URL Based Phishing Website Detection Using Machine Learning

Donatus O. Njoku[1], Callistus T. Ikwuazom[2], Stanley A. Okolie[3], Janefrances E. Jibiri[4], Emmanuel C. Ololo[5], Kelechi Onyemachi[6]

[1]Dept. of Computer Science, Federal University of Technology Owerri, Nigeria
[2]Dept. of Information Technology, Federal University of Technology Minna, Nigeria
[3]Dept. of Computer Science, Federal University of Technology Owerri, Nigeria
[4]Dept. of Information Technology, Federal University of Technology Owerri, Nigeria
[5]Dept. of Computer Science, Imo State Polytechnic, Omuma, Imo State , Nigeria
[6]Dept. of Computer Science, Federal Polytechnic Nekede, Nigeria

Abstract—**Phishing attacks are one of the most common social engineering attacks targeting users' emails to fraudulently steal confidential and sensitive information. They can be used as a part of more massive attacks launched to gain a foothold in corporate or government networks. Over the last decade, a number of antiphishing techniques have been proposed to detect and mitigate these attacks. However, they are still inefficient and inaccurate. Thus, there is a great need for efficient and accurate detection techniques to cope with these attacks. In this paper, we proposed a phishing attack detection technique based on machine learning. We modeled these attacks by selecting 10 relevant features and building a large dataset. This dataset was used to train, validate, and test the machine learning algorithms. For performance evaluation, four metrics have been used, namely probability of detection, probability of miss-detection, probability of false alarm, and accuracy. The experimental results show that better detection can be achieved using an artificial neural network.**

Keywords—URL based, phishing, machine learning, algorithm, detection

1.0 INTRODUCTION

A. Background of the Study

Due to the rapid developments of the global networking and communication technologies, lots of our daily life activities such as social networks, electronic banking, e-commerce, etc. are transferred to the cyberspace. The open, anonymous and uncontrolled infrastructure of the Internet enables an excellent platform for cyberattacks, which presents serious security vulnerabilities not only for networks but also for the standard computer users even for the experienced ones. Although carefulness and experience of the user are important, it is not possible to completely prevent users from falling to the phishing scam [4]. Because, to increase the success of the phishing attacks, attackers also get into consideration about the personality characteristics of the end user especially for deceiving the relatively experienced users [10]. End-user-targeted cyberattacks cause massive loss of sensitive/personal information and even money for individuals whose total amount can reach billions of dollars in a year [15].

Phishing attacks' analogy is derived from "fishing" for victims, this type of attacks has attracted a great deal of attention from researchers in recent years. It is also a promising and attractive technique for attackers (also named as phishers) who open some fraudulent websites, which have exactly similar design of the popular and legal sites on the Internet. Although these pages have similar graphical user interfaces, they must have different Uniform Resource Locators (URLs) from the original page. Mainly, a careful and experienced user can easily detect these malicious web pages by looking at the URLs.

Phishing is defined as a a cybercrime in which a target or targets are contacted by email, telephone or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking and credit card details, and passwords. The information is then used to access important accounts and can result in identity theft and financial loss.

Machine learning based phishes detection gadget relies upon efficiently on the aspects of accuracy. The most of antiphishers researchers center of attention on optimizing new feature proposals or classification algorithms, where developing proper features analysis and selection techniques is not the important plan. The 12 features of this site are legitimate, phishing-enabled, reaching an effective positive rate of 97% and a false positive rate of 4%. The features are obtained by META tagging, web pages content, URLs, hyperlinks, TF-IDF, and more. Therefore, extraneous aspects might also nonetheless exist, which will increase the price of the technology (i.e. Training time, storage, electricity, etc.), however, it does not affect the average accuracy. Therefore, identifying a truly effective compact feature set requires an efficient Machine Learning based technique for Phishing detection.

The first phishing lawsuit was filed in 2004 against a Californian teenager who created the imitation of the website "America Online". With this fake website, he was able to gain sensitive information from users and access the credit card details to withdraw money from their accounts.

Other than email and website phishing, there's also 'vishing' (voice phishing), 'smishing' (SMS Phishing) and several other phishing techniques cybercriminals are constantly coming up with. The study wants to focus on the various ways phishing can be done and possible solutions to them in form of a machine learning based software.

B. Objectives of the Study

The primary aim of the work is to design a URL based phishing website detector. The specific objectives are:

- To develop a novel approach to detect malicious URL and alert users.
- To apply Machine Learning techniques in the proposed approach in order to analyze the real time URLs and produce effective results.
- Creating a reporting platform for other users of the platform to report fake websites in order to build the knowledge base.

- Studying previous work on the proposed topic and looking for ways to improve them.

2.0 LITERATURE REVIEW

C. Theoretical Framework

A theoretical framework for a URL-based phishing website detector would likely draw on concepts from the fields of computer science, information security, and human-computer interaction. The first component of the theoretical framework would be a technical understanding of how phishing attacks work and the methods that attackers use to spoof legitimate websites. This would include knowledge of techniques such as domain spoofing, URL redirects, and the use of malicious scripts or payloads. The second component would be an understanding of the psychological and social factors that make individuals susceptible to phishing attacks, such as trust in familiar brands or a willingness to provide personal information [15] This would inform the design of user interfaces and interactions that aim to educate and empower users to protect themselves against phishing. The third component would be the use of machine learning and data mining techniques to analyze and detect patterns in website URLs and other features that are indicative of phishing websites. This could include using models such as Random Forest, SVM and Neural Network.

The fourth component would be the use of browser extension or security software that can interact with the user's web browser to warn them of potentially malicious websites or block them entirely, and also providing feedback to machine learning model. The final component would be the evaluation of the effectiveness of the detector, through the use of datasets that contains both phishing and legitimate website URLs, and comparing the performance of the detector with existing state-of-the-art methods, [2]. By combining these various components, the theoretical framework would provide a comprehensive approach to detecting and defending against phishing attacks by using machine learning and user interface strategies, while also taking into account the social and psychological factors that make individuals susceptible to phishing scams.

D. Conceptual Framework

Phishing is the fraudulent attempt to obtain sensitive information or data, such as usernames, passwords, credit card numbers, or other sensitive details by impersonating oneself as a trustworthy entity in a digital communication [7]. Typically carried out by email spoofing, instant messaging, and text messaging, phishing often directs users to enter personal information at a fake website which matches the look and feel of the legitimate site. As of 2020, phishing is by far the most common attack performed by cyber-criminals, with the FBI's Internet Crime Complaint Centre recording over twice as many incidents of phishing than any other type of computer crime.

The first recorded use of the term "phishing" was in the cracking toolkit AOHell created by Koceilah Rekouche in 1995 [7] however it is possible that the term was used before this in a print edition of the hacker magazine *2600*, [1]. The word is a leetspeak variant of *fishing* (*ph* is a common replacement for *f*), probably influenced by phreaking, and alludes to the use of increasingly sophisticated lures to "fish" for users' sensitive information.

Attempts to prevent or mitigate the impact of phishing incidents include legislation, user training, public awareness, and technical security measures [11]. The types of phishing include:

- E-mail phishing: Email phishing is the general term given to any malicious email message meant to trick users into divulging private information. Attackers generally aim to steal account credentials, personally identifiable information (PII) and corporate trade secrets. However, attackers targeting a specific business might have other motives.
- Spear phishing: These email messages are sent to specific people within an organization, usually high-privilege account holders, to trick them into divulging sensitive data, sending the attacker money or downloading malware.
- Whaling and CEO fraud: These messages are typically sent to high-profile employees of a company to trick them into believing the CEO or other executive has requested to transfer money. CEO fraud falls under the umbrella of phishing, but instead of an attacker spoofing a popular website, they spoof the CEO of the targeted corporation.
- Voice phishing: Also known as Vishing, here attackers use voice-changing software to leave a message telling targeted victims that they must call a number where they can be scammed. Voice changers are also used when speaking with targeted victims to disguise an attacker's accent or gender so that they can pretend to be a fraudulent person
- SMS phishing: This type of phishing attack is also known as smishing. Using SMS messages, attackers trick users into accessing malicious sites from their smartphones. Attackers send a text message to a targeted victim with a malicious link that promises discounts, rewards or free prizes.
- Watering hole: A compromised site provides endless opportunities, so an attacker identifies a site used by numerous targeted users, exploits a vulnerability on the site, and uses it to trick users into downloading malware. With malware installed on targeted user machines, an attacker can redirect users to spoofed websites or deliver a payload to the local network to steal data [8]

E. Anti-Phishing Systems

Anti-phishing software consists of computer programs that attempt to identify phishing content contained in websites, e-mail, or other forms used to accessing data (usually from the internet) and block the content, usually with a warning to the user (and often a choice to view the content regardless). It is often integrated with web browsers and email clients as a toolbar that displays the important name for the web site the viewer is visiting, in an effort to prevent fraudulent websites from masquerading as other legitimate websites [16].

Most popular web browsers come with built-in anti-phishing and anti-malware protection services, but almost none of the alternate web browsers have such protections. [13] Password managers also can be wont to help defend against phishing, as can some mutual authentication techniques.

An independent study conducted by Carnegie Mellon University CyLab titled "Phinding Phish: An Evaluation of Anti-Phishing Toolbars" and released November 13, 2018 tested the power of ten anti-phishing solutions to block or warn about known phishing sites and not block or warn about legitimate sites (not exhibit false-positives), also because the usability of every solution. Of the solutions tested, Netcraft Toolbar, EarthLink ScamBlocker and SpoofGuard were able to correctly identify over 75% of the sites tested, with Netcraft Toolbar receiving the highest score without incorrectly identifying legitimate sites as phishing. Severe problems were however discovered using SpoofGuard, and it incorrectly identified 38% of the tested legitimate sites as phishing, resulting in the conclusion that "such inaccuracies might nullify the benefits SpoofGuard offers in identifying phishing sites." [12]. Google Safe Browsing (which has since been built into Firefox) and Internet Explorer both performed well, but when testing ability to detect fresh phishes Netcraft Toolbar scored as high as 96%, while Google Safe Browsing scored as low as 0%, possibly thanks to technical problems with Google Safe Browsing. The testing was performed using phishing data obtained from Anti-Phishing working party, PhishTank, and an unnamed email filtering vendor.

Another study, conducted by SmartWare for Mozilla and released November 14, 2018, concluded that the anti-phishing filter in Firefox was more effective than Internet Explorer by over 10%. The results of this study are questioned by critics, noting that the testing data was sourced exclusively from PhishTank, which itself is an anti-phishing provider. The study only compared Internet Explorer and Firefox, leaving out (among others) Netcraft Toolbar and therefore the Opera browser, both of which use data from PhishTank in their anti-phishing solutions. This has led to speculations that, with the limited testing data, both Opera and Netcraft Toolbar would have gotten an ideal score had they been a part of the study.

While the two directly aforementioned reports were released just one day apart, Asa Dotzler, Director of Community Development at Mozilla, has skilled the criticism of the Mozilla commissioned report by saying, "so you're agreeing that the most recent legitimate data puts Firefox ahead. Good enough for me." Since these studies were conducted, both Microsoft and Opera Software have started licensing Netcraft's anti-phishing data, bringing the effectiveness of their browser's built-in anti-phishing on par with Netcraft Toolbar and beyond [6].

F. Machine Learning

With machine learning algorithms, AI was able to develop beyond just performing the tasks it was programmed to do. Before Machine Learning entered the mainstream, AI programs were only used to automate low-level tasks in business and enterprise settings. This included tasks like intelligent automation or simple rule-based classification. This meant that AI algorithms were restricted to only the domain of what they were processed for. However, with machine learning, computers were able to move past doing what they were programmed and began evolving with each iteration.

Machine learning is fundamentally set apart from artificial intelligence, as it has the capability to evolve. Using various programming techniques, machine learning algorithms are able to process large amounts of data and extract useful information. In this way, they can improve upon their previous iterations by learning from the data they are provided, [9]

We cannot talk about machine learning without speaking about big data, one of the most important aspects of machine learning algorithms. Any type of AI is usually dependent on the quality of its dataset for good results, as the field makes use of statistical methods heavily. Machine learning is no exception, and a good flow of organized, varied data is required for a robust Machine learning solution. In today's online-first world, companies have access to a large amount of data about their customers, usually in the millions. This data, which is both large in the number of data points and the number of fields, is known as big data due to the sheer amount of information it holds.

Big data is time-consuming and difficult to process by human standards, but good quality data is the best fodder to train a machine learning algorithm. The more clean, usable, and machine-readable data there is in a big dataset, the more effective the training of the machine learning algorithm will be.

As explained, machine learning algorithms have the ability to improve themselves through training [14].

3. 0 METHODOLOGY AND SYSTEM ANALYSIS

A. Facts Finding

Fact finding is an approach taken to acquire data about a specific or subject with the aim of analyzing and synthesizing the analyzed data to come up with a better system. Fact finding for this study was done by examining related publications, research work, journals and books.

The phishing detection systems are generally divided into two groups: List Based Detection Systems and Machine Learning Based Detection Systems.

1. *List based detection systems:* List-based phishing detection systems use two list, whitelists and blacklists, for classifying the legitimate and phishing web pages. Whitelist-based phishing detection systems make secure and legitimate websites to provide the necessary information. Blacklists are created by URL records, which are known as phishing websites. These list entries are derived from a number of sources, such as spam detection systems, user notifications, third party organizations, etc. The use of blacklists makes it impossible for attackers to attack again via same URL or IP address, which are previously used for attack.

2. *Machine learning based detection systems:* One of the popular methods of malicious websites' detection is the use of machine learning methods. Mainly, detection of phishing attack is a simple classification problem. In order to develop a learning-based detection system, training data must contain lots of features, which are related to phishing and legitimate website classes. By the use of a learning algorithm, it can be easy to detect the unseen or not classified URLs with a dynamic mechanism.

B. Proposed System Design

The system as extensively described in previous chapters seeks to use the standard software development models which in this case is the Waterfall model, to create a standardized anti-phishing system. To achieve this goal above, we:

- Ensure that user details are kept secure.
- Ensure proper maintenance in terms of update of the knowledge base.
- Ensure only admins are granted admin a privilege access to affect the database tables.

C. Architectural Design of the Proposed System

This is where the programs that will run the modules identified in the control centre are specified. This will enable the researcher to capture the complete working picture of the application and how each component is related to another. The system architecture is shown below:

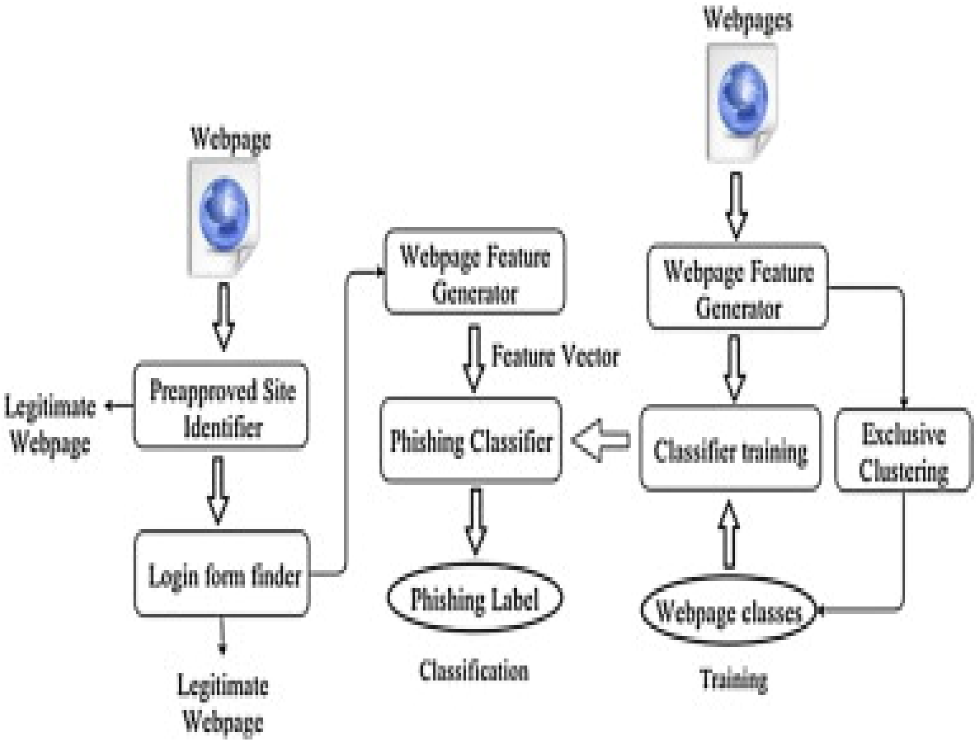

Fig. 1. Architectural design of the anti-phishing system

4.0 SYSTEM DESIGN AND IMPLEMENTATION

This chapter discusses the deployment and testing of the phishing detection system after the design and development. The Hardware and Software Requirements as well as Development tools are identified in this chapter.

A. *Objectives of design / overall system description*

- To accurately identify and classify phishing websites: The detector should use machine learning algorithms and data mining techniques to analyze website URLs and other features to accurately distinguish phishing websites from legitimate ones.
- To provide real-time protection: The detector should be integrated with a user's browser as an extension and operate in real-time, providing users with immediate warnings or blocks when they attempt to access a potentially malicious website.
- To be user-friendly: The detector's user interface should be intuitive and easy to understand, providing clear and concise warnings to users when a potentially malicious website is detected.
- To be efficient: The detector should be designed to be computationally efficient, using minimal system resources and memory to avoid slowing down the user's browsing experience.

- To improve over time: The detector should be designed to continuously improve its performance over time through the use of machine learning and feedback mechanisms’

The overall system description for such a detector would likely include the following components:

- URL Analysis: A module that analyzes website URLs using machine learning algorithms and data mining techniques to identify patterns and features that are indicative of phishing websites.
- Real-time protection: A module that integrates with a user's browser as an extension, providing real-time warnings or blocks when a potentially malicious website is detected.
- User interface: A module that provides a user-friendly interface for users to interact with the detector, receive warnings, and access educational resources.
- Feedback mechanism: A module that allows users to provide feedback on the detector's performance and provide feedback to the machine learning algorithm.
- Update mechanism: A module that allows detector to update itself with latest phishing websites and improve the performance over time.

B. Program / system design

1. *Data collection:* The first step would be to collect a dataset of both phishing and legitimate website URLs. This data would be used to train and test machine learning models.
2. *Feature extraction:* Next, the URLs would be preprocessed and features would be extracted, such as domain name, path, number of subdomains, presence of special characters, etc, which would be used as input to the machine learning model

3. Model development: Machine learning models such as Random Forest, SVM, Neural Network, etc. would be developed using the extracted features. The models would be trained and tested using the collected dataset.

4. *Model evaluation:* The performance of the models would be evaluated using metrics such as accuracy, precision, recall, F1 score, etc. to determine which model performs best.

5. *Integration with browser*: The detector would be integrated with a user's browser as an extension, providing real-time warnings or blocks when a potentially malicious website is detected.

The objectives of the design and overall system description for a URL-based phishing website detector would include the following:

- To accurately identify and classify phishing websites: The detector should use machine learning algorithms and data mining techniques to analyze website URLs and other features to accurately distinguish phishing websites from legitimate ones.

- To provide real-time protection: The detector should be integrated with a user's browser as an extension and operate in real-time, providing users with immediate warnings or blocks when they attempt to access a potentially malicious website.
- To be user-friendly: The detector's user interface should be intuitive and easy to understand, providing clear and concise warnings to users when a potentially malicious website is detected.
- To be efficient: The detector should be designed to be computationally efficient, using minimal system resources and memory to avoid slowing down the user's browsing experience.
- To improve over time: The detector should be designed to continuously improve its performance over time through the use of machine learning and feedback mechanisms.

This approach would provide a systematic way to develop a URL-based phishing website detector that can effectively detect and protect against phishing attacks while also being user-friendly and efficient.

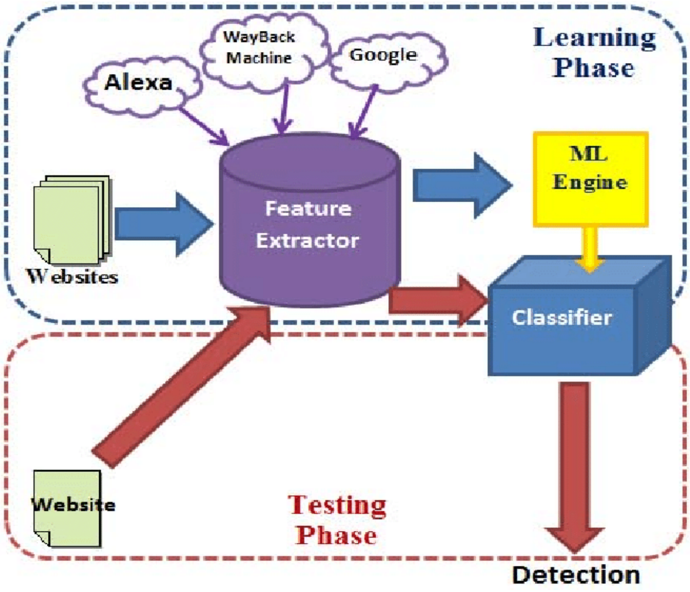

Fig. 2. System architecture

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

C. Algorithm

Natural language algorithm:

1. Take the URL of the website in question as input.
2. Check if the URL is on a list of known phishing websites.

    a. If the URL is found on the list, flag it as a phishing website and display a warning to the user.
    b. If the URL is not found on the list, proceed to the next step.

3. Check if the URL's domain name matches that of a known legitimate website, but with slight variations such as the addition or replacement of certain characters (e.g. "g00gle.com" instead of "google.com").
    a. If a match is found, flag it as a phishing website and display a warning to the user.
    b. If no match is found, proceed to the next step.
4. Check if the website has a valid SSL/TLS certificate.
    a. If the certificate is valid, proceed to the next step.
    b. If the certificate is invalid or missing, flag it as a phishing website and display a warning to the user.
    c. Perform a Google Safe Browsing check on the website.
5. If the website is found to be unsafe, flag it as a phishing website and display a warning to the user.
6. If the website is found to be safe, proceed to the next step.
    a. Check if the website has been reported as a phishing website by a reputable source.
7. If the website has been reported, flag it as a phishing website and display a warning to the user.
8. If the website has not been reported, proceed to the next step.
    a. Perform a machine learning classification on the website using a pre-trained model to detect phishing websites.
9. If the model detects the website as a phishing website, flag it as a phishing website and display a warning to the user.
10. If the model does not detect the website as a phishing website, flag it as safe.

D. Specification

The review of the existing system brought about identification of key areas that need to be improved on and they were also considered in the development of this project.

They include:

1. Data Validation
2. Speed and Reliability
3. Correctness
4. Understandability

E. Hardware and software requirements

Tables 4.1 and 4.2 identify the requirements both hardware and software required to successfully implement the system.

TABLE I. MINIMUM HARDWARE REQUIREMENTS

Minimum Hardware Requirements		
S/N	Server-Side Specification	Client-Side Specification
1	2GHz and above of CPU speed	2GHz and above of CPU speed
2	2GB and above of RAM	512MB and above of RAM
3	10 GB and above of hard disk space	512MB and above of hard disk space
4	Webserver (Apache)	Internet Connectivity
5	Database server (Sql)	

TABLE II. MINIMUM SOFTWARE REQUIREMENTS

Minimum Software Requirements		
	Server-Side Specification	Client-Side Specification
1	Windows OS	Windows OS
2	Apache	JavaScript enabled web browser
3	MySQL	

F. Communication Interfaces

Communication interfaces in this project include (and not limited to) TCP/IP (Transmission Control Protocol/Internet Protocol), HTTPS (Secured Hyper Text Transfer Protocol), FTP (File Transfer Protocol)

G. System Maintenance

Maintaining a machine learning-based phishing detector system that operates on URLs would involve several key steps to ensure its continued effectiveness. Some of these steps might include:

- Regularly updating the system's training data: As new phishing techniques are developed, the system's training data should be updated to reflect these changes so that it can continue to accurately detect new types of phishing attempts.
- Monitoring the system's performance: Regularly monitoring the system's performance metrics such as accuracy, false positive rate and false negative rate, allows to detect if there is any drift and retrain or fine-tune the model.
- Refining the system's parameters: As the system is used, its parameters may need to be adjusted to optimize its performance. This might involve adjusting the weights of different features used by the system, or tuning its threshold for classifying URLs as phishing or legitimate.
- Managing the system's infrastructure: Regularly updating and maintaining the underlying infrastructure of the system is important to ensure its continued reliability. This might involve patching security vulnerabilities, scaling the system to handle increasing traffic, and monitoring for potential system failures.

H. User Competence

The end user of this system just like any other web based application that employs a payment system, should be at least literate in English Language to be able to understand the options and the requests made from and to the server. The users should also have a basic understanding of internet security.

I. Experimental results

This section gives the experimental details of the proposed model's classification algorithms and used feature extraction types (NLP based features, Word Vectors, and Hybrid) are detailed.

1. *Used classification algorithms:* We have used seven different classification algorithms (Naive Bayes, Random Forest, kNN (n = 3), Adaboost, K-star, SMO and Decision Tree) as machine learning mechanism of the proposed system and then compared their performances. The Naïve Bayes classification is a probabilistic machine learning method, which is not only straightforward but also powerful. Due to its simplicity, efficiency and good performance, it is preferred in lots of application areas such as classification of texts, detection of spam emails/intrusions, etc. It is based on the Bayes theorem, which describes the relationship of conditional probabilities.

By the use of data preprocessing as detailed in previous sections, it can be easy to extract some distinctive features. These features are extracted by using the Natural Language Processing (NLP) operations. Therefore, these features depend on the used language. For the efficiency of the system, features are extracted according to the English language; however, according to aim it can be easily adapted to any language. Selection and design of these features are very trivial issues to accomplish, and most of the works focus on phishing detection used different feature list according to their algorithms. The selected features mainly need to parameterize the URL of the web page. Therefore, the text form of web address must be decomposed to the words that it contains. However, this is not an easy task. Because a web address can contain some combined texts in, which finding each word is a trivial task. In this decomposition operation, firstly the URL was parsed by taking into account some special characters such as ("?", "/", ".", "= ", "&").

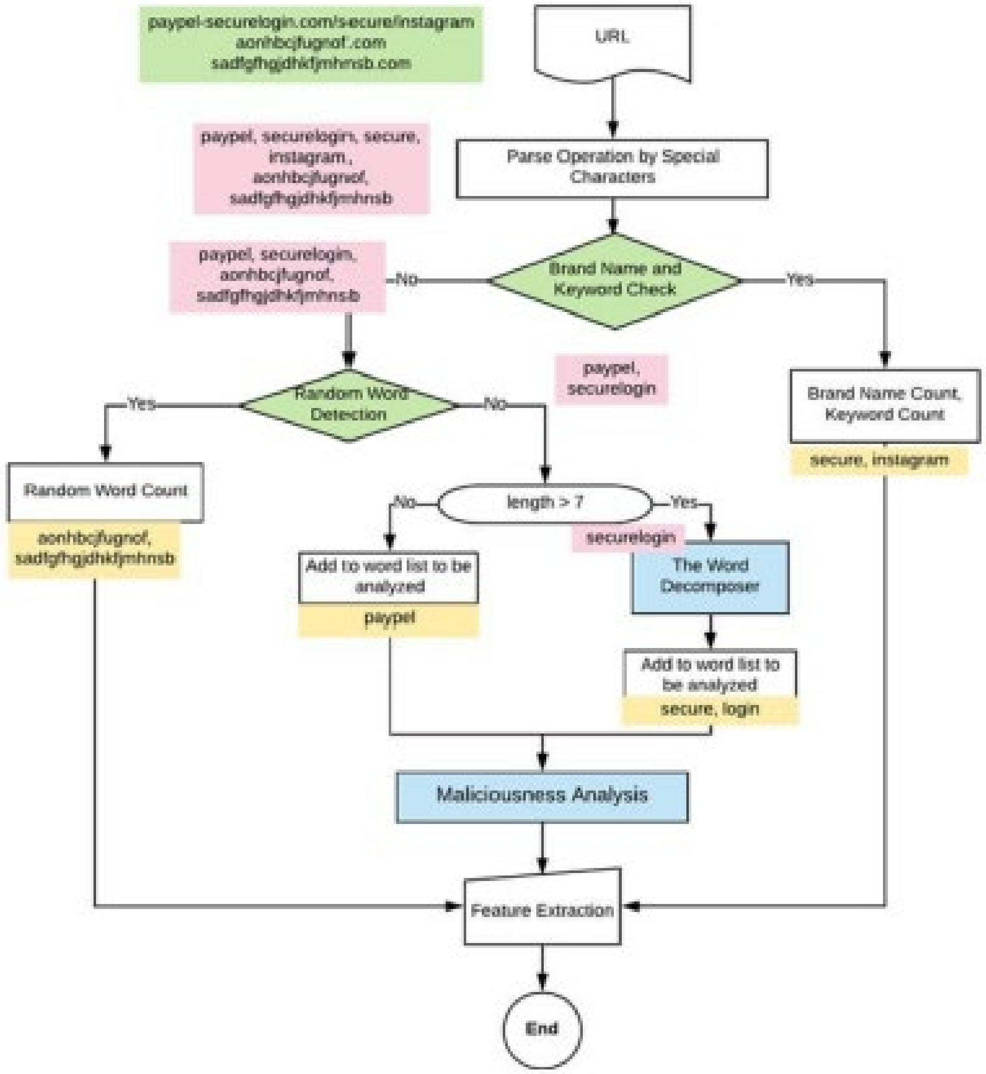
paypel-securelogin.com/secure/instagram
aonhbcjfugnof.com
sadfgfhgjdhkfjmhnsb.com
URL
paypel, securelogin, secure, instagram, aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
Parse Operation by Special Characters
paypel, securelogin, aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
No
Brand Name and Keyword Check
Yes
Yes
Random Word Detection
No
paypel, securelogin
Brand Name Count, Keyword Count
secure, instagram
Random Word Count
aonhbcjfugnof, sadfgfhgjdhkfjmhnsb
No
length > 7
Yes
securelogin
Add to word list to be analyzed
paypel
The Word Decomposer
Add to word list to be analyzed
secure, login
Maliciousness Analysis
Feature Extraction
End

Fig. 3. Execution of malicious analysis module

Then, a raw word list is reached in, which each word can have meaning alone or can be combined with the use of two or more distinct words in a meaningful order. The latter one is especially preferred for the attackers to convince the victim as if it is a legitimate web page. To deceive the users, attackers can use different techniques.

2. *Word Vectors:* In the text processing or text mining approaches, converting words into vectors is mostly preferred for reaching some crucial features. In our system, we are related to the URL of the web page, which is mainly constructed as a text that contains lots of words in it. Instead of converting these words manually, an automatic vectorization process is preferred. In this

module, each URL is converted into word vectors with the help of a specific function of Weka named as "StringtoWordVector". After obtaining the related vectors, they can be easily used in the selected machine learning algorithm. In the proposed system, 73,575 URLs are used for the testing. In the vectorization process, 1701 word-features are extracted. Then a feature reduction mechanism is applied to decrease the number of features in the list by using a feature selection algorithm named as "CfsSubsetEval"algorithm, which runs with the best first search method. With this reduction mechanism, the sufficient number of features has dropped from 1701 to 102.

3. *Hybrid features***:** To increase the efficiency of the proposed system we wanted to combine both features (NLP features and word vectors) in a hybrid model. After the implementation of the word vectorization step, we have totally 1701 word-features, and then we joined them with the 40 NLP features and there were 1741 total features before making a hybrid test. Then a similar feature reduction mechanism is executed, and the total number is decreased to 104 features.
4. *Test results:* One of the important problems for testing the proposed system is the use of a worldwide accepted dataset. We cannot reach this dataset, therefore, produced our own dataset as detailed. The dataset is also published in (Ebbu2017 Phishing Dataset, 2017). Due to its huge size and lack of test de- vice capacity, we have performed our test on this dataset, which contains 73,575 URLs. This dataset contains 36,400 legitimate URL and 37,175 phishing URLs. Experiments are executed on a MacBook Pro device with 2.7 GHz Intel Core i5 processor and 8 GB of 1867 MHz DDR3 RAM. For testing the proposed system Weka was used with some pre- developed libraries. 10-fold Cross Validation and the default parameter values of all algorithms were used during the tests. Each test set is executed with seven different machine learning algorithms. Firstly, the confusion matrix for the tested learning algorithms is constructed.

5.0 SUMMARY, CONCLUSION, AND RECOMMENDATIONS

A. Summary

A URL-based phishing website detector using machine learning is a system that uses machine learning algorithms to analyze the features of a website's URL and determine whether it is a legitimate website or a phishing website. These features can include the structure of the URL, the presence of certain keywords, and other characteristics that are commonly associated with phishing websites. The system is trained on a large dataset of both legitimate and phishing URLs, allowing it to learn the patterns and characteristics that differentiate the two. Once it is trained, the system can be used to automatically classify new URLs as legitimate or phishing. It is important to regularly update the system's training data and fine-tune the system to maintain its effectiveness in detecting new phishing techniques. Additionally, regular monitoring of the system's performance and compliance with laws and regulations is also essential

B. Conclusion

In conclusion, a URL-based phishing website detector using machine learning is a powerful tool for detecting and protecting against phishing attacks. By analyzing the features of a website's URL, the system can accurately identify and flag potential phishing websites, helping to protect users and organizations from falling victim to these types of attacks. However, it's important to keep in mind that machine learning-based phishing detectors are not foolproof, they require maintenance and monitoring to keep up to date with the latest phishing techniques. Additionally, regular monitoring of the system's performance and compliance with laws and regulations is also essential.

C. Recommendation

For future works, improvements can be made in terms of user identification and verification. Data security, data retrieval and fraud detection and reporting should be a vital consideration in development of any further web based machine learning systems.

Based on the analysis of existing and the proposed anti-phishing website detection systems, some recommendations for developing an anti-phishing website detection system will include:

- Incorporate multiple detection techniques: To increase the system's accuracy in identifying phishing websites, it should incorporate multiple detection techniques such as website structure analysis, content analysis, URL analysis, and reputation analysis.
- Use of Machine Learning techniques: To increase the system's ability to adapt to new and evolving phishing tactics, machine learning techniques such as supervised, unsupervised, and deep learning should be used.
- Incorporate real-time processing: To increase the system's effectiveness in blocking phishing attempts, it should incorporate real-time processing capabilities, which allow it to quickly identify and block phishing websites.
- Incorporate user feedback: To improve the system's accuracy, it should incorporate user feedback by allowing users to report potential phishing websites and receive alerts when a potential phishing website is detected.
- Incorporate browser extension: The system could also be integrated with browser extensions, to enable real-time identification and blocking of phishing websites while a user is browsing the internet.

REFERENCES

[1] Ramzan, Zulfikar (2019). "Phishing attacks and countermeasures". In Stamp, Mark; Stavroulakis, Peter (eds.). *Handbook of Information and Communication Security*. Springer. ISBN 978-3-642-04117-4.

[2] Van der Merwe, A J, Loock, M, Dabrowski, M. (2020), Characteristics and Responsibilities involved in a Phishing Attack, Winter International Symposium on Information and Communication Technologies, Cape Town, January 2020.

[3] "Landing another blow against email phishing (Google Online Security Blog)". (2021).

[4] Dudley, Tonia.(2019) "Stop That Phish". Archived from the original on 21 March 2021.

[5] "What is Phishing?". (2016). Archived from the original on 16 October 2016.

[6] "Internet Crime Report (2020)" (PDF). *FBI Internet Crime Complaint Centre*. U.S. Federal Bureau of Investigation. Retrieved 21 March 2021.

[7] Wright, A; Aaron, S; Bates, DW (2016). "The Big Phish: Cyberattacks Against U.S. Healthcare Systems". *Journal of General Internal Medicine*. **31** (10): 1115–8. doi:10.1007/s11606-016-3741-z. PMC 5023604. PMID 27177913.

[8] Ollmann, Gunter(2016). "The Phishing Guide: Understanding and Preventing Phishing Attacks". *Technical Info*. Archived from the original on 2011-01-31.

[9] Mitchell, Anthony (2020). "A Leet Primer". TechNewsWorld. Archived from the original on April 17, 2019

[10] "Phishing". *Language Log, September 22, 2019*. Archived from the original on 2016-08-30.

[11] Jøsang, Audun; et al. (2007). "Security Usability Principles for Vulnerability Analysis and Risk Assessment". *Proceedings of the Annual Computer Security Applications Conference 2017 (ACSAC'07)*. Archived from the original on 2021-03-21. Retrieved 2020-11-11.

[12] Aleksandersen, Daniel (16 August 2016). "Most of the alternate web browsers don't have fraud and malware protection". *Slight Future*. Retrieved 25 August 2016.

[13] Carnegie Mellon University(2016)"Phinding Phish: An Evaluation of Anti-Phishing Toolbars" (PDF). Archived from the original (PDF) on 2017-06-10. Retrieved 2018-05-25.

[14] Barraclough, P.A., Hossain, M.A., Tahir, M.A., Sexton, G., &Aslam, N. (2018) Intelligent Phishing Detection and Protection Scheme for Online Transactions. Expert Systems with Applications, 40, pp 4697-4706.

[15] Purkait, S. (2019) Phishing Counter Measures and Their Effectiveness – Literature Review.Information Management and Computer Security, 20 (5), pp 382-420.

[16] Ma, L., Ofoghi, B., Watters, P. & Brown, S. (2019) Detecting Phishing Emails Using Hybrid Features. Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, p. 493–497