Netw. Traffic Meas. Anal. Conf. (TMA),Jun. 2019, p. 800.

[5] A. K. Murthy and Suresha, “XML URL classification based on their semantic structure orientation for web mining applications,” Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.

[6] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, “PhishAri: Automatic realtime phishing detection on Twitter,” in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.

[7] S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security—ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.

[8] P. George and P. Vinod, “Composite email features for spam identification,” in Cyber Security. Singapore: Springer, 2018, pp. 281–289.

[9] H. S. Hota, A. K. Shrivas, and R. Hota, “An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique,” Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.

[10] G. Sonowal and K. S. Kuppusamy, “PhiDMA—A phishing detectionmodel with multi-filter approach,” J. King Saud Univ., Comput. Inf. Sci.,vol. 32, no. 1, pp. 99–112, Jan. 2020.

[11] R. Prasad and V. Rohokale, “Cyber threats and attack overview,” in Cyber Security: The Lifeline of Information and Communication Technology. Cham, Switzerland: Springer, 2020, pp. 15–31.

[12] Zhang, L.; Zhang, P. PhishTrim: Fast and adaptive phishing detection based on deep representation learning. In Proceedings of the 2020 IEEE International Conference on Web Services (ICWS), Beijing, China, 19–23 October 2020; pp. 176–180.

[13] Janet, B.; Reddy, S. Anti-phishing System using LSTM and CNN. In Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangaluru, India, 6–8 November 2020; pp. 1–5.

[14] Mahdavifar, S.; Ghorbani, A. Application of deep learning to cybersecurity: A survey. Neurocomputing 2019, 347, 149–176.

[15] Chai, J.; Zeng, H.; Li, A.; Ngai, E.W.T. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Mach. Learn. Appl. 2021, 6, 100134.

[16] Adebowale, M.A.; Lwin, K.T.; Hossain, M.A. Deep Learning with Convolutional Neural Network and Long Short-Term Memory for Phishing Detection. In Proceedings of the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Island of Ulkulhas, Maldives, 26–28 August 2019; pp. 1–8.

[17] Bahnsen, A.C.; Bohorquez, E.C.; Villegas, S.; Vargas, J.; González, F.A. Classifying phishing URLs using recurrent neural networks. In Proceedings of the 2017 APWG Symposium on Electronic Crime Research (eCrime), Phoenix, AZ, USA, 25–27 April 2017; pp. 1–8.

[18] Chen, W.; Zhang, W.; Su, Y. Phishing detection research based on LSTM recurrent neural network. In International Conference of Pioneering Computer Scientists, Engineers and Educators; ICPCSEE 2018: Zhengzhou, China, 2018; pp. 638–645.

[19] Ariyadasa, S.; Fernando, S.; Fernando, S. Detecting phishing attacks using a combined model of LSTM and CNN. Int. J. Adv. Appl. Sci. 2020, 7, 56–67.

[20] Pham, T.; Hoang, V.; Ha, T. Exploring Efficiency of Character-level Convolution Neuron Network and Long Short Term Memory on Malicious URL De-

tection. In Proceedings of the 2018 VII International Conference on Network, Communication and Computing–ICNCC 2018, Taipei City, Taiwan, 14–16 December 2018.

[21] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank," in Proc. Australas. Comput.Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY,USA: Association for Computing Machinery, 2020, pp. 1–11, Art. no. 3,doi: 10.1145/3373017.3373020.

General Instructions

- Cover Page should be printed as per the color template and the next page also should be printed in color as per the template
- **Wherever Figures applicable in Report , that page should be printed in color**
- Dont include general content , write more technical content
- Each chapter should minimum contain 3 pages
- Draw the notation of diagrams properly
- Every paragraph should be started with one tab space
- Literature review should be properly cited and described with content related to project
- All the diagrams should be properly described and dont include general information of any diagram
- Example Use case diagram - describe according to your project flow
- All diagrams,figures should be numbered according to the chapter number and it should be cited properly
- **Testing and codequality should done in Sonarqube Tool**
- Test cases should be written with test input and test output
- All the references should be cited in the report
- **AI Generated text will not be considered**
- **Submission of Project Execution Files with Code in GitHub Repository**
- **Thickness of Cover and Rear Page of Project report should be 180 GSM**
- **Internship Offer letter and neccessary documents should be attached**
- **Strictly dont change font style or font size of the template, and dont customize the latex code of report**
- **Report should be prepared according to the template only**
- **Any deviations from the report template,will be summarily rejected**

- **Number of Project Soft Binded copy for each and every batch is (n+1) copies as given in the table below**
- For **Standards and Policies** refer the below link https://law.resource.org/pub/in/manifest.in.html
- Plagiarism should be less than 15%
- **Journal/Conference Publication proofs should be attached in the last page of Project report after the references section**

width=!,height=!,page=-