IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Phishing URL Identification: An Actual Situation Using Login URLs

Putthuru Hema[1], Mr. Dharmaiahvari Prasad[2]

[1]PG Scholar, Master of Computer Applications, VEMU Institute of Technology, P. Kothakota, hemaputtur99@gmail.com

[2] *Assistant Professor, Dept.t of CSE, VEMU Institute of Technology, P. Kothakota*

***Abstract*— The likelihood and severity of network information insecurity are now rising quickly. The methods mostly used by hackers today are to attack end-to-end technology and exploit human vulnerabilities. These techniques include social engineering, phishing, pharming, etc. One of the steps in conducting these attacks is to deceive users with Phishing Uniform Resource Locators (URLs). As results, Phishing URL detection is of great interest nowadays. There have been several scientific studies showing a number of methods to detect Phishing URLs based on machine learning. In this project propose a Phishing URL detection method using machine learning techniques based on our proposed URL behaviours and attributes. Moreover, bigdata technology is also exploited to improve the capability of detection Phishing URLs based on abnormal behaviours. To put it briefly, the proposed detection method is made up of big data technologies, a machine learning algorithm, and a new collection of URL properties and behaviours. The experimental results show that the proposed URL attributes and behaviour can help improve the ability to detect Phishing URL significantly. This is suggested that the proposed system may be considered as an optimized and friendly used solution for Phishing URL detection.**

***Keywords*—URL, Machine Learning, Phishing, Big Data**

I. INTRODUCTION

It is highly challenging to counteract phishing attacks since they prey on user vulnerabilities, yet improving phishing detection methods is crucial. The "blacklist" method, which is another name for the overall technique of identifying phishing websites, involves adding Internet Protocol (IP) addresses to the antivirus database and updating banned URLs. Attackers employ inventive methods to trick people into thinking the URL is authentic through obfuscation in order to avoid being added to blacklists. Other basic strategies include fast-flux, which generates proxies automatically in order to host the webpage, algorithmic creation of new URLs, and many others. Nowadays Phishing Websites becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of Phishing attack. Main aim of the attacker is to steal banks account credentials. In United States businesses, there is a loss of US$2billion per year because their clients become victim to phishing. In 3rd Microsoft Computing Safer Index Report released in February 2014, it was estimated that the annual worldwide impact of phishing could be as high as $5 billion. Phishing attacks are becoming successful because lack of user awareness. Since Phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques.

.

Problem Identification

Phishing URL detection using machine learning is a significant problem in the field of cyber security. It aims to develop an efficient and accurate system that can identify and classify URLs as either benign or Phishing. The problem statement can be summarized as follows: Given a set of URLs and their associated features, the goal is to build a machine learning model that can effectively distinguish between Phishing and benign URLs. The model should be trained on a labelled dataset containing both Phishing and benign URLs, and it should be capable of generalizing to new, unseen URLs during the testing or deployment phase. Phishing URL detection is a crucial task in cybersecurity, as it helps protect users from phishing attacks, malware distribution, and other Phishing activities that leverage compromised or deceptive URLs.

Objectives

- To aanalysis Phishing URL detection method using machine learning techniques based on our proposed URL behaviours and attributes. Moreover, bigdata technology is also exploited to improve the capability of detection Phishing URLs based on abnormal behaviours.
- To suggest that the proposed system may be considered as an optimized and friendly used solution for Phishing URL detection.
- In short, the proposed detection system consists of a new set of URLs features and behaviours, a machine learning algorithm, and a bigdata technology. The experimental results show that the proposed URL attributes and behaviour can help improve the ability to detect Phishing URL significantly.

II. MODE OF PYTHON-PLATFORM

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications.

A. *Web frameworks*

A web framework is a code library designed to facilitate the development of dependable, expandable, and easily maintained web applications for developers. Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Frameworks facilitate the reuse of code for frequently used HTTP operations and the organization of projects so that other developers who are familiar with the framework may rapidly construct and manage the application compare-python-web-frameworks where the same web application is being coded with varying Python web frameworks, tinplating engines and object. Web framework resources:

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
- Frameworks is a really well-done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
- What is a web framework? Is an in-depth explanation of what web frameworks being and their relation to web servers?
- Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
- Python’s web frameworks benchmarks are a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the
- output is fun to read about nonetheless.
- What web frameworks do you use and why are they awesome? Is a language agnostic
- Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
- This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?” The votes aren't as important as the list of the many frameworks that are available to Python developers.

B. *Existing System*

An existing system surveys the literature on the detection of Phishing attacks. Phishing attacks target vulnerabilities that exist in systems due to the human factor. Many cyber attacks are spread via mechanisms that exploit weaknesses found in end-users, which makes users the weakest element in the security chain. The Phishing problem is broad and no single silver-bullet solution exists to mitigate all the vulnerabilities effectively, thus multiple techniques are often implemented to mitigate specific attacks. aims at surveying many of the recently implemented Phishing mitigation techniques. A high-level overview of various categories of phishing mitigation techniques is also presented, such as: detection, offensive defence, correction, and prevention, which we belief is critical to present where the phishing detection techniques fit in the overall mitigation process.

C. *Proposed System*

Presence of IP address in URL: If IP address present in URL, then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that attacker is trying to steal sensitive information. Presence of @ symbol in URL: If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add special symbol @ in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol. Number of dots in Hostname: Phishing URLs have many dots in URL. Prefix or Suffix separated by (-) to domain: If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example, Actual site is http://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.com to confuse the innocent users. HTTPS token in URL: If HTTPS token present in URL, then the feature is set to 1 else to0. Phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example, http://https-wwwpaypal-it-mpp-home.soft-hair.com.
Information submission to Email: Phisher might use “mail()” or “mailto:” functions to redirect the user’sin formation to his personal email[4]. If such functions are present in the URL, then feature is set to 1 else to 0. URL Shortening Services “Tiny URL”: Tiny URL service allows phisher to hide long phishing URL by making it short. The goal is to redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0 Length of Host name: Average length of the benign URLs is found to be a 25, If URLs length is greater than 25 then the feature is set to 1 else to 0. Presence of sensitive words in URL: Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that found in many phishing URLs: - 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'sign in', 'mail', 'install', 'toolbar', 'backup', 'paypal', 'password', 'username', etc;

III. SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements

A. *UML DIAGRAMS*

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express

the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

B. USE CASE DIAGRAM

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal.

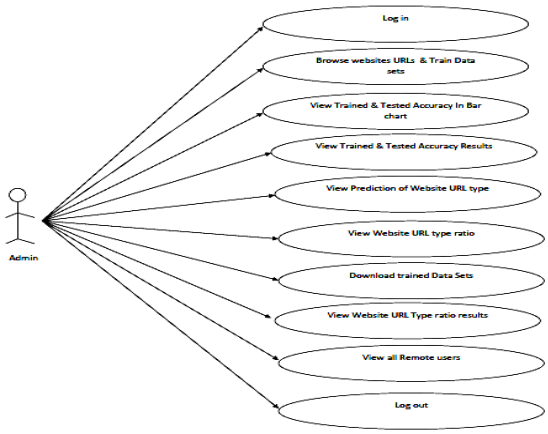

Fig. 1 Use case diagram of user/admin

C. CLASS DIAGRAM

UML class diagrams model static class relationships that represent the fundamental architecture of the system. Note that these diagrams describe the relationships between classes, not those between specific objects instantiated from those classes. Thus, the diagram applies to all the objects in the system.

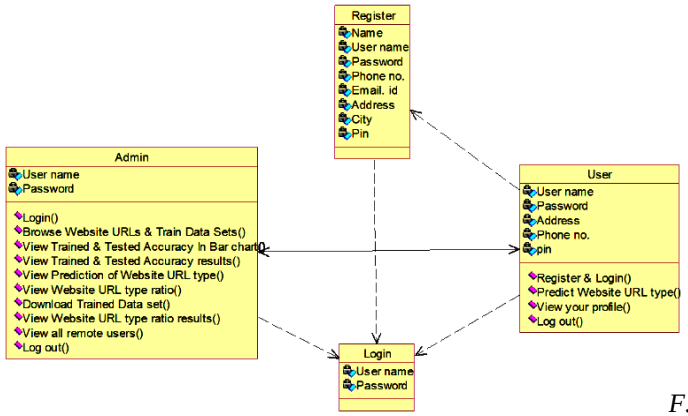

Fig. 2 Class diagram for Facial Expression

D. SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behaviour of a system. The sequence diagram is an element that is used primarily to showcase the interaction that occurs between multiple objects. This interaction will be shown over certain period of time. Because of this, the first symbol that is used is one that symbolizes the object. Objects will also be given the ability to call methods upon themselves, and they can add net activation boxes.

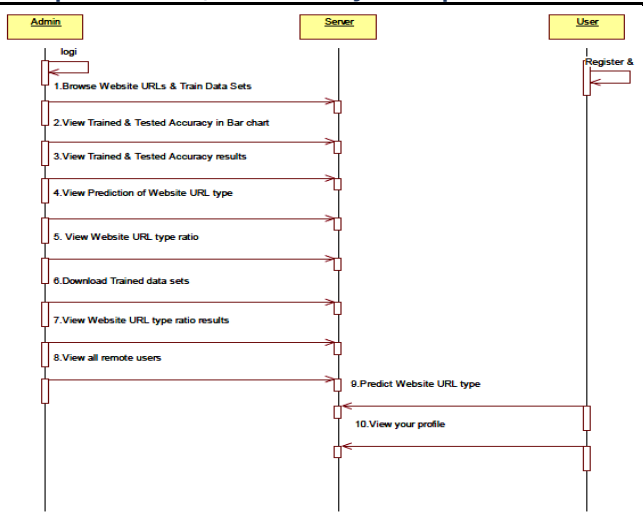

Fig. 3 Sequence diagram for Phishing Detection

E. ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

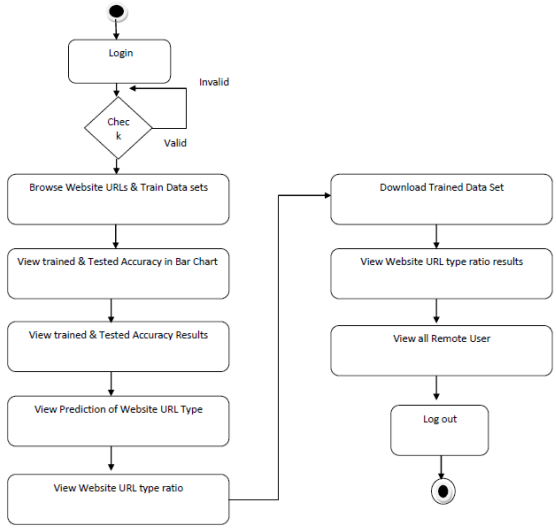

Fig. 4 Activity diagram for Facial Expression

F. System Testing

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

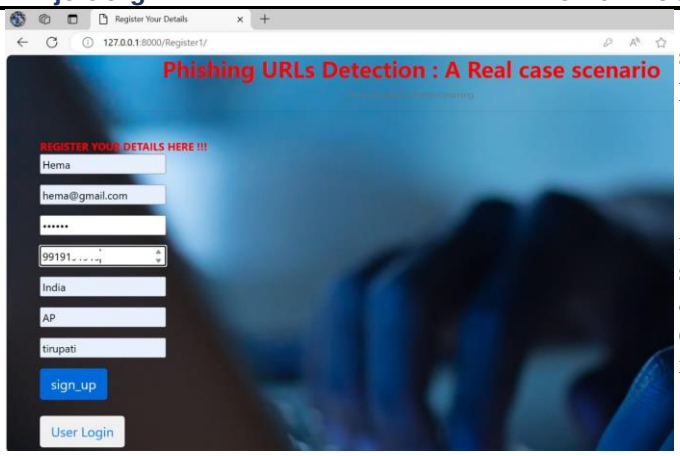

Fig. 5 Registration of the Phasing URLs Detection

Functional tests offer methodical proof that the functions being tested are available in accordance with the technical and business requirements, system documentation, and user manuals. Invoking interface systems or procedures is necessary.

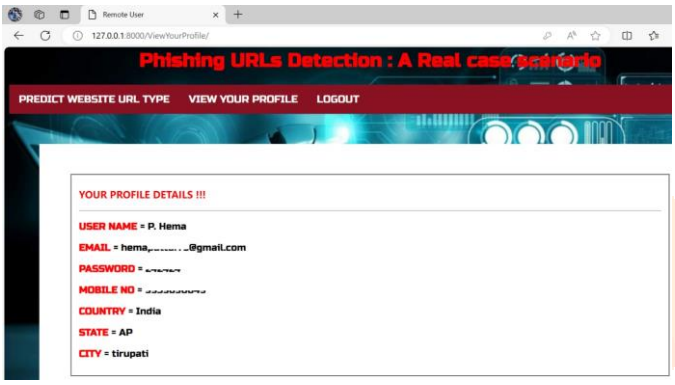

Fig. 6 Remote user

Functional test preparation and organization are centred on requirements, important features, or unique test cases. Furthermore, testing needs to take into account data fields, specified procedures, sequential processes, and systematic coverage related to identifying business process flows. Additional tests are identified and the efficacious value of current tests is ascertained prior to the completion of functional testing.

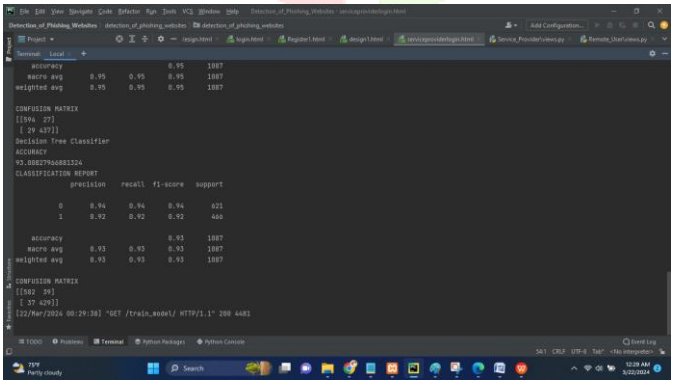

Fig. 7 Evaluation of training data

Each machine learning model underwent multiple Training and Testing Iterations, where it was trained on the designated training set and rigorously evaluated on the reserved test set. This iterative process ensured a thorough understanding of each model's behaviour and predictive capabilities.

In the critical phase of Model Training and Testing, our methodology prioritized effective evaluation strategies and model selection. First, the dataset underwent meticulous Data Splitting, with a division into training (75%) and testing (25%) sets. This separation facilitated the evaluation of model performance on unseen data

IV. Results and Discussions

The results are derived incorporated Manual Testing. This feature allowed real-time user input enabling practical testing scenarios. Users could input for immediate evaluation using all trained models. This real-time user interaction not only enhanced the user experience but also offered crucial insights into the models' performance on diverse inputs.

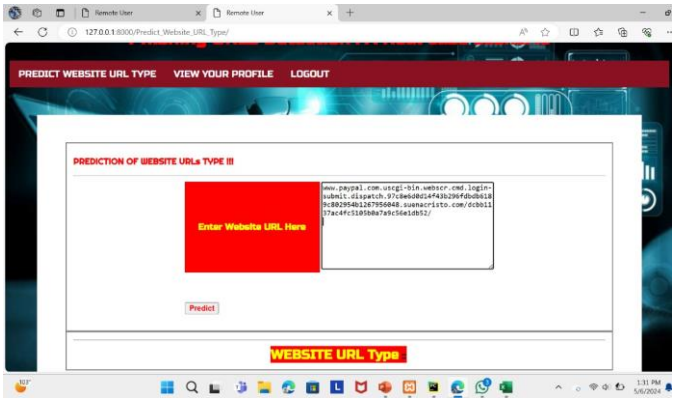

Fig. 8 Prediction of Website

The approach to Model Training and Testing emphasizes a comprehensive evaluation, model comparison, and real-world applicability through manual testing. This robust methodology ensures the selection of a highly effective algorithm for detection, grounded in thorough analysis and user engagement.

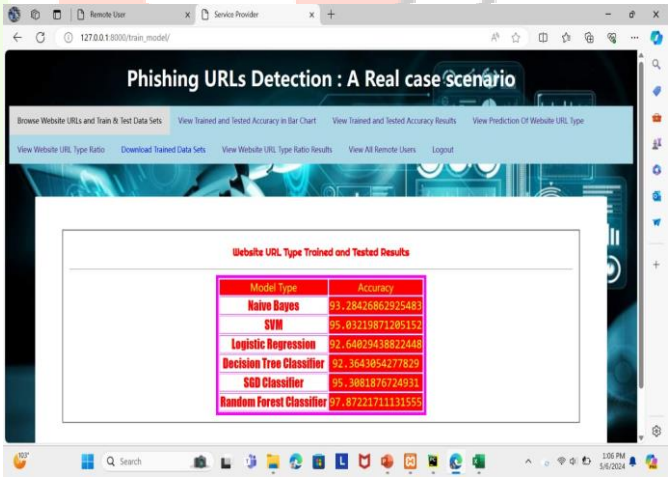

Fig. 7 Training and test results

The process of designing test cases for unit testing ensures that the core logic of the program is operating correctly and that program inputs result in legitimate outputs. Validation should be done on all internal code flows and decision branches. It is the testing of the application's separate software components. Prior to integration, it is completed following the conclusion of a single unit. This is an intrusive structural test that depends on an understanding of its structure. Unit tests evaluate a particular application, system configuration, or business process at the component level.

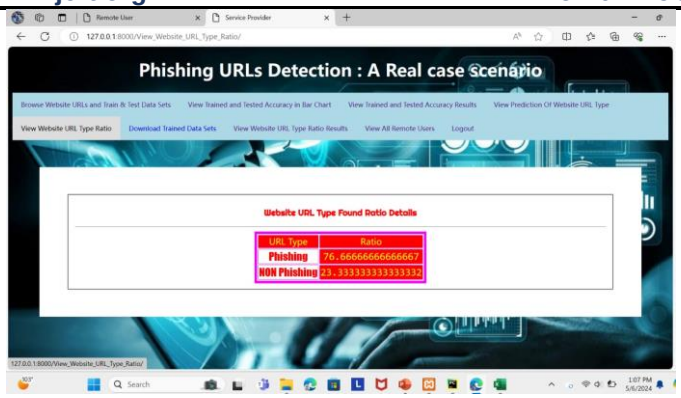

Fig. 9 Image to test of resolutions

The study opens the door for further developments in this important field by providing a comparative analysis of machine learning models and insightful information about the state of phasing detection

V. CONCLUSIONS

This project aims to enhance detection method to detect Phishing websites using machine learning technology. achieved good detection accuracy using Random Forest algorithm with lowest false positive rate. Also result shows that classifiers give better performance when we used more data as training data. The detection models leverage diverse algorithms, including Logistic Regression, Decision Tree, Gradient Boosting, and SVM, to effectively identify and classify phishing. These models integrate the expertise of machine learning specialists with the ability to extract significant features indicative of detection. The evaluation results showcase remarkable accuracy for each algorithm, with Decision Tree and Gradient Boosting models achieving particularly high accuracy levels. The trend in employing different machine learning algorithms for detection is evident, emphasizing continuous efforts to enhance accuracy and effectiveness compared to conventional methods.

In future hybrid technology will be implemented to detect phishing websites more accurately, for which random forest algorithm of machine learning technology and blacklist method will be used.

REFERENCES

[1] Priyanshu Garg, Priyanshu Sharmab, P (2024) Fake News Detection Using Machine Learning Algorithms, Tuijin Jishu/Journal of Propulsion Technology, vol. 45, No.2, 1001-4055

[2] C. Buntain and J. Golbeck, "Automatically Identifying Fake News in Popular Twitter Threads," 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017, pp. 208-215, DOI: 10.1109/SmartCloud.2017.40.

[3] Gupta and R. Kaushal, “Improving spam detection in Online Social Networks,” 2015 International Conference on Cognitive Computing and Information Processing (CCIP), 2015, pp. 1-6, DOI: 10.1109/CCIP.2015.7100738.

[4] Mahmoud Kohji, Youssef Iraqi, "Phishing Detection: A Literature Survey IEEE, andAndrew Jones, 2013

[5] Gunter Ollmann, “The Phishing Guide Understanding & Preventing Phishing Attacks”, IBMInternet Security Systems, 2007.