

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum-590 014, Karnataka.



An
Internship Report
On

“DETECTION OF PHISHING WEBSITES” USING MACHINE LEARNING

Submitted in the partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING

Submitted by

**AKSHAY
ARAVIND
(1EW18IS006)**

Under the Guidance of

Mrs. Shruthi T. V
Asst.Prof Dept. of ISE
EWIT, Bangalore



DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

EAST WEST INSTITUTE OF TECHNOLOGY

BANGALORE - 560 091

2021-2022

EAST WEST INSTITUTE OF TECHNOLOGY

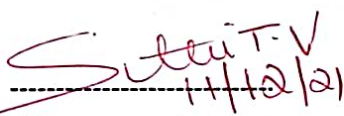
Sy. No.63, Off. Magadi Road, Vishwaneedam Post, Bangalore - 560 091
(Affiliated to Visvesvaraya Technological University, Belgaum)


DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

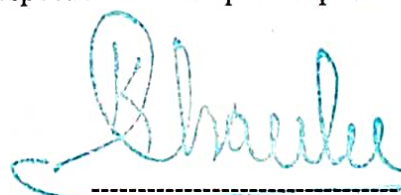


CERTIFICATE

This is to certify that the Internship work entitled “**DETECTION OF PHISHING WEBSITES**” presented by **AKSHAY ARAVIND (1EW18IS006)**, Bonafede student of **EAST WEST INSTITUTE OF TECHNOLOGY**, Bangalore in partial fulfillment for the award of **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belgaum** during the year **2021-2022**. It is certified that all corrections/suggestions indicated have been incorporated in the report. The internship work has been approved as it satisfies the academic requirements in respect of internship work prescribed for the said degree.


Signature of Guide
Mrs. Shruthi T. V
Asst.Prof, Dept. of ISE
EWIT, Bangalore


Signature of HOD
Dr. Suresh M B
Prof & Head, Dept. of ISE
EWIT, Bangalore


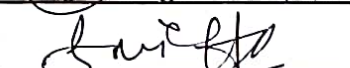

Signature of Principal
Dr. K Channakeshavalu
Principal
EWIT, Bangalore

External Viva

Name of the Examiners

1. Gnanakumari. b
2. Sarthe p

Signature with date



25/2/2022

CERTIFICATE FROM THE ORGANIZATION



CERTIFICATE

This is to certify that **Mr. AKSHAY ARAVIND** bearing registration number **1EW18IS006**, a student of **INFORMATION SCIENCE & ENGINEERING**, from **EAST WEST INSTITUTE OF TECHNOLOGY, BANGALORE** has successfully completed his internship on "**MACHINE LEARNING & PYTHON**" in our company. The duration of his internship was from **1ST SEPTEMBER 2021** to **1ST OCTOBER 2021**.

During this tenure, he has shown keen interest in learning. He was also enthusiastic and proactive in understanding the concepts.

For **MINDSET IT SOLUTION**,


HARISH S
PROJECT MANAGER

A circular stamp with the text "MINDSET IT SOLUTIONS" around the top inner edge and "BANGALORE" around the bottom inner edge. There are two small stars on either side of the word "BANGALORE". The center of the stamp contains a dotted line.

MINDSET IT SOLUTIONS & CONSULTANTS

#2/E, 2nd Floor, 14th Main Road, Vijayanagar, Bangalore – 40, Opp. Vijayanagar Metro Station Next To Police station, Mob: +91 9844628808, +91 7676851841, www.mindsetit.in

EAST WEST INSTITUTE OF TECHNOLOGY

Sy. No.63, Off. Magadi Road, Vishwaneedam Post, Bangalore - 560 091
(Affiliated to Visvesvaraya Technological University, Belgaum)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING



DECLARATION

I, **AKSHAY ARAVIND**, Student of Seventh Semester B.E ,in the Department of Information Science and Engineering, **East West Institute of Technology**, Bangalore hereby declare that the internship entitled "**DETECTION OF PHISHING WEBSITES**" using **MACHINE LEARNING** has been carried out by me and submitted in partial fulfillment of course requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** discipline of **Visvesvaraya Technological University**, Belgaum during the academic year **2021- 2022**. Further, the matter embodied in internship report has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Place: Bangalore

Date:

NAME: AKSHAY ARAVIND

USN: 1EW18IS006

ABSTRACT

Phishing is a cyber-attack which targets naive online users tricking into revealing sensitive information such as username, password, social security number or credit card number etc. Attackers fool the Internet users by masking webpage as a trustworthy or legitimate page to retrieve personal information. There are many anti-phishing solutions such as blacklist or whitelist, heuristic and visual similarity-based methods proposed to date, but online users are still getting trapped into revealing sensitive information in phishing websites. A novel classification model is proposed based on features that are extracted from URL, source code, and third-party services to overcome the disadvantages of existing anti-phishing techniques. Proposed model has been evaluated using five different machine learning algorithms.

ACKNOWLEDGEMENT

I am grateful to our institute **East West Institute of Technology** with its ideals and inspiration for having provided us with the facilities, which has made this project a success

I would like to express my gratitude to **Dr. K Channakeshavalu, Principal, EWIT** for providing us with all the facilities that helped me to carry out the work easily.

I express my sincere thanks to **Dr. Suresh M B, Professor and Head, Dept. of ISE, EWIT** for his valuable guidance and support.

I would like to express my sincere thanks to my internship guide **Mrs. Shruthi T. V, Asst. Professor, Dept. of ISE, EWIT** for her valuable guidance, encouragement in carrying out the internship work.

I would like to express my sincere gratitude to my supervisor **Mr. Harish S** for providing his invaluable guidance, comments and suggestions throughout the course of the internship. During the period of my internship work. I have received generous help from many quarters, Without the help of them, it was impossible to finish my work.

Finally, I express sincere thanks to my parents. well-wishers and friends for their moral support, encouragement that help me in completing the internship work.

**AKSHAY ARAVIND
(1EW18IS006)**

TABLE OF CONTENTS

	Page. No
ABSTRACT	i
ACKNOWLEDGEMENT	ii
LIST OF FIGURES	vi
 CHAPTERS	
CHAPTER 1: INTRODUCTION	1-6
1.1 DETECTION OF PHISHING WEBSITES	1-2
1.2 PROBLEM STATEMENT	3-6
1.3 PROPOSED SYSTEM	6-7
CHAPTER 2: COMPANY PROFILE	8-10
2.1 INTRODUCTION	8-9
2.2 SERVICES	9
2.3 CLIENTELE	9
2.4 SOFTWARE APPLICATIONS	10
CHAPTER 3: SYSTEM DESIGN	11-16
3.1 MACHINE LEARNING	11-12
3.2 SOFTWARE DESCRIPTION	11-12
3.3 JUPYTER NOTEBOOK	13
3.4 NUMPY	13-15
3.5 PANDAS	15

3.6 ANACONDA	15-16
3.7 PYTHON	16-17
3.8 PANDAS	17
3.9 SEQUENCE	18
CHAPTER 4: SYSTEM REQUIREMENTS	19-23
4.1 FUNCTIONAL REQUIREMENTS	19
4.2 NON-FUNCTIONAL REQUIREMENTS	19-22
4.3 HARDWARE REQUIREMENTS	23
4.4 SOFTWARE REQUIREMENTS	23
CHAPTER 5: IMPLEMENTATION	24-25
5.1 OVERVIEW	24-25
CHAPTER 6: TESTING	26-31
6.1 TYPES OF TESTING	26
6.1.1 UNIT TESTING	26
6.1.2 INTEGRATION TESTING	26
6.1.3 VALIDATION TESTING	26-27
6.2 TRAINING AND TESTING PHASE	27-28
6.3 PERFORMANCE EVALUATION	29-31

CHAPTER 7 : RESULTS	32-34
----------------------------	--------------

7.1 ALGORITHM CODES	32-34
---------------------	-------

CONCLUSION

REFERENCES

LIST OF FIGURES

Figure No	Title	Page No
1.1	STEPS FOR PROPOSED MODEL	3
1.2	PHISHING ATTACK OVERVIEW	4
3.6	ANACONDA NAVIAGATOR HOMEPAGE	16
3.9	SEQUENCE DIAGRAM	18
5.1	IMPLEMENTATION DIAGRAM	24
6.1	TESTING PROCESS	27
6.2	TRAINING AND TESTING PHASE	28
6.3	LOGISTIC REGRESSION PERFORMANCE EVALUATION	29
6.4	DECISION TREE CLASSIFIER PERFORMANCE EVALUATION	29
6.5	RANDOM FOREST CLASSIFER PERFORMANCE EVALUATION	30
6.6	K-NEAREST NEIGHBORS PERFORMANCE EVALUATION	30
6.7	XG BOOST CLASSIFER PERFORMANCE EVALUATION	31
6.8	STORE THE MODEL PERFORMANCE RESULTS	31
7.1	LOGISTIC REGRESSION ALGORITHM CODE	32
7.2	DECESION TREE CLASIFIER ALGORITHM CODE	32
7.3	RANDOM FOREST CLASSIFER ALOGITHM CODE	33
7.4	K-NEAREST NEIGHTBORS ALGORITHM CODE	33
7.5	XGBOOST CLASSIFER ALGORITHM CODE	34
7.6	COMPARISON OF MODELS	34

CHAPTER 1

INTRODUCTION

1.1 Detection of Phishing Websites

Phishing costs Internet users billions of dollars yearly. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting Internet users. Phishers use spoofed e-mail, phishing software to steal personal information and financial account details such as usernames and passwords. The paper deals with methods for detecting phishing Web sites by analyzing various features of benign and phishing URLs by Machine learning techniques. Here, the discussion of methods used for detection of phishing Web sites based on lexical features, host properties and page importance properties. Consider various machine learning algorithms for evaluation of the features in order to get a better understanding of the structure of URLs that spread phishing. The fine-tuned parameters are useful in selecting the apt machine learning algorithm for separating the phishing sites from benign sites.

The criminals, who want to obtain sensitive data, first create unauthorized replicas of a real website and e-mail, usually from a financial institution or another company that deals with financial information. The e-mail will be created using logos and slogans of a legitimate company. The nature of website creation is one of the reasons that the Internet has grown so rapidly as a communication medium, it also permits the abuse of trademarks, trade names, and other corporate identifiers upon which consumers have come to rely as mechanisms for authentication. Phisher then send the "spoofed" e-mails to as many people as possible in an attempt to lure them in to the scheme. When these e-mails are opened or when a link in the mail is clicked, the consumers are redirected to a spoofed website, appearing to be from the legitimate entity.

Security attacks on legitimate websites to steal users' information, known as phishing attacks, have been increasing. This kind of attack does not just affect individuals' or organizations' websites. Although several detection methods for phishing websites have been proposed using machine learning, deep learning, and other approaches, their detection accuracy still needs to be enhanced.

Advantages

- This system can be used by many E-commerce or other websites in order to have good customer relationship. User can make online payment securely.
- Data mining algorithm used in this system provides better performance as compared to other traditional classifications algorithms.
- With the help of this system user can also purchase products online without any hesitation.
- User can make online payments without any hesitation.

Disadvantages

- If Internet connection fails, this system won't work.
- All websites related data will be stored in one place.

Problem Definition

Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data. This type of accessing is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user clicks on the link, he will see the website and think it's original and try to provide his credentials.

To overcome this problem, the proposed system is using some of the machine learning algorithms in which it will help us to identify the phishing websites based on the features present in the algorithm. By using these algorithms, we can be able to keep the user personal credentials or the sensitive data safe from the intruders.

Project Purpose

The main purpose of the project is to detect the fake or phishing websites who are trying to get access to the sensitive data or by creating the fake websites and trying to get access of the user personal credentials. The proposed system is using machine learning algorithms to safeguard the sensitive data and to detect the phishing websites who are trying to gain access on sensitive data.

The steps involved in achieving phishing detection are as follows:

The study uses a dataset which contains approximately 10,000 data containing the data of websites store in a .csv file.

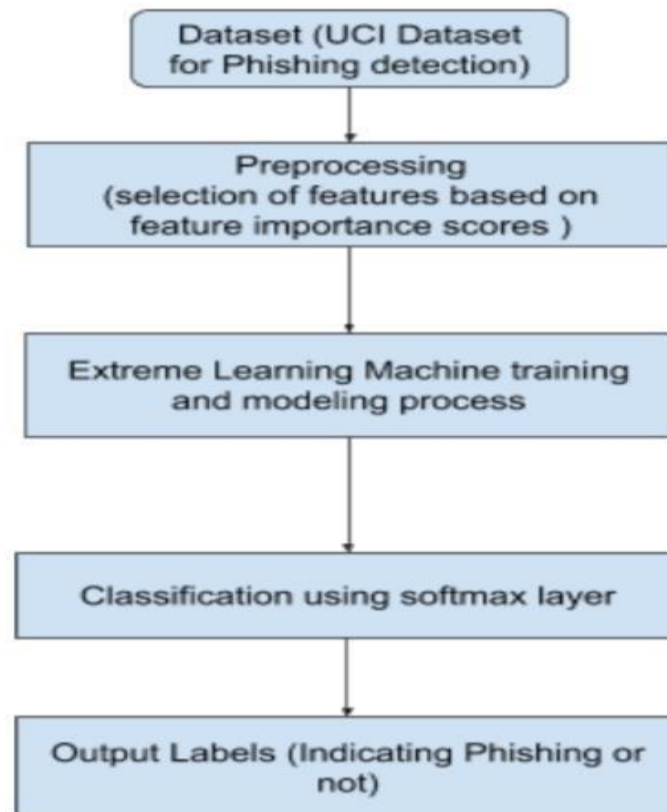


FIGURE 1.1: STEPS FOR PROPOSED MODEL

Features extracted based on the features of websites in UC Irvine Machine Learning Repository database. For classification, a neural network named Extreme Learning Machine (ELM) will be used. Extreme Learning Machine (ELM) is a feed-forward artificial neural network (ANN) model with a single hidden layer. In ELM Learning Processes, differently from ANN that renews its parameters as gradient-based, input weights are randomly selected while output weights are analytically calculated. The given data set will be divided into three parts as training, validation and test data by three-phase division in K-Fold method, and model selection and performance status will be simultaneously performed. This way the performance of the model will be measured in a reliable manner.

1.2 PROBLEM STATEMENT

In website phishing, attacker builds a website which looks like a replica of legitimate site and draws the online user to the website either through advertisements in other websites or social networks such as Facebook and Twitter etc. Some of the attackers are able to manage phishing websites along with security

indicators such as green padlock, HTTPS connection etc. Hence, HTTPS connection is no longer guaranteed to decide legitimacy of a website. This problem to be effectively handled through implementing an efficient detection system.

OBJECTIVE OF STUDY

Phishing attacks are one of the most common and least defended security threats today. Objective of study to identify phishing attacks using five machine learning algorithms. The proposed system handles feature selection through learning algorithm, after feature selection, training and prediction is done. The objective of our study to find an efficient algorithm, which achieves highest accuracy.

OVERVIEW

Phishing attacks take place through various forms such as email, websites and malware. To perform email phishing, attackers design fake emails which claim to be arriving from a trusted company. They send fake emails to millions of online users assuming that at least thousands of legitimate users would fall for it. Phishing is the process whereby someone attempts to obtain your confidential information, such as your passwords, your credit card number, your bank account details or other information protected by the Data Protection Act. Such attempts, are often referred to as Phishing attacks.



Figure 1.2: PHISHING ATTACK OVERVIEW

In website phishing, attacker builds a website which looks like a replica of legitimate site and draws the online user to the website either through advertisements in other websites or social networks

such as Facebook and Twitter etc. Some of the attackers are able to manage phishing websites along with security indicators such as green padlock, HTTPS connection etc.

In Malware phishing, attacker inserts a malicious software such as Trojan horse into a compromised legitimate site without the knowledge of a victim. According to APWG report, 20 million new malware samples were captured in the first quarter of 2016. The vast majority of the late malwares are multifunctional, i.e., they steal the information, make the victims system as a part of botnet or download and install different malicious software without client's notification.

Spear Phishing target a specific group of people or community belonging to an organization or a company. They send emails which pretend to be sent by a colleague, manager or a higher official of the company requesting sensitive data related to the company. The main intention of general phishing is financial fraud, whereas spear phishing is a collection of sensitive information.

Whaling is a type of spear phishing where attackers target bigger fish like executive officers or high-profile targets of private business, government agencies or other organizations. There are many anti-phishing techniques proposed in the literature to detect and prevent phishing attacks. We have categorized these anti-phishing techniques into 4 categories.

List-based techniques where most of the modern browsers such as Chrome, Firefox and Explorer etc. follow list-based techniques to block phishing sites. There are two types of list-based techniques such as whitelist and blacklist. The whitelist contains a list of legitimate URLs which can be accessed by the browsers. The browser downloads the website, only if the URL is present in the whitelist. Due to this behavior, even the legitimate websites which are not whitelisted are also blocked resulting in high false positives. The blacklist contains phishing or malicious URLs which are blocked by the browsers in downloading the webpages. Due to this behavior, the phishing sites which are not blacklisted are also downloaded by the browser resulting in high false negatives. These non-blacklisted phishing sites are also called as Zero-day phishing sites. A small change in the URL is sufficient to bypass the list-based techniques. Frequent update of these lists is mandatory to counter the new phishing sites.

EXISTING SYSTEM

Heuristic-based techniques

These techniques use features extracted from the phishing website to detect phishing attack. Some of the phishing sites do not have common features resulting in poor detection rate using this mechanism. As this approach does not use list-based comparison, it results in less false positives and less false negatives. This technique detects zero-day phishing attacks which the list-based techniques fail to detect.

Disadvantages

- It has less accuracy compared to list-based techniques as there is no guarantee of existence of these features in all phishing websites.
- An attacker can bypass the heuristic features once he knows the algorithm or features used in detecting phishing sites thereby reaches his goal of stealing sensitive information.

Visual Similarity-based Approach

The main objective of the phisher is to deceive the user by designing an exact image of legitimate site such that the user does not get any suspicion on the phishing site. Hence, the anti-phishing techniques compare suspicious website image with legitimate image database to get the similarity ratio, used for the classification of suspicious websites. The website is classified as phishing when the similarity score is greater than a certain threshold else it is treated as legitimate.

Disadvantages

- Image comparison of suspicious website with entire legitimate database store takes more time complexity.
- More space to store legitimate image database.
- Web page with animated website compared with phishing website leads to the low percentage of similarity that leads to high false negative rate. This technique fails, when the background of web page is slightly changed without deviating from visual appearance of legitimate site.

Machine learning-based techniques

Nowadays, most of the researchers are concentrating on the use of machine learning algorithms (ML) applied on the features extracted from the websites to detect phishing attacks. These techniques are a combination of heuristic methods and machine learning algorithms, i.e., dataset used by the machine learning algorithms is extracted through heuristic methods. Some of the machine learning algorithms are sequential minimum optimization (SMO), J48 tree, Random Forest (RF), logistic regression (LR), multilayer perceptron (MLP), Bayesian network (BN), support vector machine (SVM) and AdaBoostM1 etc. As ML-based techniques are based on heuristic features, they are able to identify the zero-day phishing attacks which make them advantageous than list-based techniques. These techniques work efficiently on the large sets of data.

1.3 PROPOSED SYSTEM

- Add new heuristic features with machine learning algorithms to reduce the false positives in detecting new phishing sites.
- Made an attempt to identify the best machine learning algorithm to detect phishing sites with high accuracy than the existing techniques.
- Used Five Machine Learning Algorithms
 - a. **Logistic regression (LR)**
 - b. **K-Nearest Neighbors (KNN)**
 - c. **Random Forest Classifiers (RFC),**
 - d. **XG Boost**
 - e. **Decision Tree Classifiers**
- Based on the experimental observations, **XG Boost** outperformed the others.
- The choice of considering these machine learning algorithms is based on the classifiers used in the recent literature.

CHAPTER 2

Company Profile

2.1 MINDSET IT SOLUTIONS A global solutions company providing custom solutions to high technology companies worldwide. Combining proven expertise in technology, vast knowledge of hardware product design cycle, system design cycle (Board design / development), Embedded software services and an understanding of emerging business domains. range of services that includes. At Mindset Technologies, we go beyond providing software solutions. We work with our client's technologies and business changes that shape their competitive advantages.

Address Bar based Features

At Mindset Technologies, we go beyond providing software solutions. We work with our client's technologies and business changes that shape their competitive advantages. Founded in 2000, Mindset Technologies (P) Ltd. is a software and service provider that helps organizations deploy, manage, and support their business-critical software more effectively. Utilizing a combination of proprietary software, services and specialized expertise, Mindset Technologies (P) Ltd. helps mid-to-large enterprises, software companies and IT service providers improve consistency, speed, and transparency with service delivery at lower costs. Mindset Technologies (P) Ltd. helps companies avoid many of the delays, costs and risks associated with the distribution and support of software on desktops, servers and remote devices. Our automated solutions include rapid, touch-free deployments, ongoing software upgrades, fixes and security patches, technology asset inventory and tracking, software license optimization, application self-healing and policy management. At Mindset Technologies, we go beyond providing software solutions. We work with our clients' technologies and business processes that shape their competitive advantages.

2.2 Services

As a team we have the prowess to have a clear vision and realize it too. As a statistical evaluation, the team has more than 40,000 hours of expertise in providing real-time solutions in the fields of Embedded Systems, Control systems, Micro-Controllers, c Based Interfacing, Programmable Logic Controller, VLSI Design and Implementation, Networking With C, ++, java, client Server Technologies in Java, (J2EE\J2ME\J2SE\EJB), VB & VC++, Oracle and operating system concepts with LINUX.

Vision

“Dreaming a vision is possible and realizing it is our goal”.

Mission

We have achieved this by creating and perfecting processes that are in par with the global standards and deliver high quality, high value services, reliable and cost-effective IT products to clients around the world.

2.3 Clientele

- *Aray InfoTech*
- *Inquirre consultancy (U.S.A)*
- *K square consultancy pvt Ltd (U.S.A)*
- *Opal solutions*
- *Texlab Solutions*
- *Vertex Business Machines*
- *JM InfoTech*

MINDSET IT SOLUTIONS was founded by a group of tech savvy professionals with a multifaceted hardware and software background, with a vision to offer the Silicon world refreshing and cost-effective Silicon, System Design and Embedded software services. Reduced costs, quicker time-to-market, huge value-adds and enhanced productivity are our way of life. The very cornerstone of our success has been our unerring path to ensuring that QA processes and procedures are met with unwavering dedication. We follow Hardware Methodologies and Software Processes that are a combination of policies and processes. Processes that have been derived from best practices from within the software and hardware industry. We follow ISO 9001:2000 processes for all the activities we execute, and aim to achieve SEI CMM Level 5 in the due course. These processes are continuously refined and defined for ongoing measurement and improvement for both process and product quality.

Mindset IT solutions provides complete solutions in embedded systems and system level programming. Our team has a breadth of experience in design and development for embedded systems that spans many CPU architectures, chipsets and peripherals across a variety of platforms. We generate custom software including device drivers, firmware and board support packages. By leveraging our experience and mature processes.

2.4 Software Applications

Programming Languages & OS	C, C++, Unix, Linux, Windows 9X/NT
Real Time Operating Systems	VxWorks, RtLinux, WinCE, QNX, RTX-51
Communication Protocols & Device Drivers	ISDN, HDLC, T1/E1/J1, DSL, ATM, TCP/IP, PPP, Ethernet.
Processors and Controllers	Intel (8-bit to 32-bit), ARM, Power PC, DSP (TI, Analog Devices)
VLSI Tools	Xilinx, Altera & Cypress
Applications	Industrial automation, Consumer Automobiles, Security Systems, Telecom

Our areas of expertise include Networking and Communication software, PDA software, Digital Signal Processing, Security Applications and Real-Time Embedded Systems. We have created applications to provide functionality such as multi-conference messaging, asset tracking with GPS, remote monitoring and control of equipment and personnel, network test equipment, and VoIP. In addition, we have a vast amount of experience creating software for embedded devices, particularly 802.11 access points, rugged communications equipment and embedded authentication systems.

CHAPTER 3

SYSTEM DESIGN

The purpose or goal behind phishing is data, money or personal information stealing through the fake website. The best strategy for avoiding the contact with the phishing web site is to detect real time malicious URL. Phishing websites can be determined on the basis of their domains. They usually are related to URL which needs to be registered (low-level domain and upper-level domain, path, query). Recently acquired status of intra-URL relationship is used to evaluate it using distinctive properties extracted from words that compose a URL based on query data from various search engines such as Google and Yahoo. These properties are further led to the machine-learning based classification for the identification of phishing URLs from a real dataset. This paper focus on real time URL phishing against phishing content by using phish-STORM. For this a few relationships between the register domain rest of the URL are consider also intra URL relentless is consider which help to dusting wish between phishing or non-phishing URL. For detecting a phishing website certain typical blacklisted urls are used, but this technique is unproductive as the duration of phishing websites is very short. Phishing is the name of avenue. It can be defined as the manner of deception of an organization's customer to communicate with their confidential information in an unacceptable behavior. It can also be defined as intentionally using harsh weapons such as Spasm to automatically target the victims and targeting their private information. As many of the failures being occurred in the SMTP are exploiting vectors for the phishing websites, there is a greater availability of communication for malicious message deliveries.

Proposed a novel classification approach that use heuristic-based feature extraction approach.

In this, they have classified extracted features into different categories such as URL Obfuscation features, Hyperlink-based features.

Moreover, proposed technique gives 92.5% accuracy. Also, this model is purely depending on the quality and quantity of the training set and Broken links feature extraction.

3.1 MACHINE LEARNING

Writing review is the most critical advance in programming improvement process. Before building up the instrument it is important to decide the time factor, economy and friends quality. When these things are fulfilled, at that point following stages is to figure out which working

framework and dialect can be utilized for building up the instrument. When the developers begin fabricating the instrument the software engineers require part of outside help. This help can be gotten from senior software engineers, from book or from sites. Before building the framework, the above thought is considered for building up the proposed framework.

AI (ML) is a class of calculation that enables programming applications to turn out to be progressively precise in anticipating results without being expressly customized. The fundamental reason of AI is to assemble calculations that can get input information and utilize factual examination to foresee a yield while refreshing yields as new information winds up accessible. The procedures engaged with AI are like that of information mining and prescient displaying. Both require scanning through information to search for examples and modifying program activities as needs be. Numerous individuals know about AI from shopping on the web and being served advertisements identified with their buy. This happens on the grounds that suggestion motors use AI to customize online promotion conveyance in practically continuous. Past customized advertising, other regular AI use cases incorporate misrepresentation location, spam separating, arrange security risk identification, prescient support and building news sources.

Benefits of Machine learning:

- Simplifies Product Marketing and Assists in Accurate Sales Forecasts.
- Utilization and efficiency improvement
- Very high Scalability
- High Computing power

3.2 SOFTWARE DESCRIPTION

Selection of programming language - Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form

without charge for all major platforms and can be freely distributed.

Programmers prefer python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy. A bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. On the other hand, often the quickest way to debug a program is to add a few print statements to the source. The fast edit-test debug cycle makes this simple approach very effective.

3.3 JUPYTER NOTEBOOK

The Jupyter Notebook App is a server-customer application that permits altering and running note pad records by means of an internet browser. The Jupyter Notebook App can be executed on a nearby work area requiring no web access (as portrayed in this report) or can be introduced on a remote server and got to through the web. Notwithstanding showing/altering/running note pad archives, the Jupyter Notebook App has a "Dashboard" (Notebook Dashboard), a "control board" indicating nearby records and permitting to open note pad reports or closing down their portions. A scratch pad part is a "computational motor" that executes the code contained in a Notebook record. The IPython part, referenced in this guide, executes python code. Portions for some, different dialects exist (official parts).

When you open a Notebook report, the related part is consequently propelled. At the point when the scratch pad is executed (either cell-by-cell or with menu Cell - > Run All), the portion plays out the calculation and produces the outcomes. Contingent upon the sort of calculations, the piece may expend critical CPU and RAM. Note that the RAM isn't discharged until the part is closed down, the Notebook Dashboard is the part which is indicated first when you dispatch Jupyter Notebook App. The Notebook Dashboard is essentially used to open note pad archives, and to deal with the running portions (picture and shutdown).

The Notebook Dashboard has different highlights like a record director, in particular exploring organizers and renaming/erasing documents.

3.4 NUMPY

NumPy is, much the same as SciPy, Scikit-Learn, Pandas, and so forth one of the

bundles that you can't miss when you're learning information science, principally in light of the fact that this library gives you a cluster information structure that holds a few advantages over Python records, for example, being increasingly reduced, quicker access in perusing and composing things, being progressively advantageous and increasingly productive.

NumPy exhibits are somewhat similar to Python records, yet at the same time particularly unique in the meantime. For those of you who are new to the subject, how about we clear up what it precisely is and what it's useful for. As the name gives away, a NumPy cluster is a focal information structure of the NumPy library. The library's name is another way to say "Numeric Python" or "Numerical Python".

At the end of the day, NumPy is a Python library that is the center library for logical registering in Python. It contains an accumulation of apparatuses and strategies that can be utilized to settle on a PC numerical model of issues in Science and Engineering. One of these apparatuses is an elite multidimensional cluster object that is an incredible information structure for effective calculation of exhibits and lattices. To work with these clusters, there's a tremendous measure of abnormal state scientific capacities work on these grids and exhibits. Since you have set up your condition, it's the ideal opportunity for the genuine work. In fact, you have officially gone for some stuff with exhibits in the above Data Camp Light pieces. Be that as it may, you haven't generally gotten any genuine hands-on training with them, since you originally expected to introduce NumPy all alone pc. Since you have done this current, it's a great opportunity to perceive what you have to do so as to run the above code pieces without anyone else.

A few activities have been incorporated underneath with the goal that you would already beable to rehearse how it's done before you begin your own. To make a NumPy exhibit, you can simply utilize the `np.array()` work. You should simply pass a rundown to it, and alternatively, you can likewise indicate the information sort of the information. In the event that you need to find out about the conceivable information types that you can pick, go here or consider investigating DataCamp's NumPy cheat sheet. There's no compelling reason to proceed to retain these NumPy information types in case you're another client; But you do need to know and mind what information you're managing. The information types are there when you need more power over how your information is put away in memory and on plate. Particularly in situations where you're working with broad information, it's great that you know to control the capacity type.

Remember that, so as to work with the `np.array()` work, you have to ensure that the NumPy

library is available in your condition. The NumPy library pursues an import tradition: when you import this library; you need to ensure that you import it as np. By doing this, you'll ensure that different Pythonistas comprehend your code all the more effectively.

3.5 PANDAS

Pandas is an open-source, BSD-authorized Python library giving elite, simple to-utilize information structures and information examination instruments for the Python programming language. Python with Pandas is utilized in a wide scope of fields including scholastic and business areas including money, financial matters, Statistics, examination, and so on. This instructional exercise has been set up for the individuals who try to become familiar with the essentials and different elements of Pandas. It will be explicitly valuable for individuals working with information purging and examination. In the wake of finishing this instructional exercise, you will wind up at a moderate dimension of ability from where you can take yourself to more elevated amounts of skill. You ought to have a fundamental comprehension of Computer Programming phrasings. A fundamental comprehension of any of the programming dialects is an or more. Pandas library utilizes the vast majority of the functionalities of NumPy. It is recommended that you experience our instructional exercise on NumPy before continuing with this instructional exercise.

3.6 ANACONDA

Anaconda constrictor is bundle director. Jupyter is an introduction layer. Boa constrictor endeavors to explain the reliance damnation in python—where distinctive tasks have diverse reliance variants—in order to not influence distinctive venture conditions to require diverse adaptations, which may meddle with one another. Jupyter endeavors to fathom the issue of reproducibility in investigation by empowering an iterative and hands-on way to deal with clarifying and imagining code; by utilizing rich content documentations joined with visual portrayals, in a solitary arrangement. Boa constrictor is like pyenv, venv and minconda; it's intended to accomplish a python situation that is 100% reproducible on another condition, autonomous of whatever different forms of a task's conditions are accessible. It's somewhat like Docker, however limited to the Python biological system. Jupyter is an astounding introduction device for expository work; where you can display code in "squares," joins with rich content depictions among squares, and the consideration of organized yield from the squares, and charts created in an all-around planned issue by method for another square's code. Jupyter is extraordinarily great in expository work to guarantee reproducibility in somebody's

exploration, so anybody can return numerous months after the fact and outwardly comprehend what somebody attempted to clarify, and see precisely which code drove which representation and end.

Regularly in diagnostic work you will finish up with huge amounts of half-completed note pads clarifying Proof-of-Concept thoughts, of which most won't lead anyplace at first. A portion of these introductions may months after the fact—or even years after the fact— present an establishment to work from for another issue.

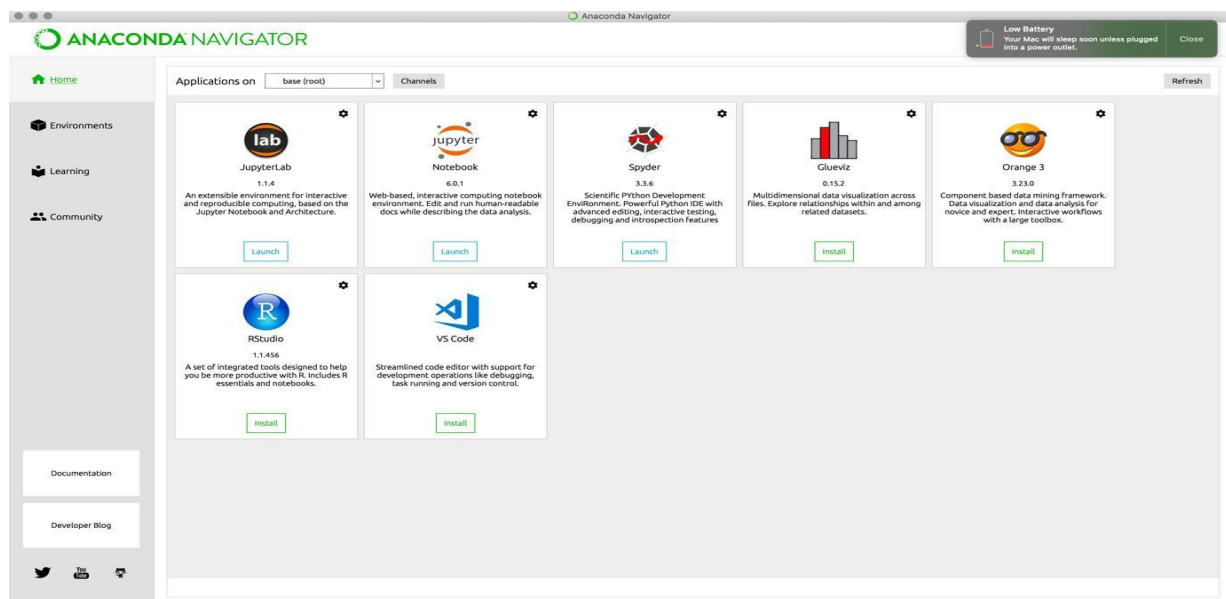


FIGURE 3.6: ANACONDA NAVIGATOR HOMEPAGE

3.7 PYTHON

Python is a translated, object-oriented, abnormal state programming language with dynamic semantics. Its abnormal state worked in information structures, joined with dynamic composing and dynamic authoritative, make it appealing for Rapid Application Development, just as for use as a scripting or paste language to interface existing segments together. Python's basic, simple to learn language structure underlines intelligibility and hence decreases the expense of program support. Python underpins modules and bundles, which empowers program seclusion and code reuse. The Python translator and the broad standard library are accessible in source or parallel structure without charge for every single significant stage, and can be openly appropriated.

Frequently, software engineers begin to look all starry eyed at Python on account of the expanded efficiency it gives. Since there is no aggregation step, the alter test-troubleshoot cycle is staggeringly quick. Troubleshooting Python programs is simple: a bug or awful information

will never cause a division blame. Rather, when the mediator finds a blunder, it raises a special case. At the point when the program doesn't get the special case, the translator prints a stack follow. A source level debugger permits assessment of nearby and worldwide factors, assessment of discretionary articulations, setting breakpoints, venturing through the code a line at any given moment, etc. The debugger is written in Python itself, vouching for Python's contemplative power. Then again, frequently the speediest method to troubleshoot a program is to add a couple of print proclamations to the source: the quick alter test-investigate cycle makes this straightforward methodology successful.

Python is an item situated, abnormal state programming language with incorporated unique semantics essentially for web and application improvement. It is amazingly alluring in the field of Rapid Application Development since it offers dynamic composing and dynamic restricting alternatives.

Moreover, Python underpins the utilization of modules and bundles, which implies that projects can be planned in a secluded style and code can be reused over an assortment of tasks. When you've built up a module or bundle you need, it very well may be scaled for use in different tasks, and it's anything but difficult to import or fare these modules.

A standout amongst the most encouraging advantages of Python is that both the standard library and the mediator are accessible for nothing out of pocket, in both parallel and sourcestructure. There is no restrictiveness either, as Python and all the important instruments are accessible on every single real stage. In this way, it is a tempting alternative for designers who would prefer not to stress over paying high improvement costs.

3.8 TensorFlow:

TensorFlow is Google Brain's second-age framework. Form 1.0.0 was discharged on February 11, 2017. TensorFlow is an open-source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework. TensorFlow is accessible on 64-bit Linux, macOS, Windows, and portable processing stages including Android and iOS. Its adaptable design considers the simple sending of calculation over an assortment of stages (CPUs, GPUs), and from work areas to bunches of servers to portable and edge gadgets. TensorFlow calculations are communicated as dataflow diagrams.

3.9 Sequence

The user must request on a website for the system. Once the system has requested the webpage, which is located in the server, user would get the desired particular details. The user can return the page for more results. This is the process in which a user would access a webpage.

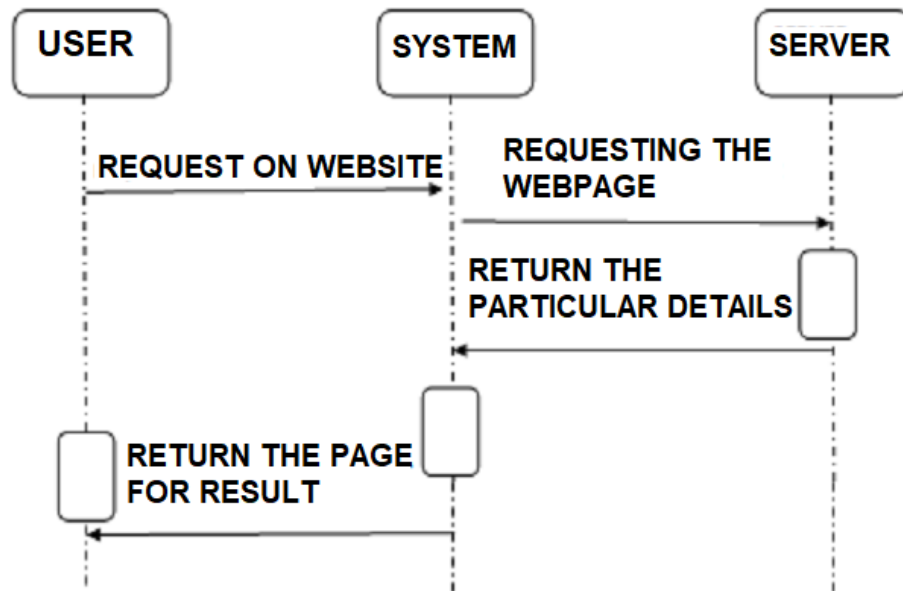


FIGURE 3.9: SEQUENCE DIAGRAM

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

A function of software system is defined in functional requirement and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

- The system should be able to load air quality data and preprocess data.
- It should be able to analyze the air quality data.
- It should be able to group data based on hidden patterns.
- It should be able to assign a label based on its data groups.
- It should be able to split data into trainset and test set.
- It should be able to train model using trainset.
- It must validate trained model using test set.
- It should be able to display the trained model accuracy.
- It should be able to accurately predict the air quality on unseen data.

4.2 NON-FUNCTIONAL REQUIREMENTS

Nonfunctional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's *quality attributes*. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the customer are described by the specification. Some Non-Functional Requirements are as follows:

4.2.1 ACCESSIBILITY

Availability is a general term used to depict how much an item, gadget, administration, or condition is open by however many individuals as would be prudent.

In our venture individuals who have enrolled with the cloud can get to the cloud to store and recover their information with the assistance of a mystery key sent to their email ids.

UI is straightforward and productive and simple to utilize.

4.2.1 MAINTAINABILITY

In programming designing, viability is the simplicity with which a product item can be altered so as to:

- Correct absconds
- Meet new necessities

New functionalities can be included in the task based the client necessities just by adding the proper documents to existing venture utilizing ASP.net and C# programming dialects. Since the writing computer programs is extremely straightforward, it is simpler to discover and address the imperfections and to roll out the improvements in the undertaking.

4.2.2 SCALABILITY

Framework is fit for taking care of increment all out throughput under an expanded burden when assets (commonly equipment) are included. Framework can work ordinarily under circumstances, for example, low data transfer capacity and substantial number of clients.

4.2.3 PORTABILITY

Convey ability is one of the key ideas of abnormal state programming. Convenient is the product code base component to have the capacity to reuse the current code as opposed to making new code while moving programming from a domain to another. Venture can be executed under various activity conditions gave it meet its base setups. Just framework records and dependent congregations would need to be designed in such case.

The functional requirements for a system describe what the system should do. Those requirements depend on the type of software being developed, the expected users of the software. These are the statement of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation.

- Extracting data from CSV files
- Cleaning the data.
- Vector Representation.

Non-functional requirements are not about functionality or behavior of system, but rather are used to specify the capacity of a system. They are more related to properties of system such as quality, reliability and quick response time. Non- functional requirements come up via customer needs, because of budget, interoperability need such as software and hardware requirement, organizational policies or due to some external factors such as: -

- Basic Operational Requirement
- Organizational Requirement
- Product Requirement
- User Requirement

The four primary functions of systems engineering are all performed by the end users, which is the customers. Operational requirements which are given by:-

- **Mission profile or scenario:** It is a map which describes the procedures and leads us to the final goal/ objective. The goal of proposed system is, to predict the crop yield prediction for future year using previous year dataset.
- **Performance:** It basically gives system parameters to reach our goal. Parameters for the proposed system are accurate predicted value which is compared to the existing system.
- **Utilization environments:** It enlists the different permutations and combinations a system can be reused in many other applications which gives better prediction, as well as gives a new approach to prediction techniques.
- **Life cycle:** It discuss about the life span of a system. As number of data increases the number of iterations increases, which will give more accuracy to the output.

4.2.5 Organizational Requirement

The Organizational requirement consists of the following types:

- **Process Standards:** To make sure the system is a quality product, IEEE standards have been used during system development.
- **Design Methods:** Design is an important step, on which all other steps in the engineering process are based on.
- It takes the project from a theoretical idea to an actual product. It gives us the basis of our solution. Because all the steps after designing are based on the design itself, this step affects the quality of the product and is a major player in how the testing and maintenance of a project take place and how successful they are. Following the design to the 'T' is of utmost importance.

Product Requirement

- **Portability:** As the system is Python based, it will run on a platform which is supported by ANACONDA.
- **Correctness:** The system has been put through rigorous testing after it has followed strict guidelines and rules. The testing has validated the data.
- **Ease of Use:** The user interface allows the user to interact with the system at a very comfortable level with no hassles.
- **Modularity:** The many different modules in the system are neatly defined for ease of use and to make the product as flexible as possible with different permutations and combinations.
- **Robustness:** During the development of the system special care is being taken to make sure that the end results are optimized to the highest level and the results are relevant and validated. Python language is used for the development, itself provides robustness to the system and thus makes it highly unlikely to fail.

'System quality' and 'Non-functional requirements' are interchangeable terms. These qualities mainly consist of two things i.e., evolution and execution. Evolution includes scalability, maintainability and testability whereas, execution include usability and privacy of system.

The kernel cell is for working with the kernel that is running in the background. Here we can restart the kernel, reconnect to it, shut it down, or even change with kernel your notebook is using.

4.3 Hardware Requirements

The following is the hardware requirements of the system for the proposed system:

4.3.1	Processor	: Any Processor above 500 MHz
4.3.2	RAM	: 8 GB
4.3.3	Hard Disk	: 1 TB
4.3.4	Input device	: Standard Keyboard and Mouse

4.4 Software Requirements

The following is the software requirements of the system for the proposed system:

4.4.1	OS	: Windows 10
4.4.2	Platform	: Jupyter Notebook
4.4.3	Language	: Python
4.4.4	IDE/Tool	: Anaconda 3-5.0.3

CHAPTER 5

IMPLEMENTATION

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

5.1 Overview

The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort and any site-specific implementation requirements.

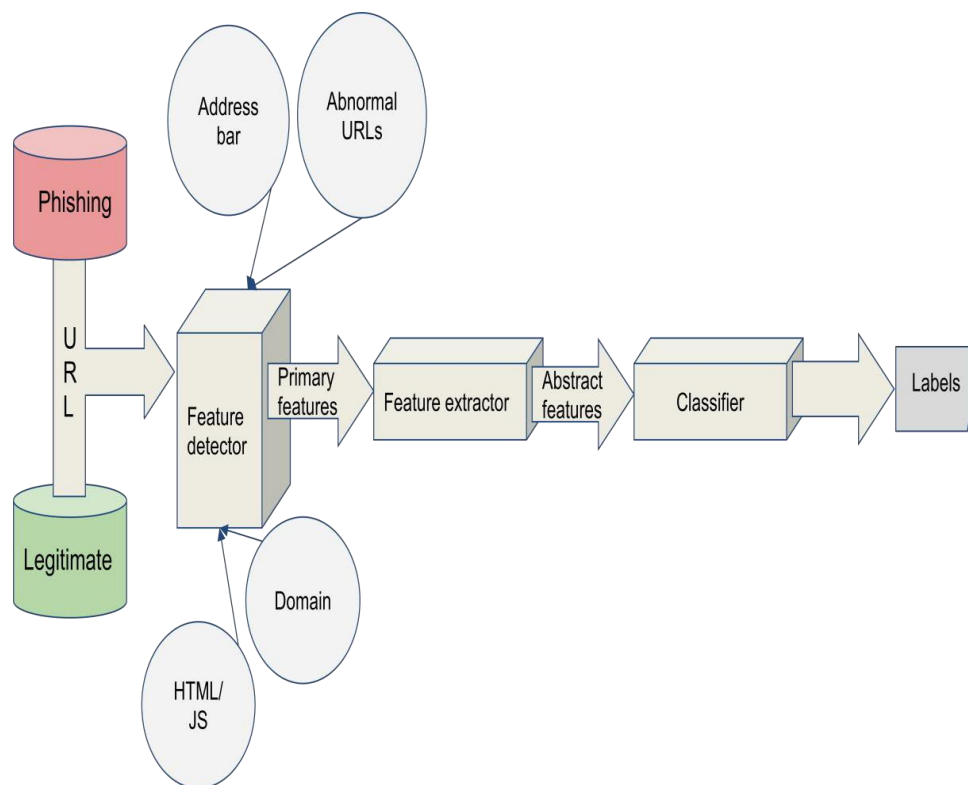


FIGURE 5.1: IMPLEMENTATION DIAGRAM

Implementation support

Anaconda is a free and open-source distribution of the Python and R programming languages for data science and learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify management and deployment.

Anaconda3 includes Python 3.6. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage anaconda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux. The following are the system requirements:

- License: Free use and redistribution under the terms of the Anaconda End User License Agreement.
- Operating system: Windows Vista or newer, 64-bit macOS 10.10+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others. Windows XP supported on Anaconda versions 2.2 and earlier. See lists. Download it from our archive.
- System architecture: 64-bit x86, 32-bit x86 with Windows or Linux, Power8 or Power9.

CHAPTER 6

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

The testing process overview is as follows:

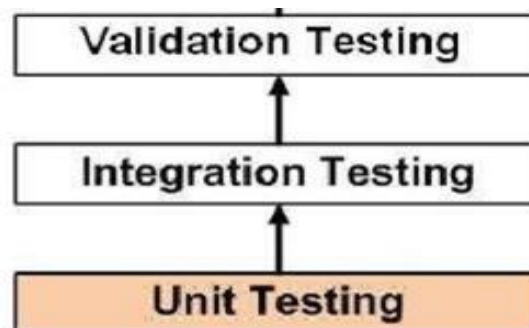


Figure 6.1: TESTING PROCESS

6.2 TRAINING AND TESTING PHASE

In the training phase, random forests, Logistic Regression, Random Forest Classifier, XGBoost, Decision Tree Classifiers, and KNN are trained without applying an optimization method. These classifiers are then optimized using the genetic algorithm, which selects the optimal values of parameters for several ensemble models. The optimized classifiers are then ranked and used as base classifiers for the stacking ensemble method. Finally, new websites are collected and used as a testing dataset in order to predict the final class label of these websites. The reason behind this is twofold: on

the one hand, to obtain a general insight into the performance of ensemble classifiers before optimizing them, and on the other hand, to explore which of the phishing websites' characteristics is most useful. The aforementioned classifiers were then optimized using the genetic algorithm. Here, the genetic algorithm was used for selecting the optimal values of model parameters in order to improve the overall accuracy of the proposed model. Later, in the ranking phase, the optimized classifiers were ranked and used as a base classifier for the ensemble classifier—the stacking method. In the testing phase, new websites were collected and used as testing data

Figure 6.2.2 refers to this phase as the detection phase, as these steps will be applied to any website in future in order to detect whether it is a benign or malicious website. In order to extract the features of the websites, we followed the methodology presented in. A set of benign and malicious websites was collected from the malware and phishing blacklist of the dataset of verified phishing pages. In order to extract the same features as those used in the training dataset.

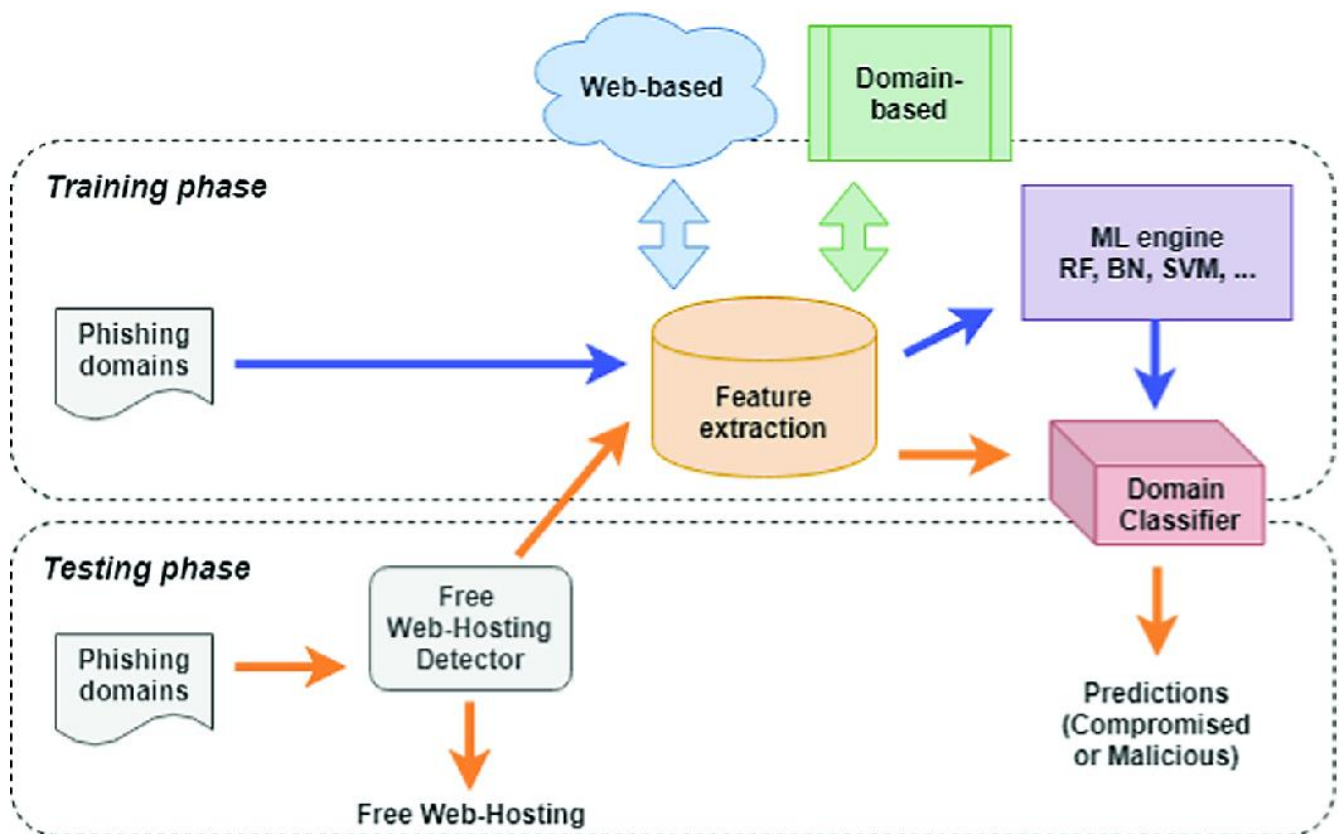


Figure 6.2: TRAINING AND TESTING PHASE

6.3 PERFORMANCE EVALUATION

```
In [14]: #computing the accuracy of the model performance
acc_train_model = accuracy_score(y_train,y_train_model)
acc_test_model = accuracy_score(y_test,y_test_model)

print("LogisticRegression: Accuracy on the Model: ",model_score)
print("LogisticRegression: Accuracy on training Data: {:.3f}".format(acc_train_model))
print("LogisticRegression: Accuracy on test Data: {:.3f}".format(acc_test_model))
print(metrics.classification_report(y_test, y_predict))
print(metrics.confusion_matrix(y_test, y_predict))

LogisticRegression: Accuracy on the Model: 0.807
LogisticRegression: Accuracy on training Data: 0.799
LogisticRegression: Accuracy on test Data: 0.807
      precision    recall  f1-score   support

      0       0.75      0.93      0.83       1012
      1       0.91      0.68      0.78        988

   accuracy          0.81       2000
  macro avg       0.83      0.81      0.80       2000
 weighted avg       0.83      0.81      0.80       2000

[[943  69]
 [317 671]]
```

FIGURE 6.3: Logistic Regression Performance Evaluation

```
In [17]: #computing the accuracy of the model performance
acc_train_tree = accuracy_score(y_train,y_train_tree)
acc_test_tree = accuracy_score(y_test,y_test_tree)

print("Decision Tree: Accuracy on the Model: ",tree_score)
print("Decision Tree: Accuracy on training Data: {:.3f}".format(acc_train_tree))
print("Decision Tree: Accuracy on test Data: {:.3f}".format(acc_test_tree))
print(metrics.classification_report(y_test, y_predict))
print(metrics.confusion_matrix(y_test, y_predict))

Decision Tree: Accuracy on the Model: 0.806
Decision Tree: Accuracy on training Data: 0.812
Decision Tree: Accuracy on test Data: 0.817
      precision    recall  f1-score   support

      0       0.74      0.95      0.83       1015
      1       0.93      0.65      0.77        985

   accuracy          0.81       2000
  macro avg       0.84      0.80      0.80       2000
 weighted avg       0.83      0.81      0.80       2000

[[967  48]
 [340 645]]
```

FIGURE 6.4: Decision Tree Classifier Performance Evaluation

```
In [20]: #computing the accuracy of the model performance
acc_train_forest = accuracy_score(y_train,y_train_forest)
acc_test_forest = accuracy_score(y_test,y_test_forest)

print("Random forest: Accuracy on the Model: ",model_score)
print("Random forest: Accuracy on training Data: {:.3f}".format(acc_train_forest))
print("Random forest: Accuracy on test Data: {:.3f}".format(acc_test_forest))
print(metrics.classification_report(y_test, y_predict))
print(metrics.confusion_matrix(y_test, y_predict))

Random forest: Accuracy on the Model: 0.798
Random forest: Accuracy on training Data: 0.814
Random forest: Accuracy on test Data: 0.818
      precision    recall  f1-score   support

      0       0.73       0.93       0.82       987
      1       0.91       0.67       0.77      1013

   accuracy                   0.80       2000
  macro avg       0.82       0.80       0.79       2000
 weighted avg       0.82       0.80       0.79       2000

[[921  66]
 [338 675]]
```

FIGURE 6.5: Random Forest Classifier Performance Evaluation

```
In [23]: #computing the accuracy of the model performance
acc_train_knn = accuracy_score(y_train,y_train_knn)
acc_test_knn = accuracy_score(y_test,y_test_knn)

print("KNeighborsClassifier: Accuracy on the Model: ",model_score)
print("KNeighborsClassifier: Accuracy on training Data: {:.3f}".format(acc_train_knn))
print("KNeighborsClassifier: Accuracy on test Data: {:.3f}".format(acc_test_knn))
print(metrics.classification_report(y_test, y_predict))
print(metrics.confusion_matrix(y_test, y_predict))

KNeighborsClassifier: Accuracy on the Model: 0.7915
KNeighborsClassifier: Accuracy on training Data: 0.810
KNeighborsClassifier: Accuracy on test Data: 0.791
      precision    recall  f1-score   support

      0       0.79       0.79       0.79       987
      1       0.80       0.79       0.79      1013

   accuracy                   0.79       2000
  macro avg       0.79       0.79       0.79       2000
 weighted avg       0.79       0.79       0.79       2000

[[784 203]
 [214 799]]
```

FIGURE 6.6: K-Nearest Neighbors Performance Evaluation


```
In [26]: #computing the accuracy of the model performance
acc_train_xgb = accuracy_score(y_train,y_train_xgb)
acc_test_xgb = accuracy_score(y_test,y_test_xgb)

print("XGBoost: Accuracy on the Model: ",model_score)
print("XGBoost: Accuracy on training Data: {:.3f}".format(acc_train_xgb))
print("XGBoost : Accuracy on test Data: {:.3f}".format(acc_test_xgb))
print(metrics.classification_report(y_test, y_predict))
print(metrics.confusion_matrix(y_test, y_predict))
```

XGBoost: Accuracy on the Model: 0.864
XGBoost: Accuracy on training Data: 0.867
XGBoost : Accuracy on test Data: 0.864

	precision	recall	f1-score	support
0	0.82	0.82	0.82	1008
1	0.82	0.81	0.82	992
accuracy			0.82	2000
macro avg	0.82	0.82	0.82	2000
weighted avg	0.82	0.82	0.82	2000

```
[[830 178]
 [185 807]]
```

FIGURE 6.7: XG Boost Classifier Performance Evaluation

Machine Learning Models

From the dataset above, We have learnt that this is a supervised machine learning. This dataset uses a classification problem, to train the dataset in this notebook are:

- Decision Tree
- LogisticRegression
- RandomForestClassifiers
- XGBoost
- KNeighborsClassifier

```
# Creating holders to store the model performance results
ML_Model = []
acc_train = []
acc_test = []

#function to call for storing the results
def storeResults(model, a,b):
    ML_Model.append(model)
    acc_train.append(round(a, 3))
    acc_test.append(round(b, 3))
```

FIGURE 6.8: MACHINE LEARNING MODELS

CHAPTER 7

RESULTS

7.1 Algorithm Codes

LOGISTIC REGRESSION

```
# instantiate the model
model = LogisticRegression(max_iter=1000)
# fit the model
model.fit(X_train,np.ravel(y_train,order='C'))
#predicting the target value from the model for the samples
y_predict= model.predict(X_test)
y_train_model = model.predict(X_train)
y_test_model = model.predict(X_test)
model_score=model.score(X_test, y_test)
```

FIGURE 7.1: LOGISTIC REGRESSION ALGORITHM CODE

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth = 5)
# fit the model
tree.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_tree = tree.predict(X_test)
y_train_tree = tree.predict(X_train)

tree_score=model.score(X_test, y_test)
```

FIGURE 7.2: DECISION TREE CLASSIFIER ALGORITHM CODE

Random Forest Classifier

```
# instantiate the model
forest = RandomForestClassifier(max_depth=5)
# fit the model
forest.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_forest = forest.predict(X_test)
y_train_forest = forest.predict(X_train)
model_score=model.score(X_test, y_test)
```

FIGURE 7.3: RANDOM FOREST CLASSIFIER

KNeighborsClassifier

```
# instantiate the model
knn = KNeighborsClassifier(n_neighbors =1)
# fit the model
knn.fit(X_train,np.ravel(y_train,order='C'))
#predicting the target value from the model for the samples
y_predict= knn.predict(X_test)

#predicting the target value from the model for the samples
y_test_knn = knn.predict(X_test)
y_train_knn = knn.predict(X_train)
model_score=knn.score(X_test, y_test)
```

FIGURE 7.4: K-NEAREST NEIGHBORS CLASSIFIER ALGORITHM CODE

XGBoost Classifier

```
# instantiate the model
xgb = XGBClassifier(use_label_encoder =False,learning_rate=0.4,max_depth=7)
#fit the model
xgb.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_xgb = xgb.predict(X_test)
y_train_xgb = xgb.predict(X_train)
model_score=xgb.score(X_test, y_test)
```

FIGURE 7.5: XG BOOST CLASSIFIER ALGORITHM CODE

Comparision of Models

To compare the models performance, a dataframe is created. The columns of this dataframe are the lists created to store the results of the model.

```
In [101]: #creating dataframe
results = pd.DataFrame({ 'ML Model': ML_Model,
                        'Train Accuracy': acc_train,
                        'Test Accuracy': acc_test})
results
```

```
Out[101]:
```

	ML Model	Train Accuracy	Test Accuracy
0	LogisticRegression	0.805	0.794
1	Decision Tree	0.814	0.801
2	Random forest	0.820	0.809
3	KNeighborsClassifier	0.794	0.789
4	XGBoost	0.867	0.859

```
In [102]: #Sorting the dataframe on accuracy
results.sort_values(by=['Test Accuracy', 'Train Accuracy'], ascending=False)
```

```
Out[102]:
```

	ML Model	Train Accuracy	Test Accuracy
4	XGBoost	0.867	0.859
2	Random forest	0.820	0.809
1	Decision Tree	0.814	0.801
0	LogisticRegression	0.805	0.794
3	KNeighborsClassifier	0.794	0.789

From the above comparison, it is clear that the XGBoost Classifier works well with this dataset.

FIGURE 7.6: COMPARISON OF MODELS

CONCLUSION

It is outstanding that a decent enemy of phishing apparatus ought to anticipate the phishing assaults in a decent timescale. The possibility to accept that the accessibility of an enemy of a phishing device at a decent time scale is additionally imperative to build the extent of anticipating phishing sites. This apparatus ought to be improved continually through consistent retraining. As a matter of fact, the accessibility of crisp and cutting-edge preparing dataset which may gained utilizing our very own device will help us to retrain our model consistently and handle any adjustments in the highlights, which are influential in deciding the site class. Albeit neural system demonstrates its capacity to tackle a wide assortment of classification issues, the procedure of finding the ideal structure is very difficult, and much of the time, this structure is controlled by experimentation. The implemented model takes care of this issue via computerizing the way toward organizing a neural system conspire. Hence, on the off chance that we construct an enemy of phishing model and for any reasons we have to refresh it, then at that point the anti-phishing model will encourage this procedure. It would mechanize the organizing procedure and will request scarcely any client defined parameters.

REFERENCES

- 1] <https://www.kaggle.com/> (for dataset)
- 2] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- 3] https://www.w3schools.com/python/python_intro.asp
- 4] “Datasets | Research | Canadian Institute for Cybersecurity | UNB.” *University of New Brunswick / UNB*, <https://www.unb.ca/cic/datasets/index.html>.
- 5] APWG, Aaron G, Manning R (2013) APWG phishing reports. APWG, 1 February 2013.
- 6] Dua, D.; Graff, C. UCI Machine Learning Repository; School of Information and Computer Science, University of California: Irvine, CA, USA. Available online: <https://archive.ics.uci.edu/ml/datasets/Phishing+Websites>
- 7] Tan, C.L. Phishing Dataset for Machine Learning: Feature Evaluation. *Mendeley Data* 2018