URL based Email Phishing Detection Application

Roshan Ravi[1], Abhishek Arvind Shillare[2], Prathamesh Prakash Bhoir[3], K.S. Charumathi[4]

[1,2,3]B.E. Student, Department of Information Technology, Pillai College of Engineering, New Panvel, Navi Mumbai, Maharashtra – 410206, India

[4]Assistant Professor, Department of Computer Engineering, Pillai College of Engineering, New Panvel, Navi Mumbai, Maharashtra – 410206, India

---***---

Abstract – *Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information. The information is then used to access important accounts and can result in identity theft and financial loss. Phishing link dispersing is associated with a number of methods, this research focuses on email messages as a means to send phishing website links, which spoof or mimic, banks, credit card and companies or other ecommerce. The purpose of this research is to design and develop a phishing detection application to detect the phishing websites which are attached to the users' emails by using the Random Forest Algorithm. This application will report the user by classifying the website as Phishing or Safe. The requested URL to be checked will be blacklisted if classified as phishing.*

Key Words: Email Phishing, URL (Uniform Resource Locator), Phishing Detection, Random Forest Algorithm.

1. INTRODUCTION

Financial services such as banking are now easily available over the Internet making the lives of people easy. Thus, it is very important that the security and safety of such services are maintained. One of the biggest threats to web security is Phishing. Phishing is the technique of extracting user credentials by masquerading as a genuine website or service over the web. There are various types of phishing attacks such as Spear phishing, which targets specific individuals or companies, Clone phishing is a type of phishing where an original mail with an attachment or link is copied into a new mail with a different (possibly malicious) attachment or link, Whaling, etc. Phishing can lead to huge financial losses. For example, the Microsoft Consumer Safer Index (MCSI) report for 2014 has estimated the annual worldwide impact of Phishing and other identity thefts to be nearly USD 5 Billion [1]. Similarly, the IRS has warned of a surge in phishing attacks with over 400% increase in reported cases [2].

Several solutions have been proposed to combat phishing ranging from educating the web users to stronger phishing detection techniques. The conventional approach to phishing detection has not been successful because of the diverse and evolving nature of phishing attacks. For instance, in December 2020, the total number of unique phishing reports submitted to the Anti-Phishing Working Group (APWG) was 199,120. The number of unique phishing email subjects and the number of brands targeted by phishing campaigns were 133,038 and 515 respectively, according to the APWG report for the month of December 2020 [3]. Such huge numbers were reported despite taking preventive measures to thwart phishing. Upon investigation, it was found that each phishing attack was different from the other one. Thus, it becomes imperative to find a way to adapt our phishing detection techniques as and when new attack patterns are uncovered.

Machine learning algorithms, which make a system learn new patterns from data, are an ideal solution to the problem of phishing detection. Although there have been many papers in recent years which have attempted to detect phishing attacks using machine learning, we intend to go one first step further and build a web application which can be used by the end user systems to classify URLs sent through emails as phishing or safe.

For our project, we have created a dataset of features that represent the attributes of the URL associated with phishing pages. This dataset is then trained by a machine learning algorithm called the Random Forest Algorithm. We have created a Web Application where the user will enter the URL to be checked in an email. The requested URL can then be determined as phishing or legitimate with the help of the rules generated by the trained dataset. The result determining the nature of the URL will be reported to the user in the web application. This will prevent the user from becoming a victim of phishing done through emails.

2. LITERATURE SURVEY

Rabab Alayham Abbas Helmi et al [4] have designed and developed a tool to detect the source code of a phishing website which is attached to email by using a decision tree algorithm. In order to improve the protection of users' information from the fake website. Anti-phishing detection is suggested to overcome the problems through the following features. The first feature of an anti-phishing detection login system is by using the user's email and password. Second feature is detecting phishing websites which are attached to a user's email by using a decision tree algorithm. Lastly, a phishing website will be detected and generate a report to the user. The system testing shows high reliability and the users' feedback shows ease of use as well as learning. This anti-phishing application is able to link with the user's original emails and come out with the analysis of phishing emails that contain in the user's email. This project is using Agile Unified Process (AUP) methodology. Also, this application is able to calculate the percentages of stored phishing emails in the user's email.

Ms. Lisa Machado and Prof. Jayant Gadge [5] have proposed an efficient way for detection of the phishing websites using the C4.5 decision tree approach and features of URL (Uniform Resource Locator). The method proposed in this paper uses various URL features and also uses C4.5 decision tree approach for better results. The proposed approach uses features extracted from URL to decide the authenticity of input URL as the phishers cannot use the exact replica of the URL. The C4.5 classifier is used to generate rules. The rules generated are used to classify the submitted URL as phishing or legitimate with better efficiency. Using the proposed approach, the average accuracy achieved by the system is very high. The system is robust and precise in detecting phishing sites. In future, more URL-based features can be used to make the system more accurate.

Thuy Lam and Houssain Kettani [6] have proposed the use of a phishing detector application, PhAttApp. This application offers numerous features to detect and prevent ransomware delivery through phishing channels and thus reduces the risk of ransomware infection. Ransomware attacks often start with a delivery phase in which attackers deliver malicious content. Attackers often use multiple threat vectors for ransomware enablement such as emails, instant messages, and drive-by downloads, exploiting the vulnerabilities of a network or application. Among these attack vectors, email is the top threat vector, which most ransomware attackers attempt to use. According to this research paper, PhAttApp, powered by machine learning, is geared to identify emerging phishing threats that humans may miss. It applies psychological-analysis techniques to automatically detect malicious emails and constantly adapts its understanding as phishing threats mutate. It also responds autonomously to contain threats at the earliest sign of compromise. PhAttApp is part of a larger application that will assist users in defending against ransomware attacks.

Eint Sandi Aung et al [7] have emphasized on URL-based phishing detection techniques, because it considers the URL to be a significant criterium in preventing phishing attacks. Moreover, examining URL-based features can also encourage faster processing than other approaches. In this work, the aim is to understand the structure of URL-based features and surveying their diverse detection techniques and mechanisms. The performance is analyzed based on the combinations of URL features on different datasets. And then summarization of the findings is done to promote better URL-based phishing detection systems. In this research paper, common data sources were listed, and comparative evaluation results and matrices were shown for better survey. It concluded with the recommendations for more effective phishing detection in the future.

Ms. Sophiya Shikalgar et al [8] have proposed a method that predicts the URL based phishing attacks based on features and also gives maximum accuracy. This method uses uniform resource locator (URL) features. The authors here identified features that phishing site URLs contain. The proposed method employs those features for phishing detection. The proposed system predicts the URL based phishing attacks with maximum accuracy. Various machine learning algorithms are used which can help in decision making and to get better accuracy of prediction. Different machine learning algorithms such as XG Boost, SVM, Naive Bayes and Stacking, where stacking uses XG Boost and SVM as its base classifier and Random Forest as its meta classifier are used in the proposed system to detect URL based phishing attacks. The hybrid algorithm approach by combining the algorithms will increase accuracy. The authors of this research paper found that their system provides an 85.5 % of accuracy for XGBoost Classifier, 86.3% accuracy for SVM Classifier, 80.2 % accuracy for Naïve Bayes Classifier and finally 85.6 percentage of accuracy when using Stacking Classifier. Hence it was found that the best among all the above classifiers is SVM and Stacking Classifier which shows maximum accuracy.

R. Kiruthiga and D. Akila Sung [9] have surveyed the features used for detection and detection techniques using machine learning. It consists of an outline of algorithms used to detect Phishing websites URLs. The authors of this paper after referring to various other papers came to a conclusion that most of the detection of phishing websites work is done by using familiar machine learning algorithms like Naïve Bayesian, SVM, Decision Tree and Random Forest, where Random Forest has highest accuracy of 98.4%.

3. FEATURES EXTRACTION FROM URL

The features of URL play a vital role in detecting fake sites because phishers cannot copy the exact URLs of legitimate sites. In order to detect phishing sites, various features of the URL are extracted from the submitted URL. Following are the features that are extracted from the URL:

- Using the IP Address: If an IP address is used as an alternative of the domain name in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their personal information.
- URL's having "@" Symbol: Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.
- Adding prefix or suffix separated by (-) to the domain: The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate website. For example, http://www.Confirme-paypal.com/.

- Alexa Ranking: By reviewing our dataset, we found that in worst scenarios, legitimate websites are ranked among the top 100,000. Furthermore, if the domain has no or less traffic, or is not recognized by the Alexa database, the website is assigned as "Phishing".
- Sub Domain and Multiple Sub Domains: Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD), which in our example is "uk". The "ac" part is shorthand for "academic", the combined "ac.uk" is called a second-level domain (SLD) and "hud" is the actual name of the domain. To produce a rule for extracting this feature, we firstly have to omit the (www.) from the URL which is in fact a sub domain in itself. Then, we have to remove the (ccTLD) if it exists. Finally, we count the remaining dots. If the number of dots is greater than three, it is classified as "Phishing" since it will have multiple sub domains. Otherwise, if the URL has no sub domains, we will assign "Legitimate" to the feature.
- Using URL Shortening Services: URL shortening is a method on the "World Wide Web" in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an "HTTP Redirect" on a domain name that is short, which links to the webpage that has a long URL. For example, the URL "http://portal.hud.ac.uk/" can be shortened to "bit.ly/19DXSk4".
- Redirecting using "//": The existence of "//" within the URL path means that the user will be redirected to another website. An example of such URL's is: "http://www.legitimate.com//http://www.phishing.com". We examine the location where the "//" appears. We find that if the URL starts with "HTTP", that means the "//" should appear in the sixth position. However, if the URL employs "HTTPS" then the "//" should appear in seventh position.
- Using Suspicious Top-Level Domain: There are certain top-level domains that are frequently used by attackers for phishing and are considered as suspicious and if the domain of the requested URL matches to have suspicious top-level domain, then the website is assigned as "Phishing".

4. ARCHITECTURE OF PROPOSED SYSTEM

According to the proposed system architecture as shown in Fig. 2, we first started by creating a dataset which consists of legitimate and phishing URLs along with their feature values as 0 or 1 representing the legitimate and phishing nature of the URL respectively. Then the dataset undergoes pre-processing using SMOTE (Synthetic Minority Oversampling Technique) to remove the imbalance in the dataset. Later, this dataset is trained using Random Forest Algorithm so that the rules generated by it can be used to classify the requested URL as "Phishing" or "Legitimate".

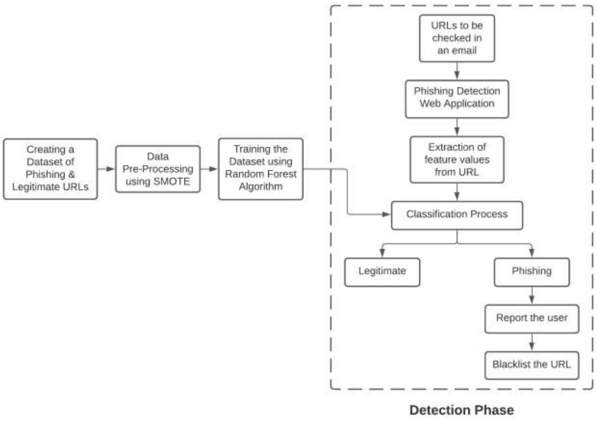

URLs to be checked in an email
Phishing Detection Web Application
Extraction of feature values from URL
Creating a Dataset of Phishing & Legitimate URLs
Data Pre-Processing using SMOTE
Training the Dataset using Random Forest Algorithm
Classification Process
Legitimate
Phishing
Report the user
Blacklist the URL
Detection Phase

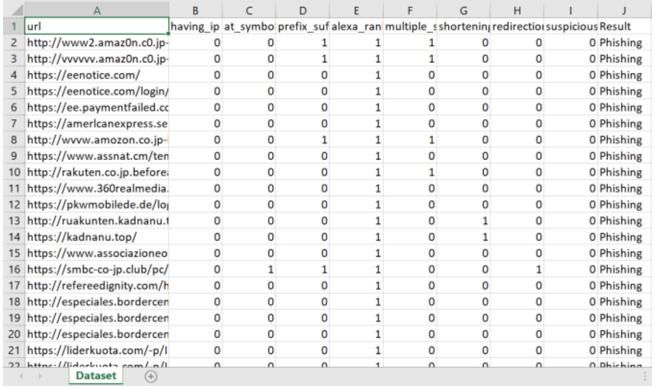
Fig. 1: Architecture of Proposed System

4.1 Creating a Dataset of Phishing and Legitimate URLs

A Dataset is created which consists of both Legitimate and Phishing URLs. Phishing and Legitimate URLs are taken from PhishTank and Kaggle respectively. It consists of 7500 URLs (instances) among which 5000 are Phishing instances and 2500 are Legitimate instances. Certain features of these URLs are used to classify between Legitimate and Phishing URLs. In our dataset, these features are given values such as 0 and 1 representing Legitimate and Phishing nature of the URLs respectively. The values of these features are given as follows based on different conditions:

- Using the IP Address: If the URL consists of an IP address, then the value is assigned as 1, else value is 0.
- URL's having "@" Symbol: If the URL consists of the "@" symbol, then the value is assigned as 1, else value is 0.
- Adding prefix or suffix separated by (-) to the domain: If the URL consists of the (-) symbol, then the value is assigned as 1, else value is 0.
- Alexa Ranking: If the domain rank is greater than 100,000 in the Alexa database, then the value is assigned as 1, else value is 0.
- Sub Domain and Multiple Sub Domains: If the domain consists of more than 3 dots, then the value is assigned as 1, else value is 0.
- Using URL Shortening Services: If such services are used, then the value is assigned as 1, else 0.
- Redirecting using "//": If the last occurrence of "//" symbol is greater than 7th position, then the value is assigned as 1, else value is 0.

- Using Suspicious Top-Level Domain: If the domain consists of Suspicious Top-Level Domain, then the value is assigned as 1, else value is 0.



	A	B	C	D	E	F	G	H	I	J
1	url	having_ip	at_symbo	prefix_suf	alexa_ran	multiple_s	shortenin	redirectio	suspicious	Result
2	http://www2.amaz0n.c0.jp-	0	0	1	1	1	0	0	0	Phishing
3	http://vvvvvv.amaz0n.c0.jp-	0	0	1	1	1	0	0	0	Phishing
4	https://eenotice.com/	0	0	0	1	0	0	0	0	Phishing
5	https://eenotice.com/login/	0	0	0	1	0	0	0	0	Phishing
6	https://ee.paymentfailed.cc	0	0	0	1	0	0	0	0	Phishing
7	https://amerlcanexpress.se	0	0	0	1	0	0	0	0	Phishing
8	http://wvvw.amozon.co.jp-	0	0	1	1	1	0	0	0	Phishing
9	https://www.assnat.cm/ten	0	0	0	1	0	0	0	0	Phishing
10	http://rakuten.co.jp.before	0	0	0	1	1	0	0	0	Phishing
11	https://www.360realmedia.	0	0	0	1	0	0	0	0	Phishing
12	https://pkwmobilede.de/lo	0	0	0	1	0	0	0	0	Phishing
13	http://ruakunten.kadnanu.t	0	0	0	1	0	1	0	0	Phishing
14	https://kadnanu.top/	0	0	0	1	0	1	0	0	Phishing
15	https://www.associazioneo	0	0	0	1	0	0	0	0	Phishing
16	https://smbc-co-jp.club/pc/	0	1	1	1	0	0	1	0	Phishing
17	http://refereedignity.com/h	0	0	0	1	0	0	0	0	Phishing
18	http://especiales.bordercen	0	0	0	1	0	0	0	0	Phishing
19	http://especiales.bordercen	0	0	0	1	0	0	0	0	Phishing
20	http://especiales.bordercen	0	0	0	1	0	0	0	0	Phishing
21	https://liderkuota.com/-p/l	0	0	0	1	0	0	0	0	Phishing

Fig. 2: Dataset

4.2 Data Pre-Processing

The Dataset does not consist of any missing values, but it is imbalanced since there are 5000 phishing instances and 2500 legitimate instances. To balance these instances, we use SMOTE (Synthetic Minority Oversampling Technique).

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them.

SMOTE synthesizes new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data. Following are the steps involved in SMOTE algorithm:

- Step 1: Setting the minority class set A, for each x∈A, the k-nearest neighbors of x are obtained by calculating the Euclidean distance between x and every other sample in set A.
- Step 2: The sampling rate N is set according to the imbalanced proportion. For each x∈A, N examples (i.e x1, x2, ...xn) are randomly selected from its k-nearest neighbors, and they construct the set A_1 .
- Step 3: For each example $x_k \in A_1$ (k=1, 2, 3...N), the following formula is used to generate a new example: x' = x + rand (0,1) * |x - x_k| in which rand (0, 1) represents the random number between 0 and 1.

4.3 Training the Dataset using Random Forest Algorithm

After the Data Pre-Processing step, the Dataset is trained by using a Machine Learning Algorithm called the Random Forest Algorithm.

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, a random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

We can understand the working of Random Forest algorithm with the help of following steps −

- Step 1 − First, start with the selection of random samples from a given dataset.
- Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- Step 3 − In this step, voting will be performed for every predicted result.
- Step 4 − At last, select the most voted prediction result as the final prediction result.

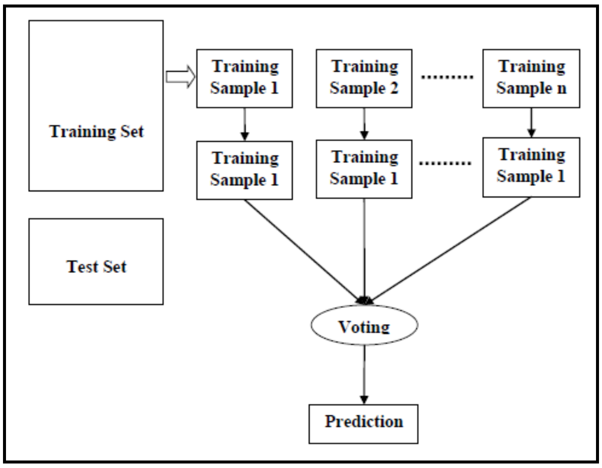


Fig. 3: Working of Random Forest Algorithm

4.4 Detection Phase

In this phase, the requested URLs or the websites are detected as Phishing or Legitimate. Following are the steps involved in this phase:

- Step 1: The URL to be checked in an email is provided to the Phishing Detection web application.
- Step 2: The URL is then transmitted to the feature extractor Python program, which extracts the

feature values through the predefined URL-based features.

- Step 3: The obtained feature values are given as an input to the Random Forest classifier.
- Step 4: With the help of trained Dataset and the feature values of the requested URL, Random Forest classifier can determine whether the requested URL is fake or legitimate. If the site is classified as phishing by the classifier, then the user is made alert about the classification result.

5. IMPLEMENTATION

Fig. 4 below shows the Phishing Detection Web Application.

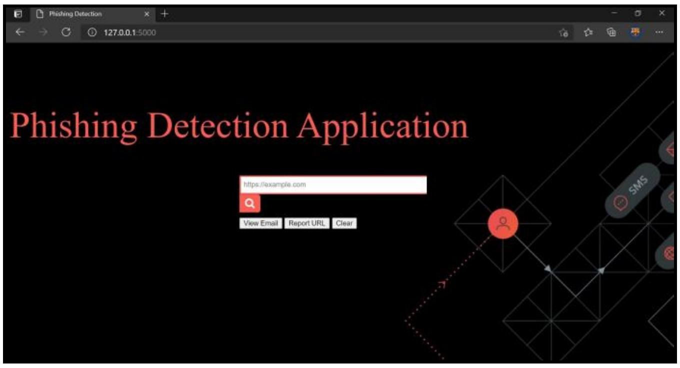
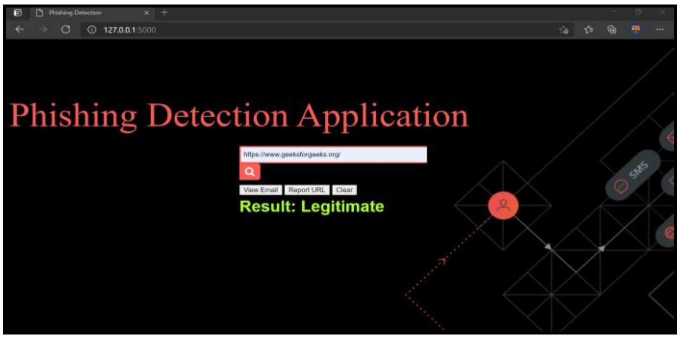


Fig. 4: Phishing Detection Application

This application consists of a text area where the user can enter the URL to be checked in an email, a "View Email" button which redirects the user to their email account and a "Report URL" button where the user can report the phishing URL to Google Safe Browsing which is a blocklist service provided by Google that provides lists of URLs for web resources that contain malware or phishing content. It has a "Clear" button to clear the Result generated as "Phishing" or "Legitimate" after checking the requested URL.

Fig. 5 shows the Result as "Legitimate" when the URL to be checked is safe. The following URL is used for testing this - https://www.geeksforgeeks.org/

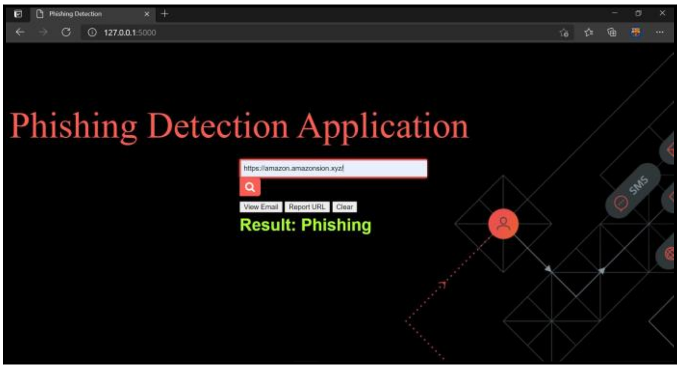


https://www.geeksforgeeks.org/

Result: Legitimate

Fig. 5: Phishing Detection Application showing result as "Legitimate"

Fig. 6 shows the Result as "Phishing" when the URL to be checked is not safe or considered as a phishing website. The following URL used for testing this, is taken from PhishTank - https://amazon.amazonsion.xyz/



https://amazon.amazonsion.xyz/

Result: Phishing

Fig. 6: Phishing Detection Application showing result as "Phishing"

Fig. 6 shows the Result as "Phishing" when the URL to be checked is not safe or considered as a phishing website.

6. CONCLUSION

Phishing presents substantial risk to business and is often cited as a top security concern. Targets of phishing are generally the weakest link in the security chain – People. Using Natural Language Processing (NLP) to understand the structure and occurrence of words in a phishing email that tricks the user to click the phishing link, or by analyzing and understanding the code behind the phishing webpage, are some of the techniques used to prevent phishing done through emails. But both these techniques are not effective enough, the first being unreliable and the other being inefficient and requiring technical knowledge. So, the most efficient and reliable method to detect phishing websites is by using the features of the URL, as used in our Phishing Detection web application. This method gives faster results and through our application users can easily check the link that they receive in their email by entering it in the application. This method and application will help the user to prevent themselves from becoming a victim of phishing done through emails.

ACKNOWLEDGEMENT

We would like to extend our deepest gratitude to our Project guide Prof. K.S. Charumathi for exemplary guidance, monitoring and constant encouragement throughout this project which helped us improve our work. We would also like to extend our gratitude to our Head of Information Technology Department Dr. Satishkumar L. Varma for providing us with an opportunity and platform to carry out this project. We would also like to express our gratitude to our Principal Dr. Sandeep M. Joshi who provided us with the golden opportunity as well as all the facilities needed to carry out our project.

REFERENCES

[1] Microsoft, Microsoft Consumer safety report. Available at https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumer-safety-index-reveals-impact-of-poor-online-safety-behaviours-in-singapore/sm.001xdu50tlxsej410r11kqvksu4nz

[2] Internal Revenue Service, IRS E-mail Schemes. Available at https://www.irs.gov/uac/newsroom/consumers-warned-of-new-surge-in-irs-email-schemes-during-2016-tax-season-tax-industry-also-targeted

[3] APWG Phishing Activity Trends Report. Available at https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf

[4] Rabab Alayham Abbas Helmi, Chua Shang Ren, Arshad Jamal and Muhammad Irsyad Abdullah. Email Anti-Phishing Detection Application. Paper presented at the 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 7 October 2019, Shah Alam, Malaysia.

[5] Ms. Lisa Machado and Prof. Jayant Gadge. Phishing Sites Detection based on C4.5 Decision Tree Algorithm. Paper presented at the 2017 Third International Conference on Computing, Communication, Control and Automation (ICCUBEA).

[6] Thuy Lam and Houssain Kettani. PhAttApp: A Phishing Attack Detection Application. ICISDM 2019, April 6-8, 2019, Houston, TX, USA© 2019 Association for Computing Machinery.

[7] Eint Sandi Aung, Chaw Thet Zan and Hayato Yamana. A Survey of URL-based Phishing Detection. Department of Computer Science and Communication Engineering, Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, 159-8555, Japan. DEIM Forum 2019 G2-3.

[8] Ms. Sophiya Shikalgar, Dr. S.D. Sawarkar and Mrs. Swati Narwane. Detection of URL based Phishing Attacks using Machine Learning. Paper presented at International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 8 Issue 11, November-2019.

[9] R. Kiruthiga, D. Akila Sung. Phishing Websites Detection Using Machine Learning. Paper presented at the International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S11, September 2019.