OPEN ACCESS
VOL. 7, NO. 1, PP.127-137, NOVEMBER 2024

ISSN : 2622-8106 (ONLINE)
DOI: 10.20895/INISTA.V7I1.1588

Journal of Informatics, Information System, Software Engineering and Applications (INISTA)

Implementation of Random Forest Classification and Support Vector Machine Algorithms for Phishing Link Detection

Felliks Feiters Tampinongkol*[1], Ahya Radiatul Kamila[2], Ariq Cahya Wardhana[3], Adi Wahyu Candra Kusuma[4], Danny Revaldo[5]

[1,2,4,5] *Program Studi Data Science, Universitas Bunda Mulia*
Jl. Lodan Raya No. 2 Ancol North Jakarta, Indonesia

[3] *Program Studi Rekayasa Perangkat Lunak, Telkom University Purwokerto*
Jl. DI Panjaitan No. 128, South Purwokerto, Indonesia

[1] ftampinongkol@bundamulia.ac.id
[2] akamila@bundamulia.ac.id
[3] ariqcahya@telkomuniversity.ac.id
[4] akusuma@bundamulia.ac.id
[5] dannyrvldo@gmail.com

Received on 31-07-2024, revised on 03-12-2024, accepted on 14-12-2024

Abstract

This research compares two machine learning methods, Support Vector Machine (SVM) and Random Forest Classification (RFC), in detecting phishing links. Phishing is an attempt to obtain sensitive information by masquerading as a trustworthy entity in electronic communications. Detecting phishing links is crucial in protecting users from this cyber threat. In this study, we used a dataset consisting of features extracted from URLs, such as URL length, the use of special characters, and domain information. The dataset was then split into training and testing data with an 80:20 ratio. We trained the SVM and RFC models using the training data and evaluated their performance based on the testing data. The results show that both methods have their respective advantages. SVM, known for handling high-dimensional data well and providing optimal solutions for classification problems, demonstrated a high accuracy rate in detecting phishing links. However, SVM requires a longer training time compared to RFC. On the other hand, RFC, an ensemble method known for its resilience to overfitting, showed performance nearly comparable to SVM in terms of accuracy but with faster training time and better interpretability. This comparison indicates that RFC is more suitable for scenarios requiring quick results and easy interpretation, while SVM is more appropriate for situations where accuracy is critical, and computational resources are sufficient. In conclusion, the choice of phishing link detection method should be tailored to specific needs and available resource constraints. This research provides valuable insights for developing more effective, efficient, and relevant phishing detection systems.

Keywords: Features Engineering, Machine Learning, Link Phishing, Random Forest, SVM

Corresponding Author:
*Felliks Feiters Tampinongkol
Power Electronics and Renewable Energy Research Laboratory (PEAR-L), University of Malaya
Balai Cerap UTM, Lengkok Suria, 81310 Skudai, Johor, Malaysia
Email: saad@um.edu.my

I. INTRODUCTION

Cybercrime has become a major threat in the increasingly connected digital world. As society becomes more reliant on technology, the internet, and gadgets for communication, various types of cybercrime continue to rise [1]. Some of these include hacking, malware, ransomware, and phishing. Cybercriminals exploit vulnerabilities in security systems to steal victims' personal data, access sensitive information, and commit financial fraud [2]. The damage from these crimes affects not only individuals but also large institutions, causing significant economic losses and reputational damage. Hacking can lead to the leakage of confidential data and system damage, while malware and ransomware can cripple organizational operations and extort victims for ransom payments [3]. Phishing is one of the most common and dangerous methods of cybercrime. Phishing works by sending fake messages that appear to come from trusted sources, such as banks or online services, to trick victims into revealing sensitive information [4]. This technique often involves emails, text messages, or fake websites designed to resemble legitimate login pages. When victims enter personal information such as usernames, passwords, or credit card numbers, this data is sent directly to the attacker, who can then misuse it for malicious purposes. Phishing can also occur through phone calls or social media, where attackers pretend to be trustworthy parties to deceive victims [5]. This method continues to evolve, with cybercriminals using increasingly sophisticated social engineering techniques to improve the success of their attacks.

The negative impacts of phishing are extensive and detrimental. For individuals, phishing can lead to identity theft, financial loss, and privacy breaches [6]. Victims may lose access to important accounts, such as email or bank accounts, resulting in direct financial losses. Besides financial damage, victims also face stress and anxiety due to threats to their personal security [7]. For organizations, phishing attacks can cause confidential data leaks, operational disruptions, and high recovery costs. Additionally, the reputation of the affected organization can be damaged, leading to a loss of trust from customers and business partners [8]. A tarnished reputation can have long-term impacts, affecting business performance and relationships with stakeholders. Various manual efforts have been made to combat phishing, such as user education and the implementation of strict security policies. Users are taught to recognize signs of phishing, such as suspicious links or unusual requests for personal information [9]. Moreover, organizations implement security measures such as two-factor authentication and regular software updates [10]. However, these manual efforts are often not sufficiently effective, as attackers continuously develop new techniques that are increasingly difficult to detect. While user education is important, it has limitations in preventing highly sophisticated attacks. The use of advanced security software and continuous monitoring is necessary to protect systems and data from the increasingly complex threat of phishing [11].

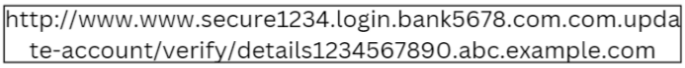
A machine learning approach can be a convincing solution for phishing link detection. Machine learning algorithms can analyze large amounts of data and identify patterns or characteristics indicative of phishing behavior [12]. By training a model on a dataset that includes examples of phishing and non-phishing links, the system can learn to distinguish between safe and dangerous links. This approach enables faster and more accurate detection and can adapt to evolving phishing techniques. Machine learning uses techniques such as pattern recognition and classification to predict the likelihood that a link is phishing based on certain features, such as URL, email content, and other metadata [13].

Two machine learning algorithms commonly used in phishing detection are Random Forest and Support Vector Machine (SVM) [14]. Random Forest works by constructing multiple decision trees from different subsets of data and combining their results to make a final decision [15]. This algorithm is known for its ability to handle complex and diverse data as well as its resistance to overfitting. Random Forest can manage various types of data and provide stable results despite variability in the data. Meanwhile, SVM seeks the optimal hyperplane that can separate data into different classes and is highly effective in high-dimensional spaces [16]. This algorithm is particularly useful in situations where there are clear boundaries between phishing and non-phishing data. SVM uses the kernel trick to map data into high-dimensional space, allowing for more effective separation. Both algorithms have their advantages and disadvantages, so a comparative analysis between Random Forest and SVM in phishing detection is important to determine the most effective algorithm in various scenarios.

II. RESEARCH METHOD

A. *Phishing Link*

Phishing attacks are cybercrimes where an individual's personal information is illegally accessed by attackers posing as a legitimate entity through websites or URLs. Phishing is carried out through messages and emails that direct victims to visit websites that appear genuine. Phishing can be categorized into several groups with different forms of crime.



http://www.www.secure1234.login.bank5678.com.com.update-account/verify/details1234567890.abc.example.com

Fig 1. *Link Phishing Example*

Some types of phishing that have increasingly victimized individuals include spear phishing, which targets specific individuals using personalized information; whaling, which primarily targets important individuals or groups within an organization; smishing, which involves sending text messages to victims; and vishing, where attackers conduct attacks through voice calls or voicemails impersonating a certain institution [17]. One characteristic that can deceive victims during a phishing attack is when a link or URL appears to be from an official institution, making it easy for victims to provide access and personal information to the attackers. The characteristics of phishing can be identified and processed into knowledge, as illustrated in Figure 1.

B. Research Stages

The research stages begin with the collection of links data confirmed as either phishing or legitimate from the website https://www.kaggle.com/, which is openly available for download. Pre-processing is conducted, encompassing several important steps: data cleansing to remove anomalies and duplicates, exploratory data analysis (EDA) to understand the characteristics of the data and feature distribution, and feature selection to choose relevant features based on feature calculations and analysis. After that, the data is split into training data and testing data for model training and evaluation purposes. The next stage is model classification, using two algorithms to build the classification model: Support Vector Machine (SVM) and Random Forest Classification (RFC).

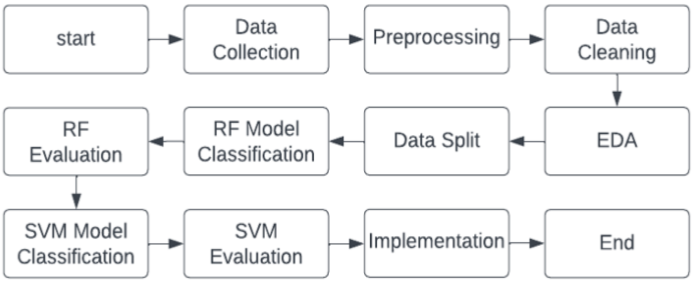


Fig 2. *Flowchart Diagram*

Support Vector Machine (SVM) is used to separate phishing links and legitimate links based on the features selected during the pre-processing stage, while Random Forest is employed for classification using an ensemble learning approach that utilizes multiple decision trees to predict phishing and legitimate links. Evaluation is performed to analyze and assess the performance of the developed models using evaluation metrics such as accuracy, precision, recall, and F1-score. The final stage is implementation, where the evaluated models are deployed into a phishing detection system that can be used in real-world environments. The research flowchart can be seen in Figure 2.

C. Pre-processing

The pre-processing process is carried out to convert raw datasets into a clean and ready-to-use dataset for data modeling [18-20]. The pre-processing steps vary depending on the data and the context of the model being developed. Based on the dataset available, distinguishing whether a link is phishing or legitimate can be identified through several indicators or features. One of the most common indicators is the URL length. In this study, there are 9 indicators or features used for the Machine Learning model, which include URL length, Hostname length, presence of ‘www’, presence of ‘.com’, URL digit ratio, number of digits in each URL, number of dot (.) characters, number of slash (/) characters, and Hostname digit ratio. The pre-processing steps can be carried out as follows:

D. Random Forest Classification Model

The Random Forest algorithm is one of the machine learning algorithms that uses a combination of several individual models to produce predictions in classification and regression tasks. This algorithm works by creating multiple decision trees that start with taking random samples from the original dataset to create several subsets of data through a technique known as bootstrap sampling [21]. Each subset is used to train one decision tree, and at each node of the tree, only a number of randomly selected features are considered for splitting, known as the random feature selection technique. One of the main techniques used in Random Forest is Bagging, where multiple decision trees are built based on data samples taken randomly with replacement from the original dataset. This means some samples may appear more than once, while others may not appear at all [22]. Each of these decision trees has a different feature selection policy and a unique structure based on the subset of data used. After the separate decision trees are built, each tree is used to make predictions on the test data, and each tree votes for a particular class.

The class with the most votes is chosen as the final prediction, making this majority voting process more robust to data variation and more resistant to overfitting compared to a single decision tree. Additionally, Random Forest can handle data with a large number of features and provide an estimate of the importance of each feature in the classification process, thereby offering additional insights into the influence of each feature on the prediction.

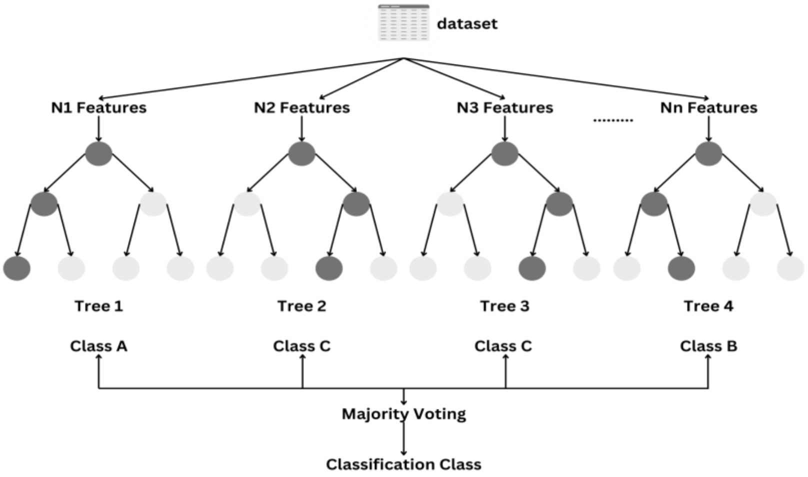


Fig 3. *Random Forest Classification Model*

Random Forest is an algorithm that can be used for supervised learning classification in the detection of phishing and legal links based on a number of relevant features. Features such as URL length, Hostname length, presence of 'www', presence of '.com', URL digit ratio, number of digits in each URL, number of dot characters (.), number of slash characters (/), and Hostname digit ratio can provide important information to distinguish between suspicious (phishing) and legal links. In phishing detection, the random feature selection process at each node in the decision tree allows the model to consider the influence of various features without overly relying on a single feature. After all the decision trees make predictions, the final prediction is determined by majority vote, leveraging the collective decision-making of the ensemble to improve accuracy and robustness.

Random Forest uses majority voting to determine the final prediction, resulting in a stable and robust model against variations in the data. This is particularly important in the context of phishing detection, as the data often has a large and diverse number of features. This allows the model to effectively distinguish between phishing and legal links based on relevant features.

E. Support Vector Machine Model

Support Vector Machine (SVM) is a widely used machine learning algorithm for classification and regression tasks. SVM works by finding the optimal hyperplane that can separate different classes in a dataset, with the main objective of maximizing the distance between the hyperplane and the nearest data points from each class, thereby enhancing stability in predicting new data [23]. SVM also uses kernel methods to transform data into higher dimensions, enabling it to handle non-linear datasets without explicitly performing dimensional transformations [24]. This capability allows SVM to easily determine class boundaries in higher-dimensional spaces. Commonly used kernels include linear, polynomial, and radial basis function (RBF) kernels, each suitable for different characteristics of datasets.

$$min\frac{1}{2}||\omega||^2 \text{ with the constraint: } y_i(wx_i + b) \geq 1, i = 1, \dots, \lambda \qquad (1)$$

During the training process, SVM finds the hyperplane that can maximize the margin between classes in an N-dimensional space (N being the number of objects) [25]. When making predictions on new data, SVM uses the relative position of the data to the determined hyperplane to classify it accurately [26]. This approach makes SVM highly effective in handling complex classification tasks with datasets that have many features. The optimal hyperplane can be determined using equation (1).

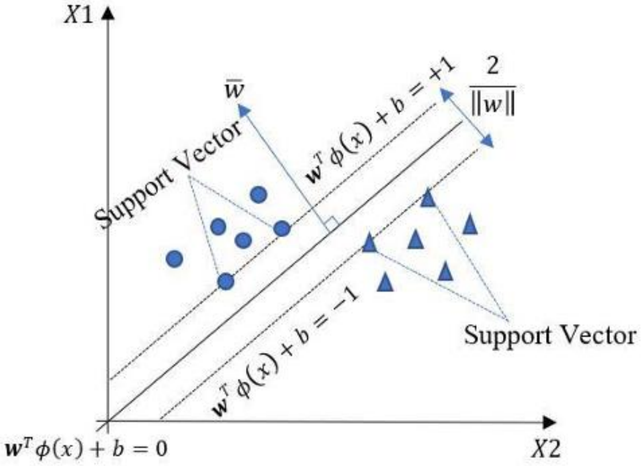

$X1$
$\bar{w}$
Support Vector
$\mathbf{w}^T\phi(x) + b = +1$
$\frac{2}{\|w\|}$
$\mathbf{w}^T\phi(x) + b = -1$
Support Vector
$\mathbf{w}^T\phi(x) + b = 0$
$X2$

Fig 4. *Support Vector Machine*

With x_i representing the input data and y_i representing the output of x_i, and ω as the classification parameter [27-29]. The main advantage of SVM is its ability to reduce the risk of overfitting by focusing on maximizing the margin between classes during optimization, rather than merely on complex correlations between the data. Support Vector Machine (SVM) is an algorithm that can be used in supervised learning for detecting phishing and legal links based on a number of relevant features. Features such as URL length, Hostname length, presence of 'www', presence of '.com', URL digit ratio, number of digits in each URL, number of dot characters (.), number of slash characters (/), and Hostname digit ratio are used to distinguish between suspicious (phishing) and legal links. SVM utilizes kernel techniques to transform data into higher dimensions, enabling clear separation between categories based on these complex features.

During the training process, SVM selects the optimal hyperplane to maximize the margin between categories, thereby improving stability and prediction accuracy on new data. SVM uses the relative position of test data to the determined hyperplane to classify links into appropriate categories, ensuring the model's ability to recognize phishing and legal links based on significant feature characteristics.

F. Evaluation Model

The concluding stage of the research focuses on assessing the performance of the classification model. A widely utilized approach for this evaluation is the confusion matrix [30], which provides a detailed summary of prediction outcomes. Specifically, True Positive (TP) denotes the number of instances correctly classified as positive, False Positive (FP) represents negative instances erroneously predicted as positive, False Negative (FN) refers to positive instances misclassified as negative, and True Negative (TN) indicates the number of instances correctly identified as negative.

This evaluation is conducted by calculating several metrics based on specific formulas to provide an overview of performance:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (5)$$

Accuracy is the proportion of correct predictions compared to the total predictions (Eq 2), Precision is the proportion of correct phishing predictions compared to all phishing predictions (Eq 3), Recall is the proportion of correctly identified phishing links compared to all actual phishing links (Eq 4) and F1-Score is the harmonic mean of precision and recall. Afterwards, compare the evaluation matrices of both models (SVM and Random Forest) to determine which model performs better in classifying phishing or legitimate links (Eq 5).

III. RESULTS AND DISCUSSION

A. *Exploratory Data Analyst (EDA)*

The results of the Exploratory Data Analysis (EDA) in Figure 5 show the analysis of several features in the URL dataset used to identify the characteristics and distribution of each feature.

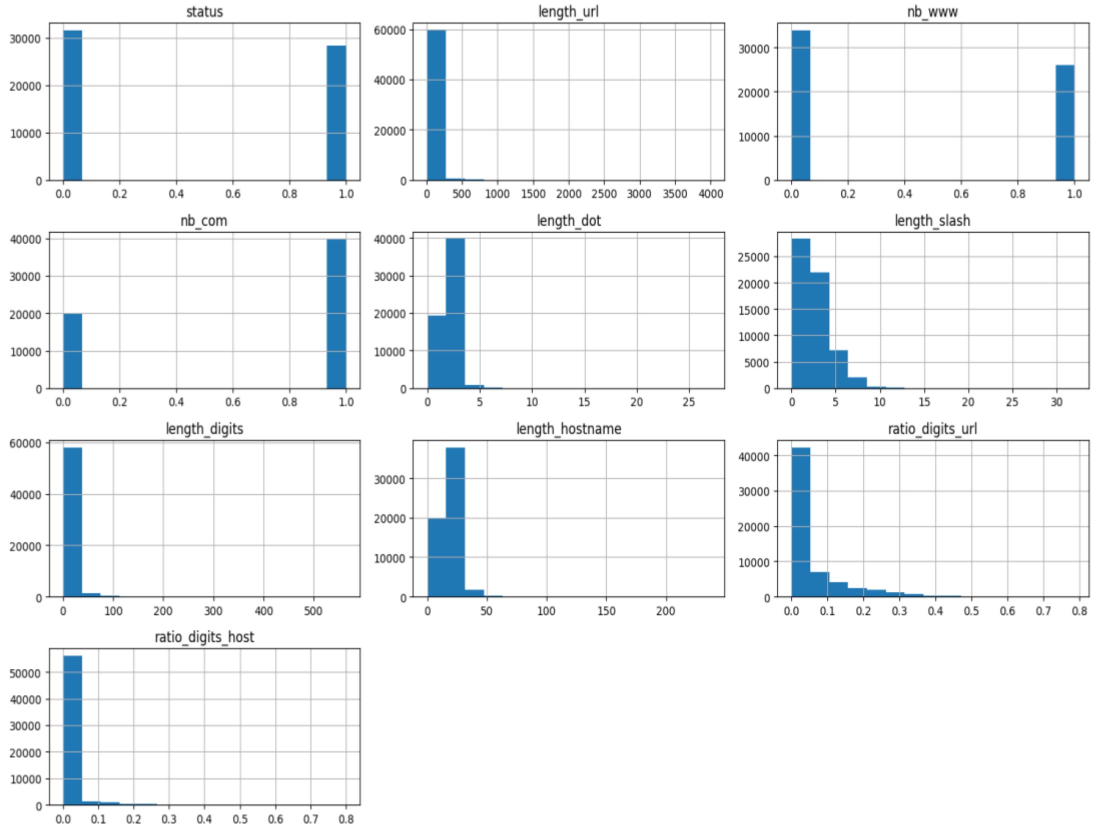


Fig 5. *Histogram Features Distribution*

1. *Distribution Histogram of All Features*

The histogram of the status shows the distribution of URL statuses, with 0 indicating safe URLs and 1 indicating dangerous URLs. It can be seen that the dataset is fairly balanced between safe and dangerous URLs, although dangerous URLs are fewer than safe ones. The URL length histogram shows that the majority of URLs are shorter than 500 characters, with a small portion of URLs exceeding 1000 characters in length. The distribution of the number of "www" in URLs is very similar to the distribution of statuses, showing that most URLs do not contain "www," while a portion of them do contain "www." The histogram of the number of ".com" in URLs shows that most URLs contain ".com," which is one of the most commonly used top-level domains. The distribution of the number of dots (.) in URLs shows that the majority of URLs have fewer than 5 dots, with the distribution decreasing as the number of dots increases.

The distribution of the number of slashes (/) in URLs shows that most URLs have fewer than 10 slashes, with some URLs having more than 20 slashes. The histogram of the number of digits in URLs shows that most URLs have fewer than 100 digits, with some URLs having a very high number of digits. The distribution of the hostname length in URLs shows that the majority of hostnames are shorter than 50 characters, with a small portion being longer than 100 characters.

2. *Bar Chart*

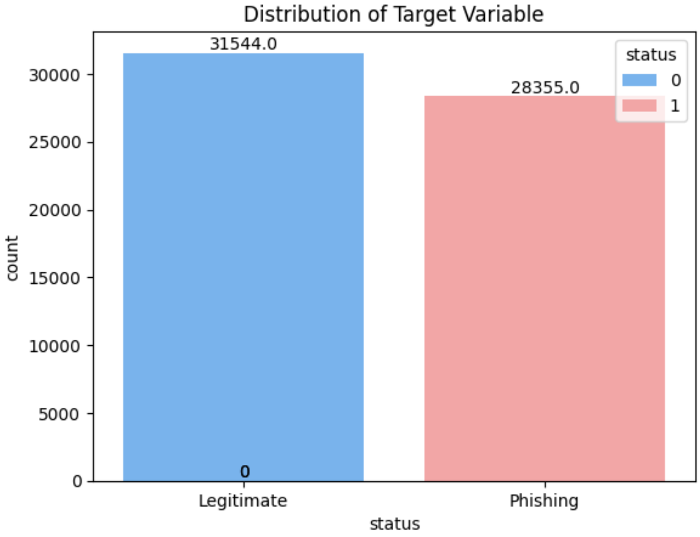


Fig 6. *Distribution of Target Variable*

The histogram of the digit ratio in URLs shows that most URLs have a low digit ratio, with some URLs having a fairly high digit ratio, indicating that these URLs may contain many numbers. The distribution of the digit ratio in hostnames shows that most hostnames have a low digit ratio, with some hostnames having a fairly high digit ratio. Figure 6 shows the distribution of the target variable in the dataset, distinguishing between legitimate and phishing URLs.

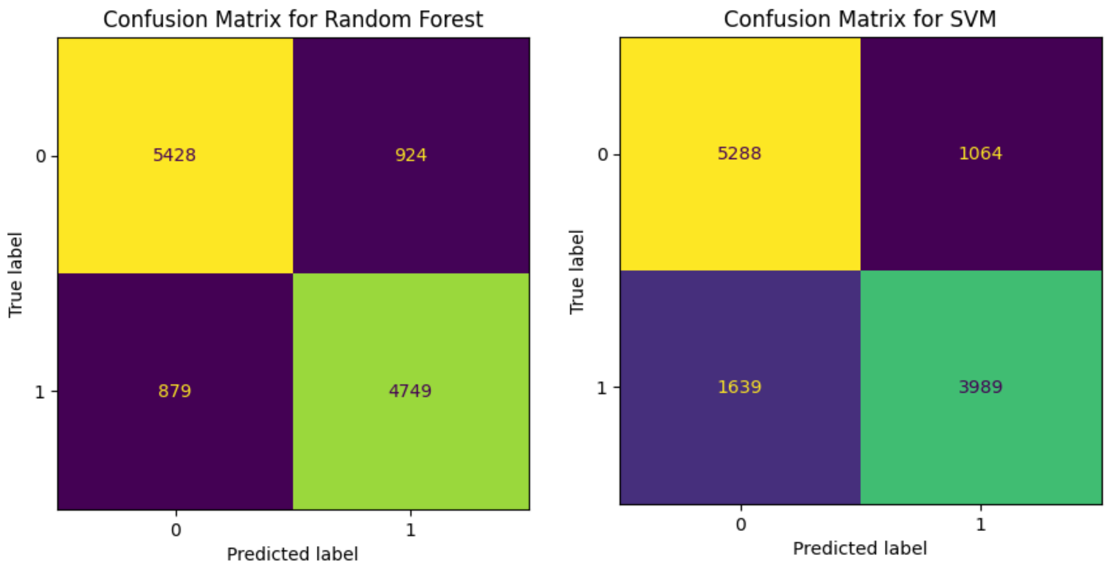
The 'Legitimate' category is represented by status 0 and marked in blue, while the 'Phishing' category is represented by status 1 and marked in pink. From the Bar Chart in Figure 6, it can be observed that the number of legitimate URLs is 31.544, slightly higher than the number of phishing URLs, which is 28.355. Despite the difference in numbers between the two categories, the distribution of this target variable is relatively balanced. This balance is important in the context of machine learning because a model trained on a balanced dataset tends to perform better and be more generalizable in classifying new URLs as legitimate or phishing.

B. *Classification Model*

1. *Random Forest Classification*

The Random Forest algorithm was implemented using the Python library provided by scikit-learn. During the modeling process, the algorithm was configured with its default parameter settings, resulting in the construction of a Random Forest comprising 100 decision trees. The performance of the model, trained to classify URLs as either legitimate or phishing, is depicted in Figure 7 through a Confusion Matrix. This matrix serves as a tool to evaluate the classification model by comparing its predicted outcomes with the actual labels. The Confusion Matrix includes four key components: True Positives (TP) totaling 4749, True Negatives (TN) amounting to 5428, False Positives (FP) recorded as 924, and False Negatives (FN) counted as 879 data.

From this matrix, several important evaluation metrics can be calculated: Accuracy, which is the proportion of all correct predictions, is approximately 87%. Precision, the proportion of correct positive predictions out of all positive predictions, is around 84%. Recall (Sensitivity), the proportion of correct positive predictions out of all actual positive cases, is also around 84%. F1-Score, which is the harmonic mean of precision and recall, is around 84%. These metrics provide insights into the performance of the Random Forest model, indicating that it performs quite well in classifying URLs as legitimate or phishing, with accuracy, precision, recall, and F1-Score all around 84%. This Confusion Matrix helps in understanding the strengths and weaknesses of the model and gives a clear picture of how well the model can identify phishing URLs compared to legitimate URLs.



a) *Random Forest Confusion Matrix*

b) SVM *Confusion Matrix*

Fig 7. *Confusion Matrix from Two Algorithm*

2. *Support Vector Machine (SVM)*

The Support Vector Machine (SVM) algorithm also uses the same Python library as Random Forests, namely scikit-learn. During the modeling stage, the parameters used to train the SVM were left unchanged, retaining their default values with C=1.0 and Gamma='scale'. Figure 7 shows the Confusion Matrix of the SVM model used to classify URLs as legitimate or phishing. This Confusion Matrix presents the evaluation results of the model's performance by comparing the model's predictions with the actual values from the test data. This matrix consists of four elements: True Positive (TP) of 3989, True Negative (TN) of 5288, False Positive (FP) of 1064, and False Negative (FN) of 1639.

Table 1. Comparison of Evaluation Model

	Random Forest	Support Vector Machine
Precision:	0.837123	0.789432
Recall:	0.843866	0.708777
F1 Score:	0.840456	0.746934
Accuracy:	0.849499	0.774290

From this matrix, several important evaluation metrics can be calculated: Accuracy, which is the proportion of all correct predictions, is approximately 77%. Precision, the proportion of correct positive predictions out of all positive predictions, is around 78.9%. Recall (Sensitivity), the proportion of correct positive predictions out of all actual positive cases, is about 71%. F1-Score, which is the harmonic mean of precision and recall, is around 74.7%.

These metrics provide insights that the SVM model has lower performance in predicting URLs as legitimate or phishing, with an accuracy of 77%. However, this model shows weaknesses in detecting phishing URLs, as evidenced by the lower recall value of 70.9%, indicating that the model tends to misclassify phishing URLs as legitimate (False Negative). Conversely, the number of False Positives indicates that the model also tends to misclassify legitimate (true) URLs as phishing (false). The Confusion Matrix in Figure 7 gives a clear picture of how the SVM model works in URL classification and helps in understanding areas where the model can be improved, particularly in reducing the number of False Negatives to increase recall.

C. Evaluation Model

The evaluation results for the Random Forest and Support Vector Machine (SVM) models using cross-validation from the Python scikit-learn library are summarized in Table 2, which shows a comparison of the performance of both models over five iterations:

- Iteration 1: Random Forest achieved an accuracy of 0.85, while SVM had an accuracy of 0.77, with a difference of approximately 0.073.
- Iteration 2: Random Forest again outperformed SVM with an accuracy of 0.85 compared to SVM's 0.77, a difference of about 0.077.
- Iteration 3: Both models showed a performance drop, with Random Forest at 0.844 and SVM at 0.77, but Random Forest still had a better performance with a difference of around 0.072.
- Iteration 4: Random Forest improved slightly with an accuracy of 0.85, while SVM also saw a minor increase to 0.77, maintaining a performance gap of approximately 0.071.
- Iteration 5: Random Forest achieved an accuracy of 0.84, while SVM had an accuracy of 0.77, with a difference of around 0.073.

Table 2. Comparison of 5-Fold Cross Validation

Iteration	*Random Forest*	*SVM*
1	0.851085	0.778130
2	0.851836	0.774290
3	0.844240	0.771786
4	0.847746	0.777045
5	0.844728	0.772185

Overall, Random Forest consistently demonstrated better performance compared to SVM in each cross-validation iteration. The performance values for Random Forest ranged between 0.844 and 0.852, indicating good stability, whereas the performance for SVM ranged between 0.772 and 0.778.

IV. CONCLUSION

The analysis of the performance of Random Forest and Support Vector Machine (SVM) models in detecting phishing links shows significant and promising results worthy of further exploration. The balanced distribution of the target variable, with 31,544 legitimate links and 28,355 phishing links, is crucial to ensure that the model does not become biased in its classification. The Exploratory Data Analysis (EDA) provided in-depth insights into various features used in detection. Features such as URL length, number of "www", hostname length, number of dots in the URL, and digit proportions in the URL show significant differences in distribution between legitimate and phishing links. The evaluation using 5-fold cross-validation provided a clearer picture of the consistency of both models' performance. Random Forest demonstrated more consistent performance compared to SVM, with higher accuracy across five iterations, averaging around 84%. In contrast, SVM's accuracy averaged around 77%, indicating lower and less stable performance compared to Random Forest. The stability and reliability of Random Forest make it a more suitable choice for detecting phishing links in the context of the analyzed dataset. Its consistent performance across various evaluation iterations provides greater confidence in its application for real-world scenarios.

ACKNOWLEDGMENT

The authors would like to express my sincere gratitude to Universitas Bunda Mulia for providing the invaluable resources and support necessary for this research. Special thanks to my research partner for their dedication, expertise, and collaborative spirit throughout the project.

REFERENCES

[1] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, 2021, doi: 10.1007/s11235-020-00733-2.

[2] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," *Front. Comput. Sci.*, vol. 3, no. March, pp. 1–23, 2021, doi: 10.3389/fcomp.2021.563060.

[3] N. Mtukushe, A. K. Onaolapo, A. Aluko, and D. G. Dorrell, "Review of Cyberattack Implementation, Detection, and Mitigation Methods in Cyber-Physical Systems," *Energies*, vol. 16, no. 13, pp. 1–25, 2023, doi: 10.3390/en16135206.

[4] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," *IEEE Access*, vol. 11, no. February, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[5] B. Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji, and J. Porras, "Mitigation strategies against the phishing attacks: A systematic literature review," *Comput. Secur.*, vol. 132, p. 103387, 2023, doi: 10.1016/j.cose.2023.103387.

[6] M. F. Ansari, P. K. Sharma, and B. Dash, "Prevention of Phishing Attacks Using AI-Based Cybersecurity Awareness Training," *Int. J. Smart Sens. Adhoc Network.*, no. July, pp. 61–72, 2022, doi: 10.47893/ijssan.2022.1221.

[7] R. Alabdan, "Phishing attacks survey: Types, vectors, and technical approaches," *Futur. Internet*, vol. 12, no. 10, pp. 1–39, 2020, doi: 10.3390/fi12100168.

[8] S. Hawa Apandi, J. Sallim, and R. Mohd Sidek, "Types of anti-phishing solutions for phishing attack," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 769, no. 1, 2020, doi: 10.1088/1757-899X/769/1/012072.

[9] R. Alazaidah *et al.*, "Website Phishing Detection Using Machine Learning Techniques," *J. Stat. Appl. Probab.*, vol. 13, no. 1, pp. 119–129, 2024, doi: 10.18576/jsap/130108.

[10] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics," *Expert Syst. Appl.*, vol. 236, no. August 2023, p. 121183, 2024, doi: 10.1016/j.eswa.2023.121183.

[11] M. S. Akhtar and T. Feng, "Comparison of Classification Model for the Detection of Cyber-attack using Ensemble Learning Models," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 9, no. 5, pp. 1–11, 2022, doi: 10.4108/eai.1-2-2022.173293.

[12] T. O. Ojewumi, G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. Ogbu, "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," *Sci. African*, vol. 16, p. e01165, 2022, doi: 10.1016/j.sciaf.2022.e01165.

[13] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing website detection based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 21, no. 24, pp. 1–18, 2021, doi: 10.3390/s21248281.

[14] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electron.*, vol. 9, no. 9, pp. 1–24, 2020, doi: 10.3390/electronics9091514.

[15] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Appl. Sci.*, vol. 13, no. 8, 2023, doi: 10.3390/app13084649.

[16] A. Ferdita Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *J. Infomedia*, vol. 7, no. 1, p. 39, 2022, doi: 10.30811/jim.v7i1.2959.

[17] S. Wajiha Zahra, S. Riaz, and A. Arshad, "Phishing Attack, Its Detections and Prevention Techniques," *Int. J. Wirel. Inf. Networks*, vol. 1, no. 2, pp. 13–25, 2023, doi: 10.37591/ijwsn.

[18] Wijaya, Deny Setiawan; Widyaningrum, Destriana. Komparasi Metode Algoritma Klasifikasi pada Aanalisis Sentimen Komentar Cyberbullying di Instagram. Jurnal Tekinkom (Teknik Informasi dan Komputer), 2024, 7.1.

[19] E. F. Morales and H. J. Escalante, "A brief introduction to supervised, unsupervised, and reinforcement learning," *Biosignal Process. Classif. Using Comput. Learn. Intell.*, pp. 111–129, 2022, doi: 10.1016/b978-0-12-820125-1.00017-8.

[20] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," *Adv. Intell. Syst. Comput.*, vol. 937, pp. 99–111, 2019, doi: 10.1007/978-981-13-7403-6_11.

[21] D. R. Hermawan, M. Fahrio Ghanial Fatihah, L. Kurniawati, and A. Helen, "Comparative Study of J48 Decision Tree Classification Algorithm, Random Tree, and Random Forest on In-Vehicle CouponRecommendation Data," *2021 Int. Conf. Artif. Intell. Big Data Anal.*, 2021, doi: 10.1109/icaibda53487.2021.9689701.

[22] Y. Elgimati, “Weighted Bagging in Decision Trees: Data Mining,” *JINAV J. Inf. Vis.*, vol. 1, no. 1, pp. 1–14, 2020, doi: 10.35877/454ri.jinav149.

[23] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” *Neurocomputing*, vol. 408, no. 1, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.

[24] D. A. Pisner and D. M. Schnyer, “Support vector machine,” *Mach. Learn.*, pp. 101–121, 2020, doi: 10.1016/b978-0-12-815739-8.00006-7.

[25] Dalimunthe, Muhammad Variansjah. Sentimen Analisis Mengenai Polusi Udara Menggunakan Algoritma Support Vector Machine dan Random Forest. 2024. PhD Thesis. Universitas Mercu Buana Jakarta.

[26] S. Islam, *Data Classification And Incremental Clustering In Data Mining And Machine Learning.* Springer Nature PP - S.L., 2022.

[27] Thenata, Angelina Pramana. Text Mining Literature Review on Indonesian Social Media. JEPIN (Jurnal Edukasi dan Penelitian Informatika), 2021, 7.2: 226-232.

[28] Tampinongkol F, Herdiyeni Y, Herliyana E. Feature Extraction of Jabon (Anthocephalus sp) Leaf Disease using Discrete Wavelet Transform. 2020. TELKOMNIKA (Telecommunication Computing Electronics and Control) 18 (2), 740-751.

[29] Herdian, C., Kamila, A., Tampinongkol, F. F., Kembau, A. S., & Budidarma, I. G. A. M.. “One-hot encoding feature engineering untuk label-based data studi kasus prediksi harga mobil bekas”. 2024. Informasi Interaktif : Jurnal Informatika Dan Teknologi Informasi, 9(1), 10–16. https://doi.org/10.37159/jii.v9i1.41

[30] N. F. Abedin, R. Bawm, T. Sarwar, M. Saifuddin, M. A. Rahman, and S. Hossain, “Phishing Attack Detection using Machine Learning Classification Techniques,” *IEEE Xplore*. pp. 1125–1130, 2020. doi: 10.1109/ICISS49785.2020.9315895.