Detecting Phishing Uniform Resource Locator (URL) using Machine Learning

Roth Bahuang
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
roth.bahuang@gmail.edu.my

Delina Beh Mei Yin
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
delina@unikl.edu.my

***Abstract* – Due to the ever-increasing threat from phishing, internets users are in dire need of a system that could help them verify if the websites that they are visiting are safe and legitimate in real-time. Therefore, we developed a Chrome browser extension powered by an ML model to perform the website's classification in this project. Three ML classifiers, Logistic Regression (LR), Naïve Bayes (NB) and Support Vector Machine (SVM), were selected and trained with a dataset from UCI. The dataset contained 30 features and 11055 samples. Five features dependent on third parties were dropped to improve the classification speed. They were the age of the domain, DNS record, page rank, web traffic and Google index.**

We performed hyperparameter tuning using Grid Search and then trained the ML classifiers using the optimal values. After that, we evaluated the ML models based on accuracy and SVM accuracy was highest at 95.80%, followed by LR (92.03%) and NB (88.91%). Thus, the SVM model was selected as the classification model for this research.

Keywords—Phishing; Detection; Support Vector Machine; Logistic Regression, Naïve Bayes, Chrome.

I. INTRODUCTION

Phishing is an online fraud in which people are led to a phishing website that looks and feels just like the real one. Typically, an email that appears from a trusted source, such as co-workers, banks, or government agencies, is sent to the victims. The email contains malicious software or links to malicious websites that attack the user's machine. If the users click the link, they will be directed to a phishing website. Failure to identify the phishing website will lead the victim to reveal sensitive information such as usernames, passwords, and credit card details. The information gathered then will be used to impersonate victims and execute financial transactions on their behalf, thus, placing them in financial and emotional distress [1].

Phishing attacks are not just for getting information from one person; they may also be used to get information from a whole corporation. In 2018, a phishing email was sent to employees at Presbyterian Healthcare in New Mexico. As a result, the attackers were able to use the phished account to obtain details about 183,370 people's healthcare plans, including their names, dates of birth, and social security numbers. It took more than a month for the company to realise they had been targeted in an attack[2]. Phishing can occur across various platforms, including the internet, text messaging, and the phone. Most targeted channels include email, instant messaging, smishing (short message phishing), vishing (voice phishing), and websites.

As our world is becoming more digitalised, attackers can now have more potential victims, thus motivating them to explore and launch more sophisticated phishing attacks. In 2020 alone, the number of phishing attacks observed by APWG members doubled over the year. Attacks peaked in October 2020, with a high of 225,304 new phishing websites, breaking all previous monthly records [3]. Traditional phishing detection methods such as blacklisting face significant challenges due to many phishing websites; the more significant the volume, the longer it takes to add to the blacklist. The longer the gap between the verification of phishing URL until it is added to the blacklist, the more vulnerable the system is to a zero-hour attack. Furthermore, more advanced tactics such as URL obfuscation and abuse of Transport Layer (TLS) or Secure Socket Layer (SSL) certificate are being used by attackers. These tactics can trick even the savviest users.

Therefore, in this research, we attempted to address the drawback of the traditional method by creating a Chrome browser extension that can classify if the URL is safe or phishing in real-time. A warning page will be displayed if the URL is classified as phishing.

II. LITERATURE REVIEW

Researchers have proposed various methods to detect phishing websites or URLs. These methods can be categorised into traditional and Machine Learning Methods [4]-[7].

A. Traditional Methods

Traditional methods are divided into two categories: The human-based Approach and the List-Based Approach. Since a phishing attack is an attempt to take advantage of the users' ignorance or inexperience, an obvious solution is educating the users about identifying a phishing website, hoping to reduce their susceptibility to phishing attacks. Over the years, numerous user training approaches have been proposed, such as web-based training, interactive game-based training, contextual training, embedded training, and non-embedded training to build and enhance users' knowledge about cyber threats [8].

According to [9], security educational interventions have a significant role in combating phishing attacks by transforming “human” or users from the weakest link in cybersecurity to the most vigorous defence. But nowadays, due to the advancement of technology, some phishing websites are hard to distinguish from safe or legitimate websites, and they can fool even the most sophisticated users [10], [11]. Since security awareness training alone is ineffective in preventing users from falling victim to phishing attacks, alternate approaches such as software enhancement are needed.

The list-based approach is classified into two categories: whitelist and blacklist. Whitelist phishing detection systems save secure and genuine URLs. Each website that is not on the whitelist is regarded as suspicious. [12] proposed an automated individual white-list strategy. The system remembers the user's last login and alerts the user when unusual access occurs. Although the whitelisting technique appears to be successful for phishing detection, a robust system with high accuracy requires an extensive list of reputable websites; otherwise, false-positive rates grow because it treats websites not listed as suspicious.

A blacklist, on the other hand, is a list of known fraudulent or phishing websites that includes IP address information, domain names, or URLs. Two prominent blacklist-based websites are phishtank.com and vxvault.net. Blacklist provides credible validation of whether the site is malicious since it is based on real feedback from those who discovered it and were impacted by it. However, a blacklist is reactive because the malicious websites need to be found first before they can be identified, verified as malicious and then added to the list. Updating the list is relatively slow and makes it susceptible or vulnerable to zero-day attacks [13].

According to [14], blacklists are updated at different rates. They projected that 12 hours after they lunched, 47 per cent to 83 per cent of phishing URLs get added to blacklists. Furthermore, the authors discovered that zero-hour protection provided by leading blacklist-based toolbars had an actual positive rate of 15 to 40 per cent. This study showed that keeping the blacklist up to date is critical. Sharifi et al. (2008) developed a novel blacklist generation methodology to solve this problem. They ran an experiment to compare the URLs acquired from the email with those gathered from the Google search engine for the actual organisations. However, their suggested method performed badly because it relied on third-party services (such as Google) to search domain names and compare top results [4].

Most modern browsers employ this method to keep their users safe from phishing; Google Chrome and Firefox browsers use Google Safe Browsing and Internet Explorer using Microsoft’s SmartScreen filter [15].

B. Machine Learning Methods

Machine learning is a subfield of artificial intelligence (AI) that focuses on developing systems that can learn from data and improve their accuracy over time without being taught to do so [16]. Machine learning can be broken down into two main categories: supervised and unsupervised [17]. Supervised machine learning models are trained on labelled data, and the model is aware of the input and desired output. The model creates a function to explain the connection between the input and output data based on this information. This function may then be used to anticipate the intended result given new data. Unsupervised machine learning methods, on the other hand, are employed when the data being trained is neither categorised nor labelled. Unsupervised learning studies how computers might infer a function from unlabelled data to describe a hidden structure.

Detecting phishing websites or URLs is a classification problem. Therefore, any supervised machine learning algorithm can be used. According to the survey conducted by [4], the most used machine learning algorithms for phishing detection are Naïve Bayes, Logistic Regression, Random Forest, Support Vector Machine, Decision Tree, Convolutional Neural Network, Decision Tree, Long Short-Term Memory and K-Means.

In machine learning, features of the dataset are the most critical factor that can influence the accuracy of the machine learning model. Three most popular features used to train ML classifiers: content-based, visual-based, and URL-based features.

The Content-Based are based on scanning the contents of the website. Therefore, content-based features may be gained by downloading the complete webpage. A webpage's content-based characteristics may be derived mainly from its HTML text and the use of JavaScript [18]. The most used features are document length, average word length, word count, distinct word count, word count in a line, number of NULL characters, string concatenation, distinctive HTML elements, links to distant script sources, and hidden objects, the number of iframes, the number of zero size iframes, the number of lines, and the number of hyperlinks and statistical aspects of the page [19]-[22]. This technique required a large quantity of storage to hold data, a high transmission cost, and the risk of downloading harmful scripts or payloads when online content is extracted. [23].

The visual-based approach utilised the visual similarity between websites to identify phishing websites. The technique relied on the fact that the attacker always uses websites that look like genuine websites to trick their victims into entering their sensitive information. The most used visual features are document object model (DOM) tree, visual features, cascading style sheet (CSS) similarity, pixel-based and visual perception [24]-[27]. Computation of visual similarity is usually performed by various algorithms such as Earth Mover’s Distance (EMD), optical character recognition (OCR) and Speed-up Robust Feature [27], [28]. The drawback of this approach is it is longer than other approaches since it involves complex computational to compare the websites.

Lastly, the URL-Based features are based on the textual properties obtained from the URL itself. Several features can be extracted from an URL. The standard features are shown in Table 2.1 [29], [30].

TABLE I. COMMONLY USED URL-BASED FEATURES

#	Feature Name	Description
1	IP address	Check if IP address is present in existing domains
2	Average words length	Count average length of meaningful words in the entire domain name
3	exe/zip	Check if exe/zip is present in the URL
4	Special symbols	Count special symbols in the URL
5	Numbers of dots	Count number of dots in URL
6	URL length	Count number of characters in URL
7	Top-level domain (TLD) feature	Validate TLD-based features.
8	"HTTP" count	Count the number of "HTTP" in the URL
9	Brand name	Extract brand name in a URL domain
10	"//" redirection	Check if "//" is included in the URL path
11	Domain separated by '-'	Check if '-' is included in the domain name
12	Multi-sub domain	Check how many # of multi-subdomains is included in the URL
13	Suspicious words	Check if suspicious words are included in the URL
14	Digits in domain	Number of digits in domain
15	Character entropy	Calculate character distribution in the entire URL using entropy.
16	Shorten URL	Check if the URL is shortened

URL-based methods performed faster than any other approach because they do not need high computational power or large storage and are independent of a third party. It can also detect zero-hour phishing attacks, which are becoming a significant concern in modern anti-phishing techniques [4].

C. *Comparison of Machine Learning Methods*

Software detection approaches include list-based, content-based, visual-based, and URL-based. In general, a software-based phishing detection approach outperforms a human-based approach since humans are prone to making casual errors, such as disregarding a website's security warning (Park et al., 2014). The advantages and disadvantages of each approach are analysed to discover the optimal technique for this research. The comparison of phishing detection techniques is shown in Table II.

TABLE II. COMPARISON OF PHISHING DETECTION TECHNIQUES

Technique	Advantages	Disadvantages
List-Based	- Low false positive - Low computational cost on the server-side	- Susceptible to zero-hour attack.
Content-Based	- Able to detect a zero-hour attack	- Higher false positive than List-Based. - Need high computational cost.
Visual-Based	- Able to detect a zero-hour attack	- Need high computational and storage costs.
URL-Based	- Able to detect zero-hour attacks. - Fast detection (suitable for real-time detection)	- Some features such as website rank susceptible to Search Engine Optimisation (SEO) poisoning.

Since no method can solve the phishing problem by itself, therefore for this research, Machine Learning using content-based and URL-based techniques were selected.

III. Methodology

A. *System Design /Architecture*

The system design flow for this research is shown in Fig. 1. When the users visit any website; the URL will be sent to the server. At the server, the URL features will be extracted. After that, the features are fed to the ML model, and then the prediction is performed. Once the prediction is obtained, it will be sent to the extension (browser). If the prediction is safe, the safe button will be displayed and if phishing, the warning page will be displayed.

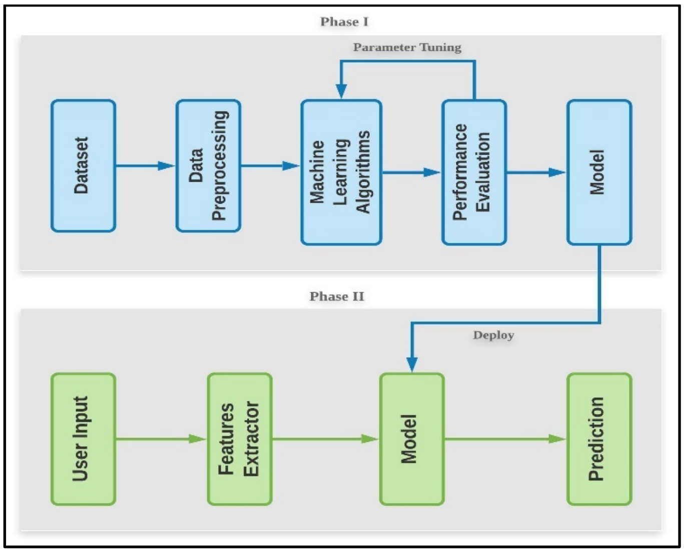


Fig. 1. System Design

This project was divided into two phases: Application Programming Interface (API) and Extension Development Phase. Machine Learning model and features extractor are performed in the API development phase and user interfaces (UIs) for the browser extension were designed and developed during the Extension Development phase.

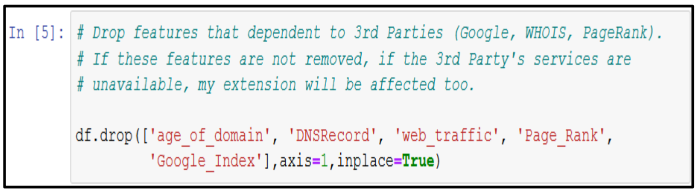
B. Dataset

The phishing website dataset is taken from UCI Machine Learning Repository. It consists of 11,055 URLs: 6157 phishing URLs and 4898 Safe URLs. Each URL in the dataset contains 30 features. The features can be divided into four categories: Address Bar-Based Features, Abnormal-Based Features, HTML and JavaScript-Based Features and Domain-based Features. The details about the features and rules are shown in Table III.

TABLE III. DATASET FEATURES

Feature Class	Feature Name
Address Bar-based Features	Using the IP Address
	Long URL to Hide the Suspicious Part
	Using URL Shortening Services
	URL contains “@” Symbol
	Redirecting using double slashes (//)
	Adding Prefix or Suffix Separated by (-) to the Domain
	Sub Domain and Multi-Sub Domains
	HTTPS
	Domain Registration Length
	Favicon
	Using Non-Standard Port
	The Existence of “HTTPS” Token in the Domain Part of the URL
Abnormal Based Features	Request URL
	URL of Anchor
	Links in <Meta>, <Script> and <Link> tags
	Server Form Handler (SFH)
	Submitting Information to Email
	Abnormal URL
HTML and JavaScript-based Features	Website Forwarding
	Status Bar Customization
	Disabling Right Click
	Using Pop-up Window
	IFrame Redirection
Domain-based Features	Age of Domain
	DNS Record
	Website Traffic
	PageRank
	Google Index
	Number of Links Pointing to Page
	Statistical-Reports Based Feature

Five features that depend on third parties were dropped to speed up the features extraction process.



```
In [5]: # Drop features that dependent to 3rd Parties (Google, WHOIS, PageRank).
        # If these features are not removed, if the 3rd Party's services are
        # unavailable, my extension will be affected too.

        df.drop(['age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
                 'Google_Index'],axis=1,inplace=True)
```

Fig. 2. Manual Feature Selection

C. Machine Learning Algorithms

Several supervised machine learning classifiers have been identified during the literature review that can perform well in website URL classification, such as Random Forest (RF), Support Vector Machine (SVM), Naïve Bayes (NB) Logistic Regression (LR) and K-Nearest Neighbours (KNN). For this project, SVM, NB and LR have been selected. The split ratio for the training and testing set was 80:20.

D. Evaluation Metrics

The performance of the ML model was evaluated using a confusion matrix. The confusion matrix is shown in using Table IV.

TABLE IV. CONFUSION MATRIX

Actual Classification	**Prediction Classification**	
	Positive	**Negative**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Note: Adapted from [32]

Based on the values of TP, FP, FN and TN from the confusion matrix, the accuracy, precision, recall, and F1-score of the models can be calculated as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (3)$$

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \qquad (4)$$

IV. RESULT AND DISCUSSION

A. Hyperparameter Tuning Result

The Grid Search result for optimal hyperparameter values and the tuned hyperparameters are shown in Table V.

Table V. Optimal Hyperparameter Values

Classifier	Optimal Hyperparameter Values
LR	C = 0.01, penalty = 'l2'
NB	var_smoothing = 1.0
SVM	C=10, gamma=0.1, kernel= 'rbf'

B. ML Model Evaluation Result

The confusion matrix for ML models is shown in Table VI. This result was obtained after the optimal hyperparameter values were applied during training.

TABLE VI. CONFUSION MATRIX FOR LR, NB AND SVM

Classifier	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)
LR	886	101	1149	75
NB	888	99	1078	146
SVM	935	52	1183	41

Table VI showed that the SVM model has the highest TP and TN prediction and lowest FP and FN prediction. Therefore, it is the highest performing ML model regarding the accuracy, as shown in Table VII.

TABLE VII: ACCURACY SCORE BEFORE AND AFTER HYPERPARAMETER TUNING

Classifier	Accuracy Score (Before tuning)	Accuracy Score (After tuning)
SVM	93.67%	95.80%
LR	91.54%	92.03%
NB	61.06%	88.91%

Table VII also showed that hyperparameter tuning benefited all ML classifiers, especially Naïve Bayes, where its accuracy had increased significantly from 61.06% to 89.51%. Therefore, it is recommended to perform hyperparameter tuning. For this project, SVM was selected as it is the best performing model.

C. ML Model Reliability Testing

This testing was conducted to ensure that the system is reliable and classify the URL correctly when deployed as an extension on the Chrome browser. Twenty (20) safe URLs and phishing URLs were tested, and the ML model was able to classify all safe websites correctly. However, for phishing URLs, it failed to classify two of them correctly. Overall, the ML model accuracy during the testing and production was acceptable, and there is no significant gap in the accuracy rate. Therefore, this extension is reliable and can differentiate between safe and phishing URLs.

V. CONCLUSION AND SUGGESTION

A. Conclusion

Due to the ever-increasing threat from phishing, researchers have proposed various solutions such as traditional and ML methods. In this research, we developed a Chrome browser extension that can classify any URL in real-time. Classification of URLs in real-time is needed because of the traditional method such as blacklist unable to detect new phishing websites. From the ML evaluation and ML reliability testing results, the objective of this research was met.

B. Limitation

Since phishing URLs have a short lifespan, it is difficult to get a higher testing sample for testing since some of the features require the website to be online. Due to this and time constrain, the reliability testing sample was small; only 20 verified phishing URLs from phishtank.com were tested.

C. Suggestion

Our suggestions for future works focus on increasing the users' coverage, classification speed and less interruption when using our extension. The suggestions are:

1. *Include Microsoft Edge and Firefox.*
   The more browser included, the more users protected. This is because users have varied browser preferences.

2. *Add Whitelist on the client-side.*
   Adding a whitelist on the client-side will speed up the classification process and decrease the computational power needed on the server-side. This also enables users to have a pleasant experience when browsing the internet.

3. *Use a bigger dataset.*
   Then dataset size for this research was small (11,055 samples). Therefore, for a future project, it is recommended to allocate more time to generate a bigger dataset since ML accuracy depends on how much they learn during the training; the bigger the dataset, the higher the accuracy.

References

[1] A. Abdulwakil, M. A. Aydin, and D. Aksu, “Detecting phishing websites using support vector machine algorithm,” *Pressacademia*, vol. 5, no. 1, pp. 139–142, Jun. 2017, doi: 10.17261/Pressacademia.2017.582.

[2] D. Barker, “Records of 85,000 involved in hospital hack | The Daily World,” Feb. 13, 2019. https://www.thedailyworld.com/news/records-of-85000-involved-in-hospital-hack/ (accessed Jun. 19, 2021).

[3] APWG, “Phishing Activity Trends Reports,” 2020. https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf (accessed Feb. 20, 2021).

[4] E. S. Aung, T. Zan, and H. Yamana, “A Survey of URL-based Phishing Detection,” pp. 1–8, 2019, [Online]. Available: https://db-event.jpn.org/deim2019/post/papers/201.pdf

[5] A. K. Jain and B. B. Gupta, “Towards detection of phishing websites on client-side using machine learning based approach,” *Telecommunication Systems*, vol. 68, no. 4, pp. 687–700, Aug. 2018, doi: 10.1007/s11235-017-0414-0.

[6] N. Dholakia and P. Agrawal, “Review on Phishing Attack Detection Techniques,” *ASIAN JOURNAL OF CONVERGENCE IN TECHNOLOGY*, vol. 6, no. 2, pp. 41–47, Aug. 2020, doi: 10.33130/AJCT.2020v06i02.008.

[7] M. H. Alkawaz, S. J. Steven, and A. I. Hajamydeen, “Detecting Phishing Website Using Machine Learning,” in *Proceedings - 2020 16th IEEE International Colloquium on Signal Processing and its Applications, CSPA 2020*, Feb. 2020, pp. 111–114. doi: 10.1109/CSPA48992.2020.9068728.

[8] M. M. Al-Daeef, N. Basir, and M. M. Saudi, “Security awareness training: A review,” *Lecture Notes in Engineering and Computer Science*, vol. 2229, pp. 446–451, 2017.

[9] N. A. G. Arachchilage, S. Love, and K. Beznosov, “Phishing threat avoidance behaviour: An empirical investigation,” *Computers in Human Behavior*, vol. 60, pp. 185–197, 2016, doi: 10.1016/j.chb.2016.02.065.

[10] N. M. Shekokar, C. Shah, M. Mahajan, and S. Rachh, “An ideal approach for detection and prevention of phishing attacks,” *Procedia Computer Science*, vol. 49, no. 1, pp. 82–91, 2015, doi: 10.1016/j.procs.2015.04.230.

[11] M. Fernando and N. A. G. Arachchilage, “Why Johnny can’t rely on anti-phishing educational interventions to protect himself against contemporary phishing attacks?,” *arXiv*, pp. 1–12, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.13262

[12] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” *Expert Systems with Applications*, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.

[13] A. Ali Ahmed, “Malicious Website Detection: A Review,” *Journal of Forensic Sciences & Criminal Investigation*, vol. 7, no. 3, Feb. 2018, doi: 10.19080/JFSCI.2018.07.555712.

[14] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An Empirical Analysis of Phishing Blacklists,” 2009.

[15] S. Wedyan and F. Wedyan, “Journal of Emerging Trends in Computing and Information Sciences An Associative Classification Data Mining Approach for Detecting Phishing Websites,” vol. 4, no. 12, 2013, [Online]. Available: http://www.cisjournal.org

[16] IBM, “What is Machine Learning? - Malaysia | IBM,” Jul. 15, 2020. https://www.ibm.com/my-en/cloud/learn/machine-learning (accessed Jun. 19, 2021).

[17] J. Delua, “Supervised vs. Unsupervised Learning: What’s the Difference? | IBM,” Mar. 12, 2021. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (accessed Jun. 19, 2021).

[18] D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,” vol. 2657, pp. 1–9, Jan. 2017, Accessed: May 30, 2021. [Online]. Available: https://arxiv.org/pdf/1701.07179.pdf

[19] H. Choi, B. B. Zhu, and H. Lee, “Detecting malicious web links and identifying their attack types,” *WebApps*, p. 11, 2011, [Online]. Available: http://dl.acm.org/citation.cfm?id=2002168.2002179

[20] D. Canali, M. Cova, G. Vigna, and C. Kruegel, “Prophiler: A fast filter for the large-scale detection of malicious web pages,” *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, pp. 197–206, 2011, doi: 10.1145/1963405.1963436.

[21] Y. T. Hou, Y. Chang, T. Chen, C. S. Laih, and C. M. Chen, “Malicious web content detection by machine learning,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 55–60, Jan. 2010, doi: 10.1016/j.eswa.2009.05.023.

[22] J. A. Jupin, T. Sutikno, M. A. Ismail, M. S. Mohamad, S. Kasim, and D. Stiawan, “Review of the machine learning methods in the classification of phishing attack,” *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1545–1555, Dec. 2019, doi: 10.11591/eei.v8i4.1344.

[23] H. Yuan, Z. Yang, X. Chen, Y. Li, and W. Liu, “URL2Vec: URL Modeling with Character Embeddings for Fast and Accurate Phishing Website Detection,” in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, Dec. 2018, pp. 265–272. doi: 10.1109/BDCloud.2018.00050.

[24] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, “Detection of phishing webpages based on visual similarity,” in *Special interest tracks and posters of the 14th international conference on World Wide Web - WWW ’05*, 2005, p. 1060. doi: 10.1145/1062745.1062868.

[25] A. P. E. Rosiello, E. Kirda, C. Kruegel, and F. Ferrandi, “A layout-similarity-based approach for detecting phishing pages,” in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, 2007, pp. 454–463. doi: 10.1109/SECCOM.2007.4550367.

[26] J. Mao, P. Li, K. Li, T. Wei, and Z. Liang, “BaitAlarm: Detecting Phishing Sites Using Similarity in Fundamental Visual Features,” in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, Sep. 2013, pp. 790–795. doi: 10.1109/INCoS.2013.151.

[27] M. Dunlop, S. Groat, and D. Shelly, “GoldPhish: Using Images for Content-Based Phishing Analysis,” in *2010 Fifth International Conference on Internet Monitoring and Protection*, 2010, pp. 123–128. doi: 10.1109/ICIMP.2010.24.

[28] A. Y. Fu, L. Wenyin, and X. Deng, “Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover’s Distance (EMD),” *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, Oct. 2006, doi: 10.1109/TDSC.2006.50.

[29] S. C. Jeeva and E. B. Rajsingh, “Intelligent phishing url detection using association rule mining,” *Human-centric Computing and Information Sciences*, vol. 6, no. 1, p. 10, Dec. 2016, doi: 10.1186/s13673-016-0064-3.

[30] I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur, “A Novel Machine Learning Approach to Detect Phishing Websites,” in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb. 2018, pp. 425–430. doi: 10.1109/SPIN.2018.8474040.

[31] G. Park, L. M. Stuart, J. M. Taylor, and V. Raskin, “Comparing machine and human ability to detect phishing emails,” in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2014, vol. 2014-January, no. January, pp. 2322–2327. doi: 10.1109/smc.2014.6974273.

[32] A. F. Nugraha and L. Rahman, “Meta-Algorithms for Improving Classification Performance in the Web-phishing Detection Process,” in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2019, vol. 6, pp. 271–275. doi: 10.1109/ICITISEE48480.2019.9003952.