A
Mini-Project Report on

Detection and Prevention of Phishing Websites using Machine Learning Techniques

Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

By

Tanmay Buchade	21106031
Sachin Sapkale	21106026
Shashikant Shukla	21106024
Devesh Sali	21106016

Under the guidance of

Dr. Jaya Gupta

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai

2023-2024

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **“Detection and Prevention of Phishing Websites using Machine Learning Techniques”** is a bonafide work of Tanmay Buchade (21106031), Sachin Sapkale(21106026), Devesh Sali (21106016), Shashikant Sukla (21106024) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

Dr. Jaya Gupta
Mini Project Guide

Dr. Jaya Gupta
Head of Department

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled **““Detection and Prevention of Phishing Websites using Machine Learning Techniques””** by **Tanmay Buchade (21106031), Sachin Sapkale(21106026), Devesh Sali (21106016) and Shashikant Shukla (21106024)** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2023-24***.

External Examiner: ________________

Internal Examiner: _______________

Place: APSIT, Thane
Date:

Declaration

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

Sachin Sapkale	Shashikant Shukla	Tanmay Buchade	Devesh Sali
(21106026)	(21106024)	(21106031)	(21106016)

ABSTRACT

Phishing attacks are a rapidly expanding threat in the cyber world, costing internet users billions of dollars each year. It is a criminal crime that involves the use of a variety of social engineering tactics to obtain sensitive information from users. Phishing techniques can be detected using a variety of types of communication, including email, instant chats, pop-up messages, and web pages. This study develops and creates a model that can predict whether a URL link is legitimate or phishing.

The data set used for the classification was sourced from an open source service called ‘Phish Tank’ which contain phishing URLs in multiple formats such as CSV, JSON, etc. and also from the University of New Brunswick dataset bank which has a collection of benign, spam, phishing, malware & defacement URLs. Over six (6) machine learning models and deep neural network algorithms all together are used to detect phishing URLs.

This study aims to develop a web application software that detects phishing URLs from the collection of over 5,000 URLs which are randomly picked respectively and are fragmented into 80,000 training samples & 20,000 testing samples, which are equally divided between phishing and legitimate URLs. The URL dataset is trained and tested base on some feature selection such as address bar-based features, domain-based features, and HTML & JavaScript-based features to identify legitimate and phishing URLs.

In conclusion, the study provided a model for URL classification into phishing and legitimate URLs. This would be very valuable in assisting individuals and companies in identifying phishing attacks by authenticating any link supplied to them to prove its validity.

Keywords: Phishing attack, Machine learning models, URL classification, Dataset

CHAPTER 1 INTRODUCTION..01
1. Introduction...1
1.1 Objectives..2
1.2 Scope..2
1.3 Methodology...2

CHAPTER 2 LITERATURE SURVEY..3
2. Literature Survey...4
2.1 History...4
2.2 Literature Review..5

CHAPTER 3 PROBLEM STATEMENT..11
3. Problem Statement...12
4. Experimental Setup...13
4.1 Hardware Setup...14
4.2 Software Setup...14

CHAPTER 5 Project Implementation...15
5. Project Implementation...16
5.1 Work Flow Diagram...16
5.2 Flowchart of the proposed system...17
5.3 Flowchart of the web interface..18
5.4 Data Collection...19
5.5 Feature extraction on the datasets..20
5.6 Data analysis & Visualization namaste..25
5.7 Data preprocessing..27
5.8 Phishing detection model...28
5.9 General working of the system..29
5.10 The home page..29
5.11 Web application source code...30
5.12 API (Application programming interface)..30
5.13 Proposed System...33

CHAPTER 6 Conclusion..34
6. Conclusion...35

References.................................... ………………………………………..36

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Now-a-days usage of internet and surfing through the browser has become a primary requirement for everyone. This could help them from gaining knowledge to fulfil requirements for their own needs. But the problem arises at security. Is it really healthy to surfthrough everywebsitewe see on the internet? Will it be safe and secure for the data present in the device? So, to resolve this problem we are going to train an ML model through various algorithm and let it study the websites to find fraudulent ones. There are many kinds of algorithms and techniques in the market that could help through to find fraudulent websites, but they aren't that accurate and precise in judging the site. One of such technique is based on using Antiphishing via black listing. Although it is a detection technique this isn't that accurate for judging the correctness of a website. This technique of ours gives the accurate and precise knowledge about the website rather than the normal procedures of detections. This system can be open-sourced to every running search engine or any other sustainable environment that works on various websites so thatit could become easier for the work force to detect the legitimacy of the site.

1.1.Objectives

The objective of this project is to develop a ML model that to detect the phishing websites with accuracy. Expected steps and procedures to be followed to fulfil the objectives are:

- To Load the dataset in to the model after feature extracting all the URLs.
- Sort out the dataset without null values and unwanted values.
- Visualize the data to know the frequent factor among the considered features from the dataset.
- Train the ML model with different algorithms and store their results for further comparison.
- Compare all the modules to find out the most accurate algorithm for the detection process.
- Sort out the dataset into phishing and legitimate to get the required output when a website is searched.

1.2.Scope

This study explores data science and machine learning models that use datasets gotten from open-source platforms to analyze website links and distinguish between phishing and legitimate URL links. The model will be integrated into a web application, allowing a user to predict if a URL link is legitimate or phishing. This online application is compatible with a variety of browsers.

1.3. Methodology

An extensive review was done on related topics and existing documented materials such as journals, e-books, and websites containing related information gathered which was examined and reviewed to retrieve essential data to better understand and know how to help improve the systemThe dataset was then pre-processed that is cleaned up from any abnormality such as missing data to avoid data imbalance. Afterward, expository data analysis was done on the dataset to explore and summarize the dataset. Once the dataset was free from all anomalies, website content-based features were extracted from the dataset to get accurate features to train and test the model. An extensive review was done on existing works of literature and machine learning models on detecting phishing websites to best decide the classification models to solve the problem of detecting phishing websites. Hence, Series of these machine learning classification models such as Decision Tree, Support Vector Machine, XGBooster, Multilayer perceptions, Auto encoder Neural Network and Random Forest was deployed on the dataset to distinguish between phishing and legitimate URLs.. Thus, a user can enter a URL link on the web application to predict if it is phishing or legitimate.

CHAPTER 2

LITERATURE SURVEY

2.LITERATURE SURVEY

2.1.HISTORY

2.1.1. The concept of phishing attacks originated in the late 1990s when malicious actors began using deceptive email tactics to trick users into disclosing sensitive information. In the early stages of phishing, the detection process relied heavily on manual scrutiny. Users were educated to be vigilant and identify common signs of phishing, such as misspelled URLs or generic greetings in emails.

As phishing attacks evolved in sophistication and scale, heuristic-based detection solutions were introduced. These systems employed predefined rules and patterns to identify potential phishing emails and websites. They analyzed email content for patterns, including keywords and suspicious links, in an attempt to flag malicious communications.

2.1.2. Around the mid-2000s, the adoption of blacklists and URL filtering gained momentum. Security organizations and companies started compiling lists of known phishing websites and malicious IP addresses. These lists were integrated into web browsers and email clients to provide warnings when users attempted to access blacklisted sites. While effective to some extent, this approach had limitations as attackers continued to refine their tactics.

Recognizing the need for more adaptive solutions, the field of phishing detection began incorporating machine learning (ML) and artificial intelligence (AI) techniques. Natural language processing and pattern recognition were applied to the detection process. These AI systems leveraged historical data to learn and adapt to evolving phishing methods, significantly enhancing detection capabilities.

2.1.3. The 2010s marked the emergence of behavioral analysis techniques in phishing detection. These methods focused on monitoring user behavior and identifying deviations from typical patterns. For example, they could detect if a user suddenly logged in from a different geographical location or interacted with emails in unusual ways. This real-time behavioral analysis proved valuable in identifying novel phishing attempts.

Simultaneously, threat intelligence services and feeds gained prominence. These services provided real-time information about emerging phishing threats, such as newly registered phishing domains or active malware campaigns. Security solutions integrated this intelligence to proactively protect users, providing an additional layer of defense.

2.1.4. As phishing attacks continued to exploit human vulnerabilities, organizations recognized the importance of user training and awareness programs. These initiatives educated employees and users on how to recognize phishing attempts, emphasizing the critical nature of avoiding suspicious links and refraining from downloading suspicious attachments.

2.1.5. In the modern era of phishing detection, systems often employ a combination of methods, leveraging both signature-based and behavior-based approaches. These multifaceted systems aim to improve detection rates while reducing false positives. The focus has shifted towards real-time protection, where phishing threats are identified and neutralized as soon as they are encountered, minimizing the window of vulnerability for users.

With the widespread use of mobile devices and messaging apps, the scope of phishing detection expanded to cover these platforms. Mobile-specific detection techniques and secure messaging features were developed to protect users across diverse communication channels. Additionally, many organizations transitioned to cloud-based security solutions, which provide scalable and up-to-date phishing detection services, ensuring protection even for remote and mobile users.

2.2.LITERATURE REVIEW

1."Variant of pishing attacks and there detection techniques" (2009), G. Jaspher Willsie Katherien, paradise Mercy Paise, Eligious Kalaivani. C, A. Amruta Rose

The concept of phishing attacks originated in the late 1990s when malicious actors began using deceptive email tactics to trick users into disclosing sensitive information. In the early stages of phishing, the detection process relied heavily on manual scrutiny. Users were educated to be vigilant and identify common signs of phishing, such as misspelled URLs or generic greetings in emails.

As phishing attacks evolved in sophistication and scale, heuristic-based detection solutions were introduced. These systems employed predefined rules and patterns to identify potential phishing emails and websites. They analyzed email content for patterns, including keywords and suspicious links, in an attempt to flag malicious communications.

Around the mid-2000s, the adoption of blacklists and URL filtering gained momentum. Security organizations and companies started compiling lists of known phishing websites and malicious IP addresses. These lists were integrated into web browsers and email clients to provide warnings when users attempted to access blacklisted sites. While effective to some extent, this approach had limitations as attackers continued to refine their tactics.

Recognizing the need for more adaptive solutions, the field of phishing detection began incorporating machine learning (ML) and artificial intelligence (AI) techniques. Natural language processing and pattern recognition were applied to the detection process. These AI systems leveraged historical data to learn and adapt to evolving phishing methods, significantly enhancing detection capabilities.

The 2010s marked the emergence of behavioral analysis techniques in phishing detection. These methods focused on monitoring user behavior and identifying deviations from typical patterns. For example, they could detect if a user suddenly logged in from a different geographical location or interacted with emails in unusual ways. This real-time behavioral analysis proved valuable in identifying novel phishing attempts.

Simultaneously, threat intelligence services and feeds gained prominence. These services provided real-time information about emerging phishing threats, such as newly registered phishing domains or active malware campaigns. Security solutions integrated this intelligence to proactively protect users, providing an additional layer of defense.

As phishing attacks continued to exploit human vulnerabilities, organizations recognized the importance of user training and awareness programs. These initiatives educated employees and users on how to recognize phishing attempts, emphasizing the critical nature of avoiding suspicious links and refraining from downloading suspicious attachments.

In the modern era of phishing detection, systems often employ a combination of methods, leveraging both signature-based and behavior-based approaches. These multifaceted systems aim to improve detection rates while reducing false positives. The focus has shifted towards real-time protection, where phishing threats are identified and neutralized as soon as they are encountered, minimizing the window of vulnerability for users.

With the widespread use of mobile devices and messaging apps, the scope of phishing detection expanded to cover these platforms. Mobile-specific detection techniques and secure messaging features were developed to protect users across diverse communication channels. Additionally, many organizations transitioned to cloud-based security solutions, which provide scalable and up-to-date phishing detection services, ensuring protection even for remote and mobile users**.**

2. “Detecting Phishing Websites Using Machine Learning" (2019) by Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, and Dr. Aram Alsedrani

Introduction

Phishing attacks remain a significant cybersecurity threat, with attackers continually refining their techniques to trick individuals into disclosing sensitive information. As a result, the development of effective phishing website detection methods is essential to protect users and organizations. In this literature survey, we explore the research conducted by Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, and Dr. Aram Alsedrani in 2019, titled "Detecting Phishing Websites Using Machine Learning." This study focuses on the use of machine learning techniques for identifying phishing websites, providing valuable insights into the state-of-the-art in phishing detection.

Phishing Attacks: A Growing Concern

Phishing attacks involve malicious actors impersonating trusted entities to deceive users into revealing sensitive information, such as login credentials or financial data. These attacks have become increasingly sophisticated, making them harder to detect using traditional methods. Thus, researchers and cybersecurity experts have turned to machine learning as a potent tool for tackling this evolving threat.

Machine Learning for Phishing Detection

The study by Alswailem et al. in 2019 explores the application of machine learning to detect phishing websites. This approach leverages algorithms and models trained on historical data to distinguish between legitimate and malicious sites. The research delves into various aspects of machine learning-based detection, including:

1. Feature Engineering: Feature selection and extraction are crucial steps in machine learning-based phishing detection. The study discusses the identification of relevant features such as URL structure, domain registration information, website content, and user behavior patterns. These features serve as input data for machine learning models.

2. Classification Algorithms: The researchers investigate different machine learning algorithms for classification, such as decision trees, random forests, and neural networks. These models are trained to predict whether a given website is legitimate or a phishing attempt based on the selected features.

3. Training and Evaluation: The study emphasizes the importance of a robust training dataset, consisting of both legitimate and phishing websites. The machine learning models are trained on this data to learn patterns indicative of phishing. Evaluation metrics like precision, recall, and F1-score are employed to assess the model's performance.

4. Real-Time Detection: Real-time phishing website detection is a critical requirement in cybersecurity. The study explores the deployment of machine learning models in real-time scenarios, allowing for the immediate identification of phishing threats as users interact with websites.

Challenges and Future Directions

While machine learning offers promising solutions for phishing website detection, the study also acknowledges several challenges, including the need for continuously updated datasets, the presence of adversarial attacks, and the risk of false positives. Researchers and practitioners are actively addressing these challenges and exploring innovative approaches to enhance the accuracy and reliability of machine learning-based detection systems.

Conclusion

The research by Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh, and Dr. Aram Alsedrani in 2019 represents a significant contribution to the field of phishing website detection. Their study underscores the potential of machine learning techniques in combating phishing threats and highlights the importance of feature engineering, classification algorithms, real-time detection, and rigorous evaluation. As phishing attacks continue to evolve, the use of machine learning will likely play an increasingly pivotal role in safeguarding users and organizations from these deceptive cyber threats.

3.“Detection and Prevention of Phishing Websites using Machine Learning Approach" (2021) by Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, and Prof. S. P. Godse

Introduction

Phishing attacks remain a significant cybersecurity challenge, necessitating the development of effective techniques to detect and prevent phishing websites. In this literature survey, we explore the research conducted by Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, and Prof. S. P. Godse in 2021, titled "Detection and Prevention of Phishing Websites using Machine Learning Approach." This study investigates the use of machine learning as a proactive approach to mitigate phishing threats, offering valuable insights into the state of the art in phishing detection and prevention.

Phishing Attacks: An Ongoing Threat

Phishing attacks involve deceptive tactics, with malicious actors impersonating trustworthy entities to trick users into revealing sensitive information. These attacks persistently evolve in sophistication and scale, demanding innovative approaches for their detection and prevention. Machine learning has emerged as a powerful tool in this context.

Machine Learning for Phishing Detection and Prevention

The study by Patil et al. in 2021 focuses on the application of machine learning to tackle phishing attacks. Key aspects of this research include:

1. **Feature Extraction:** The study delves into the importance of feature extraction in machine learning-based phishing detection. Relevant features, such as URL structure,

domain attributes, content analysis, and user behavior, are identified and used as inputs for machine learning models.

2. Machine Learning Models: The researchers explore various machine learning algorithms and models, including decision trees, support vector machines (SVMs), and ensemble methods, to effectively classify websites as legitimate or phishing. These models are trained on labeled datasets to learn the patterns indicative of phishing attempts.

3. Real-time Detection and Prevention: The study underscores the need for real-time detection and prevention mechanisms to combat phishing effectively. Machine learning models are deployed in real-time scenarios, continuously analyzing web traffic and user interactions to identify and block phishing websites promptly.

4. Evaluation Metrics: The researchers employ evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of their machine learning models. The effectiveness of these models in accurately detecting phishing websites is a critical aspect of the study.

Challenges and Future Directions

While machine learning offers promising solutions for phishing detection and prevention, the study acknowledges the challenges in maintaining up-to-date datasets, addressing adversarial attacks, and reducing false positives. Future research in this domain is expected to focus on enhancing the robustness and scalability of machine learning-based systems.

Conclusion

The research conducted by Vaibhav Patil, Pritesh Thakkar, Chirag Shah, Tushar Bhat, and Prof. S. P. Godse in 2021 underscores the growing significance of machine learning in the proactive detection and prevention of phishing websites. Their study highlights the critical role of feature extraction, machine learning models, real-time deployment, and rigorous evaluation in countering phishing threats. As phishing attacks persist in their evolution, machine learning continues to be a vital ally in safeguarding individuals and organizations from falling victim to these deceptive cyberattacks.

4.“Phishing Website Detection Based on Effective Machine Learning Approach" (2022) by Gururaj Harinahalli Lokesh and Goutham BoreGowda

Introduction

Phishing attacks continue to pose a significant threat to cybersecurity, necessitating the development of robust methods for detecting and thwarting phishing websites. In this literature survey, we delve into the research conducted by Gururaj Harinahalli Lokesh and Goutham BoreGowda in 2022, titled "Phishing Website Detection Based on Effective Machine Learning Approach." Their study explores the application of machine learning techniques to enhance the detection of phishing websites, providing valuable insights into the ever-evolving landscape of phishing detection.

Phishing Attacks: A Persistent Challenge

Phishing attacks involve deceptive tactics, with malicious actors impersonating legitimate entities to manipulate users into divulging sensitive information. These attacks continuously evolve, demanding innovative approaches for their detection and prevention. Machine learning has emerged as a potent tool in this ongoing battle.

Machine Learning for Phishing Detection

The study by Lokesh and BoreGowda in 2022 focuses on harnessing machine learning for the identification of phishing websites. Key aspects of their research include:

1. Feature Engineering: The study emphasizes the importance of feature engineering in machine learning-based phishing detection. Essential features, including URL attributes, content

analysis, and user interaction patterns, are identified and utilized as inputs for machine learning models.

2. Machine Learning Models: The researchers explore various machine learning algorithms and models, such as deep neural networks, support vector machines (SVMs), and ensemble methods, to effectively classify websites as legitimate or phishing. These models are trained on comprehensive datasets, enabling them to learn the intricate patterns associated with phishing attempts.

3. Real-time Detection and Response: The study highlights the need for real-time detection and response mechanisms to combat phishing effectively. Machine learning models are deployed in real-world scenarios, continuously analyzing web traffic and user interactions to promptly identify and neutralize phishing websites.

4. Performance Evaluation: Lokesh and BoreGowda employ rigorous evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the performance of their machine learning models. The study focuses on the effectiveness of these models in accurately detecting phishing websites while minimizing false positives.

Challenges and Future Directions

While machine learning holds immense promise in phishing detection, the study acknowledges challenges such as dataset maintenance, adaptability to evolving phishing tactics, and adversarial attacks. Future research endeavors are expected to center on further enhancing the resilience and scalability of machine learning-based solutions.

Conclusion

The research conducted by Gururaj Harinahalli Lokesh and Goutham BoreGowda in 2022 underscores the increasing significance of machine learning in the proactive detection of phishing websites. Their study emphasizes the critical role of feature engineering, diverse machine learning models, real-time deployment, and comprehensive performance evaluation in countering the pervasive threat of phishing attacks. As phishing attacks persist in their evolution, machine learning remains a pivotal asset in fortifying cybersecurity and protecting individuals and organizations from falling victim to these deceitful cyber threats.

5.“Phishing Website Detection based on Multidimensional Features driven by Deep Learning" (2019) by Peng Yang, Guangzhen Zhao, and Peng Zeng

Introduction

Phishing attacks continue to pose a significant threat to cybersecurity, necessitating innovative approaches for the detection and mitigation of phishing websites. In this literature survey, we delve into the research conducted by Peng Yang, Guangzhen Zhao, and Peng Zeng in 2019, titled "Phishing Website Detection based on Multidimensional Features driven by Deep Learning." Their study explores the application of deep learning techniques to enhance the detection of phishing websites, offering valuable insights into the evolving landscape of phishing detection

Phishing Attacks: A Persistent Challenge

Phishing attacks involve deceptive tactics, with malicious actors impersonating legitimate entities to deceive users into disclosing sensitive information. These attacks continually evolve, necessitating advanced techniques for their detection and prevention. Deep learning has emerged as a potent tool in addressing this evolving threat.

Deep Learning for Phishing Detection

The study by Yang, Zhao, and Zeng in 2019 focuses on harnessing the power of deep learning to identify phishing websites. Key aspects of their research include:

1. Multidimensional Features: The study emphasizes the importance of multidimensional features, encompassing diverse aspects of website attributes, content analysis, and user interactions. These features serve as rich inputs for deep learning models.

2. Deep Neural Networks: The researchers explore the utilization of deep neural networks, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to effectively classify websites as legitimate or phishing. These models are capable of learning complex and hierarchical patterns associated with phishing attempts.

3. Real-time Detection and Response: The study highlights the significance of real-time detection and response mechanisms to combat phishing effectively. Deep learning models are deployed in practical scenarios, continuously analyzing web traffic and user interactions to promptly identify and neutralize phishing websites.

4. Performance Evaluation: Yang, Zhao, and Zeng employ rigorous evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the performance of their deep learning models. The study focuses on the effectiveness of these models in accurately detecting phishing websites while minimizing false positives.

Challenges and Future Directions

While deep learning offers promising solutions in phishing detection, the study acknowledges challenges such as dataset diversity, model interpretability, and adversarial attacks. Future research endeavors are expected to concentrate on refining deep learning-based solutions and enhancing their robustness in dynamic cybersecurity landscapes.

Conclusion

The research conducted by Peng Yang, Guangzhen Zhao, and Peng Zeng in 2019 underscores the increasing importance of deep learning in proactive phishing website detection. Their study highlights the critical role of multidimensional features, deep neural networks, real-time deployment, and comprehensive performance evaluation in countering the evolving threat of phishing attacks. As phishing techniques continue to evolve, deep learning remains a vital asset in strengthening cybersecurity defenses and safeguarding individuals and organizations from falling victim to these deceptive cyber threats.

CHAPTER 3

PROBLEM STATEMENT

3.Problem Statement

In the realm of cybersecurity, the persistent and escalating threat of phishing attacks is a cause for serious concern. Phishing attacks manifest as deceptive emails and meticulously crafted websites, intricately designed to deceive not only unsuspecting individuals but also organizations of all sizes. These fraudulent schemes are adept at impersonating legitimate entities with a single malicious purpose: extracting sensitive information. Such information often includes critical data like login credentials, financial details, and personal information, thereby posing substantial risks to both individuals and businesses alike.

The dynamic landscape of phishing attacks is characterized by relentless innovation and adaptation on the part of malicious actors. These cybercriminals continuously evolve their tactics to evade existing security measures, making the task of effective prevention and detection an ever-mounting challenge. This evolving landscape results in two critical issues:

False Positives: Existing anti-phishing solutions, while designed with the best intentions, sometimes generate false alarms. Legitimate websites are erroneously flagged as phishing sites, leading to unnecessary disruptions and potential user frustration.

Delayed Detection: Another challenge lies in the ability to swiftly detect newly emerging and highly sophisticated phishing websites. Traditional methods often struggle to keep up with the rapid pace of phishing attacks, leaving users vulnerable for longer periods.

Recognizing the pressing need for a solution, this project assumes a pivotal role in addressing this cybersecurity crisis. Our primary objective is to develop a state-of-the-art, adaptive phishing website detection system. This system is designed to accurately distinguish between phishing websites and legitimate ones, significantly reducing the occurrence of false alarms. To achieve this, we will leverage advanced machine learning algorithms, behavioral analysis techniques, and real-time monitoring.

By achieving our objective, our aim is to bolster the digital defenses of individuals and organizations against the pervasive menace of phishing attacks. Beyond this, our research contributes to the overarching goal of enhancing online security, safeguarding personal information, and preserving user trust in digital communication. Ultimately, this project seeks to empower users and businesses with the knowledge and tools required to identify and effectively mitigate the ever-evolving risks associated with phishing attacks. In doing so, we aim to foster a safer and more secure digital ecosystem, better prepared to face the challenges of an increasingly sophisticated cyber threat landscape.

CHAPTER 4

Experimental Setup

4.Experimental Setup

4.1 Hardware Setup

1) Specifications of the Current machine:
2) CPU: Ryzen 5 5600H
3) GPU: Nvidia GTX 1650
4) RAM: 16GB
5) ROM: 512 GB M.2 SSD

4.2 Software Setup

1) Windows Operating System (8/10)
2) Anaconda Navigator (Jupyter Notebook)
3) Web browser (Preferably Chrome)
4) Visual Studio Code

CHAPTER 5

Project Implementation

5.2 Project Implementation

5.1 Work flow Diagram

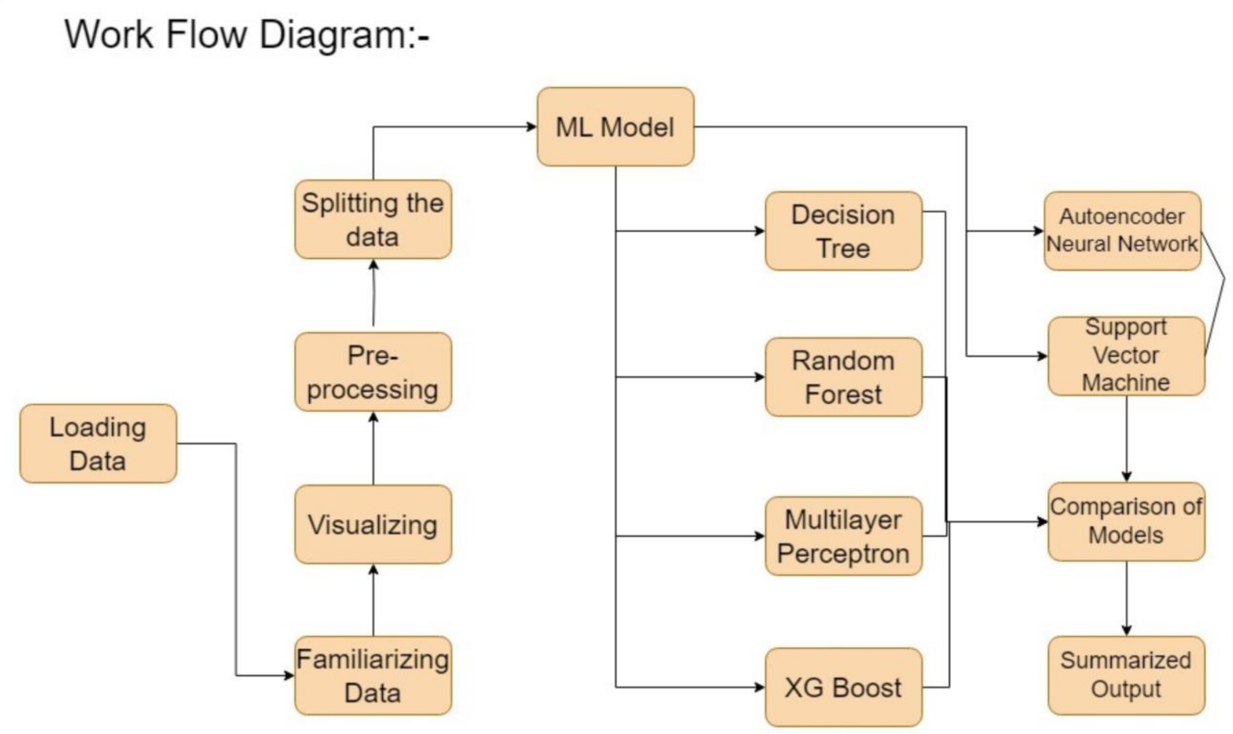


Figure 5.1: work flow diagram

5.2 Flowchart of the web interface

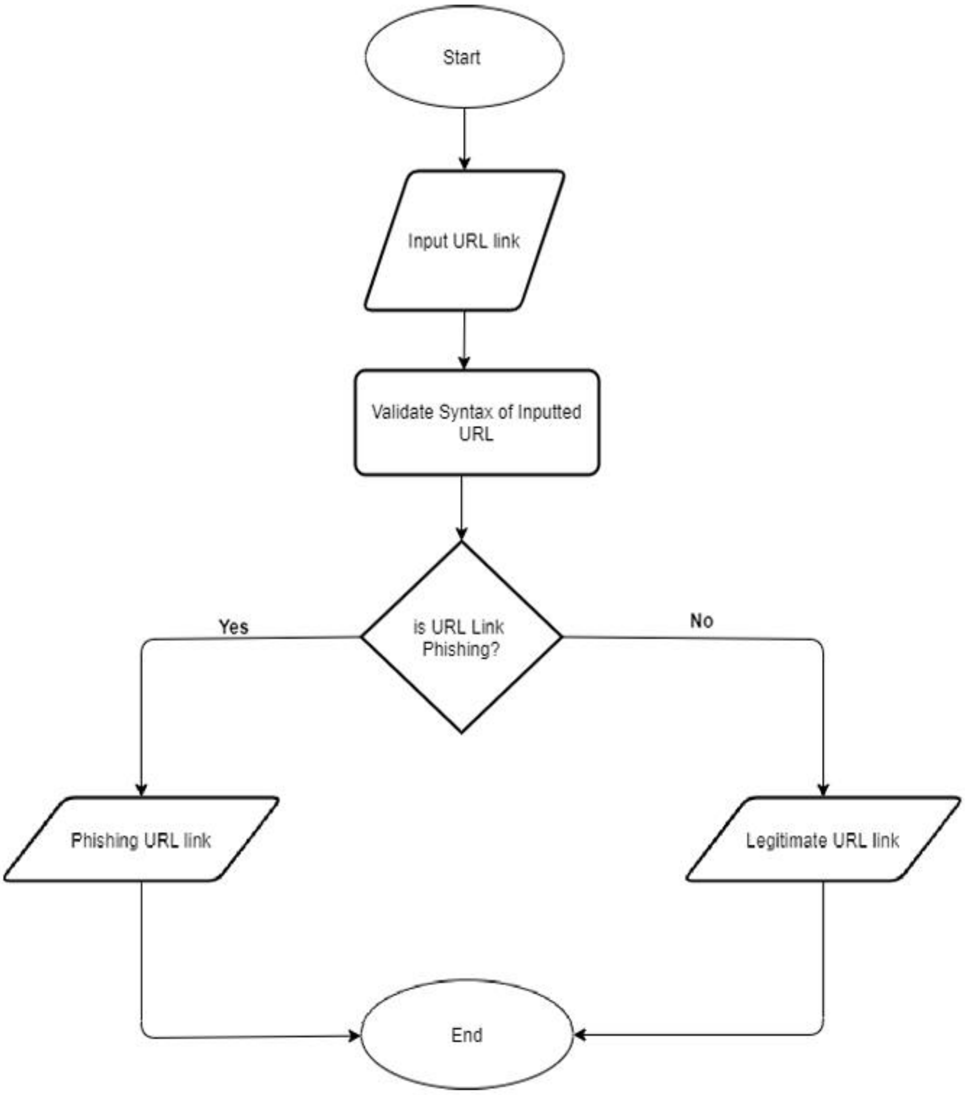


Figure 5.2: Flowchart of the web interface

5.3 Flowchart of the proposed system

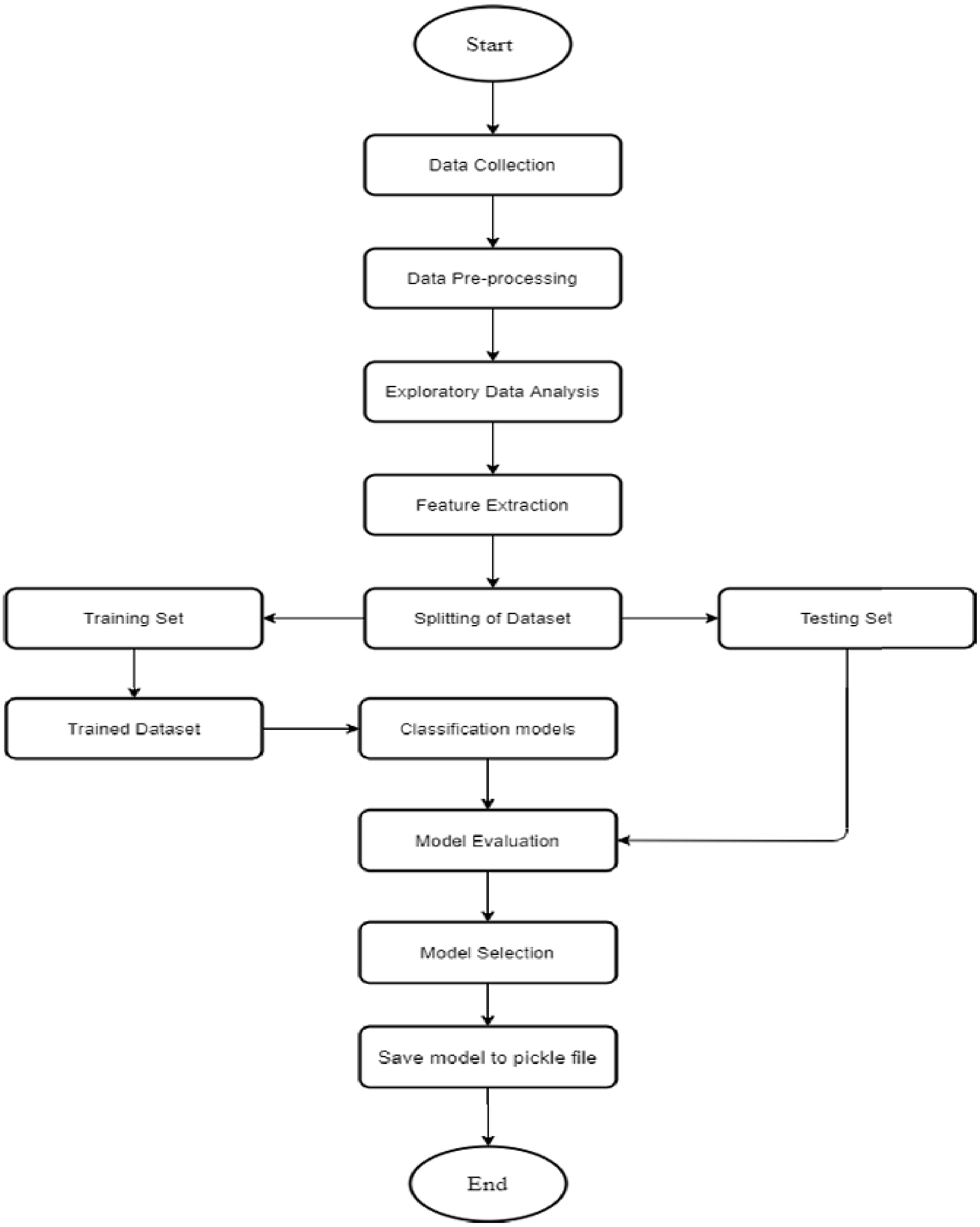
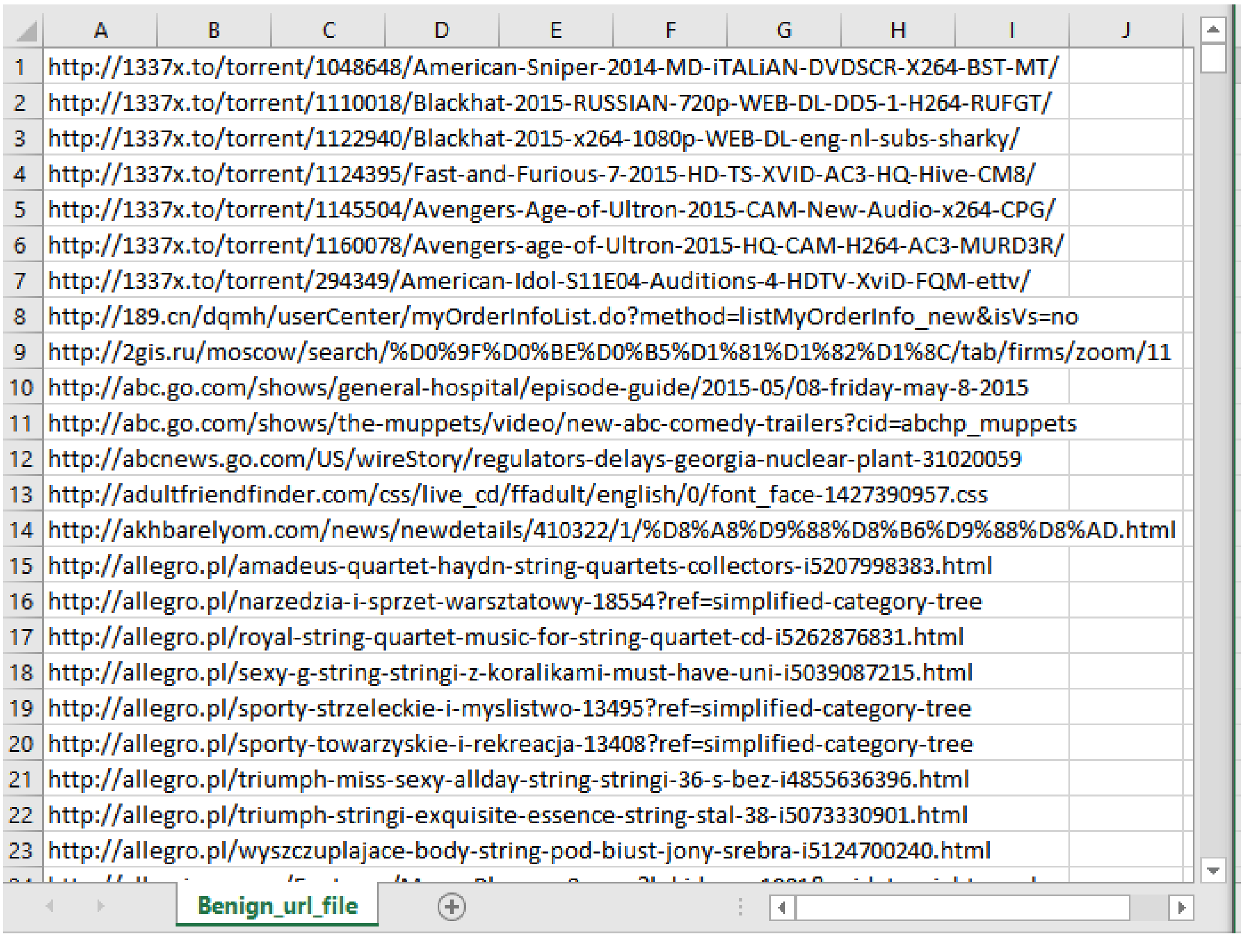


Figure 5.3: Flowchart of the proposed system

5.4. Data collection

The dataset used for the classification was sourced from was gotten from multiple sources listed in the earlier stated methodology.

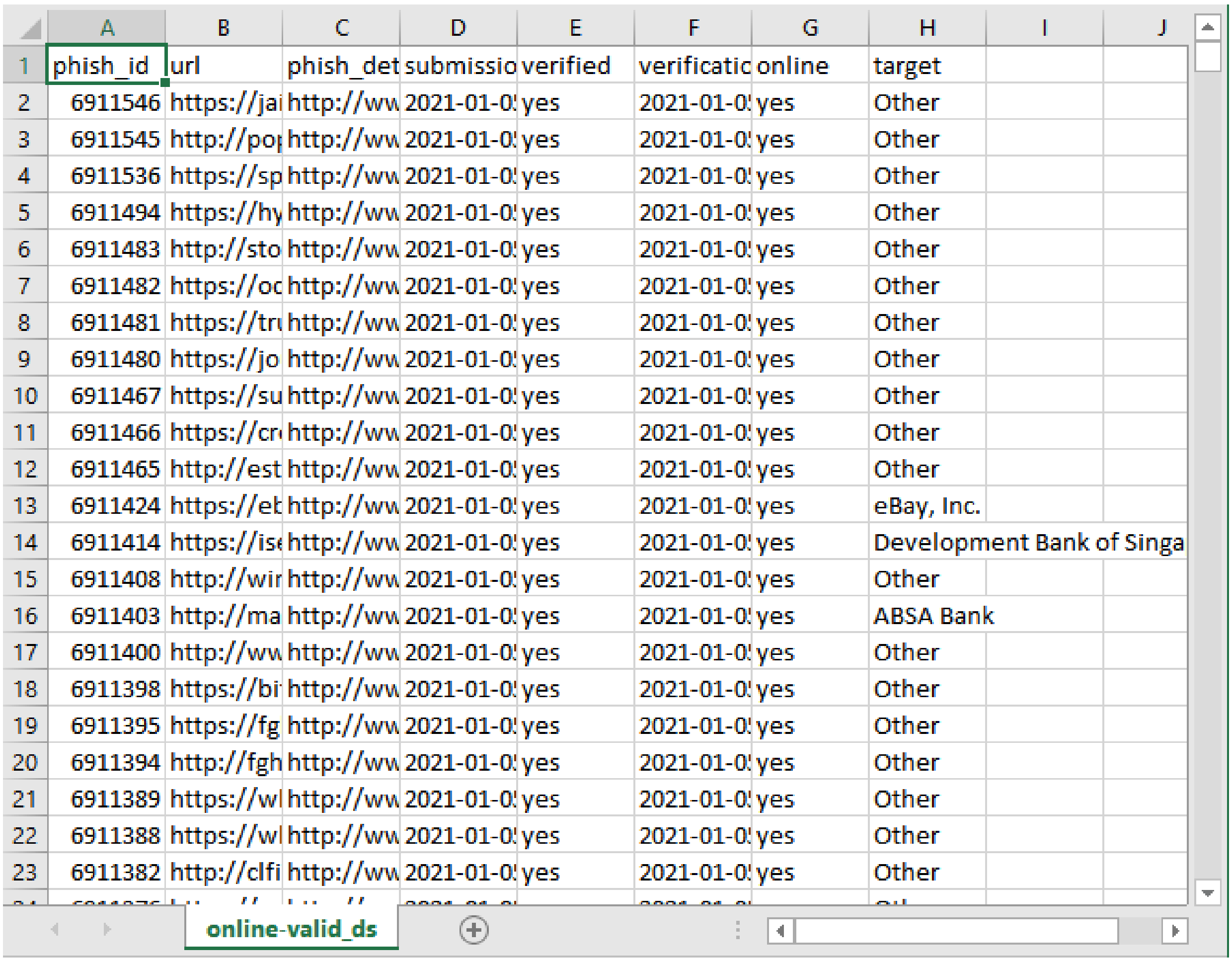
The dataset used for classifying the dataset into phishing and legitimate URLs was sourced from open source websites, samples of which are shown below in figure 5.1 and 5.2 respectively



	A	B	C	D	E	F	G	H	I	J
1	http://1337x.to/torrent/1048648/American-Sniper-2014-MD-iTALiAN-DVDSCR-X264-BST-MT/									
2	http://1337x.to/torrent/1110018/Blackhat-2015-RUSSIAN-720p-WEB-DL-DD5-1-H264-RUFGT/									
3	http://1337x.to/torrent/1122940/Blackhat-2015-x264-1080p-WEB-DL-eng-nl-subs-sharky/									
4	http://1337x.to/torrent/1124395/Fast-and-Furious-7-2015-HD-TS-XVID-AC3-HQ-Hive-CM8/									
5	http://1337x.to/torrent/1145504/Avengers-Age-of-Ultron-2015-CAM-New-Audio-x264-CPG/									
6	http://1337x.to/torrent/1160078/Avengers-age-of-Ultron-2015-HQ-CAM-H264-AC3-MURD3R/									
7	http://1337x.to/torrent/294349/American-Idol-S11E04-Auditions-4-HDTV-XviD-FQM-ettv/									
8	http://189.cn/dqmh/userCenter/myOrderInfoList.do?method=listMyOrderInfo_new&isVs=no									
9	http://2gis.ru/moscow/search/%D0%9F%D0%BE%D0%B5%D1%81%D1%82%D1%8C/tab/firms/zoom/11									
10	http://abc.go.com/shows/general-hospital/episode-guide/2015-05/08-friday-may-8-2015									
11	http://abc.go.com/shows/the-muppets/video/new-abc-comedy-trailers?cid=abchp_muppets									
12	http://abcnews.go.com/US/wireStory/regulators-delays-georgia-nuclear-plant-31020059									
13	http://adultfriendfinder.com/css/live_cd/ffadult/english/0/font_face-1427390957.css									
14	http://akhbarelyom.com/news/newdetails/410322/1/%D8%A8%D9%88%D8%B6%D9%88%D8%AD.html									
15	http://allegro.pl/amadeus-quartet-haydn-string-quartets-collectors-i5207998383.html									
16	http://allegro.pl/narzedzia-i-sprzet-warsztatowy-18554?ref=simplified-category-tree									
17	http://allegro.pl/royal-string-quartet-music-for-string-quartet-cd-i5262876831.html									
18	http://allegro.pl/sexy-g-string-stringi-z-koralikami-must-have-uni-i5039087215.html									
19	http://allegro.pl/sporty-strzeleckie-i-myslistwo-13495?ref=simplified-category-tree									
20	http://allegro.pl/sporty-towarzyskie-i-rekreacja-13408?ref=simplified-category-tree									
21	http://allegro.pl/triumph-miss-sexy-allday-string-stringi-36-s-bez-i4855636396.html									
22	http://allegro.pl/triumph-stringi-exquisite-essence-string-stal-38-i5073330901.html									
23	http://allegro.pl/wyszczuplajace-body-string-pod-biust-jony-srebra-i5124700240.html									

Figure 5.1 Dataset of Phishing URLs

Source: The Dataset is collected from an open-source service called Phish-Tank. This dataset consists of 5,000 random phishing URLs which are collected to train the ML models.



	A	B	C	D	E	F	G	H	I	J
1	phish_id	url	phish_det	submissio	verified	verificatio	online	target		
2	6911546	https://ja	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
3	6911545	http://po	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
4	6911536	https://sp	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
5	6911494	https://hy	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
6	6911483	http://sto	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
7	6911482	https://oc	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
8	6911481	https://tr	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
9	6911480	https://jo	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
10	6911467	https://su	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
11	6911466	https://cr	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
12	6911465	http://est	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
13	6911424	https://eb	http://ww	2021-01-0	yes	2021-01-0	yes	eBay, Inc.		
14	6911414	https://is	http://ww	2021-01-0	yes	2021-01-0	yes	Development Bank of Singa		
15	6911408	http://wir	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
16	6911403	http://ma	http://ww	2021-01-0	yes	2021-01-0	yes	ABSA Bank		
17	6911400	http://ww	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
18	6911398	https://bi	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
19	6911395	https://fg	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
20	6911394	http://fgh	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
21	6911389	https://wl	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
22	6911388	https://wl	http://ww	2021-01-0	yes	2021-01-0	yes	Other		
23	6911382	http://clfi	http://ww	2021-01-0	yes	2021-01-0	yes	Other		

Figure 5.2 Dataset of Legitimate URLs

Source: The Dataset were obtained from the open datasets of the University of New Brunswick, The dataset consists of collections of benign, spam, phishing, malware & defacement URLs. Out of all these types, the benign URL dataset is considered for this project. This dataset consists of 5,000 random legitimate URLs which are collected to train the ML models.

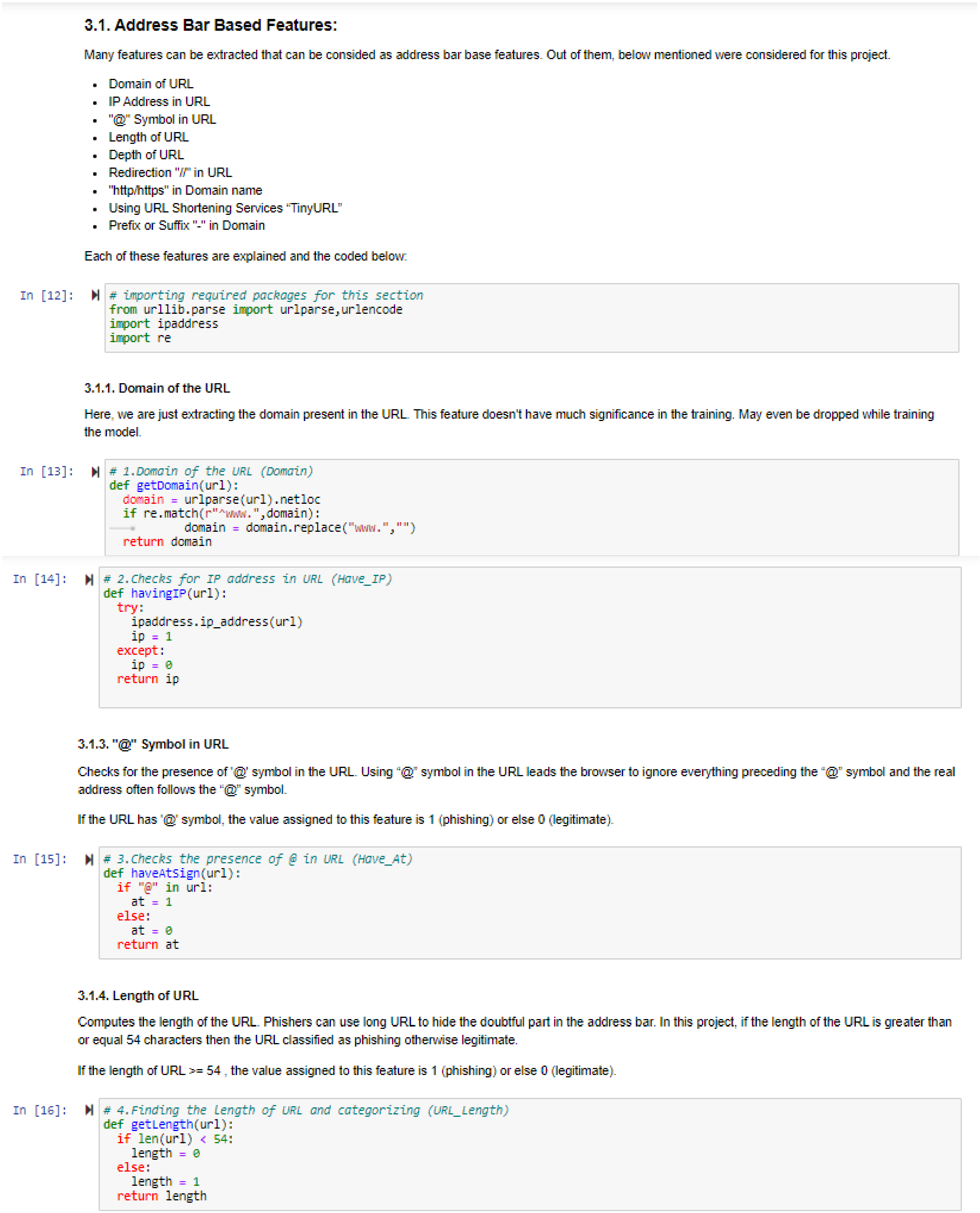
5.5 Feature extraction on the datasets

The features extraction used on the dataset are categorized into
i. Address bar based features

ii. Domain-based features

iii. Html & java-script based features

In figure 5.3, figure 5.4, and figure 5.5 the images show the list of code feature extraction done on the dataset while figure 4.6 shows the code computation for all the feature extraction used on the dataset.



3.1. Address Bar Based Features:

Many features can be extracted that can be consided as address bar base features. Out of them, below mentioned were considered for this project.

- Domain of URL
- IP Address in URL
- "@" Symbol in URL
- Length of URL
- Depth of URL
- Redirection "//" in URL
- "http/https" in Domain name
- Using URL Shortening Services “TinyURL”
- Prefix or Suffix "-" in Domain

Each of these features are explained and the coded below:

In [12]:
```
# importing required packages for this section
from urllib.parse import urlparse,urlencode
import ipaddress
import re
```

3.1.1. Domain of the URL

Here, we are just extracting the domain present in the URL. This feature doesn't have much significance in the training. May even be dropped while training the model.

In [13]:
```
# 1.Domain of the URL (Domain)
def getDomain(url):
  domain = urlparse(url).netloc
  if re.match(r"^www.",domain):
	       domain = domain.replace("www.","")
  return domain
```

In [14]:
```
# 2.Checks for IP address in URL (Have_IP)
def havingIP(url):
  try:
    ipaddress.ip_address(url)
    ip = 1
  except:
    ip = 0
  return ip
```

3.1.3. "@" Symbol in URL

Checks for the presence of '@' symbol in the URL. Using “@” symbol in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol.

If the URL has '@' symbol, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

In [15]:
```
# 3.Checks the presence of @ in URL (Have_At)
def haveAtSign(url):
  if "@" in url:
    at = 1
  else:
    at = 0
  return at
```

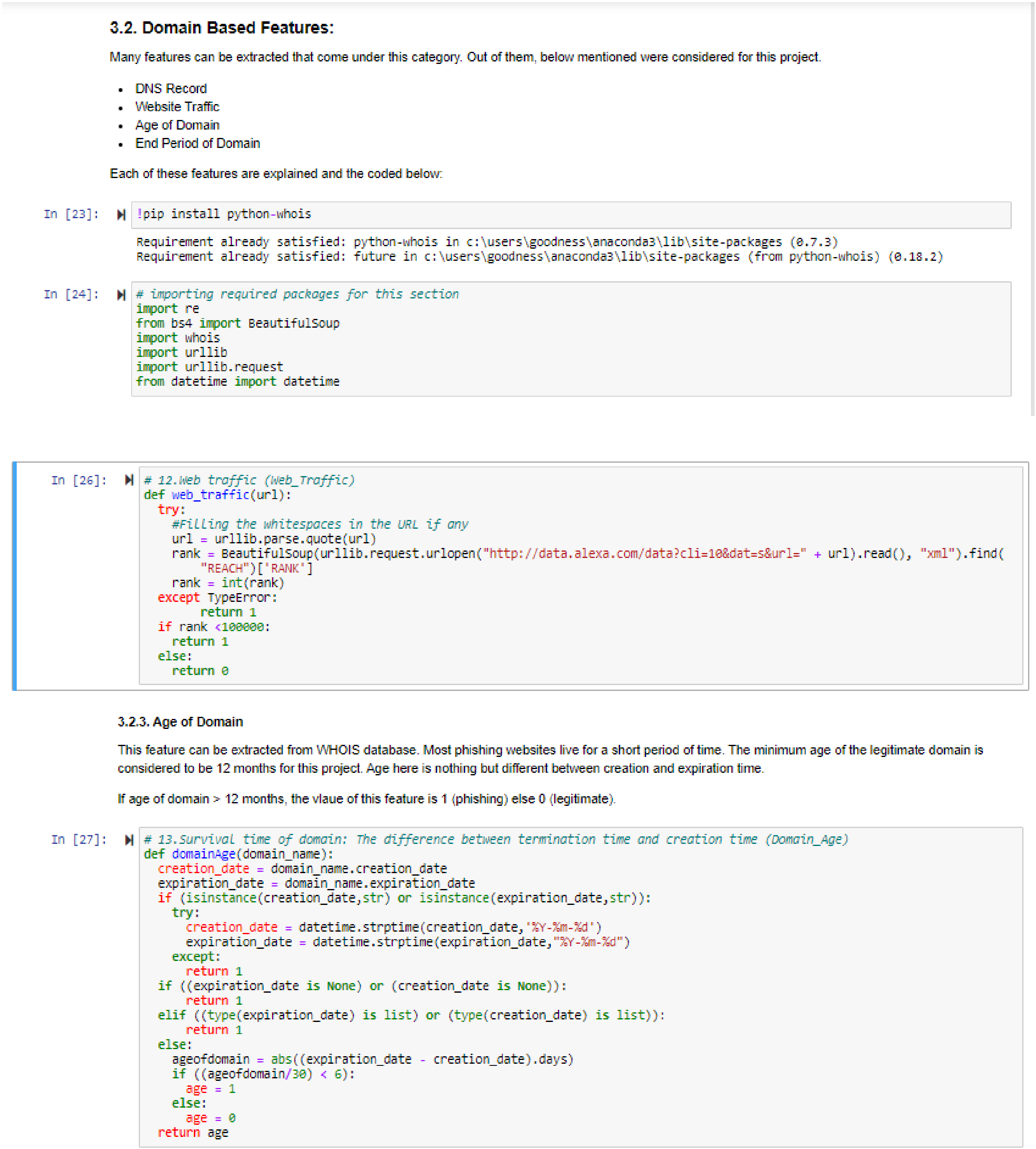
3.1.4. Length of URL

Computes the length of the URL. Phishers can use long URL to hide the doubtful part in the address bar. In this project, if the length of the URL is greater than or equal 54 characters then the URL classified as phishing otherwise legitimate.

If the length of URL >= 54 , the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

In [16]:
```
# 4.Finding the length of URL and categorizing (URL_Length)
def getLength(url):
  if len(url) < 54:
    length = 0
  else:
    length = 1
  return length
```

Figure 5.3: Code for Address bar based feature extraction



3.2. Domain Based Features:

Many features can be extracted that come under this category. Out of them, below mentioned were considered for this project.

- DNS Record
- Website Traffic
- Age of Domain
- End Period of Domain

Each of these features are explained and the coded below:

```
In [23]:  !pip install python-whois
```

```
Requirement already satisfied: python-whois in c:\users\goodness\anaconda3\lib\site-packages (0.7.3)
Requirement already satisfied: future in c:\users\goodness\anaconda3\lib\site-packages (from python-whois) (0.18.2)
```

```
In [24]:  # importing required packages for this section
          import re
          from bs4 import BeautifulSoup
          import whois
          import urllib
          import urllib.request
          from datetime import datetime
```

```
In [26]:  # 12.Web traffic (Web_Traffic)
          def web_traffic(url):
            try:
              #Filling the whitespaces in the URL if any
              url = urllib.parse.quote(url)
              rank = BeautifulSoup(urllib.request.urlopen("http://data.alexa.com/data?cli=10&dat=s&url=" + url).read(), "xml").find(
                  "REACH")['RANK']
              rank = int(rank)
            except TypeError:
                  return 1
            if rank <100000:
              return 1
            else:
              return 0
```

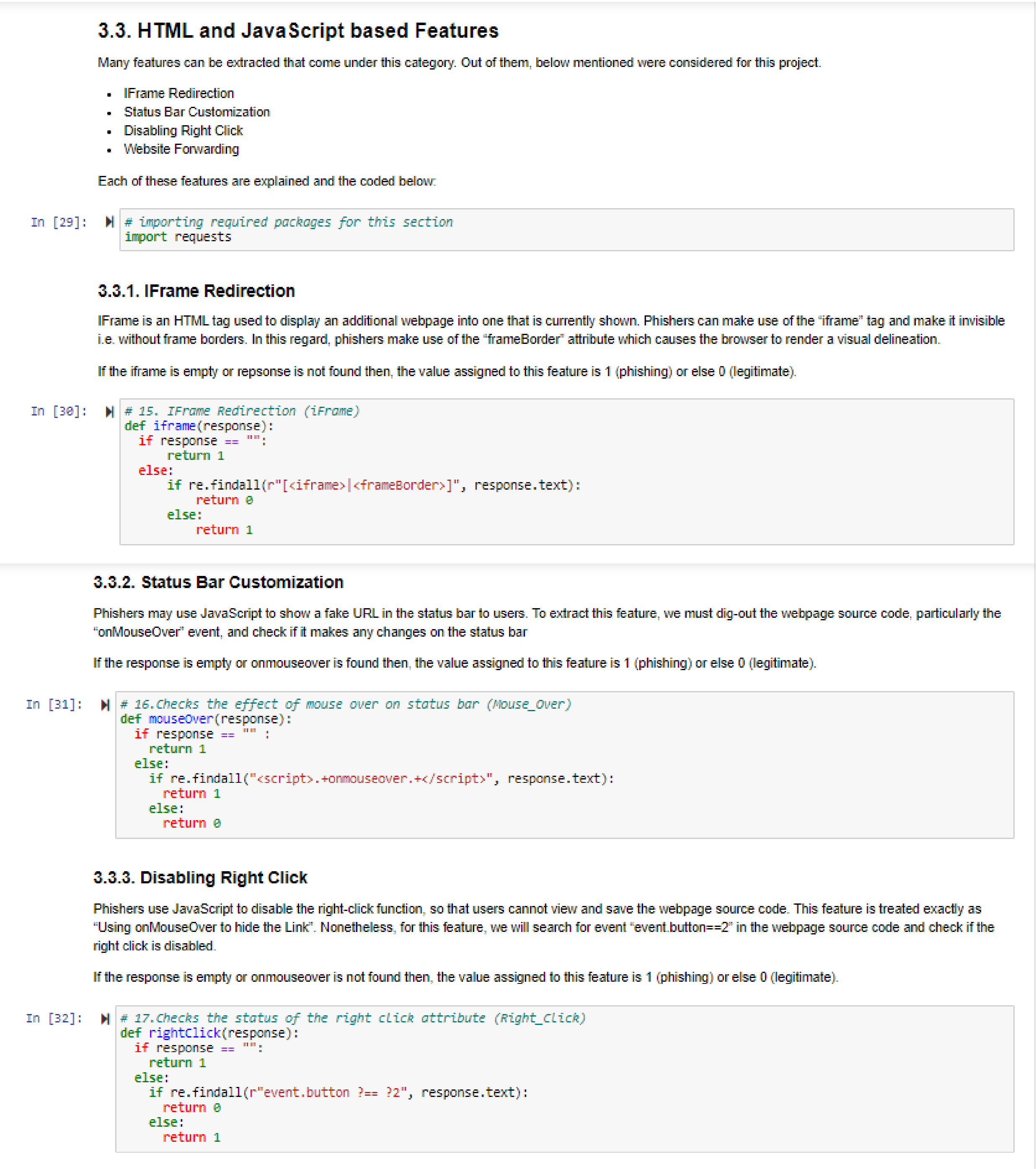
3.2.3. Age of Domain

This feature can be extracted from WHOIS database. Most phishing websites live for a short period of time. The minimum age of the legitimate domain is considered to be 12 months for this project. Age here is nothing but different between creation and expiration time.

If age of domain > 12 months, the vlaue of this feature is 1 (phishing) else 0 (legitimate).

```
In [27]:  # 13.Survival time of domain: The difference between termination time and creation time (Domain_Age)
          def domainAge(domain_name):
            creation_date = domain_name.creation_date
            expiration_date = domain_name.expiration_date
            if (isinstance(creation_date,str) or isinstance(expiration_date,str)):
              try:
                creation_date = datetime.strptime(creation_date,'%Y-%m-%d')
                expiration_date = datetime.strptime(expiration_date,"%Y-%m-%d")
              except:
                return 1
            if ((expiration_date is None) or (creation_date is None)):
                return 1
            elif ((type(expiration_date) is list) or (type(creation_date) is list)):
                return 1
            else:
              ageofdomain = abs((expiration_date - creation_date).days)
              if ((ageofdomain/30) < 6):
                age = 1
              else:
                age = 0
            return age
```

Figure 5.4: Code for domain-based features extraction



3.3. HTML and JavaScript based Features

Many features can be extracted that come under this category. Out of them, below mentioned were considered for this project.

- IFrame Redirection
- Status Bar Customization
- Disabling Right Click
- Website Forwarding

Each of these features are explained and the coded below:

In [29]:
```
# importing required packages for this section
import requests
```

3.3.1. IFrame Redirection

IFrame is an HTML tag used to display an additional webpage into one that is currently shown. Phishers can make use of the "iframe" tag and make it invisible i.e. without frame borders. In this regard, phishers make use of the "frameBorder" attribute which causes the browser to render a visual delineation.

If the iframe is empty or repsonse is not found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

In [30]:
```
# 15. IFrame Redirection (iFrame)
def iframe(response):
  if response == "":
      return 1
  else:
      if re.findall(r"[<iframe>|<frameBorder>]", response.text):
          return 0
      else:
          return 1
```

3.3.2. Status Bar Customization

Phishers may use JavaScript to show a fake URL in the status bar to users. To extract this feature, we must dig-out the webpage source code, particularly the "onMouseOver" event, and check if it makes any changes on the status bar

If the response is empty or onmouseover is found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

In [31]:
```
# 16.Checks the effect of mouse over on status bar (Mouse_Over)
def mouseOver(response):
  if response == "" :
    return 1
  else:
    if re.findall("<script>.+onmouseover.+</script>", response.text):
      return 1
    else:
      return 0
```

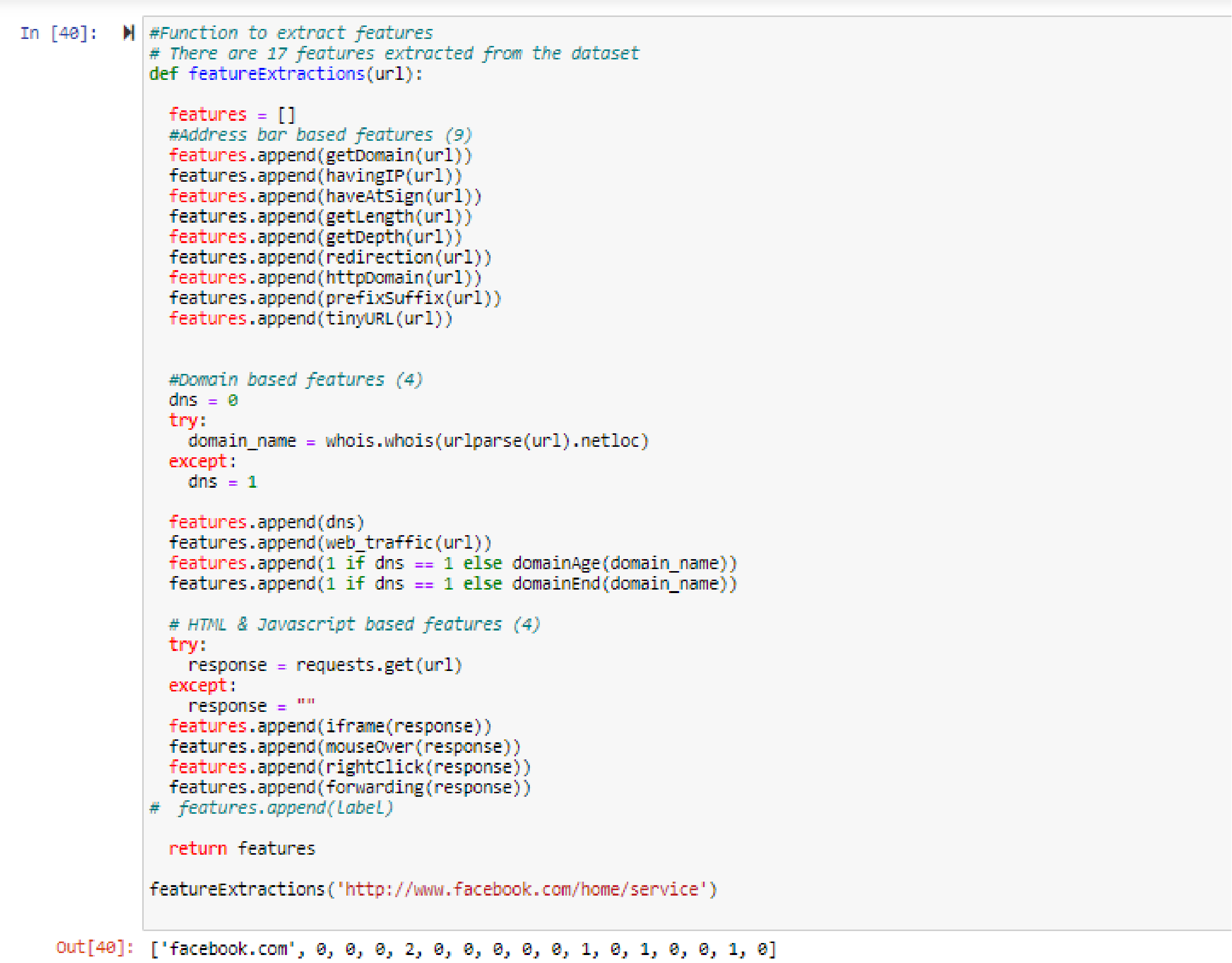
3.3.3. Disabling Right Click

Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. This feature is treated exactly as "Using onMouseOver to hide the Link". Nonetheless, for this feature, we will search for event "event.button==2" in the webpage source code and check if the right click is disabled.

If the response is empty or onmouseover is not found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

In [32]:
```
# 17.Checks the status of the right click attribute (Right_Click)
def rightClick(response):
  if response == "":
    return 1
  else:
    if re.findall(r"event.button ?== ?2", response.text):
      return 0
    else:
      return 1
```

Figure 5.5: Code for Html & java-script based features extraction



```
In [40]:
#Function to extract features
# There are 17 features extracted from the dataset
def featureExtractions(url):

  features = []
  #Address bar based features (9)
  features.append(getDomain(url))
  features.append(havingIP(url))
  features.append(haveAtSign(url))
  features.append(getLength(url))
  features.append(getDepth(url))
  features.append(redirection(url))
  features.append(httpDomain(url))
  features.append(prefixSuffix(url))
  features.append(tinyURL(url))


  #Domain based features (4)
  dns = 0
  try:
    domain_name = whois.whois(urlparse(url).netloc)
  except:
    dns = 1

  features.append(dns)
  features.append(web_traffic(url))
  features.append(1 if dns == 1 else domainAge(domain_name))
  features.append(1 if dns == 1 else domainEnd(domain_name))

  # HTML & Javascript based features (4)
  try:
    response = requests.get(url)
  except:
    response = ""
  features.append(iframe(response))
  features.append(mouseOver(response))
  features.append(rightClick(response))
  features.append(forwarding(response))
#  features.append(label)

  return features

featureExtractions('http://www.facebook.com/home/service')

Out[40]: ['facebook.com', 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0]
```

Figure 5.6: Code computation for all the feature extraction used dataset.

5.6. Data Analysis & Visualization

The image as shown in figure 5.7 shows the distribution plot of how legitimate and phishing datasets are distributed base on the features selected and how they are related to each other.
In figure 5.8 shows the plot of a correlation heat-map of the dataset. The plot shows correlation between different variables in the dataset.
In figure 5.9 and figure 5.10, it shows the feature importance in the model for Decision tree classifier and Random forest classifier respectively.

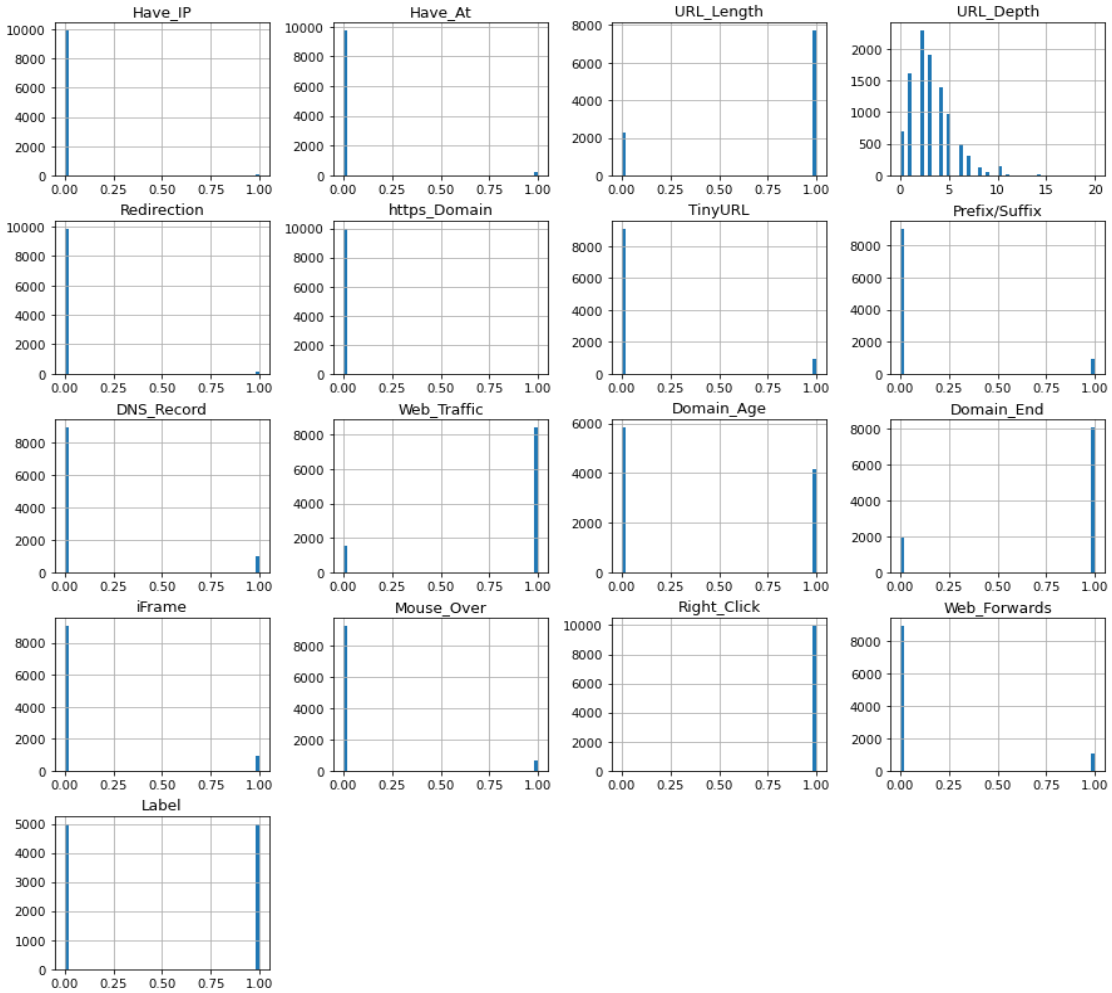


Figure 5.7: Distribution plot of dataset base on the features selected

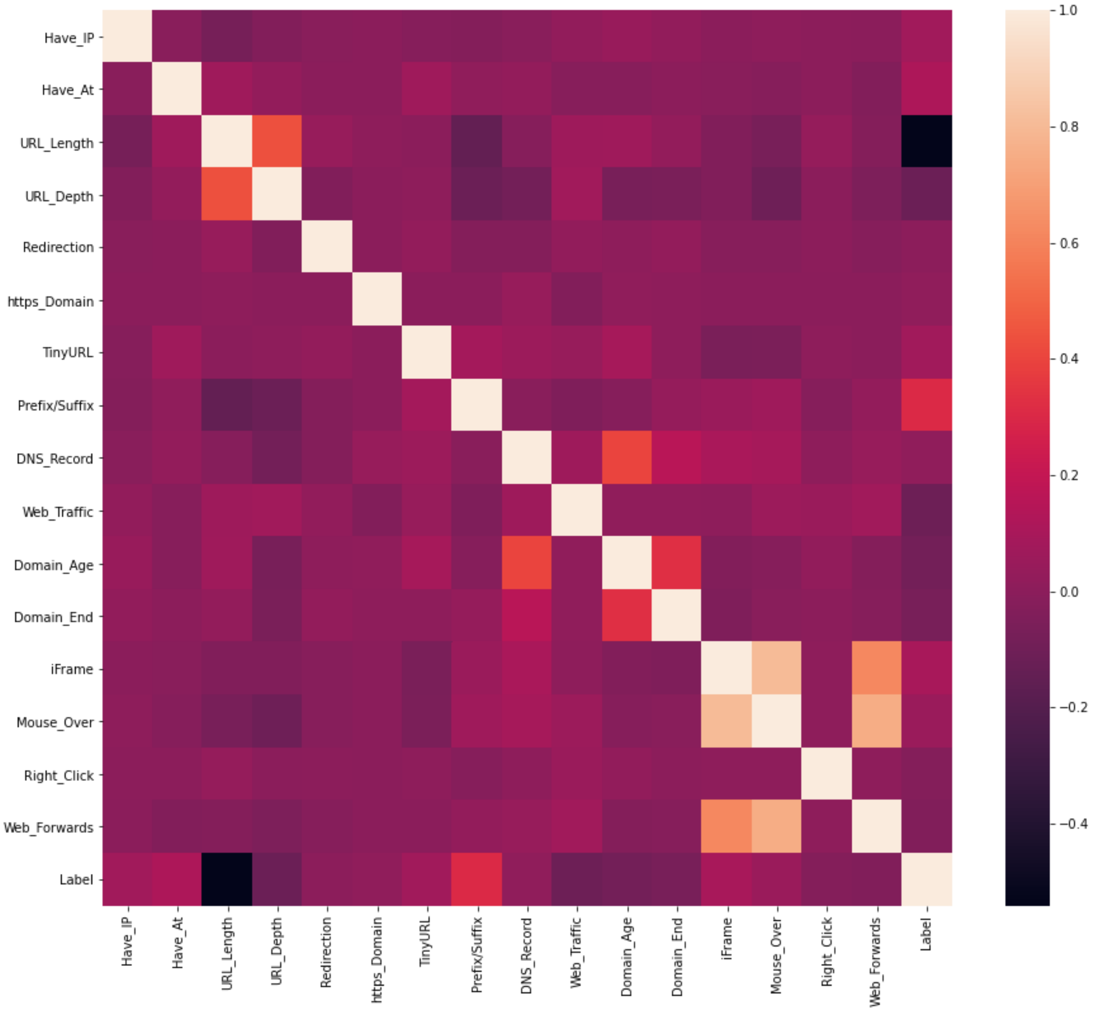


Figure 5.8: Correlation heat map of the dataset

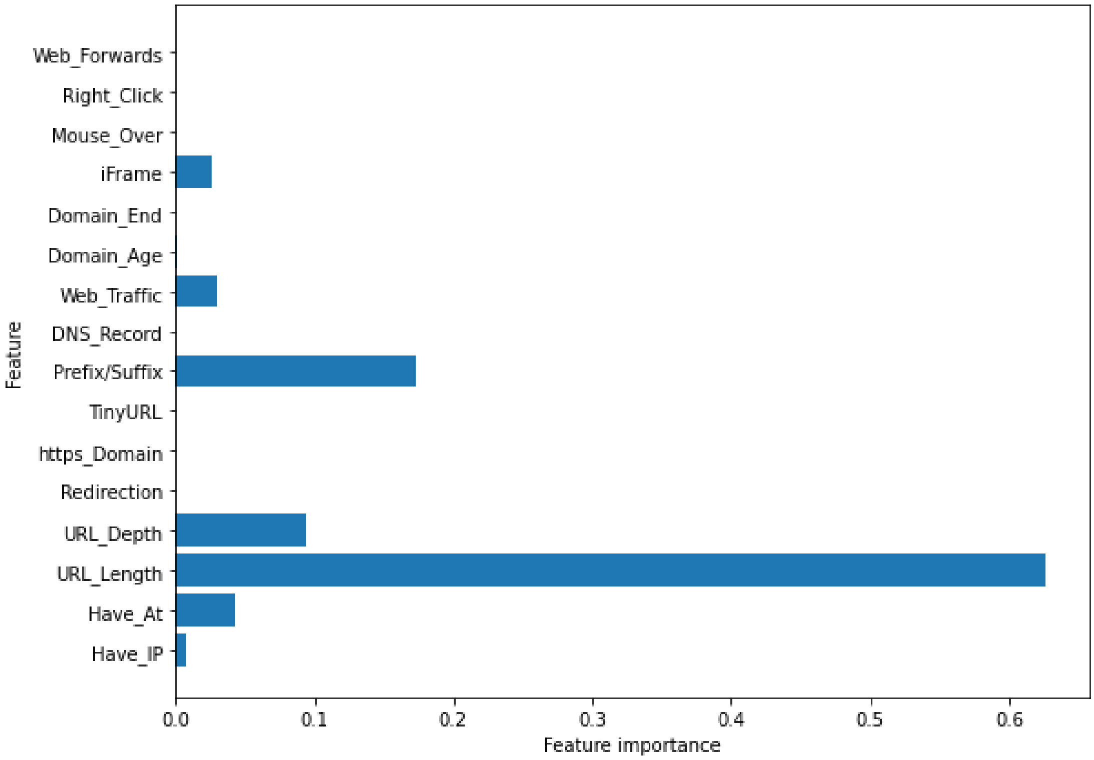


Figure 5.9: Feature importance for Decision Tree classifier

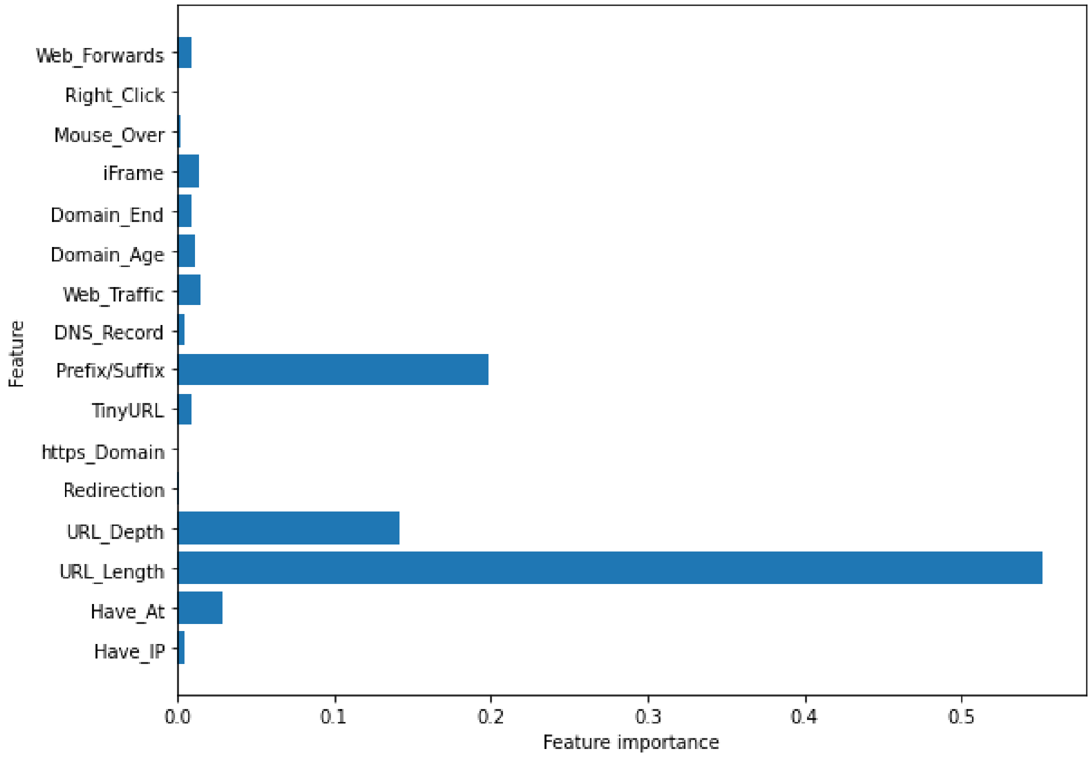
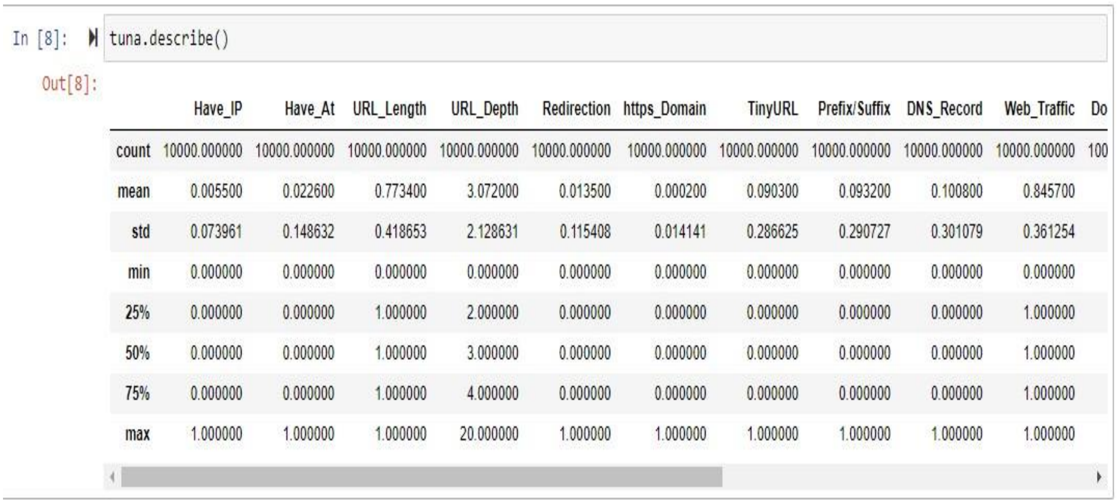


Figure 5.10: Feature importance for Random forest classifier

5.7 Data pre-processing

The datasets were first cleaned to remove empty entries and fill some entries by applying data pre-processing techniques and transform the data to use in the models. Figure 5.11 shows the summary of the dataset while figure 5.12 shows the number of missing values in the dataset which all appear to be zero.

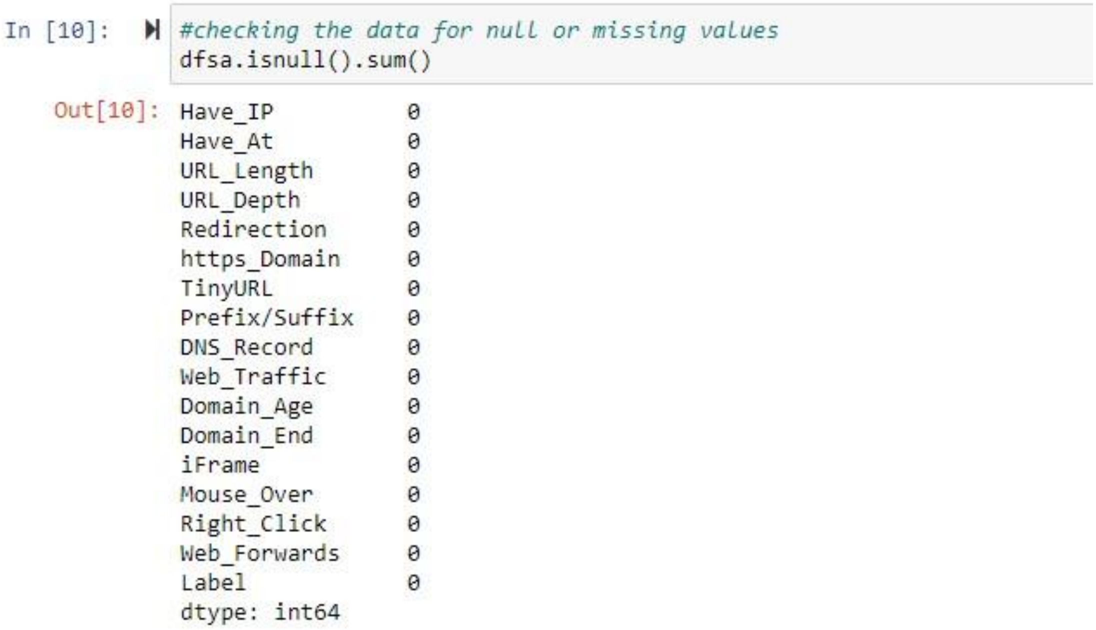


```
In [8]:  tuna.describe()
```

Out[8]:

	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	DNS_Record	Web_Traffic	Do
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	100
mean	0.005500	0.022600	0.773400	3.072000	0.013500	0.000200	0.090300	0.093200	0.100800	0.845700	
std	0.073961	0.148632	0.418653	2.128631	0.115408	0.014141	0.286625	0.290727	0.301079	0.361254	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	
50%	0.000000	0.000000	1.000000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	
75%	0.000000	0.000000	1.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	
max	1.000000	1.000000	1.000000	20.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

Figure 5.11: Summary of the dataset

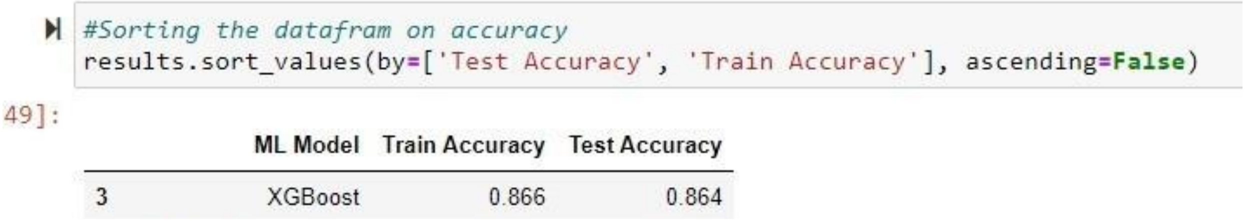


```
In [10]:  #checking the data for null or missing values
          dfsa.isnull().sum()

Out[10]:  Have_IP          0
          Have_At          0
          URL_Length       0
          URL_Depth        0
          Redirection      0
          https_Domain     0
          TinyURL          0
          Prefix/Suffix    0
          DNS_Record       0
          Web_Traffic      0
          Domain_Age       0
          Domain_End       0
          iFrame           0
          Mouse_Over       0
          Right_Click      0
          Web_Forwards     0
          Label            0
          dtype: int64
```

Figure 5.12: Number of missing values in the dataset

5.8 Phishing detection model

The based methodology stated that the proposed system utilizes machine learning model and deep neural networks. The models determine whether a website URL is phishing or legitimate. The models help give a 2-class prediction (legitimate (0) and phishing (1)). In the model development process, Here are the model, their accuracy was tested using sklearn matrices with an accuracy score and their matrices are shown in figure 5.13. The XGBooster model had the highest performance score of 86.6.

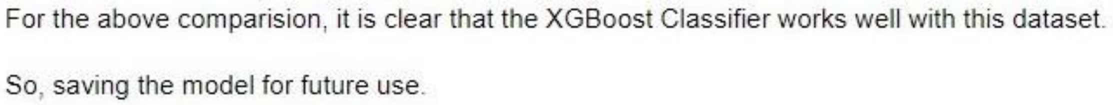


```
#Sorting the datafram on accuracy
results.sort_values(by=['Test Accuracy', 'Train Accuracy'], ascending=False)
```

49]:

	ML Model	Train Accuracy	Test Accuracy
3	XGBoost	0.866	0.864

Figure 5.13: Accuracy performance of models



For the above comparision, it is clear that the XGBoost Classifier works well with this dataset.

So, saving the model for future use.

5.9 General Working of The System

A one-page phishing detection web application called “Phish-Buster” has been developed to run on any browser. The application was developed using programming languages such as HTML, CSS, PHP, and JavaScript.
The phishing detection web application has the following pages:

5.10 The home page

The home page contains a session for a user to enter a URL and predict if it is phishing or legitimate. It predicts the state of the URL base on the feature selection as shown in figure 5.14.
The purpose of this page is to help its users validate a URL link and also provide various resources on phishing attacks. The User can also take a google phishing test to help understand how to detect phishing messages and URLs. Also, users can download a book that contains information and other resources on phishing.

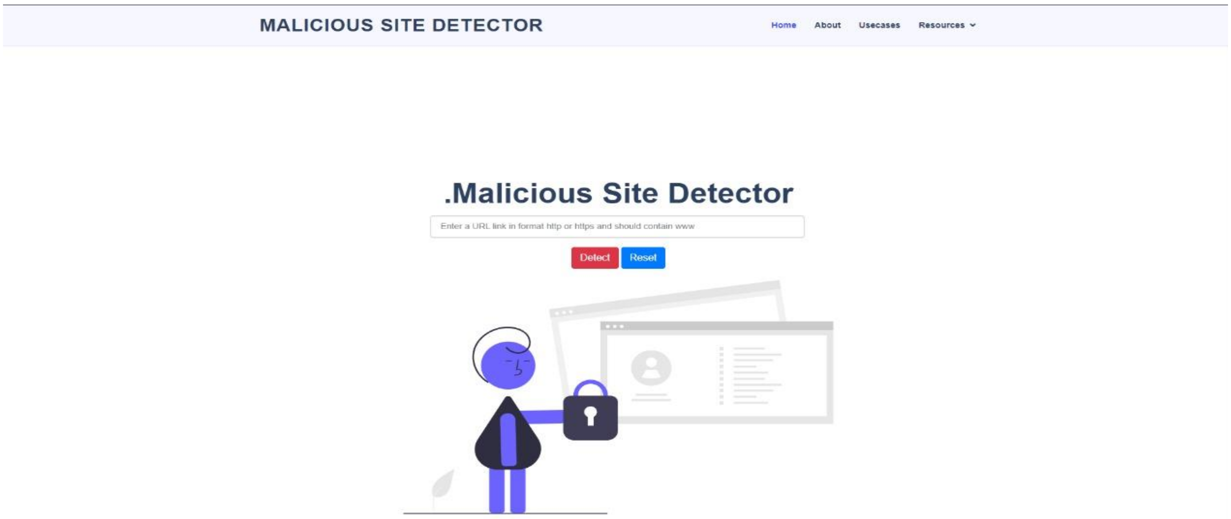
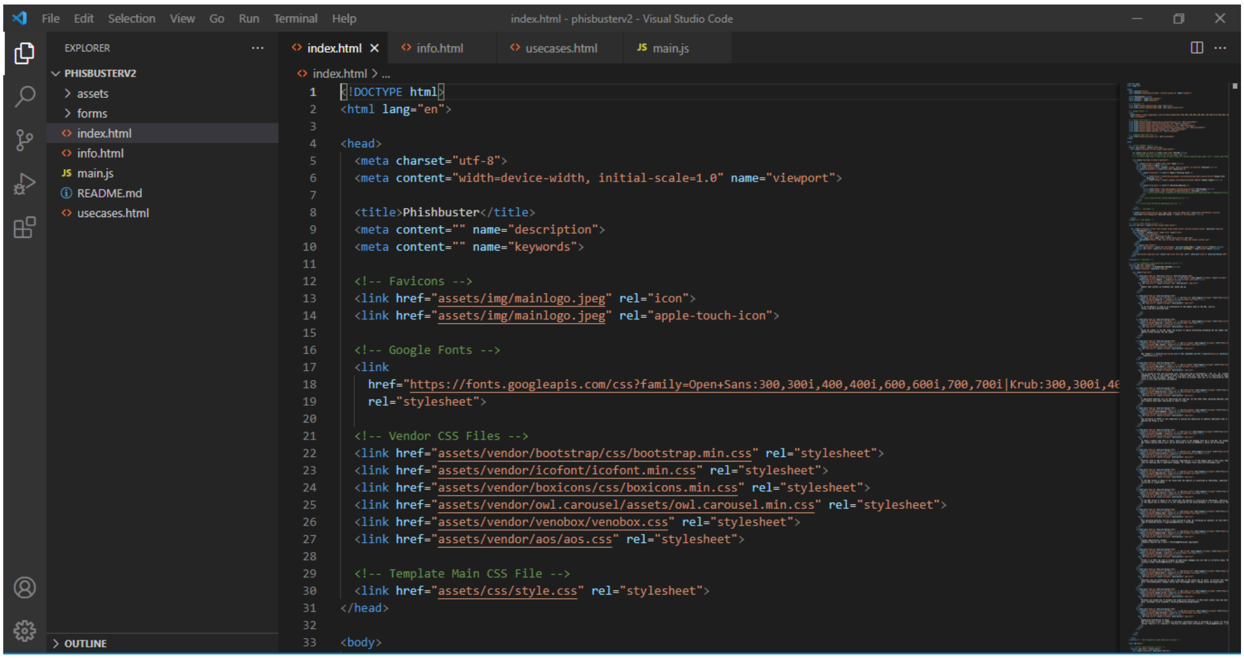


Figure 5.14 (b): The home page footer

5.11 Web application Source Code

As shown from figure 5.18, consists of pages of source code of the web application running on visual studio code, other source codes are shown in Appendix A

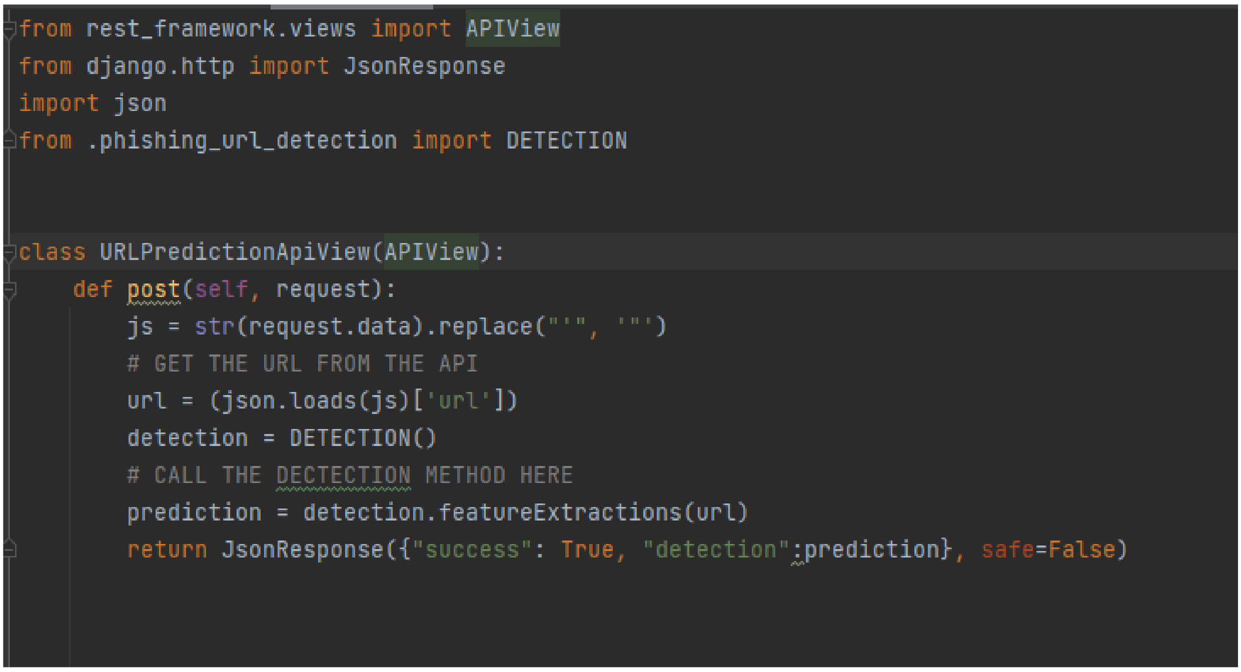


```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Phishbuster</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/mainlogo.jpeg" rel="icon">
  <link href="assets/img/mainlogo.jpeg" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link
    href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Krub:300,300i,4
    rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/icofont/icofont.min.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/owl.carousel/assets/owl.carousel.min.css" rel="stylesheet">
  <link href="assets/vendor/venobox/venobox.css" rel="stylesheet">
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">

  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">
</head>

<body>
```

Figure 5.15: Code for the web application

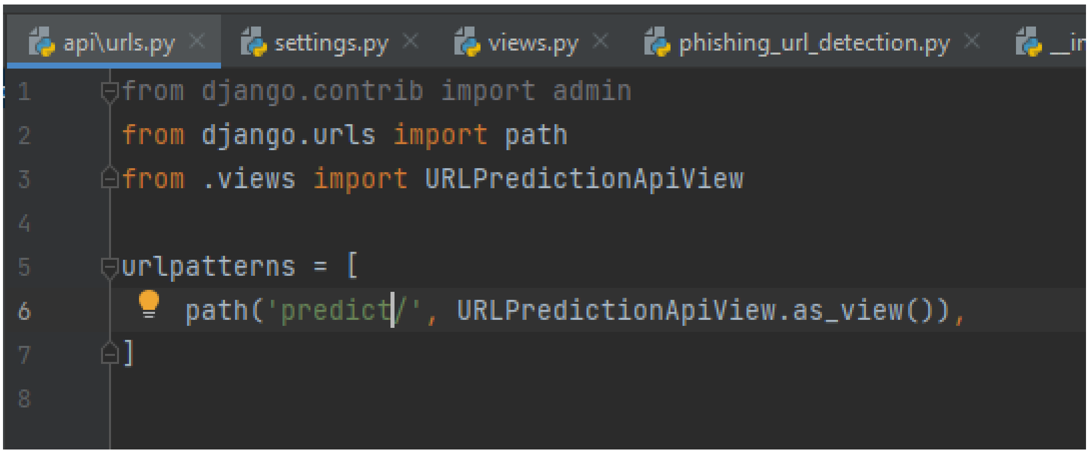
5.12 API (Application Programming Interface)

The work of an API here is that it serves as an intermediary between the web server and web application. A python framework called Django was used on the model prediction on the web application. These two ends communicate using a JSON (Javascript Object Notation) to send a request and receive a response.



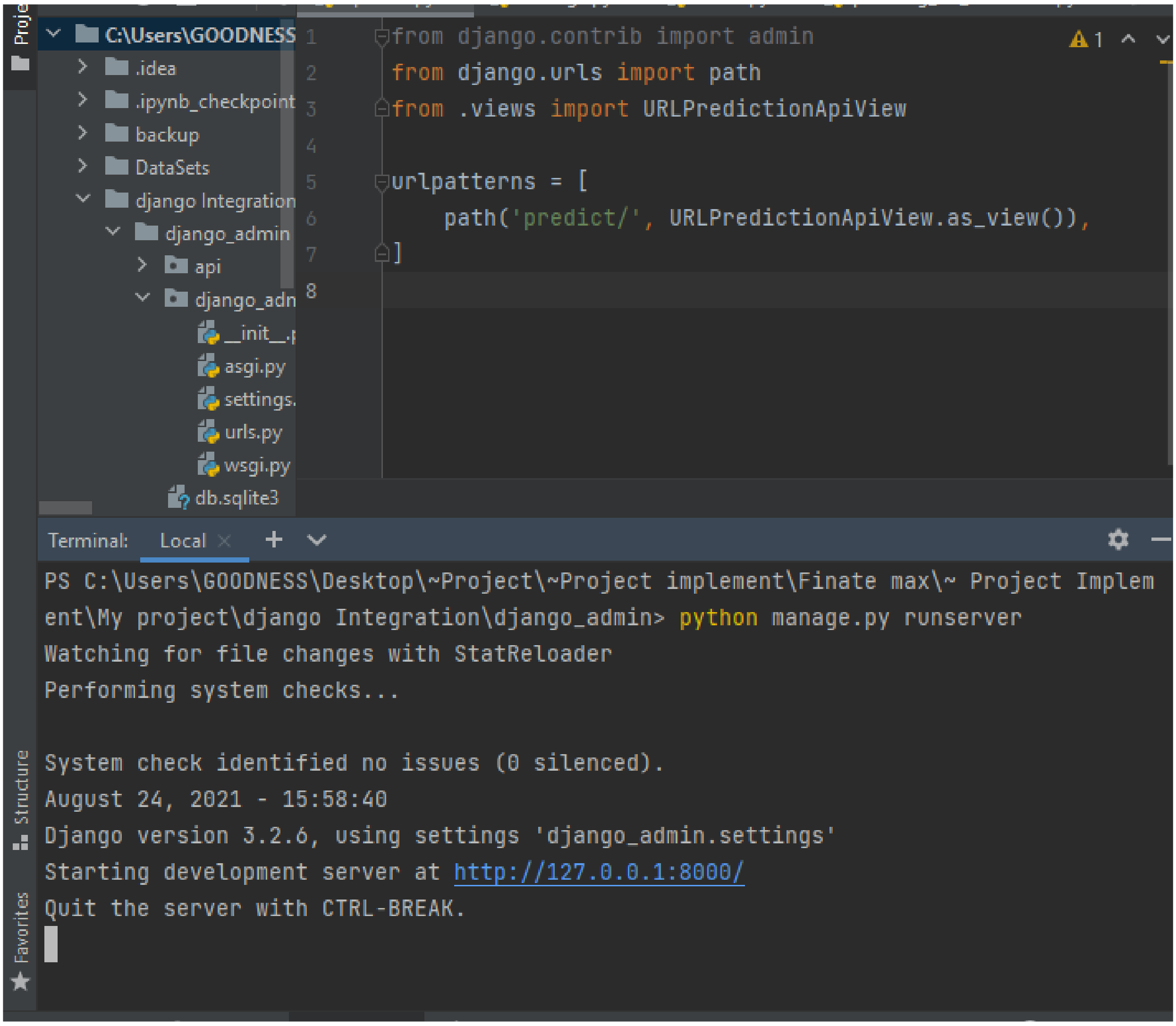
```python
from rest_framework.views import APIView
from django.http import JsonResponse
import json
from .phishing_url_detection import DETECTION


class URLPredictionApiView(APIView):
    def post(self, request):
        js = str(request.data).replace("'", '"')
        # GET THE URL FROM THE API
        url = (json.loads(js)['url'])
        detection = DETECTION()
        # CALL THE DECTECTION METHOD HERE
        prediction = detection.featureExtractions(url)
        return JsonResponse({"success": True, "detection":prediction}, safe=False)
```

Figure 5.15: python code to send a request on JSON



```python
from django.contrib import admin
from django.urls import path
from .views import URLPredictionApiView

urlpatterns = [
    path('predict/', URLPredictionApiView.as_view()),
]

```

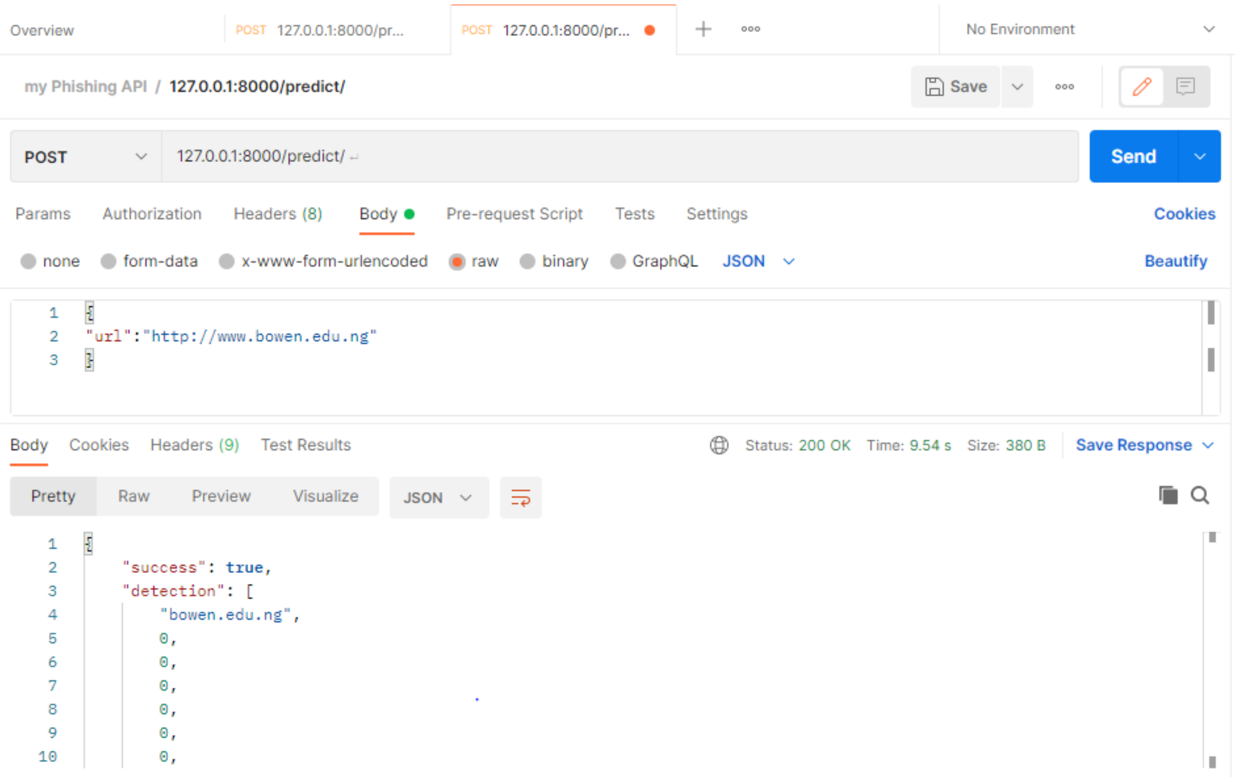
Figure 5.16: Code for API URL



```python
from django.contrib import admin
from django.urls import path
from .views import URLPredictionApiView

urlpatterns = [
    path('predict/', URLPredictionApiView.as_view()),
]

```

```
PS C:\Users\GOODNESS\Desktop\~Project\~Project implement\Finate max\~ Project Implem
ent\My project\django Integration\django_admin> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 24, 2021 - 15:58:40
Django version 3.2.6, using settings 'django_admin.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figure 5.17: Executing web-app on Django local server



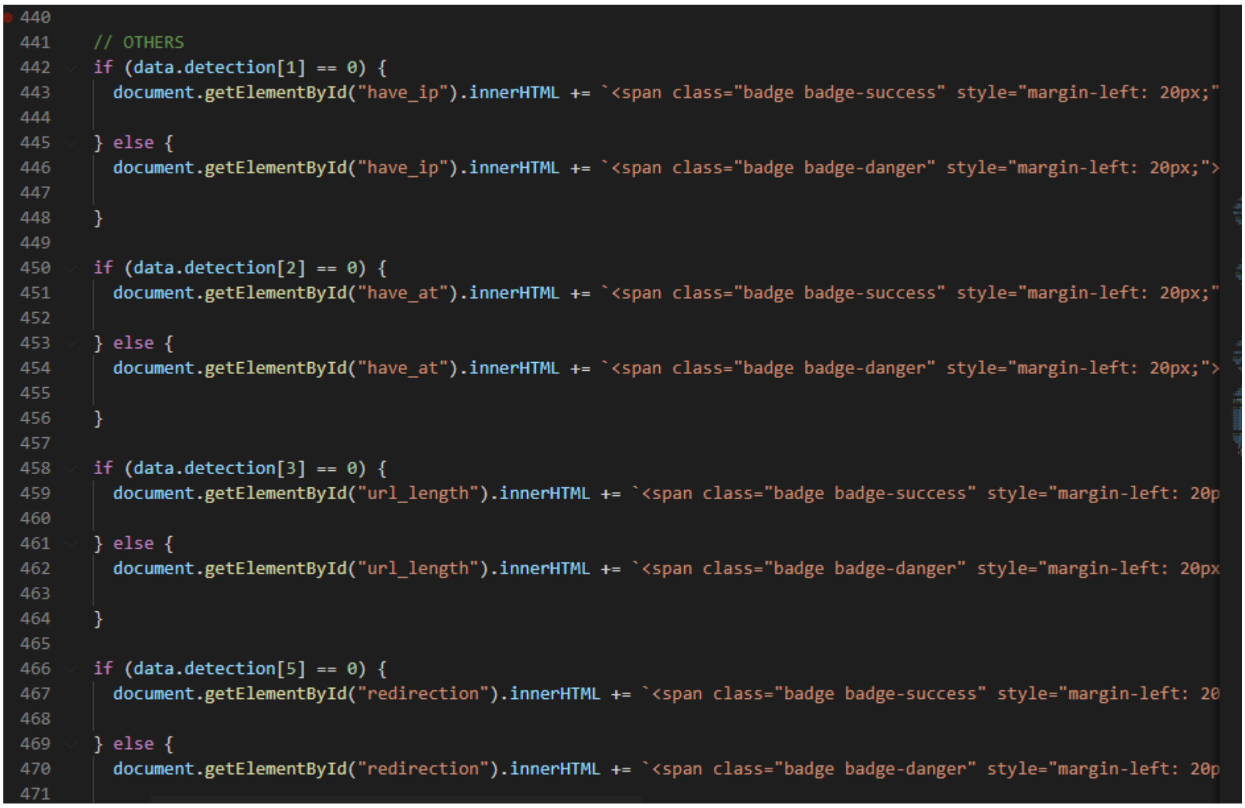
POST 127.0.0.1:8000/predict/

```
{
"url":"http://www.bowen.edu.ng"
}
```

Status: 200 OK Time: 9.54 s Size: 380 B

```
{
    "success": true,
    "detection": [
        "bowen.edu.ng",
        0,
        0,
        0,
        0,
        0,
        0,
```

Figure 5.18: Testing the API link on postman



```

// OTHERS
if (data.detection[1] == 0) {
  document.getElementById("have_ip").innerHTML += `<span class="badge badge-success" style="margin-left: 20px;"

} else {
  document.getElementById("have_ip").innerHTML += `<span class="badge badge-danger" style="margin-left: 20px;">

}

if (data.detection[2] == 0) {
  document.getElementById("have_at").innerHTML += `<span class="badge badge-success" style="margin-left: 20px;"

} else {
  document.getElementById("have_at").innerHTML += `<span class="badge badge-danger" style="margin-left: 20px;">

}

if (data.detection[3] == 0) {
  document.getElementById("url_length").innerHTML += `<span class="badge badge-success" style="margin-left: 20p

} else {
  document.getElementById("url_length").innerHTML += `<span class="badge badge-danger" style="margin-left: 20px

}

if (data.detection[5] == 0) {
  document.getElementById("redirection").innerHTML += `<span class="badge badge-success" style="margin-left: 20

} else {
  document.getElementById("redirection").innerHTML += `<span class="badge badge-danger" style="margin-left: 20p

```

Figure 5.19: JavaScript code for linking API

5.13 Proposed System

The proposed phishing detection system utilizes machine learning models and deep neural networks. The system comprises two major parts, which are the machine learning models and a web application. These models consist of Decision Tree, Support Vector Machine, XGBooster, Multilayer Perceptions, Auto Encoder Neural Network, and Random Forest. These models are selected after different comparison-based performances of multiple machine learning algorithms. Each of these models is trained and tested on a website content-based feature, extracted from both phishing and legitimate dataset. Hence, the model with the highest accuracy is selected and integrated into a web application that will enable a user to predict if a URL link is phishing or legitimate.

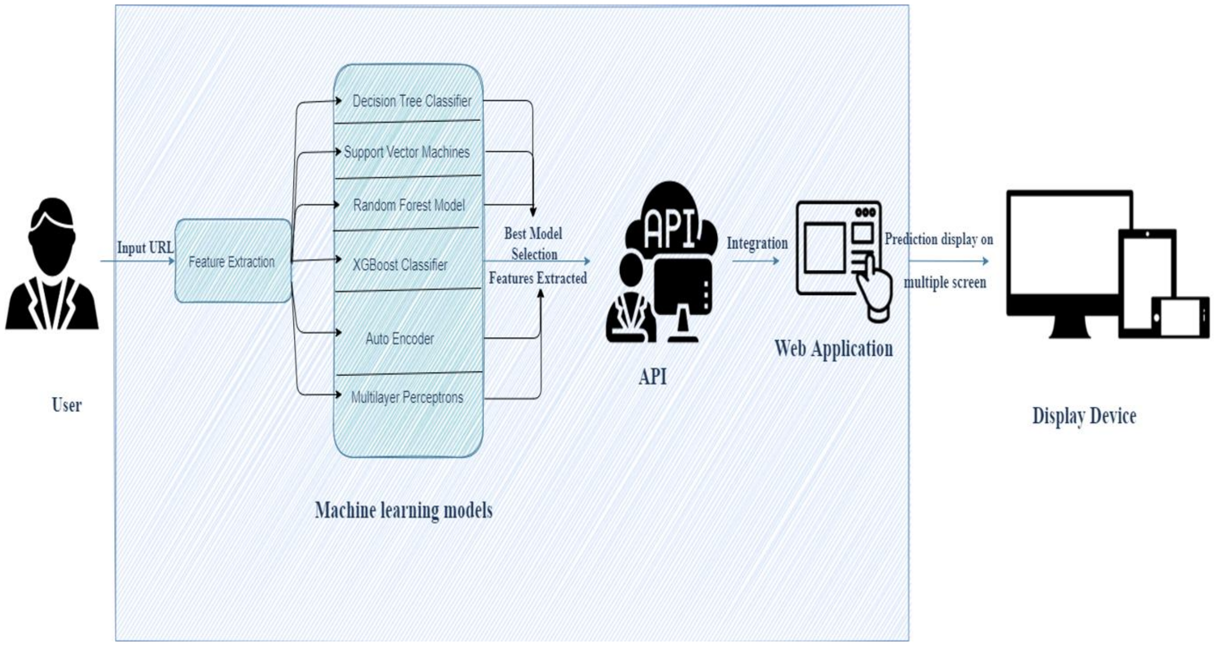


Figure 5.22: Architectural Design of the Proposed System

\

CHAPTER 6

Conclusion

6.Conclusion

The system developed detects if a URL link is phishing or legitimate by using machine learning models and deep neural network algorithms. The feature extraction and the models used on the dataset helped to uniquely identify phishing URLs and also the performance accuracy of the models used. It is also surprisingly accurate at detecting the genuineness of a URL link.

References

Journal Papers –

[1] Kathrine, G. Jaspher Willsie, Paradise Mercy Praise, A. Amrutha Rose, and Eligious C. Kalaivani. "Variants of phishing attacks and their detection techniques." In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 255-259. IEEE, 2019.

[2] Alswailem, Amani, Bashayr Alabdullah, Norah Alrumayh, and Aram Alsedrani. "Detecting phishing websites using machine learning." In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1-6. IEEE, 2019.

[3] Patil, Vaibhav, Pritesh Thakkar, Chirag Shah, Tushar Bhat, and S. P. Godse. "Detection and prevention of phishing websites using machine learning approach." In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pp. 1-5. Ieee, 2018.

[4] Harinahalli Lokesh, Gururaj, and Goutham BoreGowda. "Phishing website detection based on effective machine learning approach." *Journal of Cyber Security Technology* 5, no. 1 (2021): 1-14.

[5] Yang, Peng, Guangzhen Zhao, and Peng Zeng. "Phishing website detection based on multidimensional features driven by deep learning." *IEEE access* 7 (2019): 15196-15209.

Useful Links:-

https://machinelearningmastery.com/save-gradient-boosting-models-xgboost-python/

https://github.com/shreyagopal/t81_558_deep_learning/blob/master/t81_558_class_14_03_anomaly.ipynb

https://mc.ai/a-beginners-guide-to-build-stacked-autoencoder-and-tying-weights-with-it/

https://www.irjet.net/archives/V8/i4/IRJET-V8I4274.pdf