Grenze International Journal of Engineering and Technology, June Issue

A Survey on Secure Message—A Proposition using Machine Learning for Handling Fraudulent Messages

C Kiran Mai[1], P Manas Kumar[2], B Swena Shanthi[3], P Roshini Reddy[4], T Dhanush Reddy[5] and P Geethanjali[6]
[1-6]Dept. of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
Email: ckiranmai@vnrvjiet.in, manaskumarpalavalasa04@gmail.com, swenashanthi10@gmail.com, patlollaroshini@gmail.com, dhanushthappetareddy@gmail.com, palakittigeethanjali@gmail.com

***Abstract*— The spread of digital communication exposes oneself to running across malicious messages that overpower security features. This paper introduces Secure-Message, a new-fangled app aimed at checking the security of messages that may arrive from any app. The Machine Learning (ML) algorithms such as Decision Tree Algorithm, Support Vector Machine or Random Forest are used to analyse the received messages and provide the user a quick, animated output to let them know if it is safe enough to open the message or not. Secure-Message functions by using advanced ML algorithms trained on a varying dataset of messages that are labelled either safe or malicious. The app will evaluate several features that include text patterns, embedded links, and Metadata in assessing the probability of the message being malicious. The ease of use is built such that this application provides a transparent interface for integration with different messaging systems. In being so easy to assess online basically, it is hoped that Secure-Message alongside the rapid communications will improve security and decrease people fall victim to phishing, malware, and other forms of cyber-attacks. This Machine Learning approach enables prompt response, thus reducing the risk through fraudulent cyber messaging and providing the needful direction. This application is easily adaptable and scalable thus providing an effective solution for the present-day cyber security challenges.**

***Index Terms*— Malware, Phishing, Machine learning, Support vector machine, Decision tree algorithm, Random-forest, Fire based Cloud Message (FCM).**

I. INTRODUCTION

In the modern digital world, the internet assumes an important role in day-to-day living; while at the same time, users are subjected to several online threats such as phishing schemes, viruses, and several harmful websites. These events led to the increase in cybersecurity attacks; thus, it has become necessary to ensure a secure environment for browsing. The existing solutions of cybersecurity are often automated, in which the user has little or no control, and are highly resource consuming. Phishing is a deceptive practice entailing social engineering and technological tricks aimed at gaining sensitive customer identification and financial information. This is often made possible through manipulations on social media platforms, where fake e-mails, seemingly from well-known companies and organizations, lure users into fraudulent sites, where they might unknowingly supply their financial details such as usernames and passwords.

Grenze ID: 01.GIJET.11.2.510_25

The latest research concluded with great success in identifying the phishing attacks on the internet. It was the work of Dhamdhere and Vanjale[1] who focused on classification of the phishing emails in real time, using the k-means algorithm. 160 emails were collected from computer engineering students and sorted them into three generic classes based on technical parameters, structure, and content.

The positive rates of legitimate emails were calculated as 67%, while phishing emails had 80%; negative rates were 30% and 20%, respectively. Bahnseny et al. [2] applied two algorithms, Random Forest and Long Short-Term Memory (LSTM), to analyze the datasets PhishTank and Common Crawl and achieved an accuracy of 93.5% and 98.7%, respectively.

Alejandro Correa Bahnsen, Eduardo Contreras, and their team in their study compared two approaches—Random Forest (RF) and Long Short-Term Memory (LSTM)—for detecting phishing URLs. LSTM was found to be more accurate at 98.7% compared to RF at 93.5%, but it consumed more time to train. On the other hand, RF consumed more memory and processed URLs faster, whereas LSTM was more memory-efficient and did not require manual feature engineering.

Their results indicate that LSTM is suited for applications based on mobile devices with low memory, whereas RF yields the faster processing time and easier interpretability. The most prevalent type of phishing is attempting to get through mobile devices, this is the main attack among the several modes used by criminals. Typically, the aim here is to get to know the personal data of the victims. In the State of Phish, 2018 report Wombat Security informs those one out of every three targeted businesses turned into fake URL link tricks on the mobiles [3]. Through fraudulent URLs, phishers plan to gather private and very sensitive data from victims like financial details, personal information, usernames, passwords, etc. [4]

If users unknowingly navigate to fake websites, being assured of the authenticity, they might unwittingly give away their sensitive information without a hint of doubt or suspicion. This occurs because the fake web page is an accurate imitation of the real one [5]. According to the user's experiences with phishing attacks, which was a part of a conducted study, it was discovered that computer users are the subjects of phishing because of the poor understanding of URLs, the lack of knowledge about the identification of trustworthy websites and the impulsive/accidental clicks, the difficulty of verifying the real nature of a link, and the problems related to the identification of phishing scams.

The next successful attack will definitely happen; it is not a question of "if" but "when". The above problem shows that there is a compelling need to come up with new security strategies aimed at preventing the physical attacks from happening, recognizing, or informing users. A major problem associated with Industrial Control Systems (ICS) is that despite the implementation of detection techniques, testing the system may cause disruption on the actual physical processes, and thus could also interfere with the normal working of the system. According to Monti et al. [6], another reason that contributes to the small number of ICS datasets is the fact that most organizations are reluctant to disclose information about their architecture.

This reticence is primarily motivated by fears that the exposure of details about the architecture of organizations may give attackers information to construct vulnerabilities or, in other words, reveal sensitive business information.

II. Literature Survey

Phishing attacks are more common nowadays than ever before [7]. The cybercriminals bet heavily on various sophisticated social engineering techniques to deceive victims into downloading malware or submitting their credentials onto set-up phishing pages. Most plague enters developed thus far have considerably catered toward desktop computer users, with mobile users often taking second place. Mobile users are far more prone to attacks than their desktop counterparts, as mobile users are always online and have naturally limited capabilities in terms of screen-size and processing power.

Some detection and prevention techniques have been proposed against phishing; however, it still stands as a significant threat [8]. Various approaches include blacklisting, URL-based detection, static detection, and heuristic techniques.

Mobile applications tend to be exploitable if developers do not apply a high level of vigilance [9]. In addition, Google Play does not always implement strict security when launching new applications. Thus far, one incident implicated the popular Android game "Angry Birds," hacked by an infiltrator who inserted malicious code into its APK file to send out unauthorized text messages at a cost of 15 GBP for the affected users. Unfortunately, it affected over 1000.

An application must ask the user for the required permissions before it can access any of the system resources. Should certain permissions be disallowed, the application may not function correctly [10]. System updates or

upgrades tend to advance privacy and security. Android 11 is now the latest stable version and already brings forth several improvements toward privacy and security. Examples include scoped storage mechanisms, one-time permissions, automatic permission resets, issues regarding background location access, unveil package visibility, and foreground service measures.

The increasing volume of malware attacks, primarily on Android devices, calls for well-coordinated detection systems [11]. The unique feature of ML and DL approaches towards malware detection-create classifiers of MCA directly from training data, making independent signature definitions obsolete. This review details the progressive innovations around ML-based paradigms for Android malware detection, covering the techniques of APKs analysis and source code vulnerabilities.

The authors of this paper intend to develop a model that successfully combines real-time identification and prevention of phishing attacks on edge devices, with a feature that guarantees that user data is not sent to an external server. This makes the worker address privacy issues associated with traditional machine learning models for phishing detection. This paper aspires to develop a realized model so as to provide real-time identification and prevention of potential phishing attacks [12]. "Traditional Machine Learning Models for Phishing detection require Significant computational resources to operate. Therefore, user data is being sent to a server for phishing attack detection," thus, turning it into" a concern from a privacy perspective." To overcome this problem, "the authors have proposed a quantized model which can be deployed in low computational devices and can be used for real-time phishing attack detection at the client side without sending user data to a third party."

The 2020 Internet Crime Report revealed about 791,790 complaints regarding various types of cybercrime [13]. Among these, phishing scams encompassed around 30%, thus being the most popular cybercrime and costing about 54 million USD in losses. Therefore, it becomes indispensable that users access the Internet equipped with the capability to distinguish between original and fraudulent sites, and visual tools are hence required to assist the users in recognizing phishing sites.

When people visit a fraudulent page that looks very much like the legitimate page, many never recall the domain name of the original website, particularly in the cases of start-ups or lesser known companies. Because of this, a user might have some difficulty identifying a phishing page simply by viewing the URL. Certain browsers, including Chrome, feature a security mechanism aimed at detecting both phishing and malware thanks to warning notifications when a user tries to access an unsecure page. Google Safe Browsing was introduced in 2007 [14] and has now been incorporated into many Google services, among them Gmail and Google Search. This particular feature relies on a blacklist that is a listing of URLs that have been associated with malware or phishing activities.

Sonawal and Kuppuswamy [15] introduced “PhiDMA”, a 5-layer phasing detection model which targets both normal and visually impaired people (blind people). The model consists of multiple process of checking the legitimate websites. Starting from verification with whitelist and next URL feature layer to identify some common attributes, third comes the lexical signature-based construction of its content, the most highlighted method by screening the URL with search engine and finding string matching and estimating the phishing likelihood based on “Phishing Index” linked to new problems. The Authors developed Model where it Provides the user’s control over Voice alerts.

Authors Mao and Bian [16] developed a learning based aggregation analysis schema for detecting Phishing Websites. In specific authors considered the similarity between the web pages structures using the CSS layout. This method automatically trains classifiers to recognise phishing websites without human intervention. The results of Random Forest among the four classifiers was the best with more than 93% accuracy. This is a simple method that is language neutral and operates on a single feature category.

Awasthi and Goel [17] use two data sets, one from the UCI Machine Learning Repository that has 2456 instances and 31 attributes and the other Data set being used was from Kaggle website that is of 11055 instances and same 31 attributes. Both Datasets feature 30 URL-related more of attributes, which indicates whether the website is phishing or not. After standardising the data, the study assesses the impact of all 30 features on classifier performance. The best accuracy of 99.18% was achieved using the Extra Trees and XGBoost classifiers.

Alejandro Correa Bahnsen, Eduardo Contreras, and their team in their study compared two approaches[19]—Random Forest (RF) and Long Short-Term Memory (LSTM)—for detecting phishing URLs. LSTM was found to be more accurate at 98.7% compared to RF at 93.5%, but it consumed more time to train. On the other hand, RF consumed more memory and processed URLs faster, whereas LSTM was more memory-efficient and did not require manual feature engineering. Their results indicate that LSTM is suited for applications based on mobile devices with low memory, whereas RF yields the faster processing time and easier interpretability.

III. Proposed Methodology

The monitoring system deployed in real-time is activated when a user clicks on a web site or directly communicates a URL via the app. The processing link immediately gets initiated with no further instructions from the user. The program automatically verifies the URL. This seamless cooperation guarantees that the user does not need to wait for any manual actions or prompts. The real-time monitoring informs the user instantly of either the safety or potential risks of the connection. This immediate analysis showcasing live data on phishing threats improves user experience.

A. Database Validation

The database includes URLs that the model has already processed and is quite large. The application checks the shared link against the existing URL database, a measure that significantly enhances both speed and accuracy. This step is crucial as it facilitates the predictive stage by providing data history. If the link is not in the database, the app automatically proceeds to the second stage, feature extraction.

B. Extraction of Features

For a deeper analysis this application extracts vital info such as - the IP address, the URL, and the content of the site at this stage. The characteristics of the URL comprise the URL's length, the number of subdomains, and the presence of unique characters or ambiguous terms. The IP features analyse the domain's age, and geolocation to find out if genuine. Content features the fact that the HTML code of the page is being vigilant when it comes to spotting hidden fields, mysterious scripts, or anything that has to do with phishing like the word "password" and "login." The extracted attributes serve as a basis for classification in the Random Forest model in the next step.

C. Analysis and Classification with Random Forest Model

Data Collection & Preprocessing

The data is taken from two openly accessible datasets: the Phishing Sites Dataset from Kaggle and the UC Irvine data set. The UC Irvine dataset contains 54 elements with genuine, absolute, and whole number information types, and 235,795 occurrences. The Kaggle dataset incorporates 30 highlights and is likewise marked with phishing and authentic URLs, giving important information to preparing and assessment of the phishing recognition model.

The diverse nature of these datasets will enhance the model’s ability to classify links accurately and reliably. Standardization of URLs is implemented by removing the inquiry boundaries, switching to all lowercase characters, and normalizing the space explicit components. Tokenization separates the URL into subdomains and parts for phishing design detection. The missing data is handled either by adding or removing the incomplete passages. Clear-cut features are encoded with the help of techniques such as one-hot or label encoding. To conclude, highlight scaling is the normalization of features such as URL length or sidetracks, so that they are treated equally during model training.

Feature Extraction and Training

The extraction of highlights from the URL [18] takes crude information and makes it organized for AI. The IP-based highlights, such as the geolocation from the dangerous district. Content-based highlights, for example, malicious HTML elements, keywords like "login," malicious JavaScript, and excessively too many redirects, are some of the details that are most targeted to obtain phishing sites.

These highlight details extracted give a detailed aspect of the URL which then is used for accurate phishing detection. The Random Forest model is trained to use feature matrix consisting of length of a URL and subdomains that are given as the input to the model. The model is prepared by Decision trees that split information based on these features, with the majority of the trees casting their votes for the URL being a phishing or a safe one. The cross-validation method is mainly used for model execution evaluation thus fetching better accuracy. The features selection highlights the URL length, or key phrases as the main parameters.

Deployment and Real-Time Prediction

The model gets equipped on the valuable statistical nuances of features from training and tuning. It will be deployed in the cloud, with the ability to classify the malicious URLs instantly. when a user shares or clicks a link, the relevant features are quickly extracted from the URL. These features are then sent to the cloud where the model tags the URL as a phishing attempt, and a notification reaches the app user within seconds. And note this is where the bulk of work is done: with a model that can classify URLs almost instantaneously, yet at the same time is powerful enough to be able to process high volumes of URLs on the fly.

Push Notification for Phishing Alerts

To alert the user of a detected phishing link, FCM immediately pushes the warning, to ensure there is no delay. A user receives this notification as “Warning: This link may be phishing!” with a body, such as, “Suspicious keywords or redirects detected.” This way, users get an idea of what caused the link to be flagged and respond to it in time. The notification is delivered to the user's device in real time, even if the app is closed. The system exploits Firebase Cloud Messaging’s (FCM) low latency capabilities for real-time alerts, so it can give timely protection.

Real-Time Integration in Android Apps

The phishing detection procedure is running without any gaps in the background and checking links as they are shared or opened, Hence, intervention from user activity is not needed for real-time protection. The low-power methods are under consideration, through which the app effectively reduces battery use without shortening the time of attack detection. A real-time push notification is delivered when a phishing link is detected through FCM, letting the user know immediately. This invisible operation supports the user experience making it possible to get protection and be updated in real-time without user intervention.

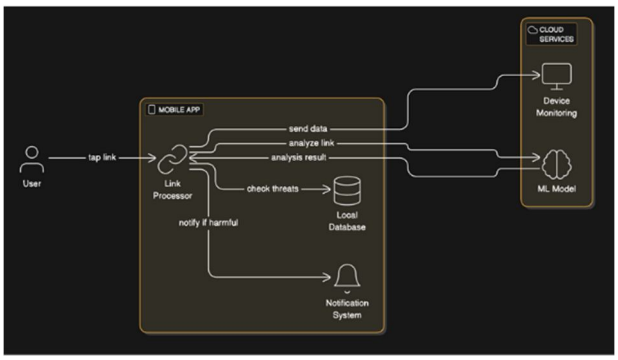

Fig 1: System Architecture

This system is able to respond immediately if a user taps on a link within the mobile application. Because the program app Link Processor takes control of the link, it will first look in a local database for known threats. In this way, it will not have to depend on an external server and can respond as fast as possible. The link is passed to a cloud-based machine learning model to be further looked at if it is not found in the local database. This model evaluates the legitimacy of the link and sends back the results to the app. In case the link is legitimate, the user will be allowed to continue without interference. In case it is malicious, the app immediately alerts the user using a notification system and avoids possible threats. In addition to this, it also includes cloud component device monitoring, which will help in boosting security on the broader scale as well. With this multi-level approach, localized efficiency can potentially be combined with the advanced level of real-time analysis of cloud-based ML modeling for strong protection against malicious links.

IV. Conclusion

We are striving for the development of a mobile app that really is a step forward in the world of cybersecurity as this app will enable the prevention and even real-time identification of virus-like Internet threats. Thus, the creation of these algorithms, which will also manage unique and dynamic threats, will occur by the means of artificial intelligence tools. We will be adding immediate alerts and easy-to-understand graphical reporting so people can decide wisely based on their own garnered knowledge of cyberspace and this doesn't require them any in-depth technical skills. We are putting a great deal of emphasis on usability and operating in real-time as these are the main tools that a user needs to get around more safely and be better equipped in a technologically driven world. The developing process of this project is a step in the right direction towards making the digital ecosystem more secure and trustworthy by protecting users from criminal scams and malware attacks.

References

[1] V. Mhaske-Dhamdhere and S. Vanjale, “A novel approach for phishing emails real time classification using k-means algorithm,” International Journal of Electrical and Computer Engineering (IJECE), vol.8, no.6, pp. 5326-5332, 2018, https://doi.org/ 10.11591/ijece.v8i6.pp.5326-5332

[2] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, “Classifying phishing URLs using recurrent neural networks,” eCrime Researchers Summit, pp. 1–8, 2017, https://doi.org/10.1109/ECRIME.2017.7945048.
[3] Crane, C. “20 Phishing Statistics to Keep You from Getting Hooked in 2019,” https://www.thesslstore.com/blog/20-phishing-statistics-to-keep-you-from-gettinghooked-in-2019/ (Last accessed on 26 May 2021)
[4] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, “Defending against phishing attacks: taxonomy of methods, current issues and future directions,” Telecommunication Systems, vol. 67, pp. 247–267, 2018. https://doi.org/10.1007/s11235-017-0334-z
[5] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, “User experiences of torpedo: Tooltip-powered phishing email detection,” Computers & Security, vol. 71, pp. 100–113, 2017. https://doi.org/10.1016/j.cose.2017.02.004
[6] Mauro Conti, Denis Donadel, and Federico Turrin. “A Survey on Industrial Control System Testbeds and Datasets for Security Research”. In: CoRR abs/2102.05631 (2021). arXiv: 2102.05631. https://arxiv.org/abs/2102.05631.
[7] J. D. Ndibwile, Y. Kadobayashi and D. Fall, "UnPhishMe: Phishing Attack Detection by Deceptive Login Simulation through an Android Mobile App," 2017 12th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Korea (South), 2017, pp. 38-47, https://doi.org/10.1109/AsiaJCIS.2017.19.
[8] A. K. J. Diksha Goel, "Mobile phishing attacks and defencemechanisms: State of art and open research: State of art and open research," computers & s e c u r i t y , vol. 73, pp. 519-544, 2018. https://doi.org/10.1016/j.cose.2017.12.006
[9] Tam, K.; Feizollah, A.; Anuar, N.B.; Salleh, R.; Cavallaro, L. The evolution of android malware and android analysis techniques. ACM Comput. Surv. (CSUR) 2017, 49, 1–41. https://doi.org/10.1145/3017427
[10] Privacy in Android 11 | Android Developers. https://developer.android.com/about/versions/11/privacy (accessed on 19 May 2021)
[11] Senanayake, J.; Kalutarage, H.; Al-Kadri, M.O. Android Mobile Malware Detection Using Machine Learning: A Systematic Review. Electronics 2021, 10, 1606. https://doi.org/10.3390/electronics10131606
[12] Ushus Maria Joseph, Mendus Jacob; Real time detection of Phishing attacks in edge devices using LSTM networks. AIP Conf. Proc. 30 August 2022; 2520 (1): 020001. https://doi.org/10.1063/5.0103355
[13] 2020 Internet Crime Report. Federal Bureau of Investigation. Available online: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf (accessed on 21 March 2021).
[14] Google Safe Browsing. Google.com. 2014. Available online: https://safebrowsing.google.com/ (accessed on 18 July 2021).
[15] Gunikhan Sonowal, K.S. Kuppusamy, PhiDMA – A phishing detection model with multifilter approach, Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 1, 2020, Pages 99-112, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2017.07.005
[16] Mao, J., Bian, J., Tian, W. et al. Phishing page detection via learning classifiers from page layout feature. J Wireless Com Network 2019, 43 (2019). https://doi.org/10.1186/s13638019-1361-0
[17] Awasthi, A., Goel, N. Phishing website prediction using base and ensemble classifier techniques with cross-validation. Cybersecurity 5, 22 (2022). https://doi.org/10.1186/s42400-022-00126-9
[18] Mahmoud Khonji, Youssef Iraqi, Andrew Jones, “Phishing Detection: A Literature Survey”, in IEEE COMMUNICATIONS SURVEYS & TUTORIALS, Vol. 15, No. 4, Fourth Quarter 2013, pp. 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009
[19] Mohammad, Rami, McCluskey, T.L. and Thabtah, Fadi Abdeljaber (2014) Intelligent Rule based Phishing Websites Classification. IET Information Security, 8 (3). pp. 153160. ISSN 17518709. https://doi.org/10.1049/iet-ifs.2013.0202
[20] Santhana Lakshim, Vijaya MS, “Efficient Prediction of Phishing Websites using Supervised Learning Algorithms”, in International Conference On Communication Technology and System Design2011, https://doi.org/10.1016/j.proeng.2012.01.930.