PHISH RECOGNIZER FROM URLS BY USING MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

HARISH.S **(820616104020)**

SURAJ.B **(820616104053)**

VENKATARAMANAN.M **(820616104059)**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

ARASU ENGINEERING COLLEGE, KUMBAKONAM

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

DECLARATION

We declare that the Project work entitled **"PHISH RECOGNIZER FROM URLS BY USING MACHINE LEARNING TECHNIQUES"** is submitted in partial fulfillment of requirement for the award of the degree in B.E., Anna University Chennai, is a record of our own work carried out by us during the academic year 2019-2020 under the supervision and guidance of **Ms.T.Veni Priya M.Tech.,(ph.D)** Assistant Professor, Department of Computer Science and Engineering, Arasu Engineering College at Kumbakonam. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at appropriate places. The matter embodied in this work is original and has not been submitted for the award of any other degree, either in this or any other university.

Name	**Register number**	**Signature**
Harish.S	820616104020	
Suraj.B	820616104053	
Venkataramanan.M	820616104059	

I certify that the declaration made above by the candidate is true.

Signature of the Guide

Ms.T.Veni Priya, M.Tech.,(Ph.D)

Assistant Professor - CSE

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**PHISH RECOGNIZER FROM URLS BY USING MACHINE LEARNING TECHNIQUES**" is the bonafide **HARISH S (820616104020), SURAJ B (820616104053), VENKATARAMANAN M (820616104059)** who carried out the project work under my supervision.

SIGNATURE

Dr.KALAIMANI SHANMUGAM M.Tech.,Ph.D.,
HEAD OF THE DEPARTMENT
Professor,
Department of CSE
Arasu Engineering College,
Kumbakonam-612501.

SIGNATURE

Ms.T.VENI PRIYA M.Tech.,(Ph.D).,
SUPERVISOR
Assistant Professor,
Department of CSE
Arasu Engineering College,
Kumbakonam-612501.

Submitted for Project Viva-Voce examination held on: _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of our project depends largely on the encouragement and guidelines of many others. It is our pleasure to give our special thanks to **ANNA UNIVERSITY**.

Also we thank **ARASU ENGINEERING COLLEGE** especially our Chairman **Mr.R.THIRUNAVUKARASU, M.A.,** and other management people who permitted us to carry out this project work.

At the outset, we wish to express our sincere gratitude to our beloved **PARENTS** who are solely responsible for the successful completion of our B.E Degree course.

It gives immense pleasure to thank **Dr. T. BALAMURUGAN, M.E., Ph.D.,** Principal of the college for providing the necessary infrastructure facilities for the project.

I am very proud and thankful to **Dr. KALAIMANI SHANMUGAM, M.Tech., Ph.D.,** Head of the Department and our guide, for her valuable advice and innovative ideas at various stages of my work.

Our heartfelt gratitude to our project coordinator **Mrs. R. KALAISELVI, M.Tech.,(ph.D)** Assistant Professor of CSE department who continuously supported us in every possible way, from initial advice to encouragement till now.

We would like to show my greatest appreciation to our project guide **Ms.T.VENI PRIYA, M.Tech.,(Ph.D)** Assistant Professor of CSE department, we feel motivated and encouraged every time when we attend her meeting. Without her encouragement and guidance this project would not have materialized.

I also extend my hearty thanks to all the staff members and friends in our college who encouraged us to successfully complete this project.

ABSTRACT

Phishing web sites have been a serious challenge to cyber safety which employs each social engineering and technical trick to steal clients' non-public identity statistics and monetary account information. Recent years have proved the hasty increase of phishing attacks. Phish recognizer is targeted on getting rid of the identity robbery and fraud that result from the growing problem of phishing. We propose phish recognizer, a deep learning based total method which uses logical regression to build accurate function representation of URLs, which we use to teach a phishing URL classifier. Our target is to seek a better overall performance classifier via knowledge of the features of the phishing internet site and choose the satisfactory mixture of them to educate the classifier.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI - Artificial Intelligence

AJAX - Asynchronous JavaScript and XML

ANN - Artificial Neural Networks

CNN - Convolutional Neural Networks

DOM - Document Object Model

DT - Decision Tree

IDE- Integrated Development Environment

JS - JavaScript

ML- Machine Learning

NB - Naive Bayes

NPM- Node package manager

PIP - Python Package Manager

PSQL -Pervasive SQL

SQL - Structured Querying Language

TF - Tensorflow

TFJS - Tensorflow Js

TS - TypeScript

URL - Unified Resource Identifier

1D - One Dimensional

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF DOMAIN

1.1.1 Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions , relying on patterns and inference instead. It is seen as a subset of artificial intelligence. It is just to give trained data to a program and get better results for complex problems. Machine learning algorithms build a mathematical model based on sample data, known as “trained data”. In order to make predictions or decisions without being explicitly programmed to perform a task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. Data mining is a field of study within machine learning , and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as Predictive analytics.

1.1.2 Naive Bayes

The naive bayes classifier originates from the Bayes Theorem, and is powerful While the inputs are excessive. But, the approach is complex and calls for the Consumer to have considerable know-how of the model getting used.Classifiers used require variable parameters which are exceptionally scaled to output the most opportunity. Moreover, those classifiers are used to assess near-shape parameters the usage of linear time in place of the iterative estimate.

1.1.3 Random Forest

Random forest works because of the random link within the version. It assumes that a character is aware of the formation of a single category. The forest Area bureaucracy different 3 classifications within the technique. The testing and education records procedure requires the person to report any end result from the implementation technique The implementation procedure also calls for the user to take several experiments to Validate the Significance of the results.

1.1.4 Neural Networks

Neural networks are networks of units (neurons) primarily based on the real neural Structure of the brain. Gadgets are arranged in layers. Neural networks "learn" through Processing statistics and evaluating their classification of the record with the acknowledged actual classification of the report. The errors from the modern classification step are used to modify the network's set of rules within the next step, and so on for many iterations.

1.1.5 Deep Learning Methodologies

Deep learning (also known as **deep structured learning** or **differential programming)** is Part of a broader family of machine learning methods based on artificial neural networks With representation learning. Learning can be supervised, semi-supervised or Unsupervised. Deep learning architectures such as deep neural networks,deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering ,machine translation, bioinformatics, medical image analysis.

Artificial neural networks (ANNs) were inspired by information processing and distributed Communication nodes in biological systems. ANNs have various differences from biological brains. Specifically , neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.

1.2 PROJECT DESCRIPTION

Cyber security is the organization and collection of resources, processes and structures used to defend cyberspace and cyberspace enabled systems from events that misrelate by default ownership rights. Cyber security is the accumulation of tools such as policies, security safeguards, training, risk management approaches guarantee and technologies that can be used to protect the cyber organization and environment. Due to the extensive growth in the number of internet users, lots of our daily life operations are transferred from the real world to the cyber world such as communication, coordination, commerce, banking, registrations, applications, etc. Because of this, the malicious people and attackers also transferred to this world and made their threats and crimes easily and anonymously. To ensure the security and privacy of cyber data, technology must be used and organized carefully by using Cyber Security. Cyber security prevents fraud or thieves who want to seize person/public/national information or connection, **"Identity theft"** or specifically **"phishing"** is one of the most threatening security deficits of the users on the Internet. In this type of crimes, attackers use some malicious web pages which impersonate as legitimate websites, to collect the victims' critical information such as username,passwords, financial data, etc.Typically, a phishing attack starts with an electronic mail which seems to come from a reputable company as depicted in security issues.

The content of the mail encourages the victim to click on the address, which can also be hidden as a hypertext. This address directs the victim to a fake

website, which is designed exactly similar to a valid website, such as an e-mail social engineering site of generally financial institutions websites. To overcome this type of attack there's need to construct a dynamic and efficient algorithm which can learn the structure of the legitimate web pages and classify the abnormal ones. Therefore, this project is aimed to set up a classification system, which can identify whether an URL is either phishing or legitimate. To train the system a dataset which contains about 74,000 items in both these types is used. To compare the efficiency of the different algorithms and select the best one, both Classical ML classifier algorithms and Deep Neural Network(DNN) approaches are used for training and testing the system in the Tensorflow framework and scikit frameworks are used and experimental results showed that the proposed approaches produce very good accuracy rates for detecting phishing URLs and email spoofing, phishing is a criminal mechanism which employs both social engineering and technical subterfuge to steal customers' personal identity information and financial account data. Recent years have witnessed the rapid growth of phishing attacks Implementation of a system which collected many legit and phishing URLs and extracted effective features for URL classification. The features are extracted from both URL and website content information. With the collected dataset, they evaluated and compared various classification algorithms such as Decision Tree(DT), Random Forest (RF), Naive Bayes (NB), and Convolutional Neural Network (NN),machine learning techniques ,and deep learning methodology.

Phishing websites have been a serious threat to cyber security which employs both social engineering and technical tricks to steal customers' personal identity information. and financial account data. Recent years have proved the hasty increase of phishing attacks. Phish Recognizer is focused on eliminating the identity theft and fraud that results from the growing problem of phishing. The proposed phish recognizer is a deep learning based approach which uses

CNN to build accurate feature representation of URLs to train a phishing URL classifier. Target is to seek a higher performance classifier by understanding the features of the phishing website and choose the best combination of them to train the classifier.

CHAPTER 2

LITERATURE SURVEY

There are varieties of anti-phishing browser extensions and anti-phishing techniques available today using different techniques/methods like black list, white list, heuristic or user ratings to identify a fraudulent web site. A few of them are studied below showing the technique used along with their merits and demerits.

Protecting users against phishing attacks with AntiPhish
Authors: E.Kirda, C. Kruegel 2005

AntiPhish is a browser extension or plug-in tool that keeps track of a user's sensitive information like password and prevents this information from being passed to a website that is not considered safe. The checking is done by the application instead of the users (who are considered as lay men without any experience. Automated form-filler applications have inspired the development of AntiPhish. Most browsers such as the Internet Explorer or Mozilla have integrated common functionality that allows form contents to an online banking website to be stored and automatically inserted if the user desires which is protected by a master password. When AntiPhish is installed, a new master password is requested by the browser as soon as the user enters inputs into a form for the first time. The sensitive information is captured and stored by the AntiPhish menu after the password is entered. The encryption of the sensitive information is done by the master password before it is stored. The algorithm that is used for encryption and decryption is a symmetric DES algorithm. In this implementation of AntiPhish, the user has to tell the browser extension which piece of information available in the page is important and need to be protected from phishing attacks. The AntiPhish menu scans the page once the user enters sensitive information like a password. It captures and stores this information as well as a mapping of where its information belongs to.

Anomaly Based Web Phishing Page Detection
Authors: Ying Pan , Xuhua Ding 2006

They proposed a novel approach, which is independent of any specific phishing implementation. Their idea is to examine the anomalies in Web pages, in particular, the discrepancy between a Web site's identity and its structural features and HTTP transactions. It demands neither user expertise nor prior knowledge of the Web site. The evasion of our phishing detection entails high cost to the adversary. Their phishing detector functions with low miss rate and low false-positive rate

Phishing URL Detection via CNN and Attention-Based Hierarchical RNN
Authors: Yongjie Huang, Qiping Yang, Jinghui Qin, Wushao Wen 2019

They proposed PhishingNet, a deep learning-based approach for timely detection of phishing Uniform Resource Locators (URLs). Specifically, they used a Convolutional Neural Network (CNN) module to extract character-level spatial feature representations of URLs meanwhile, They employed an attention-based hierarchical Recurrent Neural Network(RNN) module to extract word-level temporal feature representations of URLs. They fused these feature representations via a three-layer CNN to build accurate feature representations of URLs, on which they trained a phishing URL classifier. Extensive experiments on a verified dataset collected from the Internet demonstrate that the feature representations extracted automatically are conducive to the improvement of the generalization ability of our approach on newly emerging URLs, which makes their approach achieve competitive performance against other state-of-the-art approaches.

Detection and Prevention of Phishing Attack: An Approach for Eradication of Phishing

Authors: Shruti Ashok Mandake , R H Goudar 2016

Phishing is like masquerading the trusted party to acquire the sensitive information from users. Phishing attacks are usually carried out through fake websites, fake URLs, fake attachments in emails, fake messages. The main aim of phishing attacks is to fool the users by finding the weakness of the user. One of the best steps to be taken to avoid this attack is to educate the users about the fake links given in the website, where they should not visit such links and give the required credentials. Attackers find many ways to fool the users for browsing the fake website where they are given their personal credentials. In their proposed system there are two methods, in the first method urls are considered from email, keyword search, website and compared with the database and in the second method to detect Phishing through image.

Detection of phishing attacks

Authors: Muhammet Baykara, Zahit Ziya Gürel

Phishing is a form of cybercrime where an attacker imitates a real person institution by promoting them as an official person or entity through e-mail or other communication mediums. In this type of cyber attack, the attacker sends malicious links or attachments through phishing emails that can perform various functions, including capturing the login credentials or account information of the victim. These emails harm victims because of money loss and identity theft. In this study, a software called “Anti Phishing Simulator” was developed, giving information about the detection problem of phishing and how to detect phishing emails. With this software, phishing and spam mails are detected by examining mail contents. Classification of spam words added to the database by Bayesian algorithm is provided by them.

CHAPTER 3

PROBLEM DEFINITION

3.1 PROBLEM STATEMENT

Cybersecurity is the agency and series of assets, methods and systems used to shield cyberspace and cyberspace enabled systems from events that misrelate through default possession rights. Cyber safety is the build-up of equipment along with regulations, security safeguards, education, chance management techniques guarantee and technology that may be used to protect the cyber organisation and environment. Because of the giant increase inside the wide variety of net users, lots of our everyday life operations are transferred from the real international to the cyber international which includes communique, coordination, trade, banking, registrations, packages, and many others. Due to The malicious peoples and attackers also transferred to this international and make their threats and crimes effortlessly anonymous. To make sure the safety and privacy of cyber statistics, era need to be used and prepared carefully with the aid of the use of Cyber safety. Cyber protection prevents fraud or thieves who want to seize person/public/countrywide information or connection, **"identification robbery"** or especially **"phishing"** is one of the maximum threatening safety deficits of the users within the internet. On this form of crimes,attackers use a few malicious net pages which impersonate as valid web websites, to collect the victims' critical records consisting of username, passwords,monetary data, and many others. Usually, a phishing attack starts with an e-mail which appears to return from a good enterprise as depicted in safety problems.The content of the mail encourages the victim to click on the cope with, which can also be hidden as a hypertext. This address directs the sufferer to a fake web website online, which is designed exactly similar to a valid internet site, together with an e-mail web page social engineering site of normally financial institutions web websites. To

overcome this form of attack there is need to construct a dynamic and green algorithm that could learn the shape of the valid web pages and classify the strange ones. Therefore, on this mission, we aimed to set up a classification system that can discover whether or not an URL is both phishing or legitimate. examine the efficiency of the one of a kind algorithms and select the quality one, we used each artificial Neural network (ANN) and Deep Neural community(DNN) processes for schooling and checking out the system with the assist of Tensor float framework. And experimental consequences showed that the proposed methods produce superb accuracy fees for detecting phishing URLs and electronic mail spoofing, phishing is a criminal mechanism which employs both social engineering and technical subterfuge to spouse borrow customers' personal identification data and financial account facts.Implementation of a machine which accrued many respectable and phishing URLs and extracted powerful features for URL classification. The functions are extracted from both URL and internet site content information. With the amassed dataset, the authors evaluated and in comparison various classification algorithms which includes choice Tree (DT), Random wooded area (RF), Naïve Bayes (NB), and Neural Networks (NN), system learning strategies, deep studying technique.

3.2 PROPOSED SYSTEM

The Proposed Phish algorithm is primarily based on a computerized real-time phishing detection and advice mastering procedure. The phishing URLs in most cases have a couple of connections among the part of the URL which means an inter-relatedness and with the aid of the use of it the capabilities of phishing URLs are extracted. Then the extracted capabilities are used for a device-gaining knowledge of class to come across Phishing websites on actual time. The characteristics of phishing web sites which used to differentiate from legitimate websites. The values are then assigned to each phishing indicator with

the variety defined for phishing website risk. The URLs are pre-processed for the feature extraction method. The pre-processed dataset is used to extract the phishing functions for each URL under 4 categories: Addressed based totally characteristic, odd function, HTML, JavaScript feature and domain function. These primary functions have a number of 30. The evaluation data set is composed of 11,745 phishing websites collected from PhishTank, and 6,000 legitimate websites collected from Random Yahoo! Link, a website that returns random URLs of other websites.

3.3 ADVANTAGES OF PROPOSED SYSTEM

- The better accuracy is achieved with less features.
- Our approach outperforms the existing solutions significantly with the help of Random Forest.
- The proposed system is an internet browser extension that automatically notifies the user when it detects a phishing website.

CHAPTER 4

SYSTEM STUDY

4.1 FEASIBILITY STUDY

This project is feasible provided given unlimited resources and infinite time. Unfortunately the development of a computer-based system is more likely to be plagued by resource scarcity and stringent schedules. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Wastage of manpower and financial resources and untold professional embarrassment can be avoided if an ill-conceived system is recognized early in the development phase. So a detailed study was carried out to check the workability of the proposed system. Feasibility study is a test of system proposal regarding its workability, impact on the organization, ability to meet user needs and effective use of resources. Thus, when an application is proposed, it normally goes through a feasibility study before it is approved for development.

Feasibility and risk analysis is related in many ways. If project risk is great, the feasibility of producing quality is reduced. Thus during feasibility analysis for this project, following three primary areas for interest was considered very carefully. There are several types of feasibility.

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

4.1.1 Technical Feasibility

A study of resource availability that may affect the ability to achieve an acceptable system. This system is just required to be installed as a browser extension and thus requires no additional requirements and hence this system is technically feasible.

4.1.2 Operational Feasibility

It is essential that the process of analysis and definition can be conducted in parallel with an assessment of technical feasibility. No hardwares and additional software resources are required by the system and hence this system is operationally feasible.

4.1.3 Economic Feasibility

An evaluation of development cost weighted against the ultimate income or benefit derived from the proposed system. No extra hardwares, platforms or operations are required by the system and hence this system is economically feasible.

CHAPTER 5

SYSTEM REQUIREMENTS & SPECIFICATION

5.1 SYSTEM REQUIREMENTS

5.1.1 Hardware Requirements

Processor : Intel Pentium inside

RAM : 2GB

Hard Disk : 500 GB

Input Devices : Keyboard and mouse

Output Devices : Monitor

Browser: Modern web browser

5.1.2 Software Requirements

Operating system : Windows,linux,Mac os

Front-End : Angular JS

Back-End : postgreSQL

Framework : Python Flask

Language : Python

IDE : Visual Studio code

5.2 SOFTWARE DESCRIPTION

PYTHON

- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.
- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Locating Modules

- When you import a module, the Python interpreter searches for the module in the following sequences.
- The current directory.
- If the module isn't found, Python then searches each directory in the shell variable PYTHONPATH.
- If all else fails, Python checks the default path. On UNIX, this default path is normally /usr/local/lib/python/.

- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

- The operand to the left of the = operator is the name of the variable and the operand to the right of the = operator is the value stored in the variable.

- Python has been an object-oriented language since it existed. Because of this, creating and using classes and objects are downright easy. This chapter helps you become an expert in using Python's object-oriented programming support.
- If you do not have any previous experience with object-oriented (OO) programming, you may want to consult an introductory course on it or at least a tutorial of some sort so that you have a grasp of the basic concepts.
- However, here is a small introduction of Object-Oriented Programming (OOP) to bring you at speed.

OVERVIEW OF OOP

TERMINOLOGY

Class − A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

Class variable − A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.

Data member − A class variable or instance variable that holds data associated with a class and its objects.

Function overloading − The assignment of more than one behavior to a particular function. The operation performed varies by the types of objects or arguments involved.

Instance variable − A variable that is defined inside a method and belongs only to the current instance of a class.

Inheritance − The transfer of the characteristics of a class to other classes that

are derived from it.

Instance − An individual object of a certain class. An object obj that belongs to a class Circle, for example, is an instance of the class Circle.

Instantiation − The creation of an instance of a class.

Method − A special kind of function that is defined in a class definition.

Object − A unique instance of a data structure that's defined by its class. An object comprises both data members (class variables and instance variables) and methods.

Operator overloading − The assignment of more than one function to a particular operator.

SOFTWARE ENVIRONMENT

Using JavaScript

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) have covered in much more detail in other parts of the Learning Area. **postgresql** is a fast and powerful yet easy-to-use database system that offers just about anything a website might need in order to find and serve up data to browsers. When Javascript and python allies with My SQL to store and retrieve this data, you have the fundamental parts required for the development of social networking sites and the beginnings ofWeb2.0. And when you bring JavaScript and CSS into the mix too, you have a recipe for building highly dynamic and interactive websites.

Using Flask

Like most widely used Python libraries, the Flask package is installable from the Python Package Index (PPI). First create a directory to work in (something like **flask_todo** is a fine directory name) then install the **flask** package. You'll also want to install **flask-sqlalchemy** so your Flask application has a simple way to talk to a SQL database.

Using PostgreSQL

Of course, there's not a lot of point to being able to change HTML output dynamically unless you also have a means to track the changes that users make as they use your website. In the early days of the Web, many sites used "flat"text files to store data such as usernames and passwords. But this approach could cause problems if the file wasn't correctly locked against corruption from multiple simultaneous accesses.

Also, a flat file can get only so big before it becomes unwieldy to manage— not to mention the difficulty of trying to merge files and perform complex searches in any kind of reasonable time. That's where relational databases with structured querying becomes essential. And postgreSQL, being free to use and installed on vast numbers of Internet web servers, rises superbly to the occasion. It is a robust and exceptionally fast database management system that uses English-like commands.

The highest level of postgreSQL structure is a database, within which you can have one or more tables that contain your data. For example, let's suppose you are working on a table called users, within which you have created columns for surname, first name, and email, and you now wish to add another user. One command that you might use to do this is: INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com'); Of course, as mentioned earlier, you will have issued other commands to create the database and table and to set up all the correct fields, but the INSERT command here shows how simple it

can be to add new data to a database.. It is well suited,however, to database queries, which is why it is still in use after all this time. It's equally easy to look up data. As you'd expect, there's quite a bit more that you can do with postgreSQL than just simple INSERT and SELECT commands. For example, you can join multiple tables according to various criteria, ask for results in a variety of different orders, make partial matches when you know only part of the string that you are searching for, return only the nth result, and a lot more. Using Flask , you can make all these calls directly to postgreSQL without having to run the postgreSQL program yourself or use its command-line interface. This means you can save the results in arrays for processing and perform multiple lookups, each dependent on the results returned from earlier ones, to drill right down to the item of data you need. For even more power, as you'll see later, there are additional functions built right into postgreSQL that you can call up for common operations and extra speed.

Using an IDE

As good as dedicated program editors can be for your programming productivity, their utility pales into insignificance when compared to Integrated Development Environments (IDEs), which offer many additional features such as in- editor debugging and program testing, as well as function descriptions.

CHAPTER 6

SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

The architecture diagram describes the overall function of the project.User can install the browser extension which captures URL when the user browses a website.

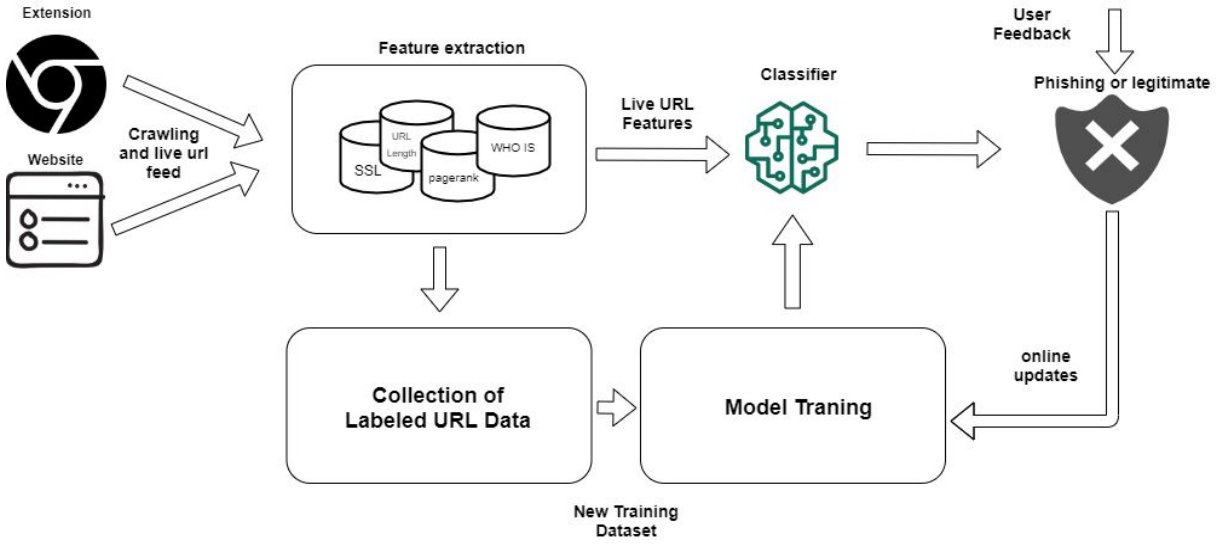


Fig 6.1: System Architecture

Features are extracted from the captured email and then with the help of classification model trained with thousands of URLs the captured URL can be classified as either legitimate or phishing website. Then the new URL is added to the dataset for training the model further with new data.

6.2 DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). Input URL is split into arrays and features are extracted. When features are extracted then it is classified by model into either legitimate or phishing.

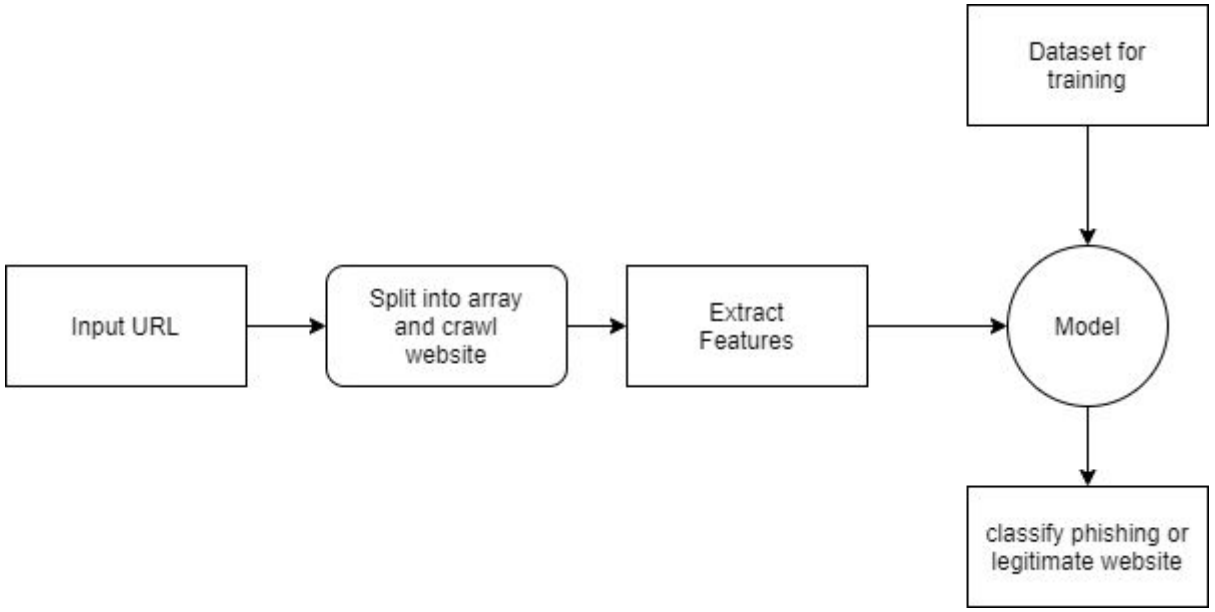


Fig 6.2: Data Flow Diagram for Working System

6.3 UML DIAGRAMS

The unified modeling language is a general-purpose, developmental, Modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

6.3.1 Use Case Diagram

Use case diagrams are referred to as behaviors diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

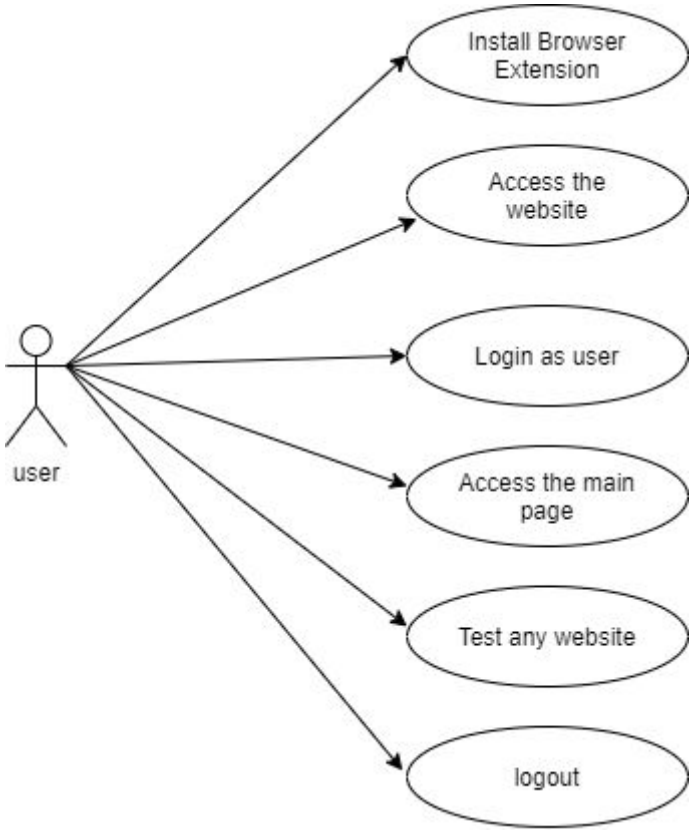


Fig 6.3: Use Case Diagram

6.3.2 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing a system's classes, their attributes, operations (or methods), and the relationships among objects.

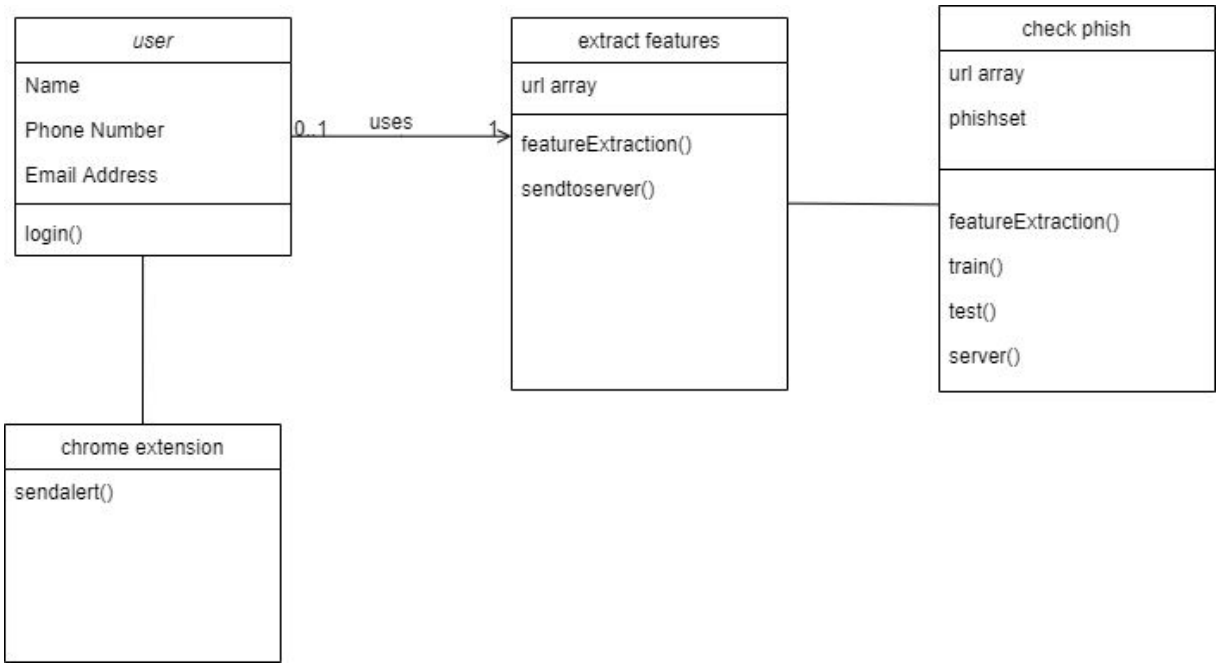


Fig 6.4: Class Diagram

6.3.3 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

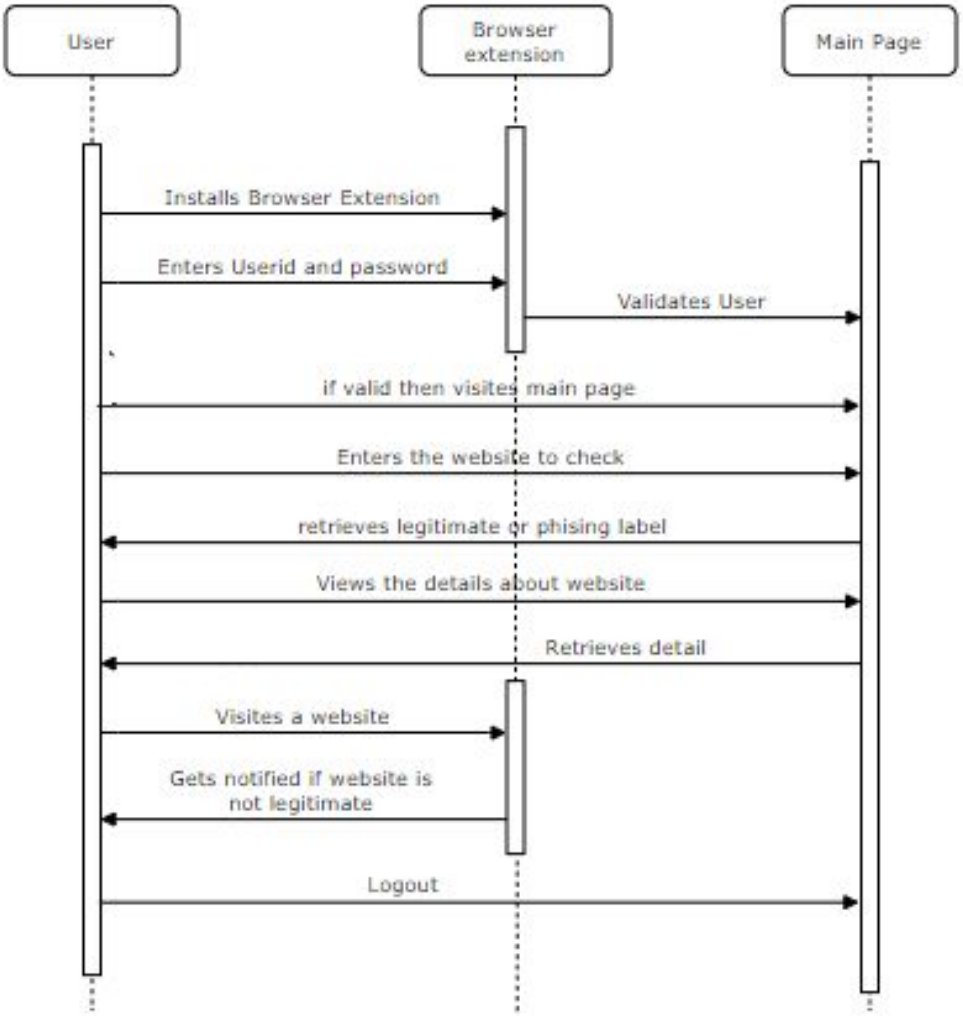


Fig 6.5: Sequence Diagram

6.3.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes.

Fig 6.6: Activity Diagram

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 SYSTEM DESCRIPTION

System implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively.

System implementation is the stage in the project where the theoretical designs turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively.

7.2 MODULES DESCRIPTION

- Preprocessing
- Future Extraction
- Classification
- Web page Prediction

7.2.1 Preprocessing

Preprocessing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

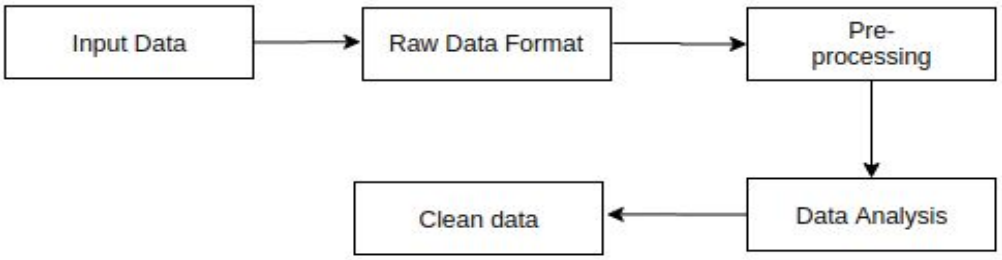


Fig 7.1: Preprocessing

7.2.2 Feature Extraction

This module is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.

Tabulation 7.2: Features of extraction

Feature category	Attriutes	values
URL	having_IPhaving_IP_Address	{0,1}
	URL_Length	{-1,0,1}
	port	{-1,0,1}
	having_At_Symbol	{-1,1}
	double_slash_redirecting	{-1,0,1}
	URL_of_Anchor	{-1,0,1}
	Prefix_Suffix	{-1,1}
Domain features	Favicon	{-1,0,1}
	Shortining_Service	{-1,1}
	HTTPS_token	{-1,1}
	Page_Rank	{-1,0,1}
	Request_URL	{0,1}
Abnormal Features	Links_in_tags	{-1,0,1}
	SFH	0,1}
	on_mouseover	{-1,0,1}
	Submitting_to_email	{-1,0,1}

7.2.3 Classification

Random forest works because the random link within the version. It assumes that a character is aware of the formation of a single category. The forest area bureaucracy different 3 classifications within the technique. The testing and education records procedure requires the person to report any end result from the implementation technique. The implementation procedure also calls for the user to take several experiments to validate the significance of the results.

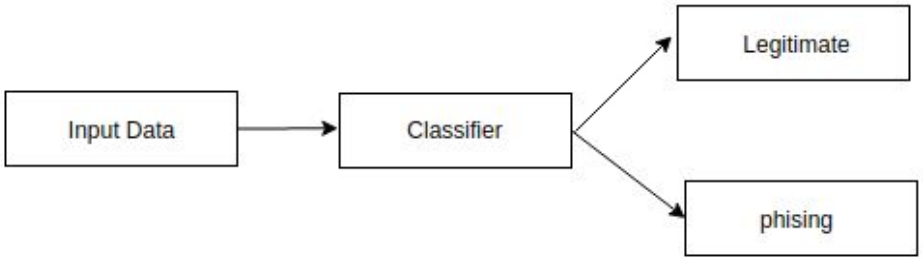


Fig 7.3: Classification Process

CHAPTER 8

SAMPLE CODE

CODE

```
from flask import Flask,jsonify,request
from flask_cors import CORS, cross_origin
app = Flask(__name__)
import re
import requests
from bs4 import BeautifulSoup
from mechanize import Browser
import OpenSSL
import ssl, socket
import numpy as np
import pandas as pd
from scipy import stats
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
cors = CORS(app)



validLink =
re.compile("^(?:http(s)?:\/\/)?[\w.-]+(?:\.[\w\.-]+)+[\w\-\._~:/?#[\]@!\$&'\(\)\*\+,
;=.]+$")

ipaddressRegEx =
re.compile("^(http:\/\/www\.|https:\/\/www\.|http:\/\/|https:\/\/)?[0-9]+([\-\.]{1}[a-
z0-9]+)*\.[a-z]{2,5}(:[0-9]{1,5})?(\/.*)?|^((http:\/\/www\.|https:\/\/www\.|http:\/\
/|https:\/\/)?([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]
|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\/.*)?$")


redirectionRegex = re.compile("([^:]\/{2,3})")

subdomainRegex = re.compile("(?:http[s]*\:\/\/)*(.*?)\.(?=[^\/]*\..{2,5})")
```

```
dict =
{"having_IP_Address":1,"URL_Length":1,"Shortining_Service":1,"having_At_
Symbol":1,"double_slash_redirecting":1,"Prefix_Suffix":1,"having_Sub_Domai
n":1,"Favicon":1,"port":1,"HTTPS_token":1,"Request_URL":1,"URL_of_Anch
or":1,"metaData":1,"SFH":1,"Submitting_to_email":1,"on_mouseover":1,"Ifram
e":1,"Page_Rank":1}

@app.route('/checkurl2',methods=["GET"])
@cross_origin()
def checkURL2():
    url = request.args.get('url')
    print(url)
    return jsonify({"data":"jbadjb"})


@app.route('/checkurl',methods=["GET"])
@cross_origin()
def checkURL():
  url = request.args.get('url')
  r = requests.get(url)
  soup = BeautifulSoup(r.text, "html.parser")
  browser = Browser()
  def split(word):
    return [char for char in word]
  letterArray = split(url)
  length = len(letterArray)
  # "having_At_Symbol"
  for letter in letterArray:
      if(letter == '@') :
          print("found @ symbol")
          dict["having_At_Symbol"] = -1
          break
  # having ip address
  if ipaddressRegEx.match(url):
          print("found a match")
          dict["having_IP_Address"] = -1

  # having subdomain
  subdomaintest = subdomainRegex.match(url)
  if subdomaintest:
      if subdomaintest.group(1) != "www":
          print("found a subdomain match")
```

```
        dict["having_Sub_Domain"] = -1



#length of the url
if length >= 52:
    dict["URL_Length"] = -1
elif length >25 and length < 52:
    dict["URL_Length"] = 0

# doubleslash redirection regex
redirectiontest = redirectionRegex.search(url)
if redirectiontest:
        print("found a match")
        dict["double_slash_redirecting"] = -1
        print(redirectiontest.groups())


#find url shortener
browser.open("http://checkshorturl.com/expand.php")
browser.select_form(nr=0)
browser['u'] = url
response = browser.submit()
content = response.read()
shortenedLink = BeautifulSoup(content,"html.parser")
# for td in soup.find_all('td'):
#         print(td)
if len(shortenedLink.find_all('td')) > 0:
    # print(soup.find_all('td')[1].get_text())
        dict["Shortining_Service"] = -1



# "having_hyphen_Symbol"
for letter in letterArray:
    if(letter == '-') :
        print("found - symbol")
        dict["Prefix_Suffix"] = -1
        break

linkTagWithFavicon = soup.find('link',{"rel" : "icon"})
if linkTagWithFavicon:
    linkhref = linkTagWithFavicon['href']
```

```
    print(linkhref)
    redirectiontest = validLink.match(linkhref)
    if redirectiontest:
        dict["Favicon"] = -1

#http port
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
try:
    s.connect((url, 80))
    s.shutdown(2)
except:
    dict["port"] = -1



#https inside domain name
splittedURL = url.split("//")
domainname = splittedURL[1]
if 'https' in domainname :
        dict["HTTPS_token"] = -1



#multimedia are loaded from same site
srcArr = []
count = 0
images = soup.find_all('img')
try:
    for img in images:
        print(img['src'])
        srcArr.append(img['src'])
except:
    print('Error')


videos = soup.find_all('video')
try:
    for video in videos:
        srcArr.append(video['src'])
except:
    print('Error')
```

```
iframes = soup.find_all('iframe')
try:
    for i in iframes:
        srcArr.append(i['src'])
except:
    audio = soup.find_all('audio')
    for a in audio:
        srcArr.append(a['src'])

for src in srcArr:
    print(src)
    if validLink.match(src):
        count+=1

if count >= (len(srcArr)/2):
    dict["Request_URL"] = -1
# print("phish")
# print(count,len(srcArr))


#URL_of_Anchor
anchor = soup.find_all('a')
anchorArray = []
try:
    for a in anchor:
            anchorArray.append(a['href'])
            dict["URL_of_Anchor"] = -1
            continue
except:
    print('No href in anchor')
# print(anchorArray)
anchorArrayhreflen = len(anchorArray)
anchorArraylen = len(anchor)
if anchorArraylen != anchorArrayhreflen:
    dict["URL_of_Anchor"] = -1



#metaData
metadata = soup.find_all('meta')
if len(metadata) == 0:
    print("no meta data available")
    dict['metaData'] = -1
```

```
elif len(metadata) < 2:
    dict['metaData'] = 0
    print("Meta Data Available.")


#SFH
formactionArray = []
forms = soup.find_all('form')
try:
    for f in forms:
        formactionArray.append(f['action'])
except:
    print('error in sfh')
if len(formactionArray) < len(forms):
    dict["SFH"] = -1
print(formactionArray)

#mailto
mailtoRegex = re.compile('\'mailto\:([^">]+)\'([^>]*)')
for action in formactionArray:
    print(action)
    mailtotest = validLink.match(action)
    print(mailtotest)
    if mailtotest:
        dict["Submitting_to_email"] = -1


#on_mouseover

for a in anchor:
    try:
      if a['onMouseOver']:
          dict["on_mouseover"] = -1
    except:
        1+1

#iframes
for i in iframes:
    try:
      if i['frameborder'] == 0:
          dict['Iframe'] = -1
    except:
          1 + 1
```

```
  browser.open("https://checkpagerank.net/index.php")
  browser.select_form(nr=1)
  browser['name'] = url
  prresponse = browser.submit()
  prcontent = prresponse.read()
  pagerank = BeautifulSoup(prcontent,"html.parser")
  try:
    pageRankofurl = pagerank.find_all('h2')[2].find_all('b')[1].text.split('/')[0]
    print("PageRank of the URL is:"+pageRankofurl)
    if len(pageRankofurl) > 0:
        if float(pageRankofurl) < 5:
            dict["Page_Rank"] = -1
  except:
      1+1

  print(dict)
  datatoPredict = list(dict.values())
  columnsTitles =
["having_IPhaving_IP_Address","URLURL_Length","Shortining_Service","ha
ving_At_Symbol","double_slash_redirecting","Prefix_Suffix","having_Sub_Do
main","Favicon","port","HTTPS_token","Request_URL","URL_of_Anchor","L
inks_in_tags","SFH","Submitting_to_email","on_mouseover","Iframe","Page_R
ank"
  ]
  df= pd.read_csv(r'C:\Users\imhar\Documents\dataset.csv')

  result=df['Result']
  features=df.reindex(columns=columnsTitles)
  # print(features.head())
  # print(features.columns)
  train_features, test_features, train_labels, test_labels = train_test_split(features,
                                            result,
                                            test_size = 0.01,
                                            random_state = 500)

  forest_model = RandomForestRegressor(random_state=500)
  forest_model.fit(train_features, train_labels)
  predictions = forest_model.predict(test_features)
  guess = forest_model.predict([datatoPredict])
  print(mean_absolute_error(test_labels, predictions))
  print(100-mean_absolute_error(test_labels, predictions))
  print([datatoPredict])
```

```
  print(guess.tolist()[0])
  if guess.tolist()[0] <= 0:
    return jsonify({"data":"phish"})
  else:
    return jsonify({"data":"legitimate"})
```

ANGULAR JS CODE:

```
 <mat-toolbar color="accent" id="navbar">
<mat-toolbar-row>
    <span>phisherman</span>
    <span class="example-spacer"></span>
    <mat-icon class="example-icon" aria-hidden="false" aria-label="Example
user verified icon">verified_user</mat-icon>
  </mat-toolbar-row>
</mat-toolbar>
<div class="container__lander flex">
  <div class="brand flex">
      <img src="../assets/Capture.PNG" id="brand-logo" alt="logo">
      <h1 class="brand-name">phisherman</h1>
  </div>
  <p class="explaining-text">
    Analyze suspicious URLs to detect types of malware, automatically share
them <br> with the security community
  </p>

<form #form="ngForm" class="form-group">
    <mat-form-field appearance="outline" class="search-formfield">
        <mat-label>URL to check</mat-label>
        <input matInput ngModel class="form-control" name="url"
placeholder="Placeholder">
        <mat-icon class="check-button" type="button" matSuffix
matTooltip="Check" (click)="onSubmit(form.value)">verified_user</mat-icon>
      </mat-form-field>

</form>
<div class="flex result-container">
  <i class="material-icons" id="result" [ngClass]="{'redText': phish, 'greenText':
phish}">
    {{resultemoji}}
    </i>
    <b>{{result_message}}</b>
</div>
<p id="terms_conditions">
```

By submitting your file to phisherman you are asking phisherman to share your submission with the
 security community and agree to our Terms of Service and Privacy Policy. Learn more.
</p>

JS CODE:

```
import { Component } from "@angular/core";
import {
FormBuilder,
FormControl,
FormGroup,
Validators
} from "@angular/forms";
import {CheckurlService} from "../../src/app/checkurl.service"
import { prepareSyntheticListenerFunctionName } from
'@angular/compiler/src/render3/util';


@Component({
selector: "app-root",
templateUrl: "./app.component.html",
styleUrls: ["./app.component.scss"]
})
export class AppComponent {
resultemoji :any = "";
result_message : string = "";
phish:boolean = false;
constructor(private fb: FormBuilder,private checkurl:CheckurlService) {}
ngOninit() {
}

onSubmit(url: any){
  this.phish = false;
  this.resultemoji = "security";
  this.result_message = "Analyzing the URL.....";
    this.checkurl.checkUrl(url.url).subscribe(res => {
      console.log(res.data)
      if(res.data === "legitimate"){
        this.phish = false;
        this.resultemoji = "mood";
        this.result_message = "You can surf without any fear!!!";
        console.log("legitimate")
```

```
      }else if(res.data === "phish"){
        this.phish = true;
        this.resultemoji = "mood_bad";
        this.result_message = "oww!! phishing website!!!";
        console.log("phish")
      }
    });
  }
}
```

CSS:

```
.example-spacer {
   flex: 1 1 auto;
 }

 #navbar{
   font-family: 'Gloria Hallelujah', ;
 }
 .container__lander{
     min-height:88vh;
     flex-direction: column;
     margin:0 8em;
 }

#terms_conditions{
  margin-top:auto;
 font-family: 'Gloria Hallelujah', ;
  font-size:0.7em;
  margin-bottom:1em;
}
.brand{
 margin-top:10vh;

}
 .brand-name{
   font-family: 'Gloria Hallelujah', ;
   margin-left:1em;
 }


.flex{
 display:flex;
```

```
  align-items: center;
  justify-content: center;
}
.result-container{
  font-family: 'Gloria Hallelujah', ;
  flex-direction: column;
}

#result{
  font-size:5em;
}

.greenText{
  color:green;
}

.redText{
  color:red;
}

  .check-button{
    cursor: pointer;
  }
  .explaining-text{
    font-family: 'Gloria Hallelujah', ;
    padding:3em;
    text-align: center;
  }

  #brand-logo{
    max-height:20vh;
    max-width: 20vw;
  }
    .search-formfield{
      width:60vh;
    }

  button{
      margin-left:1em;
    //   height:
    padding:0.6em;
  }
```

CHAPTER 9

SYSTEM TESTING AND MAINTENANCE

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results testing is used later on dTuring maintenance also.

9.1 TESTING

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of the testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work.Testing is the process of executing a program with the intent of finding errors.

9.2 SYSTEM TESTING

Testing is the analysis of source/executable code and the controlled execution of executable code to reveal defects that comprise a JavaScript program executable integrity. Defects often lead to erratic behavior or the premature termination of an executing program.

The Software testing process commences once the program is created and the documentation and related data structures are designed. Software testing essential for correcting errors. Otherwise the program or the project is said to be not complete. Software testing is a process of checking whether the developed system is working according to the original objectives and requirements.

The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. After the coding phase, computer programs

are available that can be executed for testing purposes. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases.

SOFTWARE TESTING FUNDAMENTALS

Testing presents an interesting task for software engineers. Earlier in the software process, the engineer attempts to build software from an abstract concept to a tangible implementation.. The engineer creates the series of test cases that are intended to "demolish" the software that has been built.To test any program we need to have a description of its expected behavior and a method of determining whether the observed behavior conforms to the expected behavior for this we need a test -oracle. A test-oracle is a mechanism different from the program itself that can be used to check the correctness of the output of the program for the test cases. Human-oracles are human beings who mostly compute by hand what the output of the program should be.

Human oracles can make mistakes. So test -oracle is defined in the tool to automate testing and avoids mistakes.

- A successful test is one that uncovers an as yet undiscovered error.
- The testing objectives are summarized in the following three steps.
- Testing is a process of executing a program with the intent of finding an error.

TESTING PRINCIPLES

All the tests should be traceable to customer requirements. Tests should be planned long before testing begins, that is the test planning can bring as soon as the requirement model is complete.Testing should begin "in the small" and progress towards testing "in the large". The first planned and executed generally focus on individual program modules.As testing progresses, testing shifts focus

and attempts to find errors in integrated clusters of modules and ultimately in the entire system.

The number of path permutations for even a moderately sized program is exceptionally large. For this reason, it is possible to execute every combination of paths during testing. It is possible, however, to adequately cover program logic and to ensure that all conditions in the procedural design have been exercised. To be more effective, testing has the highest probability of finding errors.

The following are the attributes of the good test

- A good test has a high probability of finding an error.
- A good test is not redundant.
- A good test should be “best of breed”.
- A good test should be neither too simple nor too complex.

TESTING STRATEGIES

System testing is a stage of implementation which is aimed at ensuring that the system works accurately and efficient before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

The testing steps are

- Unit Testing
- IntegrationTesting
- ValidationTesting
- Output Testing
- User acceptanceTesting

LEVELS OF TESTING

- This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the

intended specifications and satisfies the client's requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

- By performing acceptance tests on an application, the testing team will reduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system

- This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined together is known as alpha testing. During this phase, the following aspects will be tested in the application

- This test is performed after alpha testing has been successfully performed. In beta testing, a sample of the intended audience tests the application. Beta testing is also known as **pre-release testing**. Beta test versions of software are ideally distributed to a wide audience on the web, partly to give the program a "real-world" test and partly to provide a preview of the next release.

9.3 TESTING PROCESS

9.3.1 Unit Testing

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as "Module Testing". The modules are tested separately. This testing is carried out during the programming stage itself. Unit testing specifies paths in the module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit.

Unit testing is done for the tokenization module by providing URL as input and got the tokenized array as output.The tokenization module is found to be working as per the requirements.

Unit testing is done for the padding module by providing a tokenized array as input and getting the padded array as output.The padding module is found to be working as per the requirements.

Unit testing is done for the input module by providing Padded URL as input and got the tensor array as output.The input module is found to be working as per the requirements.

9.3.2. Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing programstructure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets are conducted.

Tokenization and Padding modules are tested by providing URL as input and got the padded array as output.The integrated module is found to be working as per the requirements.

Padding and Input modules are tested by providing tokenized URL as input and got the tensor array as output.The integrated module is found to be working as per the requirements.

9.3.3. Validation Testing

The outputs that come out of the system are as a result of the inputs that go into the system. So, for the correct and the expected outputs the inputs that go the system should be correct and proper. System is tested with input as URL and got the classification as phishing or legitimate URL.

9.3.4. Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system underconsideration.

System is tested with input as URL and got the classification as phishing or legitimate URL.

9.3.5. System Testing

A system testing does not test the software but rather the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specifications, and system documentation.System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

9.3.6 PerformanceTesting

Performance testing is designed to test the run-time performance of software within the context of an integrated system. It requires both hardware and software instrumentation. It is often necessary to measure resource utilization in an exacting fashion.

Hence no extra computation is required by the system, it performs very well even on low end computers.

9.4 SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. This is the most crucial stage in achieving a new successful system and in giving confidence to the new system for the users that it will work efficiently.

Implementation of software refers to the final installation of the package in itsreal environment, to the satisfaction of the intended users and the operations of the system. In many organizations someone who will not be operating it,will commission the software development project. The people who are not sure that the software is meant to make their job easier. In the initial stage they doubt their software but we have to ensure that the resistance does not build up as

- The active user must be aware of the benefits of using the system.
- Their confidence in the software is buildup.
- Proper guidance be imparted to the user so that he is comfortable in using the application.

The implementation procedures involve careful planning, investigation of the current system and the constraints on implementation, design of methods to achieve the changeover,an evaluation of change over methods.Initially a preliminary implementation plan is prepared to schedule and manage many different activities that must be completed for a successful system implementation.

The preliminary plan serves as a basis for the initial scheduling and assignment of resources to important implementation activities. The preliminary plan has been updated throughout the implementation phase in order to reflect the current state. A complete implementation plan includes the following items: selection of quality personnel, system training plan, system test plan, equipment installation pan, system conversion plan, and overall implementation plan. Apart from

planning, a major task of preparing the implementation procedures are education and training to the users. The more complex the system being implemented, the more involved be the system's analysis and design effort required just for implementation. An implementation coordinating committee based on policies of individual organizations is appointed.

The implementation process begins with preparing a plan for implementation of the system. According to this plan the activities have been carried out; discussions have been made regarding the equipment and resources. According to the above plan the necessary equipment has to be acquired to implement the new system.

To achieve the objective and benefits expected from a computer based system it is essential for the people who will be involved then in understanding the overall system and its effect on the organization, and in being able to carry out effectively their specific tasks. As systems become more complex the need for education and training is more and more important.

Training the user is one of the most important jobs of the developer for this purpose system and user manuals were prepared. In system manuals, details about the system, which were used to develop, were specified.

In user manuals, data flow diagrams, menu and screen formats are given. The user for the system is shown the screens and they are taught how to operate the system.

9.5 SYSTEM MAINTENANCE

The maintenance phase of the software cycle is the time in which a software product performs useful work. After the system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle.

CHAPTER 10

SCREENSHOTS

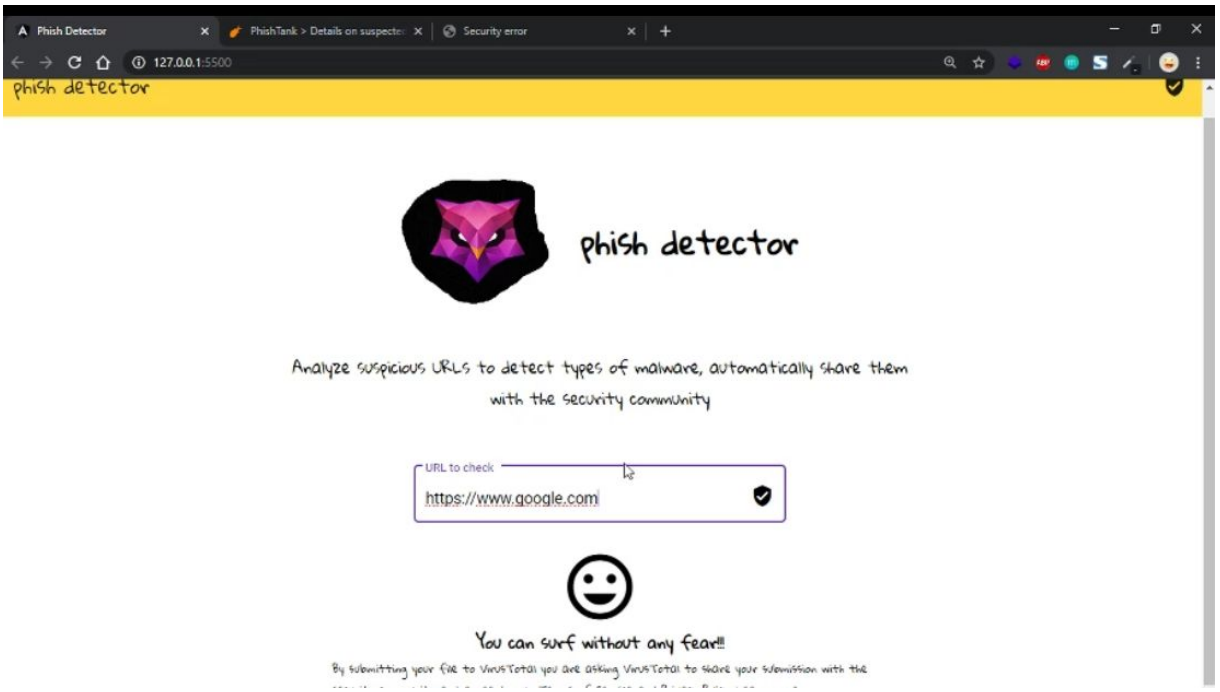


Fig 10.1: Detection of Legitimate Website

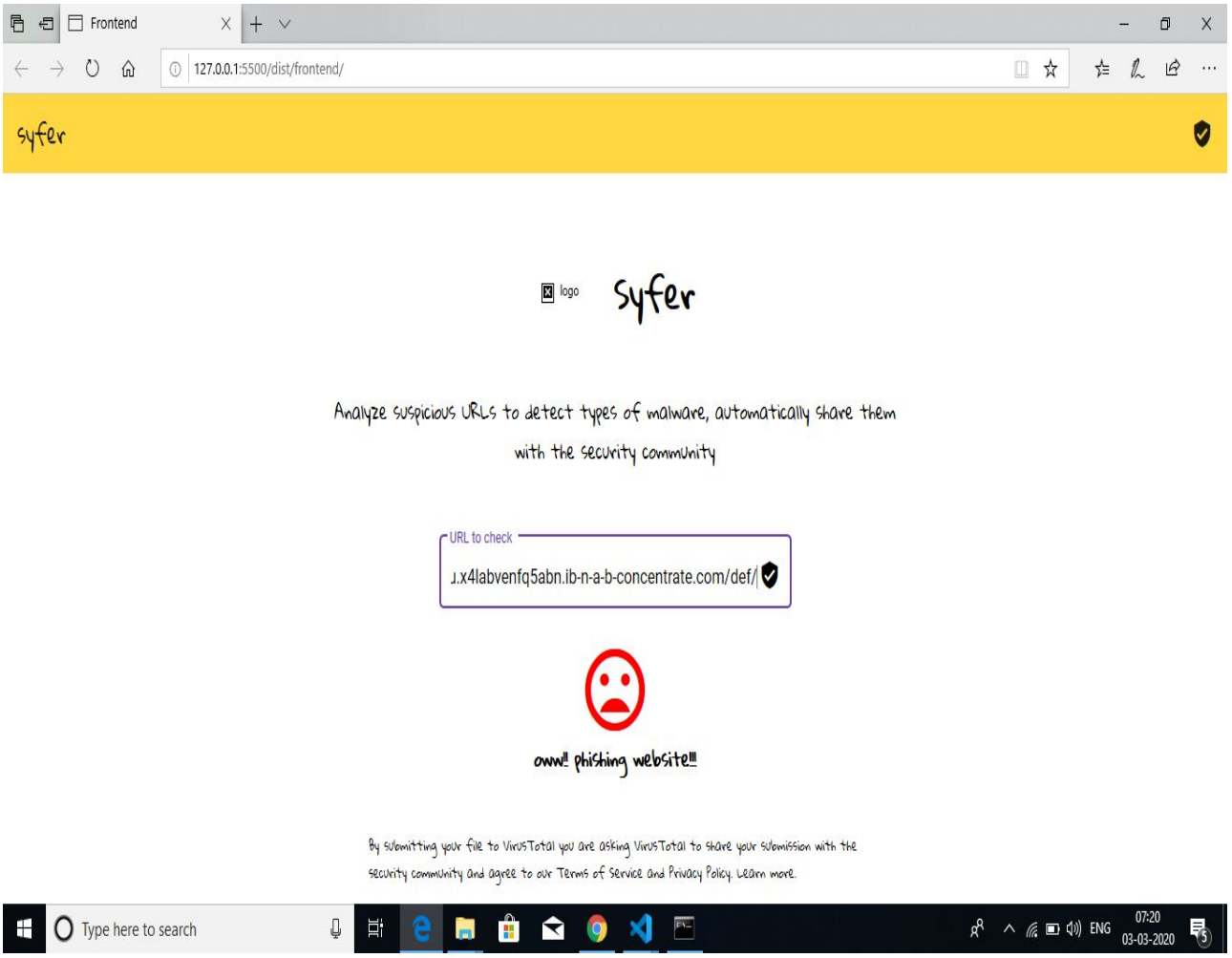


Fig 10.2: Detection of Phishing Website

CHAPTER 11

CONCLUSION AND FUTURE ENHANCEMENT

11.1 CONCLUSION

As a result, conclude our paper with accuracy of 97.33% and combination of 20 features. that is an automatic gadget gaining knowledge of approach that rely upon traits of phishing URL homes to stumble on and save you phishing web sites and to make certain excessive stage safety. The classification is accomplished in the use of Deep gaining knowledge of Methodologies, Random wooded area and Naive Bayes classifiers. As a future work the equal method is used to expand a device, based on an internet browser add-on component which could stumble on and prevent phishing websites on real time further to, imposing records mining techniques to discover new patterns of phishing URL.

11.2 FUTURE ENHANCEMENT

For future work, the phish detector can be moreover improved by adding more data sets and by using 1D CNN which is great for classifying texts.The idea of this project can be further developed with using Browser based Models like Tensorflow JS deep learning models which results in better response times even in poor network connections.

REFERENCES

[1] F. D. Abdi and L. Wenjuan, “Malicious url detection using convolutional neural networks,” Journal International Journal of Computer Science, Engineering and Information Technology, vol. 7, no. 6, pp. 1–8, 2017.

[2] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, “A comparison of machine learning techniques for phishing detection,” in Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit, ser. eCrime ’07. New York, NY, USA: ACM, 2007, pp. 60–69.

[3] M. Babagoli, M. P. Aghababa, and V. Solouk, “Heuristic nonlinear regression strategy for detecting phishing websites,” Soft Computing, Feb 2018.

[4] E. Buber, B. Dırı, and O. K. Sahingoz, “Detecting phishing attacks from url by using nlp techniques,” in 2017 International Conference on Computer Science and Engineering (UBMK), Oct 2017, pp. 337–342.

[5] D. L. Cook, V. K. Gurbani, and M. Daniluk, “Phishwish: A stateless phishing filter using minimal rules,” in Financial Cryptography and Data Security, G. Tsudik, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 182–186.

[6] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, “Defending against phishing attacks: taxonomy of methods, current issues and future directions,” Telecommunication Systems, vol. 67, no. 2, pp. 247–267, Feb 2018.

[7] Gupta, Rajendra, and Piyush Kumar Shukla."Performance Analysis of Anti-Phishing Tools andStudy of Classification Data Mining Algorithms for aNovel Anti-Phishing System." International Journal ofComputer Network and Information Security(IJCNIS) 7.12 (2015): 70.

[8] A. K. Jain and B. B. Gupta, “A novel approach to protect against phishing attacks at client side using auto-updated white-list,” EURASIP Journal on

Information Security, vol. 2016, no. 1, p. 9, May 2016.

[9] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: A literature survey,” IEEE Communications Surveys Tutorials, vol. 15, no. 4, pp. 2091–2121, Fourth 2013.

[10] E. Kirda, C. Kruegel, ”Protecting users against phishing attacks with AntiPhish”

[11] R. M. Mohammad, F. Thabtah, and L. McCluskey, “Predicting phishing websites based on self-structuring neural networks,” Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, Aug 2014.

[12] Muhammet Baykara , Zahit Ziya Gürel, “Detection of phishing attacks”

[13] H. H. Nguyen and D. T. Nguyen, ”Machine Learning Based Phishing Web Sites Detection,” in AETA 2015: Recent Advances in ElectricalEngineering and Related Sciences, V. H. Duy, T. T Dao, I. Zelinka, H.- S. Choi, and M. Chadli, Eds.Cham: Springer International Publishing, 2016, pp.123-131.

[14] M. Nguyen, T. Nguyen, and T. H. Nguyen, “Deep learning model with hierarchical lstms and supervised attention for anti-phishing,” arXiv preprintarXiv: 1805.01554, 2018.

[15] T. Peng, I. Harris, and Y. Sawa, “Detecting phishing attacks using natural language processing and machine learning,” in 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Jan 2018, pp. 300–301.

[16] Sananse, Bhagyashree E., and Tanuja K. Sarode."Phishing URL Detection: A Machine Learning andWeb Approach." International Journal of ComputerApplications 123.13 (2015).

[17] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, “Who falls for phish?: A demographic analysis of phishing susceptibility and effectiveness of interventions,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 373–382.

[18] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An

Empirical Analysis of Phishing Blacklists," 7 2009.

[19] T. Shibahara, K. Yamanishi, Y. Takata, D. Chiba, M. Akiyama, T. Yagi, Y. Ohsita, and M. Murata, "Malicious url sequence detection using event denoising convolutional neural network," in 2017 IEEE International Conference on Communications (ICC). IEEE, 2017, pp. 1–7.

[20] Shruti Ashok Mandake R. H. Goudar, "Detection and Prevention of Phishing Attack: An Approach for Eradication of Phishing "

[21] A. Stone, "Natural-language processing for intrusion detection," Computer, vol. 40, no. 12, pp. 103–105, Dec 2007.

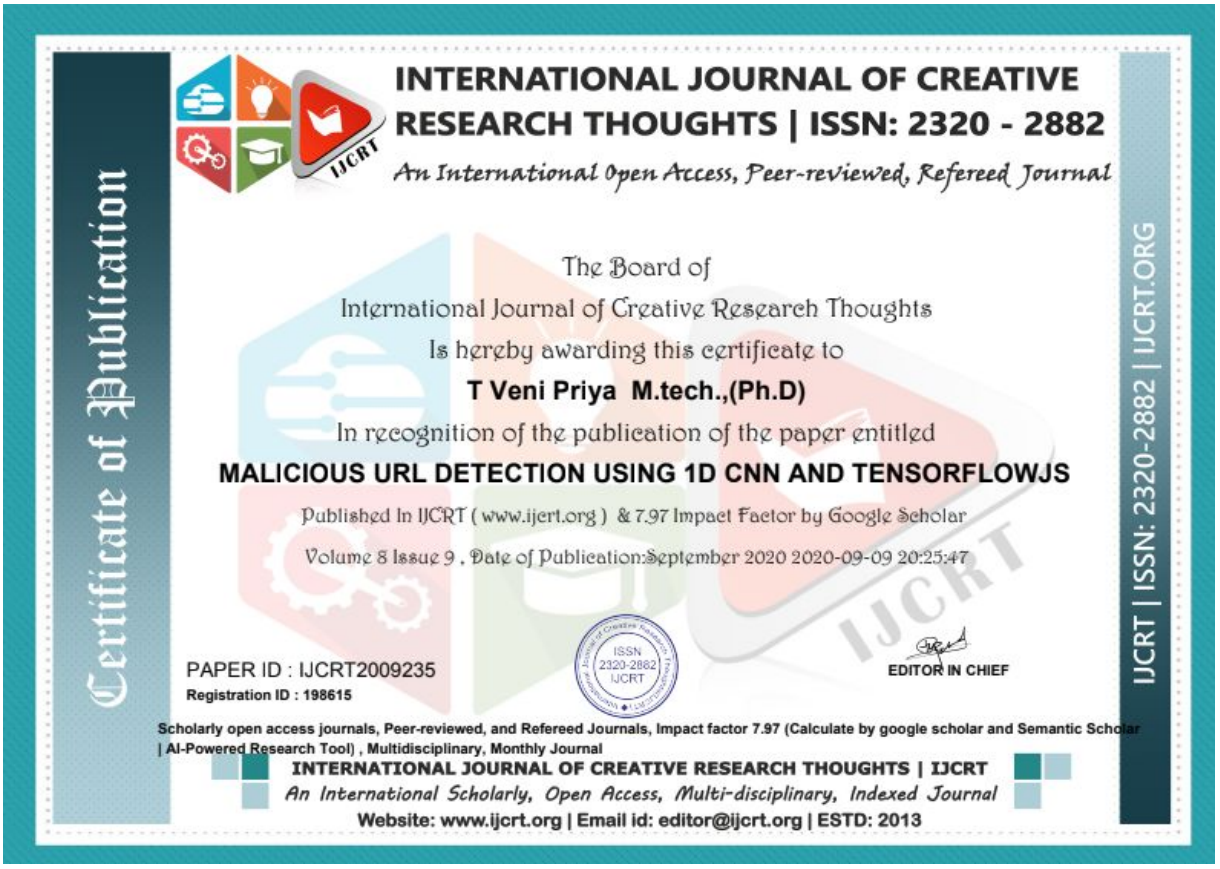
[22] M. A. U. H. Tahir, S. Asghar, A. Zafar, and S.Gillani, "A Hybrid Model to Detect 76 Phishing-Sites Using Supervised Learning Algorithms," in the 2016International Conference on Computational Science and Computational Intelligence (CSCI), 2016, pp.1126-1133.

[23] Ying Pan , Xuhua Ding, "Anomaly Based Web Phishing Page Detection"

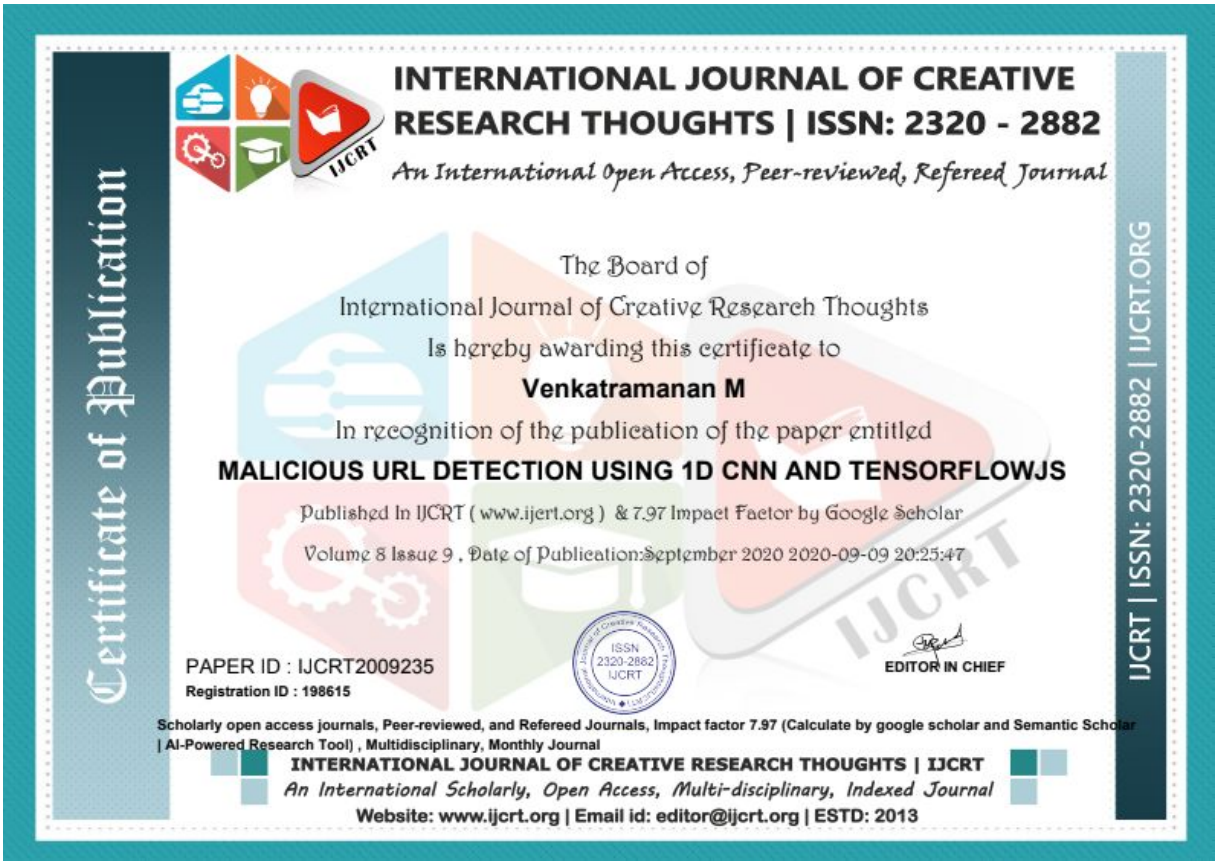
[24] Yongjie Huang, Qiping Yang, Jinghui Qin, Wushao Wen, "Phishing URL Detection via CNN and Attention-Based Hierarchical RNN"

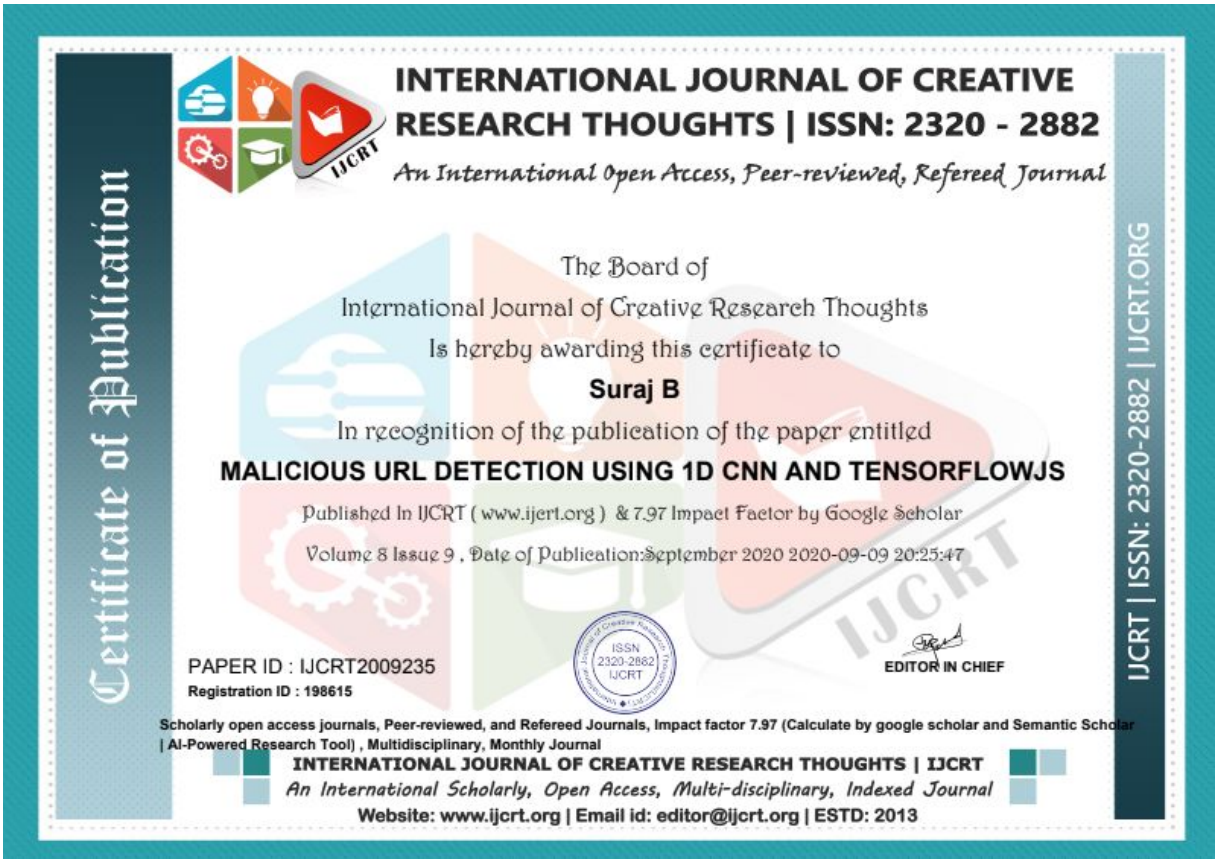
[25] W. Zhang, Y.-X. Ding, Y. Tang, and B. Zhao,"Malicious web page detection based on on-linelearning algorithm," in Machine Learning andCybernetics (ICMLC), 2011 International conference on, vol. 4. IEEE, 2011, pp. 1914–1919.

[26] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in In Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, p. 639–648.


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
T Veni Priya M.tech.,(Ph.D)
In recognition of the publication of the paper entitled
MALICIOUS URL DETECTION USING 1D CNN AND TENSORFLOWJS
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-09 20:25:47
PAPER ID : IJCRT2009235
Registration ID : 198615
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
IJCRT | ISSN: 2320-2882 | IJCRT.ORG
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
Harish S
In recognition of the publication of the paper entitled
MALICIOUS URL DETECTION USING 1D CNN AND TENSORFLOWJS
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-09 20:25:47
PAPER ID : IJCRT2009235
Registration ID : 198615
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
Venkatramanan M
In recognition of the publication of the paper entitled
MALICIOUS URL DETECTION USING 1D CNN AND TENSORFLOWJS
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-09 20:25:47
PAPER ID : IJCRT2009235
Registration ID : 198615
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
Suraj B
In recognition of the publication of the paper entitled
MALICIOUS URL DETECTION USING 1D CNN AND TENSORFLOWJS
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 8 Issue 9 , Date of Publication:September 2020 2020-09-09 20:25:47
PAPER ID : IJCRT2009235
Registration ID : 198615
ISSN 2320-2882 IJCRT
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG

E.G.S. PILLAY ENGINEERING COLLEGE
(AUTONOMOUS)
NAGAPATTINAM - 611002

INTERNATIONAL WEB CONFERENCE ON ADVANCES IN SCIENCE, ENGINEERING AND MANAGEMENT (ICASEM - 2020)
3rd JULY 2020

Certificate

This is to certify that Harish S, Venkataramanan M, Suraj B, from Arasu engineering college has Presented the paper titled "Malicious URL detection using 1d cnn" in the "International Web Conference On Advances In Science, Engineering And Management (ICASEM - 2020)" organized by E.G.S. PILLAY ENGINEERING COLLEGE (AUTONOMOUS) NAGAPATTINAM on 03.07.2020.

Convener
(Dr. G. Ganesan @ Subramanian)

Principal
(Dr. S. Ramabalan)

Certificate number XWUMNW-CE000121