**A**

**FIELD BASED PROJECT REPORT**

**on**

|  |
| --- |
| **PHISHING URL DETECTION USING MACHINE LEARNING** |

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING (AIML)**

**Submitted by**

**(BATCH :MIP-9)**

**Student Name:**

**K.Gowthami(227Y1A67E6)**

**Md.Noman Malik(227Y1A67G1)**

**Under the Guidance**

**of**

**Dr. N.Pushpa Latha**

**Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(AIML)**

**MARRI LAXMAN REDDY**

**INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(AUTONOMOUS)**

**JUNE 2024**

****

|  |  |
| --- | --- |
| **CERTIFICATE** | |
| This is to certify that the project report titled “phishing url detection using ml**”** is being submitted by **Prajwal Nilopant (207Y1A0xxxxx)** in **II B.Tech II Semeste**r **Computer Science & Engineering(AIML)**is a record bonafide work carried out by him. The results embodied in this report have not been submitted to any other University for the award of any degree. | |
|  |  |
| **Internal Guide** | **HOD** |
|  |  |
| **Principal** |  |

****

**DECLARATION**

|  |  |
| --- | --- |
| I hereby declare that the Major Project Report entitled, “**Phising Url Detection Using Machine Learning”**submitted for the B.Tech degree is entirely my work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree. | |
|  | |
| **Date:** | |
|  | |
|  |  |
| **K.Gowthami(227Y1A67E6)**  **Md.Noman Malik(227Y1A67G1)** |  |

****

**ACKNOWLEDGEMENT**

|  |
| --- |
| I am happy to express my deep sense of gratitude to the principal of the college **Dr. R. Murali Prasad,** Professor, Marri Laxman Reddy Institute of Technology & Management, for having provided me with adequate facilities to pursue my project.  I would like to thank **Dr N.Pushpa Latha,** Professor and Head, Department of Computer Science and Engineering(DS), Marri Laxman Reddy Institute of Technology & Management, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.  I am very grateful to my project guide **Dr N.Pushpa Latha,** Professor and Head, Department of Computer Science and Engineering(DS), Marri Laxman Reddy Institute of Technology & Management, for his extensive patience and guidance throughout my project work.  I sincerely thank my seniors and all the teaching and non-teaching staff of the Department of Computer Science for their timely suggestions, healthy criticism and motivation during the course of this work.  I would also like to thank my classmates for always being there whenever I needed help or moral support.With great respect and obedience, I thank my parents and brother who were the backbone behind my deeds.  Finally, I express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to my need at right time for the development and success of this work. |

****

**CONTENTS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S NO.** | |  | **TITLE** | **PAGE NO.** | | |  |
|  | | ABSTRACT | | **Vi** | | |  |
|  | | LIST OF FIGURES | | **Vii** | | |  |
|  | | LIST OF TABLES | | **Viii** | | |  |
|  | | SYMBOLS & ABBREVIATIONS | | **IX** | | |  |
| **1** | | **INTRODUCTION** | | **1** | | |  |
|  | | 1.1 | Motivation | 5 | | |  |
|  | | 1.2 | Problem | 6 | | |  |
|  | | 1.3 | Solution | 6 | | |  |
|  | | 1.4 | Scope | 6 | | |  |
|  | | 1.5 | Problem Definition | 7 | | |  |
|  | | 1.6 | Objective | 7 | | |  |
|  | | 1.7 | Limitations | 7 | | |  |
|  | | 1.8 | Organization Of Document | 7 | | |  |
| **2** | | **LITERATURE SURVEY** | | **8** | | |  |
|  | | 2.1 | Farm Monitoring System | 8 | | |  |
|  | | 2.2 Overview | | 10 | | |  |
|  | | 2.3 | Existing System | 11 | | |  |
|  | | 2.4 | Disadvantages of Existing System | 11 | | |  |
|  | | 2.5 Proposed System | | 11 | | |  |
|  | | 2.6 Advantages of Proposed System | | 12 | | |  |
|  | | 2.7 Summary | | 12 | | |  |
| **3** | | **ANALYSIS** | | **13** | | |  |
|  | | 3.1 | Introduction | 13 | | |  |
|  | | 3.2 | Software Requirement Specification | 13 | | |  |
|  | | 3.3 | Content Diagram | 19 | | |  |
|  | | 3.4 Algorithms and Flowcharts | | 20 | | |  |
|  | | 3.5 | Summary | 21 | | |  |
| **4** | | **DESIGN** | | **22** | | |  |
|  | | 4.1 | Introduction | 22 | | |  |
|  | | 4.2 | System Models | 23 | | |  |
|  | | 4.3 | Module Design and Organization | 28 | | |  |
|  | | 4.4 | Summary | 32 | | |  |
| **5** | | **IMPLEMENTATION AND RESULT** | | **33** | | |  |
|  | | 5.1 | Introduction to Arduino Microcontroller | 34 | | |  |
|  | | 5.2 | Software tools | 37 | | |  |
|  | | 5.3 ThingSpeak cloud registration | | 38 | | |  |
|  | | 5.4 | Execution | 39 | | |  |
|  | | 5.5 | Summary | 39 | | |  |
| **6** | | **TESTING AND VALIDATION** | | **40** | | |  |
|  | | 6.1 | Introduction | 40 | | |  |
|  | 6.2 | | Design of Test Cases Scenarios & Validation | | 41 |
|  | 6.3 | | Summary | | 42 |
| **7** | **CONCLUSION AND FUTURE ENHANCEMENTS** | | | | **43** |
| **8** | **REFERENCES** | | | | **44** |
|  | **APPENDIX A** | | | | **45** |
|  | **APPENDIX B** | | | | **51** |
|  | **APPENDIX C** | | | | **54** |

**ABSTRACT**

|  |
| --- |
| With raising in-depth amalgamation of the Internet and social life, the Internet is  looking differently at how people are learning and working, meanwhile opening us to  growing serious security attacks. The ways to recognize various network threats, specifi-  cally attacks not seen before, is a primary issue that needs to be looked into immediately.  The aim of phishing site URLs is to collect the private information like user’s identity,  passwords and online money related exchanges. Phishers use the sites which are visibly  and semantically like those of authentic websites. Since the majority of the clients go  online to get to the administrations given by the government and money related orga-  nizations, there has been a vital increment in phishing threats and attacks since some  years.  As technology is growing, phishing methods have started to progress briskly and this  should be avoided by making use of anti-phishing techniques to detect phishing. Machine  learning is a authoritative tool that can be used to aim against phishing assaults. There  are several methods or approaches to identify phishing websites.  The machine learning approaches to detect phishing websites have been proposed earlier  and have been implemented. The central aim of this project is to implement the system  with high efficiency, accuracy and cost effectively. That is been achieved. The project is  implemented using 4 machine learning supervised classification models. The four classi-  fication models are K-Nearest Neighbor, Kernel Support vector machine, decision tree  and random forest classifier. It was established that the Random forest classifier provides  best accuracy for the selected dataset and gives an accuracy score of 96.82%. |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG. NO** | **FIG. NAME** | **PAGE NO.** |
| 1.1 | Internet Of Things | 4 |
| 1.2 | Wireless Sensor Network (WSN) | 5 |
| 3.1 | ArduinoAtmega 328(UNO) | 15 |
| 3.2 | Arduino Ethernet Shield | 16 |
| 3.3 | ML8511 UV Sensor | 16 |
| 3.4 | DHT11 Temperature and Humidity Sensor | 17 |
| 3.5 | MQ3 Gas Sensor | 17 |
| 3.6 | Water Level Sensor | 18 |
| 3.7 | 5V Solar Panel | 18 |
| 3.8 | Rechargeable Li Ion Battery | 19 |
| 3.9 | Architecture Of The Setup | 19 |
| 3.10 | Flow Chart Of The System | 20 |
| 4.1 | Essential Use Case Diagram | 24 |
| 4.2 | Class Diagram | 25 |
| 4.3 | Sequence Diagram | 26 |
| 4.4 | Activity Diagram | 27 |
| 4.5 | Collaboration Diagram | 27 |
| 4.6 | Smart Farm Using Renewable Energy Resources Circuit Diagram | 34 |
| 5.1 | Block Diagram | 33 |
| 5.2 | Arduino Microcontroller | 34 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TABLE TITLE** | **PAGE NO.** |
| 3.1 | Software Requirements | 14 |
| 3.2 | Hardware Requirements | 14 |
| 5.1 | Summary | 35 |
| 6.1 | Above prescribed (favorable) levels | 41 |
| 6.2 | Below Threshold Level Use Case | 42 |

**SYMBOLS & ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **IDE** | **:** | Integrated Development Environment |
| **IOT** | **:** | Internet Of Things |
| **I/O** | **:** | Input and Output |
| **LED** | **:** | Light Emitting Diode |
| **SMS** | **:** | Simple Message Service |
| **UML** | **:** | Unified Modeling Language |
| **URL** | **:** | Uniform Resource Locator |
| **USB** | **:** | Universal Serial Bus |
|  |  |  |
|  |  |  |
|  |  |  |

1.INTRODCUTION

Artificial intelligence is a new innovative science that reviews and creates hypotheses, strategies, procedures, and applications that recreate, grow and broaden human knowledge. ML is an arm of artificial intelligence and it is analogous to (and frequently overlap with) computational measurements, [1] that also concentrates on making predictions with the use of PCs. Machine leaning has solid relationship with scientific improvement, which tells methods, hypothesis and utilization regions to the field. ML is sometimes, in a while combined with data mining [2], but the data mining subfield focuses more on preparatory information investigation and is called as unsupervised learning. ML can likewise be unsupervised and be utilized to learn and set up pattern profiles for various entities and then used to find important anomalies. [3].Cyber security is a set of innovations and procedures intended to secure PCs, networks ,projects and information from assaults and unapproved access, modification, or annihilation [4] A system security framework comprises of a system assurance framework and furthermore a PC protection framework. Every one of these frameworks incorporates firewalls, antivirus programming, and intrusion detection system (IDS). IDSs help find, decide and distinguish unapproved system conduct [5], for instance, use, replicating, change and annihilation .There are three important kind of network analysis for Intrusion detection system: misuse-

Introduction Detection of phishing websites using machine learning techniques based, also known as anomaly-based, signature-based, and hybrid.

misuse based detection strategies mean to distinguish realized attacks by utilizing the marks of these attacks.

• Anomaly-based methods study the typical system and its conduct and distinguish

anomalies as deviations from ordinary behavior.

• Hybrid detection conflates anomaly and misuse detection . It is utilized to expand

the rate of detection of accepted intrusions and to decrease the rate of false positives

of unknown attacks.

1.1MOTIVATION:

There are many Anti phishing techniques that helps us protect from phishing sites.

Mozilla Firefox, Safari and Google chrome makes use of Google Safe Browsing (GSB) service that will block the phishing websites. There are also many such tools like McFee Site Advisor, Quick Heal, Avast and Net craft which are widely used. GSB analyzes a URL by making use of the blacklist approach. The main disadvantage of GSB was that it was unable to detect the phishing website since updating of blacklist was not done. In case of Net craft, a website that phishing was recorded as phishing although it wasn't blocked. The blocking is done by Netcraft only when it is sure 100% that the website is phishing. The warning is given only when the user clicks the right button on the icon to find the risk rating. The risk is when the individual doesn’t check the rating or makes decision to use it after checking the rating. Security against security attacks online is provided by some soft wares like QuickHelp and Avast. The functioning of Avast anti-virus was checked after installing it. The Avast browser was not able to successfully find the phisher URL that was successfully determined by Net craft and GSB.

This above mentioned points accepts the necessity of anti phishing tools that are advanced in nature. It is noteworthy that these tools must be installed independently. A lay person might never install tools if he is not aware of practices like phishing. If that is the case, then people rely only on GSB service. Hence, the awareness considering such anti phishing tools and phishing is very important. Also, no individual should fully rely on tools because it is seen that they might lead to misclassification.

The motivation behind using machine learning (ML) for phishing URL detection stems from the need to combat increasingly sophisticated and prevalent cyber threats. Here are key motivations for employing ML in phishing URL detection:

### **1. Increasing Frequency and Sophistication of Phishing Attacks**

* **Rapid Evolution**: Phishing attacks evolve quickly, employing tactics like domain spoofing, URL obfuscation, and social engineering to deceive users.
* **Volume of Attacks**: The sheer volume of phishing attacks makes manual detection and response impractical, necessitating automated solutions.

### **2. Accuracy and Efficiency**

* **Enhanced Detection**: ML algorithms can analyze large datasets of URLs and learn patterns indicative of phishing behavior with high accuracy.
* **Real-time Detection**: ML models enable quick identification of phishing URLs, reducing response time to mitigate potential threats.

### **3. Adaptability to New Threats**

* **Dynamic Learning**: ML models can adapt to new phishing techniques and variations by continuously learning from updated datasets and evolving attack patterns.
* **Zero-day Detection**: ML can detect previously unseen phishing URLs or variations that evade traditional signature-based detection methods.

### **4. Scalability**

* **Handling Large Datasets**: ML techniques can efficiently process and analyze vast amounts of data, making them suitable for scaling to handle the ever-growing volume of URLs generated daily.
* **Automation**: ML allows for automated scanning and classification of URLs, freeing up human resources for more strategic cybersecurity tasks.

1.2 PROBLEM:

The problem is derived after making a thorough observation and study about the method

of classification of phishing websites that makes use of machine learning techniques. We

must design a system that should allow us to:

• Accurately and efficiently classify the websites into legitimate or phishing.

• Time consumed for detection should be less and should be cost effective.

1.3SOLUTION:

Detecting phishing URLs using machine learning involves several steps and considerations. Here’s a comprehensive approach to building a solution for phishing URL detection:

### **1. Problem Understanding and Dataset**

* **Problem Definition**: Detect phishing URLs with high accuracy to prevent users from accessing malicious websites.
* **Dataset**: Obtain or create a dataset with labeled examples of phishing and legitimate URLs. Ensure the dataset is balanced and representative of real-world scenarios.

### **2. Data Preprocessing**

* **Feature Extraction**:
  + **URL Features**: Extract features from URLs such as length, domain age, presence of special characters, etc.
  + **Domain Features**: Use WHOIS data, domain reputation scores, and IP address information.
  + **Content-based Features**: Analyze webpage content if available (not always possible due to privacy concerns).
* **Normalization**: Normalize and preprocess URLs
* 1.4 Scope

The focus of the project is on machine learning (ML) methods for network analysis of

intrusion detection especially phishing websites attack.

### **1. Data Collection and Preprocessing**

* **Automated Data Gathering**: Systems collect URLs from diverse sources such as web scraping, APIs, and security feeds.
* **Data Cleaning and Normalization**: Ensuring data consistency and preparing it for feature extraction..

### **2. Machine Learning Models**

* **Model Selection**: Choosing appropriate ML algorithms (e.g., logistic regression, random forest, neural networks) based on dataset characteristics and detection requirements.
* **Training**: Using labeled datasets to train models to distinguish between phishing and legitimate URLs.

### **3. Feature Extraction and Analysis**

* **URL Features**: Analyzing structural elements (e.g., URL length, domain structure) and content-based characteristics (e.g., presence of suspicious keywords).
* **Domain Reputation**: Utilizing external sources (e.g., blacklists, reputation databases) to evaluate domain trustworthiness.

**4. Real-Time Detection and Monitoring**

* **Deployment**: Integrating ML models into systems capable of real-time URL scanning and classification.
* **Monitoring**: Continuously monitoring system performance, detecting anomalies, and updating models to adapt to new phishing tactics.

### **5. Adaptability and Resilience**

* **Adapting to New Threats**: ML models capable of learning and adapting to emerging phishing techniques and zero-day attacks.
* **Feedback Loop**: Incorporating feedback from detected phishing URLs to improve model accuracy and response.

**1.5 PROBLEM DEFINITION:**

Define the goal clearly: Detect phishing URLs with a high level of accuracy using machine learning techniques.

* **Dataset Selection**:
* Choose or collect a dataset that includes labeled examples of phishing and legitimate URLs. The dataset should ideally be balanced to avoid biases.

**1.6 OBJECTIVE:**

The project’s objectives are as follows:

• To study various automatic phishing detection methods

• To identify the appropriate machine learning techniques and define a solution using

the selected method

• To select an appropriate dataset for the problem statement

• To apply appropriate algorithms to achieve the solution to phishing attacks.

**1.7LIMITATIONS:**

The challenges faced during the project are as follows:

• Finding the appropriate dataset.

• Feature extraction required the study of various modules and understanding each

module and getting the expected outcome from it.

**1.8 ORGANIZATION OF DOCUMENT:**

Chapter 1 incorporates a presentation about the application of ML in cyber security.

It details the problem statement, objectives and scope of the project. It also tells about the challenges faced during the development of the project. Chapter 2 incorporates the study and research about the phishing attacks and its detection using Machine learning techniques. It gives a detailed description of the earlier works done in this front and the limitations of those related works. discusses about the software and hardware requirements which is necessary for the system. The chapter details about the minimum requirements needed for the project and also about the modules of Python that are used. Tells about the system design and its representation using architecture, dataflow diagrams and activity diagram. It gives a graphical and diagrammatic representation of the system for better understanding and the system’s, user’s and run time perspective of the project. the implementation of this project is being examined. The chapter details about the dataset used, the steps involved in the implementation, the classifiers used, etc. the test cases are being examined and a comparison of the expected output and the actual output is being made to validate our result. In the outcome obtained and the environmental setup up of the project is

being discussed. I conclude the project in chapter 8 and also discuss about the future

enhancements to the project.

**2.LITERATURE SURVEY:**

**2.1 OVERVIEW:**

Phishing URL detection using machine learning (ML) involves leveraging algorithms and techniques to automatically identify malicious URLs designed to deceive users. Here's an overview of how this process works:

### **1. Problem Definition**

* **Objective**: Detect and classify URLs as either phishing (malicious) or legitimate based on their characteristics.
* **Challenge**: Phishing URLs are designed to mimic legitimate URLs, making manual detection challenging.

### **2. Data Collection and Preparation**

* **Dataset**: Gather a dataset consisting of labeled examples of phishing and legitimate URLs.
* **Features**: Extract relevant features from URLs, such as:
  + URL length
  + Presence of special characters
  + Domain

### **3. Feature Engineering**

* **Normalization**: Normalize URLs by converting them to a standardized format (e.g., lowercase, removing unnecessary characters).
* **Feature Selection**: Choose informative features that contribute most to distinguishing between phishing and legitimate URLs.
* **Dimensionality Reduction**: Use techniques like PCA (Principal Component Analysis) to reduce feature dimensionality if needed.

### **4. Model Selection and Training**

* **Classification Models**: Select appropriate ML algorithms for classification, such as:
  + Logistic Regression
  + Decision Trees
  + Random Forests
  + Gradient Boosting Machines (GBM)
  + Support Vector Machines (SVM)
  + Neural Networks
* **Training**: Train the selected models using the labeled dataset. This involves optimizing model parameters to achieve the best performance.

### **5. Evaluation Metrics**

* **Performance Evaluation**: Assess model performance using metrics such as:
  + Accuracy: Overall correctness of predictions.
  + Precision: Proportion of correctly predicted phishing URLs among all predicted phishing URLs.

### **6. Model Validation and Optimization**

* **Cross-validation**: Validate models using techniques like k-fold cross-validation to ensure robustness.

### **7. Deployment and Integration**

* **Production Deployment**: Integrate the trained model into a system capable of real-time URL scanning.
* **Scalability**: Ensure the solution can handle large volumes of URLs efficiently and effectively.

### **9. Security and Privacy Considerations**

* **Security**: Ensure the solution is secure against attacks that attempt to evade detection.
* **Privacy**: Handle sensitive data (like URL content) with appropriate privacy measures.

### **10. Continuous Improvement**

* **Adaptation**: Continuously update the model based on new data and emerging phishing trends to enhance detection accuracy.

**2.2 EXISTING SYSTEM:**

Moghimi and Vorjani [32] proposed a system independent from third services like Google Page Rank or WHOIS. They used two handcrafted feature sets, extracted from the URL and the Document Object Model (DOM) of the website. The first set has ninth at are related to the structure of the website and (iii) 12 deep features related to the text and image similarity. Combining these sets of features e legacy features including a set of keywords, while the second has eight novel features which inform of whether the website's resources are loaded using SSL protocol or not. They used Levenstein distance [33] to detect typo-squatting by comparing the website and resources URLs. These features were used to train an SVM classifier and obtained an accuracy of 98:65% on their banking websites dataset.

Adebowale *et al.* [34] created a browser extension to protect users by extracting features from the URL, the source code, the images, and features extracted using third party services like WHOIS. Those features were introduced into an Adaptive Neuro-Fuzzy Inference System (ANFIS) and combined with the Scale-Invariant Feature Transform (SIFT) algorithm, obtaining an accuracy of 98:30% on Rami *et al.* [35] dataset.

Rao and Pais [28] developed a phishing website classifier using the URL, the hyperlinks on the HTML code and third-party services including the age of the domain and the page rank on Alexa. They reached 99:31% accuracy with a Random Forest classifier. Yang *et al.* [36] proposed an Extreme Learning Machine (ELM) model and established three different groups of features: (i) Surface features, composed of 12 URL handcrafted and 4 Domain Name System (DNS) features related to the registration date and the DNS records for that domain; (ii) 28 Topological features and the ELM classifier, they obtained 97:5% accuracy.

Sadique *et al.* [37] presented a framework for real-time phishing detection using four sets of URL features: (i) Lexical features related to the number of characters, dots and symbols found in different parts of the URL, (ii) host-based Features that are related to the host, (iii) WHOIS features are related to the registration date and (iv) GeoIP-based features like the Autonomous System Number (ASN). A total of 142 individual features were evaluated using 98; 000 samples from Phishtank, where legitimate samples are also picked from false positives collected at PhishTank. They obtained a 90:51% accuracy on a Random Forest classifier using the proposed descriptors.

Li *et al.* [29] presented a stacking model which was the combination of three models: Gradient Boost Decision Tree (GBDT), eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Model (LGBM). This stacking model was fed with a set of features from different sources: eight from the URL, 11 from the HTML and HTML string embeddings inspired by Word2Vec model [38]. They obtained 97:30% accuracy using a 49; 947 samples dataset.

**2.3 DISADVANTAGES OF EXISTING SYSTEM**

* From an existing perspective, and to the best of our knowledge, publicly available datasets are not reflecting conditions that represent some real problems for phishing URL detection.
* it is observed that recent machine learning proposals obtained high accuracy using outdated datasets, i.e., typically containing URLs collected from 2009 to 2017. We demonstrate that models trained with old URLs decrease their performance when they are tested with URLs coming from recent phishing pages.

**2.4 PROPOSED SYSTEM:**

This paper presents a phishing URL dataset using legitimate login websites to obtain the URLs from such pages. Then, we evaluate machine and deep learning techniques for recommending the method with higher accuracy. Next, we show how models trained with legitimate homepages struggle to classify legitimate login URLs, demonstrating our hypothesis about phishing detection and legitimate login URLs. Additionally, we show how the accuracy decreases with the time on models trained with datasets from 2016 and evaluated on data collected in 2020. Finally, we provide an overview of current phishing encounters, explaining attacker tricks and approaches.

\_ We extended our previous dataset PILU-60K (Phishing Index Login URL) [20], from 60K to 90K URLs equally distributed among three classes: phishing, the legitimate homepage, and legitimate login. We make this extended dataset, PILU-90K, publicly available for research purposes.

\_ Using PILU-90K, we implemented and evaluated three pipelines for URL phishing detection: (i) we use the 38 handcrafted feature descriptors proposed by Sahingoz *et al.* [21] for training eight supervised machine learning classifiers and also (ii) automatic feature extraction using Term Frequency Inverse Document Frequency (TF-IDF) at character N-gram level combined with Logistic Regression (LR) algorithm, and (iii) a Convolutional Neural Network (CNN) at character level too.

\_ We demonstrated empirically how an URL phishing detection model struggles in classifying login URLs when it was trained on the URLs of the homepage of phishing and legitimate URLs.

\_ We evaluated the robustness of the proposed phishing detection over time. We trained the model on a dataset collected between March 2016 and April 2016, and weevaluated the model on other datasets collected between 2017 and 2020.

\_ Phishing websites were analyzed using domain frequency. We found six different phishing domains depending on the service hired by the attacker.

**2.4 ADVANTAGES OF PROPOSED SYSTEM**

* Machine learning models to detect unreported phishing encounters. Depending on their input data, these approaches can be classified into two categories: URL-based and content based.
* present an extended version of the Phishing Index Login URL (PILU-60K) dataset [20] and we name it PILU-90K. PILU-90K contains 90K URLs divided into three classes.

**3.ANALYSIS:**

**3.1 INTRODUCTION**

In ML and statistics, classification method is an approach involving supervised learning where computer program gains information from input and afterward utilizes this figuring out how to characterize new observations. Here are few classification techniques used in the detection of phishing URLs.

**3.2 SOFTWARE REQUIREMENT SPECIFICATION:**

Hardware Requirements:

• Processor CPU - Intel Pentium Dual Core and Higher

• Hard Disk capacity - 512MB Space required minimum

• RAM - 4GB minimum

Software requirements

• Programming language - Python

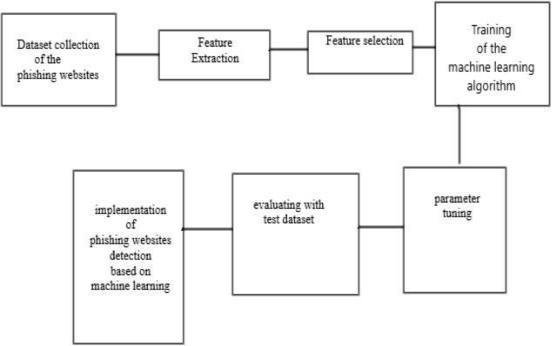
• Operating system - Windows 8.1 or above

• IDE - Anaconda , iPython version 3.x

**3.3 FEASIBILITY STUDY:**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**3.4 ARCHITECTURE DIAGRAM:**



**3.4 ALGORTIHMS:**

**Decision tree classifiers**

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labeled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting

Gradient boosting is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique used in [regression](https://en.wikipedia.org/wiki/Regression_(machine_learning)) and [classification](https://en.wikipedia.org/wiki/Classification_(machine_learning)) tasks, among others. It gives a prediction model in the form of an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) of weak prediction models, which are typically [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning).[[1]](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en-US&rs=en-US&wopisrc=https%3A%2F%2Fmy.microsoftpersonalcontent.com%2Fpersonal%2F911d7b312152ca2c%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F365d4ee19c814fd4b3310d54db1de1b4&wdorigin=MARKETING.WORD.SIGNIN,APPHOME-WEB.BANNER.UPLOAD&wdprevioussession=e53d55cb-3f50-4630-830b-e212ff9d855b&wdprevioussessionsrc=AppHomeWeb&wdenableroaming=1&mscc=1&wdodb=1&hid=00CB35A1-9073-5000-DA28-5422D5E129A6.0&uih=onedrivecom&wdlcid=en-US&jsapi=1&jsapiver=v2&corrid=71add319-b3b6-e03b-e16b-b45527b6979a&usid=71add319-b3b6-e03b-e16b-b45527b6979a&newsession=1&sftc=1&uihit=docaspx&muv=1&cac=1&sams=1&sfp=1&sdp=1&hch=1&hwfh=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fonedrive.live.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&instantedit=1&wopicomplete=1&wdredirectionreason=Unified_SingleFlush#cite_note-:1-1)[[2]](https://word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=en-US&rs=en-US&wopisrc=https%3A%2F%2Fmy.microsoftpersonalcontent.com%2Fpersonal%2F911d7b312152ca2c%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F365d4ee19c814fd4b3310d54db1de1b4&wdorigin=MARKETING.WORD.SIGNIN,APPHOME-WEB.BANNER.UPLOAD&wdprevioussession=e53d55cb-3f50-4630-830b-e212ff9d855b&wdprevioussessionsrc=AppHomeWeb&wdenableroaming=1&mscc=1&wdodb=1&hid=00CB35A1-9073-5000-DA28-5422D5E129A6.0&uih=onedrivecom&wdlcid=en-US&jsapi=1&jsapiver=v2&corrid=71add319-b3b6-e03b-e16b-b45527b6979a&usid=71add319-b3b6-e03b-e16b-b45527b6979a&newsession=1&sftc=1&uihit=docaspx&muv=1&cac=1&sams=1&sfp=1&sdp=1&hch=1&hwfh=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fonedrive.live.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Normal&instantedit=1&wopicomplete=1&wdredirectionreason=Unified_SingleFlush#cite_note-hastie-2) When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms [random forest](https://en.wikipedia.org/wiki/Random_forest).A gradient-boosted trees model is built in a stage-wise fashion as in other [boosting](https://en.wikipedia.org/wiki/Boosting_(machine_learning)) methods, but it generalizes the other methods by allowing optimization of an arbitrary [differentiable](https://en.wikipedia.org/wiki/Differentiable_function) [loss function](https://en.wikipedia.org/wiki/Loss_function).

K-Nearest Neighbors (KNN)

* Simple, but a very powerful classification algorithm
* Classifies based on a similarity measure
* Non-parametric
* Lazy learning
* Does not “learn” until the test example is given
* Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example

* Training dataset consists of k-closest examples in feature space
* Feature space means, space with categorization variables (non-metric variables)
* Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset
* Logistic regression Classifiers

*Logistic regression analysis* studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name *logistic regression* is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name *multinomial logistic regression* is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modeling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis.

Naïve Bayes

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the

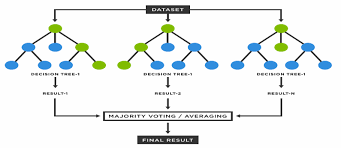
literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset We try above all to understand the obtained results.

**Random Forest**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.



The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breyman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc. The extension combines Breyman's "bagging" idea and random selection of features, introduced and later independently by Amit and German in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "black box" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

SVM

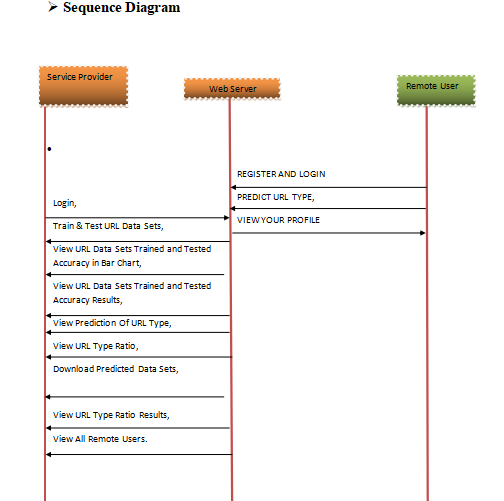
In classification tasks a discriminant machine learning technique aims at finding, based on an *independent and identically distributed* (*iid*) training dataset, a discriminant function that can correctly predict labels fornewly acquired instances. Unlike generative machine learning approaches, which require computations ofconditional probability distributions, a discriminant classification function takes a data point *x* and assignsit to one of the different classes that are a part of the classification task. Less powerful than generativeapproaches, which are mostly used when prediction involves outlier detection, discriminant approachesrequire fewer computational resources and less training data, especially for a multidimensional featurespace and when only posterior probabilities are needed. From a geometric perspective, learning a classifieris equivalent to finding the equation for a multidimensional surface that best separates the different classesin the feature space.

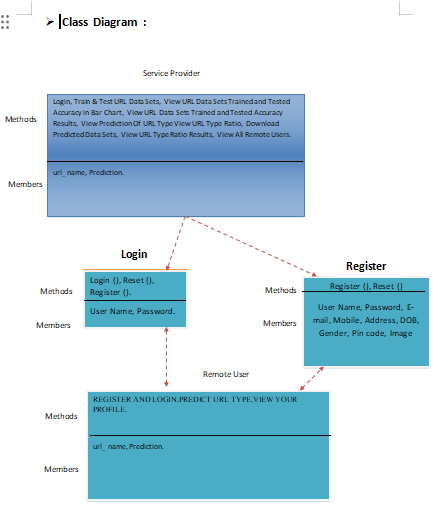
SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to *genetic algorithms* (*GAs*) or *perceptron*, both of which are widely used for classification in machine learning. For perceptron's, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes’ meeting this requirement.

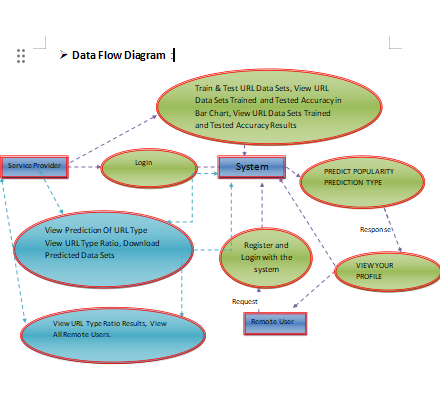
4.DESIGN

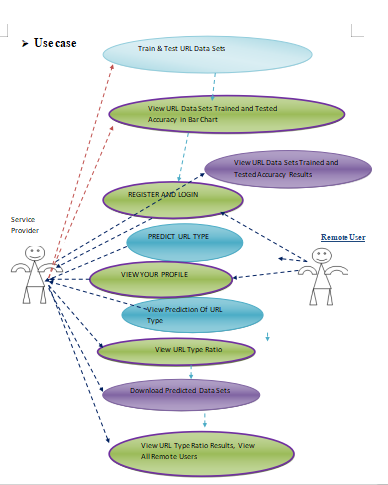
* 4.1 SYSTEM MODELS

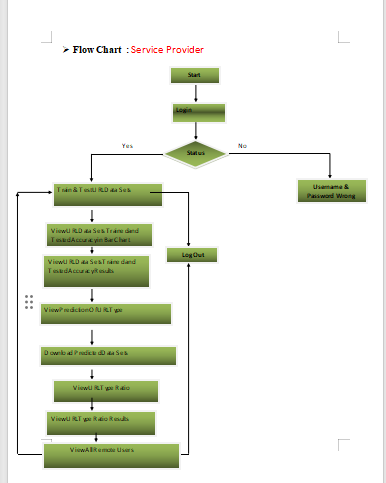
SEQUENCE DIAGRAM

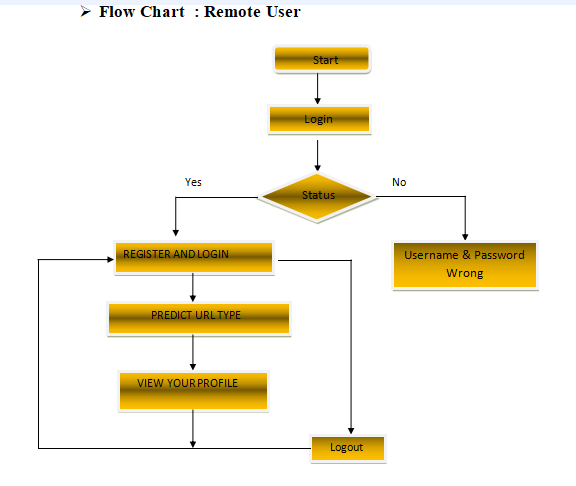












**5.IMPLEMNTATION AND RESULT**

* **5.1 SOFTWARE TOOLS**

1. Programming Languages and Frameworks:

* Python: Widely used for its rich ecosystem of machine learning libraries and ease of integration with other tools.
  + NumPy and Pandas: Fundamental libraries for data manipulation and preprocessing.
  + Scikit-learn: Offers a wide range of machine learning algorithms and tools for model selection, training, and evaluation.
  + TensorFlow or PyTorch: Deep learning frameworks for building and training neural networks, especially useful for complex pattern recognition tasks in phishing detection.

### 2. Feature Extraction and Data Analysis:

* Natural Language Toolkit (NLTK): Useful for text processing tasks, such as tokenization and stemming, which can be relevant if analyzing textual content associated with URLs.
* Beautiful Soup and Scrapy: Python libraries for web scraping, useful for collecting URLs from various sources on the internet.

### 3. Data Visualization:

* Matplotlib and Seaborn: Popular libraries for creating static, animated, and interactive visualizations to analyze data distributions and model performance.

### 4. Machine Learning Model Interpretability:

* SHAP (SHapley Additive exPlanations): Provides explanations for ML model p**redictions, helping to understand the importance of features in model decisions.**

### **5.cloud services**

* Google Cloud Platform (GCP) or Amazon Web Services (AWS): Offer scalable computing resources and machine learning services (e.g., AI Platform, SageMaker) for training and deploying ML models.
* Azure Machine Learning: Microsoft's cloud service for building, training, and deploying ML models.

### 6. Integrated Development Environments (IDEs):

* Jupyter Notebook or JupyterLab: Interactive environments for data analysis and machine learning experimentation.
* PyCharm, Visual Studio Code (VS Code): IDEs with rich plugin ecosystems for Python development and ML model integration.

### 7. Model Deployment and Monitoring:

* Docker: Containerization tool for packaging applications and their dependencies, facilitating consistent deployment across different environments.
* Kubernetes: Container orchestration platform for managing containerized applications, useful for scaling and monitoring deployed ML models.

### 8. Collaborative Tools and Version Control:

* GitHub or GitLab: Version control platforms for collaborative development and tracking changes in code and model development.
* Slack or Microsoft Teams: Communication platforms for team collaboration and project management.

which all works as an ensemble Each separate tree of the Random forest gives out a

class forecast and the class with the most votes transforms into our model’s desire

As Parameters used:

– N estimators: The number of trees in the forest. The number used in the

algorithm is 10.

– criterion: the function that is used to measure the quality of a split. The one

that is used in the algorithm is “entropy”

**5.2 SYSTEM MODULES**

**Service Provider**

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Train & Test URL Data Sets, View URL Data Sets Trained and Tested Accuracy in Bar Chart, View URL Data Sets Trained and Tested Accuracy Results, View Prediction Of URL Type, View URL Type .Download Predicted Data Sets, View URL Type Ratio Results, View All Remote Users.

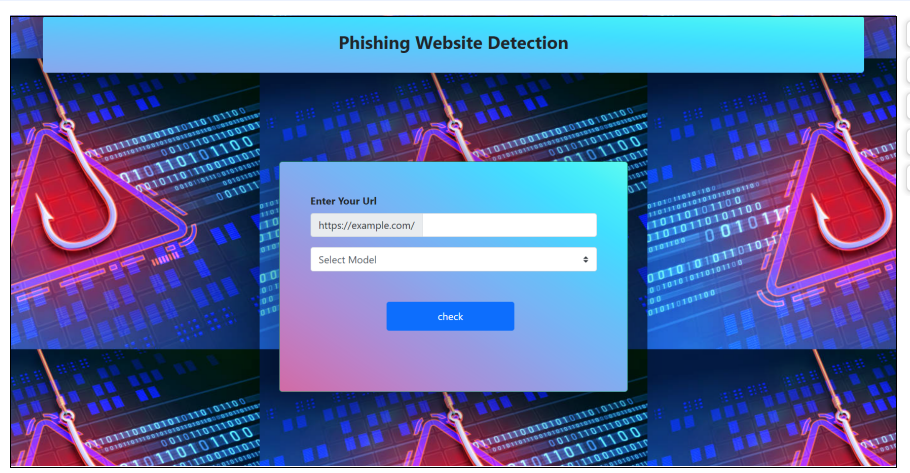
**View and Authorize Users**

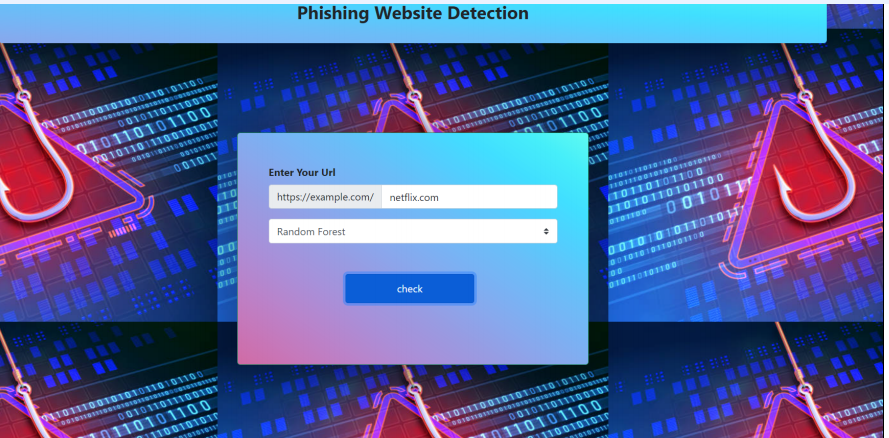
In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

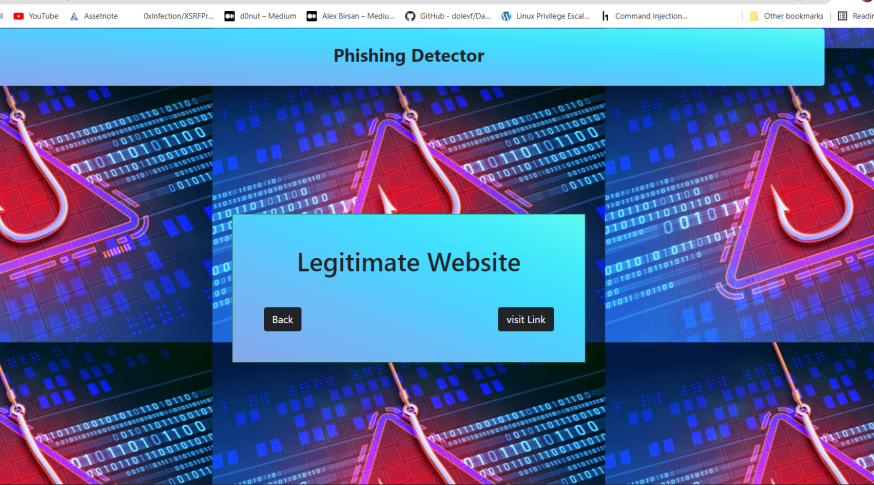
**Remote User**

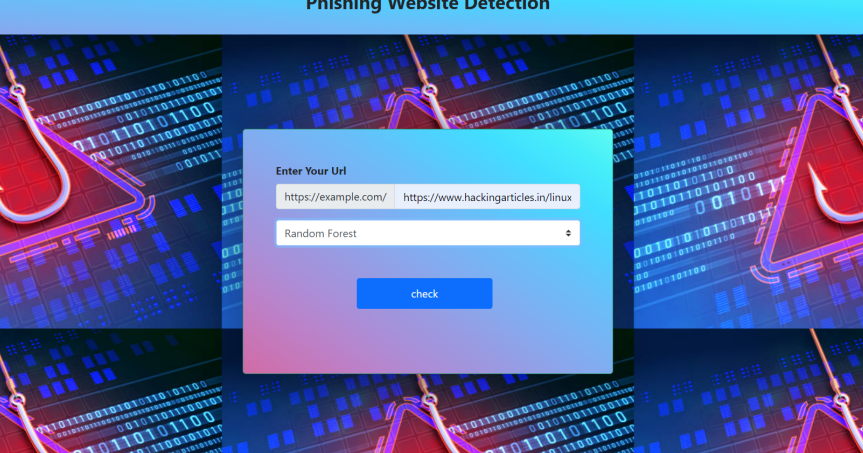
In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT URL TYPE, VIEW YOUR PROFILE..

**SUMMARY:**









**6.TESTING AND VALIDATION**

### **6.1 TESTING METHODOLOGIES:**

The following are the Testing Methodologies

**Unit Testing.**

* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a 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. Hence, the naming is Unit Testing.
* During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.
* **Integration Testing**
* 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 tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.
* **User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developingand making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

* **7.1.4 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 under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

* **.1.5 Validation Checking**
* Validation checks are performed on the following file.
* **Text Field:**
* The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

* **Numeric Field:**
* The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.
* **Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

* **Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easy.

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is

achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**UNIT TESTING:**

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

WORK:

Service Provider

Login,

Train & Test URL Data Sets,

View URL Data Sets Trained and Tested Accuracy in Bar Chart,

View URL Data Sets Trained and Tested Accuracy Results,

View Prediction Of URL Type,

View URL Type Ratio,

Download Predicted Data Sets,

View URL Type Ratio Results,

View All Remote Users.

Remote User

REGISTER AND LOGIN,

PREDICT URL TYPE,

VIEW YOUR PROFILE.

Members

url\_name,

Prediction.

**7.CONLUSION:**

The demonstration of phishing is turning into an advanced danger to this quickly de-

eloping universe of innovation. Today, every nation is focusing on cashless exchanges, business online, tickets that are paperless and so on to update with the growing world. Yet phishing is turning into an impediment to this advancement. Individuals are not feeling web is dependable now. It is conceivable to utilize AI to get information and assemble extraordinary information items. A lay person, completely unconscious of how to recognize a security danger shall never invite the danger of making money related exchanges on the web. Phishers are focusing on installment industry and cloud benefits the most. The project means to investigate this region by indicating an utilization instance of recognizing phishing sites utilizing ML. It aimed to build a phishing detection mechanism using machine learning tools and techniques which is efficient, accurate and cost effective.

The project was carried out in Anaconda IDE and was written in Python.

The proposed method used four machine learning classifiers to achieve this and a comparative study of the four algorithms was made. A good accuracy score was also achieved.The four algorithms used are K-Nearest neighbor, Kernel Support Vector Machine, De-51

**8.2 Future Enhancement**

Further work can be done to enhance the model by using ensembling models to get greateraccuracy score. Ensemble methods is a ML technique that combines many base models to generate an optimal predictive model. Further reaching future work would be combining multiple classifiers, trained on different aspects of the same training set, into a single classifier that may provide a more robust prediction than

anyof the single classifiers on their own. The project can also include other variants of phishing like smishing, vishing, etc. To complete the system. Looking even further out, the methodology needs to be evaluated on how it might handle collection growth. The collections will ideally grow incrementally over time so there will need to be

way to apply a classifier incrementally to the newdata, but also potentially have this classifier receive feedback that might modify it over time.

**8.REFERENCES:**

[1] Reid G. Smith and Joshua Eckroth. Building ai applications: Yesterday, today, andtomorrow. AI Magazine, 38(1):6–22, Mar. 2017.

[2] Panos Louridas and Christof Ebert. Machine learning. IEEE Software, 33:110–115,09 2016.

[3] Michael Jordan and T.M. Mitchell. Machine learning: Trends, perspectives, and

prospects. Science (New York, N.Y.), 349:255–60, 07 2015.

[4] Steven Aftergood. Cybersecurity: The cold war online. Nature, 547:30+, Jul2017.7661.

[5] Aleksandar Milenkoski, Marco Vieira, Samuel Kounev, Alberto Avritzer, and BryanPayne. Evaluating computer intrusion detection systems: A survey of common practices. ACM Computing Surveys, 48:12:1–, 09 2015.

[6] Chirag N. Modi and Kamatchi Acha. Virtualization layer security challenges and

intrusion detection/prevention systems in cloud computing: a comprehensive review.

The Journal of Supercomputing, 73(3):1192–1234, Mar 2017.

[7] Eduardo Viegas, Altair Santin, Andre Fanca, Ricardo Jasinski, Volnei Pedroni, andLuiz Soares de Oliveira. Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems. IEEE Transactions on Computers, 66:1–1, Jan2016.

[8] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. Machine learning and deep learning methods for cybersecurity. IEEE Access, 6:35365–35381, 2018.

[9] Neha R. Israni and Anil N. Jaiswal. A survey on various phishing and anti-phishingmeasures. International journal of engineering research and technology, 4, 2015.

[10] Pingchuan Liu and Teng-Sheng Moh. Content based spam e-mail filtering. pages218–224, 10 2016.

[11] N. Agrawal and S. Singh. Origin (dynamic blacklisting) based spammer detection and spam mail filtering approach. In 2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC),pages 99–104, 2016.

[12] Vikas Sahara, Sheetal Kumar Jain, and Manish Giri. Survey: anti-phishing framework using visual cryptography on cloud. JAFRC, 2, 01 2015.

[13] S. Patil and S. Dhage. A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework. In 2019 5th International

Conference on Advanced Computing Communication Systems (ICACCS), pages 588–593, 2019.

[14] Dipesh Vaya, Sarika Khandelwal, and Teena Hadpawat. Visual cryptography: A

review. International Journal of Computer Applications, 174:40–43, 09 2017.

[15] Saurabh Saoji. Phishing detection system using visual cryptography, 03 2015.

[16] C. Pham, L. A. T. Nguyen, N. H. Tran, E. Huh, and C. S. Hong. Phishing-aware:

A neuro-fuzzy approach for anti-phishing on fog networks. IEEE Transactions on

Network and Service Management, 15(3):1076–1089, 2018.

[17] K. S. C. Yong, K. L. Chiew, and C. L. Tan. A survey of the qr code phishing:

the current attacks and countermeasures. In 2019 7th International Conference on

Smart Computing Communications (ICSCC), pages 1–5, 2019.

[18] G. Egozi and R. Verma. Phishing email detection using robust nlp techniques. In

2018 IEEE International Conference on Data Mining Workshops (ICDMW), pages

7–12, 2018.

[19] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang. Phishing-alarm: Robust and efficient phishing detection via page component similarity. IEEE Access, 5:17020–17030,2017.

[20] G. J. W. Kathrine, P. M. Praise, A. A. Rose, and E. C. Kalaivani. Variants of phishing attacks and their detection techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pages 255–259, 2019.

[21] Muhammet Baykara and Zahit Gurel. Detection of phishing attacks. pages 1–5, 032018.

[22] Prof. Gayathri Naidu . A survey on various phishing detection and prevention

techniques. International Journal of Engineering and Computer Science, 5(9), May

2016.

[23] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu. Ofs-nn: An effective phishing websites detection model based on optimal feature selection and neural network. IEEE Access,7:73271–73284, 2019.

[24] Mahdieh Zabihimayvan and Derek Doran. Fuzzy rough set feature selection to

enhance phishing attack detection, 03 2019.

[25] P. Yang, G. Zhao, and P. Zeng. Phishing website detection based on multidimensional features driven by deep learning. IEEE Access, 7:15196–15209, 2019.

[26] T. Nathezhtha, D. Sangeetha, and V. Vaidehi. Wc-pad: Web crawling based phishing attack detection. In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–6, 2019.

[27] Y. Huang, Q. Yang, J. Qin, and W. Wen. Phishing url detection via cnn and

attention-based hierarchical rnn. In 2019 18th IEEE International Conference On

Trust, Security And Privacy In Computing And Communications/13th IEEE Inter-

national Conference On Big Data Science And Engineering (TrustCom/BigDataSE),

pages 112–119, 2019.

[28] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee. An adaptive machine learning based approach for phishing detection using hybrid features. In 2019 5th International Conference on Web Research (ICWR), pages 281–286, 2019.

[29] C. E. Shyni, A. D. Sundar, and G. S. E. Ebby. Phishing detection in websites using parse tree validation. In 2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS), pages 1–4, 2018.

[30] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe. A new method for detection o

n. In 2018 Second International Conference on In-

ventive Communication and Computational Technologies (ICICCT), pages 949–952,

2018.

[31] J. Li and S. Wang. Phishbox: An approach for phishing validation and de-

tection. In 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure

Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl

Conf on Big Data Intelligence and Computing and Cyber Science and Technology

Congress (DASC/Pi Com/DataCom/Cybersmith), pages 557–564, 2017.

[32] H. Shirazi, K. Haefner, and I. Ray. Fresh-phish: A framework for auto-detection of phishing websites. In 2017 IEEE International Conference on Information Reuse

and Integration (IRI), pages 137–143, 2017.

[33] A. J. Park, R. N. Quadair, and H. H. Tsang. Phishing website detection framework through web scraping and data mining. In 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pages680–684, 2017.

[34] Lisa Machado and Jayant Gadge. Phishing sites detection based on c4.5 decision

tree algorithm. pages 1–5, 08 2017.

[35] S. Haruta, H. Asahina, and I. Sasase. Visual similarity-based phishing detection

scheme using image and css with target website finder. In GLOBECOM 2017 - 2017

IEEE Global Communications Conference, pages 1–6, 2017.

[36] Afroz Chakure. ”k-nearest neighbors (knn) algorithm”. https://

towardsdatascience.com/k-nearest-neighbors-knn-algorithm-bd375d14eec7,

07 2019.

[37] Kitti pong Chambon, Pasa pitch Churai, Pongwarin Teerarassammee, Kittaka

Kardarases, and Nittaya Kerdasa. An empirical study of distance metrics for

k-nearest neighbor algorithm. pages 280–285, 01 2015.

[38] Natasha Sharma. ” importance of distance metrics in ma-

chine learning modelling”. https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d,01 2019.

[39] Czako Zoltan. ”sum and kernel sum”. https://towardsdatascience.com/

svm-and-kernel-svm-fed02bef1200, 12 2018.

[40] Afroz Chekuri. ” decision tree classification, an introductioto decision tree classifier”. https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac, 07 2019.

[41] Tony Yiu. ” understanding random forest, how the algorithm works

and why it is so effective”. https://towardsdatascience.com/understanding-random-forest-58381e0602d2, 06 2019.

[42]” modules - python documentation”. https://docs.python.org/3/tutorial/

modules.html, 2020.

[43] ”dfd symbols”. https://www.visual-paradigm.com/guide/data-flow-diagram/

what-is-data-flow-diagram/, 2020.

[44] ”what is activity diagram?”. https://www.visual-paradigm.com/guide/

uml-unified-modeling-language/what-is-activity-diagram/, 2020.

[45] Theodoros Iliou, Christos-Nikolaos Anagnostopoulos, Marina Nerantzaki, and

George Anastassopoulos. A novel machine learning data preprocessing method for

enhancing classification algorithms performance. 09 2015.

[46] Sebastian Raschka. About feature scaling and normalization and the effect of standardization for machine learning algorithms. 07 2014.