Performance Assessment of Machine Learning Models for Phishing URL Detection: Random Forest, XGBoost,and LightGBM

Bryan Orville Audric
Computer Science Departement
School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
bryan.audric@binus.ac.id

Michael Kurniawan
Computer Science Departement
School of Computer Science
Bina Nusantara University
Tangerang, Indonesia
michael.kurniawan009@binus.ac.id

***Abstract*—. The recent surge in online platforms has escalated cyber security threats, particularly phishing attacks aimed at acquiring users' sensitive data. In this context, machine learning models emerge as a promising solution. This study aims to evaluate the performance of several feature selection methods using various machine learning models. These feature selection methods, which are Pearson Correlation, Chi Square and Random Forest, were applied to the utilized dataset, with Random Forest, LightGBM, and XGBoost employed to assess the performance of each method. The research findings indicate that Every feature selection method contributes identical accuracy of 97% with every model except for Random Forest, showing an accuracy of 96% when using Chi Square's subset of features.**

Keywords—*Phishing, Machine Learning, Pearson Correlation, Chi Square, Random Forest, LightGBM, XGBoost*

I. INTRODUCTION

The exponential growth of online platforms and digital services has revolutionized the way individuals and organizations operate, bringing numerous benefits but also posing significant security challenges. Among these challenges, cyber threats have become increasingly sophisticated, with phishing attacks emerging as a major concern. Phishing involves deceptive practices aimed at tricking individuals into disclosing sensitive information, such as usernames, passwords, and financial details. These attacks are not merely confined to data theft; they encompass a range of malicious activities, including identity theft, financial fraud, and corporate espionage. The attackers often impersonate legitimate entities, creating a facade of trust to lure victims into divulging their private information [1].

In response to these challenges, the application of machine learning (ML) in cybersecurity has gained significant traction. ML models can analyse vast amounts of data, identify patterns, and make predictions with high accuracy, making them well-suited for detecting phishing attacks [3]. By leveraging ML, it is possible to develop automated systems that adapt to new phishing strategies, thereby enhancing the robustness and effectiveness of phishing detection mechanisms [4].

This study focuses on evaluating the effectiveness of various machine learning models, which are Random Forest, XGBoost and LightGBM, in detecting phishing URLs. The study also emphasizes the importance of feature selection during feature engineering, employing Pearson correlation, Chi Square and Random Forest to identify the most relevant features for the models. By comparing the performance of these models, this research aims to provide insights into their capabilities and limitations, contributing to the ongoing efforts to combat phishing attacks in an increasingly digital world.

II. RELATED WORKS

A. Research that highlights the use of feature selection

There are several attempts at creating an accurate phishing URL detection by using a specific filter method to filter out unnecessary features before inserting into a model. This research aimed to detect phishing websites using machine learning with filter features, filtered by Pearson correlation. It achieves the highest accuracy rate of 96.3% [6]. The research utilizes Random Forest for feature selection and ensemble methods as the model which produces a superior performance up to 95% accuracy [3]. This study utilized several filter methods to rank and select features for experiments aimed at detecting phishing websites. These methods included the Chi-Square Filter, Pearson Correlation Filter, Information Gain Filter, and Relief Filter. The Chi-Square Filter ranked features based on their relevance to phishing detection, demonstrating improved accuracy with a smaller feature set when used in conjunction with the Random Forest classifier. The study's analysis highlighted the Random Forest classifier's superior accuracy across various feature classes, with the Chi-Square Filter notably outperforming others with a higher accuracy of 96.83% while utilizing fewer features [7].

B. Research that highlights the use of machine learning models

Other study has proved that a machine learning algorithm with no feature selection can also detect phishing URL accurately. The study focused on classifying URLs into phishing, suspicious, or legitimate categories using machine learning techniques. Four classifiers were evaluated: decision tree, Naïve Bayes' classifier, SVM, and Neural Network. The classifiers were tested, and the results showed successful classification of websites with highest accuracy of 90% achieved by pruned decision tree[8]. This study employed machine learning algorithms in detecting phishing websites, with the XGBoost algorithm achieving the highest accuracy of 94% in the training phase and 91% in the testing phase. Other algorithms like Multilayer Perceptron, Random Forest, and Decision Tree also showed accuracies ranging from 90%

to 91%[9]. The research uses seven machine learning algorithm such as Decision Tree, Adaboost, Kstar, kNN, Random Forest, SMO and Naïve Bayes and all model reaches over 90% accuracy[2]. This research focuses on the detection of phishing websites using machine learning methods. It compares and evaluates the performance of various machine learning models, such as Logistic Regression, Decision Tree, Random Forest, Ada-Boost, Support Vector Machine, KNN, Neural Networks, Gradient Boosting, and XGBoost[10].This study proposed machine learning model that is RNN-LSTM model. The result shows promising accuracy of 97% when identifying phishing URLs[11]. The ensemble models, particularly Ensemble Model 5 and Ensemble Model 6, achieved high accuracy scores of 99.95% and 99.94%, respectively[12]. The research utilized a machine learning-based technique for detecting phishing websites, specifically employing a Support Vector Machine (SVM) classifier. The Support Vector Machine model achieved an accuracy of 95.66% in detecting phishing websites [4].

C. *Research that highlights the use of deep learning models*

Other researchers also used deep learning to create an accurate model. This study proposed a real-time phishing website detection framework based on deep learning, achieving a remarkable accuracy of 99.18% with the RNN-GRU model on the KPT-12 dataset[13]. This research aimed to enhance phishing URLs detection by applying parallel processing to machine learning and deep learning models, providing insights into the comparison between sequential and parallel ML execution[14]

III. METHODOLOGY

There are several processes shown on Fig. 1 that need to be followed. The process starts with gathering a dataset that contain numerous features, representing the characteristic of phishing and benign URL. The features are then filtered using Pearson correlation, Chi Square and Random Forest. The filtered features are then used for the machine learning models to train and predict. The machine learning models chosen for this study are Random Forest, XGBoost and LightGBM. The models are then evaluated using metrics such as accuracy, recall, precision, and F1 Score.

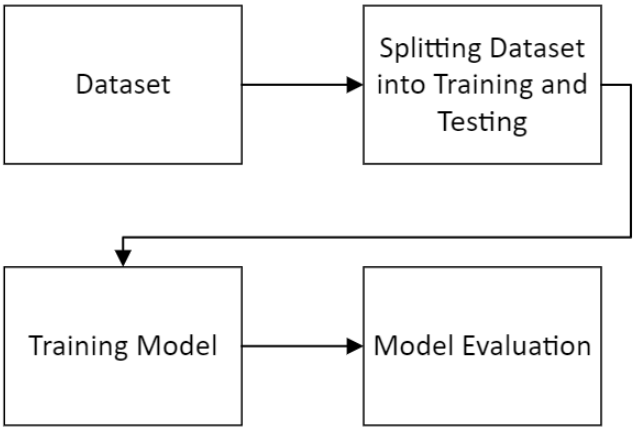

Fig. 1. Project Process

A. *Dataset*

This study employed the same dataset from this research [7], which contains 11430 URLs and 87 features such as IP Address, URL lengths, Shortening Service, Special Character, HTTPs Token and so on. The list of Features including the details can be seen in [7].

B. *Pearson Correlation*

Pearson's Correlation is implemented during feature selection. Pearson correlation is known to produce relevant features that can contribute higher accuracy when implemented on machine learning models as shown on this research [6]. Pearson's correlation coefficient, a statistical metric, is instrumental in the feature selection process. This coefficient quantifies the linear relationship between features and the target variable, discerning their relevance. By calculating correlation coefficients for each feature in the dataset with respect to the target variable, researchers can gauge their strength of association. Features exhibiting strong correlations, closer to 1 or -1, are considered more pertinent for distinguishing phishing from legitimate URLs. Conversely, features with coefficients near 0 indicate minimal linear relationship and are thus deemed less relevant for classification. The formula for calculating the Pearson correlation coefficient between two variables is given by:

$$r_{xy} = \frac{\sum(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum(X_i - \overline{X})^2 \sum(Y_i - \overline{Y})^2}} \quad (1)$$

Where r_{xy} is the Pearson correlation coefficient between variables X and Y, X_i, and Y_i are individual values of variable X and Y , and $\overline{X}$ and $\overline{Y}$ are the means of the values of X and Y, respectively. This formula measures the extent to which variables X and Y vary together, providing a deeper understanding of the relationship between features in the dataset and the target variable. Pearson correlation coefficient is broadly admitted in the feature selection algorithm to determine the best feature set.

C. *Chi Square*

Chi-Square is also implemented during feature selection and is also known to produce relevant features that can contribute to higher accuracy as shown in this [7]. Chi-square (χ^2) test is a statistical method utilized in feature selection, focusing on assessing the independence between categorical variables. In the context of phishing URL detection, features can be numerical or categorical variables representing different attributes of URLs. The Chi-square test calculates the statistical significance of the association between each feature and the target variable (i.e., whether a URL is phishing or benign). It is computed using the formula:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

Where χ^2 is the Chi-Square statistic, O_i is the observed frequency, and E_i is the expected frequency under the null hypothesis of independence. This statistical test will return the Chi-Square value and p value of every feature. The most relevant features are indicated by high Chi-Square value and low p value.

D. *Random Forest*

Random Forest is a widely used ensemble learning model comprised of multiple decision trees. Each tree is constructed independently and randomly, with a subset of features selected randomly during the tree-building process. This

feature selection technique helps mitigate overfitting and decorrelate the trees within the ensemble. Through bootstrap aggregating or bagging, Random Forest generates multiple bootstrap samples from the training data and builds a decision tree on each sample. During classification tasks, the model employs a majority voting mechanism, where each tree "votes" for the most prevalent class[15]. In this study, Random Forest is also used to select most important features by calculating the total decrease in node impurity.

E. XGBoost

XGBoost is a scalable tree boosting system widely used in machine learning for its effectiveness and efficiency. It works by sequentially building an ensemble of decision trees, where each tree corrects the errors of the previous ones through boosting. By optimizing an objective function that combines a loss function and regularization term, XGBoost minimizes errors while preventing overfitting. Leveraging gradient boosting, XGBoost fits trees to the gradient of the loss function, learning from previous mistakes. With parallel and distributed computing capabilities, XGBoost can efficiently utilize multiple CPU cores and scale to large datasets. It also provides insights into feature importance and is designed for scalability, handling billions of examples with minimal resources[5].

F. LightGBM Classifier

LightGBM is a state-of-the-art Gradient Boosting Decision Tree (GBDT) algorithm that revolutionizes traditional implementations by introducing novel techniques like Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS optimizes the training process by sampling data instances based on gradient values, focusing on instances with significant gradients to accurately estimate information gain while reducing computational overhead. EFB enhances efficiency by bundling related features together, improving cache hit rates and spatial locality. By leveraging these techniques, LightGBM accelerates model training, reduces memory consumption, and maintains high accuracy, making it a powerful tool for handling large feature dimensions and datasets in machine learning tasks[16].

G. Accuracy

Accuracy is one of the commonly used evaluation metrics in classification modeling to assess how well a model can classify data correctly overall. In this context, "correct" means the model's predictions match the true labels of the observed data. Accuracy is calculated by dividing the number of correct predictions by the total number of predictions made by the model. The formula for calculating the accuracy of a classification model is as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Prediction}} \quad (3)$$

H. Precision

Precision is a crucial evaluation metric in classification tasks, particularly in scenarios where minimizing false positives is important. It measures the accuracy of positive predictions made by the model, i.e., the proportion of correctly predicted positive instances out of all instances predicted as positive. The formula for calculating precision in a binary classification setting is as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4)$$

I. Recall Score

Recall, also known as sensitivity or true positive rate, is an essential evaluation metric in classification tasks, especially when it's crucial to capture all positive instances. It measures the ability of the model to correctly identify positive instances from all actual positive instances in the dataset. The formula for calculating recall in a binary classification setting is as follows:

$$\text{Recall Score} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (5)$$

J. F1-Score

The F1 score is a commonly used evaluation metric in classification tasks, which combines precision and recall into a single metric. It provides a balance between precision and recall, making it particularly useful when both false positives and false negatives need to be minimized. The formula for calculating the F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

Where Precision and Recall are as previously defined. This formula computes the harmonic mean of precision and recall, emphasizing the balance between the two metrics. The F1 score ranges from 0 to 1, where a higher score indicates better performance. It is a useful metric for evaluating classification models, especially in scenarios where achieving a balance between precision and recall is important, such as in information retrieval systems or medical diagnosis.

IV. Result and Discussion

A. Feature Filtration Result

Pearson correlation produces 48 features where each feature has an absolute correlation coefficient value above a threshold value which was set to be 0.1. The features are shown in Table I.

TABLE I. Features Extracted using Pearson Correlation

Pearson Features	
Length URL	Number of Semicolon
Length Hostname	‘www’ String in URL
IP Address in URL	‘Com’ String in URL
Number of Dots	HTTPS Token
Number of Hyphens	Ratio Digits URL
Number of @ symbol	Ratio Digits Host
Number of Question mark	TLD in Subdomain
Number of & Symbol	Number of Subdomains
Number of = Symbol	Prefix Suffix
Number of Slash	Shortening Service

Pearson Features	
Length Words Raw	Ratio External Media
Shortest Word Host	Sage Anchor
Longest Words Raw	Empty Title
Longest Word Host	Domain in Title
Longest Word Path	Domain with Copyright
Average Words Raw	Domain Registration Length
Average Word Host	Domain Age
Average Word Path	DNS Record
Phish Hints	Google Index
Suspicious TLD	Page Rank
Statistical Report	Ratio External Hyperlinks
Number of Hyperlinks	External Favicon
Ratio Internal Hyperlinks	Links in Tags
Ratio Internal Media	Abnormal Subdomains

After the Chi Square value and p value of every feature are obtained, the features are ranked descending based on Chi Square value and ranked ascending based on p value. The top 48 features as shown in Table II are then selected

TABLE II. FEATURES EXTRACTED USING CHI SQUARE

Chi Square Features	
Web Traffic	Number of Question mark
Domain Age	Number of External CSS
Number of Hyperlinks	Average Words Raw
Domain Registration Length	Average Word Host
Length URL	Number of Semicolon
Longest Word Path	Number of Slash
Ratio Internal Media	Number of Hyphens
Longest Words Raw	TLD in Subdomains
Ratio External Media	Statistical Report
Safe Anchor	Empty Title
Links in Tags	Prefix Suffix
Page Rank	Longest Word Host
Average Word Path	Number of Dots
Length Hostname	'com' in URL
Google Index	Domain in Title
Phish Hints	Number of @ Symbol
Number of = Symbol	Shortest Word Path
Length Words Raw	Ratio Digits URL
Shortest Word Host	Ratio Digits Host
Number of & Symbol	Domain with Copyright
'www' String in URL	Abnormal Subdomain
IP Address in URL	DNS Record
Ratio Internal Hyperlinks	External Favicon
Number of % Symbol	Suspicious TLD

After the importances value of every feature is obtained when fitted into Random Forest, the features are ranked descending based on importances value. The top 48 features as shown in Table III are then selected.

TABLE III. FEATURES EXTRACTED USING RANDOM FOREST

Random Forest Features	
Google Index	Number of Dots
Page Rank	Ratio Digits Host
Number of Hyperlinks	Shortest Word Path
Web Traffic	Number of Slash
Domain Age	Average Words Raw
'www' String in URL	Number of Hyphens
Phish Hints	Average Word Host
Ratio Internal Hyperlinks	Number of Question mark
Ratio External Hyperlinks	Shortest Words Raw
Longest Word Path	Longest Word Host
Safe Anchor	Domain with Copyright
Ratio Digits URL	Number of = Symbol
Ratio External Redirection	IP Address in URL
Length URL	Ratio External Media
Longest Words Raw	Ratio Internal Media
Average Word Path	Ratio External Errors
Duplicate Character	Number of External CSS
Length Words Raw	Number of Redirection
Domain in Title	Number of Subdomains
HTTPS Token	Empty Title
Length Hostname	Domain in Brand
Shortest Word Host	Shortening Service
Links in Tags	Number of Underscore
Domain Registration Length	Prefix Suffix

B. Performance Result

After training and evaluating each machine learning modes based on every subset of features, the accuracy, precision, recall and F1 score are obtained. Table IV shows the evaluation results of features selected using Pearson Correlation in which all three models show identical overall accuracy of 97% and exhibit very similar performance metrics for both benign and phishing classifications.

TABLE IV. CLASSIFICATION REPORT OF FILTERED FEATURES USING PEARSON CORRELATION

Model	Type	Accuracy	Precision	Recall	F1 Score
Random Forest	Benign	0.97	0.96	0.97	0.97
	Phishing		0.97	0.96	0.97
XGBoost	Benign	0.97	0.97	0.98	0.97
	Phishing		0.97	0.97	0.97
LGBM Classifier	Benign	0.97	0.97	0.98	0.97
	Phishing		0.97	0.97	0.97

Table V shows the evaluation results of features selected using Chi Square in which the highest accuracy of 97% was achieved by XGBoost and LightGBM Classifier. Random Forest shows slightly lower accuracy of 96% for both benign and phishing classification

TABLE V. CLASSIFICATION REPORT OF FILTERED FEATURES USING CHI SQUARE

Model	Type	Accuracy	Precision	Recall	F1 Score
Random Forest	Benign	0.96	0.96	0.96	0.96
	Phishing		0.96	0.96	0.96
XGBoost	Benign	0.97	0.97	0.97	0.97
	Phishing		0.97	0.97	0.97
LGBM Classifier	Benign	0.97	0.97	0.97	0.97
	Phishing		0.97	0.97	0.97

Table VI shows the evaluation results of features selected using Random Forest in which all models also show identical overall accuracy of 97% and exhibit similar performance for both benign and phishing amongst three models.

TABLE VI. CLASSIFICATION REPORT OF FILTERED FEATURES USING RANDOM FOREST

Model	Type	Accuracy	Precision	Recall	F1 Score
Random Forest	Benign	0.97	0.96	0.97	0.97
	Phishing		0.97	0.96	0.96
XGBoost	Benign	0.97	0.97	0.97	0.97
	Phishing		0.97	0.97	0.97
LGBM Classifier	Benign	0.97	0.97	0.97	0.97
	Phishing		0.97	0.97	0.97

C. Discussion

Every feature selection method contributes identical accuracy of 97% with every model except for Random Forest, showing an accuracy of 96% when using Chi Square's subset of features. When comparing the result with [7] in which the highest was 96.86% using Chi Square's feature in Random Forest Model, the use of Pearson Correlation's feature in XGBoost, the use of Chi Square's feature in XGBoost and the use of Random Forest's feature in XGBoost exhibit slightly higher accuracy.

V. CONCLUSION

This study highlights the effectiveness of machine learning techniques, specifically Random Forest, XGBoost, and LightGBM, in accurately detecting phishing websites. By utilizing features extracted through methods like Pearson correlation, Chi Square and Random Forest, most models achieved higher accuracy scores of 97%, slightly higher in comparison with the result in [7] when distinguishing between benign and phishing URLs.

The study also emphasizes the importance of feature selection in enhancing the performance of machine learning models for cybersecurity applications. However, it is important to note that phishing patterns found in URL will vary every year, hence modifying the dataset over time is necessary to maintain the same performance.

REFERENCES

[1] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," Frontiers in Computer Science, vol. 3. Frontiers Media S.A., Mar. 09, 2021. doi: 10.3389/fcomp.2021.563060.

[2] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Syst Appl, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.

[3] A. A. Ubing, S. Kamilia, B. Jasmi, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning," 2019. [Online]. Available: www.ijacsa.thesai.org

[4] J. Rashid, T. Mahmood, M. W. Nisar, and T. Nazir, "Phishing Detection Using Machine Learning Technique," in Proceedings - 2020 1st International Conference of Smart Systems and Emerging Technologies, SMART-TECH 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 43–46. doi: 10.1109/SMART-TECH49988.2020.00026.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[6] V. Aprelia Windarni, A. Ferdita Nugraha, S. Tri Atmaja Ramadhani, D. Anisa Istiqomah, F. Mahananing Puri, and A. Setiawan, "DETEKSI

WEBSITE PHISHING MENGGUNAKAN TEKNIK FILTER PADA MODEL MACHINE LEARNING," 2023.
[7] A. Hannousse and S. Yahiouche, "Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An experimental study," Oct. 2020, doi: 10.1016/j.engappai.2021.104347.
[8] A. D. Kulkarni, L. L. Brown, and A. Kulkarni, "Phishing Websites Detection using Machine Learning," 2019. [Online]. Available: http://hdl.handle.net/10950/1862www.ijacsa.thesai.org
[9] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning," Sensors, vol. 23, no. 19, Oct. 2023, doi: 10.3390/s23198070.
[10] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing Detection Using Machine Learning Techniques," Sep. 2020, [Online]. Available: http://arxiv.org/abs/2009.11116
[11] A. K. Dutta, "Detecting phishing websites using machine learning technique," PLoS One, vol. 16, no. 10 October, Oct. 2021, doi: 10.1371/journal.pone.0258361.
[12] A. Mittal, H. Kommanapalli, R. Sivaraman, T. Chowdhury, T. Chowdhury, and D. W. Engels, "Phishing Detection Using Natural Language Processing and Machine Learning," 2022. [Online]. Available: https://scholar.smu.edu/datasciencereviewAvailableat:https://scholar.smu.edu/datasciencereview/vol6/iss2/14http://digitalrepository.smu.edu.
[13] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," IEEE Access, vol. 10, pp. 1509–1521, 2022, doi: 10.1109/ACCESS.2021.3137636.
[14] N. Nagy et al., "Phishing URLs Detection Using Sequential and Parallel ML Techniques: Comparative Analysis," Sensors, vol. 23, no. 7, Apr. 2023, doi: 10.3390/s23073467.
[15] G. Biau and G. B. Fr, "Analysis of a Random Forests Model," 2012.
[16] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." [Online]. Available: https://github.com/Microsoft/LightGBM.