Malicious URL Classification using ML Models*

Lior Vinman (ID: 213081763)[1], Nevo Gadassi (ID: 325545887)[2], Yoad Tamar (ID: 213451818)[3]

I. ABSTRACT

Our research and project focus on the development of machine learning (ML) models for the detection of phishing or malicious URLs. We introduce a novel approach leveraging ensemble learning techniques, particularly emphasizing the integration of the 'soft' voting mechanism. This innovation enhances the decision-making process within the ensemble, allowing for more robust and accurate identification of malicious URLs compared to previous methods. Through extensive experimentation and comparative analysis, our approach demonstrates superior performance and efficacy in combating online threats. Our submitted proof-of-concept (POC) task achieved remarkable results, attaining an accuracy rate of 99.8178% on our dataset, further validating the effectiveness of our methodology.

II. INTRODUCTION

In the modern cybersecurity landscape, the exponential growth in networking usage has led to a significant increase in web activity, with countless individuals engaging in online browsing and information retrieval on a daily basis. As reliance on the World Wide Web (internet) intensifies for various tasks and activities, securing this expansive digital domain becomes paramount. One of the primary challenges in cybersecurity is mitigating the risks associated with malicious activities targeting internet users by using a malicious URL as the main delivery method. Who among us hasn't received a message from our best friend that says "Hi buddy! Watch this funny video! here, on: www.video.com"? or an angry email from our boss about "Hurry up and upload your documents on this link: www.docs.com"? Many adversaries are aware of this social engineering vulnerability and may fake their identities to send their phishing URLs directly to you! Let's discuss how the most common cybersecurity-attack template breaks this down.

The MITRE ATT&CK framework provides a comprehensive template for dissecting cyberattacks, outlining various stages and tactics employed by adversaries. Of particular concern is the "initial access" step, where attackers seek to infiltrate systems to establish a foothold for subsequent malicious activities. Phishing emerges as a prevalent technique for achieving this initial access, exploiting human vulnerabilities through deceptive tactics to gain unauthorized entry into networks or devices. It is very simple - create a web server on a public machine that will hold a 1:1 website copy of the national post services with a field to enter payment information, but of course, you cannot use the real domain. What to do? The solution - "forget" the last letter in the original domain. What's the worst that will happen already? The next step is to send a malicious SMS that will say "Dear customer, your package is stuck at the customs, please pay the national tax to free it" and leave the phishing URL that we have created. If the target is an elderly person, he probably will click on the URL without a doubt and enter his credit card details, the challenge "10 days to become a Millioner" - just completed! Since this story is a real event that happens at least once a day (every day!), the motivation of using (and of course, mitigating) malicious URLs is understandable.

Once attackers gain a foothold through initial access, thwarting their subsequent actions becomes increasingly challenging. This underscores the critical need for robust defenses at the initial access stage to prevent or mitigate potential breaches and subsequent damage. Connecting these two subjects, the proliferation of internet usage amplifies the risk of falling victim to initial access techniques such as phishing, especially when talking about clicking on a URL. Therefore, our research aims to address this pressing issue by developing a classification system capable of discerning suspicious URLs, thereby enhancing internet security and safeguarding users against potential cyber threats, and of-course, collect real world thread intelligence data.

III. RELATED WORK

A. "SeizeMaliciousURL: A novel learning approach to detect malicious URLs"

-In the realm of cybersecurity, the article titled "SeizeMaliciousURL: A novel learning approach to detect malicious URLs" by Dipankar Kumar Mondal, Bikash Chandra Singh, and their colleagues stands out as a significant contribution. The authors present a pioneering machine learning-based method named SeizeMaliciousURL, designed to confront the escalating threat posed by malicious URLs on the internet.

Traditional approaches, such as blacklisting and heuristic methods, have proven insufficient in effectively identifying new and emerging malicious URLs. Recognizing this limitation, the authors delve into the challenges faced by the research community in combating the ever-evolving landscape of online threats. Despite numerous attempts to

*Final project in "Methods for Detecting Cybersecurity Attacks" course at Ariel University.

[1]L. Vinman is a B.Sc student in the School of Computer Science, majoring in Cybersecurity, at Ariel University, Israel. liorvi35@gmail.com

[2]N. Gadassi is a B.Sc student in the School of Computer Science, at Ariel University, Israel. nevogad4@gmail.com

[3]Y. Tamar is a B.Sc student in the School of Computer Science, majoring in Cybersecurity, at Ariel University, Israel. yoad.tamar123@gmail.com

leverage machine learning techniques for improved detection, the persistent issues of false positives and negatives have hindered the development of a flawless solution.

SeizeMaliciousURL addresses these challenges by introducing an innovative threshold-based condition. This condition plays a pivotal role in refining the decision-making process, filtering out ambiguous results and thereby mitigating prediction errors. By employing ensemble learning with multiple classifiers, the model predicts class probabilities for URLs and combines them based on corresponding probabilities to determine the final class label for unlabeled URLs.

The authors emphasize the crucial role of the threshold in enhancing the model's overall performance. Through rigorous evaluation on two distinct datasets, the article demonstrates that SeizeMaliciousURL outperforms other existing methods in detecting malicious URLs. The comparative analysis includes precision, recall, and F1 scores, with particular emphasis on the latter, showcasing the model's prowess in achieving a balanced performance in both malicious and non-malicious URL detection.

Further enriching the study, the authors conduct experiments to determine the optimal threshold value, shedding light on the sensitivity of the model's performance to this critical parameter. The results affirm that SeizeMaliciousURL excels in accuracy and F1 scores when compared to traditional machine learning models.

Looking ahead, the authors chart a course for future endeavors in refining the SeizeMaliciousURL project. They express their commitment to continuous improvement, particularly in the domain of feature engineering, aiming to enhance the model's accuracy and classification capabilities. This commitment to ongoing development positions SeizeMaliciousURL as a promising and adaptive solution in the dynamic landscape of cybersecurity, addressing the pressing need for robust and efficient detection mechanisms in the face of evolving online threats.

In the context of our own research and development efforts, the insights and methodologies presented in the article "SeizeMaliciousURL: A novel learning approach to detect malicious URLs" by Dipankar Kumar Mondal, Bikash Chandra Singh, and their collaborators have proven invaluable. Leveraging the innovative techniques proposed in SeizeMaliciousURL, our project has incorporated ensemble learning with multiple classifiers into our model architecture. The adoption of this approach aligns with our commitment to staying at the forefront of advancements in cybersecurity and machine learning. By integrating the threshold-based condition and ensemble learning strategies detailed in the article, we aim to enhance the accuracy and efficiency of our own malicious URL detection model. This collaborative exchange of knowledge underscores the interconnected nature of research in the field, fostering a collective effort to fortify defenses against evolving cyber threats. As we implement and refine these techniques within our project, we anticipate contributing to the ongoing evolution of effective cybersecurity solutions.

B. "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions"

-The article titled "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions" provides a comprehensive review of studies conducted between 2012 and 2021 on the detection of malicious URLs using machine learning (ML) algorithms. The studies are categorized based on the language of the website content (Arabic or English), the type of ML detection techniques used (supervised, unsupervised, semi-supervised), the classification types (binary or multi-classification), and the datasets used (public sources, collected by authors, commonly used).

The paper presents several taxonomies and discusses various techniques and properties related to malicious URL attacks and their detection methods. It highlights the importance of URL features, such as lexical, content-based, and network-based features, in building effective ML models for malicious URL detection. The paper also discusses common attack techniques, including spam, phishing, malware, and defacement URLs, and how malicious URLs can be exploited for these attacks.

The reviewed studies utilized different datasets, with PhishTank and Alexa being the most common sources. The ML algorithms used in the studies are categorized into supervised, unsupervised, and semi-supervised learning, with supervised learning being the most prevalent. The paper also discusses the challenges that might impact the quality of ML detection techniques, such as dataset size, outliers, feature selection, and the sustainability of detectors.

The paper concludes by summarizing the findings, highlighting the recent advancements in the field, and suggesting possible research directions for future studies on malicious URL detection using ML algorithms.

C. "Classification of Malicious URLs Using Machine Learning"

-The article titled "Classification of Malicious URLs Using Machine Learning" from the journal Sensors (2023, 23, 7760) by Shayan Abad, Hassan Gholamy, and Mohammad Aslani, focuses on developing machine learning models to efficiently identify and classify malicious URLs to enhance cybersecurity.

The study leverages four machine learning algorithms: Support Vector Machines (SVMs), Random Forests (RFs), Decision Trees (DTs), and K-Nearest Neighbors (KNNs), in combination with Bayesian optimization for accurate URL classification. To improve computational efficiency, instance selection methods such as Data Reduction based on Locality-Sensitive Hashing (DRLSH), Border Point Extraction based on Locality-Sensitive Hashing (BPLSH), and random selection are employed.

The results indicate that RFs are effective in delivering high precision, recall, and F1 scores, with SVMs also providing competitive performance but requiring increased training time. The study also highlights the substantial impact of the instance selection method on the performance of these

models, indicating its significance in the machine learning pipeline for malicious URL classification.

The study's methodology includes data collection and preparation, model development using the selected machine learning algorithms, and performance evaluation. Instance selection methods are used to generate smaller datasets that expedite the training process. The study also employs the Minimum Redundancy Maximum Relevance (MRMR) algorithm to select the most critical features for classification.

The article concludes by emphasizing the importance of tailored model selection and the need for innovative approaches to combat cyber threats. It suggests future improvements such as expanding the range of models, incorporating additional categories of malicious URLs, and integrating larger and more diverse datasets. User

In our extensive analysis, we further corroborated the findings presented in the article "Classification of Malicious URLs Using Machine Learning" by Shayan Abad, Hassan Gholamy, and Mohammad Aslani. In our study, we employed the same machine learning algorithms, namely Support Vector Machines (SVMs), Decision Trees (DTs), K-Nearest Neighbors (KNNs), and Random Forests (RFs), coupled with Bayesian optimization to enhance the accuracy of malicious URL classification. Our investigation focused on evaluating the impact of various instance selection methods on model performance, including Data Reduction based on Locality-Sensitive Hashing (DRLSH), Border Point Extraction based on Locality-Sensitive Hashing (BPLSH), and random selection.

Interestingly, our results align with the conclusions drawn in the aforementioned article, emphasizing the efficacy of Random Forests in achieving high precision, recall, and F1 scores for malicious URL classification. Notably, our study echoes the observation that while Support Vector Machines exhibit competitive performance, they entail a longer training time compared to Random Forests.

Furthermore, in line with the methodology outlined in the article, we adopted a meticulous approach to data collection and preparation. Our model development process mirrored the selected algorithms, and we leveraged instance selection methods to streamline the training process by generating smaller, more focused datasets. Additionally, we implemented the Minimum Redundancy Maximum Relevance (MRMR) algorithm for feature selection, ensuring that only the most critical features were utilized in the classification task.

In conclusion, our study reinforces the significance of tailored model selection and echoes the call for innovative approaches to bolster cybersecurity. Building upon the insights provided by the article, our research underscores the superiority of Random Forests in the context of malicious URL classification. Moreover, we expanded on the features highlighted in the article and introduced additional refinements to optimize model performance. Looking ahead, we echo the article's suggestions for future research, emphasizing the exploration of diverse models, inclusion of additional categories of malicious URLs, and integration of larger and more varied datasets to further enhance the robustness of the classification system.

IV. EVOLUTION

The solution architecture outlined in the project encompasses a systematic approach to analyze and predict the nature of URLs, distinguishing between malicious and benign ones based on their inherent characteristics. The architecture entails several distinct stages, each contributing to the overall process from initial data preparation to final model evaluation and storage.

A. Data Exploration and Feature Extraction

The process initiates by importing essential ML libraries. The dataset, comprising URLs and their corresponding labels denoting their nature (malicious := 1 or white := 0), is loaded from a '.csv' file. Subsequently, data cleaning operations are performed to eliminate redundant columns, likely stemming from indexing artifacts. Feature extraction is then conducted to derive pertinent characteristics from the URLs, crucial for subsequent machine learning model comprehension. These features encompass various aspects such as URL length, hostname length, path length, characteristics of the top-level domain (TLD), presence of specific characters or patterns within the URL, and indicators like the use of IP addresses or shortening services.

B. Exploratory Data Analysis

The architecture incorporates exploratory data analysis to gain insights into the dataset's composition and distribution. Visualization techniques are employed to depict the distribution of data, aiding in understanding the relative proportions of benign versus malicious URLs. Furthermore, an analysis of URL lengths is conducted through histograms, stratified by classification, to discern potential trends associating URL length with malicious intent.

C. Model Training and Evaluation

Following data preparation, the dataset is partitioned into training and testing subsets. A Random Forest classifier, chosen for its efficacy in ensemble learning and classification tasks, is trained on the prepared dataset. The Random Forest model constructs numerous decision trees during training and outputs the mode of classes (classification) determined by individual trees. Subsequently, the model's performance is evaluated using the test set, with accuracy serving as the primary evaluation metric. Additionally, a confusion matrix is generated to provide a comprehensive overview of the model's performance, detailing metrics such as true positives, false positives, true negatives, and false negatives. Finally, the trained model is serialized and saved to a '.joblib'.

D. AI Metrics

The solution architecture utilizes ensemble learning techniques, specifically the Random Forest algorithm, along with soft voting, to enhance the model's predictive capabilities. In addition to the previously mentioned AI metrics.

Ensemble learning, exemplified by the Random Forest algorithm, combines multiple models to improve predictive performance. Random Forest constructs a multitude of decision trees during training, each operating independently and contributing to the final classification decision. By aggregating the outputs of multiple decision trees, Random Forest mitigates overfitting and increases robustness, resulting in more reliable predictions.

Soft voting is a method employed in ensemble learning, wherein predictions from individual base models are weighted based on their confidence scores and combined to form the final prediction. In the context of this project, soft voting leverages the outputs of multiple decision trees in the Random Forest model, considering the probabilities assigned to each class for a given input. By aggregating these probabilities across all decision trees, soft voting yields a consensual prediction with enhanced accuracy and stability.

Soft voting contributes to reducing false positives through its weighted aggregation of predictions. It considers the probabilities associated with each class prediction, allowing more confident predictions to have a greater influence on the final decision. By combining the predictions of multiple decision trees in the Random Forest, soft voting helps balance out individual errors and biases, reducing the likelihood of false positives. This approach enhances the model's ability to make accurate classifications while minimizing the occurrence of false positives, thereby improving its overall effectiveness in detecting malicious URLs.

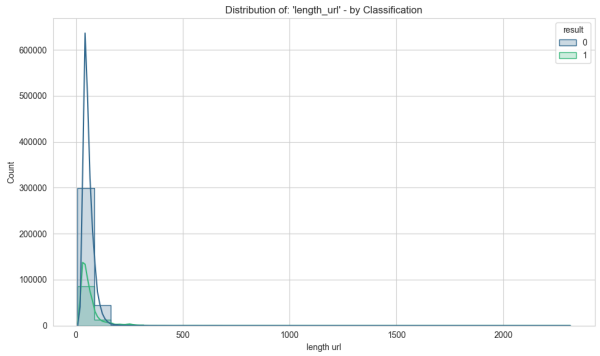
E. Used Dataset

As previously discussed, we utilized a '.csv' file for the dataset, comprising 450,175 lines of URL data with corresponding labels: 1 for malicious URLs and 0 for benign (white) URLs. Within the dataset, 23.2% of the entries are labeled as malicious, while the remaining 76.8% are categorized as white URLs. The distribution of URL lengths in the dataset is skewed towards shorter URLs, which is understandable given that URLs are primarily intended for human readability and memorization - as long URL may indicate a problem. Additionally, there are direct access URLs denoted by IP addresses (e.g., "http://192.168.1.105/"), and also URLs that are using a shorten service (such as, "http://bit.ly/").

V. DATA EXPLORATION

In our implementation, we used a dataset from the internet. Our dataset contains 450,175 lines of classified (i.e., labeled) URLs. There are many types of different URLs that we considered in training and therefore, we extracted many features to help us in the classification process. This dataset comprises URLs categorized into two classes: malicious URLs, constituting 23.3%, and white URLs, accounting for 76.8%. These URLs span a variety of types, including those with different protocols, directories, IP addresses, URL shorteners, and queries.

We created a function, named: 'extract_features(url)'. It is instrumental in extracting key features from URLs within the dataset, facilitating further analysis and classification.



length url graph

The extracted features offer insights into the structural and compositional characteristics of URLs, aiding in the identification of potential patterns indicative of malicious intent. For instance, the length features: encompassing 'length_hostname', 'length_path', 'length_fd', and 'length_tld', provide measures of the URL components such as the hostname, path, first directory, and top-level domain (TLD), respectively. Longer URLs might exhibit distinct patterns associated with certain types of attacks or phishing attempts, thereby influencing the classification process. The distribution of URL lengths reveals that there are both short and long URLs present, but there is a significantly higher proportion of shorter URLs.

Another type of extracted features is a character counts, including 'count_dash', 'count_at', 'count_question', 'count_percent', 'count_dot', and 'count_equals', shed light on the frequency of specific characters within the URLs. Such counts can unveil anomalous patterns, such as a higher occurrence of characters like '@' or '=', which might signify attempts at exploiting vulnerabilities or injecting malicious code (e.g., XSS inside the URL iteslf).

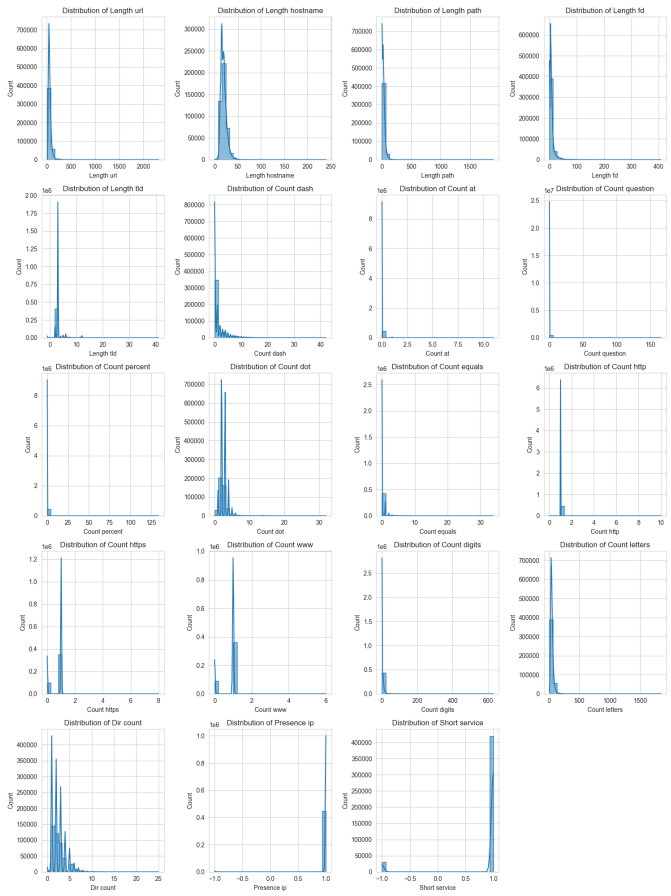
Furthermore, the presence of specific protocols and subdomains, as captured by 'count_http', 'count_https', and 'count_www', contributes to understanding URL characteristics. The usage of secure protocols (e.g., 'https') or the presence of subdomains (e.g., 'www') might carry significance in the classification process. Nowadays, websites that use HTTP are at a higher risk of being attacked, prompting a shift towards HTTPS. If we encounter a URL that employs HTTP, it could indicate either an outdated version of the protocol for a legitimate website, but more often than not, it signifies a potential malicious intent. Utilizing the web without encryption is deemed a significant security vulnerability.

The counts of digits and letters ('count_digits' and 'count_letters', respectively) offer insights into the composition of the URLs. Variations in the distribution of digits and letters across URLs may indicate discernible patterns useful for classification purposes.

Additionally, the 'dir_count' feature quantifies the number of forward slashes ('/') in the URL, reflecting the complexity

of the URL structure. URLs with multiple directories may exhibit different behavioral patterns compared to those with simpler structures, thereby influencing their classification. This was taken into consideration due to the modern structure of websites today. Usually, most websites employ a maximum of two or three levels in their source code. Therefore, if we find ourselves delving deeper than that, there may be aspects that are unclear or require clarification.

Finally, the binary feature 'presence_ip' denotes whether the URL contains an IP address. This feature is particularly pertinent, as malicious URLs may employ IP addresses to obfuscate their identities or evade detection mechanisms. In today's digital landscape, the majority of public URLs utilize domain names, making them easily understandable to users. However, encountering an IP address instead of a domain name is automatically a red flag. IPs are typically associated with backend systems or servers, and their presence in a URL may indicate an attempt to obfuscate the true identity of the website, potentially signaling malicious intent. Therefore, users should avoid entering URLs containing IP addresses.



all features

By leveraging these extracted features, machine learning models can be trained to effectively discern between malicious and white URLs, thereby bolstering cybersecurity measures and mitigating potential threats posed by malicious entities on the web.

VI. ALGORITHMS AND RESULTS

Model Training and Evaluation Overview

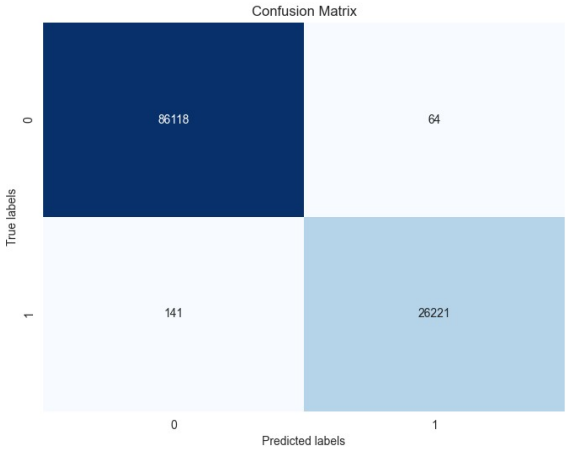
In the pursuit of building an effective classification system, a systematic approach to model training and evaluation is imperative. This process involves several key steps, each contributing to the development and assessment of the model's performance.

1. Preparation: To initiate the model training process, the extracted features undergo compilation into a comprehensive feature set. Subsequently, the dataset is partitioned into training and testing sets using the train test split function from the widely utilized sklearn library. This ensures a robust evaluation of the model's generalization capabilities.

2. Random Forest Classifier: The crux of the model lies in the utilization of a Random Forest Classifier. This sophisticated ensemble learning method proves particularly potent for classification tasks. Operating by constructing a multitude of decision trees during the training phase, the Random Forest outputs the mode of the classes (classification) determined by individual trees. This approach enhances the model's predictive accuracy and robustness.

3. Model Evaluation: The assessment of the Random Forest model hinges on the test set, with accuracy being the primary metric of interest. The model exhibits an impressive accuracy score of 99.8178 % , showcasing its proficiency in correctly classifying instances. Beyond accuracy, a detailed examination is conducted through the generation of a confusion matrix. This matrix provides insights into the model's performance, breaking down results into true positives, false positives, true negatives, and false negatives. Such granularity aids in understanding the model's strengths and areas for improvement.

Incorporating the insights derived from the confusion matrix contributes to a nuanced understanding of the model's behavior, allowing for targeted improvements and refinements. This iterative evaluation process is essential for ensuring the model's reliability and effectiveness in real-world applications.



confusion matrix

4. Model Saving: In the culmination of the training process, the accomplished model is safeguarded for future use. This is achieved by saving the trained model to a .joblib file using the joblib.dump function. This strategic preservation enables the seamless loading of the model for subsequent predictions, obviating the need for time-consuming retraining.

In essence, the outlined model training and evaluation methodology not only establishes a robust foundation for classification tasks but also facilitates the future deployment of the trained model, ensuring efficiency and sustainability in predictive capabilities. This meticulous approach, coupled with the remarkable accuracy scores, serves as a cornerstone in the development of reliable and scalable machine learning solutions.

VII. SUMMARY

This project presents a robust and encompassing machine learning pipeline designed for the in-depth analysis of URLs, specifically targeting the identification of potential malicious ones. The pipeline spans crucial stages, encompassing meticulous data loading, preprocessing, sophisticated feature engineering, exploratory data analysis, comprehensive model training, rigorous evaluation, and strategic model preservation for future use. With a primary focus on automating the detection of harmful URLs, this approach emerges as a pivotal advancement in the realm of cybersecurity. It not only provides a systematic and efficient framework but also serves as a proactive means of identifying and mitigating potential threats, making it an invaluable contribution to the field.

REFERENCES

[1] Mondal, D.K., Singh, B.C., Hu, H., Biswas, S., Alom, Z., & Azim, M.A. (2021). SeizeMaliciousURL: A novel learning approach to detect malicious URLs. Journal of Information Security and Applications, 62, 102967. https://doi.org/10.1016/j.jisa.2021.102967

[2] Abad, S., Gholamy, H., & Aslani, M. (2023). Classification of Malicious URLs Using Machine Learning. Sensors, 23(18), 7760. https://doi.org/10.3390/s23187760

[3] Aljabri, M., Altamimi, H.S., Albelali, S.A., Al-Harbi, M., Alhuraib, H.T., Alotaibi, N.K., Alahmadi, A.A., Alhaidari, F., Mohammad, R.M.A., & Salah, K. (2022). Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions. *IEEE Access*, 10, 1-1. doi:10.1109/ACCESS.2022.3222307

[4] Ferreira, M. (2019). Malicious URL Detection using Machine Learning Algorithms. In *Proceedings of the Digital Privacy and Security Conference 2019* (pp. 114). doi:10.11228/dpsc.01.01

[5] VirusTotal. Retrieved from https://www.virustotal.com/

[6] PhishTank. Retrieved from https://www.phishtank.com/

[7] Kaspersky Threat Intelligence Portal. Retrieved from https://www.kaspersky.com/enterprise-security/intelligence-portal

[8] Google Safe Browsing. Retrieved from https://safebrowsing.google.com/

[9] FireEye Threat Research. Retrieved from https://www.fireeye.com/cyber-threat-intelligence/threat-research.html

[10] Cisco Talos Intelligence. Retrieved from https://talosintelligence.com/