Covenant Journal of Informatics & Communication Technology. Vol. 8 No. 1, June, 2020
ISSN: *print* 2354 – 3566 *electronics* 2354 – 3507 DOI:

An Open Access Journal Available Online

Approach for Identifying Phishing Uniform Resource Locators (URLs)

Nureni Azeez, Oluwaseyi Awotunde, Florence Oladeji

Department of Computer Sciences, Faculty of Science, University of Lagos, Lagos, Nigeria.
nurayhn1@gmail.com, seyi.juliana@gmail.com, foladeji@unilag.edu.ng

Received: 06.01.2020 Accepted: 14.05.2020
Date of Publication: June, 2020

Abstract—Phishing attacks are still very rampant and do not show signs of ever stopping. According to Santander Bank Customer Service, reports of phishing attacks have doubled each year since 2001. This work is based on identifying phishing Uniform Resource Locators (URLs). It focuses on preventing the issue of phishing attacks and detecting phishing URLs by using a total of 8 distinctive features that are extracted from the URLs. The sample size of study is 96,018 URLs. A total of four supervised machine learning algorithms: Naive Bayes Classifier, Support Vector Machine, Decision Tree and Random Forest were used to train the model and evaluate which of the algorithms performs better. Based on the analysis and evaluation, Random Forest performs best with an accuracy of 84.57% on the validation data set. The uniqueness of this work is in the choice of the selected features considered for the implementation.

Keywords/Index Terms—Cyber-attacks, Decision Tree, Phishing, Random Forest, Support Vector Machine

1. Introduction

Phishing is a cyber-attack carry out by fraudulent people to defraud people of their confidential information, login credentials and also finances. They do this for either their personal gain and this attack is not just done on individuals but also on Organizations. Phishers use legitimate sites to steal internet users' private and confidential information (Nureni and Irwin, 2010).

The term "Phishing" can be backdated to the early 1990s where a group of scammers came together and created an algorithm that allows them to generate random credit card numbers which they used to create accounts on America Online (AOL) (Adebowale, et. al., 2019). This was stopped by AOL but the "phishers" did not stop there but started pretending to be AOL employees and were messaging customers for their information. As people started becoming inclined to these scams, the group of scammers moved on to emails which was harder to track. They sent multiple emails to different people and robbed them of their information. These threats started becoming rampant and these scammers moved on from emails to other platforms and started hitting other major businesses.

Phishing is a really big and serious threat which keeps increasing year by year. In 2017, phishing attacks increased by 65% and over 1 million phishing sites were created. 76% of businesses were affected by these attacks in 2018 (Azeez, et. al., 2020). These few statistics go to show how serious and dangerous phishing is to the regular users, to businesses and organizations.

There are various types of phishing attacks and some of the popular ones are:

1. Spear phishing (Gupta, et. al., 2018): with this phishing attack, attackers pose as authentic company owner by using some of the features of the authentic and target sites to trick customers into giving out their personal and confidential information
2. Pharming: attackers convert the domain name system (DNS) to numerical Internet Protocol (IP) address, so users will put in the correct website link of their choice but they get redirected to the phishing site without knowing
3. Vishing: this is where phishers call people in the pretence of family members or relatives and collect information or funds from them
4. Smishing (Gupta, et. al., 2018): phishers send SMS messages to people with fake link for them to put in their information

There was a 65% increase of phishing in 2016, with a total of 1,220,525 attacks for the year and half a billion dollars was reportedly lost to phishing in the United States every year (Azeez, et. al., 2020). Phishing attacks are still very rampant and do not show signs of ever stopping. Reports of phishing attacks have doubled each year since 2001 (Azeez and Ademolu, 2016). This goes to show that many people still fall victim to this cyber-attack. These attacks are done with precision on the part of the attackers.

Phishers study their victims to know the sites they visit regularly and ensure to contact these victims stating the need for them to change their passwords as their account could be blocked or disabled. The victims who want to preserve their accounts, will go ahead and change their password or login details, providing access for the attack. Due to this danger, a lot of individuals and companies have lost valuable information and a lot of money (Nureni and Irwin ,2010).

Because the victims do not notice the minute details that differentiate these sites from the legitimate ones, they fall prey to the attack. Through the adoption of whitelist, users will be notified when such changes occur, thereby saving them from impending danger and monumental loss.

Different techniques to combat phishing and prevent phishing have been implemented over the years. One of them is the Whitelist approach.

Whitelist, being the opposite of a blacklist is a list of sites that a user frequently visits that requires login details and are considered to be legitimate. This in turn blocks other sites that are not on the list from accessing the user's information.

This basically checks the sites that are safe and notifies the user if the site is not legitimate or if the site is not on the whitelist. Efforts were made to adopt Machine Learning (ML) approach: Naive Bayes Classifier, Support Vector Machines (SVM), Decision Tree and Random Forest for the implementation of this work.

The work aims at preventing phishing through the Whitelist approach. The objectives include:

1. To prevent phishing through the Whitelist approach
2. To identify illegal sites
3. To protect the interest and privacy of users while surfing the internet

To reduce the rapid increase in phishing attacks to a minimal level

2. Methodology

Machine learning algorithms were used in the implementation of this system. The steps taken to implement this are:

1. Data gathering: Data was gathered from PhishStorm (Azeez and Ademolu, 2016). 48,009 non-phishing sites were gathered, and 48,009 phishing sites were gathered from site. 10 features were extracted from the data.
2. Data Cleaning: Incorrect data entry was manually filtered out of the data gathered to allow the models to train using correct and authentic data
3. Model Training: Models were trained using some selected supervised machine learning algorithms (Support Vector Machines, Decision Tree, Naive Bayes Classifier and Random Forest).
4. Model Comparison: Trained models were compared based on their performances by using the following metrics: True Positive, True Negative, False Positive and False Negative.
5. Creation of Web Browser Extension: The best model based on its performance was then used to create a dataset which was used to detect phishing sites as an extension on the web browser

2.1 Data Gathering

This is the first step in implementation of the solution. It involves collecting several phishing and non-phishing sites. Data was gathered from PhishStorm. A total number of 48,009 non-phishing sites and 48,009 phishing sites were gathered.

2.2 Data Cleaning

The data gathered contained some inaccurate entries which were inconsequential to the research. Data cleaning was done by manually going through the data and filtering out the incorrect entries in order to help the models to better understand what a phishing and a non-phishing URL looks like.

2.3 Feature Extraction

This is where the data is converted to dataset of lesser number of variables based on the features selected containing the right amount of information to work with. Some features were selected to check the URLs and how well the models perform. A total of 8 features were selected to check the legitimacy of the URLs.

The Features are:

1. Length of URL
2. HTTPS token
3. Number of dots
4. Number of sub-domains
5. Digit count in the URL
6. Suspicious characters like @ and %40
7. Multiple occurrence of https, http

The features were divided into numerical and categorical features.

2.3.1 Numerical Features

These are the features that have continuous numeric data. They are data that signify a measurement or a count of values.

1. Length of URL: Most phishing sites are very lengthy because they are trying to cover the illegitimacy of their sites such that users will not be able to see it due to the length. Because URLs are broken down into three major parts with various sub-parts, this feature will be broken down to best classify the site

$$f_1 = length\ of\ the\ host\ name$$

$$f_2 = length\ of\ path$$

2. Number of dots: Phishing sites tend to have a lot of dots in their host name unlike legitimate sites with less than two dots. URLs that have many numbers of dots are most times categorised as phishing sites.

$$f_3 = number\ of\ dots$$

3. Number of sub-domains: Phishing sites are known to want to duplicate original sites and they tend to use the same name but add extra words to it, making the user think he is on a safe site. These extras are most times added between domain of a legitimate site and they are most times more than one.

$$f_4 = number\ of\ sub - domains$$

4. Digit count in the URL: The occurrence of digits in a legitimate URL is very rare and if it exists, the digits are always very few. Phishing sites tend to have a lot of digits in their URL.

$$f_5 = number\ of\ digits$$

2.3.2: Categorical Features

These are the features that have discrete numeric data. They are data that signify uncountable data and data that can be described using intervals.

1. HTTPS token: Websites are said to be secure when they have an https token but illegitimate and not secure sites do not have that but instead have http

$$f_6 = \begin{cases} 1, & HTTPS\ token \\ 0, & not\ HTTPS\ token \end{cases}$$

2. Suspicious characters like @ and %40: Legitimate sites do not have

the occurrence of '@', '_' and '% in their URLs. URLs that have any one of these suspicious characters can be categorized as phishing sites

$$f_7 = \begin{cases} 1, has\ no\ suspicious\ characters \\ 0.\ has\ suspicious\ characters \end{cases}$$

3. Multiple occurrence of https, http: Websites are required to have just one occurrence of https or http but when a URL has more than one of these tokens, it can be said to be a phishing site

$$f_8 = \begin{cases} 1, & one\ occurrence\ of\ https,http \\ 0, & multiple\ occurrences\ of\ https,http \end{cases}$$

2.4 Model Training and Algorithms Used

Decision was reached on the three algorithms because of their popularity along with observable contradictory results obtained on them from previous researches. What is more, they can also provide relatively good performance on the classification task in this work.

The data collected was separated into training and testing sets. Some part of the data was used to train the model using the features extracted based on the aforementioned supervised machine learning algorithms (Naive Bayes Classifier, Support Vector Machine, Decision Tree and Random Forest) and results were obtained. The testing data was then fed into the model to see how well it has trained.

2.4.1: Naive Bayes Classifier

Naive Bayes Classifier is a machine learning model or classifier that uses the Naive Bayes' theorem of probability. It is used to predict a class of unknown circumstances. The classifier assumes that the predictions on a class are not dependent on each other.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \text{...............} (1)$$

Where P (c | x) is the posterior probability of class given predictor
P (x | c) is the likelihood i.e. probability of attribute given class
P(c) is the prior probability of class
P(x) is the prior probability of predictor

This algorithm assumes that the features are independent of each other so it tests the data based on the features individually (Jain and Gupta, 2016). How it works is that:

1. It converts the data set into a frequency table
2. Creates a table of likelihood to derive the probabilities of each feature
3. The algorithm was implemented using Python

2.4.2: Support Vector Machine

Support Vector Machine (SVM), is a supervised learning model that is used to analyse data for classification and regression problems. It is a model that best splits data. It works as follows: each data item is plotted as a point in n-dimensional space (n is the number of features) and the values of each of the features is the value for a specific coordinate.

A margin of best fit is plotted to show how best the data can be split and this margin is referred to as a Hyperplane (Azeez and Babatope, 2016). The points closest to the hyperplane on opposite sides are referred to as the Support Vectors. The distance between the support vectors and the hyperplane should be as far as possible.

The algorithm uses support vectors and hyperplanes, where support vectors are the vectors closest to the plane and the hyperplane is the line of best fit that passes through the points or vectors (Nivedha et. al., 2017). The steps taken to use this are:

1. Identify the right hyperplane
2. Classify the two classes in the data
3. Implement it using Scikit-learn libraries

2.4.3: Decision Tree

A Decision Tree is a prediction model used in machine learning to solve problems of classification and regression. It is designed in the form of a tree-like graph and the data set is split using different features or conditions. It represents decisions and decision making. It represents the if-else statement (Nivedha et. al., 2017).

How this works is:

1. Start with a training data set that has attributes and classification
2. Ascertain the best attribute in the dataset
3. Split this set into subsets with values of this best attribute
4. Generate decision tree nodes based on the best attribute
5. Keep generating nodes using the subset from (3) till you cannot classify further

2.4.4: Random Forest

This model makes use of many decision trees, hence the word "Forest". It is used for classification and regression. To classify a new instance, each decision tree provides a classification for the input data. The classification from all the trees are taken and the prediction with the highest "vote" is selected (Chiew et. al., 2020).

How it works is:

1. When classifying a new object, different decision trees are used
2. Each decision tree classifies the input data
3. All the classifications made by the trees are taken and compared
4. Vote is taken for the classification
5. The classification with the highest vote is selected

2.5: Model Evaluation

The model results were accessed for each of the machine learning algorithms used. The models were accessed based on their performances (Wu et. al., 2018). The following metrics were used to evaluate the models:

1. **Confusion Matrix** (True Positive, False Positive, True Negative and False Negative): ***True positive*** is when the assumed class of a data is 1 (true) and the predicted result is 1 (true) (Al-Janabi et. al., 2017). ***False Positive*** is when the assumed class is 0 (false) and the predicted is 1 (true). True Negative is when both the assumed and the predicted result are 0 (false) and ***False Negative*** is when the assumed data class is 1 (true) and the predicted result is 0 (false)

2. **Accuracy**: This refers to the amount of correct predictions made by the model

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total}$$ … (2)

3. Precision: This refers to how concise and exact the predictions are in that, the sites we predicted as phishing sites are actually phishing sites, same for legitimate sites

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Negative}$$ … (3)

4. Recall or Sensitivity: This refers to the correctness of the models in diagnosing the sites as phishing or non-phishing or legitimate. The sites which are phishing should be predicted as phishing, same for non-phishing

$$Sensitivity = \frac{True\ Positive}{Actual\ True}$$ … (4)

5. Specificity: This refers to the correctness also in that, the sites that are legitimate were predicted to be legitimate by the models

$$Specificity = \frac{True\ Negative}{Actual\ False}$$……. (5)

2.6 Implementation of Web Browser Extension

Implementation was carried out using JavaScript, HTML, and CSS and it is categorized into five steps.

1. Create the project: This is where the file and the folder to house these files were created. A manifest file is created which tells the browser what it needs to know in order to open the extension. The HTML and CSS files are also created which contains the display of the extension. A separate file was created to hold any script file and it references the HTML file
2. Update the manifest file: Code was added to the manifest file which is in a JSON format
3. Create the UI: Writing of the code in the HTML page that allows you to click on the extension icon
4. Implement how the UI should work: Write the script such as event listeners
5. Test the Implementation: This is where the extension created was tested to know if it is working fine or needs any improvement

3.0: System Design

The application is in the form of a web browser extension where once there is a change in the URL, the Whitelist system scans the URL and compares it to the ones already on the whitelist.

If there are similarities between the new URL and one of the URLs on the list, the user will be notified that progress can be made. Whereas, if there is no similarity, the user is notified about the change and required to stop all transactions on that site.

4. Machine Learning Techniques

The model was evaluated using the four machine learning algorithms (Naïve Bayes, Support Vector Machine, Decision Tree, and Random

Forest). The result gotten from the comparison of the evaluation was used to determine the algorithm that will then be used to create the Web Extension.

4.1 Naïve Bayes

The confusion matrix for Naive Bayes was able to correctly classify 8663 URLs as authentic (True negatives), wrongly classified 3600 URLs as authentic (False negatives), wrongly classified 1015 URLs as phishing (False positives) and correctly classified just 5904 URLs as phishing (True positives).

Table 1 shows a total Precision of 0.71 and 0.85 for both non Phishing and Phishing when using Naïve Bayes. The corresponding graphical interpretation is shown in Figure 1.

Table 1 Model Evaluation for Naïve Bayes

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Phishing	0.71	0.90	0.79	9678
Phishing	0.85	0.68	0.72	9504
Total/Average	0.78	0.76	0.75	19182

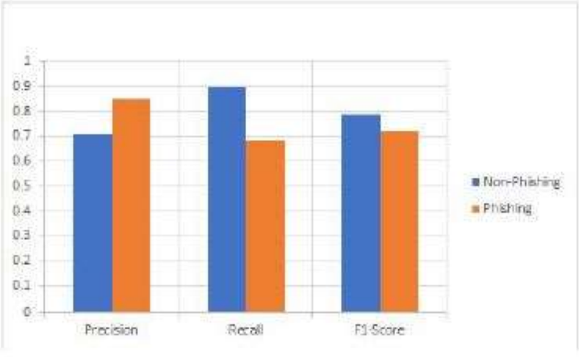

Figure 1. Graph of Model Evaluation for Naïve Bayes

4.2 Support Vector Machine (SVM)

The confusion matrix for SVM was able to correctly classify 8762 URLs as authentic (True negatives), wrongly classified 3663 URLs as authentic (False negatives), wrongly classified 916 URLs as phishing (False positives) and correctly classified just 5841 URLs as phishing (True positives)

Table 2 Model Evaluation for SVM

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Phishing	0.71	0.91	0.79	9678
Phishing	0.86	0.61	0.72	9504
Total/Average	0.78	0.76	0.76	19182

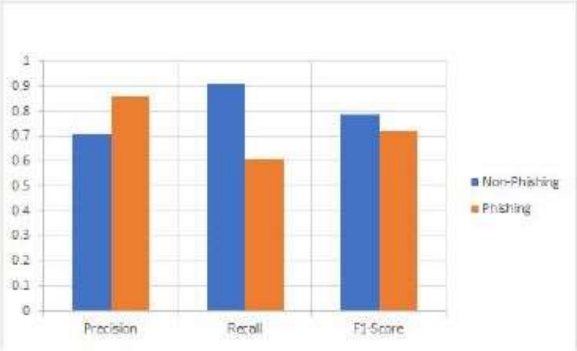

Figure 2. Graph of Model Evaluation for SVM

Table 2 shows 0.71 and 0.86 as values for Precision for both non-Phishing and Phishing with SVM. The graphical interpretation is shown in Figure 2.

4.3 Decision Tree

The confusion matrix for Decision Tree was able to correctly classify 8624 URLs as authentic (True negatives), wrongly classified 2178 URLs as authentic (False negatives), wrongly classified 1054 URLs as phishing (False positives) and correctly classified just 7326 URLs as phishing (True positives).

Table 3 Model Evaluation For Decision Tree

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Phishing	0.80	0.89	0.84	9678
Phishing	0.87	0.77	0.82	9504
Total/Average	0.84	0.83	0.83	19182

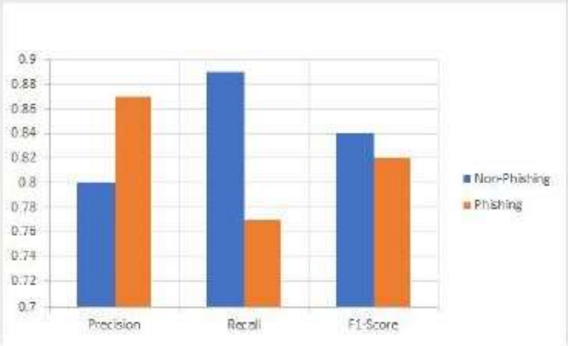

Figure 3. Graph of Model Evaluation for Decision Tree

Table 3 and Figure 3 provide the values obtained for both categories (Non Phishing and Phishing) when Decision Tree was considered.

4.4 Random Forest

The confusion matrix shows Random Forest was able to correctly classify 8545 URLs as authentic (True negatives), wrongly classified 1895 URLs as authentic (False negatives), wrongly classified 1133 URLs as phishing (False positives) and correctly classified just 7609 URLs as phishing (True positives).

Table 4 Model Evaluation for Random Forest

Category	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Phishing	0.82	0.88	0.85	9678
Phishing	0.87	0.80	0.83	9504
Total/Average	0.84	0.84	0.84	19182

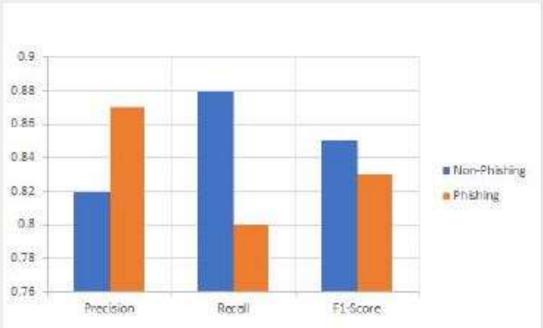

Figure 4. Graph of Model Evaluation for Random Forest

Table 4 and Figure 4 provide the values obtained for both categories (Non Phishing and Phishing) when Random Forest was considered.

Based on this comparison shown below, Random Forest has a higher model evaluation compared to the rest. It has the highest recall, that is, it is correctly classifying the non-phishing URLs as non-phishing, therefore, it is the best option to use to create the Web Extension.

Table 5. Comparison of the algorithms performances

Category	Precision	Recall	F1-Score
Naïve Bayes	0.78	0.76	0.75
SVM	0.78	0.76	0.76
Decision Tree	0.84	0.83	0.83
Random Forest	0.84	0.84	0.84

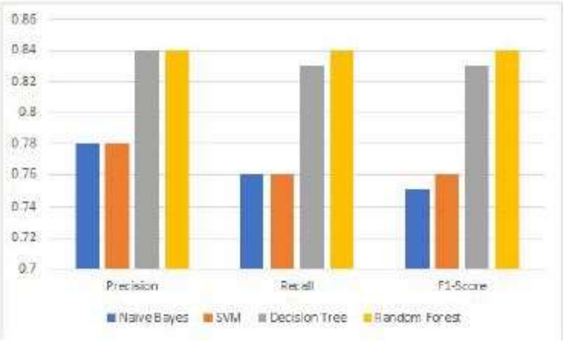

Figure 5. Graph of Comparison of the Algorithms Performances

Table 5 and Figure 5 provide the summary of the values obtained for both categories (Non Phishing and Phishing) when all the Machine Learning algorithms were evaluated.

5. Related Work

This part shows the review of articles of journals, documents from the internet on what phishing is about and the methods or approaches used to detect and prevent phishing. These methods were reviewed based on their benefits and their weaknesses in solving phishing.

In the work of Dudhe and Ramteke, they discussed the use of various approaches to detect phishing. The use of known and new features was applied in preventing phishing. They made use of Blacklist-Whitelist based approach, Fizzy rule-based approaches, Machine learning approaches, heuristic approach, CANTINA based approaches and Image based approaches to prevent and detect phishing for users (Dudhe and Ramteke, 2015). These approaches were used to determine which of them is the best among the anti-phishing techniques listed and the heuristic approach was said to be the best or at least better than the other approaches. The weakness of this is in its inability to work on server-side security.

A desktop application called PhishShield that takes a URL as input and brings the status of the URL (either phishing or legitimate website) as the output was implemented and discussed in the work of Rao and Ali. It has an accuracy rate of 96.57% and it can detect phishing sites that trick users by changing the contents to images (Rao and Ali, 2015). This implementation made use of the heuristic approach and was said to detect phishing attacks that blacklists cannot detect. It is considered to be faster than visual based assessment techniques that have been used in phishing detection (Strinzel, 2019). However, the result can still be improved upon in terms of its performance and the cost of computation using techniques like generic algorithms, neural network.

In 2015, Sedgewick et. al., developed Application Whitelisting which uses whitelists to determine the applications that are allowed to execute on a host, thereby preventing malware and other unapproved software. They wanted to educate organizations on the use and implementation of application whitelisting (Sedgewick et al., 2015). They discussed how highly recommended these solutions are when it comes to security. Organizations who want to make use of these solutions should be risk conscious when it comes to deploying the solutions. It requires diligence among staff to maintain and manage the solutions.

A very promising method to avoid phishing, Zero Knowledge Authentication (ZeKo), was developed by Shar et al., in 2015. The solution protects users from phishing attacks. The reasons phishing ((Matumba et. al., 2019). is still a rampant and growing attack is due to the ignorance of the users when it comes to computer and its usage. Users fail to see the slightest change in the URL; they fail to notice security warnings when they are on a website. They studied human behaviour in relation to phishing and realised that the attackers go for users that are gullible and extract their classified information directly from them. The attackers do this either via SMS, known as SMSishing and Voice conversation, known as Vishing. With this solution in place, phisher can easily be checked and prevented from carrying out his nefarious activities (Shar et al., 2015).

A content-based approach to detecting phishing using CANTINA as a good phishing site detector was implemented (Dudhe and Ramteke, 2015). The implementation made use of PHP and MYSQL, also making use of web crawlers. It basically crawls the original website URL, the location of the server and 'whois' information. When a user gets an email attached with a phishing link, the system takes the URL, that is, the link, and compares it with the original URL. It also does that for the location of the server and the 'whois' information. It analyses these for similarities, then conveys the result to the user. This implementation is said to be effective as it has a 6% false positive performance, then coupled with the heuristic approach, has a 1% false positive performance but it still needs to be improved on as because it is not user friendly (Gupta et al., 2015).

Fraud Website Detection application which discovers fraud websites through the use of RIPPER algorithm to categorize the websites was implemented by Prajapati et al., in 2016. This application takes corrective measures against fraudulent websites by reporting the prospective sites to the concerned authority. They went on to discuss different approaches used to detect fraud websites and how Heuristic approach is the better approach as it can detect fraud websites before they are blacklisted (Rao and Ali, 2015). The application still needs to be improved upon as it can be a plug-in to the browser, thereby, notifying the users when they are surfing the internet.

A novel approach for phishing protection that makes use of auto-updated whitelist of all authentic sites that a user access was implemented. A whitelist has a list of all the legitimate sites a user can visit while blacklist contains all the sites that a user should not visit as it is a phishing site. This approach has the likelihood of detecting attacks very well and very fast. It is sufficient for a real-time environment and it can be improved upon by using other features to detect phishing and legitimate sites even if these new features will increase running time complexity of the system (Gupta and Jain, 2016).

Rao and Ali made use of an enhanced heuristic approach to combat phishing where blacklist and whitelist were made use of. Websites that are not legitimate and are not already on the blacklist are discovered and the blacklist is updated, same for the whitelist where it is updated on the legitimate sites that are not already on it (Rao and Ali, 2015). The solution was implemented using PHP programming and Database and has a high accuracy level (Okunoye et al., 2016). It is said to be highly effective and user-friendly but it still needs to be further worked on as it does not use visual similarities approach which makes it time consuming.

6. Conclusion

Having fully known the danger of phishing in the global community, it is an understatement to say that it has caused financial damages in most financial institutions. The essence of carrying out this research is, therefore, in the right direction. The machine approach adopted has clearly revealed how the adopted approach can be fully utilized in identifying phishing URLs and curtailing phishers. The summary of the results obtained as shown in Table 5 revealed that Random Forest performed has the best performance with the metrics considered. Phishing URLs can easily be detected if users are conscious of the change in the URLs and also when web extensions can notify the user if the URL is a phishing or non-phishing one. In order to achieve maximum accuracy, we propose that neural networks should be used for future research instead of traditional ML approach adopted in this case. Consequently, the proposed application can identify phishing URLs with an accuracy of 84.57%.

References

Adebowale MA, Lwin KT, Sánchez E, Hossain MA (2019) Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text. Expert Systems with Applications 115 (2019) 300–313.

Azeez NA, Salaudeen BB, Misra S, Damaševičius R, Maskeliūnas R et al (2020). Identifying phishing attacks in communication networks using URL consistency features. Int. J. Electronic Security and Digital Forensics, Vol. 12, No. 2, pp 200-213

Gupta BB, Arachchilage NAG, Psannis KE (2018) Defending against Phishing Attacks: Taxonomy of Methods, Current Issues and Future Directions. Telecommunication Systems volume 67, pp 247–267, https://doi.org/10.1007/s11235-017-0334-z

Azeez NA, Ademolu O (2016) CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification. 2016 International Conference Computational Science and Computational Intelligence (CSCI) Las Vegas, NV, USA: IEEE, 959-965.

Jain AK, Gupta BB (2016) A novel approach to protect against phishing attacks at client side using auto-updated whitelist. EURASIP Journal on Information Security. doi:10.1186/s13635-016-0034-3

Azeez NA, Babatope AB (2016) AANtID: an alternative approach to network intrusion detection. The Journal of Computer Science and its Applications. An International Journal of the Nigeria Computer Society, 129-143.

Nivedha S, Gokulan S, Karthik C, Gopinath R et al (2017). Improving Phishing URL Detection Using Fuzzy Association Mining. International Journal of Engineering and Science (IJES), 21-31.

Chiew KL, Tan CL, Wong K, Yong KSC. , Tiong WK (2020) A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences 484 (2019) 153–166

Wu C, Shi J, Yang Y, Li W (2018) Enhancing Machine Learning Based Malware Detection Model by Reinforcement Learning. ICCNS 2018, November 2–4, 2018, Qingdao, China.

Al-Janabi M, Quincey E, Andras P (2017) Using supervised machine learning algorithms to detect suspicious URLs in online social networks. 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Nureni AA, Irwin B (2010). Cyber security: Challenges and the way forward. Computer Science & Telecommunications, 29, 56-69.

Nivedha S, Gokulan S, Karthik C, Gopinath R et al (2017). Improving Phishing URL Detection Using Fuzzy Association Mining. International Journal of Engineering and

Science (IJES), 21-31.

Dudhe P.D, R. P. (2015). A Review on Phishing Detection Approaches. International Journal of Computer Science and Mobile Computing, 166-170.

Rao R.S, A. S. (2015). PhishShield: A Desktop Application to Detect Phishing Webpages through Heuristic Approach. Eleventh International Multi-Conference on Information Processing-2015, 147-156.

Sedgewick A, S. M. (2015). *Guide to Application Whitelisting.* National Institute of Standards and Technology Special Publication 800-167.

Shar K, S. T. (2015). Phishing: An Evolving Threat. *International Journal of Students Research in Technology & Management*, 216-222.

Gupta B.B, A. A. (2015). Defending against Phishing Attacks: Taxonomy of Methods, Current Issue and Future Directions.

Prajapati U, S. N. (2016). Fraud Website Detection using Data Mining. *International Journal of Computer Applications*, 0975-8887.

Gupta B.B., J. A. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. *EURASIP Journal on Information Security*.

Okunoye B, A. N. (2016). PHISHDETECT: A Web Enabled Anti-Phishing Technique using Enhanced Heuristic Approach. *Department of Computer Sciences, University of Lagos, Nigeria*.

Matumba L, M. F. (2019). Blacklisting or Whitelisting? Deterring Faculty in Developing Countries from Publishing in Substandard Journals. *University of Toronto Press*, 83-95.

Strinzel M, S. M. (2019). Blacklists and Whitelists To Tackle Predatory Publishing: a Cross-Sectional Comparison and Thematic Analysis. *Swiss National Science Foundation, Bern, Switzerland*.