EMERGING ADVANCES IN INTEGRATED TECHNOLOGY
e-ISSN: 2773-5540

Vol. 6 No. 1 (2025) 9-18
https://publisher.uthm.edu.my/ojs/index.php/emait

EmAIT

# Performance Comparison of Machine Learning Models for Phishing Website Detection Based on Multilayer Perceptron

**Hui Ching Mah[1]*, Nor Hazlyna Harun[2]**

[1] *School of Computing,University Utara Malaysia (UUM), Sintok, Kedah, MALAYSIA*
[2] *Data Science Research Lab (DSRL), Institute for Advanced and Smart Digital Opportunities (IASDO), University Utara Malaysia (UUM), Sintok, Kedah, MALAYSIA*

*Corresponding Author: hazlyna@uum.edu.my

DOI: https://doi.org/10.30880/emait.2025.06.01.002

### Article Info

Received: 29 December 2024
Accepted: 30 May 2025
Available online: 10 June 2025

### Keywords

Comparison, multilayer perceptron, random forest, phishing detection, KNN, naive bayes, SVM, cybersecurity

### Abstract

Phishing is a common cybercriminal activity in which attackers lure people into providing data by posing as genuine websites. Currently, alerts and blacklists are well-used methods of detection though they have been proven less effective in the evolving types of phishing. This paper focuses on the impact of a Multilayer Perceptron (MLP) in minimizing the deficiencies of conventional techniques as a tool of identifying phishing websites. The goal is to improve the identification of new and initially unseen phishing sites by building upon MLP capability to resolve multiple relationships between webpage characteristics including URL, HTML, and HTTP properties. The given experiment is conducted on a total 500 phishing and 500 legitimate websites through different machine learning classifiers such as SVM, k-NN, Decision Trees, Naïve Bayes, and MLP using both 5 and 10-fold cross-validation. The performance of the models is measured using commonly used measures these include accuracy, precision, recall, and F1-score with the MLP having the best performance with 98.1% accuracy. This analysis shows that MLP is the best in improving the detection of phishing threats with the best scalability and adaptability in fighting phishing attacks. Hence, this paper reveals that MLP has a great capacity to enhance real-time phishing detection and minimize false alarm rates to afford a reliable defense against one of the constantly evolving cyber threats.

## 1. Introduction

Phishing is one of the most common types of cybercrime that engages users in sharing personal information or installing unauthorised software [1]. Phishing attacks are a dangerous type of cyber attack where malicious agents create a clone of legitimate websites to steal sensitive information [2]. However, cybersecurity, phishing and intrusion are interconnected because phishing and intrusion are risks that cybersecurity seeks to minimize. Phishing typically precedes other attack methods, in which a person is defrauded into providing information. This stolen information can lead to intrusion where the attackers gain access to one system without permission. They are linked through cybersecurity, which equips them with means through which they can identify, prevent, and combat both phishing attempts and intrusions, to safeguard systems and data from deterioration.

Unfortunately, current detection techniques struggle to adapt to changes in these attacks' behaviour, making real-time methods like blacklisting ineffective [3]. Have the statistic prove that even the current solutions cannot effectively to address this problem. As there are more phishing incidents are now being perpetrated, owing to the

APWG data made in 2022, of about 4.7 million phishing attacks, which were 1.5 times higher than those in 2019 [4,5]. Phishing attacks for instance, raise from 180.4 million in the third quarter of the year 2023 to 493.2 million, a rise of 173% in the same quarter only [6]. Therefore, this project plan to address using the multilayer perceptron system which is accurate and adaptive nature of the system to detect phishing websites, which the traditional approach of blacklisting is not capable of handling efficiently.

The aim is to increase the detectability of new, previously undisclosed phishing sites by applying the multilayer perceptron capacity to model non-linear, complex relationships extracted out of website attributes like URL, HTML, and HTTP. This is particularly important as phishing has continued to threaten many sectors and become a constant form of threat [7]. The capability of neural networks in multi-dimensional feature learning and high precision makes multilayer perceptron most suitable for refining current augmented detection rates and minimizing false positives [8]. There will be an emphasis on quantifiable enhancements, where the standard of measurement will include accuracy, precision, recall, and F1-score in relation to the multilayer perceptron model, compared to k-Nearest Neighbours (k-NN), Decision Tree, Random Forest, Naive Bayes, SVM, and Random Forest. This problem is highly important to modern anti-phishing endeavours, as phishing remains one of the most popular attack vectors, with MLP's potential to learn new patterns providing a scalable and viable safeguard.

The rest of the paper is as follows: Section 2 provides background to the study by reviewing existing literature on the use of machine learning algorithms in detecting phishing. It introduces and summarizes the difficulties and previous techniques laid on this basis. Section 3 applied following classifiers for the training and validation of data: SVM, k-NN, Decision Trees, Naive Bayes, SGD, MLP, and used both 5 folds cross validation and 10 fold cross validation. This provides an overview of the experimental setup where the selection of tools and frameworks are described, as well as data preprocessing techniques employed, and the performance measures for accuracy, precision, recall and F1-score matrices. Section 4 show the efficacy of each model, where the MLP produced a top result of 98.1% being the highest among all models, followed by Random Forest, SVM, Decision Tree, Naïve Bayes and k-NN. They also identify the advantages and the drawbacks of each classifier which was the focus for the analysis.

## 2. Related Works

Multilayer Perceptron (MLP) (Fig 1) has been widely used in cybersecurity with its ability to solve various tasks including the analysis of phishing and intrusion. The present literature review as tabulated in Table 1 aims to present the current research comparing MLP to other ML algorithms within a cybersecurity setting.

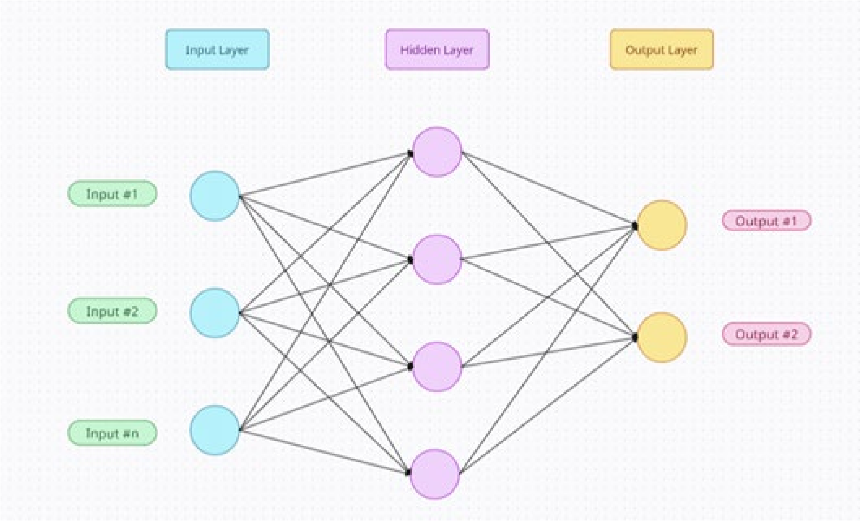


**Fig. 1** *Architecture neural network*

### 2.1 Phishing Detection

The MLP was used to classify the patterns for a specific study about phishing websites and the results revealed higher efficiency of the MLP as compared to other models. The research revealed MLP to be accurate of 98.4%, which is even better than conventional classifiers such as the SVMs and Decision Tree [9]. The authors emphasized that one of the biggest strengths of MLP is its ability to deal with noisy data, which is a characteristic frequently found in situations where phishing is detected. Further, the study preserved the aspect of feature selection approaches and highlighted that feature exclusion usually improves the model performance assessment [10].

## 2.2 Intrusion Detection System (IDS)

A comparison of MLP with other machine learning approaches has now been made when performed for intrusion detection. A detailed examination of the findings presented here by the authors also showed that although traditional methods such as decision tree achieved very high levels of accuracy (up to 99.67% for identification of Distributed Denial of Service attacks), MLP stands out as one of the most consistent classifiers with slightly low speed is due its backpropagation training algorithm [11]. Another work showed that the MLP could be used for the classification of the traffic indicating potential to differentiate between normal and malicious in current datasets [12].

**Table 1** *Literature review matrix*

| Ref | Aim | Datasets | Model/Techniques | Result |
|---|---|---|---|---|
| [13] | To construct a rule-based model for identifying and preventing internet banking Web sites from phishing based on webpage content and URL features. | 3066 phishing, 1271 legitimate instances collected. | - Support Vector Machine (SVM) | - True positive rate: 99.14%.<br>- False negative rate: 0.86%.<br>- Outperformed existing detection systems in accuracy and reliability. |
| [14] | To develop a system for detecting phishing links using machine learning to give enhance the safety on the web. | From PhishTank and Yandex Search API | Decision Tree, Adaboost, kNN, Random Forest, SVM, Naive Bayes | Decision Tree with NLP features achieved 97.02% accuracy. |
| [15] | To examine how the use of phishing feature datasets can improve the recognition of phishing attacks. | Clean URLs from Alexa | SVM, Naïve Bayes | - Accuracy: 99.96% True Positives and True Negatives.<br>- False Positive Rate: 0.04%.<br>- Runtime: Less than 2,000 ms. |
| [16] | To develop a cloud-based intrusion detection model based on machine learning, with emphasis on the anomaly detection algorithm employing the random forest (RF) classifier. | NSL-KDD, Bot-IoT | Random Forest | Accuracy: 98.3% on Bot-IoT, 99.99% on NSL-KDD |

# 3. Methodology

## 3.1 Data Collection and Preprocessing Techniques

The data set for the phishing attack detection system is obtained from GitHub from the repository phishing_dataset/phishing_dataset_test.csv at main.sibelkapan/phishing_dataset.GitHub [17]. These features are important for analyzing this dataset in an effort to detecting phishing attempts. Thus, 500 phishing websites are downloaded from PhishTank and equal number of 500 legitimate websites are obtained from Alexa. This makes the system's construction accurate since it provides a balanced training and testing data set. Data divides train set and test set are 70% and 30%, respectively so the amount of data the model is trained on is enough but the model will also be trained and tested on different independent instances.

Each feature can thus be said to be having a classification value as a function of being an indicator. The features are categorized into different groups, as outlined in Table 2. This section defines the processing that has been conducted for the purpose of introducing the dataset for machine learning models for the identification of phishing websites. First, an explorative data cleaning process was performed such that the URL in the dataset were checked and made unique such that there are no duplicates and each URL was confirmed to be properly formatted and reachable. To standardize the input, the entries that were missing any of the 25 numerical features (Table 2) were also removed for the sake of data accuracy.

Preprocessing methods such as Min-Max scaling and Z-score normalization were used in the dataset to reduce the impact of extreme values on the model outputs. The feature extraction was done by gathering 25 numerical values from different sources of web pages and the HTTP connections such as the URL addresses, the values found in the HTML codes, and the HTTP response codes. URL features included the length of the URL, counts of special characters, the use of HTTPS, and the presence of IP addresses. HTML characteristics derived from the visited web pages included the number of links present, occurrences of critical keywords such as "login" and "update," and information from meta tags. HTTP primitive characteristics were extrapolated based on server characteristics, including the HTTP status code, HTTP response time, and redirection status.

In the textual data preprocessing, the following processes were done conversion of textual data into tokenized data by breaking them into smaller parts, and the removal of stopwords that add little to the texts' meaning. Besides, the categorical data were transformed to a format suitable to the machine learning algorithms as either one-hot encoded or labelled encoded data. The final dataset to be used was obtained from the cleaned data where 70% was dedicated to training purposes while 30% was used for testing purposes. Lastly, to compile the preprocessing did not introduce new issues, various tests and checks were carried out in addition to exploratory data analysis to analyze the features and their distributions if any.

**Table 2** *Features used in phishing attack detection*

| No | Feature | Feature Group | Description |
|---|---|---|---|
| 1 | domain_similarity | URL | Measures the similarity of the domain name of the visited web site and the URL domain name obtained through Alexa or PhishTank. |
| 2 | url_lenght | URL | The total count of characters that exist in a URL. |
| 3 | http_protocol | URL | Type of HTTP: standard (0) or secure (1). |
| 4 | num_dot | URL | Count of dots in a URL. |
| 5 | num_slash | URL | Count of slash in a URL. |
| 6 | num_double_slash | URL | Count of double slash in a URL. |
| 7 | num_hypen | URL | Count of dashes in a URL. |
| 8 | num_underscore | URL | Count of underscore in a URL. |
| 9 | num_equal | URL | Count of equals in a URL. |
| 10 | num_paranthesis | URL | Count of parenthesis in a URL. |
| 11 | num_curly_bracket | URL | Count of curly brackets in a URL. |
| 12 | num_square_bracket | URL | Count of square brackets in a URL. |
| 13 | num_less_and_greater | URL | Count of less that and greater than in a URL. |
| 14 | num_tilde | URL | Count of tilde in a URL. |
| 15 | num_asterisk | URL | Count of asterisk in a URL. |
| 16 | num_plus | URL | Count of plus in a URL. |
| 17 | url_inc_at | URL | If the URL includes an at symbol (1) or not (0). |
| 18 | url_inc_ip | URL | If the URL includes an IP address (1) or not (0). |
| 19 | response_history | HTTP | The HTTP response code that a server sends to a client with information about the result of the request made by the client to the server. |
| 20 | redirect | HTTP | Whether the website has a link to another site (1) or does not have a link to another site (0) identified using HTTP redirection response codes. |
| 21 | num_a_href | HTML | Count of <a> or link anchor tags on a website a webpage and is used to create links. |
| 22 | num_input | HTML | Count of <input> tags in a website used in creating input from elements. |
| 23 | num_button | HTML | Count of <button> tags in a website that is used for creating Button html elements to perform some actions. |
| 24 | num_link_href | HTML | Count of <link> tags in a website used to accept or refer other documents besides HTML. |
| 25 | num_iframe | HTML | Count of <iFrame> tags within a web page employed to import other documents into the website being analyzed. |

## 3.2 Multilayer Perceptron Model Architecture

According to the model, 25 input features extracted from the URLs, HTML content and HTTP responses of the websites will be used. Such features as domain_similarity, url_length, http_protocol, num_dot, num_slash, num_double_slash, num_hyphen, num_underscore, num_equal, num_paranthesis, num_curly_bracket, num_square_bracket, num_less_and_greater, num_tilde, num_asterisk, num_plus, url_inc_at, url_inc_ip, response_history, redirect, num_a_href, num_input, num_button, num_link_href, num_iframe, were found to be relevant in the identification of the nature of the website.

In the architecture, there will be one layer of hidden neurons. The number of nodes in the hidden layer will be 128 and 64. This structure enables the model to learn many features and relations in the dataset. The number of nodes is chosen in the range between a lower bound which restricts the model from memorizing the data and a higher bound which reduces the model representation power.

Hence the output layer will have one node and its activation function will be the sigmoid function. This code will give an output of the likelihood of the given website being that of phishing only but as a probability value between 0 and 1. A threshold of 0.5 will be used to classify the websites into two classes: 0 for legitimate and 1 for phishing. This binary output is useful where decisions have to be made immediately as in a real-time real life anti-phishing system.

The selection of the current MLP architecture is informed by its flexibility in capturing non-linear relationships inherent in the data set. The ReLU (Rectified Linear Unit) activation function will be used. It is effective in training deep networks, because, unlike its predecessor, the logistic sigmoid function, it helps us avoid the vanishing gradient problem [18]. ReLU enables the model to learn quickly and gives much better results in complex data sets than in other activations. The sigmoid function in the output layer is suitable in binary classification as it maps the output to a probability measure.

## 3.3 Evaluation Metrics

For the assessment of the performance of the phishing detection model, a set of evaluation parameters will be used such as accuracy, precision, recall, and F1 score.

Accuracy is another simple measure of performance that estimates the general accuracy of a model by evaluating the proportion of accurately classified instances: true positives and true negatives in the given array of data [13]. The formula for accuracy is given by:

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

The high accuracy consequently means that the model was also efficient in separating between the real and phishing sites.

Negative predictions do not require much attention and thus precision plays an important role in measuring the validity of identified positive instances predicted by the model. It was established as the percentage of true positives divided by the total number of cases predicted to be positive, including the false positives [19]. The formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

The technique indicates a good number of actual positives, which in turn gives a view of a low percentage of false positives; this proves the ability of the model to correctly classify the phishing sites.

Recall is also classified as sensitivity or the true positive rate, demonstrating the potential of a model to find all the actual positives. This metric should remind us to maximize the number of captured phishing instances and therefore minimize false negatives [20]. The formula for the recall is:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1 measures both and gives a fair measure of the test efficacy of the model. This metric is most useful in situations where high values are both precision and recall values because it provides an overall measure of the ability of the model to detect phishing websites while at the same time minimizing the number of misclassifications made by the model [22]. The formula for the F1 score is:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

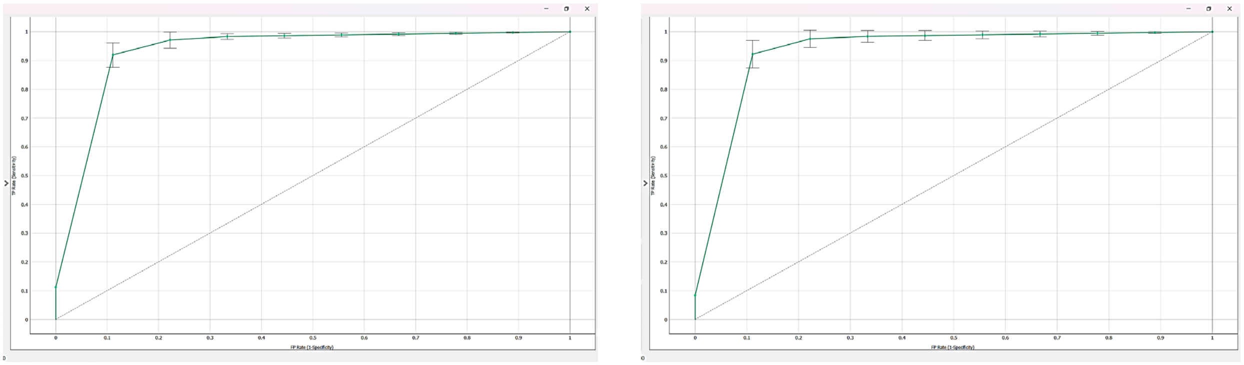
For the purpose of model validation, both 5-fold and 10-fold cross-validation approaches will be required. These techniques will guarantee that the model performance is tested credibly on the different parts of the data set so that its merits can be evaluated appropriately

## 4. Results and Discussion

This study compares the performance of multilayer perceptron with five machine learning models for phishing attack detection using two different cross-validation methods include 5-fold and 10-fold cross-validation. The models assessed are k-Nearest Neighbors (kNN), Decision Tree, Random Forest, Naive Bayes, Support Vector Machine (SVM) and Multilayer Perceptron (MLP). For each model, the accuracy namely Complete Accuracy (CA), F1 score, Precision and Recall were calculated.

### 4.1 k-Nearest Neighbors (kNN)

In the case of k-Nearest Neighbors (kNN) model, a mild enhancement was observed with 10-fold cross-validation where CA, F1 score, precision and recall ranged from 89.5% to 90.2% (Fig. 2). This means that a large number of folds impacts slightly on the enhanced stabilities and performances of the two methods.

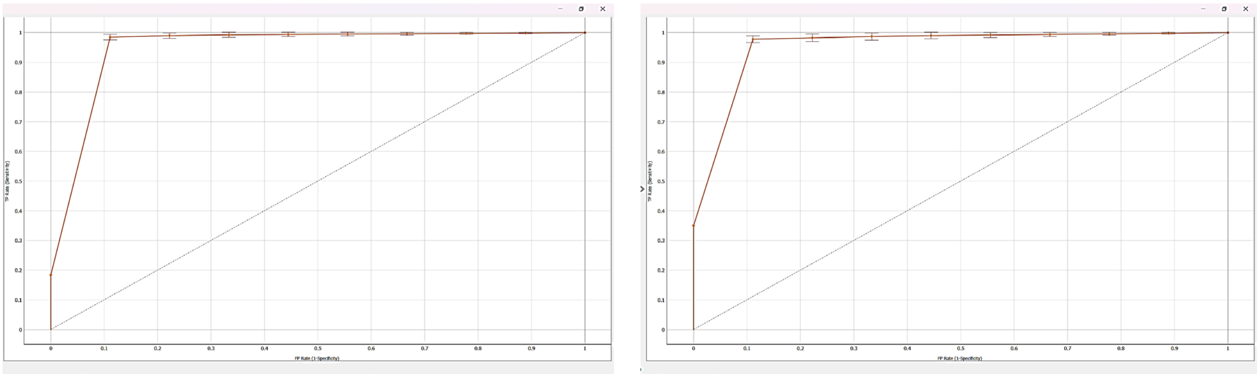


**(a)** **(b)**

**Fig. 2** *Comparison kNN (a) 5 folds; and (b) 10 folds*

### 4.2 Decision Tree

The Decision Tree model proved almost equally effective across both 5-fold and 10-fold cross-validation settings. Consequently, when using the 5-fold cross-validation method, the accuracy, f1-score, precision, and relative recall were 96.8% while, in turn, the accuracy and f1-score values, yielded from the 10-fold cross Validation, slightly dropped to 96.4% (Fig. 3). These slight variations indicated that the Decision Tree model is not very sensitive to changes in the fold count, and therefore, its classification ability remains almost unaffected. The fact that the benefit of such stability is that it shows the Decision Tree is insensitive to the number of fold having comparable results from few splits on data set.

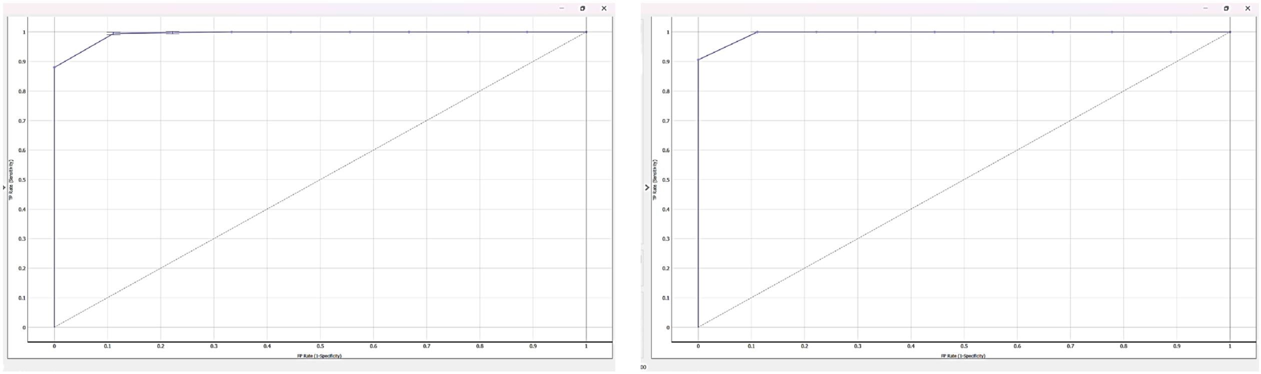


**(a)** **(b)**

**Fig. 3** *Comparison decision tree (a) 5 folds; and (b) 10 folds*

## 4.3 Random Forest

Both configurations of cross-validation proved to be effective in the Random Forest model, with the 5-fold configuration justifying a little better result than the 10-fold configuration. Classification accuracy, F1 score, the precision, and recall: using 5-fold cross-validation, the estimates of these metrics were 98.1%, whereas, when using 10-Fold cross-validation the estimates were slightly lower and made 97.4% (Fig. 4). This result suggests that Random Forest with less number of folds yielded a slightly better classification accuracy, which might be due to increased splits in 10 folds configuration that results to slight variations effects. However, both configurations prove that Random Forest is extremely suitable to phishing attack detection.

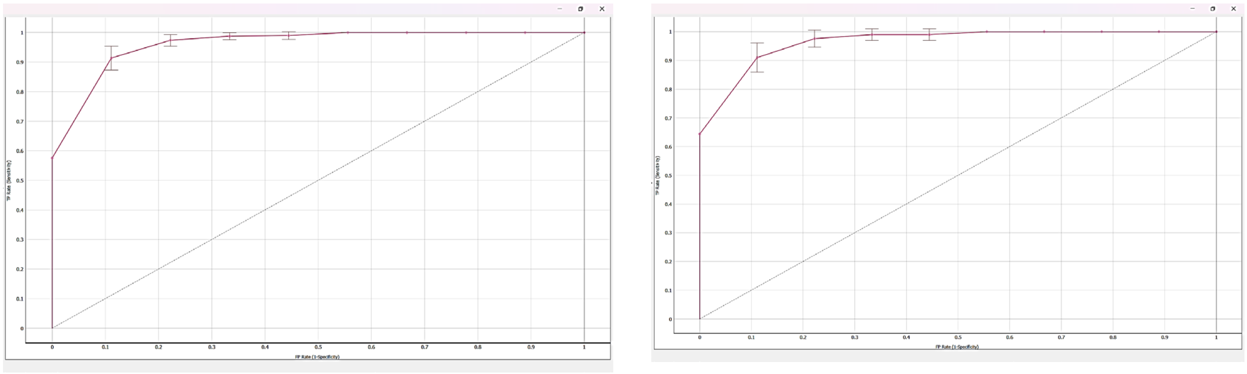


**(a)** **(b)**

**Fig. 4** *Comparison random forest (a) 5 folds; and (b) 10 folds*

## 4.4 Naïve Bayes

For both configurations of cross-validation, 5-fold as well as 10-fold, the precision attained by the Naive Bayes model is approximately 91.9% for classification accuracy, F1 score, precision, and recall (Fig. 5). This stability means that Naive Bayes is able to remain constant in terms of moving average in whichever fold count used in the analysis and appears to generalize the data appropriately. Such robustness shows that Naive Bayes does not heavily depend on data split structure, which is desirable in situations where different validation structures are to be used.

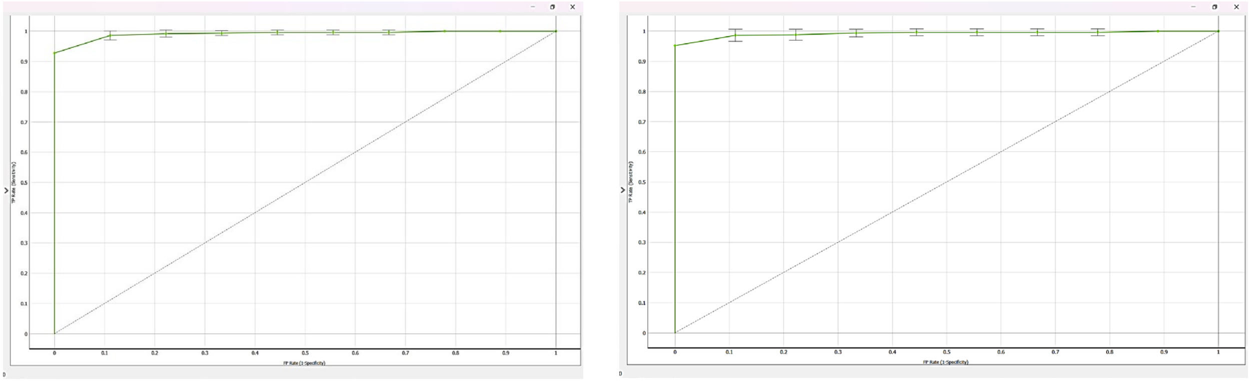


**(a)** **(b)**

**Fig. 5** *Comparison naïve bayes (a) 5 folds; and (b) 10 folds*

## 4.5 Support Vector Machine (SVM)

For the SVM model, minor enhancement in classification performance was noted when executing 10-folds the cross-validation (Fig. 6). In this configuration, we captured the classification accuracy, F1 score, precision, and recall as 97.2% towards the baseline of 96.9% stemming from the 5-fold setup. This increase in metrics is, however, small for the fold count with the higher value, which may indicate that SVM has more improvements when more folds are utilized, because it enables the model to learn across the variations in the data across more folds. This improvement corresponds to the fact that SVM is just able to slightly improve the generalization and consistency when more validation splits are used and yields better performance in average.

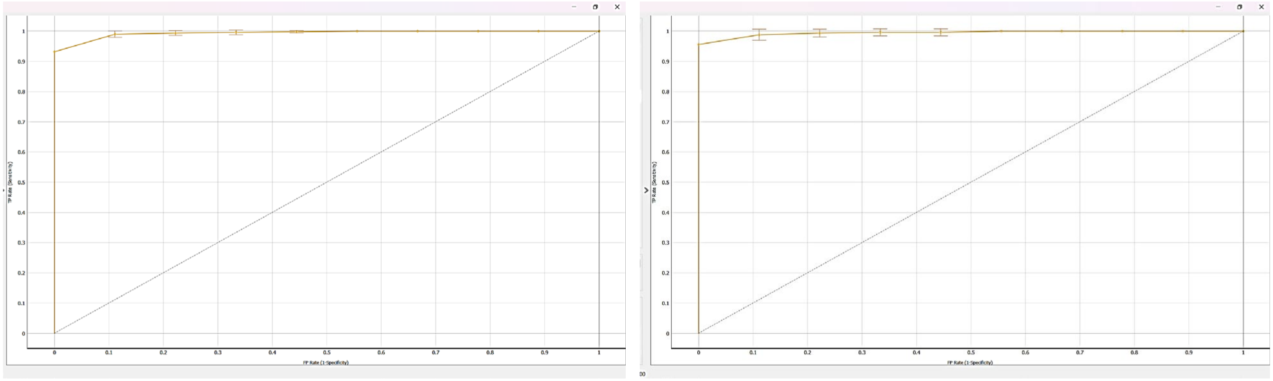


**(a)** **(b)**

**Fig. 6** *Comparison SVM (a) 5 folds; and (b) 10 folds*

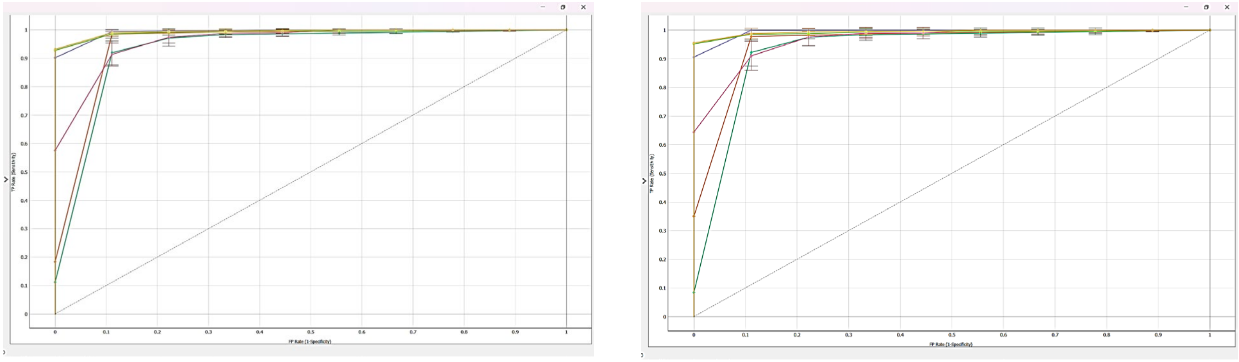
## 4.6 Multilayer Perceptron (MLP)

The MLP model did not have any difference in their performance metrics between 5-fold and 10-fold (Fig. 7 and Fig. 8). In both configurations, classification accuracy, F1 score, precision and recall were measured and obtained at 98.1%. This clearly means that the MLP possesses high level of robustness and stability by meaning to say that it's performance is not affected by the number of cross-validation folds specified. Such stability is useful as it shows that the model accurately transfers the learned data no matter the kind of cross validation used.



**(a)** **(b)**

**Fig. 7** *Comparison MLP (a) 5 folds; and (b) 10 folds*



**(a)** **(b)**

**Fig. 8** *Comparison five machine learning models (a) 5 folds; and (b) 10 folds with multilayer perceptron*

## 5. Summary

The findings based on these cross-validation identify the MLP model as the best choice regarding accuracy and stability. The results by 5-fold and 10-fold cross-validation show that this model provides good prediction accuracy and the feature importance rank robustness that would suit phishing attack detection. The performance of the Random Forest model was also good, especially when the 5-fold cross-validation was used, which makes it quite equal rival. Although it was slightly worse than the MLP, Random Forest could be valuable under conditions when interpretation of the model is needed and when training time is slightly faster. Therefore, the MLP has been suggested as the most accurate and stable model for this cybersecurity task. Nevertheless, Random Forest stays a reasonable backup, notably when interpretability enters into play.

## Acknowledgement

Communication of this research is made possible through monetary assistance by Universiti Utara Malaysia.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

## Author Contribution

*The authors confirm contribution to the paper as follows: **study conception and design:** Hui Ching Mah, Nor Hazlyna Harun; **data collection:** Hui Ching Mah, Nor Hazlyna Harun; **analysis and interpretation of results:** Hui Ching Mah, Nor Hazlyna Harun; **draft manuscript preparation:** Hui Ching Mah, Nor Hazlyna Harun. All authors reviewed the results and approved the final version of the manuscript.*

## References

[1] Cisco.com. (2021). *What Is Phishing? Examples and Phishing Quiz - Cisco*. https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html

[2] Cyberark. (2024). *What is phishing? | F-Secure*. https://www.cyberark.com/what-is/phishing/

[3] Qabajeh, I., Thabtah, F., & Chiclana, F. (2018). A recent review of conventional vs. automated cybersecurity anti-phishing techniques. In *Computer Science Review* (Vol. 29, pp. 44–55). Elsevier. https://doi.org/10.1016/j.cosrev.2018.05.003

[4] Smith, G. (2024). *Top Phishing Statistics for 2024: Latest Figures and Trends*. StationX. https://www.stationx.net/phishing-statistics/

[5] Palatty, N. (2023). *81 Phishing Attack Statistics 2024: The Ultimate Insight*. Astra. https://www.getastra.com/blog/security-audit/phishing-attack-statistics/

[6] Ozsahan, H., & Worthington, D. (2024). *50+ Phishing Attack Statistics for 2024*. Jumpcloud. https://jumpcloud.com/blog/phishing-attack-statistics

[7] Tian, C. (Annie), Jensen, M. L., & Durcikova, A. (2023). Phishing susceptibility across industries: The differential impact of influence techniques. *Computers & Security*, *135*, 103487. https://doi.org/10.1016/J.COSE.2023.103487

[8] Chan, K. Y., Abu-Salih, B., Qaddoura, R., Al-Zoubi, A. M., Palade, V., Pham, D. S., Ser, J. Del, & Muhammad, K. (2023). Deep neural networks in the cloud: Review, applications, challenges and research directions. *Neurocomputing*, *545*, 126327. https://doi.org/10.1016/j.neucom.2023.126327

[9] Mohsin, M. I., & Harun, N. H. (2024). Classifying Phishing Websites Using Multilayer Perceptron. *Emerging Advances in Integrated Technology*, *5*(1), 59–64.

[10] Dasgupta, D., Akhtar, Z., & Sen, S. (2022). Machine learning in cybersecurity: a comprehensive survey. *Journal of Defense Modeling and Simulation*, *19*(1), 57–106. https://doi.org/10.1177/1548512920951275

[11] Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. *Cybersecurity*, *5*(1), 1–22. https://doi.org/10.1186/s42400-021-00103-8

[12] Kapan, S., & Sora Gunal, E. (2023). Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features. *Applied Sciences (Switzerland)*, *13*(24). https://doi.org/10.3390/app132413269

[13] Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert Systems with Applications*, *53*, 231–242. https://doi.org/10.1016/j.eswa.2016.01.028

[14] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029

[15] Orunsolu, A. A., Sodiya, A. S., & Akinwale, A. T. (2022). A predictive model for phishing detection. *Journal of King Saud University - Computer and Information Sciences*, *34*(2), 232–247. https://doi.org/10.1016/j.jksuci.2019.12.005

[16] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(20), 1–22. https://doi.org/10.1109/AVSS.2018.8639152

[17] Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU). *ArXiv*. https://arxiv.org/abs/1803.08375v2

[18] Amer, M. (2022). *Classification Evaluation Metrics: Accuracy, Precision, Recall, and F1 Visually Explained*. Classification Evaluation Metrics: Accuracy, Precision, Recall, and F1 Visually Explained. https://cohere.com/blog/classification-eval-metrics

[19] GeeksforGeeks. (2024). *Methods to Minimize False Negatives and False Positives in Binary Classification*. https://www.geeksforgeeks.org/methods-to-minimize-false-negatives-and-false-positives-in-binary-classification/

[20] Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 1–14. https://doi.org/10.1038/s41598-024-56706-x