

DETECTION OF MALICIOUS URL USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

NOMULA SHIVANI REDDY (37110521)

PAMULAPATI RUPASREE (37110539)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

MARCH - 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **NOMULA SHIVANI REDDY (Reg. No. 37110521)** and **PAMULAPATI RUPASREE (Reg. No.37110539)** who carried out the project entitled “**ONLINE CRIME REPORTING SYSTEM**” under my supervision from **August 2020** to **March 2021**.

Internal Guide

Ms.D.Deepa.,M.E.,(Ph.D)

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **NOMULA SHIVANI REDDY** hereby declare that the Project Report entitled “**DETECTION OF MALICIOUS URL USING MACHINE LEARNING**” is done by me under the guidance of Ms.D.Deepa,M.E.,(Ph.D) Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. D.Deepa, M.E., (Ph.D.)**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

We address our experience training and testing a malicious URL detection system in this article. Our research is inspired by a range of technical and security developments. To begin with, the internet has become a more dangerous environment. Semaneme announced a 36 percent rise in cyber threats year over year in 2011. This equates to about 4,500 new attacks every day. The rate at which new attacks are launched has far outpaced the capabilities of conventional anti-malware tools. Second, both personal and business use of mobile web data has improved significantly. Semaneme observed in their 2012 State of Flexibility Survey that, while smartphones were once largely banned by IT, they are now used by hundreds and thousands of workers around the world. As a result, the attackable demographic for attackers has not only expanded, but also contains a potentially more appealing community from a commercial or financial perspective.

With the increased usage of smart phones and tablets for both personal and professional purposes, web deficiencies are on the rise. This work aims on a machine learning approach that includes a lot of URL feature vectors, Python core enhancements, and density value to recognise malicious URLs.

We obtain a performance of 0.81 and a F1-measure of 0.74 using an SVM with a polynomial kernel. The user is, however, expected to take some action in all situations, such as click on a preferred resource on the internet (URL). The web security organizations have developed blacklisting programs to help identify malicious websites.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 INTRODUCTION TO MACHINE LEARNING	1
	1.2 OVERVIEW OF MACHINE LEARNING	1
	1.3 INTRODUCTION TO PROJECT	3
	1.4 PROBLEM DESCRIPTION	4
	1.5 PROJECT DEVELOPMENT	5
2	LITERATURE SURVEY	6
3	METHODOLOGY	12
	3.1 GENERAL	12
	3.2 EXISTING WORK	12
	3.3 FUTURE WORK	12
	3.4 HARDWARE REQUIREMENTS	13
	3.5 SOFTWARE REQUIREMENTS	14
	3.6 ARCHITECTURE DIAGRAM	14
	3.7 SYSTEM DESIGN	15
	3.8 DATA FLOW DIAGRAMS	16
	3.9 UML DIAGRAMS	17
	3.10 SYSTEM IMPLEMENTATIONS	22
	3.11 LANGUAGE SPECIFICATION	24
	3.12 FEATURES OF ANACONDA NAVIGATOR	26
	3.13 FLOW DIAGRAMS	29
	3.14 BLOCK DIAGRAM	30
	3.15 SYSTEM TESTING	31
	3.16 TYPES OF TESTING	31
	3.17 SYSTEM TEST	32

4	RESULTS AND DISCUSSIONS	34
5	CONCLUSION AND FUTURE WORK	37
	REFERENCES	38
	APPENDICES	
	A. SAMPLE CODE	40

LIST OF FIGURES

FIGURE NAME	FIGURE NO
3.1 ARCHITECTURE DESIGN	15
3.2 DFD LEVEL 0	16
3.3 DFD LEVEL 1	17
3.4 DFD LEVEL 2	17
3.5 USECASE DIAGRAM	19
3.6 CLASS DIAGRAM	20
3.7 SEQUENCE DIAGRAM	20
3.8 ACTIVITY DIAGRAM	21
3.9 STATE DIAGRAM	21
3.10 DEPLOYMENT DIAGRAM	22
3.11 ANACONDA	26
3.12 ANACONDA NAVIGATOR	27
3.13 FLOW DIAGRAM	29
3.14 SYSTEM ARCHITECTURE	30
3.15 BLOCK DIAGRAM	30
4.1 FINDING ACCURACY	34
4.2 DETECTING MALICIOUS URL	35
4.3 WEBPAGE FOR USERS TO CHECK	36
4.4 OUTPUT PREDECTION	36

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO MACHINE LEARNING

Quantum computing is an scientific analysis of algorithmic problems, mathematical models that computerizes systems use to execute the task perfectly without any usage of special commands, but centering on patterns, estimation. Information technology is known as a subset of it. ML algorithms create a statistical model of sample data, referred to as training data in-order to make projections, assumptions without having be specifically trained to do. Machine learning algorithm are involved in applications, like email filtering, network intrusion detection, computerized vision where developing a complex algorithm of instructions is impossible for performing the task. Algorithmic statistics, which focuses on making calculations for machines, is closely linked to computer science. The area of machine learning benefits from the study of mathematical optimization because it provides methodology, theory, and implementation domains. Data mining is a branch of machine learning that relies on unsupervised learning for exploratory data processing. Machine learning is also known as predictive analytics when it is used to solve market issues.

1.2 OVERVIEW OF MACHINE LEARNING

Arthur Samuel invented the name in 1959. "A computer programming is said to be learned from practice E with the other to any classes of functions T , output measure P if it was success at tasks at T , as calculated by the P , increases with the experience E ," according to Tom M. Mitchell, a commonly cited, more systematic description of the algorithms learned in the software development industry. Instead of describing the field in cognitive terms, this definition of machine learning gives a practically operational definition. This is in reference to Alan Turing's suggestions in those papers "Virtualization Machinery and Intellect," which asks, "Do machines think?" replaces "Can devices do what we (as conscious entities) can do?" with "Can machines do anything we (as conscious entities) can do?" The numerous features that may be possessed by a thought machine, as well as the various consequences of building it, are revealed in Turing's plan.

DATASET

A data set (usually known as the dataset) is a cumulative of the information. A data set is often the objects of a single primary key or mathematical data matrix, where each column represents represent a certain variable and each row represents a specific member of the sample group in question. For each component of the datasets the data sets lists of values for each and every of the quantities, such like an asteroid's height, weight. Any attribute is referred to as a datum. The data set may contain data for one sometimes more members, with the number of items respect to the number of members.

The term data set would also be used more generally to refer to the information contained in a set and closely connected tables that correspond to a certain procedure or occurrence. Data corpus also dataset stock are less often used terms for any such data collection. Statistics obtained by space agencies conducting experiments with instruments onboard scientific instruments are an example of this kind. Big data refers to data collections that are so huge that conventional data analysis applications can't handle them.

The data collection is the unit of evaluation in the open data discipline for the knowledge published in a public open data repository. Many as half a million data sets are collected through the European Big Data database. Other meanings have been proposed in this area, but there is currently no official one. Other problems (real-time data streams, non-relational sample sizes, etc.) make it more difficult to reacquaint yourself with the data consensus about it.

URL's

A domain name, or uniform resource locator (url), is the reference to the website resources that describes its primary interface is a graphical as well as a method for retrieving it. A Universal Resource Identifier (URI) is the type of a Uniform Resource Identifiers. And the fact that many people confuse the two concepts.] [a] [a] URLs are widely used to refer to web sites (http), but they can also be used to refer to files and, email (mailto), database access and many other applications.

The URLs of the web pages is normally shown over up in the page in the top address bar by most web browsers. A standard URL could look like this: www.example.com/index.html,

with http as protocol, example.com as a domain name, and the index.html as the url (index.html)

Tim Berners Lee, the inventors of this World Wide Webs and a URI steering committee of the Internet Protocol Suite (IETF) identified Standardized Resource Locators in RFC 1738 in 1994, as a result of consultation that began at the IETF Breathing Papers Birds of a Flock event in 1992.

The style incorporates the domain name scheme (created in year 1985) with the file path code, which uses slashes to distinguish directory and filenames. There were already conventions in place for prefixing server names to full file routes, which were followed by a double slash (/).

Bernie bros longer expresses disappointment for using points to distinguish all the pieces of a search engine with in the URIs, thinking that he would use slash instead, and then also mentioned that the two slashes before the domain name were excessive, considering the colon surrounding the first section of a URI.

"Common" Resource Locators were stated in an early (1993) draught of the Xml Specification. About June 1994, (RFC in1630), October 1994, this has been withdrawn (draft-ietf-url-08.txt).

1.3 INTRODUCTION TO PROJECT

New connectivity tools have had a huge effect on company development and promotion through a wide range of applications, including online banking, u t, and instant messaging. In reality, having an internet presence is almost required to operate a successful business in today's world. As a result, the Massive Global Web's value has been steadily growing. Regrettably, technical advances are accompanied by new advanced tactics for attacking and defrauding people. Rogue web-sites that has capacity to sell the counterfeit a goods, the financial manipulation that fool users in to exposing confidential details that leads to the money or the identification of theft, and even an malware installation in the most users device is examples of such attacks.

Usually, phishing attacks use sql injection to deceive the user into clicking on a spoofed connection that leads to a false web page. The spoof connection is posted on famous websites or sent to the victim by email. The false website is designed to look like the real one. As a result, rather than sending the victim's request to the actual web application, it

will be sent to the fake web server.

1.4 PROBLEM DESCRIPTION

Let's look at the URL layout to get a clearer sense of what attackers are considering as they create a phishing domain. To address web sites, the Universal Resource Locator (URL) was developed. The diagram below illustrates the related parts of a standard URL's structure. It all starts with the protocol that is used to reach the website.

Security threats of malicious URL's

Malicious websites are a widespread and severe cybersecurity problem. Malicious URLs hosts the unsolicited and irrelevant content (malware, spoofing, driven by hacks, and so on) and trick unwitting consumers into becoming scam victims (monetary loss and theft of the privatized information and the malware installation), resulting in billion dollars in losses per years. It was critical for identifying and respond to certain threatens as soon as possible.

Scope

The reach of this strategy is restricted to the host. If the host is untrustworthy but the URL is secure, it will also be labelled as malicious due to the host, resulting in a false positive. However, if the URL is malicious and hosted by a well-known host, it could be misclassified as benign, resulting in a false negative.

Purpose

Cyber protection is jeopardised by a malicious URL. Every year, there are various instances of data loss, monetary loss, and adware and malware installation systems that result in thousands of euros in damages. It is important to browse safely. The aim of the project is to achieve cyber protection by identifying malicious URLs and preventing access to them. To recognise and use the capabilities of already current malicious URLs to identify new malicious URLs, an automated solution for censoring malicious URLs using machine learning is needed.

Goal

The URL is categorised as malicious or benevolent based on its lexical features, host-based characteristics, and popularity attributes.

Design

The first step in lexical features derived is to isolate the two sections of the URL: the host name and the route. Then, since malicious websites have a large number of tokens, we search for them in the web domain and route. In the next page, we examine the duration of the URL, as malicious URLs are usually long, as well as suspicious term tokens. We analyse the credibility and validity of hosts in host-based function review since malicious websites are hosted by less trustworthy and approved hosts. We look at the prevalence of URLs in popularity attribute review since malicious URLs are less common than benevolent ones.

Deviations

The only difference from the plan is that it is an idea. I claimed that URLs should be classified as benign(0), spam(1), or malicious(1) (2). Since spam URLs make up only a small percentage of the dataset as opposed to benevolent and malicious URLs, I transformed them to malicious URLs in the dataset. As a consequence, the only choices for designation are benign(0) or malicious(1) (1).

1.5 PROJECT DEVELOPMENT:

i) Dataset description:

- My information was copied from GitHub by someone who was working on a related python project.
- 832 rows of 22 variables
- Each attribute of a URL is a lexical or host-based function.
- I used correlation in R to search the dataset for size structure, limiting the classification procedure to only the most effective attributes. (code in Python is available on my website)

ii) Design Methodology:

- For this research, I've chosen to use a variety of supervised learning classification algorithms.
- The efficient algorithms for differentiation regression analysis that are currently in use.
- The following analytic activity involves choosing the best attributes needed for the decision tree, since I expect to use the decision tree for classification as well.
- Binaural beats are used to test performance for each method.

CHAPTER 2

LITERATURE SURVEY

1.CANTINA:

Centered on the TF IDF informative retrieve algorithm, Hong et al proposed a content-based methods for finding phish websites. The design and methodology of a few heuristics are also enlightened in this paper. It was created in-order to reduces the count of false cases. The results of this study enhance CANTINA is capable of identifying phishing sites, effectively labelling about 95% of phishing targets.

CANTINA, 1 the novel based on technique for detecting phishing-sites, was implemented at with the preparation, execution, and evaluation. Unlike other methodologies that looks a surface attributes of a site page, such as the URL, domain name, CANTINA looks at the idea of a site to determine if it is genuine or fake. CANTINA make open of the known TF-IDF formula actually, the Robust calculation recently developed by Phelps and Wilensky

for conquering hyperlinks is used in data recovery. The results execute the CANTINA is effective at detecting phishing sites, with a detection rate of 94-97 percent. It was an exhibiting that a CANTINA will be worked out in the collaboration with the heuristic usage by the various devices to inferior fake positivity, whether only slightly lowers the rate of phishing discovery CANTINA is compared to two well-known anti-phishing tools that are represents the best devices for detecting phishing destinations that are currently available. The tests reveal that CANTINA is on par with or better than SpoofGuard in terms of execution, with far less false positives, and performs similarly to NetCraft. The combination of a bar and heuristics are effectivity at the detecting phishing URLs in clients' legitimate email, and the most common blunder is misclassifying spam URLs as phishing.

ISSUES:

CANTINA is a program which detects small-scale versions of all websites.

The larger data websites are not protected by this.

2.CANTINA+:

Cranor et al. proposed CANTINA+, an element rich AI scheme that aims to use AI to exploit the expressiveness of a rich array of highlights in order to gain the high Accurate Positive rate (AP), on novels phishes while limits the False Positive rate to the lowest level using sifting calculations. CANTINA+, most used element bases approach in the writing, which elaborates the HTML Documentation Objective Model (DOM), web engine tools, outsider administrations using AI technique for identification of phishes, includes eight novel highlights. They devised two channels to aid in the reduction of FP. The firstly are a close-copy phish locator that hashing to generate phish that is extremely similar. The second is a login framework channel, which categorizes Web pages that have no known login structure as genuine. At last, CANTINA+ has been demonstrated for being a serious phishing adversary.

ISSUES:

The contents are downloaded from web pages and depend solely on the Google search engine.

The system's forecast is entirely dependent on the results of a search engine query.

3. The semi-directed learning approaches for the location of hacked site page:

Another phishing site page discovery proposed by the Zhao dependent on sort of a semi-regulated learns technique trans-reductive helping machinery.

The highlight for the web's picture is extricated a supplementing the impediment of phishing recognition just dependent on archive objective model (DOM) incorporates dark histograms, shading histograms, spatial connection between the sub diagrams. The most highlights that delicate data can be analyzed utilizing pages examination dependent on objects. As opposed to the downside for helping vectorized machine calculation whether basically prepares classifies by then learning pretty much nothing an helpless delegate marked examples, this technique acquaints the TSVM with train classifier that it considers the dissemination data certainly encapsulated in the huge amount of the unlabeled examples and have preferred execution over SVM.

ISSUES:

The recognition pace of this technique is somewhat lower.

It depends just on Google internet searcher and the substance that can be downloaded from those pages.

4. A Multitier phishing identification:

This was mainly proposed by Islam et al. another methodology called as the multi-level grouping modeling for the phishing emails separating. It was a creative strategy to the extricating most highlights of the phishing of emailing dependent in numbers of message substance through messages headings that then selected highlights will be indicated by the needed ranking system and thoroughly examined

and then the impact of re-scheduling the classification algorithms in a different tier of classifications in the process to finding out the most rarely optimized scheduling algorithm. The correct and the exact proof is that, this methodology diminishes the bogus most positive issues considerably with decreased intricacy.

ISSUES:

It is the extended test for build up the vigorous malware identification technique holding precision for future phishing messages.

Highlight recovery is wasteful.

5.Assessing the severity:

Guo et describe the frames work to determining the magnitude for the phish attack on common terms of highly risky levels and probability of possible market value losses and profit incurred by the targeted businesses. The administered arrangement procedures, a significance of data mining, are used to determine the seriousness of the highest phishing attacks. Asynchronously, the important factors that contributes to the highly risk level or the incredible financial loss as a result of a phishing attack are into the light. Guo et al. used the hybrid approach that fused key expression and the extraction, supervised characterization techniques with the literary awareness depiction of the phishing attack and the tax information of the target system. A firm that determines the magnitude of a phishing attack based on the level of risk or the potential for financial loss.

ISSUES:

This strategy was only supported if the cost of misclassification was comparable.

The tests cannot be carried out if the cost of misclassification is unequal.

The consequences of mis-classifying concept involved in the high-risk or high-CAR phishing attack are most to be devastating.

6. Delicate registering based attribution:

Nishanth et al. utilized the novel with two stage delicate registering approaches for information attribution to survey the seriousness of the phishing assaults. The ascription strategy includes K-implies calculation, multilayer perceptron (MLP) working couple. The half breed is applicable for supplant missing of the estimations of monetary information which then utilized for the anticipating the seriousness of the phishing assaults in monetary firms. In the wake of crediting the missing qualities, mine the monetary information identified with the organizations alongside the organized type of the printed information utilizing Multi Layers Perceptron (MLP) and Probabilistic Neural organization (PNO), Decision Tree (DT) individually.

To start with, supplant the missing qualities in the monetary information utilizing the delicate processing-based information attribution approach. At that point, applying the text mining on text based (unstructured) information of phishing cautions. Subsequently, text-based information is changed over into organized information. At last, anticipate the danger level of phishing assaults utilizing the consolidated monetary information from the budget report of the organizations and text-based information utilizing MLP and DT independently. The general of grouping exactness of the three danger classifications of phishing assaults utilizing the classifiers like MLP, PNO, and DT are predominant.

ISSUES:

Monetary information alone can be done by the general precision utilizing PNN isn't the awesome.

Accuracy will not be constant with the most different levels.

7.Visual-similitude:

Kruegel demonstrated an efficient method for detecting by analogizing the visual similarity among the one of the suspected phishing site and unique spoofed legitimate site, phishing attempts can be identified. A phishing attempt is made when the two pages are "as well" identical. alert is shown. They use three features to assess page similarity in this system: The page's overall visual appearance as seen by the client, including texting pieces (this includes the style bases features), image inside and in the

page, and page's over virtual appearances as seen by the client (after the program has delivered it). They used these highlights to assess the similarity between the goal and the authentic page, calculating a single similitude ranking.

ISSUES:

DOM Anti Phish was not much effective again the phish sites that they depend on pictures for the most part.

These phishing endeavors were not identified altogether sites.

8.Detecting phishing website pages with visual similitude:

proposed a viable methodology for recognizing phishing pages, that then utilizes Earth Movers Distances (EMD) that ascertain a nearest visually closeness of web page. which utilizes Earth Movers to calculate the comparability of webpage, use the EMD method. The key reason why net users can become the phishing victims that is phishing webpages have a strong visual resemblance to legitimate Web pages, for example, outwardly comparative square designs, predominant tones, pictures, and text styles, and so forth They use the counter-phishing technique to gain the suspicious webpages, that should be gained from the URLs in messages that contains watch words linked with the protected web pages. First believer them into standardized pictures and afterward address their picture marks with highlights made out of predominant shading classification and its comparing centroid facilitate to figure the visual similitude of two Web pages.

ISSUES:

Initial adherent them into normalized pictures and a while later location their image marks with features made out of dominating concealing order and its contrasting centroid encourage with figure the visual likeness.

9.Fighting phishing:

Phishing is type of the online data fraud related with the social designing and

specialized ploy. In particular, phishers endeavor to fool Internet clients into uncovering delicate or private data, for example, their ledger and Visa numbers. Chen et al. introduced a viable picture put together enemy of phishing plan based with respect to discriminative central issue highlights in site pages. Then they utilize an dis variant substance descriptor, Contrasting Contexts Histogram (CCH), for figuring out the likeness level between dubious pages, valid.

ISSUES:

The method is vulnerable to change the webpage ratio and color palette.

10. Phishing discovery:

Using the AC method in data mining to solve the problem of phishing venue. They compare and contrast MCAC, a newly developed AC calculation, with the other AC, rule enlistment calculations with phishing data. The information on phishing was gathered from Phishing tank document, that one is the less community site. The actual pages, on the other hand, were gathered from the Hurray Registry. Thabtah et al. demonstrated that MCAC can extract rules that resolve relationships between site highlights. These guidelines are then used to determine the site's type.

ISSUES:

Rather than using an intelligent data mining method, the rule is based on human experience.

To maximize the number of features collected, this method did not consider content-based features.

CHAPTER 3

METHODOLOGY

3.1 GENERAL

The aim of the Machine Analysis is to create a brief analysis task as well as to gather complete details about the definition, actions, and other constraints like performance

measurement and machine optimization. The aim of System Analysis is to thoroughly define the technical specifics of the main definition in a simple and succinct manner.

3.2 EXISTING SYSTEM

Malicious identification systems based on outlier-detection strategies for a CR cooperative sensing device are currently in use. Both sensors transmit their energy transmitter outputs to an access point, which uses a big data and understanding to determine the presence of a primary signal. Throughout each sensing iteration, we explored different rigorous methods for assigning outlier variables to the apps. These outlier variables are used by malicious user identification systems to identify malicious users and reduce their effect on the sensor system's performance.

Disadvantages:

1. This model is based on observational evidence, and requires a large amount of computational work.
2. Sensing data rapidly becomes redundant due to problems such as channel loss and receiver instability, rendering future decision-making impossible.
3. The channel has a lot of interference.

3.3 PROPOSED SYSTEM

The machine is given a dataset of malicious and valid URLs, which is then pre-processed so that the data can be used for analysis. Around 30 attributes of malicious websites are included in the features, which are used to separate them from legitimate ones. Each type has its own set of malicious qualities and standards that must be adhered to. For each URL, the defined characteristics are extracted, and appropriate input ranges are defined. Each malicious website danger is then given one of these values. The ranges for each input vary from 0 to 10, while the output ranges from 0 to 1. The values of malicious attributes are represented by binary numbers 0 and 1, showing whether the attribute is present or not.

After the data has been conditioned, we can make the usage of a machine learn algorithm to analyze the dataset. The algorithms for machine learning have already been discussed in a proper section. Following that, we can use a hybrid classification method in which we

combine two classifiers, SVM and Logistic regression, to estimate the precision of the phishing URL detection, resulting in the desired outcome. This technique, also known as a hybrid approach to data testing, proposes the use of a combination of two classifiers, as discussed above. The data will then be checked, and the forecast accuracy will be evaluated, which will be higher than the current system. Now we'll look at the various classifiers and talk about the hybrid mix we used with our proposed scheme.

Advantage

1. accuracy level is very high we can predict level is increased.

Support Vector Machine:

Support vector machines (also called as support vector networks) are the supervised learn model that process data for classifying or regression analyzing and come with related learning algorithms. SVMs that performs the non-linear classify as well as linear classification by using a technique known as the kernel trick, which involves indirectly mapping inputs into high-dimensional feature space.

Logistic regression:

We suggested logistic regression as a solution. The Sigmoid Function is the type of the functions that is most used to describe (Logistic Function). The linear equation with the independent predicts the often uses in the logistic regressive algorithms to maintain the value. The estimated value ranges from negative to positive infinity. A algorithm's output must be the class vector, no, yes.

EXPERIMENTAL METHODS

3.4 HARDWARE REQUIREMENTS

Physical computing tools, also known as hardware, are the most basic set of specifications specified by the operating system and the software application. In these cases of operating systems, the hardware specification list is the often followed by the hardware configuration process. The below are the minimum hardware requirements:

- 1.PROCESSOR: PENTIUM IV
- 2.RAM: 8 GB

- 3.PROCESSOR: 2.4 GHZ
- 4.MAIN MEMORY: 8GB RAM
- 5.PROCESSING SPEED: 600 MHZ
- 6.HARD DISK DRIVE: 1TB
- 7.KEYBOARD:104 KEYS

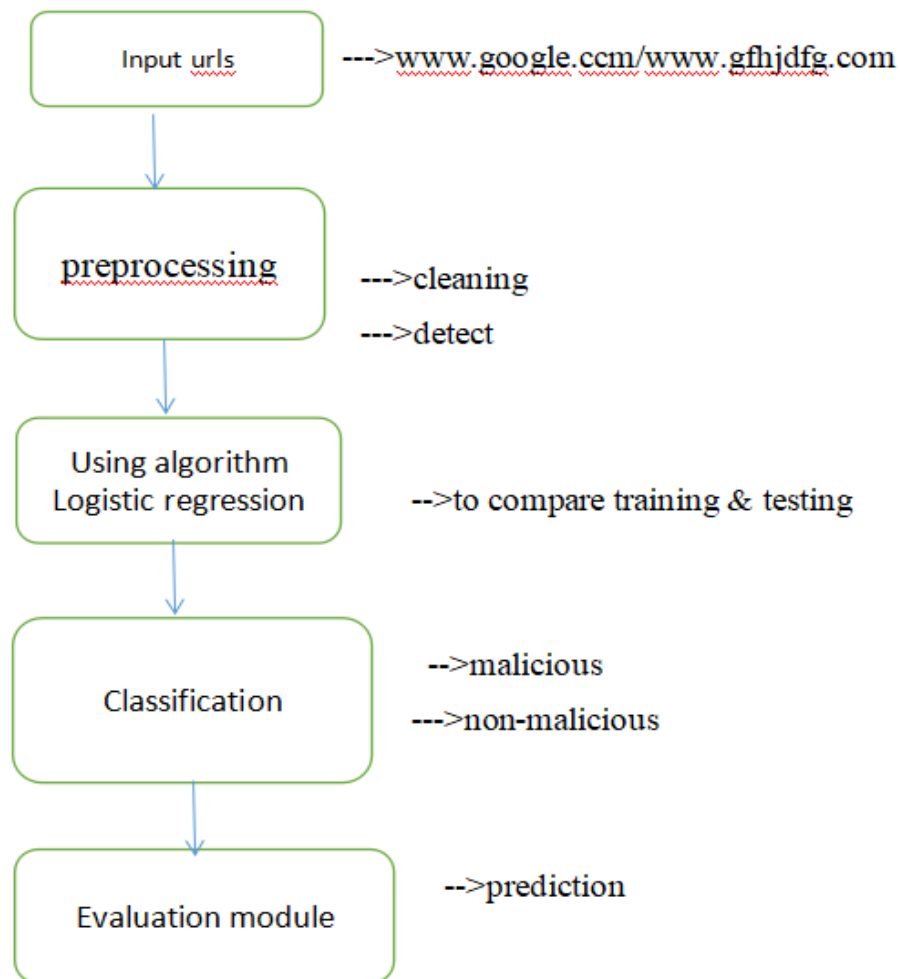
3.5 SOFTWARE REQUIREMENTS:

Computer specifications are concerned with specifying the tools and prerequisites that must be implemented on a device in order for an application to work. These prerequisites must be installed before the app can be installed. The below are the minimum programme requirements:

- 1. FRONT END: PYTHON
- 2. DATASET: CSV
- 3. IDE: ANACONDA
- 4. OPERATING SYSTEM: WINDOWS 10

3.6 ARCHITECTURE DIAGRAM

A system architecture, also known as systems design, is the mathematical models that says describes the systems configuration, behavior, and some aspects. The systematic meaning and represents of a system arranged in the manner that needs for the facilitates that thinking about the systematic mechanisms is otherwise called as the architecture description. Device elements, publicly observable properties of certain components, and interactions between them can all be used in the system architecture. It will offer a blueprint for obtaining goods and developing processes that can work together to execute the overall



structure.

Figure 3.1: Architecture diagram

3.7 SYSTEM DESIGN

The method of translating specifications into a representation of software is known as system design. A specification is an engineering image of something that would be constructed. Design provides us with software representations that can be evaluated for consistency. If product production as a whole "blended the perspectives of advertising, architecture, and production in to the single entity," single approaches to the products development," then the design is most act of taking up the advertising information and creates the design for the product that to be manufactured. Most commonly used approaches for computer system architecture are object oriented analyzes and design methods. In objective oriented research or architecture, the UML has become the de facto standard. It's commonly used for modelling computing systems, and it's becoming more

popular for non-software systems and organizations. As a result, systems architecture is the practice of defining and implementing systems to meet the user's specific specifications.

3.8 DATA FLOW DIAGRAM:

1. The bubble map is another name for the DFD. It is the most basic graphic formalism that is being used to describe the device in common terms of the set it collects, the process and it performs on the data, and the data does produce the output.
2. One of those is the most important modelling techniques is data flow diagrams (DFD). It's made to represent the various component of each and every machine. The control itself, the data that process uses, the external attribute that communicates with those system, and then knowledge flow in the system are all examples of these elements.
3. Illustrates how the data flows through a system and transformed by the sequence of the transformations. It is the regular schematic representation of data flow and mostly the transformation that occur when data traverse from the input to result.
4. DFD is oftenly referred to the bubble map. At any higher level of the abstraction, a DFD can effectively be used to describe a system. DFD can be categorized into categories, each reflecting a particular level of knowledge flow and functional detail.

DFD Level 0

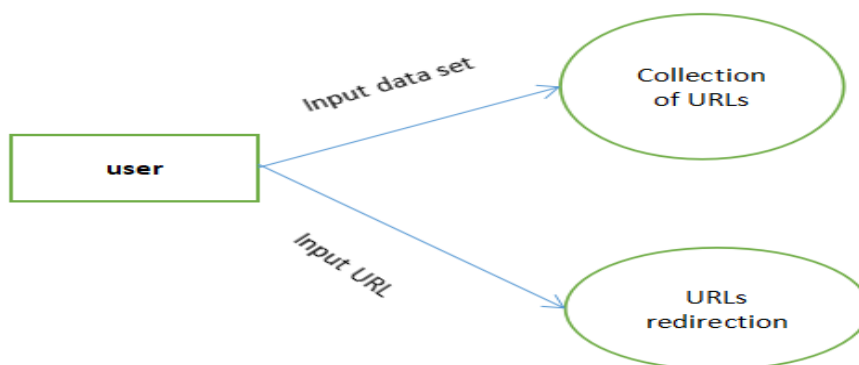


Figure 3.2: DFD Level 0

DFD LEVEL 1

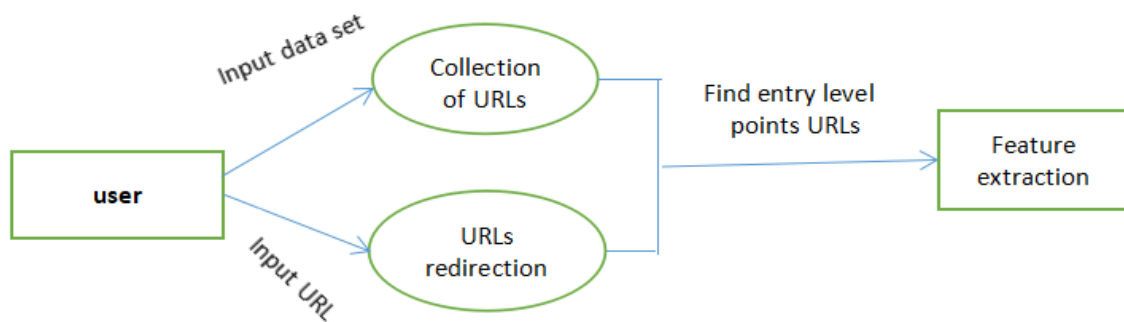


Figure 3.3: DFD Level 1

DFD LEVEL 2

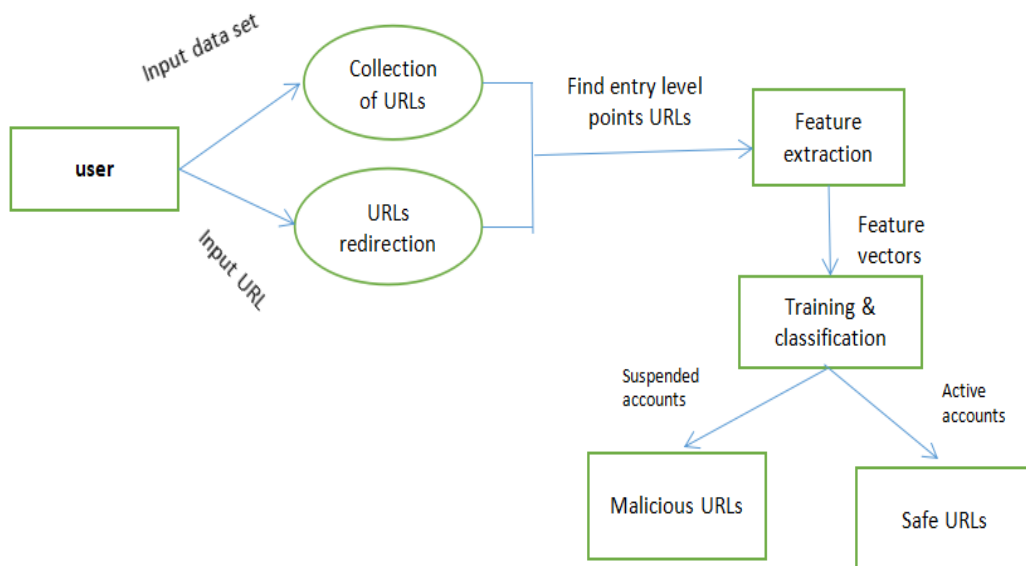


Figure 3.4: DFD Level 2

3.9 UML DIAGRAMS

Unified Modeling Language (UML) is an acronym for "Unified Modeling Language."

UML is a standardized general-purpose modelling language used in the field of object-oriented software creation. The Object Management Community in charge of the standard, and it was developed by them.

The ultimate aim is for the UML that becomes a standard of modelling objective-oriented computer application. UML has two major components into present form: meta model and notation. Any kind of system and procedure can be applied to, connected with the UML in our future.

The Unified Modeling (UML) is the basic language to describing, visualizing, constructing, reporting software systematic objects, as well the business modelling, other non-software structure.

The UML is collection of validated design principles for modelling large and complex structures. The UML is a critical components of the object oriented software architecture and the software creation process. To express the architecture of software projects, the UML primarily employs graphical notes.

GOALS:

The first goals in designing of the UML are follows:

1. User should be able to create and share meaningful templates using a ready to use visual modelling language.
2. To expand the key principles, include frameworks for extendibility and specialization.
3. Be unconstrained by programming languages or implementation processes.
4. With a ready-to-use, expressive digital modelling language, users should be able to build and exchange meaningful models.
5. Build a systematic foundation for comprehending the modelling language.
6. Encourage the demand for OO tools to extend.
7. Higher-level programming principles like alliances, models, trends, and modules should be supported.
8. Good practices should be included.

USE CASE DIAGRAM:

A usecases diagram is the type of behavior diagrams described by generated from the Use-case studies in the Unified Modeling Language (UML). It is the aim is to provide a graphical representation of a system function in common terms of actor, priorities (represents as the use cases), any dependency between these use cases. A usage cases diagram key aim is to demonstrate what framework is being used will performed for either actor. Roles of these actors in those systems are being depicted.

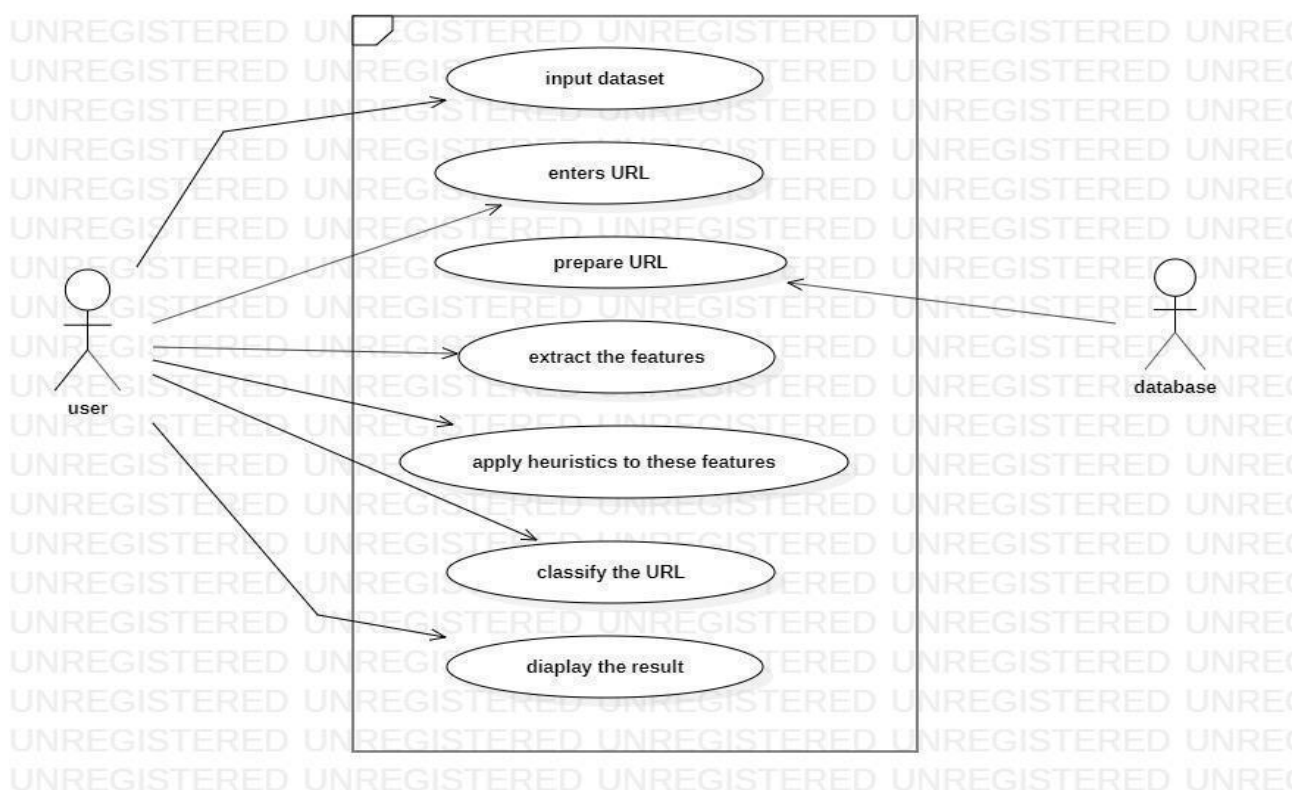


Figure 3.5: Use case Diagram

CLASS DIAGRAM:

A class diagram would be used to describe, characterize, and record numerous elements of a system, and perhaps to create executable code for a computer system.

The attributes and procedures of a class, and perhaps even the system's limitations, are portrayed in a class diagram. Class measures are widely used in the modelling of attribute structures since they are the only UML diagrams that can be explicitly mapped for

object-oriented languages.

A class diagram portrays a set of classes, interfaces, alliances, collaborations, and constraints.

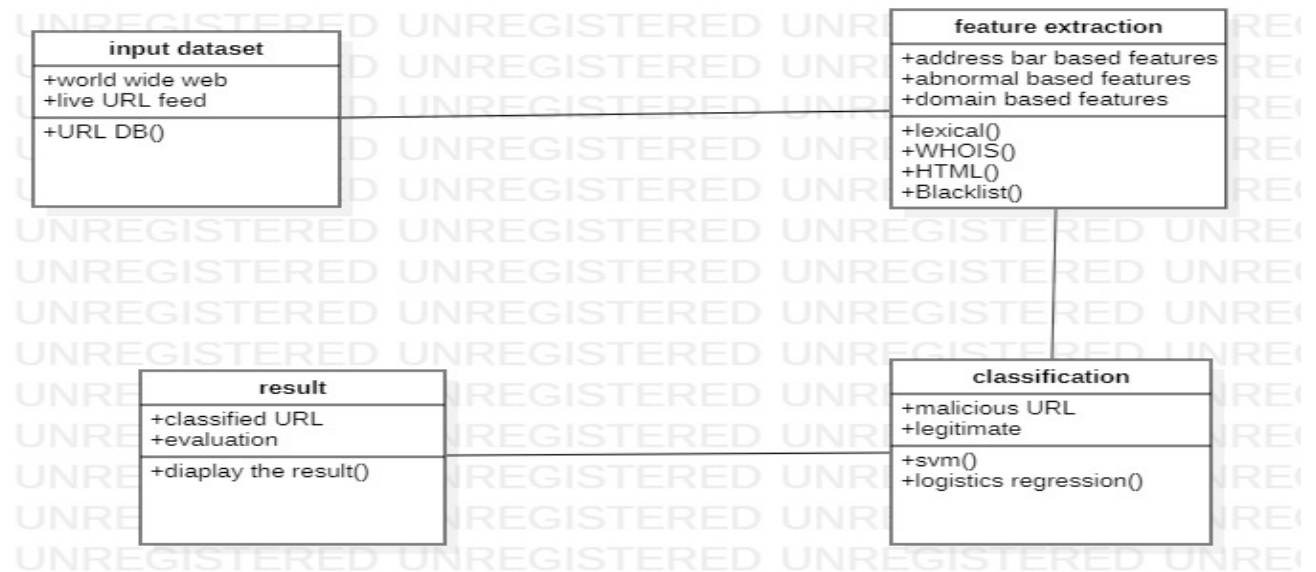


Figure 3.6: Class Diagram

SEQUENCE DIAGRAM:

In the Unified Modeling (UML), a sequence is the type that the activity diagrams that depicts however processes communicate one another and in which else order. It's a Line Chart Map construct. Case diagrams, scatter graph, and timing graphs are all terms used to describe sequence diagrams.

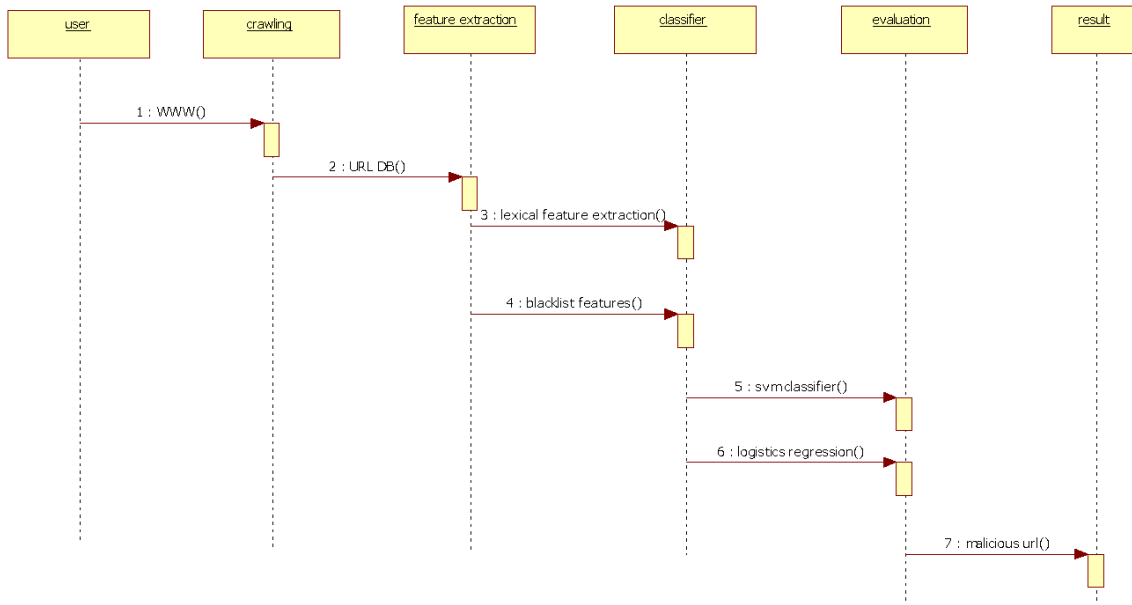


Figure 3.7: Sequence Diagram

Activity diagram

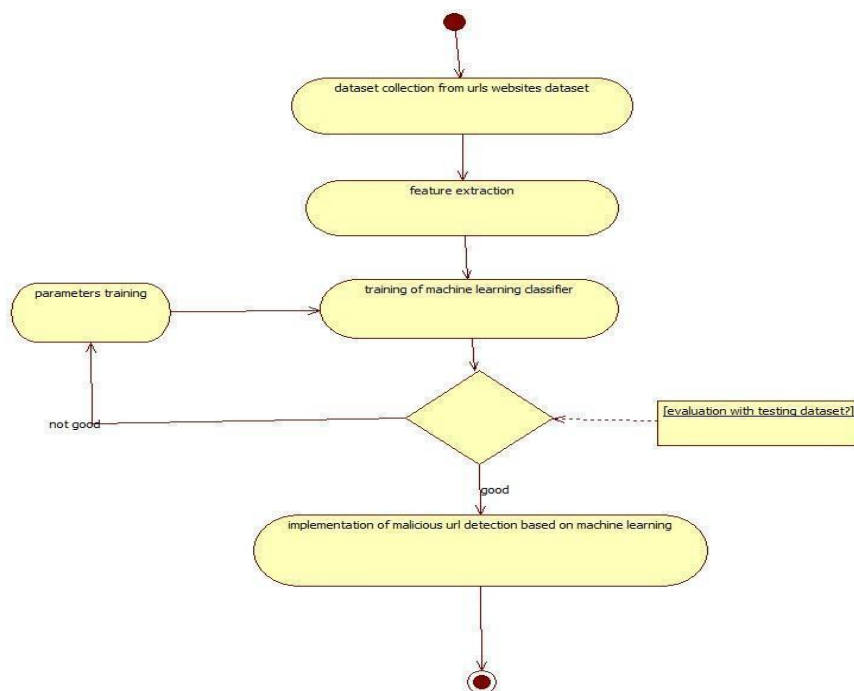


Figure 3.8: Activity Diagram

State diagram

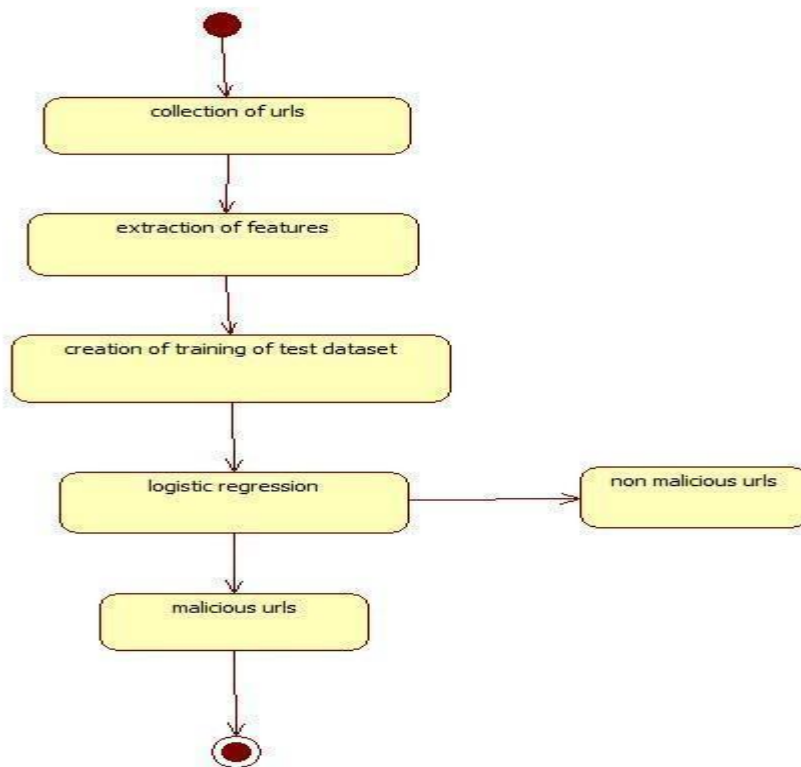


Figure 3.9: State Diagram

DEPLOYMENT DIAGRAM

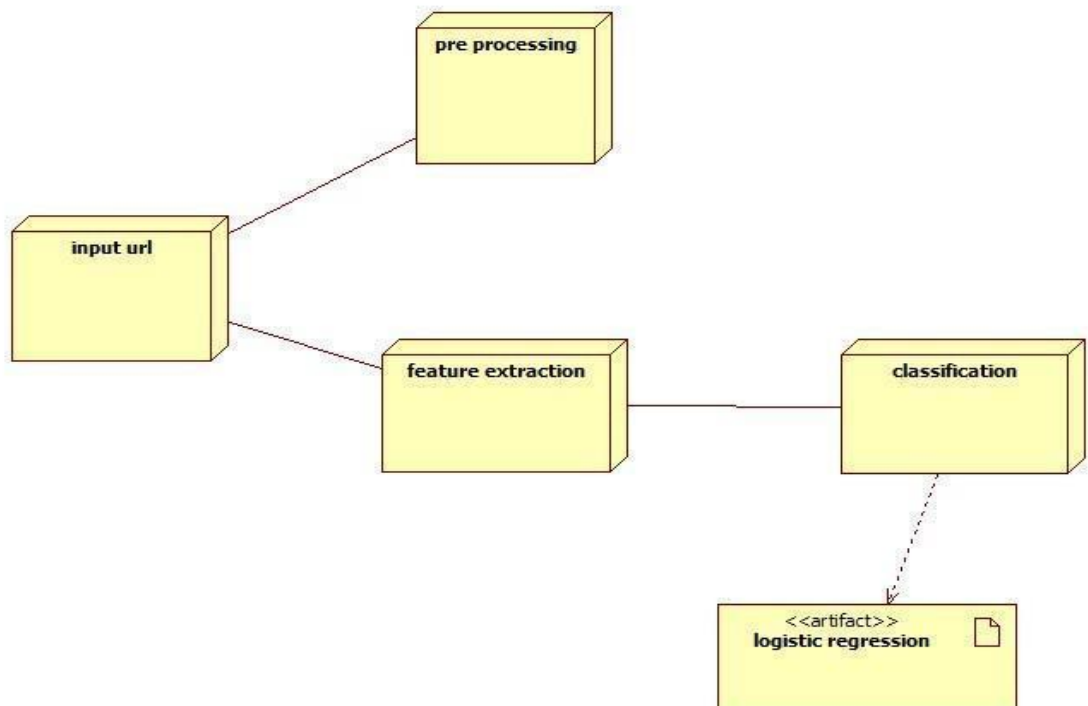


Figure 3.10: Deployment Diagram

3.10 SYSTEM IMPLEMENTATION

Software is categorized into modules, which are individually labelled and addressable elements that are combined to fulfil issue specifications. Modularity is the single feature of software that enables a program to be managed mentally. A module is a program's component. Programs are made up of one or more separately created modules that aren't joined together until the end. A single module can contain one or several routines.

MODULES:

- Data Pre-processing
- Feature Extraction
- Model Creation
- Prediction

DATA PRE-PROCESSING

We load the metadata into this pre-processing step, then apply the metadata to the data and replace the transformed data with metadata. The data would then be carried on, with the unnecessary data in the list being removed and the data being divided into train and test data. To break the data into train and test, we'll need to import train test split from scikit-learn. This will assist the pre-processed data in splitting the data into train and test based into the weights defined on the platform code. The test and train are divided by 0.2 and 0.8, or 20 and 80 percent, respectively.

FEATURE EXTRACTION

Feature extraction is a dimensionality reduction method that reduces a large collection of

raw data into smaller categories for processing. The vast number of variables in these large data sets necessitates a lot of computational power to process. Feature extraction refers to methods for selecting and/or combining variables into features in order to reduce the quantity of the data that needs to be processed while still correctly and fully describing the existing environment.

MODEL CREATION

We create data into two models:

- Training model
- Testing model

The test and train are divided by 0.2 and 0.8, which equals 20% and 80%, respectively.

For the training portion, we use a machine learning algorithm to assess the model's accuracy. The svm and logistic regression algorithms are used.

PREDICTION

This module is built on the user interface. Bootstrap is used to build a web page. Enter the URL for the web page. We now obtain data from the customer in order to compare the dataset values. Finally, it can determine whether the user is malicious or not.

3.11 LANGUAGE SPECIFICATION:

PYTHON LANGUAGE

Guido Rossum developed Python in 1989 as an object-oriented programming language. It's perfect for quick prototyping of complicated software. It can be extended to C or C++ and has interfaces to several OS device calls and libraries. NASA, Google, YouTube, BitTorrent, and other major organisations use the Python programming language. Python programming is commonly used in specialised areas of computer science such as Artificial Intelligence, Natural Language Generation, Neural Networks, and others. Python puts a heavy emphasis on code readability, and this class will teach you the fundamentals of the

language.

PYTHON PROGRAMMING CHARACTERISTICS

It has a larger number of data types and a simpler syntax than any other programming language.

It's a scripting language that works on every platform and has direct access to operating system APIs.

It has more run-time stability than other programming languages.

It contains Perl and Awk's simple text manipulation features.

In Python, a module can contain one or more classes and free functions.

Python libraries are cross-platform, meaning they work on Linux, Macintosh, and Windows.

Python can be converted to bytecode for use in massive applications.

Python embraces both functional and formal programming, as well as object-oriented programming (OOP).

It has an immersive feature that helps you to communicate with it.

Software fragments are tested and debugged.

Since there is no compilation phase in Python, writing, debugging, and checking are all possible.

APPLICATIONS OF PYTHON PROGRAMMING

Web Applications

Frameworks and CMS (Content Management Systems) based on Python can be used to build scalable Mobile Applications. Django, Plone, Pyramid, and Django CMS is some other of most common tools to designing Apps. Python used to control websites such as Mozilla, Instagram, PBS, Reddit.

Scientific, Numeric Computing

Python has a plethora of libraries for the scientific and numerical programming. In general purpose programming, libraries such as SciPy and NumPy are used. There are also specialised libraries, such as EarthPy for earth science and AstroPy for astronomy, among others. Computer learning, data mining, and deep learning both use the term extensively.

Creating software Prototypes

When compared to compiled languages like C++ and Java, Python is sluggish. If resources are scarce and performance is needed, it can not be the best solution. Python, on the other hand, is an excellent language for prototyping. Consider the following scenario: You can start by making a demo for your game using Pygame (a game creation library). If you like the demo, you can make the game using a language like C++.

Good Language to Teach Programming

Many businesses use Python to teach programming to children and beginners. It's a good language with a lot of skills and functionality. Despite this, it is one of the simplest languages to learn due to its easy syntax.

About Opencv Package

Python was a high program language created by name Guido Rossum that has quickly gained popularity due to its ease and readability of code. It allows the program for communicate the thoughts in lesser lines of coding while maintaining readable.

Python is lesser than other language like C, Cpp. Python could also been conveniently expanded with C/C++, which is an important feature. This function allows used to written computationally intensity C, C++ code and wrap it in a Python wrapper, which we can then use as modules. This gives us with twice benefits: our code is the quick as original C, Cpp code (due to the real Cpp code running at the background), and secondly, Python is much simple to coding. OpenCV-Python is a Python wrapper over the original C++ implementation.

Numpy's assistance makes the job much simpler. Numpy is a numerical operations library that is highly structured. It has a syntax similar to that of MATLAB. Both array structures in OpenCV are translated to and from Numpy arrays. the number of guns you have in your

arsenal Aside from that, there are many other libraries, such as Sc. So, what-ever Numpy operation can you do, can combine them with CV, which makes iPy more efficient Matplotlib which support the Numpy can be calculated with these.

3.12 FEATURES OF ANACONDA NAVIGATOR

ANACONDA

Anaconda is the free and open-source Python and R programming language distribution that is simple to set up. Anaconda is a software environment for mathematical computation, computer science, predictive analysis, and deep learning.

Anaconda 5.3 is the most recent distribution, which was launched in October of 2019. It contains the module, an environmental manager, and the library at over 1000 open-source packagers, all of which come with free community support.

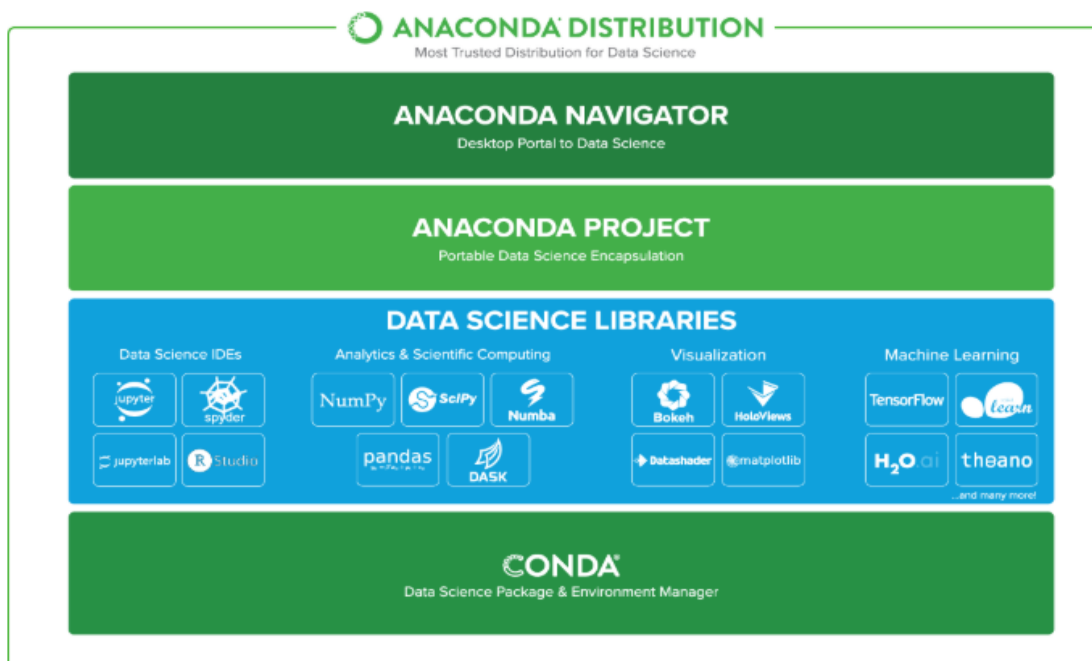
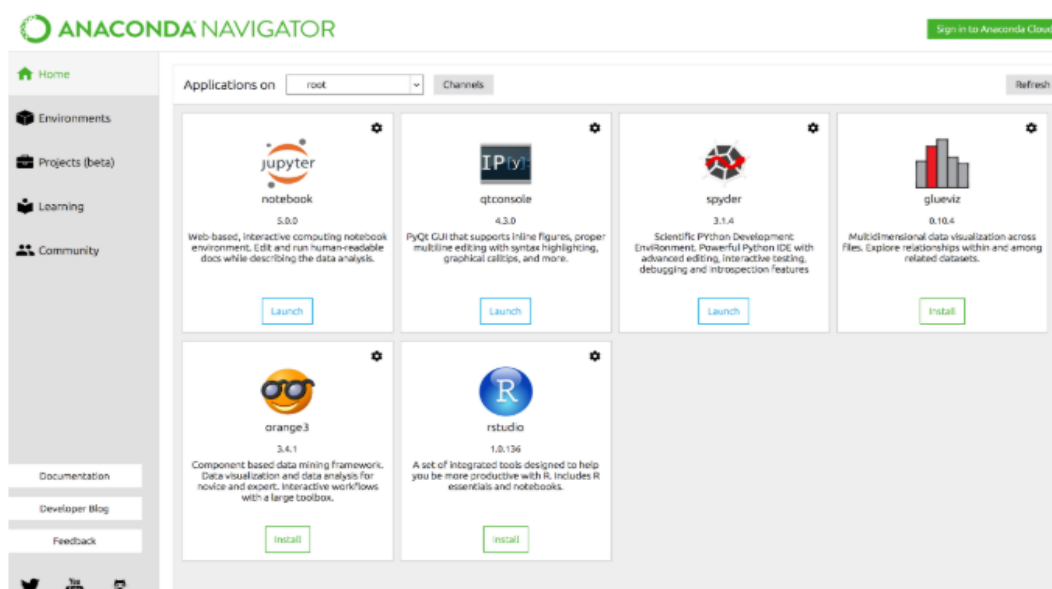


Figure 3.11: Anaconda

What is Anaconda Navigator?

Anaconda navigator is the graphics users interface (GUI) for desktop that comes with the Anaconda distribution. It helps us to use the Anaconda distribution's software and control conda packagers, environmental, and networks withheld having to use the command line command. It is most compatible for the system, Mac OS X, Linux.



Anaconda Navigator

Figure 12: Anaconda Navigator

Applications Provided in Anaconda Distribution

The Anaconda distribution with the following application with the usage of the Anaconda Navigator.

1. Jupyter Notebook
2. JupyterLab
3. Qt Console
4. Glueviz
5. Spyder
6. RStudio
7. Orange3
8. Visual Studio Code

- **JupyterLab:** This is based on Jupyter Notebook, Structure, this is an expandable worker platform of the collaborative, reproducible computing.
- **Jupyter Notebook:** This is an immersive programming notebook that runs on the internet. When explaining the data processing, we can be able to edit and run human readable docs.
- **Qt Console:** In line figure are, clear multilevel editors with the syntax highlighting, graphics call tip, more are all supported by the PyQt GUI.
- **Spyder:** This is the Python Programming Environment for scientists. It is the robust Python with functionality such as advancement editing, dynamic checking, debugging concept, introspection.
- **VS Code:** This is a streamline coded editor that includes features for debugging process, mission execution, version control.
- **Glueviz:** This was mostly used to visualize multi-dimensional dataset spanning several directories. The searches for links between and beyond similar data set.
- **Orange 3:** It is the data mining platform built on components. This will be used for data processing and visualization. Orange 3's workflows are much collaborative, having the largest toolbox.
- **Rstudio:** This is a set of the resource that work together to help you get things done with R. It comprises R basics as well as notebooks.
-

New Features of Anaconda 5.3



Compiled with Latest Python release: Anaconda 5.3 is compiled with Python 3.7, taking advantage of Python's speed and feature improvements.

- **Better Reliability:** The stability has been enhanced with a new update, which now collects and saves packages meta data for the installed packages.
- **Enhanced CPU Performance:** In Anaconda 5.3, the Intellic Math Kernel Libraries 2019 for Deepest Neural Networks (MKL 2019) has been included. MKL 2019 for the Deepest Neural Networks will be used by Tensor Flow apps. These Python binary packages are offered to help you get the most out of your CPU.
- **New packages are added:** The latest version contains over 230 packages that have been revised or added.
- **Work in Progress:** With Python 3.7, there is a casting flaw in Numpy, but the entire team is now patching it until Numpy is revised.

3.13 FLOW DIAGRAM:

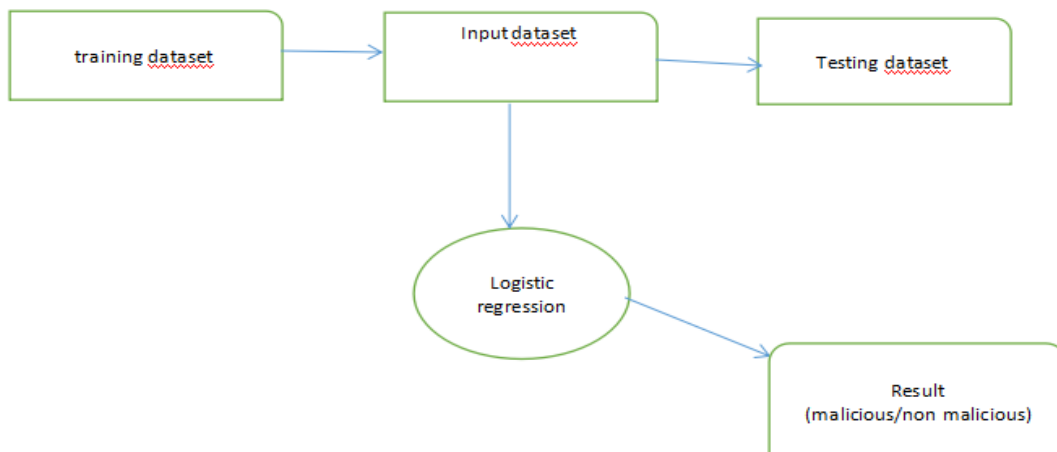


Figure 3.13: Flow diagram

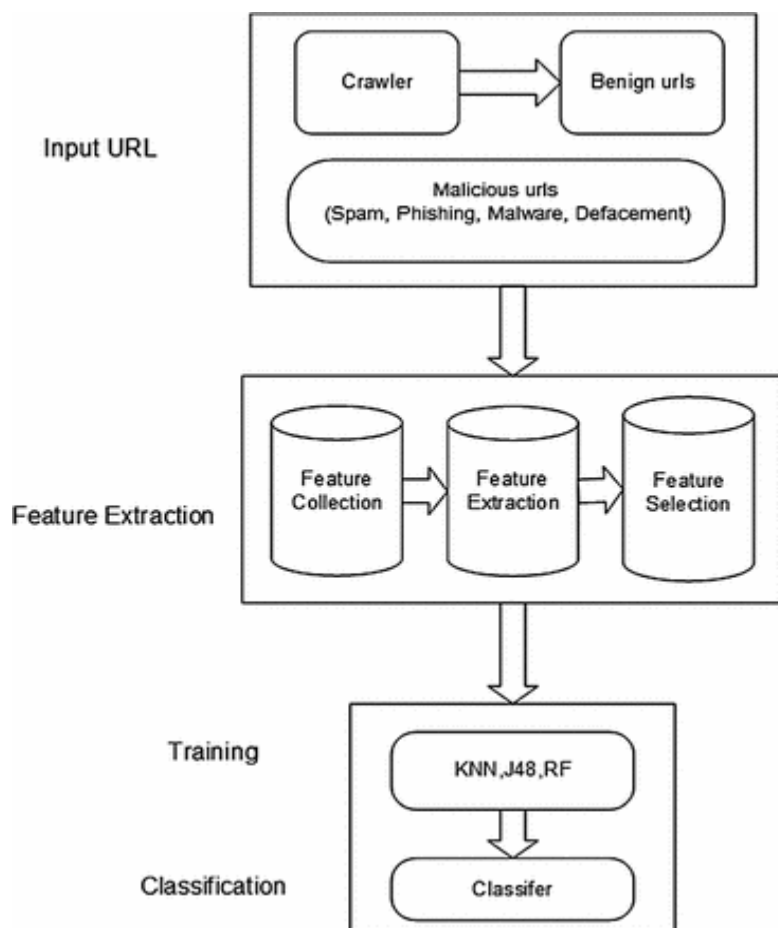


Figure 3.14: System Architecture

3.14 BLOCK DIAGRAM

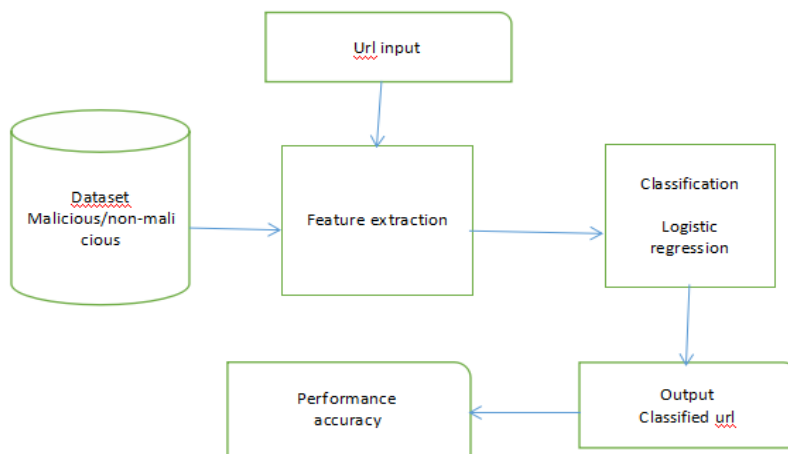


Figure 3.15: Block Diagram

3.15 SYSTEM TESTING

The aim of research is to found mistakes. Testing was the methodology of attempting to finding any possibility flaws and weaknesses in the work object. This allows you for testing the functional of individual parts, sub assembly, assemblies, and a completed project. It was the method for evaluating software to that ensure the it satisfies its specification and meets consumer needing, that it does not mal functionate in any inappropriate way. There is also difference kinds of tests. Each and every test form is designed to meet a particular research need.

3.16 TYPES OF TESTS

UNIT TESTING

Unit checking requires generating test cases to ensure that the software's internal logic is running correctly and that program inputs result in correct outputs. Validation can be performed on both judgement branches and internal code traffic. That is the testing of the application's individual device modules. It is performed after an independent unit has been completed and before it is integrated. This is an intrusive structural examination that

depends on prior understanding of the structure. Unit checks are used to analyze a complex business method, program, or device configuration at the component stage. Unit checks ensures that those each particular direction of the business processes meets the documentation requirements and has specifically specified input, outputs.

INTEGRATION TESTING

This check is used for seeing how two or more software modules will function together as a single application. Research is event-driven, with an emphasis on the specific result of screen and field. Integration checks prove that, while the components is individually satisfying, the arranging of component is rights and high reliable, as shown in the good unit testing. This testing is a form of testing that focuses on exposing issues that result from then the combination of all the components.

FUNCTIONAL TESTING

Functionality checks demonstrate that the features being evaluated are accessible in accordance with the market and operational specifications, device documents, user manuals.

Functional testing was centered in the following terms:

Valid Input: The agreed validating input groups must should be defined.

Invalid Input: The class of invalid input must should be detected and rejects.

Functions: the functionality that have been defined must be included.

Application outputs must be exercised in the specified groups.

Interfacing mechanisms and protocols must be invoked.

Functional assessments are organized and prepared around criteria, main features, or unique test cases. Furthermore, comprehensive coverage of for testing, consider business process flows, data areas, predefined procedures, and subsequent processes. Additional tests are described and the beneficial value of current tests is evaluated before functional testing is completed.

3.17 SYSTEM TESTING

Device verification guarantees that the whole implemented development system complies with the specifications. It checks a configuration to ensure that the outcomes are

known and predictable. System testing is an example of memory allocator system integration testing. System testing incorporates method parameters and flows, with an emphasis on pre-driven process connections and integration points.

WHITE BOX TESTING

This is the method of software tester in which the software testing is familiar with a software's inside workings, configuration, languages, or at very least it is purpose. This serves a reason. It was used to testing places those aren't available with a black box standard.

BLACK BOX TESTING

Testing applications without understanding the inside workings, function, and vocabulary of the most module being evaluated is known as black box testing. For most other kinds of tests, black box tests require a definitive source document, such as a specification or a set of standards. It is a form of testing in which the software under test is treated as a black box in a particular manner. It is hard to “see” through it. The test responsible for understanding and responds to outputs without considering how the program performs.

UNIT TESTING

This testing is typically performed as the part of a combination programming and then the unit test process of these software development lifecycles, although it was not unusual for code and unit testing for being done independently.

Test strategies and approaches

Field testing can be performed by manually, functional tests would be written in neat and clear detail.

Test objectives

Both field entry must behave correctly.

The identified relation must be used to trigger the sites.

There must be no delays in the entry screen, calls, or replies.

Features to be tested

Check that the submissions are in the proper format.

There should be no duplicate entries allowed.

All links should lead to the correct page for the user.

Integrity Checking

The gradual integrity tester of two and more connected softwares module in an unified system for producing errors triggered by interfacing fault is known as software system development.

The integrity tests aim is guaranteed the modules and software systems, such as those used in this software framework or – a step up software at this corporate level, function together flawlessly.

Test Results:

Every one of the above-mentioned test case were successful. There were no flaws found.

Acceptance Testing

Acceptance by the users Testing is an important aspect of every project, and it necessitates active input from the end user. It also guarantees that the device satisfies the operating specifications.

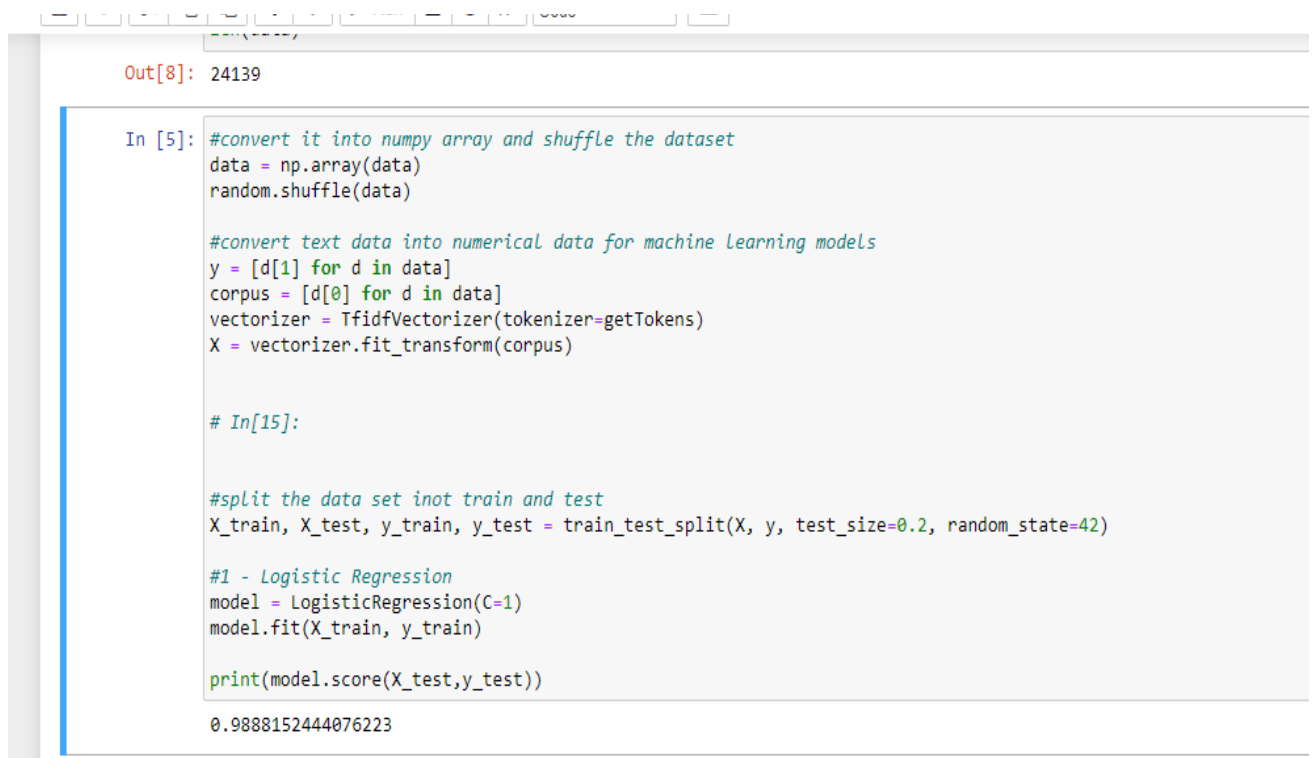
Test Results:

All the above-mentioned test inputs were successful. There were no flaws found.

CHAPTER 4

RESULTS AND DISCUSSIONS

The work on show is still in its early stages. The aim of this paper is to provide a short overview of our approach. The extraction of lexical features may be used to detect malicious URLs, according to one theory. We used the Classifying approach based on the TF - IDF word association to complete the basic investigation. The features extracted from URL bigrams can be supported, and term frequency and inverse term frequency can provide the simplest classification setting. The main task, however, is to identify using the proposed features, and we have completed the preprocessing stage. The work presented here is an early effort in malicious URL detection; in a future work, we will cover the post-processing of the Feature set and include the classifying coefficients that are used as separating parameters.



```
Out[8]: 24139

In [5]: #convert it into numpy array and shuffle the dataset
data = np.array(data)
random.shuffle(data)

#convert text data into numerical data for machine learning models
y = [d[1] for d in data]
corpus = [d[0] for d in data]
vectorizer = TfidfVectorizer(tokenizer=getTokens)
X = vectorizer.fit_transform(corpus)

# In[15]:

#split the data set into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

#1 - Logistic Regression
model = LogisticRegression(C=1)
model.fit(X_train, y_train)

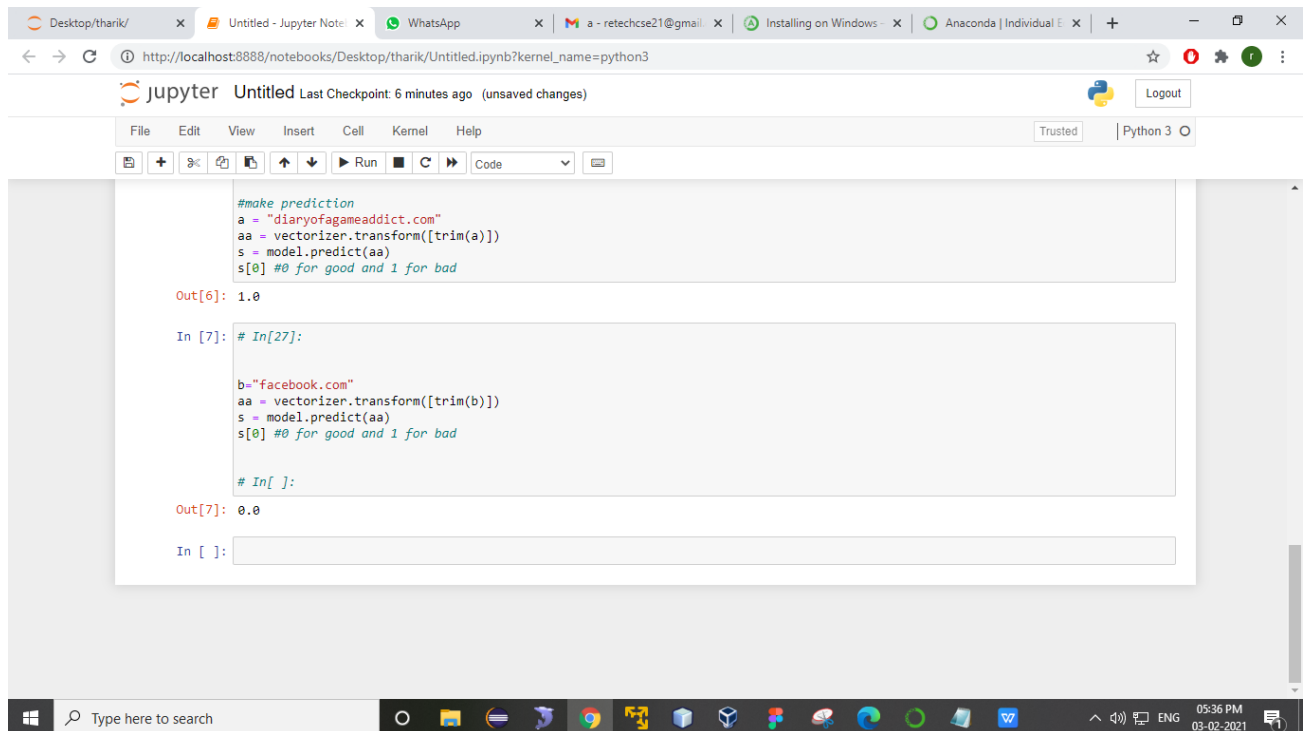
print(model.score(X_test, y_test))

0.9888152444076223
```

Figure 4.1: Finding Accuracy

In fig 4.1 we can see that the algorithm used accuracy has been predicted this has been predicted using the training and testing model of logistic regression. First the dataset is trained and tested, training is done with 80% and testing is done at 20%. Here we can see

the accuracy of 98% previously there is drawback of accuracy. In this model there is high accuracy level we used svm and logistic regression for gaining more accuracy. Before finding accuracy the url in alphabetical order will be changed into numerical data by using vectorizer.



```
#make prediction
a = "diaryofagameaddict.com"
aa = vectorizer.transform([trim(a)])
s = model.predict(aa)
s[0] #0 for good and 1 for bad

Out[6]: 1.0

In [7]: # In[27]:

b="facebook.com"
aa = vectorizer.transform([trim(b)])
s = model.predict(aa)
s[0] #0 for good and 1 for bad

# In[ ]:

Out[7]: 0.0

In [ ]:
```

Figure 4.2: Detecting malicious url

The malicious url has been detected value in binary form. If the output is 0 then it is non-malicious if the output is 1 then input url is malicious. Here the url will be trimmed i.e., it removes the unnecessary things and takes the words in form of tokens and then tokenizes the word with the pre-trained data then compares and then predicts output as either malicious or non-malicious. Whole code is debugged/run in jupyter notebook. All the files are saved with .py extension and then extracted in the jupyter notebook. This code later will be deployed into the website and the users can use the website to check the link whether it is malicious or not. Next the code is deployed in the vs code where the output will be the website.

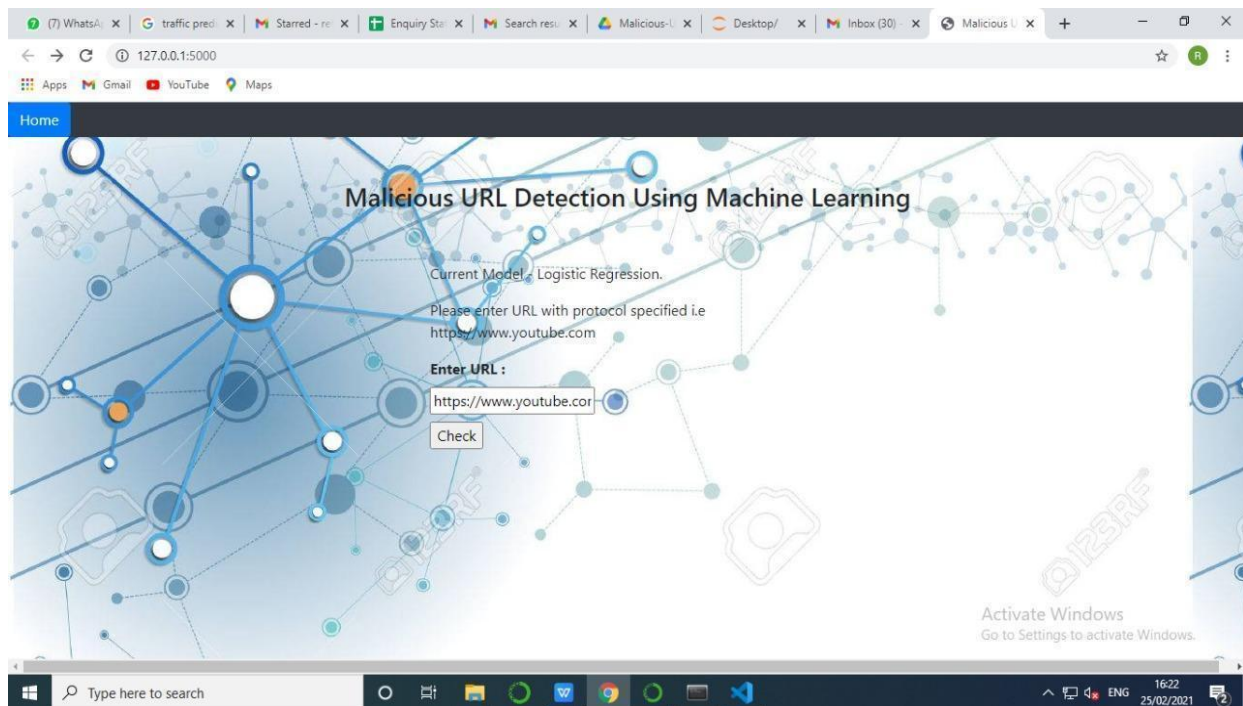


Figure 4.3: Webpage for user to check

This webpage is written using HTML, CSS and Bootstrap. This consists of a paragraph and textbox where user can type the link and search. Here is a protocol that the link should be written in some specific format that is it should be written by using domain and protocol like https and .com, .in, .org etc., When the user enters the url the code at the backend will run and predicts the result in the next page. In the next page this shows as whether the link is malicious or non-malicious to the user and with the input the user given also as shown below.

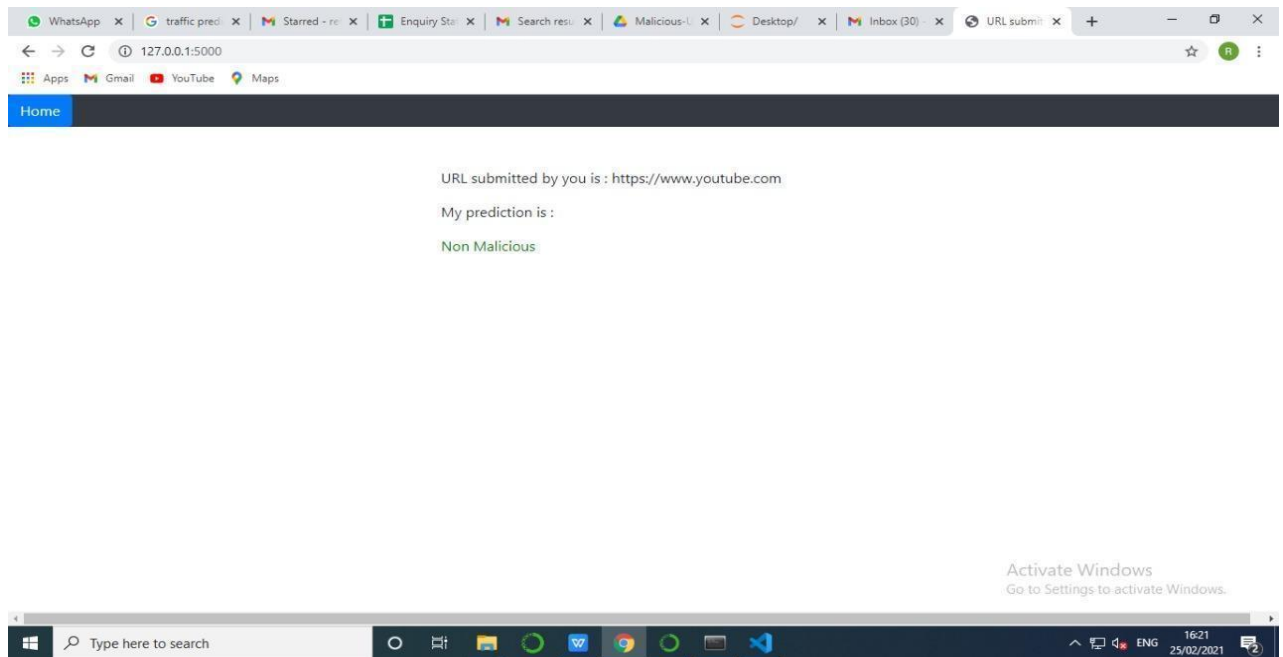


Figure 4.4: Output Prediction

CHAPTER 5

CONCLUSION AND FUTURE WORK

Many cyber security applications depend on malicious URL identification, and machine learning techniques are obviously a promising path. We conducted a thorough and ordered analysis of Malicious Detection using AI approaches in the work. We provided the methodical description of Malicious detection from an AI standpoint, followed by nitty gritty information. Current investigations finds malicious URL identification, especially in the types of growing new component portrayals and preparing new learning calculations for determining vindictive URL position assignments We sorted most, if not all, of the existing obligations for malevolent URL position in writing in this overview, as well as acknowledged the requirements and challenges for creating tasks for detecting malicious URLs. In this analysis, we summarize the majority, if not all, of the existing commitments for malignant URL location in writing, as well as the requirements challenges for the develop of Malicious Detection as the Services for the real-world cyber security application. At long last, some featured less useful issue for application space, demonstrated other significant new issues to additional exploration examination. Specifically, despite extensive research and enormous progress in recent years, automatic detection of spam URLs using AI remains as the challenging open issue. More viable aspect extraction, portrayal learning (example: by profound learning) are expected in the future, as well as more efficient AI calculations for developing predictive models especially for managing idea floats (example: successful internet learning), other arising difficulties (example: area dividing while applying the model to another space), Finally, a clever scheme for protecting named details and client criticism in a closed circle setup (example: coordinating online dynamic learning approach in the genuine system). Finally, we highlighted some fair concerns for the application room as well as some major open issues that need further investigation. Specifically, despite extensive research and enormous progress in recent years, robotized identification of vindictive URLs using AI remains as the challenging open issue. More convincing part extraction and representation learning (e.g., by way of profound learning approaches), and more successful AI calculations are among the future bearings for planning vision model, and particular for handling concept floats (example: efficient internet learning), other arising difficulties (example: space adaptation while adapting a concept to the another area), and

finally a clever plan of closed circle structure for obtaining marked information and client feedback (example: coordinating an online dynamic e-learning approach in the genuine framework).

REFERENCES

1. Abdelhamid N, Ayesh A, Thabtah F (2014) Phishing detection based associative classification data mining. *Science-Direct* 41:5948–5959.
2. Chen KT, Chen JY, Huang CR, Chen JY (2009) Fighting phishing with discriminative key point features of webpages. *IEEE Internet Comput* 13:56–63.
3. Chen X, Bose I, Leung ACM, Guo C (2011) Assessing the severity of phishing attacks: a hybrid data mining approach. *Expert Syst Appl* 50:662–672.
4. Fu AY, Wenyin L, Deng X (2006) Detecting phishing web pages with visual similarity assessment based on earth mover's distance. *IEEE Trans Dependable Secure Comput* 3(4):301–321.
5. Islam R, Abawajy J (2013) A multi-tier phishing detection and filtering approach. *J Netw Comput Appl* 36:324–335.
6. Li Y, Xiao R, Feng J, Zhao L (2013) A semi-supervised learning approach for detection of phishing webpages. *Optik* 124:6027–6033.
7. Nishanth KJ, Ravi V, Ankaiah N, Bose I (2012) Soft computing-based imputation and hybrid data and text mining: the case of predicting the severity of phishing alerts. *Expert Syst Appl* 39:10583–10589.
8. Medvet E, Kirda E, Kruegel C (2008) Visual-similarity-based phishing detection. *SecureComm*. In: *Proceedings of the 4th international conference on Security and privacy in communication networks*. pp 22–25.

9. Xiang G, Hong J, Rose CP, Cranor L (2011) CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. *ACM Trans Inf Syst Secur* 14:21.
10. Zhang Y, Hong JI, Cranor LF (2007) CANTINA: a content-based approach to detecting phishing web sites. In: *Proceedings of the 16th international conference on world wide web, Banff*, p 639–648.
11. R.k. Nepali and Y. Wang “You Look suspicious!!” Leveraging the visible attributes to classify the malicious short URLs on Twitter. in *49th Hawaii International Conference on System Sciences (HICSS) IEEE*, 2016, pp. 2648-2655.
12. Rakesh Verma, Avisha Das, “What’s in a URL: Fast Feature Extraction and Malicious URL Detection” proceeding IWSPA ‘17 *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics* Pages 55-63.

APPENDIX

SOURCE CODE

```
from flask import Flask, render_template
from flask_wtf import FlaskForm as Form
from wtforms import StringField
from wtforms.validators import InputRequired, URL
import joblib
import os

import re

app = Flask(__name__)
app.config['SECRET_KEY'] = os.urandom(24)

def trim(url):
    return re.match(r'(?:\w*://)?(?:.*\.)?([a-zA-Z-1-9]*\.[a-zA-Z]{1,}).*', url).groups()[0]

def getTokens(input):
    tokensBySlash = str(input.encode('utf-8')).split('/')
    allTokens = []
    for i in tokensBySlash:
        tokens = str(i).split('-')
        tokensByDot = []
        for j in range(0, len(tokens)):
            tempTokens = str(tokens[j]).split('.')
            tokensByDot = tokensByDot + tempTokens
        allTokens = allTokens + tokens + tokensByDot
```

```

allTokens = list(set(allTokens))
if 'com' in allTokens:
    allTokens.remove('com')
return allTokens

class LoginForm(Form):
    url = StringField('Enter URL : ', validators=[InputRequired(), URL()])

@app.route('/', methods=['GET', 'POST'])
def index():
    form = LoginForm()
    if form.validate_on_submit():
        model = joblib.load('pre-trained/mal-logireg1.pkl')
        vectorizer = joblib.load("pre-trained/vectorizer1.pkl")
        prediction = model.predict(vectorizer.transform([trim(form.url.data)]))

        if prediction[0] == 0:
            #prediction = "NOT MALICIOUS"
            return render_template("success.html", url = form.url.data, status
= "Non Malicious")
        else:
            #prediction = "MALICIOUS"
            return render_template("success.html", url= form.url.data, status
= "Malicious")
        #return render_template('success.html', url = form.url.data, prediction =
prediction)
    return render_template('index.html', form=form)

if __name__ == '__main__':
    app.run(debug=True)

```

Webpage Code:

```
{% from "_formhelpers.html" import render_field %}

<html>
<head>
<title>Malicious URL Detection</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></scri
pt>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
</head>
<body style="background-image: url('static/bg2.jpg');">
    <div class="nav bg-dark">
        <a href="#" class="btn btn-primary">Home</a>
    </div>
    <br>
    <br>
    <div class="row">
        <div class="col-sm-12">
            <h3> <center>Malicious URL Detection Using Machine Learning
</center></h3>
        </div>
```



```

</div>
<br>
<br>
<div class="row">
    <div class="col-sm-4"></div>
    <div class="col-sm-4">
        <p> Current Model - Logistic Regression. </p>
        <p> Please enter URL with protocol specified i.e
https://www.youtube.com
        <form method="POST" action="/">
            <dl>
                {{ form.csrf_token }}
                {{ render_field(form.url) }}
                <input type="submit" value="Check">
            </dl>
        </form>
    </div>
<div class="col-sm-4"></div>
</div>

</body>
</html>

```