## WEB PHISHING DETECTION

**TEAM ID : PNT2022TMID32234**

**SUBMITTED BY ,**

**TEAM LEADER: VAISHNAVI**

**TEAM MEMBER:**

**LAKSHMI DHARSHINI**

**KARTHIKEYAN**

**PAVISH KUMAR**

|  |  |
| --- | --- |
| **SERIAL NUMBER** | **TITLE** |
| **1** | **INTRODUCTION** |
| **1.1** | **PROJECT OVERVIEW** |
| **1.2** | **PURPOSE** |
| **2** | **LITERATURE SUREVEY** |
| **2.1** | **Existing problem** |
| **2.2** | **References** |
| **2.3** | **Problem Statement Definition** |
| **3** | **Empathy Map Canvas** |
| **3.1** | **Ideation & Brainstorming** |
| **3.2** | **Proposed Solution** |
| **3.3** | **Problem Solution fit** |
| **3.4** | **REQUIREMENT ANALYSIS** |
| **4** | **Functional Requirements** |
| **4.1** | **Non-Functional Requirements** |
| **4.2** | **PROJECT DESIGN** |
| **5** | **Data Flow Diagrams** |
| **5.1** | Solution & Technical Architecture |
| **5.2** | **User Stories** |
| **5.3** | **PROJECT PLANNING & SCHEDULING** |
| **6** | **Sprint Planning & Estimation** |
| **6.1** | **Sprint Delivery Schedule** |
| **6.2** | **Reports from JIRA** |
| **7** | **CODING & SOLUTIONING** |
| **7.1** | **Feature 1** |
| **7.2** | **Feature 2** |
| **7.3** | **Database Schema** |
| **8** | **TESTING** |
| **8.1** | **Test Cases** |
| **8.2** | **User Acceptance Testing** |
| **9** | **RESULTS** |
| **9.1** | **Performance Metrics** |
| **10** | **ADVANTAGES & DISADVANTAGES** |
| **11** | **CONCLUSION** |
| **12** | **FUTURE SCOPE** |
| **13** | **APPENDIX** |

**Existing problem:**

Today's growing phishing websites pose significant threats due to their extremely undetectable risk. They anticipate internet users to mistake them as genuine ones in order to reveal user information and privacy, such

as login ids, pass-words, credit card numbers, etc. without notice. This paper proposes a new approach to solve the anti-phishing problem. The new features of this approach can be represented by URL character sequence without phishing prior knowledge, various hyperlink information, and textual content of the webpage, which are combined and fed to train the XG Boost classifier. One of the major contributions of this paper is the selection of different new features, which are capable enough to detect 0-h attacks, and these features do not depend on any third-party services. In particular, we extract character level Term Frequency-Inverse Document Frequency (TF-IDF) features from noisy parts of HTML and plaintext of the given webpage. Moreover, our proposed hyperlink features determine the relationship between the content and the URL of a webpage. Due to the absence of publicly available large phishing data sets, we needed to create our own data set with 60,252 webpages to validate the proposed solution. This data contains 32,972 benign webpages and 27,280 phishing webpages. For evaluations, the performance of each category of the proposed feature set is evaluated, and various classification algorithms are employed. From the empirical results, it was observed that the proposed individual features are valuable for phishing detection. However, the integration of all the features improves the detection of phishing sites with significant accuracy. The proposed approach achieved an accuracy of 96.76% with only 1.39% false-positive rate on our dataset, and an accuracy of 98.48% with 2.09% false-positive rate on benchmark dataset, which outperforms the existing baseline approaches.

**Introduction**

Phishing offenses are increasing, resulting in billions of dollars in loss1. In these attacks, users enter their critical (i.e., credit card details, passwords, etc.) to the forged website which appears to be legitimate. The Software-as-a-Service (SaaS) and webmail sites are the most common targets of phishing2. The phisher makes websites that look very similar to the benign websites. The phishing website link is then sent to millions of internet users via emails and other communication media. These types of cyber-attacks are usually activated by emails, instant messages, or phone calls3. The aim of the phishing attack is not only to steal the victims' personality, but it can also be performed to spread other types of mal ware such as ransomware, to exploit approach weaknesses, or to receive monetary profits4. According to the Anti-Phishing Working Group (APWG) report in the 3rd Quarter of 2020, the number of phishing attacks has grown since March, and 28,093 unique phishing sites have been detected between July to September2. The average amount demanded during wire transfer Business E-mail Compromise (BEC) attacks was $48,000 in the third quarter, down from $80,000 in the second quarter and $54,000 in the first.

Detecting and preventing phishing offenses is a significant challenge for researchers due to the way phishers carry out the attack to bypass the existing anti-phishing techniques. Moreover, the phisher can even target some educated and experienced users by using new phishing scams. Thus, software-based phishing detection techniques are preferred for fighting against the phishing attack. Mostly available methods for detecting phishing attacks are blacklists/whitelists5, natural language processing6, visual similarity7, rules8, machine learning techniques 9,10, etc. Techniques based on blacklists/whitelists fail to detect unlisted phishing sites (i.e. 0-h attacks) as well as these methods fail when blacklisted URL is encountered with minor changes. In the machine learning based techniques, a classification model is trained using various heuristic features (i.e., URL, webpage content, website traffic, search engine, WHOIS record, and Page Rank) in order to improve detection efficiency. However, these heuristic features are not warranted to present in all phishing websites and might also present in the benign websites, which may cause a classification error. Moreover, some of the heuristic features are hard to access and third-party dependent. Some third-party services (i.e., page rank, search engine indexing, WHOIS etc.) may not be sufficient to identify phishing websites that are hosted on hacked servers and these websites are inaccurately identified as benign websites because they are contained in search results. Websites hosted on compromised servers are usually more than a day old unlike other phishing websites which only take a few hours. Also, these services inaccurately identify the new benign website as a phishing site due to the lack of domain age. The visual similarity-based heuristic techniques compare the new website with the pre-stored signature of the website. The website’s visual signature includes screenshots, font styles, images, page layouts, logos, etc. Thus, these techniques cannot identify the fresh phishing websites and generate a high false-negative rate (phishing to benign). The URL based technique does not consider the HTML of the webpage and may misjudge some of the malicious websites hosted on free or compromised servers. Many existing approaches11,12,13 extract hand-crafted URL based features, e.g., number of dots, presence of special “@”, “#”, “–” symbol, URL length, brand names in URL, position of Top-Level domain, check hostname for IP address, presence of multiple TLDs, etc. However, there are still hurdles to extracting manual URL features due to the fact that human effort requires time and extra maintenance labor costs. Detecting and preventing phishing offense is a major defiance for researchers because the scammer carries out these offenses in a way that can avoid current anti-phishing methods. Hence, the use of hybrid methods rather than a single approach is highly recommended by the networks security manager.

This paper provides an efficient solution for phishing detection that extracts the features from website's URL and HTML source code. Specifically, we proposed a hybrid feature set including URL character sequence features without expert’s knowledge, various hyperlink information, plaintext and noisy HTML data-based features within the HTML source code. These features are then used to create feature vector required for training the proposed approach by XG Boost classifier. Extensive experiments show that the proposed anti-phishing approach has attained competitive performance on real dataset in terms of different evaluation statistics.

Our anti-phishing approach has been designed to meet the following requirements.

High detection efficiency: To provide high detection efficiency, incorrect classification of benign sites as phishing (false-positive) should be minimal and correct classification of phishing sites (true-positive) should be high.

Real-time detection: The prediction of the phishing detection approach must be provided before exposing the user's personal information on the phishing website.

Target independent: Due to the features extracted from both URL and HTML the proposed approach can detect new phishing websites targeting any benign website (zero-day attack).

Third-party independent: The feature set defined in our work are lightweight and client-side adaptable, which do not rely on third-party services such as blacklist/whitelist, Domain Name System (DNS) records, WHOIS record (domain age), search engine indexing, network traffic measures, etc. Though third-party services may raise the effectiveness of the detection approach, they might misclassify benign websites if a benign website is newly registered. Furthermore, the DNS database and domain age record may be poisoned and lead to false negative results (phishing to benign).

Hence, a light-weight technique is needed for phishing websites detection adaptable at client side. The major contributions in this paper are itemized as follows.

We propose a phishing detection approach, which extracts efficient features from the URL and HTML of the given webpage without relying on third-party services. Thus, it can be adaptable at the client side and specify better privacy.

We conducted extensive experiments using various machine learning algorithms to measure the efficiency of the proposed features. Evaluation results manifest that the proposed approach precisely identifies the legitimate websites as it has a high true negative rate and very less false positive rate.

We release a real phishing webpage detection dataset to be used by other researchers on this topic.

The rest of this paper is structured as follows: The "Related work" section first reviews the related works about phishing detection. Then the "Proposed approach" section presents an overview of our proposed solution and describes the proposed features set to train the machine learning algorithms. The "Experiments and result analysis” section introduces extensive experiments including the experimental dataset and results evaluations. Furthermore, the "Discussion and limitation" section contains a discussion and limitations of the proposed approach. Finally, the "Conclusion" section concludes the paper and discusses future work.

**Related work**

This section provides an overview of the proposed phishing detection techniques in the literature. Phishing methods are divided into two categories; expanding the user awareness to distinguish the characteristics of phishing and benign webpages14, and using some extra software. Software-based techniques are further categorized into list-based detection, and machine learning-based detection. However, the problem of phishing is so sophisticated that there is no definitive solution to efficiently bypass all threats; thus, multiple techniques are often dedicated to restrain particular phishing offenses.

**List-based detection**

List-based phishing detection methods use either whitelist or blacklist-based technique. A blacklist contains a list of suspicious domains, URLs, and IP addresses, which are used to validate if a URL is fraudulent. Simultaneously, the whitelist is a list of legitimate domains, URLs, and IP addresses used to validate a suspected URL. Wang et al.15, Jain and Gupta5 and Han et al.

16 use white list-based method for the detection of suspected URL. Blacklist-based methods are widely used in openly available anti-phishing toolbars, such as Google safe browsing, which maintains a blacklist of URLs and provides warnings to users once a URL is considered as phishing. Prakash et al.17 proposed a technique to predict phishing URLs called Phishnet. In this technique, phishing URLs are identified from the existing blacklisted URLs using the directory structure, equivalent IP address, and brand name. Eelegyhazi at al.18 developed a method that compares the domain name and name server information of new suspicious URLs to the information of blacklisted URLs for the classification process. Sheng et al.19 demonstrated that a forged domain was added to the blacklist after a considerable amount of time, and approximately 50–80% of the forged domains were appended after the attack was carried out. Since thousands of deceptive websites are launched every day, the blacklist requires to be updated periodically from its source. Thus, machine learning-based detection techniques are more efficient in dealing with phishing offenses.

Machine learning-based detection

Data mining techniques have provided outstanding performance in many applications, e.g., data security and privacy20, game theory21, blockchain systems22, healthcare23, etc. Due to the recent development of phishing detection methods, various machine learning-based techniques have also been employed6,9,10,13 to investigate the legality of websites. The effectiveness of these methods relies on feature collection, training data, and classification algorithm. The feature collection is extracted from different sources, e.g., URL, webpage content, third party services, etc. However, some of the heuristic features are hard to access and time-consuming, which makes some machine learning approaches demand high computations to extract these features.

Jain and Gupta24 proposed an anti-phishing approach that extracts the features from the URL and source code of the webpage and does not rely on any third-party services. Although the proposed approach attained high accuracy in detecting phishing webpages, it used a limited dataset (2141 phishing and 1918 legitimate webpages). The same authors9 present a phishing detection method that can identify phishing attacks by analyzing the hyperlinks extracted from the HTML of the webpage. The proposed method is a client-side and language-independent solution. However, it entirely depends on the HTML of the webpage and may incorrectly classify the phishing webpages if the attacker changes all webpage resource references (i.e., Java script, CSS, images, etc.). Rao and Pais25 proposed a two-level anti-phishing technique called Black Phishing. At first level, a blacklist of signatures is created using visual similarity based features (i.e., file names, paths, and screenshots) rather than using blacklist of URLs. At second level, heuristic features are extracted from URL and HTML to identify the phishing websites which override the first level filter. In spite of that, the legitimate websites always undergo two-level filtering. In some researches26 authors used search engine-based mechanism to authenticate the webpage as first-level authentication. In the second level authentication, various hyperlinks within the HTML of the website are processed for the phishing websites detection. Although the use of search engine-based techniques increases the number of legitimate websites correctly identified as legitimate, it also increases the number of legitimate websites incorrectly identified as phishing when newly created authentic websites are not found in the top results of search engine. Search based approaches assume that genuine website appears in the top search results.

In a recent study, Rao et al.27 proposed a new phishing websites detection method with word embedding extracted from plain text and domain specific text of the html source code. They implemented different word embedding to evaluate their model using ensemble and multimodal techniques. However, the proposed method is entirely dependent on plain text and domain specific text, and may fail when the text is replaced with images. Some researchers have tried to identify phishing attacks by extracting different hyperlink relationships from webpages. Guo et al.28 proposed a phishing webpages detection approach which they called HinPhish. The approach establishes a heterogeneous information network (HIN) based on domain nodes and loading resources nodes and establishes three relationships between the four hyperlinks: external link, empty link, internal link and relative link. Then, they applied an authority ranking algorithm to calculate the effect of different relationships and obtain a quantitative score for each node.

et al.6 work, the distributed representation of words is adopted within a specific URL, and then seven various machine learning classifiers are employed to identify whether a suspicious URL is a phishing website. Rao et al.13 proposed an anti-phishing technique called Catch Phish. They extracted hand-crafted and Term Frequency-Inverse Document Frequency (TF-IDF) features from URLs, then trained a classifier on the features using random forest algorithm. Although the above methods have shown satisfactory performance, they suffer from the following restrictions: (1) inability to handle unobserved characters because the URLs usually contain meaningless and unknown words that are not in the training set; (2) they do not consider the content of the website. Accordingly, some URLs, which are distinctive to others but imitate the legitimate sites, may not be identified based on URL string. As their work is only based on URL features, which is not enough to detect the phishing websites. However, we have provided an effective solution by proposing our approach to this domain by utilizing three different types of features to detect the phishing website more efficiently. Specifically, we proposed a hybrid feature set consisting of URL character sequence, various hyperlinks information, and textual content-based features.

Deep learning methods have been used for phishing detection e.g., Convolutional Neural Network (CNN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), and Recurrent Convolutional Neural Networks (RCNN) due to the success of the Natural Language Processing (NLP) attained by these techniques. However, deep learning methods are not employed much in phishing detection due to the inclusive training time. .3 proposed a phishing detection approach with a character level convolutional neural network based on URL. The proposed approach was compared by using various machine and deep learning algorithms, and different types of features such as TF-IDF characters, count vectors, and manually-crafted features. Le et al.29 provided a URL Net method to detect phishing webpage from URL. They extract character-level and word-level features from URL strings and employ CNN networks for training and testing. Chatterjee and Namin30 introduced a phishing detection technique based on deep reinforcement learning to identify phishing URLs. They used their model on a balanced, labeled dataset of benign and phishing URLs, extracting 14 hand-crafted features from the given URLs to train the proposed model. In recent studies, Xiao et al.31 proposed phishing website detection approach named CNN–MHSA. CNN network is applied to extract characters features from URLs. In the meanwhile, multi-head self-attention (MHSA) mechanism is employed to calculate the corresponding weights for the CNN learned features. Zheng et al.32 proposed a new Highway Deep Pyramid Neural Network (HDP-CNN) which is a deep convolutional network that integrates both character-level and word-level embedding representation to identify whether a given URL is phishing or legitimate. Albeit the above approaches have shown valuable performances, they might misclassify phishing websites hosted on compromised servers since the features are extracted only from the URL of the website.

The features extracted in some previous studies are based on manual work and require additional effort since these features need to be reset according to the dataset, which may lead to overfitting of anti-phishing solutions. We got the motivation from the above-mentioned studies and proposed our approach. In which, the current work extract character sequences feature from URL without manual intervention. Moreover, our approach employs noisy data of HTML, plaintext, and hyperlinks information of the website with the benefit of identifying new phishing websites. Table 1 presents the detailed comparison of existing machine learning based phishing detection approaches

**Proposed approach**

Our approach extracts and analyzes different features of suspected webpages for effective identification of large-scale phishing offenses. The main contribution of this paper is the combined uses of these feature set. For improving the detection accuracy of phishing webpages, we have proposed eight new features. Our proposed features determine the relationship between the URL of the webpage and the webpage content.

**System architecture**

The overall architecture of the proposed approach is divided into three phases. In the first phase, all the essential features are extracted and HTML source code will be crawled. The second phase applies feature vectorization to generate a particular feature vector for each webpage. The third phase identifies if the given webpage is phishing. Figure 1 shows the system structure of the proposed approach. Details of each phase are described as follows.

**Feature generation**

The features are generated in this component. Our features are based on the URL and HTML source code of the webpage. A Document Object Model (DOM) tree of the webpage is used to extract the hyperlink and textual content features using a web crawler automatically. The features of our approach are categorized into four groups as depicted in Table 2. In particular, features F1–F7, and F14 are new and proposed by us; Features F8–F13, and F15 are taken from other approaches9,11,12,24,33 but we adjusted them for better results. Moreover, the observational method and strategy regarding the interpretation of these features are applied differently in our approach. A detailed explanation of the proposed features is provided in the feature extraction section of this paper.

**Features extraction**

Due to the limited search engine and third-party methods discussed in the literature, we extract the particular features from the client side in our approach. We have introduced eleven hyperlink features (F3–F13), two login form features (F14 and F15), character level TF-IDF features (F2), and URL character sequence features (F1). All these features are discussed in the following subsections.

URL character sequence features (F1)

The URL stands for Uniform Resource Locator. It is used for providing the location of the resources on the web such as images, files, hypertext, video, etc. URL. Each URL starts with a protocol (http, https, and ftp) used to access the resource requested. In this part, we extract character sequence features from URL. We employ the method used in35 to process the URL at the character level. More information is contained at the character level. Phishers also imitate the URLs of legitimate websites by changing many unnoticeable characters, e.g., “www.icbc.com” as “www.1cbc.com”. Character level URL processing is a solution to the out of vocabulary problem. Character level sequences identify substantial information from specific groups of characters that appear together which could be a symptom of phishing. In general, a URL is a string of characters or words where some words have little semantic meanings. Character sequences help find this sensitive information and improve the efficiency of phishing URL detection. During the learning task, machine learning techniques can be applied directly using the extracted character sequence features without the expert intervention. The main processes of character sequences generating include: preparing the character vocabulary, creating a tokenizer object using Keras preprocessing package (https://Keras.io) to process URLs in char level and add a “UNK” token to the vocabulary after the max value of chars dictionary, transforming text of URLs to sequence of tokens, and padding the sequence of URLs to ensure equal length vectors. The description of URL features extraction is shown in Algorithm 1.

**HTML features**

The webpage source code is the programming behind any webpage, or software. In case of websites, this code can be viewed by anyone using various tools, even in the web browser itself. In this section, we extract the textual and hyperlink features existing in the HTML source code of the webpage.

**Textual content-based features (F2)**

TF-IDF stands for Term Frequency-Inverse Document Frequency. TF-IDF weight is a statistical measure that tells us the importance of a term in a corpus of documents36. TF-IDF vectors can be created at various levels of input tokens (words, characters, n-grams) 37. It is observed that TF-IDF technique has been implemented in many approaches to catch phish of webpages by inspecting URLs 13, obtain the indirect associated links38, target website11, and validity of suspected website 39. In spite of TF-IDF technique extracts outstanding keywords from the text content of the webpage, it has some limitations. One of the limitations is that TF-IDF technique fails when the extracted keywords are meaningless, misspelled, skipped or replaced with images. Since plaintext and noisy data (i.e., attribute values for div, h1, h2, body and form tags) are extracted in our approach from the given webpage using Beautiful Soup parser, TF-IDF character level technique is applied with max features as 25,000. To obtain valid textual information, extra portions (i.e., JavaScript code, CSS code, punctuation symbols, and numbers) of the webpage are removed through regular expressions, including Natural Language Processing packages (http://www.nltk.org/nltk\_data/) such as sentence segmentation, word tokenization, text lemmatization and stemming.

Phishers usually mimic the textual content of the target website to trick the user. Moreover, phishers may mistake or override some texts (i.e., title, copyright, metadata, etc.) and tags in phishing webpages to bypass revealing the actual identification of the webpage. However, tag attributes stay the same to preserve the visual similarity between phishing and targeted site using the same style and theme as that of the benign webpage. Therefore, it is needful to extract the text features (plaintext and noisy part of HTML) of the webpage. The basic of this step is to extract the vectored representation of the text and the effective webpage content. A TF-IDF object is employed to vectorize text of the webpage. The detailed process of the text vector generation algorithm as follows.

**Referance:**

on web phishing detection

Rami, M., McCluskey, L. and Fadi, T. (2015a) Phishing Website Dataset: UCI Machine Learning Repository. Available at: <https://archive.ics.uci.edu/ml/machine-learning-databases/00327/Training%20Dataset.arff> (Accessed: 12 September 2017).

Rami, M., Thabtah, F. A. and T.L., M. (2015b) Phishing Websites Dataset. Available at: <http://eprints.hud.ac.uk/24330/9/Mohammad14JulyDS_1.arff> (Accessed: 10 September 2017). References (44) R. Stathacopoulou et al. 'Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis' Information Sciences (2005) N.M. Shekokar et al.

'An ideal approach for detection and prevention of phishing attacks' Procedia Computer Science (2015) M. Moghimi et al.

'New rule-based phishing detection method' Expert Systems with Applications (2016) J. Kim et al. 'HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems' Neural Networks (1999) D. Karaboga et al. 'An adaptive and hybrid artificial bee colony algorithm (ABC) for ANFIS training' Applied Soft Computing (2016) G. Hripcsak et al.

'Agreement, the f-measure, and reliability in information retrieval' Journal of the American Medical Informatics Association (2005) P.A. Barraclough et al.

'Intelligent phishing detection and protection scheme for online transactions. (Report)' Expert Systems With Applications (2013) N.A.G. Arachchilage et al.

'Phishing threat avoidance behaviour: An empirical investigation' Computers in Human Behavior (2016) M. Aburrous et al.

'Intelligent phishing detection system for e-banking using fuzzy data mining' Expert Systems with Applications (2010) A. Abraham 'Adaptation of fuzzy inference system using neural learning' Fuzzy systems engineering (2005) A. Abunadi et al.

'Feature extraction process: A phishing detection approach' Aburrous, M., Hossain, M. A., Thabatah, F., & Dahal, K. (2008).

Intelligent Phishing Website Detection System using... Unifying the global response to cybercrime [Online] (2017) L.D.D. Babu et al.

'A Survey of Methodologies and Techniques for Detection and Prevention of Phishing Attacks' International Journal of Advanced Research in Computer Science (2010) P. Barraclough et al.

'Phishing website detection fuzzy system modelling' E. Çakıt et al. 'Predicting the occurrence of adverse events using an adaptive neuro-fuzzy inference system (ANFIS) approach with the help of ANFIS input selection' Artificial Intelligence Review (2016) Cohen, A., Nissim, N., Rokach, L., & Elovici, Y. (2016).

'SFEM: Structural Feature Extraction Methodology for the... A. Cruz et al. 'ANFIS: Adaptive neuro-fuzzy inference systems' IM, UFRJ, Mestrado NCE (2009) A.

Dariane et al. 'Forecasting streamflow by a combination of a genetic input selection algorithm and wavelet transforms using ANFIS models' Journal of Hydrological Sciences (2016) M. Deshmukh et al.

'Different Techniques for Detection of Phishing Attack' International Journal of Engineering Science (2017) A.J. Elliot et al.

'Color and psychological functioning: The effect of red on performance attainment.' Journal of Experimental Psychology (2007) J.C.S. Fatt et al. Leverage Website Favicon to Offset

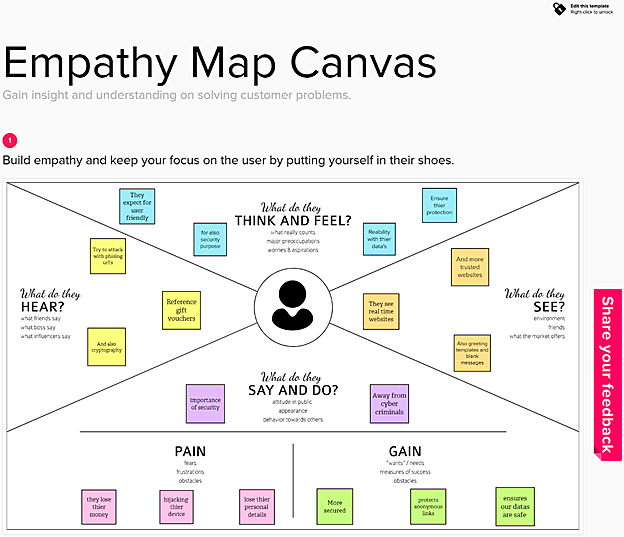
Existing problem

Polymorphic Phishing Phishing offenses are increasing, resulting in billions of dollars in loss1. In these attacks, users enter their critical (i.e., credit card details, passwords, etc.) to the forged website which appears to be legitimate. The Software-as-a-Service (SaaS) and webmail sites are the most common targets of phishing2. The phisher makes websites that look very similar to the benign websites. The phishing website link is then sent to millions of internet users via emails and other communication media. These types of cyber-attacks are usually activated by emails, instant messages, or phone calls3. The aim of the phishing attack is not only to steal the victims' personality, but it can also be performed to spread other types of malware such as ransomware, to exploit approach weaknesses, or to receive monetary profits4. According to the Anti-Phishing Working Group (APWG) report in the 3rd Quarter of 2020, the number of phishing attacks has grown since March, and 28,093 unique phishing sites have been detected between July to September2. The average amount demanded during wire transfer Business E-mail Compromise (BEC) attacks was $48,000 in the third quarter, down from $80,000 in the second quarter and $54,000 in the first. Detecting and preventing phishing offenses is a significant challenge for researchers due to the way phishers carry out the attack to bypass the existing anti-phishing techniques. Moreover, the phisher can even target some educated and experienced users by using new phishing scams. Thus, software-based phishing detection techniques are preferred for fighting against the phishing attack. Mostly available methods for detecting phishing attacks are blacklists/whitelists5, natural language processing6, visual similarity7, rules8, machine learning techniques 9,10, etc. Techniques based on blacklists/whitelists fail to detect unlisted phishing sites (i.e. 0-h attacks) as well as these methods fail when blacklisted URL is encountered with minor changes. In the machine learning based techniques, a classification model is trained using various heuristic features (i.e., URL, webpage content, website traffic, search engine, WHOIS record, and Page Rank) in order to improve detection efficiency. However, these heuristic features are not warranted to present in all phishing websites and might also present in the benign websites, which may cause a classification error. Moreover, some of the heuristic features are hard to access and third-party dependent. Some third-party services (i.e., page rank, search engine indexing, WHOIS etc.) may not be sufficient to identify phishing websites that are hosted on hacked servers and these websites are inaccurately identified as benign websites because they are contained in search results. Websites hosted on compromised servers are usually more than a day old unlike other phishing websites which only take a few hours. Also, these services inaccurately identify the new benign website as a phishing site due to the lack of domain age. The visual similarity-based heuristic techniques compare the new website with the pre-stored signature of the website. The website’s visual signature includes screenshots, font styles, images, page layouts, logos, etc. Thus, these techniques cannot identify the fresh phishing websites and generate a high false-negative rate (phishing to benign). The URL based technique does not consider the HTML of the webpage and may misjudge some of the malicious websites hosted on free or compromised servers. Many existing approaches11,12,13 extract hand-crafted URL based features, e.g., number of dots, presence of special “@”, “#”, “–” symbol, URL length, brand names in URL, position of Top-Level domain, check hostname for IP address, presence of multiple TLDs, etc. However, there are still hurdles to extracting manual URL features due to the fact that human effort requires time and extra maintenance labor costs. Detecting and preventing phishing offense is a major defiance for researchers because the scammer carries out these offenses in a way that can avoid current anti-phishing methods. Hence, the use of hybrid methods rather than a single approach is highly recommended by the networks security manager. This paper provides an efficient solution for phishing detection that extracts the features from website's URL and HTML source code. Specifically, we proposed a hybrid feature set including URL character sequence features without expert’s knowledge, various hyperlink information, plaintext and noisy HTML data-based features within the HTML source code. These features are then used to create feature vector required for training the proposed approach by XG Boost classifier. Extensive experiments show that the proposed anti-phishing approach has attained competitive performance on real dataset in terms of different evaluation statistics. Our anti-phishing approach has been designed to meet the following requirements. High detection efficiency: To provide high detection efficiency, incorrect classification of benign sites as phishing (false-positive) should be minimal and correct classification of phishing sites (true-positive) should be high. Real-time detection: The prediction of the phishing detection approach must be provided before exposing the user's personal information on the phishing website. Target independent: Due to the features extracted from both URL and HTML the proposed approach can detect new phishing websites targeting any benign website (zero-day attack). Third-party independent: The feature set defined in our work are lightweight and client-side adaptable, which do not rely on third-party services such as blacklist/whitelist, Domain Name System (DNS) records, WHOIS record (domain age), search engine indexing, network traffic measures, etc. Though third-party services may raise the effectiveness of the detection approach, they might misclassify benign websites if a benign website is newly registered. Furthermore, the DNS database and domain age record may be poisoned and lead to false negative results (phishing to benign). Hence, a light-weight technique is needed for phishing websites detection adaptable at client side. The major contributions in this paper are itemized as follows. We propose a phishing detection approach, which extracts efficient features from the URL and HTML of the given webpage without relying on third-party services. Thus, it can be adaptable at the client side and specify better privacy. We proposed eight novel features including URL character sequence features (F1), textual content character level (F2), various hyperlink features (F3, F4, F5, F6, F7, and F14) along with seven existing features adopted from the literature. We conducted extensive experiments using various machine learning algorithms to measure the efficiency of the proposed features. Evaluation results manifest that the proposed approach precisely identifies the legitimate websites as it has a high true negative rate and very less false positive rate. We release a real phishing webpage detection dataset to be used by other researchers on this topic. The rest of this paper is structured as follows: The "Related work" section first reviews the related works about phishing detection. Then the "Proposed approach" section presents an overview of our proposed solution and describes the proposed features set to train the machine learning algorithms. The "Experiments and result analysis” section introduces extensive experiments including the experimental dataset and results evaluations. Furthermore, the "Discussion and limitation" section contains a discussion and limitations of the proposed approach. Finally, the "Conclusion" section concludes the paper and discusses future work. Related work This section provides an overview of the proposed phishing detection techniques in the literature. Phishing methods are divided into two categories; expanding the user awareness to distinguish the characteristics of phishing and benign webpages14, and using some extra software. Software-based techniques are further categorized into list-based detection, and machine learning-based detection. However, the problem of phishing is so sophisticated that there is no definitive solution to efficiently bypass all threats; thus, multiple techniques are often dedicated to restrain particular phishing offenses. List-based detection List-based phishing detection methods use either whitelist or blacklist-based technique. A blacklist contains a list of suspicious domains, URLs, and IP addresses, which are used to validate if a URL is fraudulent. Simultaneously, the whitelist is a list of legitimate domains, URLs, and IP addresses used to validate a suspected URL. Wang et al.15, Jain and Gupta5 and Han et al.16 use white list-based method for the detection of suspected URL. Blacklist-based methods are widely used in openly available anti-phishing toolbars, such as Google safe browsing, which maintains a blacklist of URLs and provides warnings to users once a URL is considered as phishing. Prakash et al.17 proposed a technique to predict phishing URLs called Phishnet. In this technique, phishing URLs are identified from the existing blacklisted URLs using the directory structure, equivalent IP address, and brand name. et al.18 developed a method that compares the domain name and name server information of new suspicious URLs to the information of blacklisted URLs for the classification process. Sheng et al.19 demonstrated that a forged domain was added to the blacklist after a considerable amount of time, and approximately 50–80% of the forged domains were appended after the attack was carried out. Since thousands of deceptive websites are launched every day, the blacklist requires to be updated periodically from its source. Thus, machine learning-based detection techniques are more efficient in dealing with phishing offenses. Machine learning-based detection Data mining techniques have provided outstanding performance in many applications, e.g., data security and privacy20, game theory21, blockchain systems22, healthcare23, etc. Due to the recent development of phishing detection methods, various machine learning-based techniques have also been employed6,9,10,13 to investigate the legality of websites. The effectiveness of these methods relies on feature collection, training data, and classification algorithm. The feature collection is extracted from different sources, e.g., URL, webpage content, third party services, etc. However, some of the heuristic features are hard to access and time-consuming, which makes some machine learning approaches demand high computations to extract these features. Jain and Gupta24 proposed an anti-phishing approach that extracts the features from the URL and source code of the webpage and does not rely on any third-party services. Although the proposed approach attained high accuracy in detecting phishing webpages, it used a limited dataset (2141 phishing and 1918 legitimate webpages). The same authors9 present a phishing detection method that can identify phishing attacks by analyzing the hyperlinks extracted from the HTML of the webpage. The proposed method is a client-side and language-independent solution. However, it entirely depends on the HTML of the webpage and may incorrectly classify the phishing webpages if the attacker changes all webpage resource references (i.e., Java script, CSS, images, etc.). Rao and Pais25 proposed a two-level anti-phishing technique called Black Phish. At first level, a blacklist of signatures is created using visual similarit based features (i.e., file names, paths, and screenshots) rather than using blacklist of URLs. At second level, heuristic features are extracted from URL and HTML to identify the phishing websites which override the first level filter. In spite of that, the legitimate websites always undergo two-level filtering. In some researches26 authors used search engine-based mechanism to authenticate the webpage as first-level authentication. In the second level authentication, various hyperlinks within the HTML of the website are processed for the phishing websites detection. Although the use of search engine-based techniques increases the number of legitimate websites correctly identified as legitimate, it also increases the number of legitimate websites incorrectly identified as phishing when newly created authentic websites are not found in the top results of search engine. Search based approaches assume that genuine website appears in the top search results. In a recent study, Rao et al.27 proposed a new phishing websites detection method with word embedding extracted from plain text and domain specific text of the html source code. They implemented different word embedding to evaluate their model using ensemble and multimodal techniques. However, the proposed method is entirely dependent on plain text and domain specific text, and may fail when the text is replaced with images. Some researchers have tried to identify phishing attacks by extracting different hyperlink relationships from webpages. Guo et al.28 proposed a phishing webpages detection approach which they called High Phish. The approach establishes a heterogeneous information network (HIN) based on domain nodes and loading resources nodes and establishes three relationships between the four hyperlinks: external link, empty link, internal link and relative link. Then, they applied an authority ranking algorithm to calculate the effect of different relationships and obtain a quantitative score for each node. In et al.6 work, the distributed representation of words is adopted within a specific URL, and then seven various machine learning classifiers are employed to identify whether a suspicious URL is a phishing website. Rao et al.13 proposed an anti-phishing technique called Catc Phish. They extracted hand-crafted and Term Frequency-Inverse Document Frequency (TF-IDF) features from URLs, then trained a classifier on the features using random forest algorithm. Although the above methods have shown satisfactory performance, they suffer from the following restrictions: (1) inability to handle unobserved characters because the URLs usually contain meaningless and unknown words that are not in the training set; (2) they do not consider the content of the website. Accordingly, some URLs, which are distinctive to others but imitate the legitimate sites, may not be identified based on URL string. As their work is only based on URL features, which is not enough to detect the phishing websites. However, we have provided an effective solution by proposing our approach to this domain by utilizing three different types of features to detect the phishing website more efficiently. Specifically, we proposed a hybrid feature set consisting of URL character sequence, various hyperlinks information, and textual content-based features. Deep learning methods have been used for phishing detection e.g., Convolutional Neural Network (CNN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), and Recurrent Convolutional Neural Networks (RCNN) due to the success of the Natural Language Processing (NLP) attained by these techniques. However, deep learning methods are not employed much in phishing detection due to the inclusive training time. Aljofey et al.3 proposed a phishing detection approach with a character level convolutional neural network based on URL. The proposed approach was compared by using various machine and deep learning algorithms, and different types of features such as TF-IDF characters, count vectors, and manually-crafted features. Le et al.29 provided a URLNet method to detect phishing webpage from URL. They extract character-level and word-level features from URL strings and employ CNN networks for training and testing. Chatterjee and Namin30 introduced a phishing detection technique based on deep reinforcement learning to identify phishing URLs. They used their model on a balanced, labeled dataset of benign and phishing URLs, extracting 14 hand-crafted features from the given URLs to train the proposed model. In recent studies, Xiao et al.31 proposed phishing website detection approach named CNN–MHSA. CNN network is applied to extract characters features from URLs. In the meanwhile, multi-head self-attention (MHSA) mechanism is employed to calculate the corresponding weights for the CNN learned features. Zheng et al.32 proposed a new Highway Deep Pyramid Neural Network (HDP-CNN) which is a deep convolutional network that integrates both character-level and word-level embedding representation to identify whether a given URL is phishing or legitimate. Albeit the above approaches have shown valuable performances, they might misclassify phishing websites hosted on compromised servers since the features are extracted only from the URL of the website. The features extracted in some previous studies are based on manual work and require additional effort since these features need to be reset according to the dataset, which may lead to overfitting of anti-phishing solutions. We got the motivation from the above-mentioned studies and proposed our approach. In which, the current work extract character sequences feature from URL without manual intervention. Moreover, our approach employs noisy data of HTML, plaintext, and hyperlinks information of the website with the benefit of identifying new phishing websites. presents the detailed comparison of existing machine learning based phishing detection approaches. Comparison of machine learning based phishing detection approaches. Full size table Proposed approach Our approach extracts and analyzes different features of suspected webpages for effective identification of large-scale phishing offenses.

**IDEATION & PROPOSED SOLUTION**

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

Empathy Map Canvas:



Ideation & Brainstorming :

See a finshed version of this template to kickstart your work. Need some inspiration? Open example Brainstorm & idea prioritization Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room. 2-8 people recommended 10 minutes to prepare 1 hour to collaborate Template Share template feedback Team gathering Define who should participate in the session and send an invite. Share relevant information or pre-work ahead. Set the goal Think about the problem you'll be focusing on solving in the brainstorming session. A B Before you collaborate A little bit of preparation goes a long way with this session. Here’s what you need to do to get going. 10 minutes Learn how to use the facilitation tools Use the Facilitation Superpowers to run a happy and productive session. C Open article Define your problem statement What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm. 5 minutes 1 Key rules of brainstorming To run an smooth and productive session Stay in topic. Defer judgment. Go for volume. If possible, be visual. Listen to others. Encourage wild ideas. PROBLEM How might web phising detection using python & mechin e learning ? Brainstorm Write down any ideas that come to mind that address your problem statement. 10 minutes 2 You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing! TIP p.vaishnavi A web application that tests your knowledge on how you can escape phishing built on React An End-to-End Data Science Project: Website Phishing suspicious site detection using machine learning algorithms automated, dockerized phishing catcher person 1 automated, dockerized phishing catcher SmartiPhish integrated web browser for phishing free web browsing experience. Combining Long-term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites using URL Analysis of Phishing Detection Project using SHAP. karthikeyan person 2 A spam filter for Emails. Anti-Phishing Solution to Detect Spoofed Website Attacks suspicious site detection using machine learning algorithms Detecting phishing websites with only visual/textual characteristics, using Generative Adversarial Networks pavishkumar Classifying Phishing and Genuine Websites using Phishing data \*classifying a website as phishing or not in a form of a web app. Docker file has also been provided. Precise phishing detection with recurrent convolution neural network Tool for detecting malicious emails and SMTP mis configuration lakshmi dharshini person 4 Brainstorm Prioritize Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible. 20 minutes 4 Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard. TIP Feasibility Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.) If each of these tasks could get done without any cost, which would have the most positive impact? Importance Analysis of Phishing Detection Project using SHAP. A spam filter for E-mails. Anti-Phishing Solution to Detect Spoofed Website Attacks Anti-Phishing Solution to Detect Spoofed Website Attacks A spam filter for E-mails. Anti-Phishing Solution to Detect Spoofed Website Attacks After you collaborate You can export the mural as an image or pdf to share with members of your company who might find it helpful. Share the mural Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session. Quick add-ons Keep moving forward Strategy blueprint Define the components of a new idea or strategy. Customer experience journey map Understand customer needs, motivations, and obstacles for an experience. Strengths, weaknesses, opportunities & threats Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan. A Share template feedback Open the template Open the template Open the template Export the mural Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive. B Group ideas Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups. 20 minutes 3 Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural. TIP av web phising detection Classifying Phishing and Genuine Websites using Phishing data Detecting phishing websites with only visual/textual characteristics, using Generative Adversarial Networks Anti-Phishing Solution to Detect Spoofed Website Attacks Analysis of Phishing Detection Project using SHAP. A web application that tests your knowledge on how you can escape phishing built on React Smart Phish integrated web browser for phishing free web browsing experience.

Proposed Solution:

|  |  |
| --- | --- |
| Problem Statement (Problem to be solved) | *Web Phishing Detection* |
| Idea / Solution description | Never justify the anonymous link is the best solution for this phishing. |
| Novelty / Uniqueness | To justify the crime in digital way. |
| Social Impact / Customer Satisfaction | The customer can affected by losing the details of them and lost their money. |
| Business Model (Revenue Model) | The major business model for the problem is that cyber security. |
| Scalability of the Solution | The scalability of the solution is about the short duration. |

**Problem Solution fit:**

1. 10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, 8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. 4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design. solves a problem and matches customer behaviour. What kind of actions do customers take online? Extract online channels from #7 Project Title: WEBPHISHINGDETECTIN Project Design Phase-I- Solution Fit Template Team ID: PNT2022TMID32234 The customer of Bank's are the customers of the phishing methods. Those who are using the bank's website to the transaction of money from one place to another are the customers. The customers don't want to click the unwanted links. The people who are using the bank site's to transfer the money to transfer from one place to another have to verify the website which they are using. coupons and gift voucher are the major thing which triggers the customers to get into the phishing site's. BEFORE : Coupons and gift voucher are make the customer happy. AFTER: A client can feel vulnerable. By making the customers to avoid using the unwanted websites to transaction and and verifying the websites for once again will prevent the customers. Making limits of using or clicking the unwanted phishing site's. The real reason is that phishers can get the details of an customers while using bank site's. This causes the major problem to the people. The customer have to verify the site's before entering for transaction. The Available solutions for this phishing websites is don't want click any anonymous link. Don't want to install any unofficial software. By using this software applications, these phishing websites can be found easily. After verifying those site's the customer can use these kind of websites. 8.1: In online mode, the customer can report to the bank. 8.2: Customer can visit the bank and they can give the complain about the transaction of money. If we are in that place, we will check the twice. This is the solution for the phishing websites. I

**REQUIREMENT ANALYSIS:**

Functional requirement

1. Non-Functional requirements

**Functional requirement:**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | Cross Checking | Through External Website |
| FR-4 | User Status Verified | To verify the safety precautions |
|  | User Accessibility | Fetching the information |

**Non-Functional requirements:**

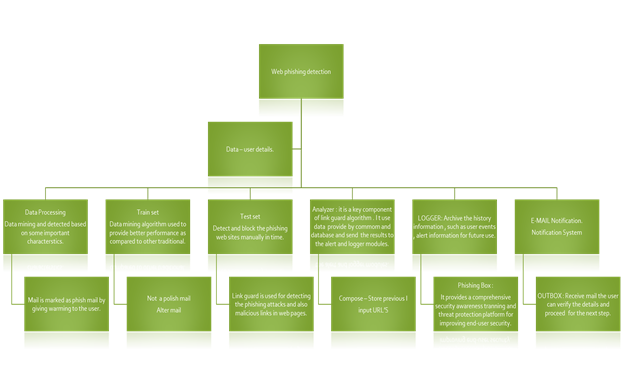
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The customer can more kindly |
| NFR-2 | **Security** | The security is Spectaculous |
| NFR-3 | **Reliability** | Solid as a rock |
| NFR-4 | **Performance** | To rise to the occasion. |
| NFR-5 | **Availability** | The customer can easily access the website |
| NFR-6 | **Scalability** | To adapt to increase the demands |

**PROJECT DESIGN:**

1. Data Flow Diagram
2. Solution & Technical Architecture
3. User Stories

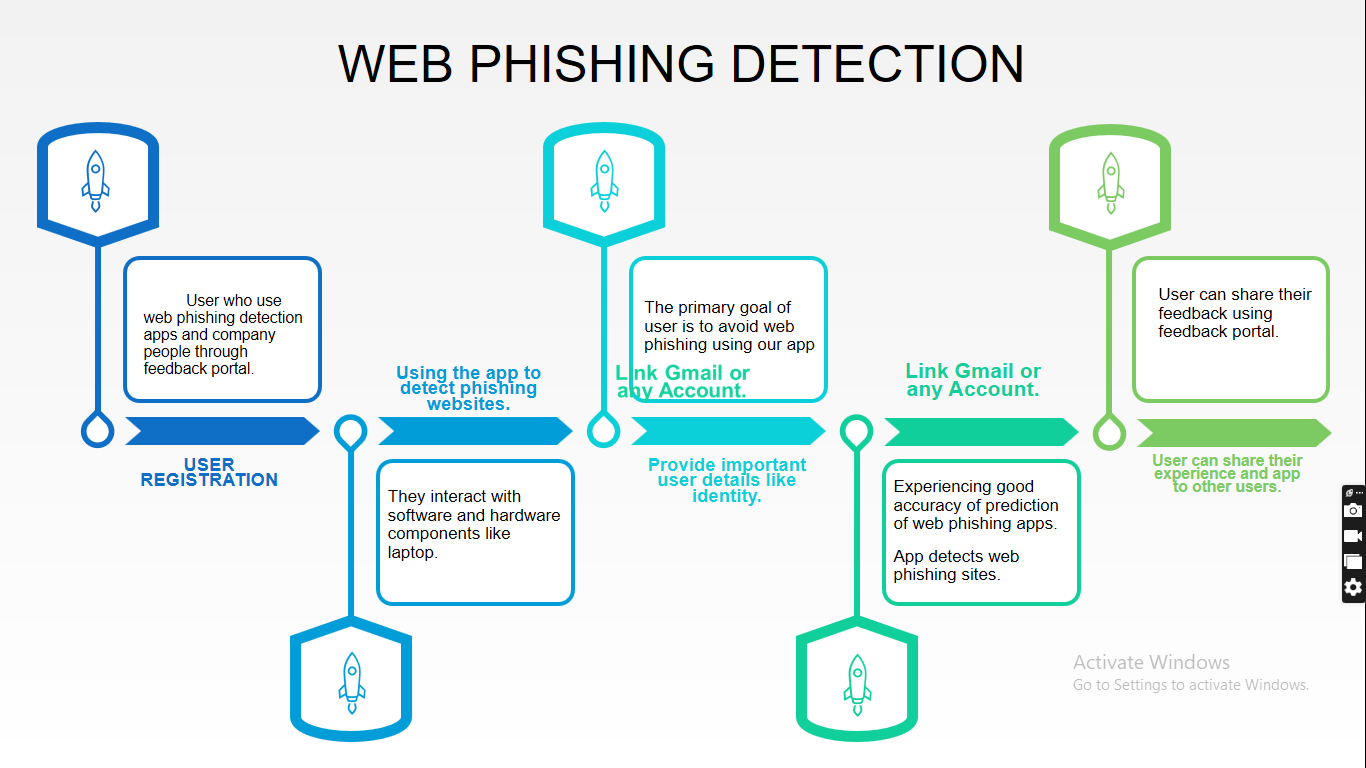
**Data Flow Diagrams:**

|  |
| --- |
|  |



|  |
| --- |
| **Solution & Technical Architecture :**    **User Stories :** |

|  |
| --- |
|  |



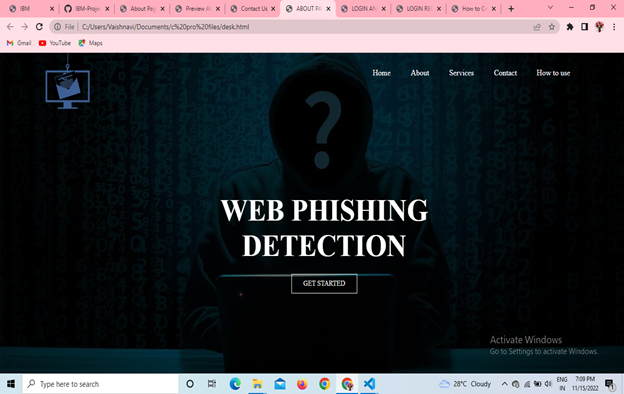
|  |
| --- |
|  |

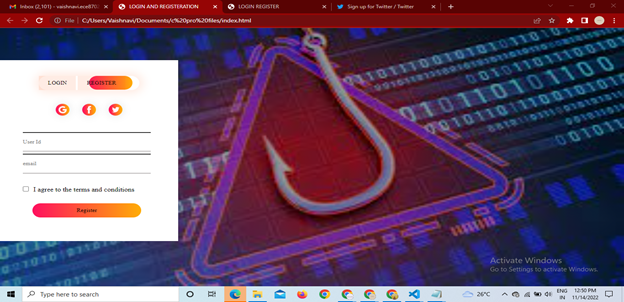
|  |
| --- |
|  |

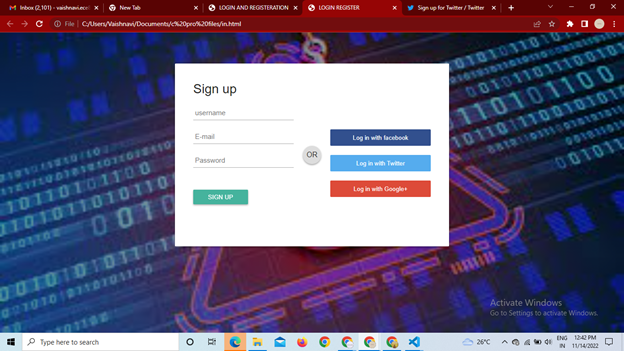
**PROJECT PLANNING & SCHEDULING :**

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

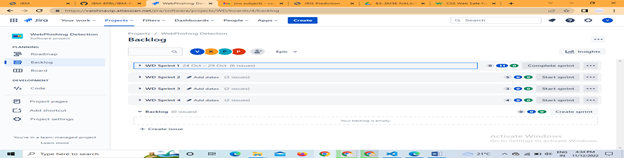
**Sprint Planning & Estimation:**

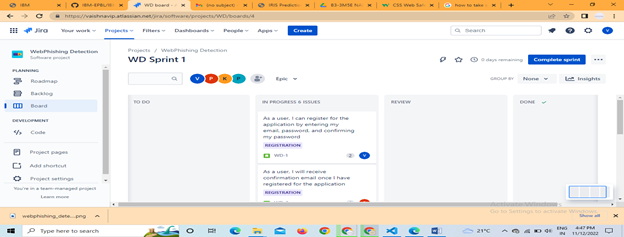


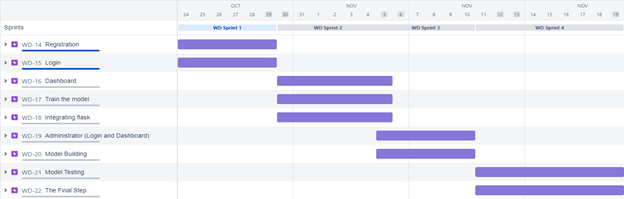


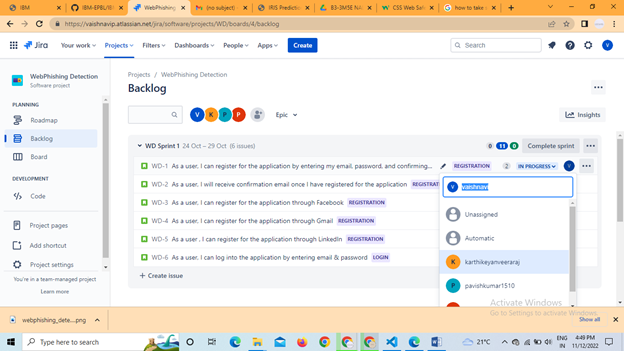


**Reports from JIRA :**









**CODING & SOLUTIONING (Explain the features added in the project along with code)**

1. Feature 1
2. Feature 2
3. Database Schema (if Applicable)

**Feature 1:**

|  |  |
| --- | --- |
|  | <html> |
|  | <head> |
|  | <title> |
|  | ABOUT PAGE |
|  | </title> |
|  | <link rel="stylesheet" href="desker.css"> |
|  | </head> |
|  | <body> |
|  | <header> |
|  | <div class="wrapper"> |
|  | <div class="logo"> |
|  | <img src="imgi.png" alt="#"> |
|  | </div> |
|  | <ul class="nav-area"> |
|  | <li><a href="#">Home</a></li> |
|  | <li><a href="#">About</a></li> |
|  | <li><a href="#">Services</a></li> |
|  | <li><a href="#">Contact</a></li> |
|  | <li> |
|  | <a href="#" title=" |
|  | 1. Login into the page |
|  | 2.After the login , click get started |
|  | 3.Paste the link and click predict |
|  | 4. Results will be display" data-bs-toggle="popover" data-bs-trigger="hover" data-bs-content="">How to use</a></li> |
|  |  |
|  | </ul> |
|  | </div> |
|  | <div class="welcome-text"> |
|  | <h1 style="color: white;">Web Phishing Detection</h1> |
|  | <a href="#">Get Started</a> |
|  | </div> |
|  | </header> |
|  | </body> |
|  |  |
|  | </html> |
|  |  |

**Feature 2:**

|  |  |
| --- | --- |
| <html> |  |
|  | <head> |
|  | <title>LOGIN REGISTER</title> |
|  | <link rel="stylesheet" href="cssFile.css"> |
| ]]-]p=]=;  ; | </head> |
|  |  |
|  | <body> |
|  | <div class="hero"> |
|  | <div id="login-box"> |
|  | <div class="left"> |
|  | <h1 id='status'>Sign up</h1> |
|  | <input type="text" id="username" name="username" placeholder="username" /> |
|  | <input type="text" id="email" name="email" placeholder="E-mail" /> |
|  | <input type="password" id="password" name="password" placeholder="Password" /> |
|  |  |
|  | <input type="submit" id='signUp' name="signup\_submit" value="Sign up" /> |
|  | </div> |
|  |  |
|  |  |
|  |  |
|  | <div class="right"> |
|  | <span class="loginwith">Sign in with<br />social network</span> |
|  |  |
|  | <button class="social-signin facebook"><a href="https://www.facebook.com"></a>Log in with facebook</button> |
|  | <button class="social-signin twitter">Log in with Twitter</button> |
|  | <button class="social-signin google">Log in with Google+</button> |
|  | </div> |
|  | <div class="or">OR</div> |
|  | </div> |
|  | </div> |
|  | </body> |
|  | <script type="module"> |
|  | // Import the functions you need from the SDKs you need |
|  | import { initializeApp } from "https://www.gstatic.com/firebasejs/9.13.0/firebase-app.js"; |
|  | import { getDatabase,set,ref } from "https://www.gstatic.com/firebasejs/9.13.0/firebase-database.js"; |
|  | import { getAuth, createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.13.0/firebase-auth.js"; |
|  |  |
|  | // TODO: Add SDKs for Firebase products that you want to use |
|  | // https://firebase.google.com/docs/web/setup#available-libraries |
|  |  |
|  | // Your web app's Firebase configuration |
|  | const firebaseConfig = { |
|  | apiKey: "AIzaSyA5eWN6Whx70Pry4\_1BhtgqbohIxqDZuEw", |
|  | authDomain: "authentication-app-d2784.firebaseapp.com", |
|  | databaseURL: "https://authentication-app-d2784-default-rtdb.firebaseio.com", |
|  | projectId: "authentication-app-d2784", |
|  | storageBucket: "authentication-app-d2784.appspot.com", |
|  | messagingSenderId: "134416282379", |
|  | appId: "1:134416282379:web:778736d7e2a45c4b6d680e" |
|  | }; |
|  |  |
|  | // Initialize Firebase |
|  | const app = initializeApp(firebaseConfig); |
|  | const database = getDatabase(app); |
|  | const auth= getAuth(); |
|  |  |
|  | signUp.addEventListener('click',(e) =>{ |
|  |  |
|  | var email = document.getElementById('email').value; |
|  | var password = document.getElementById('password').value; |
|  | var username = document.getElementById('username').value; |
|  |  |
|  |  |
|  | createUserWithEmailAndPassword(auth, email, password) |
|  | .then((userCredential) => { |
|  | // Signed in |
|  | const user = userCredential.user; |
|  | set(ref(database,'users/'+user.uid),{ |
|  | username: username, |
|  | email: email |
|  | }) |
|  | alert('user created'); |
|  | // ... |
|  | }) |
|  | .catch((error) => { |
|  | const errorCode = error.code; |
|  | const errorMessage = error.message; |
|  |  |
|  | alert('errorMessage'); |
|  | // .. |
|  | }); |
|  | }); |
|  | </script> |
|  | </html> |

**TESTING:**

* 1. Test Cases
  2. User Acceptance Testing

**Test Cases:**

4.6 Test Cases and Test Results For the URL verifier module in the ISOT phishing detection system, phishing detection is done using 16 different heuristic rules. In the system, 11 main classes were defined, and 1 class was defined with 5 sub-classes. This covers all 16 heuristic rules. To test the system, 15 test cases were designed using assertion methods. Ten test cases were designed to test the 10 main classes and 5 test cases were designed to test the class with five sub-classes. The getter-setter method was used to test the class with five sub-classes. The getter method is used to obtain or retrieve a variable value from the class, and the setter method is used to store the variables. The class with five sub-classes checks the 5 different heuristic rules, length of the URL, number of dots and slashes in the URL, presence of @ symbols in the URL, IP address mentioned in the URL, and the presence of special character such as ',', '\_', ';' in the URL. Initially, only a single test case was created for the class with five sub-classes, but it was failing as this class has five methods as. After applying the getter setter method, all the test cases passed without any issues. The test. assert Not Null () is used to check if the input URL is not empty, and assert Array Equals () is used to compare the result from the detection method with the expected result.

**User Acceptance Testing:**

**Need of User Acceptance Testing** arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

* Developers code software based on requirements document which is their “own” understanding of the requirements and **may not actually be what the client needs from the software**.
* Requirements changes during the course of the project may not be communicated effectively to the developers.

## *Acceptance Testing and V-Model*

In VModel, User acceptance testing corresponds to the requirement phase of the Software Development life cycle(SDLC).

### *Prerequisites of User Acceptance Testing:*

Following are the entry criteria for User Acceptance Testing:

* Business Requirements must be available.
* Application Code should be fully developed
* Unit Testing, Integration Testing & System Testing should be completed
* No Showstoppers, High, Medium defects in System Integration Test Phase –
* Only Cosmetic error is acceptable before UAT
* Regression Testing should be completed with no major defects
* All the reported defects should be fixed and tested before UAT
* Traceability matrix for all testing should be completed
* UAT Environment must be ready
* Sign off mail or communication from System Testing Team that the system is ready for UAT execution

## *How to execute UAT Tests*

UAT is done by the intended users of the system or software. This type of Software Testing usually happens at the client location which is known as Beta Testing. Once Entry criteria for UAT are satisfied, following are the tasks need to be performed by the testers:

UAT Process

* Analysis of Business Requirements
* Creation of UAT test plan
* Identify Test Scenarios
* Create UAT Test Cases
* Preparation of Test Data(Production like Data)
* Run the Test cases
* Record the Results
* Confirm business objectives

### *Step 1) Analysis of Business Requirements*

One of the most important activities in the UAT is to identify and develop test scenarios. These test scenarios are derived from the following documents:

* Project Charter
* Business Use Cases
* Process Flow Diagrams
* Business Requirements Document(BRD)
* System Requirements Specification(SRS)

### *Step 2) Creation of UAT Plan:*

The UAT test plan outlines the strategy that will be used to verify and ensure an application meets its business requirements. It documents entry and **exit criteria for UAT, Test scenarios and test cases approach and timelines of testing**.

### *Step 3) Identify Test Scenarios and Test Cases:*

Identify the test scenarios with respect to high-level business process and create test cases with clear test steps. Test Cases should sufficiently cover most of the UAT scenarios. Business Use cases are input for creating the test cases.

### *Step 4) Preparation of Test Data:*

It is best advised to use live data for UAT. Data should be scrambled for privacy and [security](https://www.guru99.com/ethical-hacking-tutorials.html) reasons. Tester should be familiar with the database flow.

### *Step 5) Run and record the results:*

Execute test cases and report bugs if any. Re-test bugs once fixed. [Test Management](https://www.guru99.com/test-management.html) tools can be used for execution.

### *Step 6) Confirm Business Objectives met:*

Business Analysts or UAT Testers needs to send a sign off mail after the UAT testing. After sign-off, the product is good to go for production. Deliverables for UAT testing are Test Plan, UAT Scenarios and Test Cases, Test Results and Defect Log

## *Exit criteria for UAT:*

Before moving into production, following needs to be considered:

* No critical defects open
* Business process works satisfactorily
* UAT Sign off meeting with all stakeholders

## *Qualities of UAT Testers:*

UAT Tester should possess good knowledge of the business. He should be independent and think as an **unknown user to the system**. Tester should be Analytical and Lateral thinker and combine all sort of data to make the UAT successful.

Tester or Business Analyst or Subject Matter Experts who understand the business requirements or flows can prepare test and data which are realistic to the business.

## *Best Practices:*

Following points needs to be considered to make UAT Success:

* Prepare UAT plan early in the project life cycle
* Prepare Checklist before the UAT starts
* Conduct Pre-UAT session during System Testing phase itself
* Set the expectation and define the scope of UAT clearly
* Test End to End business flow and avoid system tests
* Test the system or application with real-world scenarios and data
* Think as an Unknown user to the system
* Perform Usability Testing
* Conduct Feedback session and meeting before moving to production

## *UAT Tools*

There are several tools in the market used for User acceptance testing and some are listed for reference:

Fitness tool: It is a [Java](https://www.guru99.com/java-tutorial.html) tool used as a testing engine. It is easy to create tests and record results in a table. Users of the tool enter the formatted input and tests are created automatically. The tests are then executed and the output is returned back to the user.

: It is toolkit used to automate browser-based tests during User acceptance testing. Ruby is the programming language used for inter-process communication between ruby and Internet Explorer.

## *Example Guidelines for* UAT:

* Most of the times in regular software developing scenarios, UAT is carried out in the QA environment. If there is no staging or UAT environment
* UAT is classified into Beta and Alpha testing but it is not so important when software is developed for a service based industry
* UAT makes more sense when the customer is involved to a greater exten

**RESULTS:**

**[Quality Glossary Definition: Metrics](https://asq.org/quality-resources/quality-glossary/m)**

Performance metrics are defined as figures and data representative of an organization’s actions, abilities, and overall quality. There are many different forms of performance metrics, including sales, profit, return on investment, customer happiness, customer reviews, personal reviews, overall quality, and reputation in a marketplace. Performance metrics can vary considerably when viewed through different industries.

Performance metrics are integral to an organization's success. It's important that organizations select their chief performance metrics and focus on these areas because these metrics help guide and gauge an organization’s success. Key success factors are only useful if they are acknowledged and tracked. Business measurements must also be carefully managed to make sure that they give right answers, and that the right questions are being asked.

Traditionally, businesses have viewed the following financial measurements as indicators of success:

* Return on capital employed or return on investment (ROI)
* Profit
* Market share
* Earnings growth
* Stock price

Non-financial measurements are also useful to help assess, report, and drive success. Most notably, the [Malcolm Baldrige National Quality Award](https://asq.org/quality-resources/malcolm-baldrige-national-quality-award)'s Criteria for Performance Excellence non-financial success metrics include:

* [Customer satisfaction](https://asq.org/quality-resources/customer-satisfaction)
* Process excellence
* Employee satisfaction

Organizations across most industries rely on these indicators as well as:

* Fast, responsive time to market
* A loyal customer base
* Outstanding processes for quality and timeliness
* Mechanisms that ensure learning, growth, and [continual improvement](https://asq.org/quality-resources/continuous-improvement)

Organizations may define their own indicators of performance in key areas. Such metrics are often useful because they reduce complex measurements and results to a single value that can be tracked, managed, and improved. These “shortcuts” can be misleading, however, when used either for process improvement or for other feedback such as promotion, recognition, or compensation.

## WHAT IS METROLOGY?

[Metrology](https://asq.org/quality-resources/quality-glossary/m), or measurement science, contributes to business measurements as well as to more traditional engineering and scientific measurements.

The field of metrology has developed and fielded an approach—known as measurement assurance—that is analogous to product assurance in manufacturing. Measurement assurance uses management and [statistical techniques](https://asq.org/quality-resources/statistics) to:

* Evaluate the operation of a measurement system
* Ensure that it measures the desired quantities to the accuracy and precision required
* Monitor the performance of the measurement system

The following performance measurement necessities are the same whether you’re measuring business, service, process, or laboratory variables. Together, they constitute a measurement plan.

* **Definition of purpose:** Why is a measurement being made? What process or variable is being measured? For what will the resulting data be used?
* **Statement of the required measurement performance indicators (accuracy, precision, resolution):** These may be determined by organizational policy, adherence to a published standard or an analysis of the requirements based on use, ability to measure, or more.
* **The unit or variable being measured** and a statement as to why measuring that particular variable supports the purpose of the measurement.
* **An operational definition:** A detailed, yet easily understood, description of the measurement process.
  + Example: An operational definition for the measurement of a sales-fulfillment cycle time might be, “The time interval to be measured begins when the sales department places a validated order form in the sales order out box, and ends when the completed, boxed order is delivered to the loading dock for pickup.”
* **An analysis plan:** A typical example is a monthly report that makes comparisons to the previous month, year over year, and year to date. The different time frames provide greater context and allow the data to be presented graphically.
  + A [control chart](https://asq.org/quality-resources/control-chart) is a simple analysis plan template. It provides a graphical context that shows the continuity of changes over time, plus some analysis (control limits) that enables the viewer to differentiate among common causes, special causes, and random variation.

**[Driving Higher Workplace Performance: Using Analytics, Dashboard Metrics, and Soft Skills to Improve Results](https://asq.org/quality-resources/articles/case-studies/driving-higher-workplace-performance?id=33d5ebaa8e854c8eb934ce326834b2f0)** (PDF) When assigned the task of improving warehouse performance for a Western Canadian industrial distribution center, a Lean Six Sigma Black Belt discovered the differences between "human" and "automated" business processes.

**[Statistics Roundtable: Metrics for Uncertainty](https://asq.org/quality-progress/articles/metrics-for-uncertainty?id=dc9861e1447f43f7adda18538b21a1da)** (***Quality Progress***) A look at probability, evidence, and a seldom-used additive metric.

**[Measure for Measure: By Their Measures Shall Ye Know Them](https://asq.org/quality-progress/articles/column-measure-for-measure-by-their-measures-shall-ye-know-them?id=a966e01a2ee7473c9854970d7e5d93d3)** (***Quality Progress***) Measurements are so commonplace and universal they sometimes seem to be part of the air. They’re so usual, they’re not noticed. Still, they play a profound role in our lives and the way we behave.

**[The Education of a Metrologist](https://asq.org/quality-progress/articles/the-education-of-a-metrologist?id=83882d2624684c3f9de12e3e719db608)** (***Quality Progress***) Metrology is an interesting, detailed, and vast profession that allows members to focus on many different elements. Learn more about finding your way in this field with specific courses and training options from today’s top schools.

**[Measures of Software System Difficulty](https://asq.org/quality-resources/articles/case-studies/measures-of-software-system-difficulty?id=db1ce44dfac84bba84515ce1c2ca15c6)** (***Software Quality Professional***) Predicting and monitoring quality early in the software development life cycle can help provide initial estimates of software product quality.

**ADVANTAGES & DISADVANTAGES :**

The paper discuss with the importance of Phishing Detection Websites[5] and a review of various models in detection. Phishing attacks are significant threat to users of the Internet causing tremendous loss year by. The goal here is to combine the best aspects of human verified blacklists and heuristic-based methods which are have the low false positive rate of the owner. The key insight behind our detection algorithm is to define the existing human-verified blacklists and apply various techniques. The features introduced in Carnegie Mellon AntiPhishing and Network Analysis Tool (CANTINA)[3], in similarity feature to a machine learning based phishing detection. The heuristic detection model mainly makes the use of various characteristics of the URL that includes URL similarity calculation, domain name probability evaluation, IP address, the port number, etc. It will get the information of website ranking, registration information, category in which phishing websites and other information by querying the third party libraries such as Google Page rank. This will increase the phishing detection efficiency compared to older signature based models. The method can detect the various websites containing phishing attacks and abnormal behaviors. Web content, which is the main display channels for phishing fraud, well expresses the various intention of the website. The different machine learning module, Decision Tree, Support Vector Machine, Naive Bayes, Neural Network Feon Jaison et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014, pg. 696-699 © 2014, IJCSMC All Rights Reserved 697 and other machine learning algorithm have been applied into the model training and predicting whether the given website is phishing website or not. Clustering methods and classification are both fundamental tasks in Data Mining. Classification is used mostly as a supervised learning method, clustering for unsupervised learning. The goal of clustering is descriptive, that of classification is predictive. Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances belong to different groups. The instances are thereby categorized into an efficient representation. Formally, the clustering structure is represented as a set of sub-sets. Distance of clustering is the grouping of similar instances/objects. There are two main type of measures used to estimate this relation: distance measures and similarity measures. Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two instances xi and xj as d(xi,xj). Similarity Functions is considered as an alternative concept to that of the distance is the similarity function s(xi, xj) that compares the two vectors xi and xj. This function should be symmetrical i.e s(xi,xj)=s(xj,xi) and have a large value when xi and xj are somehow similar and constitute the largest value for identical vectors. Cosine Measure is the angle between the two vectors is a meaningful measure of their similarity, the normalized inner product may be an appropriate similarity measure: The existing phishing models describes as Blacklist/White list based method in which a user visits a Web site so that anti-phishing tool searches the address of that site in a blacklist stored in the database. If the visited site is present on the list, the anti-phishing tool will instruct the users. Tools in this category include Scam Blocker from the EarthLink Company, Phish Guard, and Net craft, etc. In the Rule-based method the various tools uses certain rules in their software, and checks the security of a Web site according to these rules. Examples of this type of tools include Spoof Guard developed by Stanford, Trust Watch of the Geo Trust, etc. Spoof Guard checks the domain name, URL of Web site, it also checks whether the browser is directed to the current URL through the links in the contents of e-mails. If it identifies that the domain name of the visited Web site is similar to a well-known domain name, or if they are not using the standard port, Spoof Guard will instruct the users about the phishing website. Both Spoof Guard and Trust Watch provide a toolbar in the browsers to notify their users whether the Web site is verified and trusted from phishing. The Intelligent architecture for the phishing website detection is that Feature Extraction module Heterogeneous Classifier that are built from the features extracted, Ensemble Classification Process, Hierarchical Clustering Algorithm for categorizing the various phishing websites. 1) Feature selection [1] is an important problem for pattern classification systems. The process includes how to select good features according to the maximal statistical dependency criterion based on mutual information. Because of the difficulty in directly implementing the maximal dependency condition, the procedure derives an equivalent form, called minimal redundancy-maximal-relevance criterion, for firstorder incremental feature selection. The process basically include selecting features from N samples .i.e. the feature selection problem is to find from M-dimensional observation space R(m),a subspace of m features. Selecting the features with highest relevance to target class C is de-fined as Max-relevance. Feon Jaison et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014, pg. 696-699 © 2014, IJCSMC All Rights Reserved 698 2) A naive Bayes classifier[4] considers all of these properties to independently contribute to the probability that this fruit is an apple. It only requires a small amount of training data to estimate the parameters necessary for classification, i.e. mean and variance The naive Bayes probabilistic model. The probability model for a classifier is a conditional model over a dependent class variable C with a small number of classes, conditional on several feature variables F1 through Fn is given by. 3) The best way to define statistical learning is called as supervised Learning. In this each data points consist of a vector of features denoted as x and a class label y, and it is assumed that there is some underlying function f such that y=f(x) for each training data point (x,y).The goal of learning algorithm is to find a good approximation h to f that can be applied to assign labels to new x values. The function h is called a classifier, because it assigns class labels y to input data points x. supervised learning can be applied to many problems including hand-writing recognition, medical diagnosis. Ordinary machine learning algorithms work by searching through a space of possible functions that is called as the hypothesis. To find one function h which is the best approximation to a unknown function f we uses the ensemble classifier process. To determine which hypothesis h is the best, a learning algorithm can measure how well h matches f on the training data points, and it can also assess how consistent h is with any variable prior to knowledge about the existing problem. 4) A clustering [2] is the data mining technique used to place data elements into related groups. The data mining analyze data in different perspective and classifies the data and summarize it into useful information. Analyzing is done using cluster analysis, Induction, Decision tree etc. This most popular methods to separate data into disjoint groups. Hierarchical clustering will measure the distance between 2 tuples and which specifies the dissimilarity in the sets as a function of the pair-wise distance.

**CONCLUSION**

Phishing websites are being designed to trick people into submitting credentials to access private information and assets by making the sites look like legitimate websites. Phishing attacks through URLs embedded in emails or SMS messages are one of the major issues faced by Internet users because of the huge number of online transactions performed daily. Phishing attacks cause losses to organizations, customers and Internet users. In this project, testing utilities were developed to support test automation for different aspects of the ISOT phishing detection system. In particular, two kind of test utilities were developed. 1. A mechanism to populate live test email accounts from different datasets, including spam, phishing, and legitimate emails, allowing online testing of the ISOT phishing detection system. 2. Automated test cases were used to unit test one of the modules of the ISOT phishing detection system. In this report, the data collection process for phishing, spam and legitimate emails was discussed. Further, the ISOT message security system was explained, and the design and implementation of the service to read the eml files from the collected datasets and send them to the test accounts was explained. The testing was done using the JUnit framework to check the functionality of the different heuristic rules of the system. The results of the unit tests were verified using assertion methods in the JUnit framework.

**FUTURE SCOPE:**

Future Work In the future, optimization can be done in the test units and these units can be made fully automated using Robot-Framework. This is important if more heuristics rules are included in the detection system. If the URL length is very long i.e. more than a million characters, then the system may crash. To prevent this situation, a timeout feature can be added when determining the URL length.

**APPENDIX:**

In this , we have found that the URL enters that they are phishing or not , By using these methods we got this idea.

By implementing this , This can be tested and implemented.