e-ISSN: 2686-2573
DOI: 10.21512/emacsjournal.v6i3.11844

Machine Learning-Based Malicious Website Detection Using Logistic Regression Algorithm

Puan Bening Pastika [1*], **Alamsyah**[2]

[1,2] Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang,
Semarang, Indonesia 50229
puanbening04@students.unnes.ac.id; alamsyah@mail.unnes.ac.id

*Correspondence: puanbening04@students.unnes.ac.id

Abstract – *Cybercrime is an increasing threat that occurs while exploring the internet. Cybercrime is committed by cybercriminals who exploit the web's vulnerability by inserting malicious software to access systems that belong to web service users. It is detrimental to users, therefore detecting malicious websites is necessary to minimize cybercrime. This research aims to improve the effectiveness of detecting malicious websites by applying the Logistic Regression algorithm. The selection of Logistic Regression is based on its ability to perform binary classification, which is important for distinguishing between benign and potentially malicious websites. This research emphasizes a preprocessing stage that has been deeply optimized. Data cleaning, dataset balancing, and feature mapping are enhanced to improve detection accuracy. Hybrid sampling addresses data imbalance, ensuring the model is trained with representative data from both classes. Experimental results show that the Logistic Regression implementation achieves an excellent level of accuracy. The developed model recorded an accuracy of 92.60% without cross-validation, which increased to 92.71% with 5-fold cross-validation. The novelty of this research lies in the significant increase in accuracy compared to previous methods, demonstrating the potential to improve protection against malicious website threats in an increasingly complex and risky digital environment. This research makes an important contribution to the development of digital security detection technologies to address the ever-growing challenges of cybercrime.*

Keywords: *Malicious Website; Machine Learning; Logistic Regression*

I. INTRODUCTION

The internet has become essential for communicating with the outside world (Kavici & Ayaz-Alkaya, 2024). People can interact with web services to perform their daily tasks through the internet. Cyberthieves exploit the interaction to harm online service consumers via malicious websites that they have mainly built (Saleem Raja et al., 2021). Malicious websites collect visitors' personal information when visiting them (Mondal et al., 2021) The goal of constructing a dangerous website may involve installing malware, collecting personal information, or exposing data (A. Saleem Raja et al., 2023). Most users need an essential understanding of how to browse the internet correctly. Users are less likely to be able to distinguish between malicious and non-malicious websites (Mondal et al., 2021). Therefore, detecting malicious websites is necessary to minimize cybercrimes.

Machine learning is a powerful technology for combating cyberattacks (Prasad & Chandra, 2024) and it can be utilized to detect malicious websites. Previous studies have proven the effectiveness of machine learning in detecting malicious websites. In (Mohamad Arifandy & Septia Ulfa Sunaringtyas, 2021), machine learning is applied to detect malicious websites through classification based on web page features. The machine learning model designed uses the method of Wang et al. (2017) with the Decision Tree algorithm. The results showed that the machine learning model with the Decision Tree algorithm performed best, with an accuracy of 0.921, precision of 0.925, and f-measure of 0.914. The Decision Tree algorithm shows better performance than other algorithms such as naïve Bayes (accuracy 0.738, precision 0.645, and f-measure

0.773) and support vector machine (SVM) (accuracy 0.802, precision 0.738, and f-measure 0.807). In (Aprelia Windarni et al., 2023) the Pearson correlation filter feature is utilized by applying three machine learning methods: Naïve Bayes, Decision Tree, and Random Forest to determine the most effective method in detecting web phishing. The results showed that the Naïve Bayes method had an accuracy of 60.4%, Decision Tree 94.4%, and Random Forest 96.3%. The Random Forest method was the most effective, with 96.3% accuracy. In (Jalil et al., 2023), machine learning-based URL phishing detection was conducted. The proposed technique for classifying phishing URLs involves analyzing various URL components, including full URL, protocol scheme, hostname, path area, entropy features, suspicious words, and brand name matching using the TF-IDF technique. Experiments were conducted on six different datasets using eight different machine learning classifiers, with Random Forest achieving the highest accuracy in all datasets. The framework, which uses only 30 features, gained 96.25% and 94.65% accuracy on the Kaggle dataset. Comparison results show that the model achieved accuracies of 92.2%, 91.63%, 94.80%, and 96.85% on benchmark datasets, exceeding existing approaches. In (Ramadhan, 2023), Malicious URL detection is done by extracting lexical features from URLs using the Random Forest algorithm. The dataset utilized has a high level of imbalance, which Random Oversampling overcomes. Model testing focused on optimizing lexical features to classify malicious URL types (benign, defacement, malware, phishing) with variations of 10, 15, 19, and 23 features, using 8-fold cross-validation. Experimental results show the best accuracy improvement, reaching 97.6% using 23 lexical features. A challenge faced is the detection of static URLs without a "/" at the end, which are often misclassified as phishing. This research provides important insights for optimizing malicious URL detection using the Random Forest algorithm, which is relevant in the face of the complexity of today's cyber threats.

Previous studies have shown the effectiveness of several machine learning algorithms for detecting malicious websites. This research aims to detect malicious websites by applying one of the machine learning methods, Logistic Regression. The Logistic Regression algorithm approach is utilized to solve binary classification problems by estimating the probability of belonging to one of two (Vajrobol et al., 2024), i.e., malicious and benign websites.

II. METHODS

In this research, the focus is on the accuracy obtained by the Logistic Regression algorithm in detecting malicious websites. This research was conducted using the python programming language using Colab tool on one dataset which was divided into 80% for training the model and 20% for testing the model. The research flow undertaken to detect malicious websites shown in Figure 1.

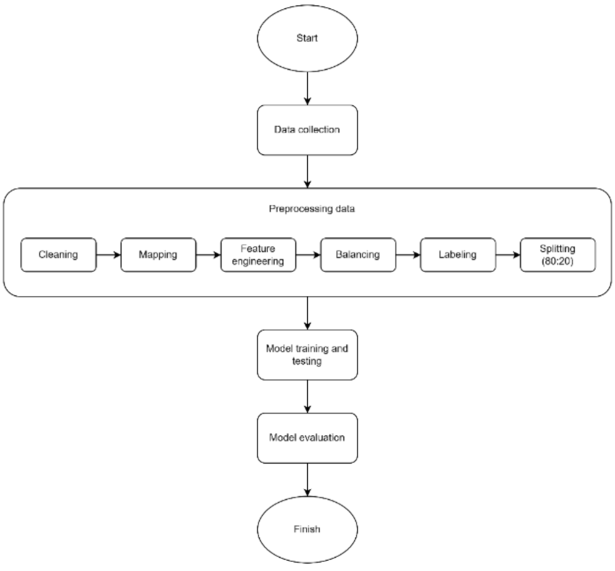

Figure 1. The research flow

Figure 1 depicts the research flow undertaken to detect malicious websites. The research flow consists of four stages. The first stage is data collection. The second stage is data preprocessing, namely by cleaning, mapping, feature engineering, balancing, labeling, and splitting with a ratio of 80:20 for training data and test data. The third stage is training and testing the model using the Logistic Regression algorithm. The last stage, which is evaluating the model with accuracy, precision, recall, and F1-score parameters and using cross-validation techniques. The following below is a more detailed approach to the flow of research conducted.

2.1 Data collection

This research utilizing a dataset obtained from Kaggle (*Malicious URLs Dataset*, n.d.). The Kaggle dataset contains 651,191 URLs organized into four classes, there are benign, defacement, phishing, and malware. There are 428,103 benign URLs, 96,457 defacement URLs, 94,111 phishing URLs, and 3,520 malware URLs.

2.2 Preprocessing data

There are several steps involved in preprocessing data. The steps taken in preprocessing data include cleaning, mapping, feature engineering, balancing, labeling, dan splitting. The cleaning process is done to clean the data from missing values (Doshi, 2011). The mapping process serves to create a target map (Chung & Fabbri, 2020) from the research target, which initially has 4 classes, namely benign, defacement, phishing, and malware, narrowed down to 2 classes, namely good (benign) and bad (malicious). The labeling process serves to provide numerical labels for the class of benign websites and malicious websites to facilitate the model of the Logistic Regression algorithm in classifying data. Next, a feature engineering process was carried out to build variables (Tiwari & Rana, 2021) which amounted to 23 features is shown in Table I. Table I provides a detailed list of extracted features along with their feature names and descriptions based on the URL, host, and area path of the URL.

Table I. List of 23 features

#	Feature names	Feature descriptions
1	url_length	Length of the URL measured in the number of characters.
2	hostname_length	Length of the hostname in the URL measured in the number of characters.
3	count-www	Number of occurrences of "www" in the URL.
4	count@	Number of occurrences of the "@" character in the URL.
5	count_dir	Number of directories in the URL path.
6	count_embed_domain	Number of occurrences of double slashes ("//") in the URL.
7	short_url	Indicates whether the URL uses a URL shortening service.
8	count-https	Number of occurrences of "https" in the URL.
9	count-http	Number of occurrences of "http" in the URL.
10	count%	Number of occurrences of the "%" character in the URL.
11	count?	Number of occurrences of the "?" character in the URL.
12	count-	Number of occurrences of the hyphen ("-") in the URL.
13	count=	Number of occurrences of the equal sign ("=") in the URL.
14	url_length	Length of the URL measured in the number of characters.
15	hostname_length	Length of the hostname in the URL measured in the number of characters.
16	sus_url	Indicates whether the URL contains suspicious words related to online scams.
17	fd_length	Length of the first directory in the URL path.
18	tld_length	Length of the top-level domain (TLD) in the URL measured in the number of characters.
19	count-digits	Number of digit characters (numbers) in the URL.
20	count-letters	Number of letter characters (alphabet) in the URL.
21	abnormal_url	Indicates whether the URL includes parts of the hostname in its path.
22	use_of_ip_address	Indicates whether the URL uses an IP address.
23	google_index	Indicates whether the URL appears in Google search results.

The balancing process is performed to address data imbalance (Barella et al., 2021). A proper balancing process on unbalanced data can reduce defects and eliminate imbalances in the data (Felix & Lee, 2019). In this research, the balancing process is carried out using a hybrid sampling technique. Hybrid sampling is a combination of oversampling and undersampling balancing techniques to help improve the generalization ability of the model and reduce the possibility of overfitting (Gao et al., 2020) (Qian et al., 2014). This research applies an undersampling method to reduce the number of 'good' class type. In contrast, an oversampling method is applied to increase the number of 'bad' class type. The labeling process serves to give numerical labels (Alobaid et al., 2020) to the class of benign websites and dangerous websites to facilitate the model of the Logistic Regression algorithm in classifying data. The splitting process is carried out to divide the data into training data and test data (Liu et al., 2019) with a ratio of 80:20. The splitting process is useful for controlling the error rate in research (Dai et al., 2023).

2.3 Model training and testing

In classifying data, the training process is carried out so that the model built is able to classify data according to the target research label (Tashev et al., 2022). During the training stage, the algorithm receives input training data and learns patterns (Alban et al., 2020) to be able to distinguish between malicious and non-malicious websites. After the training process is complete, the next step is to perform the testing process. The testing stage aims to test the extent to which the trained algorithm can correctly categorize data when given new data that has never been seen before (Verma et al., 2020). In this research, the model utilized for the testing process is Logistic Regression. The Logistic Regression model that has been built during the training stage, is tested using test data to evaluate its performance in classifying malicious websites according to the target research label.

2.4 Model evaluation

The model is evaluated to measure how well the Logistic Regression model has been built. The performance measurements utilized to evaluate the model in addition to accuracy during model testing are accuracy, precision, recall, and F1-score (Yacouby & Axman, 2020). The description of these performance measurements as follows (Yu et al., 2020):

- ***Accuracy***
  The percentage of valid predictions from observations to the total number of observations is expressed as,

$$\textit{Accuracy} = TP+TN/TP+FP+TN+FN \tag{1}$$

- ***Precision***
  This matrix is a good performance matrix when the false-positives is high written as,

$$\textit{Precision} = TP/TP+FP \tag{2}$$

- ***Recall***
  This matrix is a good performance matrix when the false-negative is high written as,

$$\textit{Recall} = TP/TP+FN \tag{3}$$

- ***F1-score***
  This matrix is a holistic average of precision and recall, written as,

$$\textit{F1-score} = 2\times \textit{Precision}\times \textit{Recall}/\textit{Precision}+\textit{Recall} \tag{4}$$

In addition to the above four matrices, evaluation measurements using cross-validation are also performed. Cross-validation is a technique that estimates the average prediction error of the model against an unseen training set (Bates et al., 2023) to ensure that the model is thoroughly evaluated. This research using cross-validation with 5-fold.

III. RESULTS AND DISCUSSION

3.1 Data collection

The dataset utilized in this research consists of 651,191 URLs grouped into four classes: benign, defacement, phishing, and malware. The benign class consists of 428,103 URLs, the defacement class consists of 96,457 URLs, the phishing class consists of 94,111 URLs, and the malware class consists of 3,520 URLs. The benign class dominates the dataset, while the malware class is the class with the least number of URLs. The dataset consists of 2 columns, the url column which displays the website and the type column which displays the class of the URL. The first 5 rows of the dataset are shown in Table II.

Table II. Display of the first 5 rows of the dataset

URL	Type
br-icloud.com.br	phishing
mp3raid.com/music/krizz_kaliko.html	benign
bopsecrets.org/rexroth/cr/1.htm	benign
http://www.garage-pirenne.be/index.php?option=...	defacement
http://adventure-nicaragua.net/index.php?option=...	defacement

3.2 Preprocessing data

The preprocessing data performed are cleaning, mapping, feature engineering, balancing, labeling, and splitting. To improve accuracy, the cleaning, mapping, and balancing processes were optimized. The cleaning process is done by removing missing values, as shown in Table III.

Table III. Cleaning process result

Before cleaning process		After cleaning process	
Feature	Number of missing values	Fitur	Number of missing values
fd_length	287317	fd_length	0

The original dataset had four unique categories: benign, defacement, phishing, and malware, each reflecting a different sort of website. To make the classification work easier, these groups were combined into two larger categories. The benign websites, which are not malicious in nature, were placed into a single category designated "good." Meanwhile, defacement, phishing, and malware websites, which all represent different types of malicious conduct, were grouped together under a single "bad" category. The results of the mapping process are shown in Table IV.

Table IV. Mapping process result

Before mapping process		After mapping process	
URL	Type	URL	Type
br-icloud.com.br	phishing	br-icloud.com.br	bad
mp3raid.com/music/krizz_kaliko.html	benign	mp3raid.com/music/krizz_kaliko.html	good
bopsecrets.org/rexroth/cr/1.htm	benign	bopsecrets.org/rexroth/cr/1.htm	good
http://www.garage-pirenne.be/index.php?option=...	defacement	http://www.garage-pirenne.be/index.php?option=...	bad
http://adventure-nicaragua.net/index.php?option=...	defacement	http://adventure-nicaragua.net/index.php?option=...	bad

The next data preprocessing process is the feature engineering process to create new variables. There are 23 new variables created, namely url_length, hostname_length, count-www, count@, count_dir, count_embedded_domain, short_url, count-https, count-http, count%, count?, count-, count=, url_length, hostname_length, sus_url, fd_length, tld_length, count-digits, count-letters, abnormal_url, use_of_ip_address, and google_index. The description of each variable or feature has been explained in the methods chapter. An example of the results of applying 5 features from the feature engineering process is shown in Table V.

Table V. Feature engineering process result

URL	Type	URL_ Length	Count-www	Count%	Abnormal_ URL	Google_ index
br-icloud.com.br	bad	16	0	0	0	1
mp3raid.com/music/krizz_kaliko.html	good	35	0	0	0	1
bopsecrets.org/rexroth/cr/1.htm	good	31	0	0	0	1
http://www.garage-pirenne.be/index.php?option=...	bad	88	1	0	1	1
http://adventure-nicaragua.net/index.php?option=...	bad	235	0	0	1	1

The next data preprocessing process is the balancing process to balance the data. The balancing process is done by applying a combination of oversampling and undersampling techniques or also known as Hybrid Sampling. This technique employs oversampling to raise the number of benign (good) classes and undersampling to lower the number of malicious (bad) classes. The results of the balancing process are shown in Table VI.

Table VI. Balancing process result

Before balancing process		After balancing process	
Fitur	Data amount	Fitur	Data amount
bad	428103	bad	140532
good	233088	good	140532

To simplify the model's classification, the labeling process assigned numerical values to these categories, with 1 indicating benign websites and 0 signifying malicious. The results of the labeling process are shown in Table VII.

Table VII. Labeling process result

Before labeling process		After labeling process	
URL	Type	URL	Class_ URL
br-icloud.com.br	bad	br-icloud.com.br	0
mp3raid.com/music/krizz_kaliko.html	good	mp3raid.com/music/krizz_kaliko.html	1

bopsecrets.org/ rexroth/cr/1.htm	good	bopsecrets.org/ rexroth/cr/1.htm	1
http://www. garage-pirenne. be/index. php?option=...	bad	http://www.garage-pirenne.be/index. php?option=...	0
http://adventure-nicaragua. net/index. php?option=...	bad	http://adventure-nicaragua. net/index. php?option=...	0

3.3 Model training and testing

Training and testing of the model done with a ratio of 80% for training data and 20% for test data. The Logistic Regression model is applied using a max_iter parameter of 2000 iterations and a random_state parameter of 42. After training and testing the Logistic Regression model, good results were obtained with training accuracy and testing accuracy shown in Table VIII.

Table VIII. The accuracy of model training and testing

Model	Accuracy	
	Training	Testing
Logistic Regression	92.72%	92.60%

Model evaluation

Model evaluation is carried out using accuracy, precision, recall, and F1-score performance measurements with the results obtained from each class listed in Table IX.

Table IX. Model evaluation

	Model evaluation		
	Precision	Recall	F1-score
Malicious	93.98%	91.30%	92.62%
Benign	91.54%	94.16%	92.83%
Macro average	92.76%	92.73%	92.72%

In addition to using accuracy, precision, recall, and F1-score matrices, testing is also done using cross-validation with 5-fold. The graph of testing using cross-validation with 5-fold is shown in Figure 2.

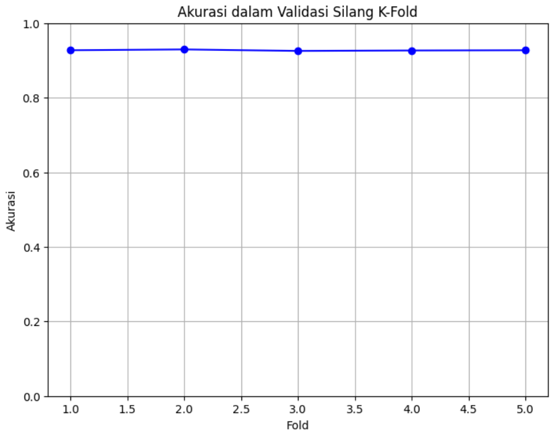

Figure 2. Testing using cross-validation

From Figure 2, it can be seen that the model has consistent accuracy based on evaluation using 5-fold cross-validation. The accuracy line between folds has an almost flat curve with little fluctuation. This shows that the model has a stable, reliable performance in predicting new data, and is not overfitting on certain subsets of the training data. The accuracy comparison without cross-validation and using cross-validation is shown in Table 10.

Table X. Comparison of accuracy without cross-validation and using cross-validation

Model	Model evaluation	
	Without cross-validation	Using cross-validation
Logistic Regression	92.60%	92.71%

3.4 Model performance measurement comparison

The algorithm's performance is evaluated by comparing it to test results from previous research. Previous research tests used the same Logistic Regression algorithm to detect dangerous websites. Limiting the comparison to the same method guarantees a fair assessment of the Logistic Regression model's performance. This method enables immediate assessment of gains in accuracy and efficacy within a consistent framework. By focusing on similar approaches, the research highlights gains in data pretreatment optimization and model evaluation, indicating the study's distinctive contributions to improving harmful website identification. Comparison of model performance measurements is listed in Table XI.

Table XI. Comparison of model performance measurements

Logistic Regression Model	Accuracy	Precision	Recall	F1-score
In (Alsaedi et al., 2022)	86.15%	82.21%	90.82%	86.30%
In (Utku & Can, 2022)	86.20%	90.60%	93.90%	92.20%
In (Shin et al., 2022)	90%	93.37%	92.29%	93.30%
Proposed method	92.60%	92.76%	92.73%	92.72%

From Table XI, it can be concluded that the Logistic Regression algorithm with the proposed method has the highest accuracy value. Although research (Shin et al., 2022) has higher precision and F1-score than the proposed method, but accuracy as the main evaluation has a lower value than the proposed method.

IV. CONCLUSION

This research applies the Logistic Regression algorithm to detect malicious websites with a focus on data preprocessing optimization and model evaluation. The data preprocessing process includes cleaning data from missing values, converting four class types (benign, defacement, phishing, and malware) into two class types (good and bad), creating 23 new features, balancing data with hybrid sampling techniques, and assigning numerical labels to classes. Model evaluation using accuracy, precision, recall, F1-score, and cross-validation metrics resulted in accuracy of 92.60%, 92.76%, 92.73%, 92.72%, and 92.71% respectively. This approach proves that Logistic Regression optimized through proper preprocessing and evaluation can be effective in detecting malicious websites so that it can contribute to the cybersecurity system to protect users from cyber threats.

The findings show that Logistic Regression may effectively detect dangerous websites when improved with correct preprocessing and evaluation approaches. This is a crucial contribution to the cybersecurity industry since it helps to safeguard consumers from cyber dangers. However, there are some constraints to consider. The dataset's sample size and diversity may restrict the results' generalizability. Furthermore, while Logistic Regression accurately identifies linear correlations, its performance may need to be improved by complicated patterns in the data.

Future research could examine different algorithms and their usefulness in detecting fraudulent websites. Real-time detection systems could improve practical applications, and user feedback mechanisms could help develop model accuracy and adaptation to emerging dangers.

REFERENCES

A. Saleem Raja, S. Peerbashab, Y. Mohammed Iqbal, B. Sundarvadivazhagan, & M. Mohamed Surputheen. (2023). Structural Analysis of URL For Malicious URL Detection Using Machine Learning. *Journal of Advanced Applied Scientific Research*, *5*(4), 28–41. https://doi.org/10.46947/joaasr542023679

Alban, A. Q., Islam, F., Malluhi, Q. M., & Jaoua, A. (2020). Anomalies Detection in Software by Conceptual Learning From Normal Executions. *IEEE Access*, *8*, 179845–179856. https://doi.org/10.1109/ACCESS.2020.3027508

Alobaid, A., Kacprzak, E., & Corcho, O. (2020). Typology-based semantic labeling of numeric tabular data. *Semantic Web*, *12*(1), 5–20. https://doi.org/10.3233/SW-200397

Alsaedi, M., Ghaleb, F., Saeed, F., Ahmad, J., & Alasli, M. (2022). Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning. *Sensors*, *22*(9), 3373. https://doi.org/10.3390/s22093373

Aprelia Windarni, V., Ferdita Nugraha, A., Tri Atmaja Ramadhani, S., Anisa Istiqomah, D., Mahananing Puri, F., & Setiawan, A. (2023). DETEKSI WEBSITE PHISHING MENGGUNAKAN TEKNIK FILTER PADA MODEL MACHINE LEARNING. In *Information System Journal (INFOS)* | (Vol. 6, Issue 1).

Barella, V. H., Garcia, L. P. F., de Souto, M. C. P., Lorena, A. C., & de Carvalho, A. C. P. L. F. (2021). Assessing the data complexity of imbalanced datasets. *Information Sciences*, *553*, 83–109. https://doi.org/10.1016/j.ins.2020.12.006

Bates, S., Hastie, T., & Tibshirani, R. (2023). Cross-Validation: What Does It Estimate and How Well Does It Do It? *Journal of the American Statistical Association*, 1–12. https://doi.org/10.1080/01621459.2023.2197686

Chung, C.-J., & Fabbri, A. G. (2020). Mineral Occurrence Target Mapping: A General Iterative Strategy in Prediction Modeling for Mineral Exploration. *Natural Resources Research*, *29*(1), 115–134. https://doi.org/10.1007/s11053-019-09494-5

Dai, C., Lin, B., Xing, X., & Liu, J. S. (2023). False Discovery Rate Control via Data Splitting. *Journal of the American Statistical Association*, *118*(544), 2503–2520. https://doi.org/10.1080/01621459.2022.2060113

Doshi, B. (2011). *Handling Missing Values in Data Mining*.

Felix, E. A., & Lee, S. P. (2019). Systematic literature review of preprocessing techniques for imbalanced data. *IET Software*, *13*(6), 479–496. https://doi.org/10.1049/iet-sen.2018.5193

Gao, X., Ren, B., Zhang, H., Sun, B., Li, J., Xu, J., He, Y., & Li, K. (2020). An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling. *Expert Systems with Applications*, *160*, 113660. https://doi.org/10.1016/j.eswa.2020.113660

Jalil, S., Usman, M., & Fong, A. (2023). Highly accurate phishing URL detection based on machine learning. *Journal of Ambient Intelligence and Humanized Computing*, *14*(7), 9233–9251. https://doi.org/10.1007/s12652-022-04426-3

Kavici, S., & Ayaz-Alkaya, S. (2024). Internet addiction, social anxiety and body mass index in adolescents: A predictive correlational design. *Children and Youth Services Review*, *160*, 107590. https://doi.org/10.1016/j.childyouth.2024.107590

Liu, H., Chen, S.-M., & Cocea, M. (2019). Subclass-based semi-random data partitioning for improving sample representativeness. *Information Sciences*, *478*, 208–221. https://doi.org/10.1016/j.ins.2018.11.002

Malicious URLs Dataset. (n.d.). Retrieved April 25, 2024, from https://bit.ly/4aigTxf

Mohamad Arifandy, & Septia Ulfa Sunaringtyas. (2021). Rancang Bangun Model Machine Learning untuk Mendeteksi Malicious Webpage dengan Metode Wang, et al. (2017). *Info Kripto*, *15*(2), 63–68. https://doi.org/10.56706/ik.v15i2.3

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

Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. *Computers & Se-*

curity, *136*, 103545. https://doi.org/10.1016/j.cose.2023.103545

Qian, Y., Liang, Y., Li, M., Feng, G., & Shi, X. (2014). A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing*, *143*, 57–67. https://doi.org/10.1016/j.neucom.2014.06.021

Ramadhan, E. (2023). *PENERAPAN LEXICAL FEATURES UNTUK MENGOPTIMASI ALGORITMA RANDOM FOREST DALAM PENDETEKSIAN MALICIOUS URL PADA WEBSITE*. UPN "Veteran" Yogyakarta.

Saleem Raja, A., Vinodini, R., & Kavitha, A. (2021). Lexical features based malicious URL detection using machine learning techniques. *Materials Today: Proceedings*, *47*, 163–166. https://doi.org/10.1016/j.matpr.2021.04.041

Shin, S.-S., Ji, S.-G., & Hong, S.-S. (2022). A Heterogeneous Machine Learning Ensemble Framework for Malicious Webpage Detection. *Applied Sciences*, *12*(23), 12070. https://doi.org/10.3390/app122312070

Tashev, I. J., Michael Winters, R., Wang, Y.-T., Johnston, D., Reyes, A., & Estepp, J. (2022). Modelling the Training Process. *2022 IEEE Research and Applications of Photonics in Defense Conference (RAPID)*, 1–2. https://doi.org/10.1109/RAPID54472.2022.9911274

Tiwari, S. R., & Rana, K. K. (2021). *Data Science and Intelligent Applications* (Vol. 52).

Utku, A., & Can, U. (2022). Machine Learning-Based Effective Malicious Web Page Detection. *International Journal of Information Security Science*, *11*(4), 28–39.

Vajrobol, V., Gupta, B. B., & Gaurav, A. (2024). Mutual information based logistic regression for phishing URL detection. *Cyber Security and Applications*, *2*, 100044. https://doi.org/10.1016/j.csa.2024.100044

Verma, V. K., Brahma, D., & Rai, P. (2020). Meta-Learning for Generalized Zero-Shot Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 6062–6069. https://doi.org/10.1609/aaai.v34i04.6069

Yacouby, R., & Axman, D. (2020). Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models. *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, 79–91. https://doi.org/10.18653/v1/2020.eval4nlp-1.9

Yu, L., Chen, L., Dong, J., Li, M., Liu, L., Zhao, B., & Zhang, C. (2020). Detecting Malicious Web Requests Using an Enhanced TextCNN. *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 768–777. https://doi.org/10.1109/COMPSAC48688.2020.0-167