LOYOLA INSTITUTE OF TECHNOLOGY,CHENNAI

ANNA UNIVERSITY::CHENNAI – 600025

DECEMBER 2022

TITLE: WEB PHISHING DETECTION

A PROJECT REPORT

SUBMITTED BY

G JOHN WESLEY (210919205026)

V HARIHARAN (210919205017)

K MATHEESWARAN(210919205031)

S HARSHAD(210919205020)

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

TABEL OF CONTENTS

<u>ABSTRACT</u>

The criminals, who want to obtain sensitive data, first create unauthorized replicas of a real website and e-mail.

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.

Phisher then send the "spoofed" e-mails to as many people as possible in an attempt to lure them into 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.

We discuss the methods used for detection of phishing Web sites based on url importance properties.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 About Detection Of Phishing Website

Phishing costs Internet users billions of dollars per year. 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. This paper deals with methods for detecting phishing Web sites by analyzing various features of benign and phishing URLs by Machine learning techniques. We discuss the methods used for detection of phishing Web sites based on lexical features, host properties and page importance properties. We 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.

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.

Disadvantages

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

1.1 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 the 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 click on the link he will see the website and think its original and try to provide his credentials .

To overcome this problem we are 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 algorithm we cam be able to keep the user personal credentials or the sensitive data safe from the intruders.

1.3 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. We are using machine learning algorithms to safeguard the sensitive data and to detect the phishing websites who are trying to gain access on sensitive data.

1.4 Project Features:

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publically, maybe because there is no agreement in literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features.

In this article, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we proposed some new features, experimentally assign new rules to some well-known features and update some other features.

1.1. Address Bar based Features

1.1.1. Using the IP Address

If an IP address is used as an alternative of the domain name in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link "http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html".

$$\textit{Rule}: \text{IF} \begin{cases} \text{If The Domain Part has an IP Address} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.1.2. Long URL to Hide the Suspicious Part

Phishers can use long URL to hide the doubtful part in the address bar. For example:

http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html

To ensure accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than or equal 54 characters then the URL classified as phishing. By reviewing our dataset we were able to find 1220 URLs lengths equals to 54 or more which constitute 48.8% of the total dataset size.

$$Rule: IF \begin{cases} URL\ lengt\ < 54 \rightarrow feature = \text{Legitimate} \\ else\ if\ URL\ lengt\ \geq 54\ and \leq 75 \rightarrow feature = Suspicious \\ ot\ erwise \rightarrow feature = \text{Phishing} \end{cases}$$

We have been able to update this feature rule by using a method based on frequency and thus improving upon its accuracy.

1.1.3. Using URL Shortening Services "TinyURL"

URL shortening is a method on the "World Wide Web" in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an "HTTP Redirect" on a domain name that is short, which links to the webpage that has a long URL. For example, the URL "http://portal.hud.ac.uk/" can be shortened to "bit.ly/19DXSk4".

$$\underline{Rule}: IF \begin{cases} \text{TinyURL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.1.4. URL's having "@" Symbol

Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

Rule: IF $\{$
$$\begin{array}{l} \text{Url Having @ Symbol} \rightarrow \text{Phishing} \\ \quad\quad \text{Otherwise} \rightarrow \text{Legitimate} \end{array}$$

1.1.5. Redirecting using "//"

The existence of "//" within the URL path means that the user will be redirected to another website. An example of such URL's is: "http://www.legitimate.com//http://www.phishing.com". We examin the location where the "//" appears. We find that if the URL starts with "HTTP", that means the "//" should appear in the sixth position. However, if the URL employs "HTTPS" then the "//" should appear in seventh position.

Rule: IF $\{$
$$\begin{array}{l} \text{ThePosition of the Last Occurrence of "//" in the URL} > 7 \rightarrow P \ is \ ing \\ \quad\quad\quad\quad \text{Otherwise} \rightarrow \text{Legitimate} \end{array}$$

1.1.6. Adding Prefix or Suffix Separated by (-) to the Domain

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example http://www.Confirme-paypal.com/.

Rule: IF $\{$
$$\begin{array}{l} \text{Domain Name Part Includes } (-) \text{ Symbol} \rightarrow \text{Phishing} \\ \quad\quad\quad \text{Otherwise} \rightarrow \text{Legitimate} \end{array}$$

1.1.7. Sub Domain and Multi Sub Domains

Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD), which in our example is "uk". The "ac" part is shorthand for "academic", the combined "ac.uk" is called a second-level domain (SLD) and "hud" is the actual name of the domain. To produce a rule for extracting this feature, we firstly have to omit the (www.) from the URL which is in fact a sub domain in itself. Then, we have to remove the (ccTLD) if it exists. Finally, we count the remaining dots. If the number of dots is greater than one, then the URL is classified as "Suspicious" since it has one sub domain. However, if the dots are greater than two, it is classified as "Phishing" since it will have multiple sub domains. Otherwise, if the URL has no sub domains, we will assign "Legitimate" to the feature.

$$\text{Rule: IF} \begin{cases} \text{Dots In Domain Part} = 1 \ \rightarrow \ \text{Legitimate} \\ \text{Dots In Domain Part} = 2 \ \rightarrow \ \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

1.1.8. HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer)

The existence of HTTPS is very important in giving the impression of website legitimacy, but this is clearly not enough. The authors in (Mohammad, Thabtah and McCluskey 2012) (Mohammad, Thabtah and McCluskey 2013) suggest checking the certificate assigned with HTTPS including the extent of the trust certificate issuer, and the certificate age. Certificate Authorities that are consistently listed among the top trustworthy names include: "GeoTrust, GoDaddy, Network Solutions, Thawte, Comodo, Doster and VeriSign". Furthermore, by testing out our datasets, we find that the minimum age of a reputable certificate is two years.

Rule:

$$\text{IF} \begin{cases} \text{Use https and Issuer Is Trusted } and \ Age \ of \ Certificate \geq 1 \ \text{Years} \rightarrow \text{Legitimate} \\ \text{Using https and Issuer Is Not Trusted} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

1.1.9. Domain Registration Length

Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent domains have been used for one year only.

$$\text{Rule: IF}\begin{cases} \text{Domains Expires on} \leq 1 \text{ years} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.1.10. Favicon

A favicon is a graphic image (icon) associated with a specific webpage. Many existing user agents such as graphical browsers and newsreaders show favicon as a visual reminder of the website identity in the address bar. If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt.

$$\text{Rule: IF}\begin{cases} \text{Favicon Loaded From External Domain} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.1.11. Using Non-Standard Port

This feature is useful in validating if a particular service (e.g. HTTP) is up or down on a specific server. In the aim of controlling intrusions, it is much better to merely open ports that you need. Several firewalls, Proxy and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only open the ones selected. If all ports are open, phishers can run almost any service they want and as a result, user information is threatened. The most important ports and their preferred status are shown in Table 2.

$$\text{Rule: IF}\begin{cases} \text{Port \# is of the Preffered Status} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

Table 1 Common ports to be checked

PORT	Service	Meaning	Preferred Status
21	FTP	Transfer files from one host to another	Close
22	SSH	Secure File Transfer Protocol	Close
23	Telnet	provide a bidirectional interactive text-oriented communication	Close
80	HTTP	Hyper test transfer protocol	Open
443	HTTPS	Hypertext transfer protocol secured	Open
445	SMB	Providing shared access to files, printers, serial ports	Close
1433	MSSQL	Store and retrieve data as requested by other software applications	Close
1521	ORACLE	Access oracle database from web.	Close
3306	MySQL	Access MySQL database from web.	Close
3389	Remote Desktop	allow remote access and remote collaboration	Close

1.1.12. The Existence of "HTTPS" Token in the Domain Part of the URL

1.2. Abnormal Based Features

1.2.1. Request URL

Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain. In legitimate webpages, the webpage address and most of objects embedded within the webpage are sharing the same domain.

$$\text{\% of Request URL} < 22\% \rightarrow Legitimate$$
$$\text{Rule: IF \{\%of Request URL} \geq 22\% \text{ and } 61\% \rightarrow \text{Suspicious}$$
$$\text{Otherwise} \rightarrow \text{feature} = \text{Phishing}$$

1.2.2. URL of Anchor

An anchor is an element defined by the <a> tag. This feature is treated exactly as "Request URL". However, for this feature we examine:

1. If the <a> tags and the website have different domain names. This is similar to request URL feature.

2. If the anchor does not link to any webpage, e.g.:

    A. <a href="#">

    B. <a href="#content">

    C. <a href="#skip">

    D. <a href="JavaScript ::void(0)">

$$\text{\% of URL Of Anchor} < 31\% \rightarrow Legitimate$$
$$\underline{Rule}: \text{IF\{\% of URL Of Anchor} \geq 31\% \text{ And} \leq 67\% \rightarrow \text{Suspicious}$$
$$\text{Otherwise} \rightarrow \text{Phishing}$$

1.2.3. Links in <Meta>, <Script> and <Link> tags

Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources. It is expected that these tags are linked to the same domain of the webpage.

Rule:

IF

$$\begin{cases} \% \text{ of Links in } "<Meta>","<Script>" \text{ and } "<Link>" < 17\% \rightarrow Legitimate \\ \% \text{ of Links in } <Meta>","<Script>" \text{ and } "<Link>" \geq 17\% \text{ And } \leq 81\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

1.2.4. Server Form Handler (SFH)

SFHs that contain an empty string or "about:blank" are considered doubtful because an action should be taken upon the submitted information. In addition, if the domain name in SFHs is different from the domain name of the webpage, this reveals that the webpage is suspicious because the submitted information is rarely handled by external domains.

$$\text{Rule: IF} \begin{cases} \text{SFH is "about: blank" Or Is Empty} \rightarrow \text{Phishing} \\ \text{SFH Refers To A Different Domain} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.2.5. Submitting Information to Email

Web form allows a user to submit his personal information that is directed to a server for processing. A phisher might redirect the user's information to his personal email. To that end, a server-side script language might be used such as "mail()" function in PHP. One more client-side function that might be used for this purpose is the "mailto:" function.

$$\text{Rule: IF} \begin{cases} \text{Using "mail()" or "mailto:" Function to Submit User Information} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.2.6. Abnormal URL

This feature can be extracted from WHOIS database. For a legitimate website, identity is typically part of its URL.

Rule: IF $\begin{cases} \text{The Host Name Is Not Included In URL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$

1.3. HTML and JavaScript based Features

1.3.1. Website Forwarding

The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. In our dataset, we find that legitimate websites have been redirected one time max. On the other hand, phishing websites containing this feature have been redirected at least 4 times.

Rule: IF $\begin{cases} \text{ofRedirect Page} \leq 1 \rightarrow \text{Legitimate} \\ \text{of Redirect Page} \geq 2 \text{ And} < 4 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$

1.3.2. 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.

Rule: IF $\begin{cases} \text{onMouseOver Changes Status Bar} \rightarrow \text{Phishing} \\ \text{It Does't Change Status Bar} \rightarrow \text{Legitimate} \end{cases}$

1.3.3. 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.

Rule: IF $\begin{cases} \text{Right Click Disabled} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$

1.3.4. Using Pop-up Window

It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window. On the other hand, this feature has been used in some legitimate websites and its main goal is to warn users about fraudulent activities or broadcast a

welcome announcement, though no personal information was asked to be filled in through these pop-up windows.

Rule: IF $\left\{ \begin{array}{l} \text{Popoup Window Contains Text Fields} \rightarrow \text{Phishing} \\ \qquad\quad \text{Otherwise} \rightarrow \text{Legitimate} \end{array} \right.$

1.3.5. 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.

Rule: IF $\left\{ \begin{array}{l} \text{Using iframe} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{array} \right.$

1.4. Domain based Features

1.4.1. Age of Domain

This feature can be extracted from WHOIS database (Whois 2005). Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.

Rule: IF $\left\{ \begin{array}{l} \text{Age Of Domain} \geq 6 \text{ months} \rightarrow \text{Legitimate} \\ \qquad\quad \text{Otherwise} \rightarrow \text{Phishing} \end{array} \right.$

1.4.2. DNS Record

For phishing websites, either the claimed identity is not recognized by the WHOIS database (Whois 2005) or no records founded for the hostname (Pan and Ding 2006). If the DNS record is empty or not found then the website is classified as "Phishing", otherwise it is classified as "Legitimate".

Rule: IF $\left\{ \begin{array}{l} \text{no DNS Record For The Domain} \rightarrow \text{Phishing} \\ \qquad\quad \text{Otherwise} \rightarrow \text{Legitimate} \end{array} \right.$

1.4.3. Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). By reviewing our dataset, we find that in worst scenarios, legitimate websites ranked among the top 100,000. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as "Phishing". Otherwise, it is classified as "Suspicious".

$$\text{Rule: IF} \begin{cases} \text{Website Rank} < 100,000 & \rightarrow \text{ Legitimate} \\ \text{Website Rank} > 100,000 & \rightarrow Suspicious \\ \text{Otherwise} & \rightarrow \text{Phish} \end{cases}$$

1.4.4. PageRank

PageRank is a value ranging from "0" to "1". PageRank aims to measure how important a webpage is on the Internet. The greater the PageRank value the more important the webpage. In our datasets, we find that about 95% of phishing webpages have no PageRank. Moreover, we find that the remaining 5% of phishing webpages may reach a PageRank value up to "0.2".

$$\text{Rule: IF} \begin{cases} \text{PageRank} < 0.2 \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

1.4.5. Google Index

This feature examines whether a website is in Google's index or not. When a site is indexed by Google, it is displayed on search results (Webmaster resources, 2014). Usually, phishing webpages are merely accessible for a short period and as a result, many phishing webpages may not be found on the Google index.

$$\text{Rule: IF} \begin{cases} \text{Webpage Indexed by Google} \rightarrow \text{Legitimate} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

1.4.6. Number of Links Pointing to Page

The number of links pointing to the webpage indicates its legitimacy level, even if some links are of the same domain (Dean, 2014). In our datasets and due to its short life span, we find

that 98% of phishing dataset items have no links pointing to them. On the other hand, legitimate websites have at least 2 external links pointing to them.

Rule:

IF

$$\{ \begin{array}{l} \text{Of Link Pointing to The Webpage} = 0 \rightarrow \text{Phishing} \\ \text{Of Link Pointing to The Webpage} > 0 \; and \leq 2 \rightarrow Suspicious \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{array}$$

1.4.7. Statistical-Reports Based Feature

Several parties such as PhishTank (PhishTank Stats, 2010-2012), and StopBadware (StopBadware, 2010-2012) formulate numerous statistical reports on phishing websites at every given period of time; some are monthly and others are quarterly. In our research, we used 2 forms of the top ten statistics from PhishTank: "Top 10 Domains" and "Top 10 IPs" according to statistical-reports published in the last three years, starting in January2010 to November 2012. Whereas for "StopBadware", we used "Top 50" IP addresses.

$$\text{Rule: IF} \{ \begin{array}{l} \text{Host Belongs to Top Phishing IPs or Top Phishing Domains} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{array}$$

Phishing is one of the most common and most dangerous attacks among cybercrimes. The aim of these attacks is to steal the information used by individuals and organizations to conduct transactions. Phishing websites are fake websites that contain various hints among their contents and web browser-based information. When a user opens a fake webpage and enters the username and protected password, the credentials of the user are acquired by the attacker which can be used for malicious purposes. Phishing websites look very similar in appearance to their corresponding legitimate websites to attract large number of Internet users.
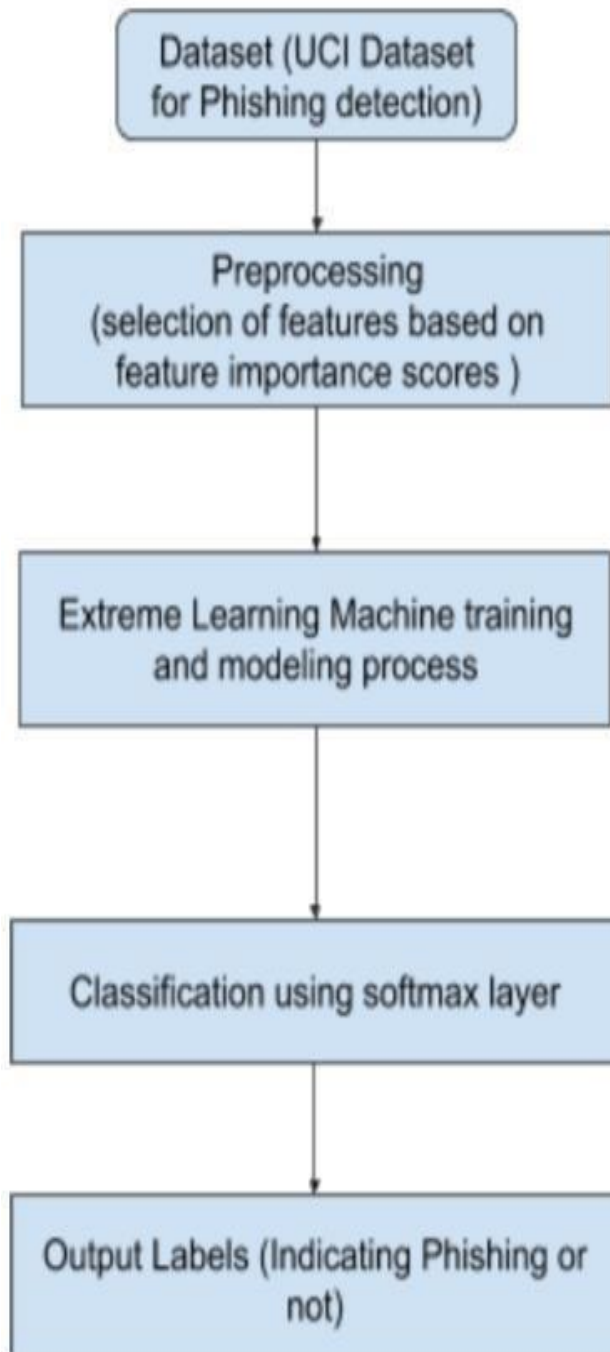
GOALS

1. Use of features extracted from websites which explain characteristics of a website for phishing detection

2. Classification of website based on such features, using Extreme Learning Machines (ELM) which is an advanced neural network leveraging generalization capabilities given by randomization of weights

METHODOLOGY

The steps involved in achieving phishing detection are as follows:

```
┌─────────────────────────┐
│   Dataset (UCI Dataset  │
│  for Phishing detection)│
└─────────────────────────┘
             │
             ▼
┌─────────────────────────────┐
│        Preprocessing        │
│ (selection of features based│
│  on feature importance      │
│  scores )                   │
└─────────────────────────────┘
             │
             ▼
┌─────────────────────────────┐
│ Extreme Learning Machine    │
│ training and modeling       │
│ process                     │
└─────────────────────────────┘
             │
             ▼
┌─────────────────────────────┐
│ Classification using        │
│ softmax layer               │
└─────────────────────────────┘
             │
             ▼
┌─────────────────────────────┐
│ Output Labels (Indicating   │
│ Phishing or not)            │
└─────────────────────────────┘
```

The study uses a dataset which contains approximately 11,000 data containing the 30 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.

CHAPTER 2

LITERATURE SURVEY

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-learningbased 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 relationship 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 behaviour. 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 thismodel is purely depends on the quality and quantity of the training set and Broken links feature extraction.

2.1 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 are considered for building up the proposed framework**.**

Machine learning

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

2.2 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-testdebug cycle makes this simple approach very effective.

2.2.1 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, he 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**.**

2.2.2 MATPLOTLIB

People are exceptionally visual animals: we comprehend things better when we see things envisioned. Notwithstanding, the progression to showing investigations, results or bits of knowledge can be a bottleneck: you probably won't realize where to begin or you may have as of now a correct configuration as a top priority, however then inquiries like "Is this the correct method to imagine the bits of knowledge that I need to convey to my group of onlookers?" will have unquestionably gone over your brain.

When you're working with the Python plotting library Matplotlib, the initial step to responding to the above inquiries is by structure up information on themes like: The life structures of a Matplotlib plot: what is a subplot? What are the Axes? What precisely is a figure?

Plot creation, which could bring up issues about what module you precisely need to import (pylab or pyplot?), how you precisely ought to approach instating the figure and the Axes of your plot, how to utilize matplotlib in Jupyter note pads, and so on.

Plotting schedules, from straightforward approaches to plot your information to further developed methods for picturing your information. Essential plot customizations, with an emphasis on plot legends and content, titles, tomahawks marks and plot format.

Sparing, appearing, your plots: demonstrate the plot, spare at least one figures to, for instance, pdf documents, clear the tomahawks, clear the figure or close the plot, and so on. In conclusion, you'll quickly cover two manners by which you can alter Matplotlib: with templates and the rc settings.

Since all is set for you to begin plotting your information, it's an ideal opportunity to investigate some plotting schedules. You'll regularly go over capacities like plot() and disperse(), which either draw focuses with lines or markers interfacing them, or draw detached focuses, which are scaled or shaded. In any case, as you have just found in the case of the primary area, you shouldn't neglect to pass the information that you need these capacities to utilize!

These capacities are just the exposed rudiments. You will require some different capacities to ensure your plots look magnificent:

2.4.3 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 models 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 DataCamp 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 be able 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.

2.2.4 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. In this instructional exercise, we will get familiar with the different highlights of Python Pandas and how to utilize them practically speaking. 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.

2.4.5 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.
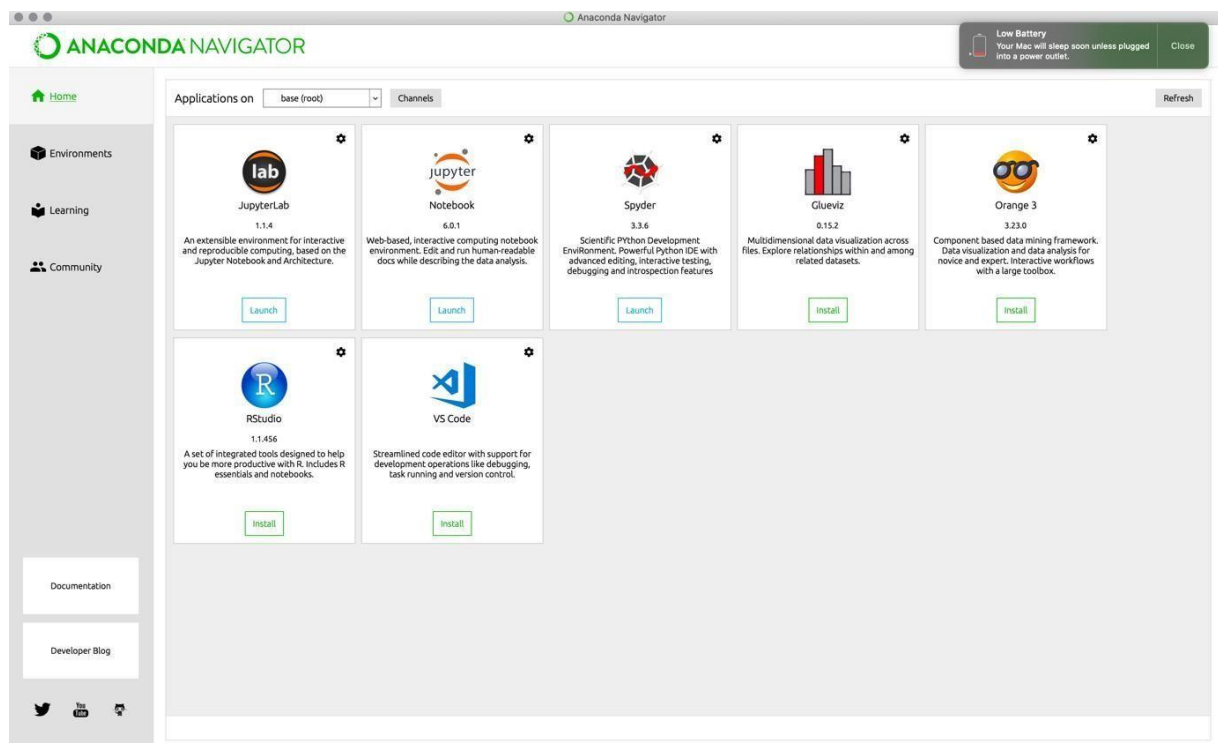
2.2.6 PYTHON

Python is a translated, object-arranged, 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.

Python is generally basic, so it's anything but difficult to learn since it requires a one of a kind language structure that centers around coherence. Designers can peruse and interpret Python code a lot simpler than different dialects. Thusly, this decreases the expense of program upkeep and improvement since it enables groups to work cooperatively without huge language and experience obstructions.

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 source structure. 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.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 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.

- Our 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 testset.
- It should be able to train model using trainset.
- It must validate trained model using testset.
- It should be able to display the trained model accuracy.
- It should be able to accurately predict the air quality on unseen data.

3.2 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. We must include only those requirements that are appropriate for our project. Some Non-Functional Requirements are as follows:

- Reliability

- Maintainability

- Performance

- Portability

- Scalability

- Flexibility

Some of the quality attributes are as follows:

3.2.1 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.

3.2.2 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.

3.2.3 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.

3.2.4 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 dependant congregations would need to be designed in such case.

The functional requirements for a system describe what the system should do.

Those requirments 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 is not about functionality or behaviour 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

➤ Basic Operational 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.

➤ 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.

User Requirement

- The user should able to have User Interface Window with Visualize Graphics.
- The user should able to configure with neat GUI all the parameters.

Resource Requirement

Anaconda 3-5.0.3: Anaconda is a free and open source distribution of the Python and R programming languages for data science, machine learning and other applications. Anaconda distribution comes with 1400 packages as well as the conda package and virtual environment manager, called Anaconda Navigator. Packages can be made using the conda build command. **Anaconda Navigator**is a desktop graphical user interface allows user to manage conda packages. The following applications are available by default in navigator: Jupyter lab, Jupyter netbook, Spyder, Orange, Rstudio etc. conda is an open source, cross platform, language-agnostic package manager and environment management system. It installs, runs and update packages and their dependencies.

1. **Jupyter Notebook**: The code is fully written in Python language using Jupyter notebook. It is the spin-off projects from the IPyton project, which used to have an IPython Notebook project itself. IPython kernel, which allows you to write your programs in Python. We can install Jupyter Notebook using command $pip installJupyter. It has serveral menus that you can use to interact with your notebook they are listed as:

- File
- Edit
- View
- Insert
- Cell
- Kernel, Widgets, Help

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.

3.3 Hardware Requirements:

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

- Processor:Any Processor above 500 MHz

- RAM   :8 GB

- Hard Disk   :1 TB

- Input device   : Standard keyboard and mouse

3.4 Software Requirements:

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

- OS   :   Windows 10

- Platform   :   Jupyter Notebook

- Language   :   Python

- IDE/tool   :   Anaconda 3-5.0.3

CHAPTER 4

DESIGN

Technologies Used

- **PYTHON**

- **TENSORFLOW (SCIKIT- LEARN)**

- **MACHINE LEARNING**

- **LIBRARIES – PANDAS , NUMPY**

1.1. Open CV

OpenCV (Open Source Computer Vision Library) is an open source PC vision and AI programming library. OpenCV was worked to give a typical foundation to PC vision applications and to quicken the utilization of machine discernment in the business items. Being a BSD-authorized item, OpenCV makes it simple for organizations to use and adjust the code. The library has more than 2500 enhanced calculations, which incorporates an exhaustive arrangement of both exemplary and best in class PC vision and AI calculations. These calculations can be utilized to distinguish and perceive faces, distinguish objects, arrange human activities in recordings, track camera developments, track moving articles, extricate 3D models of items, produce 3D point mists from stereo cameras, fasten pictures together to create a high goals picture of a whole scene, find comparative pictures from a picture database, expel red eyes from pictures taken utilizing streak, pursue eye developments, perceive landscape and set up markers to overlay it with enlarged reality, and so on.

OpenCV has in excess of 47 thousand individuals of client network and evaluated number of downloads surpassing 18 million. The library is utilized broadly in organizations, examine gatherings and by administrative bodies. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

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, TPUs), and from work areas to bunches of servers to portable and edge gadgets. TensorFlow calculations are communicated as stateful dataflow diagrams. The name TensorFlow gets from the activities that such neural systems perform on multidimensional information exhibits, which are alluded to as tensors.

Neural Networks:

The neural system itself isn't a calculation, yet rather a structure for some, extraordinary AI calculations to cooperate and process complex information inputs. Such frameworks learn to perform undertakings by thinking about models, for the most part without being modified with any errand explicit principles. For instance, in picture acknowledgment, they may figure out how to distinguish pictures by dissecting precedent pictures and utilizing the outcomes to recognize it in different pictures.

They do this with no earlier information about felines, for instance, that they have hide, tails, hairs and feline like countenances. Rather, they consequently create distinguishing qualities from the learning material that they procedure.

Convolutional Neural Networks:

As of 2011, the state of the art in deep learning feedforward networks alternated between convolutional layers and max-pooling layers,topped by several fully or sparsely connected layers followed by a final classification layer. Learning is normally managed without unsupervised pre-preparing. In the convolutional layer, there are channels that are convolved with the information. Each channel is comparable to a loads vector that must be prepared. Such directed profound learning strategies were the first to accomplish human-aggressive execution on certain tasks.

4.1 UML Diagrams:

➢ Class diagram

extensions

extreme_learning_machines

+ELMClassifier()

sklearn

ELMClassifier_Model

+fit()
+predict()
+score()

Model Selection

+KFold()
+cross_val_score()
+train_test_split()

preprocessing

+normalize()
+StandardScaler()
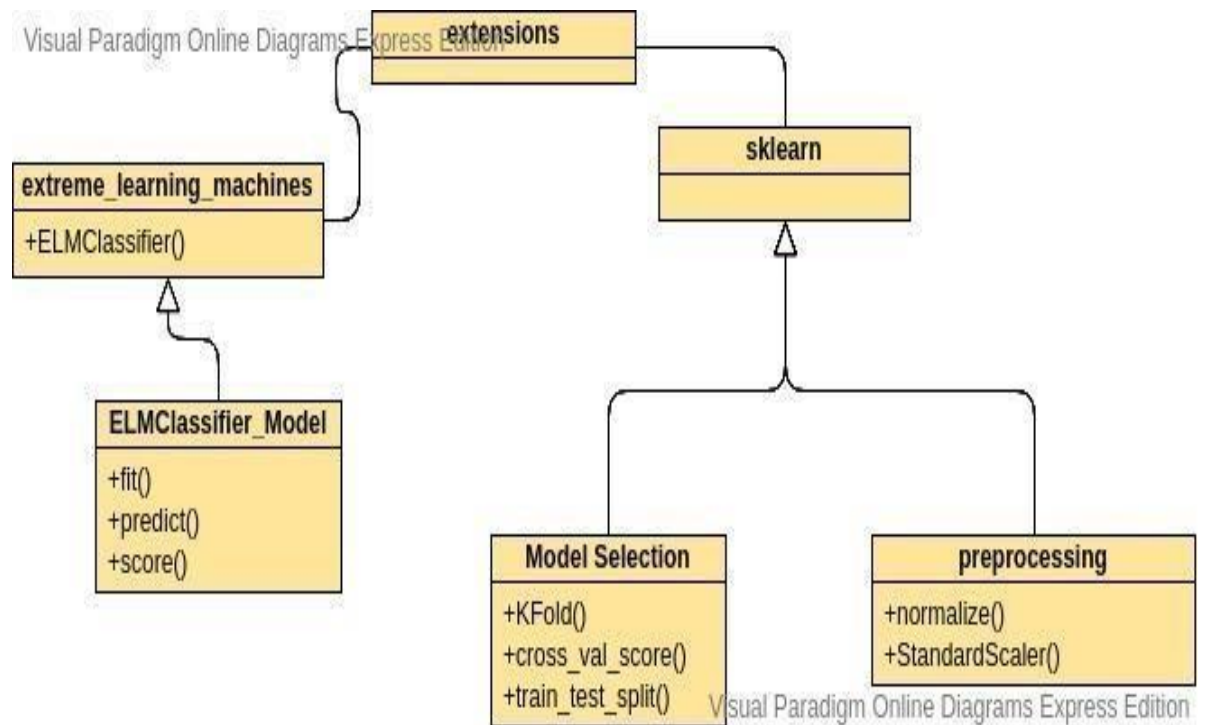
➢ Sequence Diagram:
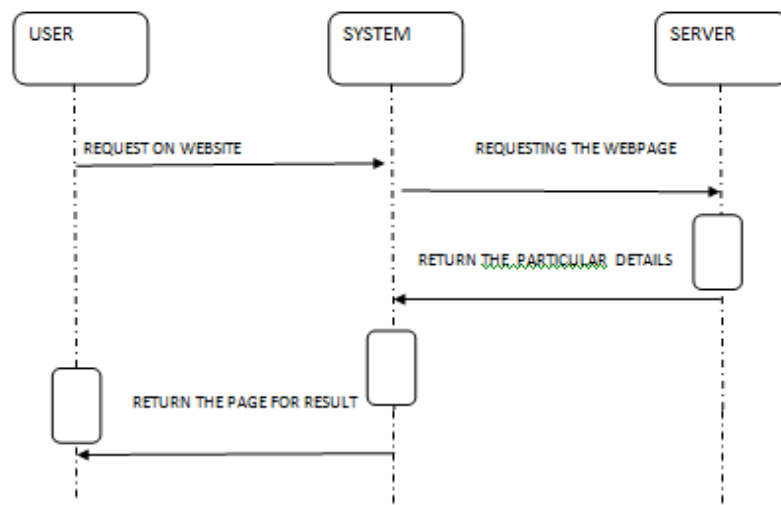
➢ Use Case Diagram:
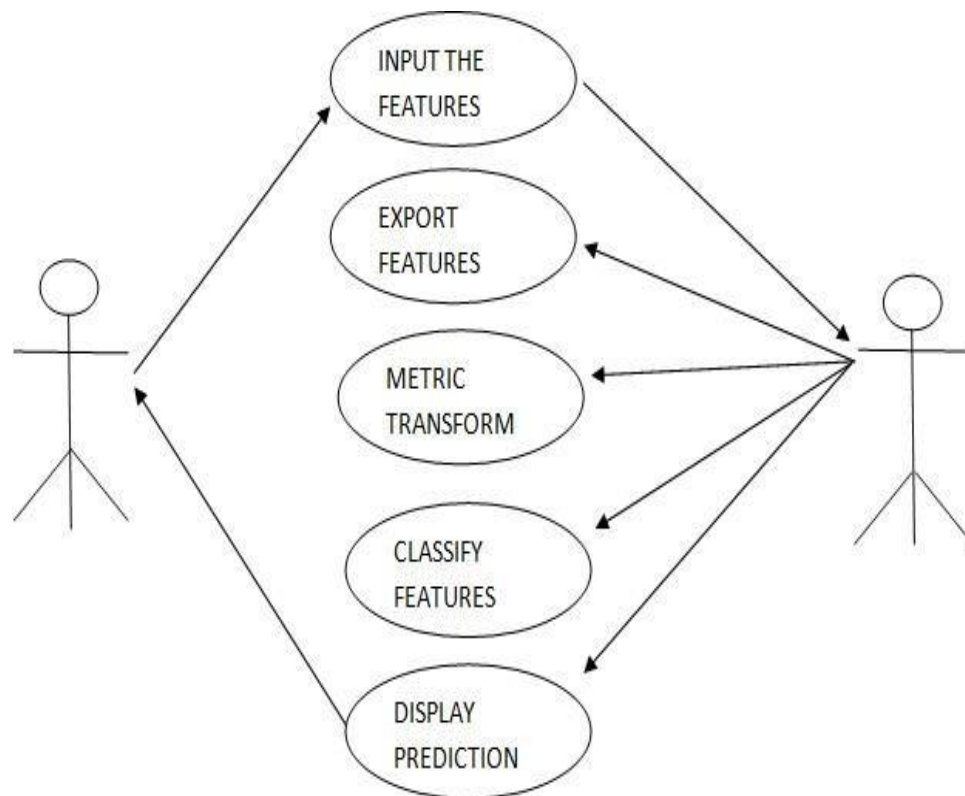
CHAPTER 5

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.
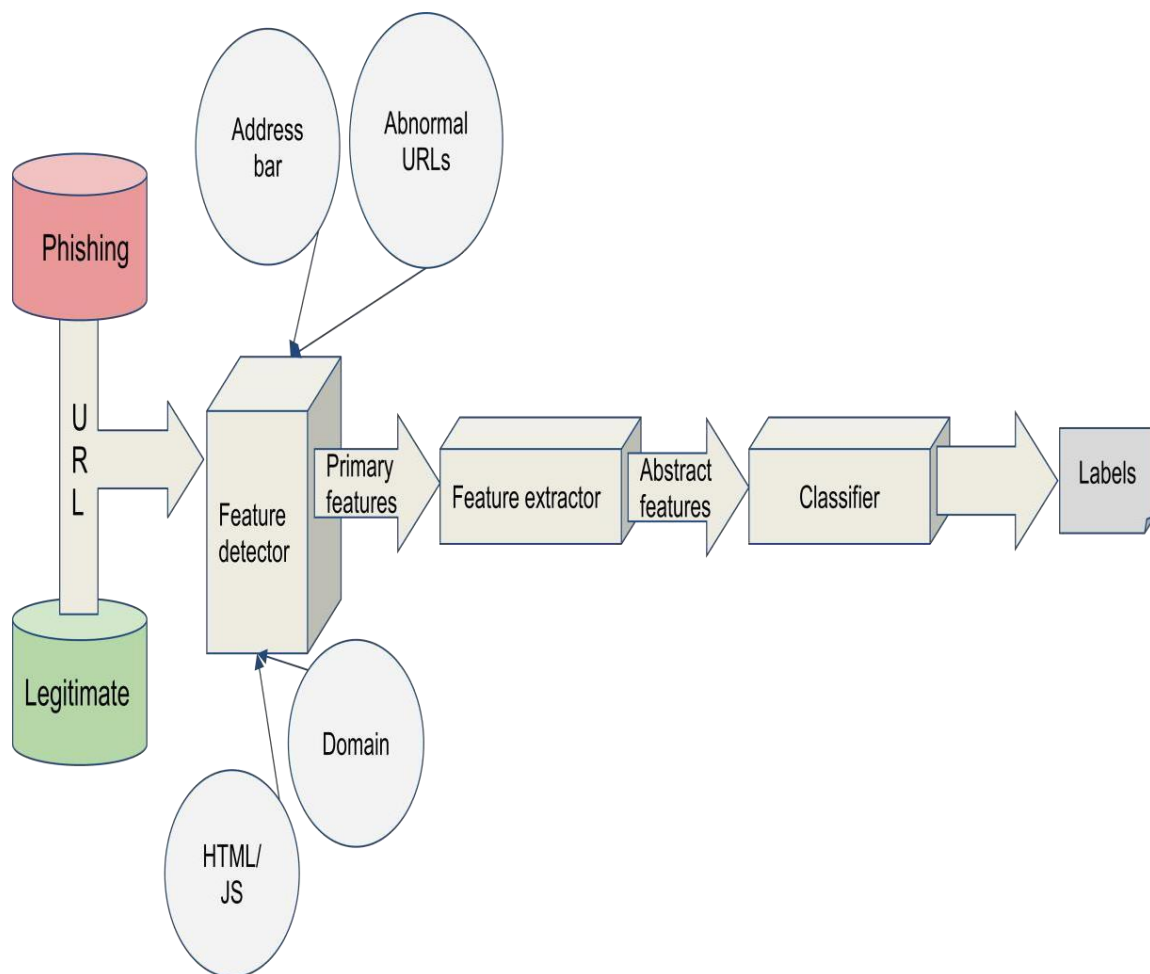
5.1 Overview of system implementation

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.

5.1.1 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-testdebug cycle makes this simple approach very effective.

5.1.2 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. Minimum 3 GB disk space to download and install.
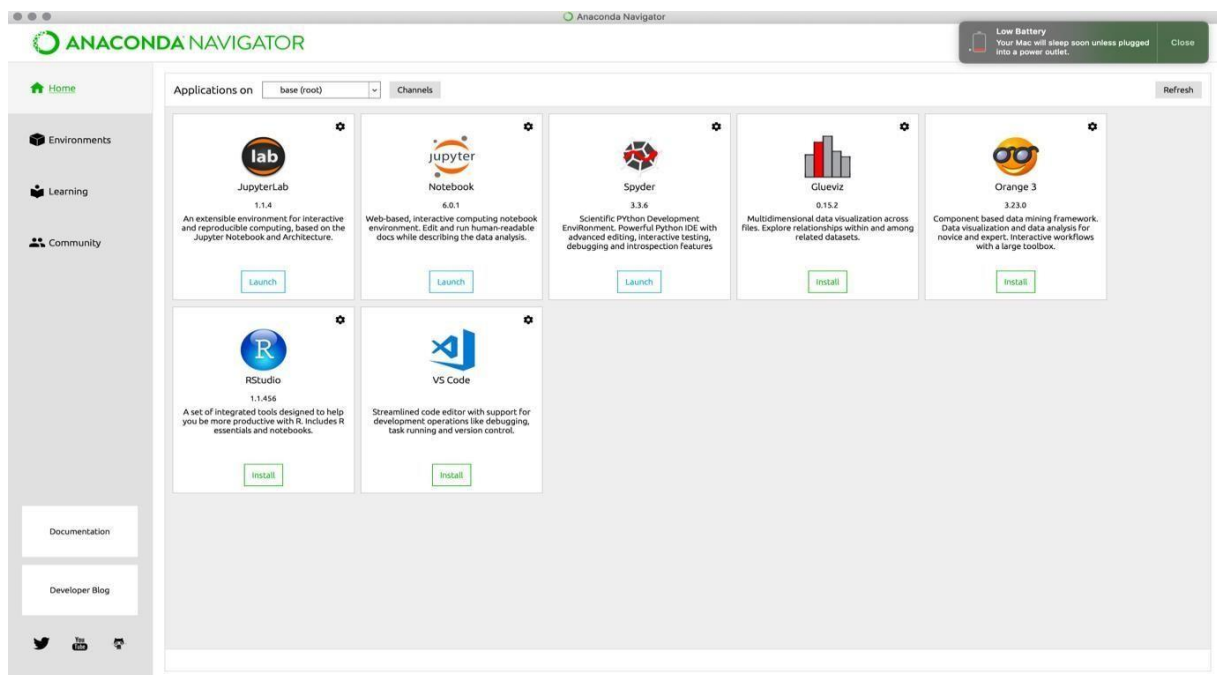
Fig 3 :Anaconda Navigator

After the installation of anaconda navigator, we were taught python programming. We were taught various inclusion of python libraries such as **NumPy** i.e. introduction to NumPy, NumPy arrays, few notes on array indexing, NumPy array indexing, NumPy operations and few exercises to recall it. We were taught how to use **Pandas**, how to include data frames, finding and replacing missing data with useful information, group-by functions, merging, joining and concatenating and other data input and output operations. We were also taught python for data visualization that is matplotlib, seaborn. **Matplotlib** is a plotting library for python and its extension NumPy. It makes use of general-purpose GUI kits and provides an object-oriented API for embedding the plots. In seaborn we were taught distribution plots, categorial plots, matrix plots, grids, regression plots etc.

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 test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

6.1 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.

6.2 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.

6.3 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 identifyingdesign 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:
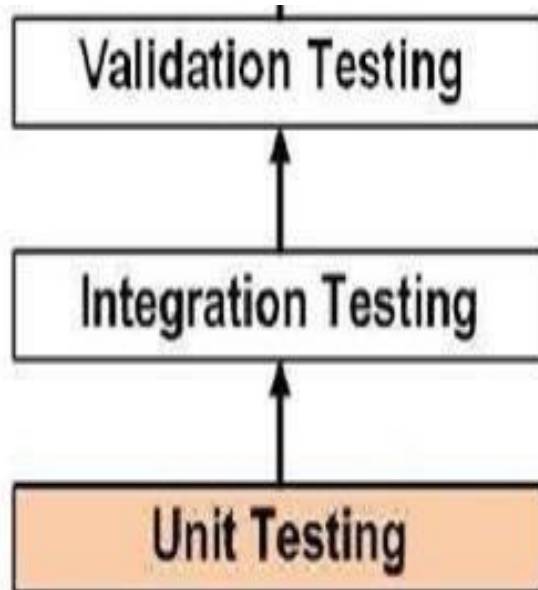
Figure 6.1: The testing process

6.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s).
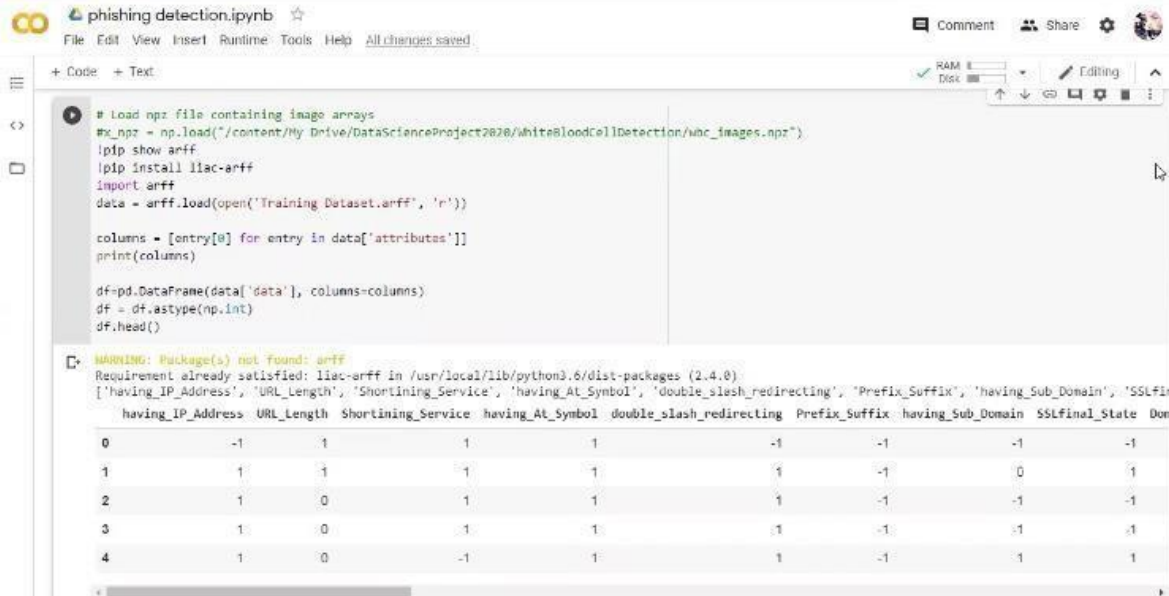
System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).

System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

CHAPTER 7

SNAPSHOTS

```
[53]    4   1
        Name: Result, dtype: int64
```

Features considered

```
x.columns
```

```
Run cell (Ctrl+Enter)        ddress', 'URL_Length', 'having_At_Symbol', 'Prefix_Suffix',
cell executed since last change Domain', 'SSLfinal_State', 'Domain_registeration_length',
                             ort', 'Request_URL', 'URL_of_Anchor', 'Links_in_tags',
executed by teja praveen     tting_to_email', 'on_mouseover', 'age_of_domain',
3.05 PM (0 minutes ago)      'web_traffic', 'Page_Rank', 'Google_Index',
executed in 1.67s            ng_to_page', 'Statistical_report'],
                 dtype='object')
```

```
[24]  from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=8)
```

```
[26]  x_train.shape
```

```
(8844, 22)
```

```
[ ]   y_train.shape
```

```
(8844,)
```

```
[53]  Statistical_report      0
      Result                  0
      dtype: int64
```

```
df.iloc[:,-1].value_counts()
```

```
1    6157
-1   4898
Name: Result, dtype: int64
```

```
[48]  corr=df.corr()
      corr.nlargest(30,"Result")["Result"]
```

```
Result                   1.000000
SSLfinal_State           0.714741
URL_of_Anchor            0.692935
Prefix_Suffix            0.348606
web_traffic              0.346103
having_Sub_Domain        0.298323
Request_URL              0.253372
Links_in_tags            0.248229
SFH                      0.221419
Google_Index             0.128950
age_of_domain            0.121496
Page_Rank                0.104645
having_IP_Address        0.094160
Statistical_report       0.079857
DNSRecord                0.075718
URL_Length               0.057430
having_At_Symbol         0.052948
on_mouseover             0.041838
```

phishing detection.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Name: 6265, dtype: int64

```
x_test.iloc[4,:]
```

```
having_IP_Address            1
URL_Length                  -1
having_At_Symbol             1
Prefix_Suffix               -1
having_Sub_Domain            1
SSLfinal_State              -1
Domain_registeration_length -1
Favicon                      1
port                         1
Request_URL                  1
URL_of_Anchor                0
Links_in_tags                0
SFH                         -1
Submitting_to_email          1
on_mouseover                 1
age_of_domain                1
DNSRecord                    1
web_traffic                  0
Page_Rank                   -1
Google_Index                 1
Links_pointing_to_page       0
Statistical_report           1
Name: 10316, dtype: int64
```

testGUI_V_1_0.py - C:\Users\Pc\Desktop\Phishing\testGUI_V_1_0.py (3.6.1)

File Edit Format Run Options Window Help

```python
from __future__ import print_function

from tkinter import *
from tkinter import messagebox

# import tensorflow as tf
import pickle
import dill
import numpy as np

import pandas as pd
from sklearn.preprocessing import LabelEncoder

window = Tk()

window.title("Phishing Prediction App")
window.geometry('600x800')

sample = []

def clicked():
    pass

def button_func():
    txt = tbl.get()
    print(txt)

def predict():
    #---------- Predictions ----------------------------
    sample = [int(tb.get()) for tb in tb_list]

    model_ELM = pickle.load(open("model.pk", "rb"))
    pred_lab = model_ELM.predict(np.array(sample).reshape(1, -1))

    print("\n\nPredicted :", pred_lab)

    if (pred_lab[0] == 1):
        pred = "Phishing Detected"
    else:
        pred = "No Phishing Detected"
```

---

phishing detection.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

🗨 Comment ⚇ Share ⚙

+ Code + Text

```
[53] Statistical_report        0
     Result                    0
     dtype: int64
```

```
df.iloc[:,-1].value_counts()
```

```
1     6157
-1    4898
Name: Result, dtype: int64
```

```
[48] corr=df.corr()
     corr.nlargest(30,"Result")["Result"]
```

```
Result                       1.000000
SSLfinal_State               0.714741
URL_of_Anchor                0.692935
Prefix_Suffix                0.348606
web_traffic                  0.346103
having_Sub_Domain            0.298323
Request_URL                  0.253372
Links_in_tags                0.248229
SFH                          0.221419
Google_Index                 0.128950
age_of_domain                0.121496
Page_Rank                    0.104645
having_IP_Address            0.094160
Statistical_report           0.079857
DNSRecord                    0.075718
URL_Length                   0.057430
having_At_Symbol             0.052948
on_mouseover                 0.041838
```

CHAPTER 8

CONCLUSION

It is outstanding that a decent enemy of phishing apparatus ought to anticipate the phishing assaults in a decent timescale. We accept that the accessibility of a decent enemy of 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 [30, 32] 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.Our 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, at that point our model will encourage this procedure, that is, since our model will mechanize the organizing procedure and will request scarcely any client defined parameters.

CHAPTER 9

REFERENCES

- Liu J, Ye Y (2001) Introduction to E-business operators: commercial center arrangements,

security issues, and market interest. In: E-business specialists, commercial centerarrangements, security issues, and market interest, London, UK

• APWG, Aaron G, Manning R (2013) APWG phishing reports. APWG, 1 February 2013.

[Online]. Accessible: http://www. antiphishing.org/assets/apwg-reports/. Gotten to 8 Feb2013

• Kaspersky Lab (2013) Spam in January 2012: love, governmental issues and game.

[Online].

Available:http://www.kaspersky.com/about/news/spam/2012

Spam_in_January_2012_Love_Politics_and_ Sport. Gotten to 11 Feb 2013

• Seogod (2011) Black Hat SEO. Search engine optimization Tools. [Online]. Accessible:http://www.seobesttools.com/dark cap website optimization/. Gotten to 8 Jan 2013

• Dhamija R, Tygar JD, Hearst M (2006) Why phishing works. In: Proceedings of the SIGCHImeeting on human factors in figuring frameworks, Cosmopolitan Montre 'al, Canada

• Cranor LF (2008) A system for thinking about the human tuned in. In: UPSEC'08

Proceedings of the first meeting on ease of use, brain science, and security, Berkeley, CA,USA

• Miyamoto D, Hazeyama H, Kadobayashi Y (2008) An assessment of AI based techniquesfor recognition of phishing destinations. Aust J Intell Inf Process Syst 10(2):54–6

• Xiang G, Hong J, Rose CP, Cranor L (2011) CANTINA?:a include rich AI structure foridentifying phishing sites. ACM Trans Inf Syst Secur 14(2):1–28