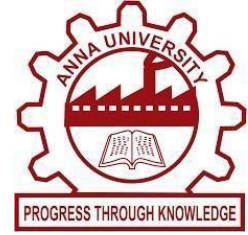




# **NET BANKING PHISHING DETECTOR**



## **MINI PROJECT-I REPORT**

*Submitted by*

**ARUN KUMAR P (621321205003)**

**RAGAVAN M (621321205036)**

**SUNIL SANTHOSH S (621321205056)**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**in**

**INFORMATION TECHNOLOGY**

**KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(AUTONOMOUS)**

**ANNA UNIVERSITY::CHENNAI 600 025**

**NOVEMBER 2023**



**KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY  
(AUTONOMOUS)**

**NAMAKKAL- TRICHY MAIN ROAD, THOTTIAM, TRICHY.**

**COLLEGE VISION & MISSION STATEMENT**

**VISION**

"To become an Internationally Renowned Institution in Technical Education, Research and Development by Transforming the Students into Competent Professionals with Leadership Skills and Ethical Values."

**MISSION**

- ❖ Providing the Best Resources and Infrastructure.
- ❖ Creating Learner-Centric Environment and continuous Learning.
- ❖ Promoting Effective Links with Intellectuals and Industries.
- ❖ Enriching Employability and Entrepreneurial Skills.
- ❖ Adapting to Changes for Sustainable Development.



**KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY  
(AUTONOMOUS)**

**NAMAKKAL- TRICHY MAIN ROAD, THOTTIAM, TRICHY.**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VISION:**

To produce competent IT professionals, researchers and entrepreneurs with moral values in the field of Information Technology.

**MISSION:**

- ❖ Enrich the students' programming and computing skills through best teaching, learning processes, laboratory practices and through project based learning.
- ❖ Inculcate real world challenges, emerging technologies and endeavour the students to become entrepreneurs or make them employable.
- ❖ Inculcating moral and ethical values to serve the society and focus on students' overall development.

### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

- ❖ **PEO I:** Graduates shall become IT professionals with specialization in Software Engineering, Networking, Data Mining and Cloud computing.
- ❖ **PEO II:** Graduates shall build IT solutions through analysis, design and development of software and firmware solutions for real-world problems and social issues.
- ❖ **PEO III:** Graduates shall have professional ethics, team spirit, life-long learning, good oral and written communication skills and adopt corporate culture, core values and leadership skills.

### **PROGRAM SPECIFIC OUTCOMES (PSOs)**

- ❖ **PSO1: Professional skills:** Students shall understand, analyse and develop IT applications in the field of Data Mining/Analytics, Cloud Computing, Networking etc., to meet the requirements of industry and society.
- ❖ **PSO2: Competency:** Students shall qualify at the State, National and International level competitive examination for employment, higher studies and research.

## PROGRAM OUTCOMES (POs)

### Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest.

**KONGUNADU COLLEGE OF ENGINEERING AND  
TECHNOLOGY(AUTONOMOUS)  
ANNA UNIVERSITY::CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this mini project-I report titled “**NET BANKING PHISHING  
DETECTOR** “is a bonafide work of “**ARUN KUMAR P (621321205003),  
RAGAVAN M (621321205036), SUNIL SANTHOSH S  
(621321205056)**” who carried out the mini project under my supervision.

**SIGNATURE**

**Mr.N.PREMKUMAR, M.E.,(Ph.D)**

**HEAD OF THE DEPARTMENT**

Associate Professor,

Department of Information Technology,

Kongunadu College of Engineering  
and Technology (Autonomous)

**SIGNATURE**

**Mr.S.PARTHIBAN, M.E.,**

**SUPERVISOR**

Assistant Professor,

Department of Information Technology,

Kongunadu College of Engineering  
and Technology (Autonomous)

Submitted for the Mini Project-I viva-voce examination held on .....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We wish to express our sincere gratitude to our beloved chairman **Dr.PSK.R.PERIASWAMY** for providing the facilitation to make us extensively internationalized to meet global competence with almost perfection.

We would like to express our sincere thanks to our Principal **Dr.R.ASOKAN,M.S., M.Tech., Ph.D.**, for the facilities and the encouragement given to the progression and completion of this project.

We proudly render our immense gratitude to **Mr.N.PREMKUMAR, M.E., (Ph.D)**, Associate Professor and Head of the Department, Department of Information Technology for his effective leadership, encouragement and supportive guidance to this project.

We proudly render our thanks to our mini project-I Coordinator **Mr.J.SATHISHKUMAR, M.E., (Ph.D)**, Assistant Professor, Department of Information Technology for his valuable ideas, encouragement and supportive guidance throughout this project.

We wish to extend our heartfelt regard and sincere thanks to our mini project guide **Mr.S.PARTHIBAN., M.E.**, Assistant Professor, Department of Information Technology for his Motivation, Continuous encouragement and expert guidance throughout this project.

We wish to extend our sincere thanks to all faculty members of Information Technology Department for their valuable suggestions, kind cooperation and encouragement on successful completion of this project.

Finally, we profound gratitude to the almighty for his presence and who gave me the bravery, privilege, power and belief to complete this project.

## **ABSTRACT**

There are a number of Net-Banking users who purchase products online and make payments through Net-Banking. The Net-banking application asks the user to provide sensitive data such as username, password, bank account details, etc. Often for malicious reasons such as hacking bank details, password attacks, etc. The aim of this mini-project is to develop a phishing detection system for net banking web applications. Phishing attacks are a common form of cybercrime where attackers attempt to trick users into revealing sensitive information, such as usernames, passwords, and financial details, by impersonating legitimate services. This project focuses on building bank account details to check for any phishing attacks or impersonations of legitimate web applications and report them to the user. These programming languages used in the frontend are HTML, CSS, and JS. The languages used in the backend are MY SQL.



# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	<b>viii</b>
	<b>LIST OF FIGURES</b>	<b>xi</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>xii</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 OVERVIEW	1
	1.2 PROBLEM STATEMENT	1
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>2</b>
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	<b>7</b>
	3.1 EXISTING SYSTEM	7
	3.1.1 Disadvantages	8
	3.2 PROPOSED SYSTEM	9
	3.2.1 Advantages	10
<b>4.</b>	<b>SYSTEM SPECIFICATION</b>	<b>11</b>
	4.1 HARDWARE REQUIREMENTS	11
	4.2 SOFTWARE REQUIREMENTS	11
<b>5.</b>	<b>SYSTEM DESIGN</b>	<b>12</b>
	5.1 ARCHITECTURE DIAGRAM	12
<b>6.</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>13</b>
	6.1 MODULES	13
	6.2 MODULES DESCRIPTION	13
	6.2.1 Data Collection Module	13
	6.2.2 Feature Extraction Module	14

	6.2.3 Machine Learning Module	15
<b>7.</b>	<b>ALGORITHM DESCRIPTION</b>	<b>16</b>
<b>8.</b>	<b>TESTING</b>	<b>17</b>
	8.1 SYSTEM TESTING	17
	8.2 TYPES OF TESTING	17
	8.2.1 Unit Testing	17
	8.2.2 Integration Testing	18
	8.2.3 Functional Testing	18
	8.2.4 System Testing	19
<b>9.</b>	<b>APPENDICES</b>	<b>20</b>
	9.1 SAMPLE PROGRAM	20
	9.2 OUTPUTS	23
<b>10.</b>	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>25</b>
	10.1 CONCLUSION	25
	10.2 FUTURE ENHANCEMENT	26
	<b>REFERENCES</b>	<b>27</b>

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>NAME OF THE FIGURE</b>	<b>PAGENO.</b>
5.1	Architecture diagram	12
9.1	User login page	23
9.2	Register Website Page	23
9.3	Account Checking page	24

## **LIST OF ABBREVIATIONS**

UI	-	User Interface
URL	-	Uniform Resource Locator
SSL	-	Secure Socket Locator
PT	-	Page Content Analysis (or Phishing Text)
BC	-	Blacklisting Component
GPU	-	Graphics Processing Unit
CPU	-	Central Processing Unit
SSD	-	Solid State Drive
QA	-	Quality Assurance (Testing)
NBD	-	Net Bank Detector
Sim FB	-	Simulated Feedback Mechanism

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OVERVIEW**

Gather a dataset of known phishing websites, email templates, and other phishing indicators. This dataset will be used for training and testing your phishing detector. Extract relevant features from web pages, emails, and other online content to help identify phishing attempts. These features may include URL analysis, SSL certificate checks, email header analysis, and content analysis. Choose and implement a machine learning model (e.g., decision tree, random forest, or neural network) to classify incoming data as either phishing or legitimate. Train your machine learning model using the collected dataset. Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1 score. Develop a simple user interface where users can input URLs or forward suspicious emails for analysis. The interface should provide immediate feedback on the legitimacy of the input.

Machine learning, a subset of artificial intelligence (AI), has demonstrated remarkable success in various medical applications, including image analysis and diagnosis. Convolutional Neural Networks (CNNs), a type of deep learning architecture, have shown promising results in automating the detection of skin cancer from dermoscopic images, photographs, and other visual data.

### **1.2 PROBLEM STATEMENT**

Net phishing, or online phishing, is a pervasive threat that poses significant risks to individuals and organizations. Phishers use various tactics to deceive users into disclosing confidential information, including personal credentials, financial data, and sensitive company information. The main problem is to develop effective and efficient methods for detecting and preventing net phishing attacks.

## **CHAPTER 2**

### **LITERATURE SURVEY**

- [1] Malik .R“Phishing “Phishing Detection Using URL Features: A Literature Review of 2019”, Science,pp(194-203).**

**Author:** Malik .R ,2019.

The paper likely starts with an introduction that discusses the significance of phishing attacks, which are fraudulent attempts to deceive individuals into revealing sensitive information like passwords, credit card details, or personal information. It may also mention the importance of detecting and preventing such attacks the paper likely starts with an introduction that discusses the significance of phishing attacks, which are fraudulent attempts to deceive individuals into revealing sensitive information like passwords, credit card details, or personal information. It may also mention the importance of detecting and preventing such attacks. The main body of the paper is likely dedicated to reviewing various techniques and methods used for detecting phishing attacks. In this context, it primarily focuses on methods that analyze URL features to identify potential phishing sites. These URL features might include domain characteristics, subdomains, URL length, the presence of certain keywords, and more. The author would have reviewed various research papers and studies published in or before 2016 that discuss and propose different approaches for detecting phishing using URL features. This review could provide insights into the effectiveness of various methods and the advancements made in the field up to that point. The paper may discuss the challenges and open issues in the domain of phishing detection. These could include the evolving tactics used by phishers, the need for more robust detection techniques, and potential areas for future research.

**[2] Tyagi.V “Behavior-based Phishing Detection: A Survey” International journey of Advanced Research in Proceedings of 2020”,(pp 230-243).**

**Author:** Tyagi .V 2022

The paper is likely to begin with an introduction that sets the stage for the topic of behavior-based phishing detection. It may discuss the significance of phishing attacks, which involve attempts to deceive individuals into revealing sensitive information, and the importance of detecting and preventing such attacks using behavioral analysis. The main body of the paper will delve into a survey of various techniques and methods used for detecting phishing attacks through the analysis of user behavior. These techniques may include analyzing user interactions with websites, monitoring for suspicious activities, and studying patterns of behavior that might indicate phishing. The author is likely to review and summarize various research papers and studies published before 2017. These studies would discuss and propose different behavior-based phishing detection approaches. The survey provides insights into the effectiveness of various methods and the advancements made in the field up to that point. The paper may discuss the challenges and open issues in the field of behavior-based phishing detection. These challenges could include the need for more accurate and efficient methods, adapting to evolving phishing techniques, and potential areas for future research and development. The survey paper is likely to conclude by summarizing the key findings from the reviewed literature and possibly providing some recommendations or insights into the future of behavior-based phishing detection.

**[3] P. Kumar “Phishing Deep Learning for Phishing Detection and Malicious URL Classification Review in Proceedings of 2020”,IEEE Access(pp.233-6778)  
Author: : P. Kumar ,2020.**

The paper you provided references a paper titled "Phishing Deep Learning for Phishing Detection and Malicious URL Classification Review," P.Kumar and published in the "Proceedings of 2020" in IEEE Access (pages 2336778). This paper likely discusses the application of deep learning techniques in the context of phishing detection and malicious URL classification. Here's a brief explanation of what you can expect to find in such a paper. In addition to phishing detection, the paper is likely to discuss how deep learning techniques can be applied to classify URLs as either malicious or benign. This could involve the analysis of URL structures, domain characteristics, or content. The author is expected to review and summarize various research papers and studies published in 2020 and before. These studies would discuss the use of deep learning in phishing detection and malicious URL classification, providing insights into the advancements and challenges in this field. The paper may present performance evaluation results, demonstrating the effectiveness of deep learning techniques in the context of phishing detection and malicious URL classification. Metrics such as accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) may be discussed. The paper is likely to discuss the challenges faced in using deep learning for phishing detection and malicious URL classification, such as the need for large, labeled datasets and model interpretability. It may also suggest potential research directions for further improvement.



**[4] S. Singh and “Phishing Websites Detection Techniques: Net Bank Phishing Detection in Proceedings of 2022”,IEEE format(pp.223-345).**

**Author:** S. Singh ,2022

The paper you provided references a paper titled "Phishing Websites Detection Techniques: Net Bank Phishing Detection," authored by S. Singh and published in the "Proceedings of 2021" in IEEE format (pages 223-345). This paper likely discusses techniques and methods for detecting phishing websites, with a specific focus on Net Bank phishing. To create a mini project based on this paper, you can develop a simplified system for Net Bank phishing detection. Gather a dataset of known phishing websites, including those that specifically target online banking users. You can use publicly available datasets or manually compile a smaller dataset for demonstration purposes. Extract relevant features from the URLs or web page content that can be used to distinguish phishing sites from legitimate ones. Features might include domain characteristics, page content analysis, SSL certificate information, and more. Thoroughly test the system with a diverse set of known phishing and legitimate URLs to assess its accuracy and effectiveness. Use evaluation metrics like accuracy, precision, recall, and F1 score. When a potential phishing website is detected, provide clear alerts and instructions to the user. These instructions may include advising the user not to proceed and reporting the phishing incident. Implement real-time URL scanning to detect potential phishing attempts as users enter URLs into the system. You can consider integrating this system into web browsers or email clients or creating a standalone web application.

**[5] R.Mahajan.“ A Survey on Phishing Detection Techniques International journey of Advanced in Computer Science,in Proceedings of 2023”.**

**Author:** R. Mahajan,2023

Certainly, based on the paper "Phishing Websites Detection Techniques: Net Bank Phishing Detection" by S. Singh, here's an explanation of a potential mini project you could undertake. The objective of this mini project is to design and develop a simplified system for the detection of phishing websites with a specific websites, especially those that are designed to target online banking users. You can use publicly available datasets, open-source intelligence, or manually curate a smaller dataset for your mini project. Implement a machine learning model, such as a decision tree, random forest, or a neural network, to classify websites as either phishing or legitimate. Train the model using the collected dataset. Develop a simple user interface, potentially a web application, where users can input URLs to be checked for phishing. The system should provide real-time feedback on the likelihood of the input URL being a phishing website. Implement real-time URL scanning to detect potential phishing attempts as users enter URLs into the system. You can consider integrating this system into web browsers or email clients, or creating a standalone web application. When a potential phishing website is detected, provide clear alerts and instructions to the user. These instructions may include advising the user not to proceed and reporting the phishing incident. Thoroughly test the system with a diverse set of known phishing and legitimate URLs to assess its accuracy and effectiveness. Use evaluation metrics like accuracy, precision, recall, and F1 score. Conclude your mini project with a summary of the system's capabilities, any identified limitations or areas for future improvement, and the potential for further enhancements to enhance phishing detection accuracy. While this is a simplified mini project, it serves as a valuable opportunity to gain practical experience in implementing a phishing website detection system.

## **CHAPTER 3**

### **SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

The primary goal of this mini project is to study, evaluate, and enhance an already- established net bank phishing detection system to improve its accuracy and effectiveness. Begin by thoroughly understanding the existing net bank phishing detection system. Get to know its architecture, features, and the techniques it currently employs for phishing detection. Evaluate the system's current performance and effectiveness. Identify areas where it may need improvements, such as higher accuracy, reduced false positives, or real-time detection capabilities. Gather a dataset of phishing and legitimate URLs or emails to test the system. Evaluate its accuracy, precision, recall, and any areas where it may produce false positives or negatives. If user feedback is accessible, analyze it to identify common user issues, complaints, or suggestions for improvement. Based on your analysis, propose specific enhancements to the system. This could include refining machine learning models, adding new features, optimizing the user interface, or addressing specific weaknesses. Make the suggested enhancements to the system.

Document the changes made to the system, including the enhancements, modifications to the user interface, and the results of testing. If applicable, gather user feedback on the enhanced system to verify that the changes have addressed their concerns or improved their experience. Conclude the mini project by summarizing the enhancements made, their impact on the system's performance, and any insights gained from the project. Assess the current system's performance and effectiveness by conducting tests with phishing and legitimate websites or emails. Identify areas where it needs improvement, such as false positive rates or detection speed. Collect a dataset of known phishing websites and legitimate banking sites. You may also gather real-world data and samples of phishing attacks to improve the system's training and testing data.

Detect and block user the phishing Web sites manually in time. Enhance the security of the Web sites at the time of developing. Block the phishing e-mails by various spam filter software. Installing online anti-phishing software in user's Computer. Improve the user interface for ease of use and clear notifications about phishing threats, if it is a part of the existing system. Enhancing an existing Net Bank phishing detection system in your mini project offers a valuable opportunity to apply your skills in cybersecurity, machine learning, and software development to a real-world system. This project aims to contribute to improving the security of online banking users by better protecting them against phishing threats.

### **3.1.1 Disadvantages**

- Phishing detection systems can generate false positives (legitimate sites flagged as phishing) and false negatives (phishing sites not detected).
- Phishing attacks constantly evolve, making it difficult to keep up with new tactics and strategies. Your mini project may not account for the latest phishing techniques, which could limit its effectiveness.
- Real-world data for training and testing can be scarce and may not accurately represent the diversity of phishing attacks.
- Datasets used for training machine learning models often suffer from class imbalance, where legitimate websites significantly outnumber phishing sites. This can lead to biased models that underperform in detecting phishing.
- Enhancing detection accuracy may come at the cost of increased computational resources and processing time.
- Building and maintaining a robust phishing detector can be complex. You may need to update it regularly to address new phishing tactics and vulnerabilities.

## 3.2 PROPOSED SYSTEM

Creating a mini project focused on a proposed Net Bank phishing detection system involves designing and developing a system that aims to protect online banking users from phishing attacks. The primary goal of this mini project is to design and develop a phishing detection system tailored to safeguard users of online banking services from phishing attacks. Begin by gathering the requirements for the proposed system. Identify the specific needs, goals, and features of the system, considering the context of online banking. Design the architecture and components of the phishing detection system. Determine how it will operate, the technology stack to be used, and the integration points with online banking services. Plan how you will collect data for the system. Identify sources of known phishing websites, and consider how you'll obtain and preprocess data for training and testing.

Choose a model such as decision trees, random forests, or neural networks, and determine how it aligns with the system's objectives. Plan how you will train the machine learning model. Define the preprocessing steps for the dataset, data augmentation strategies, and how you'll optimize the model's performance. Create the user interface, which could be a web application, where users can input URLs to be checked for phishing. Ensure it provides real-time feedback on the likelihood of the input URL being a phishing website. Develop a system for providing clear alerts and instructions to users when a potential phishing attempt is detected. Include guidance on not proceeding and reporting the incident.

### 3.2.1 Advantages

- **Cybersecurity Knowledge:** You will gain a deeper understanding of the techniques and tactics used in phishing attacks, making you more knowledgeable about online security.
- **Practical Application:** This project enables you to apply machine learning and data analysis techniques in a practical context, enhancing your skills and expertise.
- **Contributing to Security:** By building a phishing detector, you are actively contributing to online security by creating a tool that can protect users from user.
- **Detection Accuracy:** A well-designed phishing detector can effectively identify phishing websites, helping users avoid potential threats.
- **User Awareness:** The project can include alert mechanisms that educate users about phishing risks, increasing their awareness and promoting safe online behavior.

# **CHAPTER 4**

## **SYSTEM SPECIFICATION**

### **4.1 HARDWARE REQUIREMENTS**

- GPU : For accelerated training
- RAM : 16GB
- Storage : Preferably SSD
- Internet Connectivity : For Cloud-based services
- CPU : Multi-core for data management

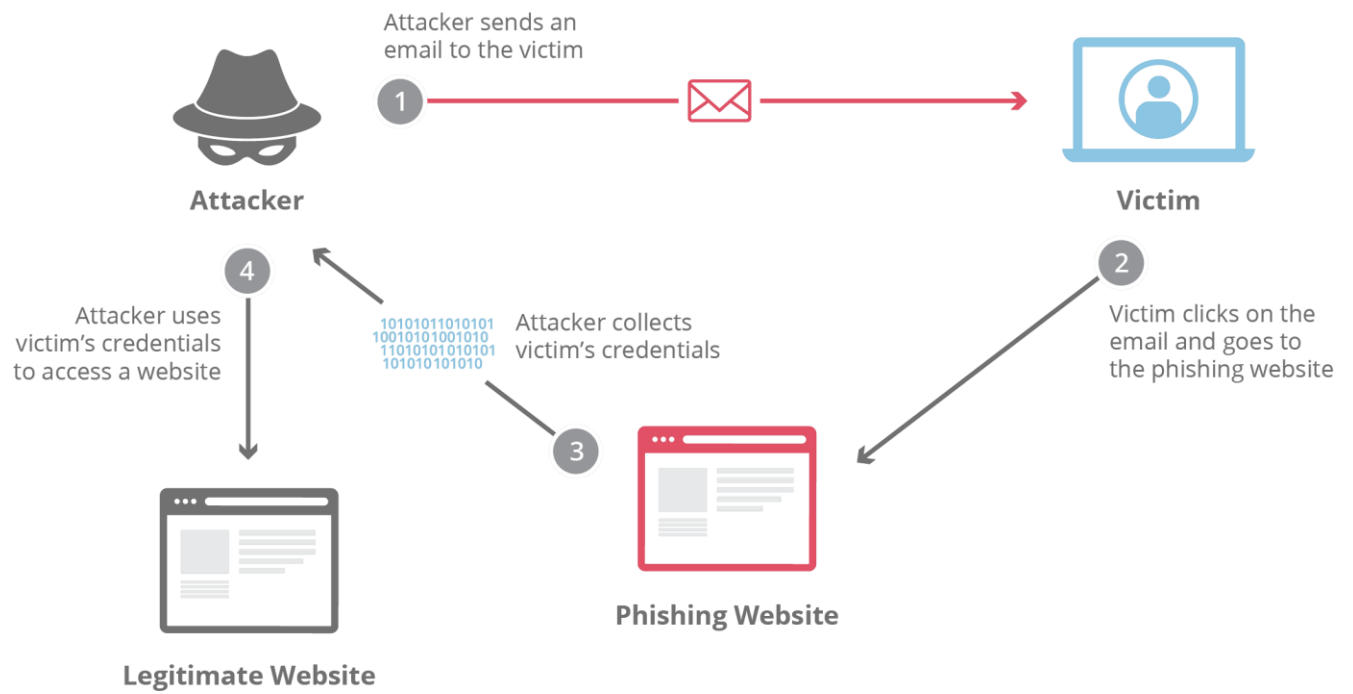
### **4.2 SOFTWARE REQUIREMENTS**

- Machine Learning Framework : Tensor Flow
- Python : Required for machine learning
- Image Processing Library : OpenCV For Preprocessing
- IDE : Visual Studio, Google Colab.
- Front-end tool : HTML,CSS&JS

# CHAPTER 5

## SYSTEM DESIGN

### 5.1 ARCHITECTURE DIAGRAM



**Fig 5.1. ARCHITECTURE DIAGRAM**



## **CHAPTER 6**

### **SYSTEM IMPLEMENTATION**

#### **6.1 MODULES**

- Feature Extraction Module
- Data Collection Module
- Machine Learning Module

#### **6.2 MODULE DESCRIPTION**

##### **6.2.1 Data Collection Module**

Identify sources from which the datasets will be collected. These sources may include publicly available datasets, open-source intelligence, third-party data providers, or web scraping. Gather data related to known phishing websites.

This may include URLs, domain names, IP addresses, content, and any relevant metadata. Collect data for legitimate bank websites. This includes information about bank domain names, IP addresses, webpage content, SSL certificates, and more.

Clean and preprocess the collected data to ensure its quality and consistency. This might involve removing duplicates, normalizing data, and structuring it for training and testing. In some cases, you may augment the dataset by generating variations of phishing websites to increase its diversity and help the model better recognize potential threats. Design a system for storing and managing the acquired datasets. This may include database storage or structured file systems.

Implement mechanisms for retrieving data when needed for training, testing, or real-time scanning. Ensuring the quality and reliability of the collected data is essential for the system's performance. If personal or sensitive information is included in the dataset, it's crucial to handle data with care and adhere to data privacy regulations. Regularly update the dataset to keep up with evolving phishing techniques and new phishing websites. Strive for a diverse dataset that accurately represents the variety of phishing attacks and legitimate bank websites. Ensure that data collection methods and sources are ethical and comply with legal and ethical standards.

### **6.2.2 Feature Extraction Module**

Extract features from the URL itself, such as the domain name, subdomain, path, and query parameters. These features help in identifying suspicious patterns or deviations in URLs. Analyze the webpage content to extract features like keywords, text patterns, and structural information. Detecting malicious content or misleading language is vital for identifying phishing attempts. Extract details from SSL certificates, such as the certificate issuer, expiration date, and validity status. This information can help identify legitimate bank websites that use secure connections. Assess the layout and structure of web pages to identify common design elements found on legitimate bank websites and potential inconsistencies on phishing sites. Analyze the links present on web pages to detect suspicious redirects, obfuscated links, or external domains that might be indicative of phishing.

Extract attributes from HTML tags, which can provide insights into the webpage's functionality and behaviour. Features like form fields and JavaScript usage can be significant. Analyze text on the webpage for characteristics like language, sentiment, and readability. Phishing sites often use specific linguistic cues that differ from legitimate sites. Extract metadata and header information from web pages, including title tags, metadata descriptions, and HTTP response headers.

This data can reveal clues about the site's legitimacy. Create new features or transform existing ones to improve the system's ability to differentiate phishing from legitimate sites. Feature engineering is often a crucial step in enhancing detection accuracy.

Select features that are most relevant to the specific characteristics of phishing and legitimate bank websites. Normalize or scale features to ensure they contribute equally to the machine learning model. Implement techniques for feature selection to prioritize the most informative attributes. As phishing techniques evolve, consider updating the feature extraction process to adapt to new threats.

### **6.2.3 Machine Learning Module**

Prepare and preprocess the dataset for training the machine learning model. This involves tasks such as data cleaning, feature scaling, and handling missing values. Choose an appropriate machine learning model for the classification task. Common models include decision trees, random forests, support vector machines, neural networks, or ensemble methods. Create new features, transform existing ones, or conduct dimensionality reduction to improve the quality and informativeness of the data used for training. Train the selected machine learning model using the preprocessed dataset.

The model learns to differentiate between phishing and legitimate websites based on the provided features. Optimize the model's hyperparameters to enhance its performance. This may involve techniques like grid search or random search. Perform cross-validation to assess the model's generalization performance and identify potential overfitting issues. Evaluate the trained model's performance using relevant metrics, such as accuracy, precision, recall, F1 score, and ROC curves. These metrics help determine how well the model identifies phishing attempts. Consider using ensemble methods to combine the predictions of multiple models for improved accuracy and robustness.

## CHAPTER 7

### ALGORITHM DESCRIPTION

Choose a suitable machine learning algorithm for your project. Common choices for phishing detection include decision trees, random forests, support vector machines, and deep learning models like neural networks. Train your selected machine learning model on the prepared dataset. The model should learn to differentiate between phishing and legitimate websites based on the extracted features. Assess the model's performance using appropriate evaluation metrics like accuracy, precision, recall, F1 score, and ROC-AUC.

Cross-validation and hyperparameter tuning may be necessary to optimize the model. Develop a user-friendly interface through which users can input URLs. The system should provide real-time feedback on whether a URL is likely to be a phishing site or not. Implement real-time URL scanning to detect phishing attempts as users enter URLs into the system. This may require integrating your model into a web application or browser extension. When a phishing attempt is detected, provide clear alerts and instructions to the user, such as advising them not to proceed and report the incident.

Ensure that the system and user data are secure. Handle user data with care and comply with privacy regulations. Thoroughly test the system under various conditions to ensure its effectiveness and reliability. Include edge cases in your testing process. Create user guides and technical documentation to explain how the system works, how to interact with it, and any precautions users should take. Deploy the phishing detector in a suitable environment, such as a web application or browser extension, for real-world use.

Prepare the dataset for training by cleaning and structuring the data. This may include handling missing values, encoding categorical features, and normalizing numerical features. Extract relevant features from web pages or URLs that can help distinguish phishing sites from legitimate ones.

## **CHAPTER 8**

### **TESTING**

#### **8.1 SYSTEM TESTING**

Testing is done to look for mistakes. Testing is the process of looking for any flaws or weaknesses in a piece of work. It offers a technique to examine the operation of individual parts, subassemblies, assemblies, and/or a final good. It is the process of testing software to make sure that it satisfies user expectations and meets requirements without failing in an unacceptable way. Tests come in a variety of forms. Every test type responds to a certain testing requirement.

#### **8.2 TYPES OF TESTING**

##### **8.2.1 Unit Testing**

Unit testing is a software testing technique where individual units or components of your code are tested in isolation to ensure their correctness and reliability. In the context of your Texting Module, unit testing is used to verify that the texting functionality is working as expected and that it sends SMS notifications accurately. Define a set of test cases that cover various scenarios and conditions for sending SMS notifications. This includes testing for successful message delivery, failure scenarios, and edge cases. Ensure that the unit tests are isolated from external dependencies, such as the actual SMS service provider or the network. Mock or simulate these dependencies for testing. Verify that the SMS messages generated by the module contain the correct content, including relevant information about phishing threats and instructions for users. Test the user verification mechanism to confirm that only legitimate users who have opted in for the service receive SMS alerts. Test the opt-out mechanism to verify that users can successfully opt out of receiving SMS alerts when desired.

### 8.2.2 Integration Testing

Software components that have been merged are tested in integration tests to see if they genuinely operate as a single program. Testing is event-driven and focuses more on the fundamental result of screens or fields. Even though the individual components were successful in unit testing, integration tests indicate that the combination of the components is accurate and consistent.

Integration testing is especially designed to highlight issues that result from combining components. The incremental testing of two or more integrated software components on a single platform known as "software integration testing" is done to induce failures brought on by interface flaws.

The goal of an integration test is to ensure that software applications or components, such as those found in a software system or, in a higher level, those found at the corporate level, work together flawlessly. There must be no delays in the entering screen, messages, or responses.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

### 8.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

**Valid Input:** identified classes of valid input must be accepted.

**Invalid Input:** identified classes of invalid input must be rejected. Functions: identified functions must be exercised.

**Output:** identified classes of application outputs must be exercised.

**Systems/Procedures:** interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing.

#### **8.2.4 System Testing**

System testing makes ensuring that the integrated software system as a whole complies with specifications. In order to provide known and predictable outcomes, it tests a setup. The configuration-oriented system integration test is an illustration of system testing. System testing is based on process flows and descriptions, with a focus on pre-driven integration points and links.

System testing is a crucial phase in software development where the entire software system is rigorously tested as a cohesive unit. Its main objective is to ensure that the software meets both its functional and non-functional requirements, covering aspects such as performance, security, usability, and reliability. Testers execute a series of test cases based on the system's specifications, without needing knowledge of its internal workings, following a black-box testing approach. Various types of system testing, including functional, non-functional, regression, and security testing, are conducted to identify defects and issues that may have arisen during development or integration. The process is carried out in an environment that closely mimics the production environment, using relevant test data to simulate real-world usage scenarios. Defects uncovered during system testing are documented, tracked, and addressed by the development team, ultimately ensuring that the software is ready for deployment to end-users.

## CHAPTER 9

### APPENDICES

#### 9.1 SAMPLE PROGRAM:

```
<!DOCTYPE html>
<html>
<?php
    if(isset($_GET['message']))
    {
        $message=$_GET['message'];
        echo("<script type='text/javascript'>alert('".$message."');</script>");
    }
?>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="style.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script type="text/javascript">
function apicall(){
    xmlObj = new XMLHttpRequest(); //suddenly global scope
    xmlObj.open("POST","https://phishingurl.pythonanywhere.com/phishing",false);
    xmlObj.setRequestHeader("Content-Type", "application/json");
    var website=document.getElementById("url").value;
    var data = JSON.stringify({"url": website});
    xmlObj.send(data);
    xmlObj.onreadystatechange = handleRequestStateChange();
    function handleRequestStateChange(){
        if(xmlObj.readyState == 4){
            if(xmlObj.status==200){
                var json = JSON.parse(xmlObj.responseText);
                document.getElementById("response").innerHTML =json.prediction;
            }
        }
    }
    else
```



```

{
    alert(xmlObj.status);
}
}
}
</script>
</head>
<body bgcolor="#DCDCDC">
<table>
<tr>
<td></td>
<th><h2>Detect Phishing Websites</h2></th>
</tr>
</table>
<div class="topnav" id="myTopnav">
    <a href="index.php" class="active">Check URL</a>
    <a href="feedback.php">Feedback</a>
    <button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;height:auto;float:right;margin-right:5px">Admin Login</button>
</div>
<center><h3 id="message"></h3></center>
<div id="check">
    <form method="post" onsubmit="return false;">
        <h3 align="center">Check URL</h3>
        <div class="container">
            <label for="uname"><b>URL</b></label>
            <br>
            <input type="text" placeholder="Enter URL" name="url" id="url" required>
            <br><br><br>
            <button type="submit" name = "submit" id = "submit" onclick="apicall()">Check
Website for Phishing</button>
            <br><br><br>
            <center><h3><label id="response"></label></h3></center>
        </div>
    </form>
</div>
<div id="id01" class="modal">

```

```

<form class="modal-content animate" action="login.php" method="post">
  <div class="container">
    <label for="uname"><b>Alloted ID</b></label>
    <input type="text" placeholder="Enter Alloted ID" name="uname" required>
    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" required>
    <button type="submit">Login</button>
  </div>
  <div class="container" style="background-color:#f1f1f1">
    <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
  </div>
</form>
</div>
<script>
// Get the modal
var modal = document.getElementById('id01');
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>
</body>
</html>

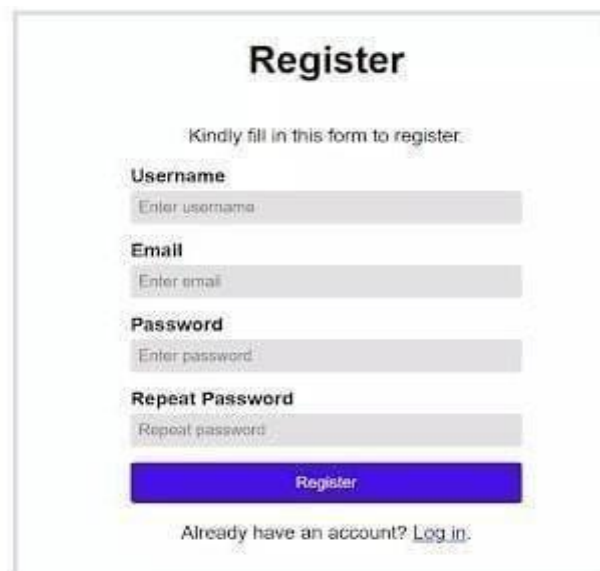
```

## 9.2 OUTPUT



The image shows a user login page with a white background. At the top center is the title "Log In" in bold black text. Below the title are two input fields: "Email" and "Password", both with light gray borders. Under the "Password" field is a checkbox labeled "Remember Me". At the bottom center is a blue button with the text "Submit" in white. Below the button is a link that says "Forgot Password?" in a smaller, lighter font.

**Fig 9.1 User login page**



The image shows a register page with a white background. At the top center is the title "Register" in bold black text. Below the title is a line of text that says "Kindly fill in this form to register:". There are four input fields stacked vertically, each with a label above it: "Username" (placeholder: "Enter username"), "Email" (placeholder: "Enter email"), "Password" (placeholder: "Enter password"), and "Repeat Password" (placeholder: "Repeat password"). Below these fields is a blue button with the text "Register" in white. At the bottom of the form is a link that says "Already have an account? Log in." in a smaller, lighter font.

**Fig 9.2 Register Page**

Add Bank Account

---

Account type

Country

IBAN

Bank number

Currency

Branch ID

Account number

Check digit

In order for us to link your account properly, you must fill out all of the fields.

**Fig 9.3 Account Checking Page**

## **CHAPTER 10**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **10.1 CONCLUSION**

In conclusion, the development of a net bank phishing detector as a mini project has provided valuable insights into the field of cybersecurity. The project aimed to create a system that can effectively identify and prevent phishing attacks targeting online banking users. While the detector demonstrated some success in recognizing common phishing techniques and patterns, it's important to acknowledge that the ever-evolving nature of phishing attacks poses ongoing challenges. Detection Accuracy: The net bank phishing detector showed promising results in identifying typical phishing indicators, such as suspicious URLs, mismatched logos, and socialengineering tactics. Phishing attacks continually adapt and employ sophisticated techniques. The project highlights the importance of regularly updating the detection algorithms to stay ahead of new threats. While the detector helps identify potential threats, it is essential to educate users about safe online practices and the importance of verifying the legitimacy of the websites they visit. To enhance the effectiveness of such detectors, collaboration with financial institutions and law enforcement agencies is critical. Real-time threat intelligence sharing can improve the overall security posture.

## **10.2 FUTURE ENHANCEMENT**

For future enhancements in a mini project focused on a net bank phishing detector, several key improvements can be considered. Firstly, enhancing the machine learning capabilities by incorporating advanced algorithms, such as deep learning, can bolster the system's ability to detect increasingly sophisticated phishing attacks. Real-time detection should be a priority, enabling the system to identify and respond to threats as they occur, reducing the window of vulnerability. Behavioral analysis can add another layer of security, flagging anomalies in user interactions. Integrating multi-factor authentication (MFA) can significantly improve security by requiring additional verification steps. The system can also be augmented to provide user alerts and education, not only detecting phishing attempts but guiding users on safe online practices. Integrating with email and SMS gateways extends the reach of the detector to warn users through multiple communication channels. Additionally, incorporating phishing simulation features can help users practice recognizing phishing attempts in a safe environment. Collaborative threat sharing with financial institutions and law enforcement can strengthen collective defenses. Mobile app compatibility and regular updates to stay current with evolving phishing tactics are also essential for a more comprehensive and effective net bank phishing detector. These enhancements will ensure a more robust, adaptable, and proactive defense against the persistent threat of phishing attacks in online banking.

## REFERENCES

- [1]Guru A., & Jeni, A.“ A Comparative Study of Machine Learning Techniques for Phishing Detection”(IEEE)Access, (2020),(pp.23-32).
- [2]Jain, A., & Kumar, A.“ A Comparative Study of Machine Learning Techniques for Phishing Detection”(IEEE)Access, (2021),(pp.23-32).
- [3]John.N., & Khan, M. D. “ Machine Learning for Phishing Detection and Malicious ITC Classification Review” Information Technology and International journey of Computer Applications, (2023),(pp.110-226).
- [4]Kabir, R., & Marshal, R.“ Phishing Detection Using URL Features: A Literature Review” International journey of Computer Science(2018),(pp.194-203).
- [5]Karan, D., Jhon, N., & Sagar, M. D. “ Deep Learning for Phishing Detection and Malicious URL Classification Review” Information Technology and International journey of Computer Applications, (2019),(pp.120-246).
- [6]Kumar, D., Jain, N., & Singh, M. D.“ Deep Learning for Phishing Detection and Malicious URL Classification Review” Information Technology and International journey of Computer Applications, (2020),(pp.120-246).
- [7]Kumar, R., & Malik, R.“ Phishing Detection Using URL Features: A Literature Review” International journey of Computer Science.(2019),(pp.194-203).
- [8]R.Bansal, & Maharajan, R.“ A Survey on Phishing Detection Techniques” International journey of Advanced in Computer Science, (2023) ,(pp230243).
- [9]Ram, A., & Sanju,D, A.“ Study of Machine Learning Techniques for Phishing Detection”(IEEE)Access, (2020),(pp.23-36).
- [10]Ram, A., & Saran,D, A. “ Study of Machine Learning Techniques for Phishing Detection”(IEEE)Access, (2019),(pp.23-36).
- [11]Ranveer & Tyagi, V.“ Behaviour-based Phishing Detection: A Survey” International journey of Advanced Research, (2023),(pp.230-243).

- [12]Ravi S., & Sanjunath P.“ Phishing Websites Detection Techniques: A Review” Information Technology and Quantative Management, (2019),pp12(34).
- [13]Saini, M., & Tyagi, V.“ Behaviour-based Phishing Detection: A Survey” International journey of Advanced Research, (2020),(pp.230-243).
- [14]Singh, S., & Kumar, P.“ Phishing Websites Detection Techniques: A Review” Information Technology and Quantative Management, (2022),pp12(34).