ABSTRACT

The purpose of the machine learning-based cyber threat detection platform is to identify and stop attacks using raw urls. Cybersecurity is seriously threatened by the spread of rogue URLs. These rogue websites mostly seek to trick users into disclosing sensitive personal or financial information. We introduce a Cyber Threat Detection platform that uses machine learning algorithms to distinguish between phishing and authentic URLs in order to combat this threat. We improve users' capacity to distinguish between secure and harmful websites by classifying raw URLs into these two groups. While phishing URLs use false messages to trick naive people, legitimate URLs are safe places to browse. Our software improves internet security by reliably classifying URLs using a trained model. Through this research, we contribute to the ongoing efforts to prevent cyber attacks, providing a robust solution for detecting and mitigating the dangers associated with malicious URLs.

Index Terms :

Cyber Threat Detection, Machine Learning, Malicious URLs, Phishing, Cyber Security, Raw URL Analysis, Online Security, Fraud Detection, Personal Information Protection, Financial Information Security.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	IV
	LIST OF ABBREVATIONS	VIII
	LIST OF FIGURES	IX
1 .	**INTRODUCTION**	1
	1.1 PROJECT OVERVIEW	3
	1.2 PURPOSE	3
	1.3 MACHINE LEARNING	4
	1.3.1 CHARACTERISTICS OF MACHINE LEARNING	4
	1.3.2 BENEFITS OF MACHINE LEARNING	5
2 .	**LITERATURE SURVEY**	7
3 .	**SYSTEM ANALYSIS**	13
	3.1 EXISTING SYSTEM	13
	3.1.1 DISADVANTAGES OF EXISTING SYSTEMV	13
	3.2 PROPOSED SYSTEM	15
	3.2.1 ADVANTAGES OF PROPOSED SYSTEM	15
4 .	**PROJECT DESCRIPTION**	16
	4.1 INPUT DESIGN AND OUTPUT DESIGN	16
	4.1.1 INPUT DESIGN	16

4.1.2 OBJECTIVES 16

4.1.3 OUTPUT DESIGN 17

5 . REQUIREMENT ANALYSIS 18

5.1 FUNCTIONAL REQUIREMENTS 18

5.2 NON-FUNCTIONAL REQUIREMENTS 19

6 . SYSTEM DESIGN 20

6.1 SYSTEM ARCHITECTURE 20

6.2 DATA FLOW DIAGRAM 22

7. UML DIAGRAMS 23

7.1 USE CASE DIAGRAM 24

7.2 CLASS DIAGRAM 25

7.3 ACTIVITY DIAGRAM 26

7.4 SEQUENCE DIAGRAM 26

8 . MODULES DESCRIPTION 28

8.1 MODULES SPLITUP 28

8.1.1 DATASET COLLECTION & PREPROCESSING 28

8.1.2 LEXICAL FEATURES 29

8.1.2.1 FEATURES 29

8.1.3 MODEL BUILDING & EXTRACTION 32

8.1.4 DEPLOYMENT 32

8.1.4.1 FLASK 33

9.	**TECHNIQUES AND ALGORITHMS**	34
	9.1 ALGORITHMS	34
	9.1.1 LOGISTIC REGRESSION	34
	9.1.2 RANDOM FOREST CLASSIFIER	38
	9.1.3 K-NEAREST NEIGHBOUR	42
10.	**SYSTEM REQUIREMENTS**	44
	10.1 HARDWARE REQUIREMENTS	44
	10.2 SOFTWARE REQUIREMENTS	44
11.	**RESULT AND DISCUSSIONS**	45
12.	**CONCLUSION**	50
13 .	**FUTURE ENHANCEMENT**	51
14.	**REFERENCE**	52
15.	**ANNEXURE**	54

LIST OF ABBREVATIONS

S.NO	ABBREVATIONS	EXPANSION
1	**API**	Application Protocol Interface
2	**CLI**	Command Line Interface
3	**GUI**	Graphical User Interface
4	**HTTPS**	Hyper-Text Transfer Protocol Secure
5	**IP**	Internet Protocol
6	**UML**	Unified Modelling Language
7	**URL**	Uniform Resource Locator
8	**SSL**	Secure Sockets Layer
9	**TLD**	Top-Level Domain

LIST OF FIGURES

FIGURE NO	LIST OF FIGURES	PAGE NO.
1	STRUCTURE OF URL	2
1.3.1	CHARACTERISTICS OF MACHINE LEARNING.	5
1.3.2	BENEFITS OF MACHINE LEARNING.	6
6.1	SYSTEM ARCHITECTURE	21
6.2	DATA FLOW DIAGRAM	22
7.1	USE CASE DIGRAM	24
7.2	CLASS DIAGRAM	25
7.3	ACTIVITY DIAGRAM	26
7.4	SEQUENCE DIAGRAM	27
11.1	LEXICAL FEATURES	58
11.2	GET STARTED PAGE	59
11.3	INPUT URL PAGE	60
11.4	URL PREDICTION PAGE	61

CHAPTER 1

INTRODUCTION

Cybersecurity attacks are a serious problem in today's linked society since new and varied threats are always emerging. Phishing, ransomware, malware, and other cyberthreats are noteworthy examples of these. Significant financial damages have been caused by these attacks to individuals, e-commerce businesses, and financial institutions.Our project's objective is to create a real-time cyber threat detection system in reaction to this ever-changing threat scenario. By proactively identifying and mitigating cyber threats as they arise, this approach seeks to improve both individual and organizational security postures online.

What is URL?

A URL, or Uniform Resource Locator, serves as a vital component of navigating the vast landscape of the internet. It functions as a unique address, guiding users to specific resources such as web pages, files, or applications hosted on remote servers. Comprising multiple components, a typical URL includes the protocol, specifying the communication method like HTTP or HTTPS, ensuring secure data transmission. The hostname or IP address identifies the server where the resource resides, facilitating the connection. Additionally, the path delineates the specific location or route within the server's directory structure, streamlining resource retrieval. Often, URLs are augmented with parameters or queries to customize resource access or functionality, enhancing user experience.Furthermore, URLs play a pivotal role in search engine optimization (SEO), influencing the discoverability and visibility of online content. As digital gateways, URLs encapsulate the essence of the Internets interconnections, empowering users to explore,

share, and access information effortlessly.Generally, there are three basic components that make up a legitimate URL.

i) **Protocol:** It specifies the communication protocol used to access the resource, such as HTTP (Hypertext Transfer Protocol) or HTTPS (HTTP Secure).

ii) **Hostname:** It identifies the network location of the resource, typically represented by a domain name or IP address.

iii) **Path:** It denotes the specific location or route to access the resource on the server, often structured hierarchically to organize content.

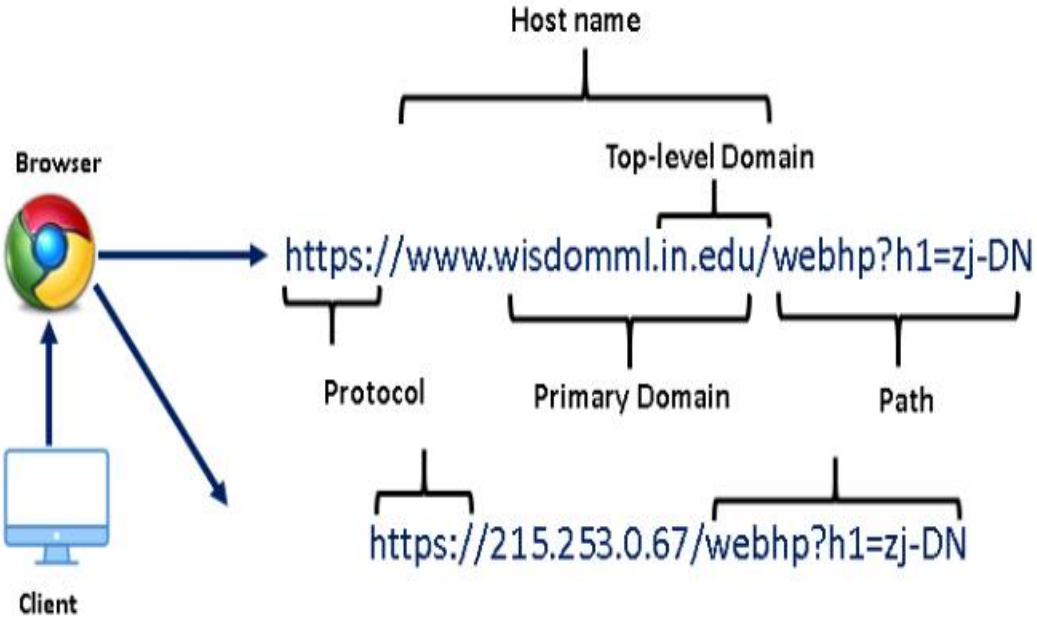


Figure 1. STRUCTURE OF URL

1.1 PROJECT OVERVIEW

Phishing is a common attack on credulous people by making them to disclose their unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishers use the websites which are visually and semantically similar to those real websites. As technology continues to grow, phishing techniques started to progress rapidly and this needs to be prevented by using anti-phishing mechanisms to detect phishing. Machine learning is a powerful tool used to strive against phishing attacks. This paper surveys the features used for detection and detection techniques using machine learning.

1.2 PURPOSE

There are a number of users who purchase products online and make payments through various websites. Many websites ask a user to provide sensitive data such as username, password or credit card details etc. often for malicious reasons. This type of website is known as a phishing website. In order to detect and predict phishing websites, we propose an intelligent, flexible and effective system based on a classification Machine Learning algorithm. We implement classification algorithms and techniques to extract the criteria for phishing data sets to classify their legitimacy. The phishing website can be detected based on some important characteristics like URL and Domain Identity, and security and encryption criteria in the final phishing detection rate. Once a user makes an online transaction, he makes payment through the website. Our system will use a Machine Learning algorithm to detect whether the website is a phishing website or not. This application can be

used by many E-commerce enterprises in order to make the whole transaction process secure.With the help of this system, users can also purchase products online without any hesitation. Admin can add phishing website URL or fake website URL into system where system could access and scan the phishing website and by using algorithm, it will add new suspicious keywords to uses machine learning technique.

1.3 MACHINE LEARNING

Machine Learning (ML) is a subset of **artificial intelligence (AI)** that focuses on the development of algorithms allowing computers to learn and improve from experience without being explicitly programmed. It encompasses a wide range of techniques that enable systems to automatically learn and make predictions or decisions based on data. ML algorithms are trained on vast datasets, extracting valuable insights and facilitating informed decision-making processes. Through iterative learning, these algorithms refine their performance over time, adapting to evolving datasets and scenarios. As organizations increasingly embrace ML technologies, the potential for innovation and transformative impact across industries continues to expand exponentially.

1.3.1 CHARACTERISTICS OF MACHINE LEARNING

Data-driven Approach: Machine learning algorithms rely on large volumes of data to train models and make predictions or decisions.

Adaptability: ML models can adapt and improve over time as they are exposed to new data, allowing them to continuously refine their performance.

Complex Patterns: ML algorithms can identify complex patterns and relationships in data that may not be apparent to human analysts.

Automation: ML enables the automation of tasks that traditionally require human intervention, streamlining processes and improving efficiency.

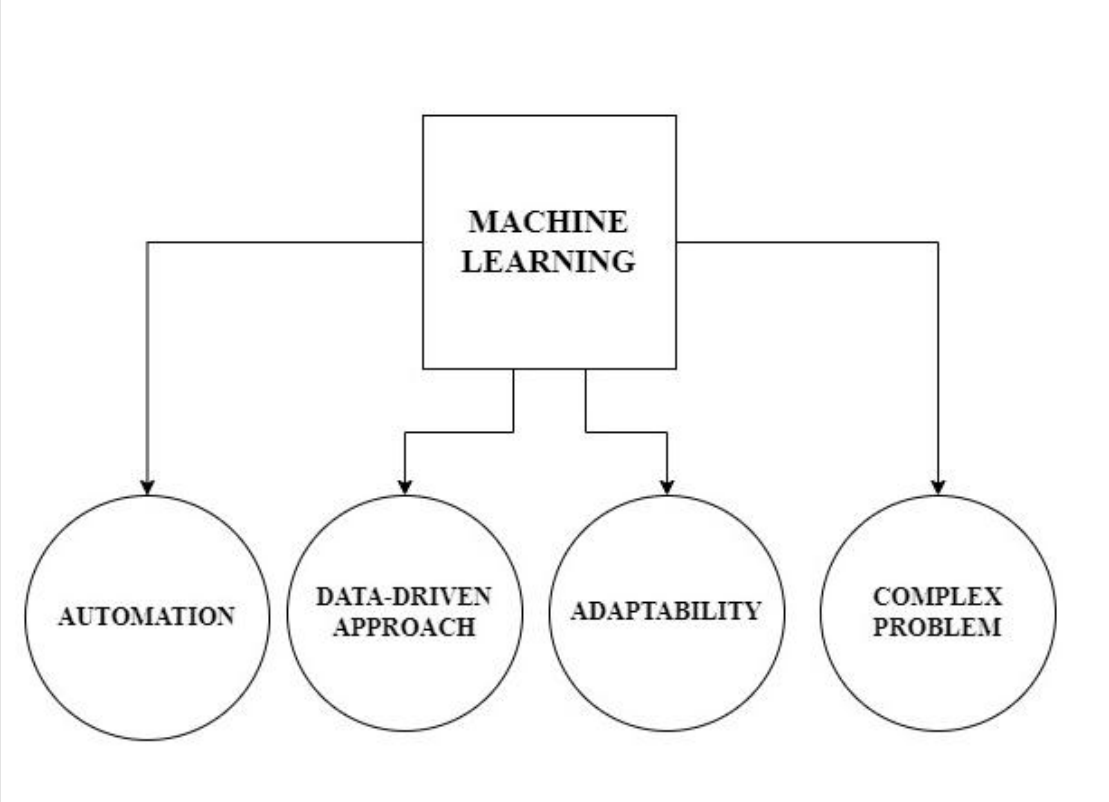


Figure 1.3.1 CHRACTERISTICS OF MACHINE LEARNING

1.3.3 BENEFITS OF MACHINE EARNING

Predictive Analytics: ML algorithms can analyze historical data to make predictions about future events or trends, enabling organizations to anticipate and prepare for various scenarios.

Personalization: ML powers recommendation systems and personalized content delivery, enhancing user experiences by providing tailored recommendations and suggestions.

Fraud Detection: ML algorithms can detect patterns indicative of fraudulent behavior in financial transactions, helping organizations prevent financial losses.

Medical Diagnosis: ML models can analyze medical data to assist in disease diagnosis and treatment planning, potentially improving patient outcomes and health care efficiency

Process Automation: ML enables the automation of repetitive tasks and decision-making processes, freeing up human resources for more complex and creative endeavors.

Enhanced Decision Making: ML empowers organizations to make data-driven decisions by uncovering hidden patterns and insights in large datasets, leading to more informed and effective strategies.

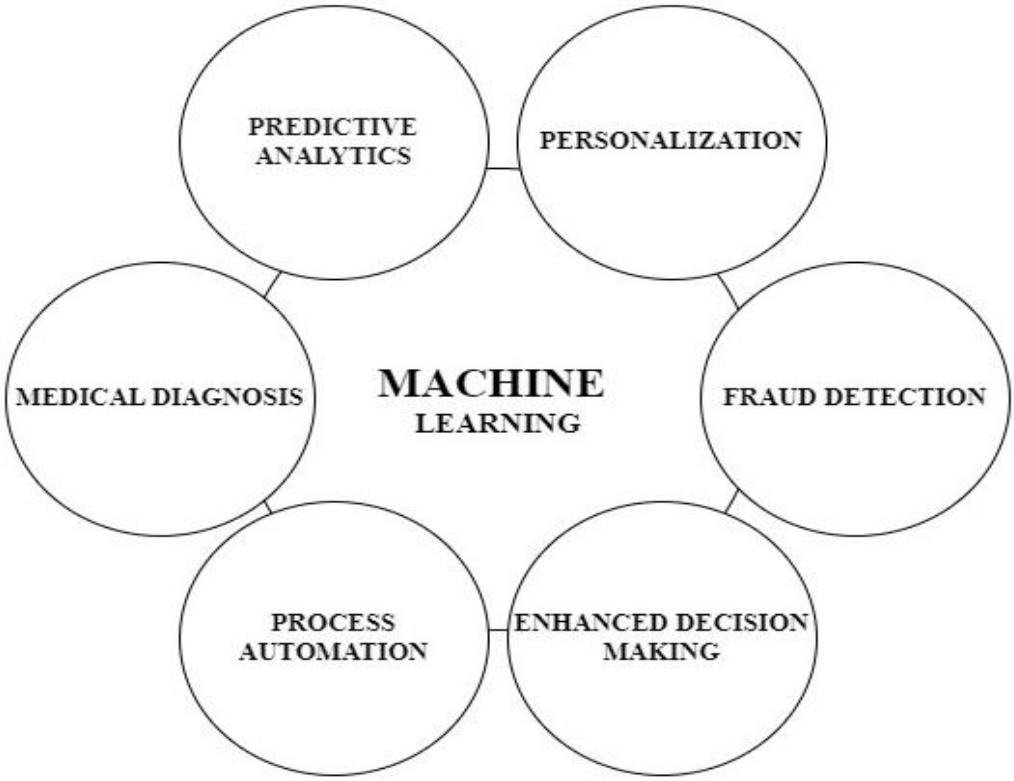


Figure 1.3.2 BENEFITS OF MACHINE LEARNING

CHAPTER 2

LITERATURE SURVEY

Phishing Attack Detection on Text Messages -2022 IEEE Pune Section International Conference.

AUTHORS: Swarangi Uplenchwar , Shilpa Deshpande , Varsha Sawant , Supriya Kelkar ,Prajakta Surve.

As the use of text messages for communication continues to rise, so does the risk of phishing attacks targeting unsuspecting users. Phishing attacks via text messages, also known as SMS phishing or smishing, pose a significant threat to cybersecurity, aiming to deceive individuals into divulging sensitive information. In this paper, we propose a novel approach for detecting phishing attacks on text messages using machine learning techniques. Through experimental evaluation and validation, we showcase the robustness and efficiency of our proposed solution in detecting and mitigating phishing attacks on text messages. Our research contributes to the ongoing efforts to enhance cybersecurity in the digital age, offering a practical and scalable approach for protecting users against evolving threats in mobile communication channels. **[7]**

Protecting Users Against Phishing Attacks with AntiPhish.

AUTHORS: Engin Kirda and Christopher Kruegel.

Phishing attacks remain a prevalent threat in the realm of cybersecurity, aiming to deceive users into disclosing sensitive information through fraudulent means. In response to this ongoing challenge, Engin Kirda and Christopher Kruegel of the Technical

University of Vienna have developed "AntiPhish," a robust solution designed to protect users against phishing attacks effectively.

Leveraging innovative techniques and advanced algorithms, AntiPhish employs machine learning and behavioral analysis to detect and thwart phishing attempts across various online platforms. By continuously analyzing user behavior and characteristics of phishing emails, websites, and other malicious content, AntiPhish provides proactive defense mechanisms to safeguard users' personal and financial information. This paper outlines the key features and methodologies of AntiPhish, highlighting its efficacy in mitigating the risks posed by phishing attacks and contributing to the advancement of cyber security technologies. **[9]**

The Emergence Threat of Phishing attack and the detection -2021 International Conference on automation , control and mechatronics for industry.

AUTHORS: Sadia Parvin Ripa,Fahmida Islam,Mohammad Arifuzzaman.

The 2021 International Conference on Automation, Control, and Mechatronics for Industry served as a platform for addressing the emerging threat of phishing attacks and exploring innovative detection methodologies. Phishing attacks, characterized by deceptive attempts to acquire sensitive information, pose significant risks to individuals and organizations alike. These attacks exploit human vulnerabilities and technological weaknesses, making detection a critical aspect of cybersecurity efforts.

This abstract highlights discussions and presentations from the conference focused on the detection of phishing attacks. Various detection techniques were explored, including machine learning algorithms, email filtering, URL analysis, user behavior analysis, and real-time threat intelligence.

Moreover, user behavior analysis emerged as a valuable approach for detecting anomalies indicative of phishing attacks, leveraging insights into user interactions with digital content to identify potential threats. Additionally, the integration of real-time threat intelligence feeds was highlighted as a crucial strategy for staying ahead of evolving phishing tactics and campaigns.

Overall, the discussions and presentations at the 2021 International Conference on Automation, Control, and Mechatronics for Industry underscored the importance of proactive measures and collaborative efforts in combating the growing threat of phishing attacks. By leveraging advanced detection techniques and staying abreast of emerging threats, organizations can better protect themselves and their stakeholders from the detrimental effects of phishing attacks in today's interconnected digital landscape. **[11]**

A Survey on the Detection of Phishing Websites-2023 University of Pavia Through the CRUI-CARE Agreement.

AUTHORS: Rasha Zieni,Luisa Massari And Carla Calzarossa.

Phishing is a security threat with serious effects on individuals as well as on the targeted brands. Although this threat has been around for

quite a long time, it is still very active and successful. In fact, the tactics used by attackers have been evolving continuously in the years to make the attacks more convincing and effective. In this context, phishing detection is of primary importance. The literature offers many diverse solutions that cope with this issue and in particular with the detection of phishing websites. This paper provides a broad and comprehensive review of the state of the art in this field by discussing the main challenges and findings. More specifically, the discussion is centered around three important categories of detection approaches, namely, list-based, similarity-based and machine learning-based. For each category we describe the detection methods proposed in the literature together with the datasets considered for their assessment and we discuss some research gaps that need to be filled.**[8]**

Modeling and Preventing Phishing detection system for e - banking using fuzzy data mining.

Authors: Markus Jakobsson, Aburrous M, Hossain M.A. , Dahal K. , Thabatah, F.

The modeling of phishing attacks, emphasizing the importance of understanding the underlying psychological and technical mechanisms employed by attackers. By examining the cognitive processes and decision-making factors involved in phishing victimization, Jakobsson provides a framework for developing more effective prevention strategies.

On the other hand, Aburrous et al.'s research presents a practical approach to phishing detection in the context of e-banking systems. Their proposed system utilizes fuzzy data mining techniques to analyze user behavior and identify patterns indicative of phishing attempts. By

incorporating fuzzy logic to handle uncertain and imprecise data, the system enhances the accuracy and reliability of phishing detection in real-world scenarios.

Both contributions underscore the multifaceted nature of phishing attacks and the need for interdisciplinary approaches to combat this evolving threat. By combining insights from psychology, cybersecurity, and data mining, researchers and practitioners can develop more robust models and prevention mechanisms to safeguard against phishing attacks effectively.This abstract highlights the significance of ongoing research efforts in understanding, modeling, and preventing phishing attacks, ultimately contributing to the advancement of online security in an increasingly interconnected digital landscape.**[6]**

Phishing email detection based on structural properties.

AUTHORS: M.Chandrasekaran, et al.,

The emergence of phishing emails as a prominent cybersecurity threat has necessitated innovative approaches for their detection and mitigation. In this abstract, we examine the research presented by M. Chandrasekaran et al. at the New York State Cyber Security Conference (NYS) in 2006, focusing on phishing email detection based on structural properties.

The authors propose a novel approach that considers the underlying structure of phishing emails.By examining properties such as email header information, embedded links, and HTML structure, the proposed detection method aims to differentiate between legitimate emails and phishing attempts. This structural analysis provides valuable insights into the origins and intentions of suspicious emails, enabling more accurate

identification and classification.The research presented at the New York State Cyber Security Conference underscores the importance of incorporating structural properties into phishing email detection algorithms. Furthermore, the insights shared at the conference highlight the need for ongoing collaboration and innovation in the field of cybersecurity to stay ahead of evolving threats. By leveraging research findings and best practices, organizations can better protect themselves and their stakeholders from the detrimental effects of phishing attacks in today's digital landscape.**[3]**

Learning to Detect Phishing Emails" Ian Fette School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213.

Phishing emails represent a persistent cybersecurity threat, exploiting human vulnerabilities to deceive individuals and organizations. In this abstract, we delve into the research conducted by Ian Fette, Norman Sadeh, and Anthony Tomasic from the School of Computer Science at Carnegie Mellon University, focusing on the development of techniques for detecting phishing emails.The researchers' work centers on leveraging machine learning algorithms to automatically identify and classify phishing emails. By training models on large datasets of known phishing emails, they aim to enable automated detection of suspicious content, thereby enhancing cybersecurity defenses.Key aspects of their research include feature engineering, where relevant attributes of phishing emails are extracted and utilized as input to machine learning models.**[4]**

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Existing phishing detection systems incorporate a variety of techniques and methodologies to identify and mitigate the threat posed by phishing emails. These systems often leverage rule-based mechanisms, machine learning algorithms, email filtering techniques, user behavior analysis, and real-time threat intelligence feeds. Rule-based systems employ predefined rules or heuristics to flag suspicious emails based on criteria such as sender domain mismatches or suspicious URLs. Machine learning models are trained on labeled datasets to automatically classify emails as legitimate or phishing based on learned patterns and features. Email filtering mechanisms are utilized to block or flag suspicious emails before they reach users' inboxes, while user behavior analysis monitors interactions to detect anomalies indicative of phishing attempts. Integration with real-time threat intelligence feeds ensures that systems remain updated on emerging threats, allowing for proactive identification and mitigation. Continuous monitoring and updates are essential for adapting to evolving threats, ensuring that detection systems remain effective in safeguarding users and organizations against phishing attacks.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

False Positives: One common issue is the generation of false positives, where legitimate emails are incorrectly flagged as phishing attempts. This can lead to user frustration and decreased trust in the detection system.

False Negatives: Conversely, false negatives occur when phishing emails are not detected by the system, allowing malicious content to reach users'

inboxes. This poses a significant security risk and undermines the effectiveness of the detection system.

Over-reliance on Static Rules: Rule-based systems may rely too heavily on static rules or heuristics, making them less effective against sophisticated phishing attacks that evolve over time. Attackers can easily bypass these rules by altering their tactics or using obfuscation techniques.

Limited Scalability: Some detection methods, such as manual rule creation or signature-based approaches, may lack scalability, particularly in large organizations with high email volumes. As a result, these systems may struggle to keep pace with the growing volume and complexity of phishing attacks.

Resource Intensive: Certain detection techniques, such as machine learning algorithms, require significant computational resources and expertise to implement and maintain. This can pose challenges for organizations with limited resources or technical capabilities.

Privacy Concerns: Phishing detection systems often involve the analysis of email content and user behavior, raising privacy concerns regarding the collection and processing of sensitive information.

Evolution of Phishing Tactics: Phishing attacks continue to evolve, with attackers employing new tactics and techniques to evade detection. Existing systems may struggle to keep pace with these changes, requiring constant updates and refinement to remain effective.

Cost: Implementing and maintaining effective phishing detection systems can incur significant costs, including software licenses, hardware infrastructure, and ongoing maintenance and support. This can be prohibitive for smaller organizations with limited budgets.

3.2 PROPOSED SYSTEM

The system aims to enhance phishing detection through the utilization of machine learning techniques.

Data Collection and Preprocessing: Emphasizes the importance of collecting a comprehensive dataset and preprocessing it to ensure suitability for model training.

Lexical Features: Identifies key features extracted from the dataset that serve as indicators for classifying websites.

Model Selection: Logistic Regression algorithm is chosen for its simplicity, interpretability, and effectiveness in binary classification tasks.

Model Evaluation: Metrics such as accuracy, precision, recall, and F1-score are utilized to evaluate the performance of the trained model.

Deployment: The trained model is deployed for real-time prediction, enabling users to assess website legitimacy on-the-fly.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

Enhanced Phishing Detection: Leveraging machine learning techniques to accurately classify websites as legitimate or phishing, thereby strengthening cybersecurity.

Automation: Automating the process of website classification, reducing the manual effort required for identifying potential phishing threats.

Scalability: The system can scale to analyze large volumes of websites efficiently, providing real-time protection against phishing attacks.

Adaptability: The model can be continuously updated with new data to adapt to evolving phishing tactics and emerging threats.

CHAPTER 4

PROJECT DESCRIPTION

4.1 INPUT DESIGN AND OUTPUT DESIGN

4.1.1 INPUT DESIGN

The input design is the link between the information system and the user. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

4.1.2 OBJECTIVES

URL Input Field: The webpage provides a text input field where users can enter the URL they want to verify for phishing. This input field allows users to input the URL they wish to check.

URL Extraction: Once the user submits the URL through the input field, the system extracts this input and passes it to the backend for processing. This step involves capturing the URL entered by the user.

URL Prediction Button: A button labeled "Predict" is provided next to the URL input field. When clicked, this button triggers the prediction process, where the system analyzes the input URL to determine if it is a phishing website or not.

4.1.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

Phishing Detection Results: After the prediction process is completed, the system displays the results of the phishing detection. This output section informs the user about the likelihood of the input URL being a phishing website. The output could include one of the following messages

- **Legitimate Website:**"You are safe! This is a Legitimate Website" If the URL is determined to be safe and not associated with phishing.

- **Phishing Website:**"You are on the wrong site. Be cautious!": If the URL is flagged as potentially phishing.

Awareness to User: In addition to displaying the phishing detection results, the output section aims to raise awareness among users about the importance of verifying website authenticity to protect their sensitive information. This awareness message emphasizes the significance of staying vigilant while browsing online and encourages users to be cautious when accessing unfamiliar websites.

CHAPTER 5

REQUIREMENT ANALYSIS

5.1 FUNCTIONAL REQUIREMENTS

Functional requirements specify the behavior that a system must exhibit to meet user needs and achieve its intended purpose. They outline the specific tasks, functions, or features that the system should perform, focusing on the actions it must take and the outcomes it must produce. These requirements define the system's capabilities and functionalities in detail, guiding the development process by providing a clear understanding of what the system should do.

Checking URL: Users should be able to check if a URL might be a phishing URL.

Copying URL: Users can copy suspected URLs and paste them into the search engine.

URL Extraction: After pasting the URL into the search engine, it should extract all information about the URL.

Data Processing: The search engine should compare the URL with a given dataset using machine learning algorithms like Logistic Regression and Decision Trees.

Predicting: The search engine should predict the result of the given URL and display whether it is negative or positive

5.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements specify how the system should behave, rather than what it should do. They encompass attributes such as performance, usability, reliability, security, and scalability. These requirements ensure that the system meets certain quality standards and user expectations.

Usability: The website should be easy to understand, and users should not face difficulties in finding the search engine.

Security: The website should prioritize security and not request any device permissions, as it is solely for security processes.

Reliability: All data processing and predictions should be hidden from end-users. The results should be accurate, and there should be no wrong predictions.

Performance: When using datasets with Python and machine learning algorithms, predictions should be faster compared to using a database with Python.

Scalability: The system should be able to handle multiple URL searches simultaneously by different users.

CHAPTER 6

SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

The architecture of a Cyber Threat Detection Platform Using Machine Learning.The system consists entities : User , Dataset , Lexical features , Model Building , Deployment.

User Interface: The user interacts with the system through a user interface, which could be a web application, command-line interface (CLI), or graphical user interface (GUI). The user interface allows users to input data, configure the system, and view the results of threat detection.

Dataset Management: The dataset management component is responsible for acquiring, storing, and preprocessing the data used for training and testing machine learning models. It may include functionalities for data ingestion, data cleaning, feature extraction, and data storage.

Lexical Features Extraction: This component focuses on extracting lexical features from the raw data. Lexical features may include patterns, keywords, n-grams, or other linguistic attributes that can help identify patterns indicative of cyber threats. Techniques such as tokenization, stemming, and vectorization may be employed to extract and represent these features.

Model Building: The model building component involves training machine learning models using the extracted features and labeled data. Various machine learning algorithms such as logistic regression, decision

trees, random forests, support vector machines, or neural networks may be employed to build predictive models. Hyperparameter tuning and model selection techniques may also be part of this component to optimize model performance.

Deployment: Once trained, the models need to be deployed into a production environment where they can make real-time predictions on incoming data. This component involves integrating the trained models into the overall threat detection system and providing APIs or other interfaces for making predictions.

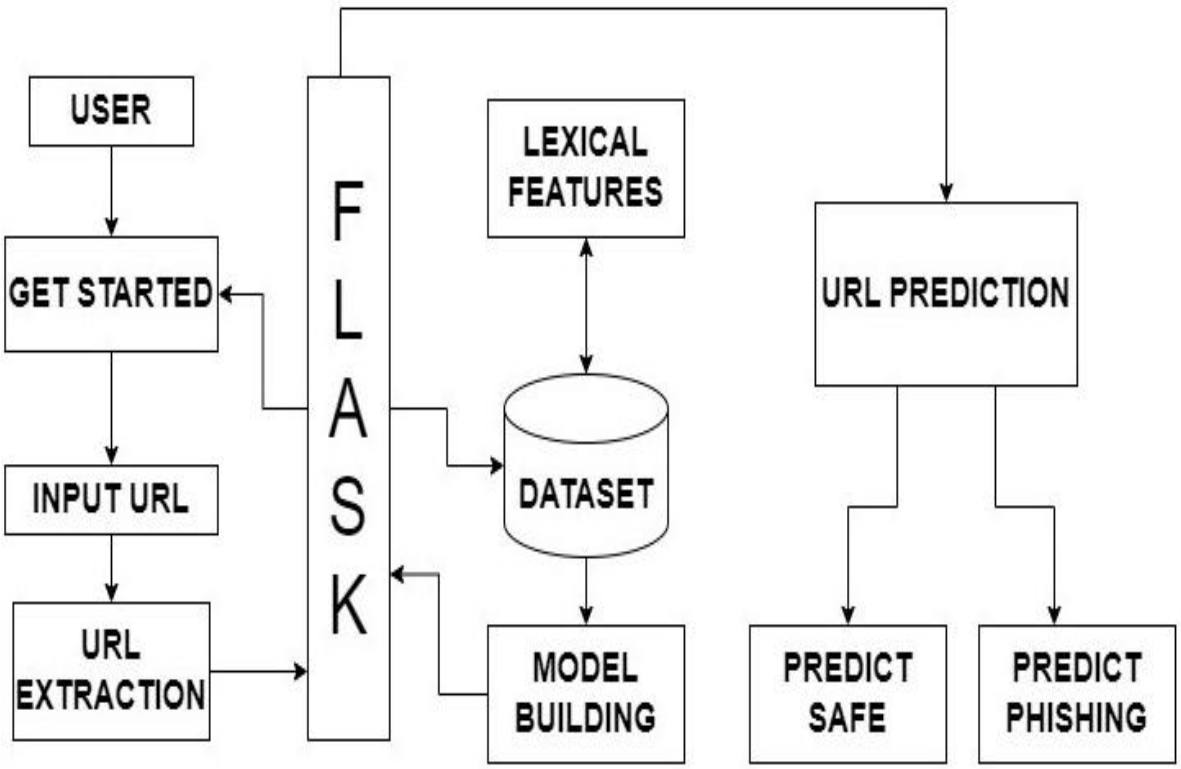


Figure 6.1 SYSTEM ARCHITECTURE

6.2 DATA FLOW DIAGRAM

USER: In this User module,individuals first obtain a search token to access the system's functionalities. Once authenticated, users input a URL into the system interface. The system then proceeds to extract relevant information from the URL through a process called URL extraction. Subsequently, the extracted data is passed through the system's URL prediction mechanism. This mechanism employs machine learning algorithms to determine the safety of the provided URL. Based on the prediction outcome, users are promptly informed whether the URL is deemed safe for browsing or flagged as potentially unsafe.

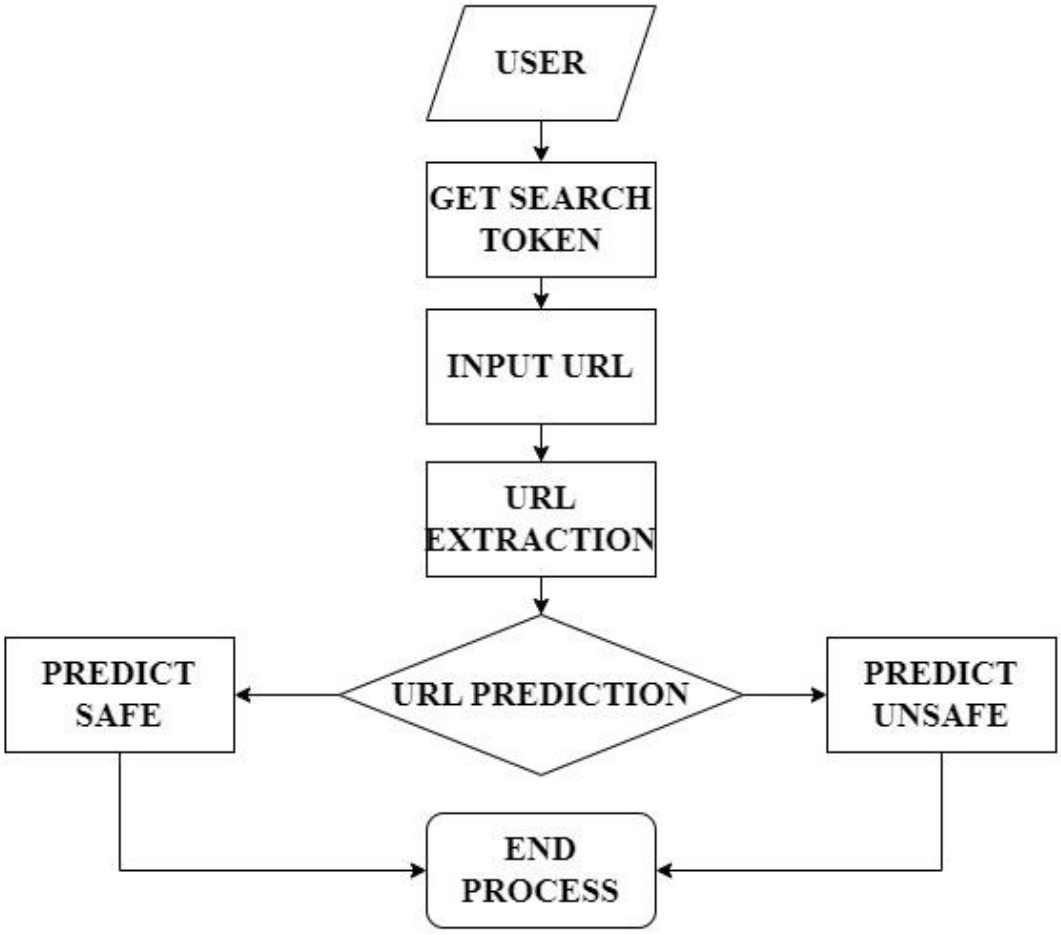


Figure 6.2 DATA FLOW DIAGRAM

CHAPTER 7

UML DIAGRAM

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

- ❖ Provide extendibility and specialization mechanisms to extend the core concepts.

7.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

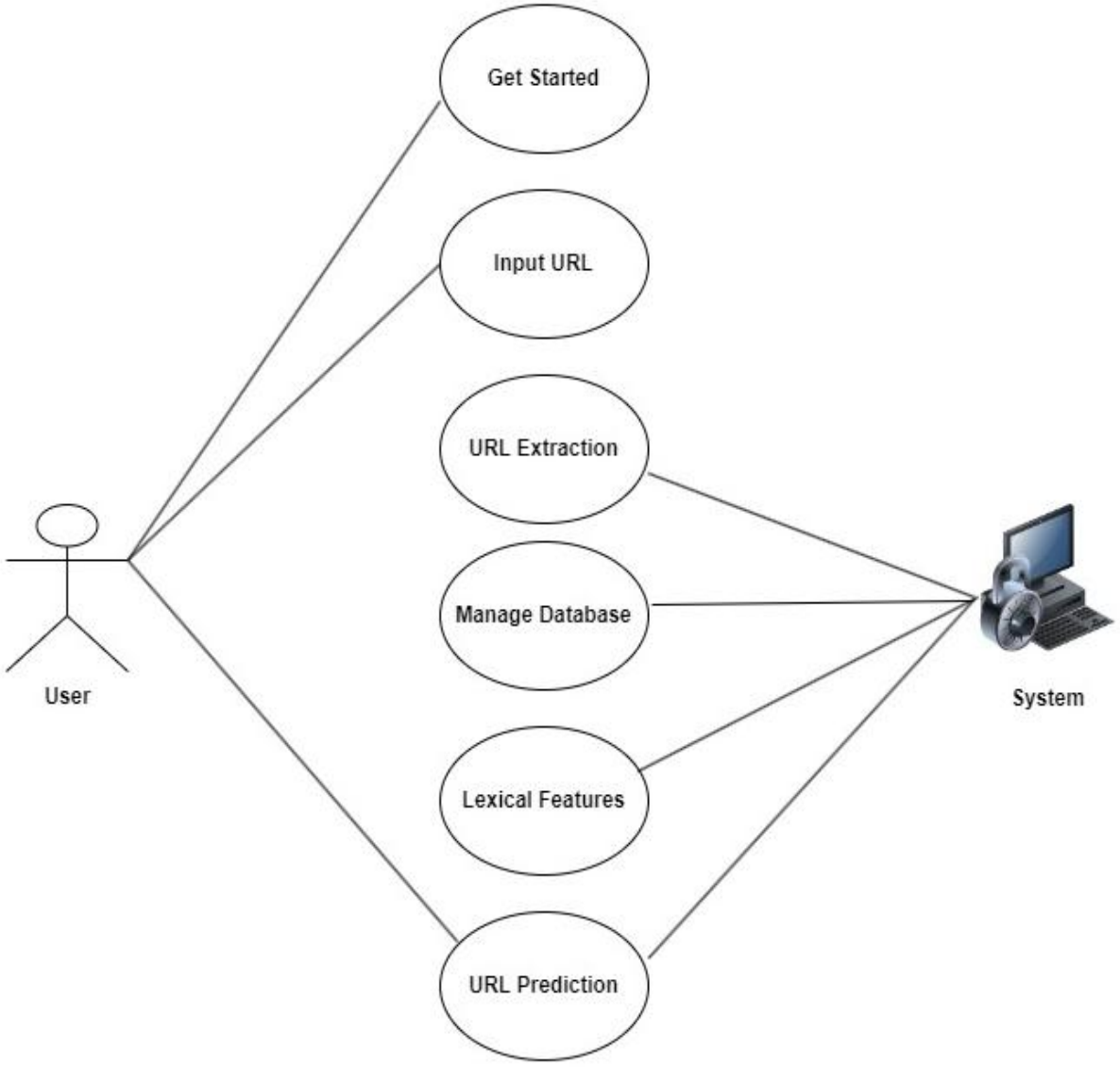


Figure 7.1 USE CASE DIAGRAM

7.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

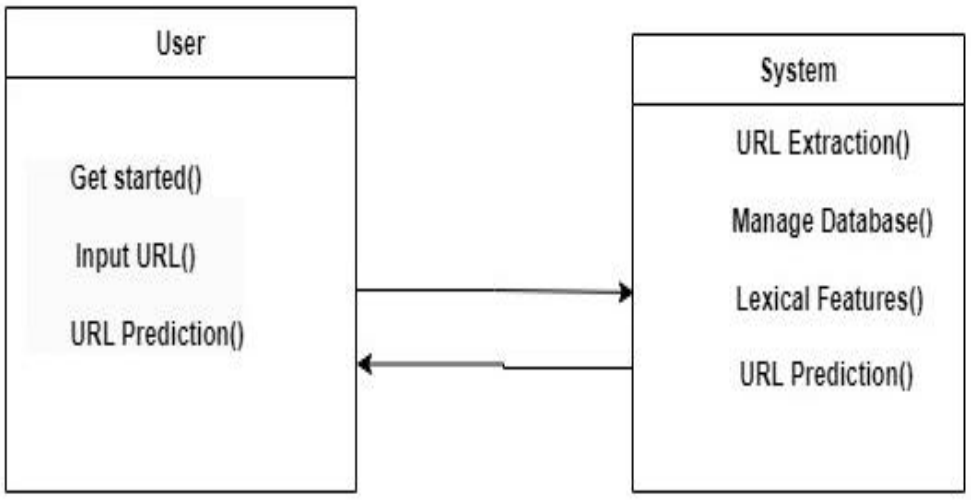


Figure 7.2 CLASS DIAGRAM

7.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

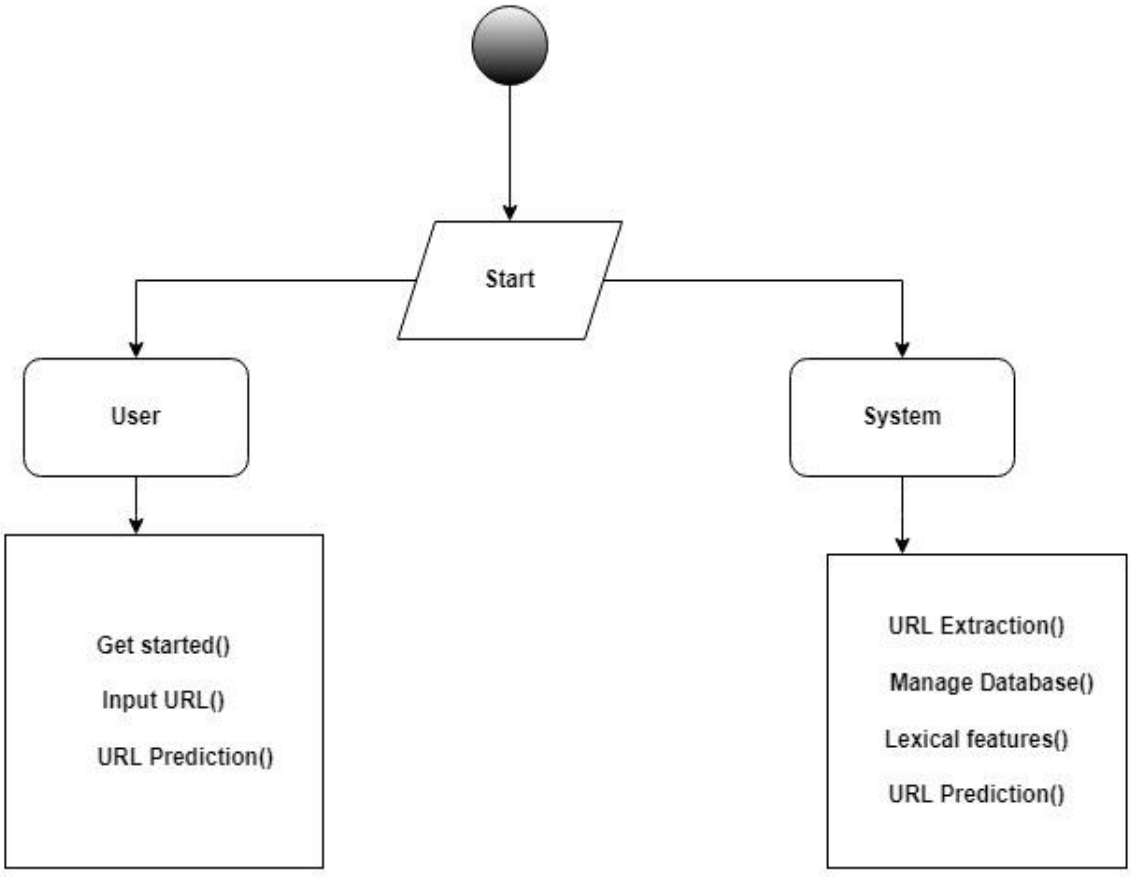


Figure 7.3 ACTIVITY DIAGRAM

7.4 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

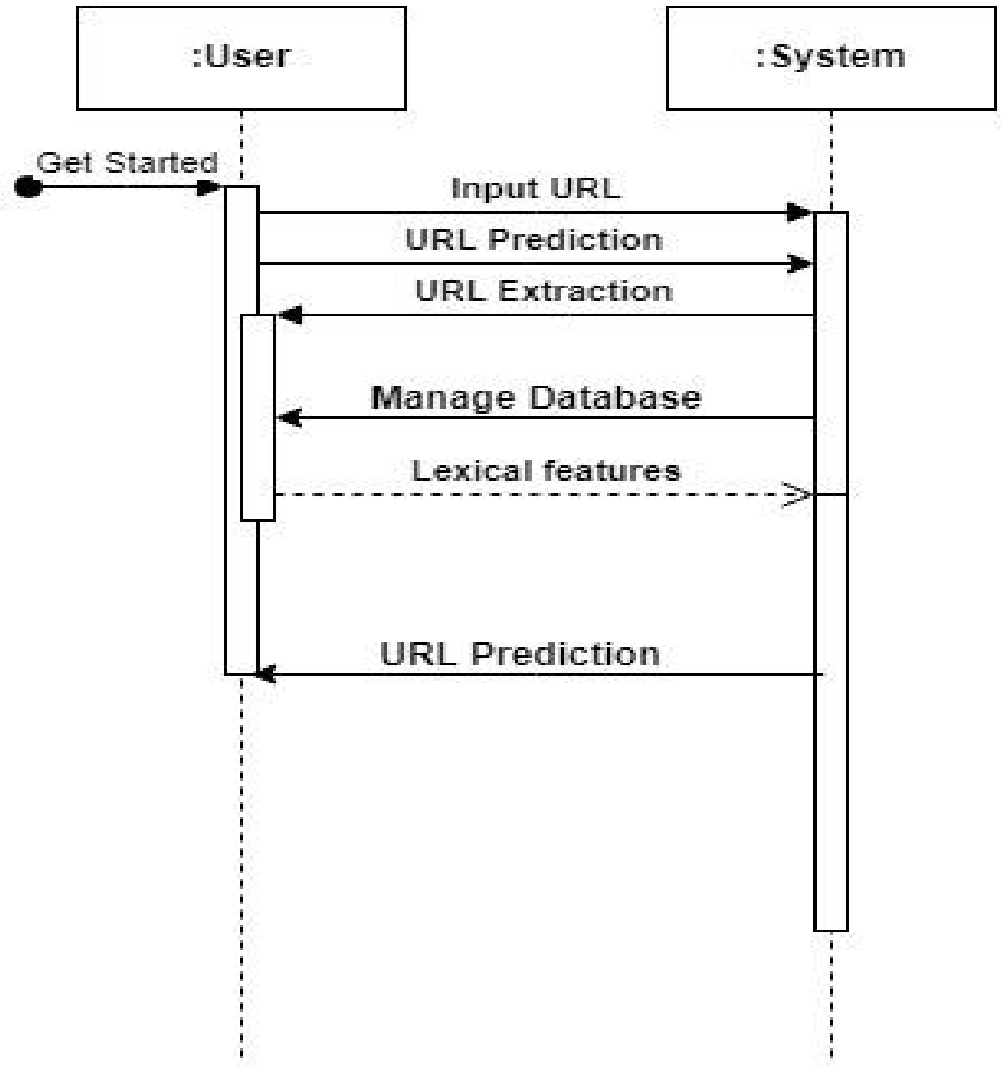


Figure 7.4 SEQUENCE DIAGRAM

CHAPTER 8

MODULES DESCRIPTION

8.1 MODULES SPLITUP

1. DATASET COLLECTION AND PREPROCESSING
2. LEXICAL FEATURES
3. MODEL BUILDING AND EXTRACTION
4. DEPLOYMENT

8.1.1 DATASET COLLECTION AND PREPROCESSING

We collected 38,800 URL from the open source platform and stored in a dataset containing a total of 38,800 URLs categorized into safe and phishing categories. To insights into the dataset's composition, we performed an analysis to examine the distribution of URLs based on their labels 19,400 labeled as safe and 19,400 as phishing.

PHISHING URL: A phishing URL is a deceptive web address crafted by cyber criminals to mimic legitimate sites, aiming to trick users into revealing sensitive information like passwords or financial data. These URLs are often distributed through email or social media, exploiting users' trust to steal personal information or distribute malware. Users should verify URLs and avoid clicking suspicious links to protect themselves from phishing attacks

LEGITIMATE URL: A legitimate URL denotes a web address affiliated with reputable and trustworthy entities, ensuring secure access to genuine content and services

Subsequently, to ensure data integrity and eliminate redundancy, duplicate URLs were removed from the dataset. This preprocessing step resulted in a refined dataset consisting of 11,000 unique URLs, ready for further analysis and model development. Finally, a preprocessed dataset containing 11,000 URLs was selected to ensure a balanced representation of both safe and phishing instances, laying the groundwork for robust machine learning model training and evaluation.

8.1.2 LEXICAL FEATURES

Lexical features in the context of web phishing detection refer to characteristics extracted from the textual components of URLs, such as domain names, path segments, and query parameters. These features provide valuable information for machine learning algorithms to analyze and classify URLs as either legitimate or phishing based on their structural patterns and syntax.

8.1.2.1 FEATURES

We have implemented python program to extract features from URL. Below are the features that we have extracted for detection of phishing URLs.

Presence of IP address in URL: If IP address present in URL then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that the attacker is trying to steal sensitive Information.

Presence of @ symbol in URL: If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add a special symbol @ in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

Number of dots in Hostname: Phishing URLs have any dots in URL. For example http://shop.fun.amazon.phishing.com, in this URL phishing.com is an actual domain name, whereas use of "amazon" word is to trick users to click on it. Average number of dots in benign URLs is if the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.

Prefix or Suffix separated by (-) to domain: If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-)to the domain name so that users feel that they are dealing with a legitimate webpage. For example Actual site is http://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.com to confuse the innocent users.

URL redirection: If "//" presents a URL path then feature is set to 1 else to 0.The existence of "//" within the URL path means that the user will be redirected to another website.

HTTPS token in URL: If HTTPS token is present in URL then the feature is set to else to 0. Phishers may add the "HTTPS" token to the domain part of a URL in order to trick users. For example, http://https-www-paypal-it-mpp,home.soft-hair.com.

Information submission to Email: Phisher might use "mail()" or "mailto:" functions to redirect the user's information to his personal email[4]. If such functions are present in the URL then feature is set to 1 else to 0.

URL Shortening Services "TinyURL" : TinyURL service allows phisher to hide long phishing URL by making it short. The goal is to

redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0.

Length of Host name: Average length of the benign URLs is found to be a 25, if URL's length is greater than 25 then the feature is set to 1 else to 0

Presence of sensitive words in URL: Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that foundin many phishing URLs :- 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal','password', 'username', etc;

Number of slash in URL: The number of slashes in benign URLs is found to be a 5; if number of slashes in URL is greater than 5 then the feature is set to 1 else to 0.

Presence of Unicode in URL: Phishers can make a use of Unicode characters in URL to trick users to click on it. For example the domain "xn--80ak6aa92e.com" is equivalent to" apple.com Visible URL to user is "apple.com" but after clicking on this URL, user will visit "xn--80ak6aa92e.com" which is a phishing site.

Age of SSL Certificate: The existence of HTTPS is very important in giving the impression of website legitimacy . But minimum age of the SSL certificate of benign website is between 1 year to 2 years.

URL of Anchor: We have extracted this feature by crawling the source code on the URL. URL of the anchor is defined by <a> tag. If the <a>tag has a maximum number of hyperlinks which are from the other domain then the feature is set to 1 else to 0.

IFRAME: We have extracted this feature by crawling the source code of the URL. This tag is used to add another web page into existing main

webpage. Phishers can make use of the "iframe" tag and make it invisible i.e. without frame orders. Since border of inserted Webpage is invisible, user seems that the inserted web page is also the part of the main web page and can enter sensitive information.

Website Rank: We extracted the ranking of websites and compare it with the first One hundred thousand websites of Alexa database. If rank of the website is greater than 10,0000 then feature is set to 1 else to 0.

8.1.3 MODEL BUILDING AND EXTRACTION

In the process of model building for web phishing detection, we first split our dataset into training and testing sets . This step ensures that our model is trained on a portion of the data and evaluated on unseen data to assess its generalization capability. Subsequently, we employ the Logistic Regression algorithm, a commonly used technique for binary classification tasks due to its simplicity and interpretability. After fitting the Logistic Regression model to the training data, we make predictions on the test set and calculate the accuracy score to evaluate the performance of our model. Finally, we print the accuracy score to assess how well the Logistic Regression model can distinguish between legitimate and phishing URLs, providing valuable insights into the effectiveness of our web phishing detection system.

8.1.4 DEPLOYMENT

Deployment involves the transition of a software application from development to production, ensuring it is accessible and operational for end-users. This process typically includes installation, configuration, and testing to ensure smooth operation in the production environment.

8.1.4.1 FLASK

Flask is a lightweight and versatile web framework written in Python, designed to create web applications quickly and with minimal hassle. It provides tools and libraries for building web applications and APIs, allowing developers to handle tasks such as URL routing, template rendering, and request handling efficiently. Flask follows a simple and modular design, making it easy to get started for beginners while offering flexibility and scalability for more complex projects.

In this deployment module, a Flask web application is created to provide a user interface for phishing website detection. Upon receiving a URL input from the user through a form submission, the application utilizes a trained machine learning model stored in a pickled file to predict whether the given URL is safe or potentially phishing. The inputScript file is imported to analyze the URL before passing it to the model for prediction. Depending on the prediction outcome, the application returns a message indicating whether the URL is deemed safe or unsafe. Finally, the application is run on a local server using the Flask run method, making it accessible for testing and use.

CHAPTER 9

TECHNIQUES AND ALGORITHMS

9.1 ALGORITHMS

9.1.1 LOGISTIC REGRESSION

Logistic regression stands as a cornerstone in the realm of statistical modeling, especially in binary classification tasks where the objective is to predict the probability of an observation belonging to one of two classes. Its simplicity, interpretability, and effectiveness make it a widely employed tool across various domains, from finance to healthcare and beyond.

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. Logistic regression is a statistical algorithm which analyze the relationship between two data factors. The article explores the fundamentals of logistic regression, it’s types and implementations.

Mathematically, logistic regression combines linear regression with the logistic function. It assumes a linear relationship between the independent variables (features) and the log-odds of the dependent variable (the binary outcome). The log-odds, also known as the logit function, represent the natural logarithm of the odds ratio, which is the ratio of the probability of the event happening to the probability of it not happening.

How does Logistic Regression work?

The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.

Let the independent input features be:

[Tex]X = \begin{bmatrix} x_{11} & … & x_{1m}\\ x_{21} & … & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & … & x_{nm} \end{bmatrix}[/Tex]

and the dependent variable is Y having only binary value i.e. 0 or 1.

[Tex]Y = \begin{cases} 0 & \text{ if } Class\;1 \\ 1 & \text{ if } Class\;2 \end{cases} [/Tex]

then, apply the multi-linear function to the input variables X.

[Tex]z = \left(\sum_{i=1}^{n} w_{i}x_{i}\right) + b [/Tex]

Here [Tex]x_i [/Tex] is the ith observation of X, [Tex]w_i = [w_1, w_2, w_3, \cdots,w_m] [/Tex] is the weights or Coefficient, and b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

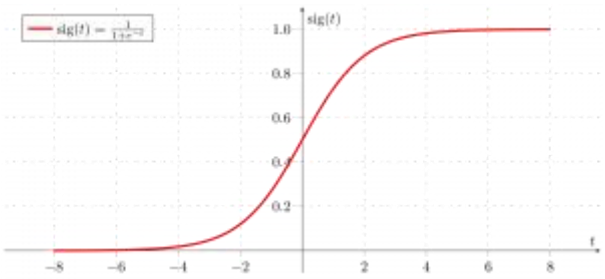
[Tex]z = w\cdot X +b [/Tex]

whatever we discussed above is the linear regression.

Sigmoid Function:

Now we use the sigmoid function where the input will be z and we find the probability between 0 and 1. i.e. predicted y.

[Tex]\sigma(z) = \frac{1}{1-e^{-z}} [/Tex]



Sigmoid function

As shown above, the figure sigmoid function converts the continuous variable data into the probability i.e. between 0 and 1.

- [Tex]\sigma(z) [/Tex] tends towards 1as [Tex]z\rightarrow\infty [/Tex]
- [Tex]\sigma(z) [/Tex] tends towards 0 as [Tex]z\rightarrow-\infty [/Tex]
- [Tex]\sigma(z) [/Tex] is always bounded between 0 and 1

where the probability of being a class can be measured as:

[Tex]P(y=1) = \sigma(z) \\ P(y=0) = 1-\sigma(z) [/Tex]

Logistic Regression Equation:

The logistic regression algorithm is a type of regression analysis used for predicting the probability of a categorical dependent variable. It's commonly used for binary classification problems, but can be extended to handle multiple classes as well. The logistic regression equation can be expressed as:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n)}}$$

Where:

• P(Y=1|X)P(Y=1|X) is the probability that the dependent variable Y is equal to 1 given the independent variables X.

• e is the base of the natural logarithm.

• β0 is the intercept.

Key Points:

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.

It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

9.1.2 RANDOM FOREST ALGORITHM

Random Forest is a versatile ensemble learning method used for both classification and regression tasks. It belongs to the family of decision tree-based algorithms and is renowned for its robustness and high performance across various domains.

At its core, Random Forest constructs multiple decision trees during the training phase. Each decision tree is trained on a random subset of the training data and a random subset of the features. This process introduces randomness and diversity into the individual trees, helping to mitigate overfitting and improve the generalization ability of the model.

During inference, the Random Forest aggregates the predictions of all individual trees to make the final prediction. For classification tasks, it typically outputs the mode (most frequent class) predicted by the constituent trees. In contrast, for regression tasks, it computes the average prediction across all trees.

The collective wisdom of multiple decision trees in a Random Forest yields several benefits. It tends to be more robust to noise and outliers compared to individual decision trees. Additionally, Random Forests can automatically handle feature interactions and nonlinear relationships in the data, making them suitable for complex real-world datasets.

Moreover, Random Forests provide insights into feature importance, allowing users to identify the most influential features in the prediction process. This information can be valuable for feature selection and model interpretation.

Overall, Random Forest stands as a powerful and popular machine learning algorithm due to its simplicity, scalability, and outstanding performance across a wide range of tasks, from classification and regression to anomaly detection and feature selection.

Ensemble Learning:

Ensemble learning techniques like Random Forest harness the collective intelligence of multiple base models to enhance predictive performance and overall robustness.

Random Forest specifically employs a technique called bagging (Bootstrap Aggregating) to train each decision tree in the ensemble. Bagging involves randomly sampling subsets of the training data, with replacement, to train each individual tree. This random sampling introduces diversity into the training process, as each tree sees a slightly different subset of the data.

By training each decision tree on a different subset of the data, Random Forest mitigates the risk of overfitting that can occur with single decision trees. Each tree learns to capture different aspects of the data, and the ensemble combines their predictions through averaging (for regression) or voting (for classification), leading to a more robust and generalizable model.

Moreover, bagging helps to reduce variance by averaging over multiple noisy models, thereby improving the stability and reliability of the final predictions. Additionally, it enables parallelization of the training process, making Random Forest efficient and scalable for large datasets.

Overall, by leveraging bagging and combining the predictions of multiple decision trees, Random Forest achieves remarkable performance across various machine learning tasks, making it a popular choice for practitioners seeking robust and accurate models..

Decision Trees:

Decision trees are hierarchical models that recursively partition the feature space based on the values of input features, ultimately leading to predictions at the leaf nodes. Each internal node represents a decision based on a feature, and each leaf node corresponds to a prediction.

Random Forest leverages the power of decision trees by creating a collection (ensemble) of them. However, it adds an element of randomness to each tree's construction. Instead of using all available features for each split, Random Forest selects a random subset of features at each node. This process introduces diversity among the trees, ensuring that they don't all learn the same patterns from the data.

Moreover, Random Forest also applies bagging (Bootstrap Aggregating) during training, where each tree is trained on a random sample of the training data with replacement. This further enhances the diversity among the trees in the ensemble.

By combining a diverse set of decision trees, each trained on a different subset of features and data points, Random Forest creates a robust and accurate model that is less prone to overfitting and more stable compared to individual decision trees.In essence, Random Forest harnesses the collective wisdom of multiple decision trees, each offering its unique perspective on the data, to create a more reliable and robust predictive model.

WORKING OF RFC:

- **Bootstrapping:** Random Forest starts by creating random subsets of the original dataset with replacement. This process is called bootstrapping. Each subset is used to train a decision tree.
- **Feature Selection:** At each node of the decision tree, a random subset of features is selected as candidates for splitting the node. This helps in reducing correlation between individual trees.
- **Decision Tree Training:** Each decision tree is trained on its respective subset of the data. The trees are grown to their maximum depth or until a stopping criterion is met (e.g., maximum depth reached, minimum samples per leaf).
- **Voting or Averaging:** For classification tasks, each decision tree predicts the class of a new data point. The class with the most votes across all trees is assigned as the final prediction. For regression tasks, the predictions of all trees are averaged to produce the final prediction.

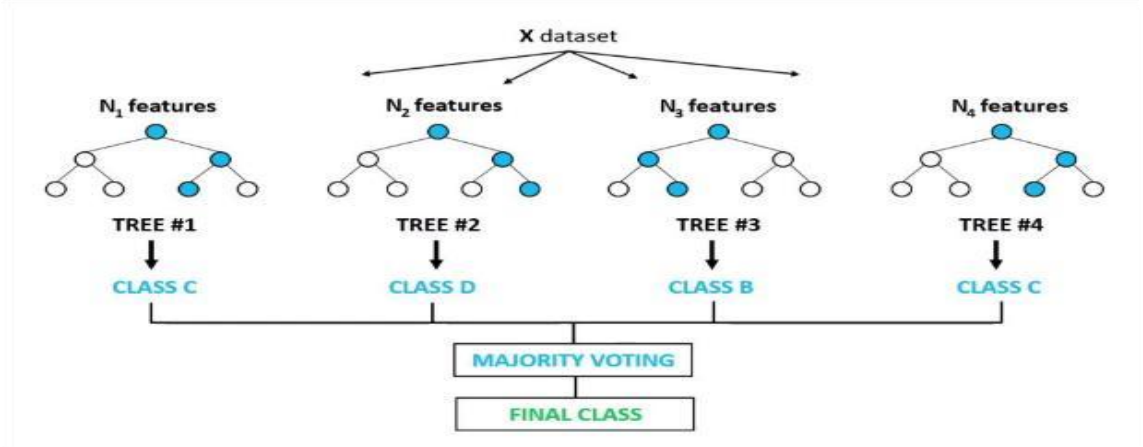


Figure 6.2 RFC

9.1.3 K-NEAREST NEIGHBOUR

K-Nearest Neighbors (K-NN) is a simple yet powerful supervised machine learning algorithm used for classification and regression tasks. It operates on the principle of similarity, where it predicts the class or value of a new data point based on the majority class or average value of its k nearest neighbors in the feature space.

In K-NN classification:

The algorithm calculates the distance between the new data point and every other point in the dataset using a distance metric such as Euclidean distance.It then selects the k nearest neighbors (data points with the smallest distances) to the new point.

Finally, it assigns the class label of the majority of the k neighbors to the new data point as its predicted class.

In K-NN regression:

Similarly, the algorithm calculates the distance between the new data point and every other point in the dataset.It selects the k nearest neighbors.Finally, it assigns the average value of the target variable of the k neighbors to the new data point as its predicted value.

The choice of the parameter k is crucial in K-NN. A small value of k can lead to noisy predictions and higher variance, while a large value of k can result in smoother decision boundaries but might oversimplify the model and lead to bias.

K-NN is a non-parametric algorithm, meaning it doesn't make any assumptions about the underlying data distribution. It's also known as a lazy learner because it doesn't explicitly learn a model during training; instead, it memorizes the training data and performs computation during inference.

K-NN is relatively simple to implement and understand, making it a popular choice for baseline models and as a starting point for exploring datasets. However, it can be computationally expensive, especially with large datasets, as it requires calculating distances between the new data point and every other point in the dataset during prediction.

Overall, K-NN is a versatile algorithm that can be effective in many scenarios, particularly when the decision boundary is complex and data is not linearly separable. However, it's essential to choose the appropriate value of k and preprocess the data properly to achieve optimal performance

CHAPTER 10

SYSTEM RQUIREMENTS

10.1 HARDWARE REQUIREMENTS

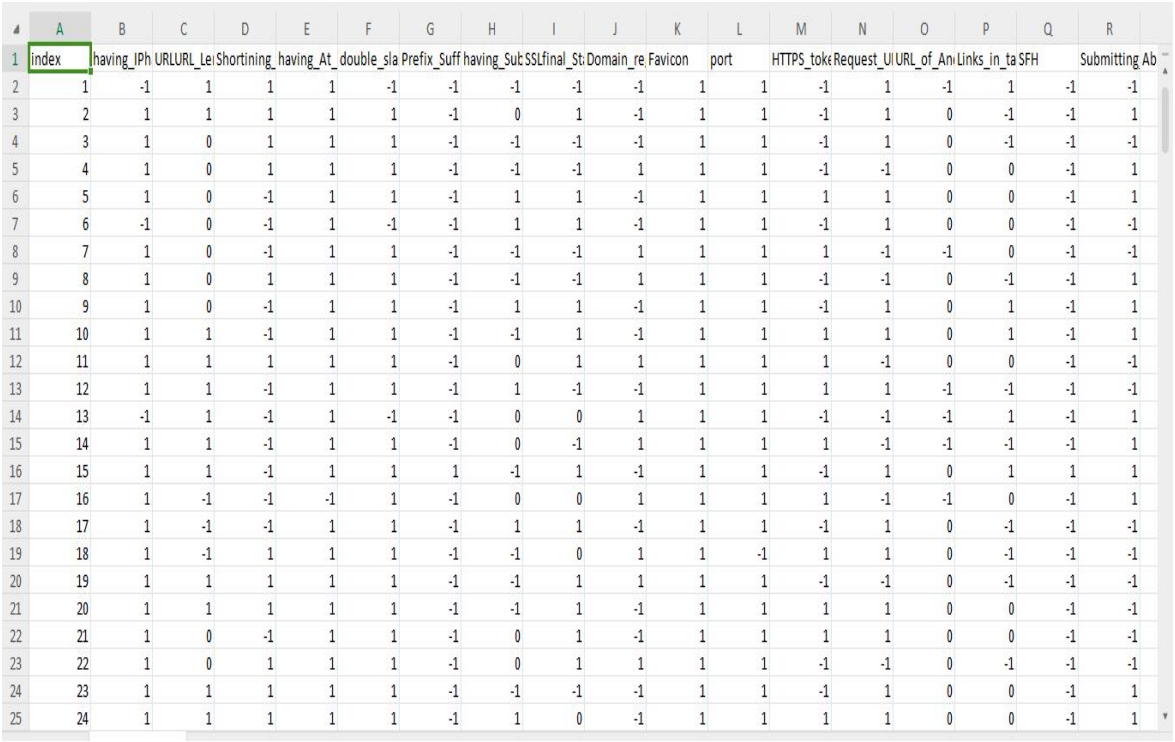
- **System** : Pentium i3 Processor
- **Hard Disk** : 500 GB.
- **Monitor** : 15'' LED
- **Input Devices** : Keyboard, Mouse
- **Ram** : 4 GB

10.2 SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 10.
- **Coding Language** : PYTHON
- **Tool** : ANACONDA, JUPYTER NOTEBOOK.
- **Libraries** : pandas, numpy, scikit-learn, BeautifulSoup, urllib, regex, tldextract, ssl, socket, whois.

CHAPTER 11
RESULT AND DISCUSSIONS

LEXICAL FEATURES



index	having_IPh	URLURL_Le	Shortining_	having_At_	double_sla	Prefix_Suff	having_Sub	SSLfinal_St	Domain_re	Favicon	port	HTTPS_toke	Request_UI	URL_of_Ani	Links_in_ta	SFH	Submitting_Ab
1	-1	1	1	1	-1	-1	-1	-1	-1	1	1	-1	1	-1	1	-1	-1
2	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1	1
3	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1	-1
4	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1	1
5	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1	1
6	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1	0	0	-1	-1
7	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	0	-1	-1
8	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	-1	-1	1
9	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1	-1	1
10	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1	1
11	1	1	1	1	1	-1	0	1	1	1	1	1	-1	0	0	-1	-1
12	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1	-1
13	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1	-1	1	-1	1
14	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1	-1	-1	-1	1
15	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1	0	1	1	1
16	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1	-1	0	-1	1
17	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	-1	-1	-1
18	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1	0	-1	-1	-1
19	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	0	-1	-1	-1
20	1	1	1	1	1	-1	-1	1	-1	1	1	1	1	0	0	-1	-1
21	1	0	-1	1	1	-1	0	1	-1	1	1	1	1	0	0	-1	-1
22	1	0	1	1	1	-1	0	1	1	1	1	-1	-1	0	-1	-1	-1
23	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	0	-1	1
24	1	1	1	1	1	-1	1	0	-1	1	1	1	1	0	0	-1	1

Figure 11.1 LEXICAL FEATURES

Lexical features refer to characteristics of text or words that can be extracted and analyzed for various purposes, such as text classification, sentiment analysis, or phishing website detection. These features are derived from the lexical properties of the text itself and do not consider the semantics or context.

In the context of phishing website detection, lexical features can be used to analyze the URL structure, domain name, presence of certain characters or patterns, and other textual attributes that may indicate whether a website is legitimate or malicious.

GET START PAGE

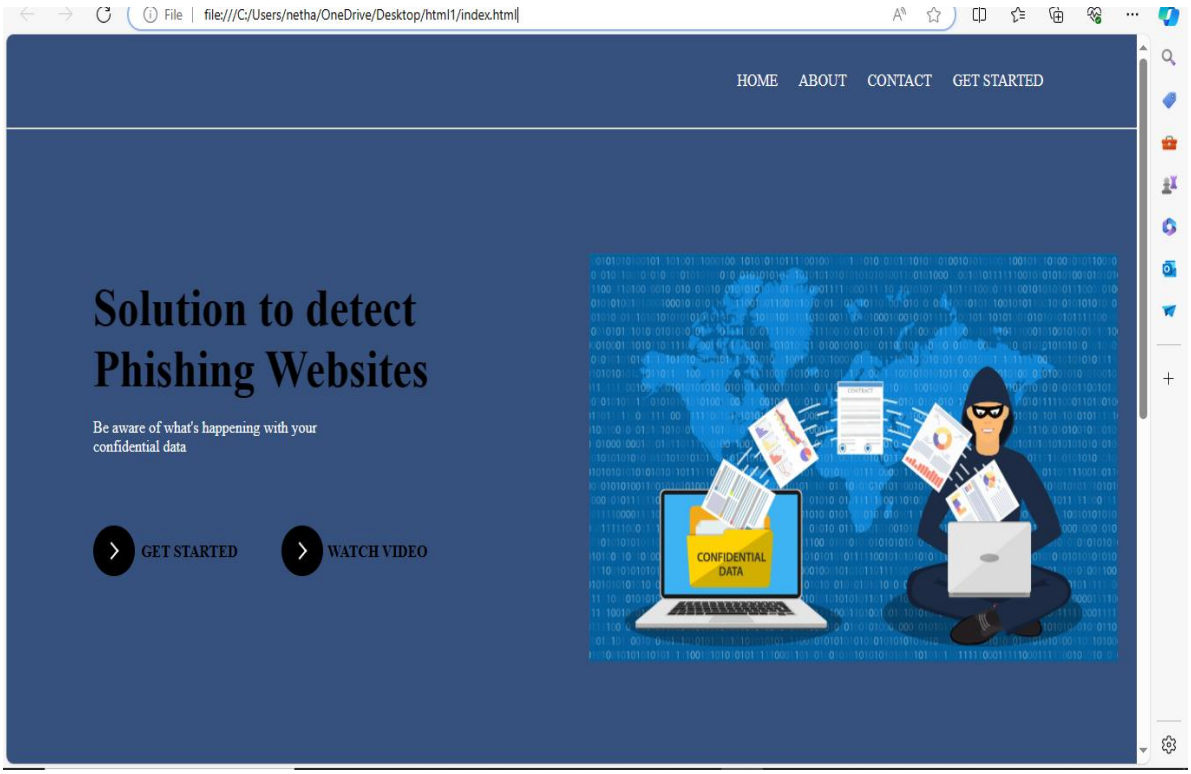


Figure 11.2 GET STARTED PAGE

The 'Get Started' page is an essential component of the 'Web Phishing Detection' platform, designed to guide users in taking proactive steps towards enhancing their online security. By offering clear instructions and accessible resources, this page enables visitors to initiate their journey towards understanding and combatting phishing threats effectively. Through engaging content and intuitive navigation, users are encouraged to explore tools and techniques for identifying and mitigating phishing risks, empowering them to protect their sensitive information and digital assets from malicious actors on the internet.

INPUT URL PAGE

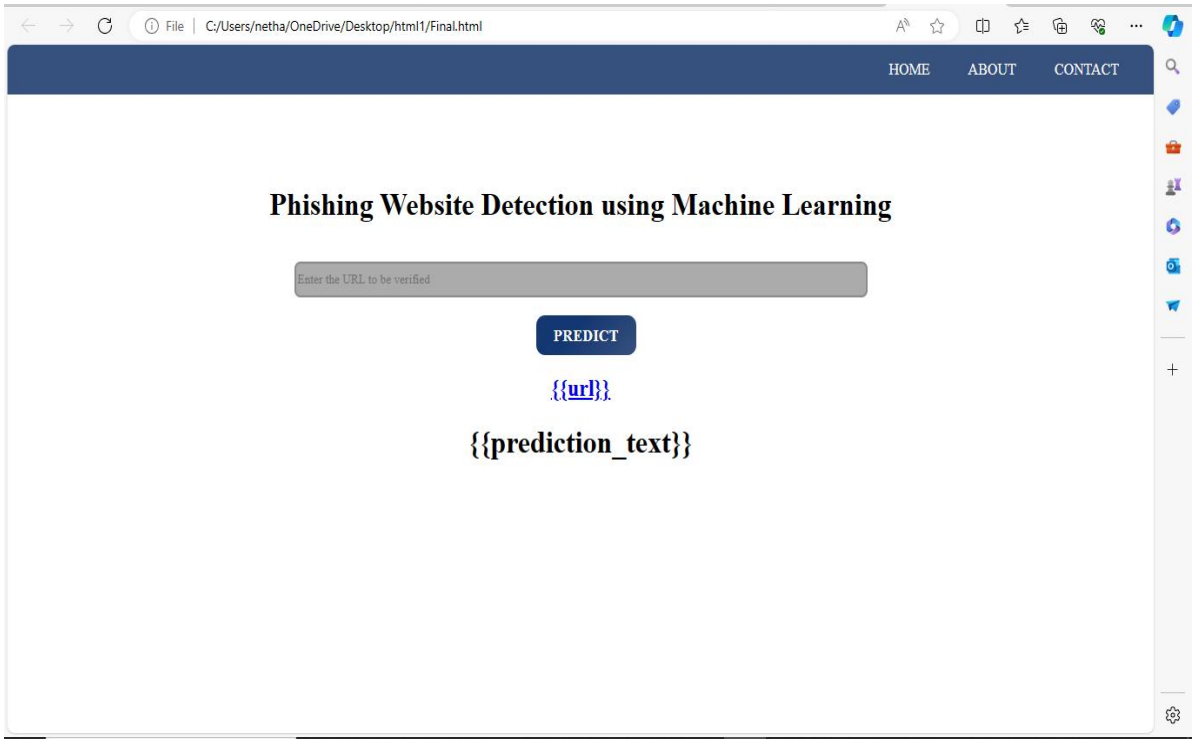


Figure 11.3 INPUT URL PAGE

The input URL page serves as the gateway for users to engage with the 'Phishing Website Detection using Machine Learning' system. With a clean and intuitive interface, users can enter the URL they wish to verify for potential phishing threats. The page provides a seamless experience, guiding users through the process with a clear call-to-action button for prediction. Upon submission, users receive instant feedback, including the predicted outcome and a direct link to the analyzed URL for further inspection.

URL PREDICTION PAGE

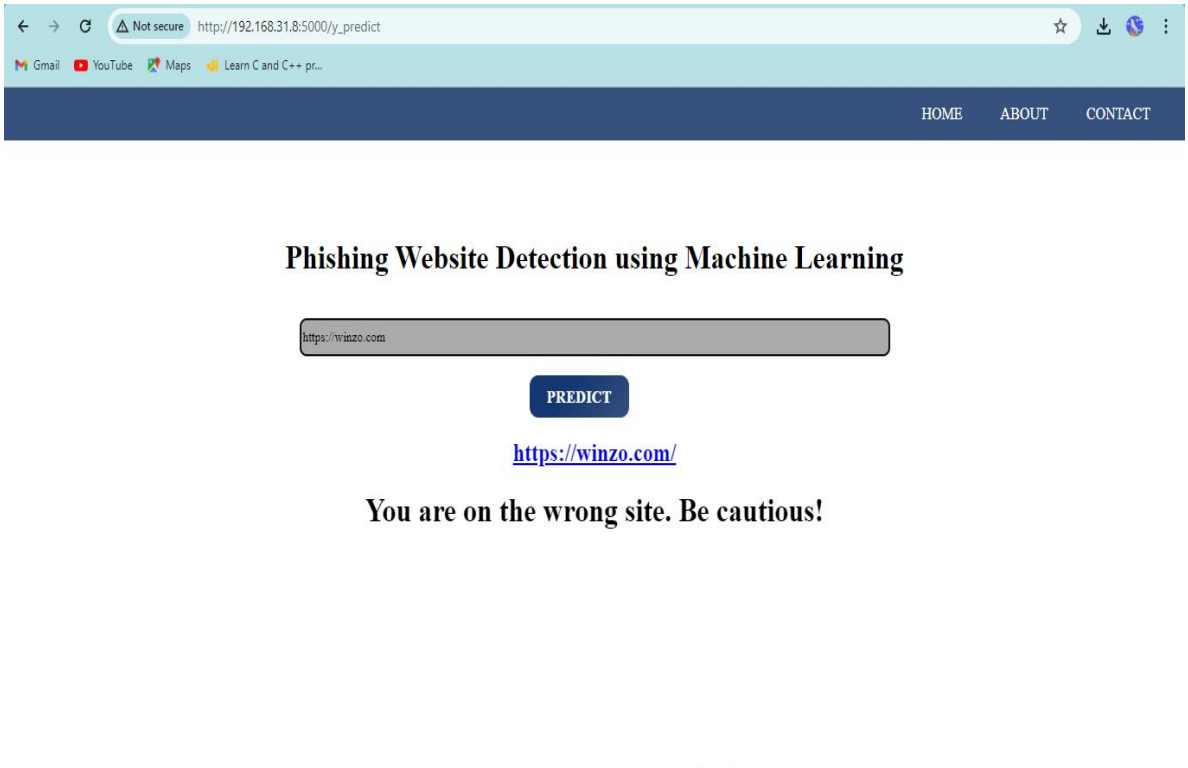


Figure 11.4 URL PREDICTION PAGE

The URL prediction page is a crucial component of the "Phishing Website Detection using Machine Learning" system, offering users a simple yet powerful interface to assess the legitimacy of URLs. Featuring an intuitive design, users can effortlessly input URLs for analysis. Upon submission, the system swiftly processes the input and delivers insightful predictions regarding the URL's phishing potential. With clear and concise feedback, users gain valuable insights into the safety of the provided URL, empowering them to make informed decisions while browsing the web.

PERFORMANCE TESTING

RANDOM FOREST CLASSIFIER:

```
In [45]: print(classification_report(y_test, y_pred_rf))
```

	precision	recall	f1-score	support
-1	0.98	0.95	0.97	1014
1	0.96	0.99	0.97	1197
accuracy			0.97	2211
macro avg	0.97	0.97	0.97	2211
weighted avg	0.97	0.97	0.97	2211

LOGISTIC REGRESSION:

```
In [46]: print(classification_report(y_test, y_pred_logistic))
```

	precision	recall	f1-score	support
-1	0.92	0.89	0.91	1014
1	0.91	0.94	0.92	1197
accuracy			0.92	2211
macro avg	0.92	0.92	0.92	2211
weighted avg	0.92	0.92	0.92	2211

K-NEAREST NEIGBHOUR:

```
In [47]: print(classification_report(y_test, y_pred_knn))
```

	precision	recall	f1-score	support
-1	0.95	0.92	0.94	1014
1	0.93	0.96	0.95	1197
accuracy			0.94	2211
macro avg	0.94	0.94	0.94	2211
weighted avg	0.94	0.94	0.94	2211

CHAPTER 12

CONCLUSION

Phishing is an appalling threat in the web security domain. In this attack, the user inputs his/her personal information to a fake website which looks like a legitimate one. We have presented a survey on phishing detection approaches based on visual similarity. This survey provides a better understanding of phishing website, various solution, and future scope in phishing detection. Many approaches are discussed in this paper for phishing detection; however most of the approaches still have limitations like accuracy, the countermeasure against new phishing websites, failing to detect embedded objects, and so forth. These approaches use various features of a webpage to detect phishing attacks, such as text similarity, font colour, font size, and images present in the webpage. Text based similarity approaches are relatively fast, but they are unable to detect phishing attack if the text is replaced with some image. Image processing based approaches have a high accuracy rate while they are complex in nature and are time-consuming. Furthermore, most of the work is done offline. These involve data collection and profile-creation phases to be completed first. A comparative table is prepared for easy glancing at the advantages and drawbacks of the available approaches. No single technique is enough for adopting it for phishing detection purposes. Detection of phishing websites with high accuracy is still an open challenge for further research and development.

CHAPTER 13
FUTURE ENHANCEMENT

In future if we get structured dataset of phishing we can perform phishing detection much more faster than any other technique.In future we can use a combination of any other two or more classifier to get maximum accuracy. We also plan to explore various phishing techniques that uses Lexical features, Network based features,Content based features, Webpage based features and HTML and JavaScript features of web pages which can improve the performance of the system. In particular, we extract features from URLs and pass it through the various classifiers.

Developing a sophisticated phishing detection system is crucial in today's digital landscape, where cyber threats continue to evolve in complexity and frequency. Looking ahead, the availability of structured datasets tailored for phishing detection holds immense potential to revolutionize the efficiency and accuracy of such systems. By harnessing these datasets, we can anticipate a substantial acceleration in phishing detection processes compared to traditional techniques.

HTML and JavaScript features offer valuable insights into the underlying code of web pages, allowing us to identify malicious scripts and payloads embedded within them. By analyzing the markup and scripting languages used to construct web pages, we can pinpoint vulnerabilities and exploit them to neutralize phishing attempts. This proactive approach enables us to mitigate the impact of phishing attacks and protect users from malicious code execution.

CHAPTER 14
REFERENCE

[1]. M. Bazarganigilani, ”Phishing E-Mail Detection Using Ontology Concept and Nave Bayes Algorithm,” International Journal of Research and Reviews in Computer Science, vol. 2,no.2, 2011.

[2]. M. Chandrasekaran, et al., ”Phoney: Mimicking user response to detect phishing attacks,” in In: Symposium on World of Wireless, Mobile and Multimedia Networks, IEEE Computer Society, 2006, pp. 668-672

[3].M. Chandrasekaran, et al., ”Phishing email detection based on structural properties”, in New York State Cyber Security Conference (NYS) , Albany, NY ,” 2006.

[4]. I. Fette, et al., ”Learning to detect phishing emails,” in Proc. 16th International World Wide Web Conference (WWW 2007), ACM Press, New York, NY, USA, May 2007, pp. 649-656

[5]. “Learning to Detect Phishing Emails” Ian Fette School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA icf@cs.cmu.edu Norman Sadeh School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA Anthony Tomasic School of Computer Science Carnegie Mellon University Pittsburgh, PA, 15213, USA .

[6]. Modeling and Preventing Phishing Attacks by Markus Jakobsson, Phishing detection system for e -banking using fuzzy data mining by Aburrous, M. ; Dept. of Comput., Univ. of Bradford, Bradford, UK ; Hossain, M.A. ; Dahal, K. ; Thabatah, F.

[7]. Phishing Attack Detection on Text Messages using Machine Learning Techniques - 2022 IEEE Pune Section International Conference.

[8]. Phishing or not phishing? A Survey on the Detection of Phishing Websites-2023 University of Pavia Through the CRUI-CARE Agreement.

[9]. "Protecting Users Against Phishing Attacks with AntiPhish" Engin Kirda and Christopher Kruegel Technical University of Vienna.

[10]. P.R.a.D.L.Ganger,"Gone phishing: Evaluating anti-phishing tools for windows. Technical report, ," September 2006

[11]. The Emergence Threat of Phishing attack and the detection technique using machine learning models -2021 International Conference on automation , control and mechatronics for industry.

CHAPTER 15

ANNEXURE

APP.py

```
@app.route('/')
@app.route('/index.html')
def home():
    return render_template('index.html')
@app.route('/contact.html')
def contact():
    return render_template('contact.html')
@app.route('/Final.html')
def predict():
    return render_template('Final.html')
#Fetches the URL given by the URL and passes to inputScript
@app.route('/y_predict', methods=['POST','GET'])
def y_predict():
url = request.form['URL']
    checkprediction = inputScript.main(url)
    prediction = model.predict(checkprediction)
    print(prediction)
    output=prediction[0]
```

```
    if(output==-1):
        pred="Your are safe!! This is a Legitimate Website."
    else:
        pred="You are on the wrong site. Be cautious!"
     return render_template('Final.html', prediction_text='{}'.format(pred),
url=url)
if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

INPUTSCRIPT.py

```
import regex
from tldextract import extract
import ssl
import socket
from bs4 import BeautifulSoup
import urllib.request
import whois
import datetime
def url_having_ip(url):
#using regular function
    symbol=regex.findall(r'(http((s)?)://)((((\d)+).)*)((\w)+)(/((\w)+))?',url)
    if(len(symbol)!=0):
        having_ip = 1 #phishing
```

```
    else:
        having_ip = -1 #legitimate
    return(having_ip)
    return 0
def url_length(url):
    length=len(url)
    if(length<54):
        return -1
    elif(54<=length<=75):
        return 0
    else:
        return 1
def having_at_symbol(url):
    symbol=regex.findall(r'@',url)
    if(len(symbol)==0):
        return -1
    else:
        return 1
 def doubleSlash(url):
    #ongoing
    return 0
def prefix_suffix(url):
```

```
    subDomain, domain, suffix = extract(url)
    if(domain.count('-')):
        return 1
    else:
        return -1
def sub_domain(url):
    subDomain, domain, suffix = extract(url)
    if(subDomain.count('.')==0):
        return -1
    elif(subDomain.count('.')==1):
        return 0
    else:
        return 1
def domain_registration(url):
    try:
        w = whois.whois(url)
        updated = w.updated_date
        exp = w.expiration_date
        length = (exp[0]-updated[0]).days
        if(length<=365):
            return 1
        else:
```

```
            return -1
    except:
        return 0
    return 0
def main(url):
check=[[url_having_ip(url),url_length(url),url_short(url),having_at_symb
ol(url),doubleSlash(url),prefix_suffix(url),sub_domain(url),SSLfinal_Stat
e(url),domain_registration(url),favicon(url),port(url),https_token(url),req
uest_url(url),url_of_anchor(url),Links_in_tags(url),sfh(url),email_submit(
url),abnormal_url(url),redirect(url),on_mouseover(url),rightClick(url),po
pup(url),iframe(url),age_of_domain(url),dns(url),web_traffic(url),page_ra
nk(url),google_index(url),links_pointing(url),statistical(url)]]
        print(check)
        return check
```

PHISHINGDETECTION.ipynb

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, accuracy_score
df= pd.read_csv("dataset_website.csv")
df.head()
x=df.iloc[:,1:31].values
y=df.iloc[:, -1].values
```

```
print(x,y)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,
random_state=0)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred1=lr.predict(x_test)
from sklearn.metrics import accuracy_score
log_reg=accuracy_score (y_test,y_pred1)
import pickle
pickle.dump(lr, open('Phishing_website.pkl', 'wb'))
```

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">
   <title>Web Phishing Detection</title>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/index.css')}}">
</head>
<body>
```

```
  <div class="banner">
    <div class="navbar">
      <ul>
        <li><a href='index.html'>Home</a></li>
        <li><a href='index.html#about'>About</a></li>
        <li><a href='contact.html'>Contact</a></li>
        <li><a href='Final.html'>Get Started</a></li>
      </ul>
    </div>
<hr>
    <div class="content">
      <h1><b>Solution to detect<br> Phishing Websites</b></h1>
      <br>
        <p>Be aware of what's happening with your<br> confidential
data</p>
    </div>
    <div class="images">
    <image src="{{ url_for('static', filename='css/th.jpeg')}}">
    </div>
    <div style="margin-left:100px; margin-top:-70ch;"class="btn">
      <button class="learn-more">

      <a  href="Final.html">
        <span class="circle" aria-hidden="true">
          <span class="icon arrow"></span>
        </span>
        <span class="button-text">Get Started</span></a>
      </button>
      <button class="learn-more">
```

```
            <a  href="https://getcssscan.com/css-buttons-examples">
               <span class="circle" aria-hidden="true">
                  <span class="icon arrow"></span>
               </span>
               <span class="button-text">Watch Video</span></a>
         </button>
         </div>
      </div>
   </div>
   <br><br><br>
                  <u><h2     id="about"     style="text-align:center;font-size:
35px">About</h2></u>
<div class="text">
   <div style="float:left" class="text1">
      <p style=" margin-left:20px";>
          Web service is one of the key combination software services for
the <br>
           Internet. Web phishing is one of many security threats to web
services<br>
          on the Internet. Web phishing aims to steal private information,
such <br>
            as usernames, passwords, and credit card details, by way of
impersonating <br>
         a legitimate entity.
      </p>
   </div>
   <div style="float: right;" class="text2">
      <p style="margin-right:30px;">
```

```
The recipient is then tricked into clicking a malicious link,which can<br>
lead to the installation of malware,the freezing of the system as part <br>
of a ransomware attack or the revealing of sensitive information. It <br>
will lead to information disclosure and property damage.
    </p>
  </div>
</div>
  <br><br><br><br><br><br><br><br>
  <br><br>
  <div  class="footer">
    <br><br><br><br>
    <h2>Check your Websites</h2>
    <br><br>
<div style="float:left;
            margin-left: 60px;
            padding: 0;
            font-size: larger;" class="foot1">
<p>Understanding if the website is a valid one or not is important and plays a vital role in securing the data.
    <br>To know if the URL is a valid one or you are information is at risk check your website.
    </p>
</div>
<div >
     <a style="margin-left:20px;
padding-:5px;
```

```
            text-decoration: none;
            font-family: Tw cen MT;
                    " class="button-24" href="Final.html">Check your
Website</a>
      </div>
      <br><br><br><br><br><br><br><br>
   </div>
 </body>
</html>
```

Final.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">
   <title>Prediction</title>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/Final.css')}}">
</head>
<body>
   <div class="navbar">
      <ul>
         <li><a href='index.html'>Home</a></li>
         <li><a href='index.html#about'>About</a></li>
         <li><a href='contact.html'>Contact</a></li>
      </ul>
```

```
</div><br>
<br>
<br>
<div>
  <form class="input" method="POST" action="/y_predict">
                <h1>Phishing Website Detection using Machine Learning</h1><br>
          <input name="URL" placeholder="Enter the URL to be verified">
    <br><br>
      <button class="button-75" role="button" type="submit"><span class="text">Predict</span></button>
    <a href="{{url}}" target="_blank"><h2>{{url}}</h2></a>
    <h1>{{prediction_text}}</h1>
  </form></div></body>
</html>
```