ISSN: 2581-1274

Comparison of the Performance of Machine Learning Classification Algorithms on Phishing URL Detection

Ahmad [a,1,*], Sartika Lina Mulani Sitio [a,2]

[a] University of Pamulang, Jl. Raya Puspitek, South Tangerang 15310, Indonesia
[1] *dosen02594@unpam.ac.id**; [2] *dosen00847@unpam.ac.id*
* *corresponding author*

ARTICLE INFO

Article history:
Published
July 30, 2026

Keywords:
Cybersecurity
Gradient Boosting
Machine Learning
Phishing Detection
URL Classification

ABSTRACT

Phishing attacks continue to evolve and remain one of the most significant cybersecurity threats because malicious Uniform Resource Locators (URLs) can closely resemble legitimate websites, making conventional blacklist-based detection increasingly ineffective against newly generated phishing URLs. Although numerous machine learning models have been proposed for phishing detection, previous studies generally focus on evaluating a limited number of classifiers or emphasize a single algorithm, resulting in limited empirical evidence regarding their comparative performance under identical experimental conditions. This study aims to provide a comprehensive evaluation of multiple machine learning and deep learning classifiers for phishing URL detection while identifying the most influential URL features contributing to classification performance. The experiments were conducted using a publicly available Kaggle dataset containing 11,054 URL instances with 31 features representing phishing and legitimate websites. Ten classification algorithms, namely Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, CatBoost, Extreme Gradient Boosting, and Multilayer Perceptron, were evaluated using Accuracy, Precision, Recall, and F1-score. Feature analysis indicates that HTTPS, AnchorURL, and WebsiteTraffic are among the most influential attributes for distinguishing phishing from legitimate URLs. Experimental results demonstrate that the Gradient Boosting classifier achieved the best overall performance with an accuracy of 97.4%, precision of 98.6%, recall of 99.4%, and F1-score of 97.7%, outperforming the other evaluated models. These findings indicate that ensemble boosting methods provide more robust and reliable phishing URL detection than conventional classifiers and can serve as an effective foundation for intelligent phishing prevention systems.

I. Introduction

The development of internet technology and digital services has increased the intensity of online information exchange, especially in the banking, e-commerce, social media, and cloud-based services sectors. This condition also increases the threat of cybercrime, one of which is phishing. Phishing attacks are carried out by manipulating users through fake URLs or websites that resemble official websites to obtain sensitive information, such as usernames, passwords, and financial data [1][2]. Phishing is one of the most dangerous forms of cyber-attacks because it is able to exploit user weaknesses through social engineering techniques. Previous research has shown that blacklist and rule-based detection methods are starting to be less effective in detecting new phishing URLs that continue to grow dynamically [3][4]. Therefore, the Machine Learning-based approach is a widely

used solution because it is able to recognize the patterns and characteristics of phishing URLs automatically with a high level of accuracy [5][6].

Previous studies have developed phishing detection models using a variety of classification algorithms. The research by [5] implemented the Random Forest, Decision Tree, Logistic Regression, and Support Vector Machine algorithms to detect phishing URLs based on URL characteristics and obtain good classification performance. Other research developed by [6] through the PhishNot system, utilizing a cloud-based machine learning approach with feature reduction using Recursive Feature Elimination (RFE), showed that Random Forest produced an accuracy of 97.5% with a very fast detection time. In addition, research by [7] compared 24 classifiers from six learning strategies and showed that Random Forest, FilteredClassifier, and J-48 performed best at detecting website phishing. Meanwhile, research by [8] compared Artificial Neural Network, Support Vector Machine, Decision Tree, and Random Forest using domain phishing datasets and found that Random Forest provided the most optimal classification results. Other research by [9] developed a voting classifier-based hybrid machine learning model that combines Logistic Regression, Support Vector Machine, and Decision Tree to improve the accuracy of URL phishing detection.

Although previous studies have produced high classification performance, most studies have focused on the use of specific algorithms or limited combinations, so they have not provided a comprehensive comparative analysis of different machine learning and deep learning methods on the same phishing dataset. In addition, some studies still focus on improving accuracy without paying attention to model efficiency, feature selection, and the model's generalization ability to new data [10]. Therefore, this study was conducted to analyze and compare the performance of several classification algorithms in detecting phishing URLs so that it can be determined that the best performing methods can be determined based on the accuracy, precision, recall, F1-score, and ROC-AUC values.

This study uses a URL phishing dataset obtained from Kaggle with a total of 11,054 data points and 32 features that represent the characteristics of phishing and legitimate URLs. The methods used in the study include Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, CatBoost, Extreme Gradient Boosting, and Multilayer Perceptron. The research stages start from data preprocessing, data sharing training and testing, model training, performance evaluation, and analysis of classification results to determine the best model in detecting phishing URLs. The contribution of this research lies in presenting a comparative analysis of various machine learning and deep learning algorithms using the same phishing dataset so that the evaluation results can be compared objectively [11]. This study also provides an overview of the influence of URL feature characteristics on the performance of classification models and provides an empirical reference for the development of more effective and adaptive phishing detection systems in modern cybersecurity environments. In addition, the results of the research are expected to be the basis for the development of an artificial intelligence-based automatic phishing detection system that is able to assist users in identifying malicious URLs quickly and accurately.

II. Method

The research method in this study uses a Machine Learning approach to detect phishing URLs based on the characteristics and features of website URLs. The research began with the collection of phishing URL datasets, and then an Exploratory Data Analysis (EDA) process was carried out to understand the structure, distribution, and quality of the data. Furthermore, the data is visualized to see the pattern of relationships between features before being separated into training and testing data. In the modeling stage, several classification algorithms such as Logistic Regression, Random Forest, Support Vector Machine, and Multilayer Perceptron were used to train phishing detection models. The results of each model are then evaluated and compared using accuracy, precision, recall, and F1-score metrics to determine the best algorithm for accurately identifying phishing URLs [8]. The stages of the research method can be seen in Figure. 1.

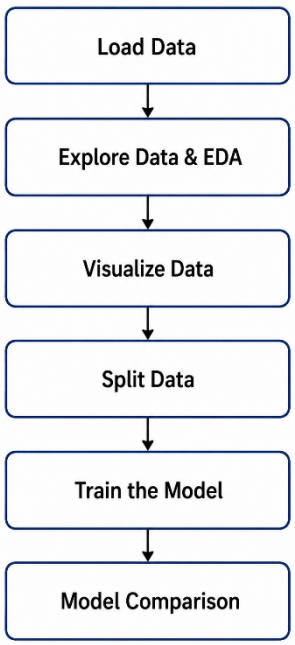

Fig. 1. Stages of Research Method

1. Load Data

Data collection in this study was carried out using a phishing URL dataset obtained from Kaggle. The dataset contains URL data that has been classified into phishing and legitimate categories so that it can be used for training processes and testing Machine Learning models. The dataset used has a total of 11,054 data points with 32 features (attributes) that represent the characteristics of URLs and websites, such as IP address usage, URL length, HTTPS usage, number of subdomains, URL redirects, and various other security features [12]. Each piece of data has a class label that indicates whether or not the URL is phishing. This dataset was chosen because it has a large enough amount of data and relevant features to support the process of analysis, comparison, and evaluation of various classification algorithms in order to detect phishing URLs effectively [13].

Table 1. Sample Data

No	Fitur						
	Having_IP	***URL_Length***	***Shortining_ Service***	***Having-At_Symbols***	***HTTPS***	***Domain Age***	***Result***
1	-1	1	1	-1	1	-1	Phishing
2	1	-1	-1	1	1	1	Legitimate
3	-1	-1	1	-1	-1	-1	Phishing
4	1	1	-1	1	1	1	Legitimate
5	-1	1	1	-1	-1	-1	Phishing

2. Explore Data and EDA

The Exploratory Data Analysis (EDA) stage is carried out to understand the structure and characteristics of the dataset before the modeling process is carried out. At this stage, the amount of data, data type, class distribution, missing value, and correlation between features are checked. This analysis is important to know the quality of the data and determine the necessary preprocessing steps. Previous research has shown that URL feature analysis has a significant effect on the performance of phishing detection models because URL features are able to effectively describe phishing website behavior patterns [14].

3. Visualize Data

The data visualization stage is carried out to facilitate the interpretation of the pattern and distribution of phishing and legitimate data. Visualization is performed using histograms, countplots, boxplots, and heatmap correlation to see the relationships between features and the distribution of dataset classes [15]. The visualization process helps in understanding the dominant features that affect the URL phishing classification process. The data visualization approach has also been used in previous studies to improve understanding of phishing attack patterns and the characteristics of malicious URLs [16][17].

4. Split Data

The data separation stage is carried out by dividing the dataset into training and testing data using the train-test split method. Data training is used to train classification models, while data testing is used to evaluate the model's ability to detect phishing URLs in new data. Data sharing is done to avoid overfitting and ensure that the model is able to generalize well to data that has not been studied before [18].

5. Train the Model

The model training stage was carried out using several classification algorithms, namely Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, CatBoost, Extreme Gradient Boosting, and Multilayer Perceptron. Each model is trained using data training, and performance evaluations are carried out using accuracy, precision, recall, and F1-score [19]. Previous research has shown that Random Forest, XGBoost, and Deep Learning have high performance in detecting URL phishing [20][21].

6. Model Comparison

The final stage is to compare the performance of all classification models to determine the best algorithm for detecting phishing URLs. Comparisons were made based on accuracy, precision, recall, F1-score, and confusion matrix values. The best-performing model is selected as the optimal model for the implementation of a phishing detection system. This comparative approach is widely used in phishing detection research because it is able to provide an objective evaluation of the effectiveness of each classification algorithm [22].

III. Results and Discussion

The results and discussion presented the results of the implementation of various Machine Learning and Deep Learning algorithms in detecting phishing URLs based on the datasets that have been used. In this section, an analysis of the Processing dataset, data visualization, model training process, and performance evaluation of each algorithm was carried out using accuracy, precision, recall, and F1-score metrics.

A. Processing Dataset

The dataset used in this study was obtained from the Kaggle repository and consists of 11,054 URL samples with 31 features, including 30 independent features representing URL characteristics and one dependent feature indicating phishing or legitimate classes. The class distribution comprises 6,157 legitimate URLs and 4,897 phishing URLs, indicating a relatively balanced dataset; therefore, no class balancing techniques were applied. All features are stored as integer (int) values, eliminating the need for categorical encoding. Data preprocessing included checking data types, missing values, duplicate records, and class distribution. The results showed that the dataset contains no missing values or duplicate data, making it suitable for model training. Although the dataset is widely used for phishing detection research, it may contain bias because it represents phishing URLs collected from specific sources and periods.

	count	mean	std	min	25%	50%	75%	max
UsingIP	11054.0	0.313914	0.949495	-1.0	-1.0	1.0	1.0	1.0
LongURL	11054.0	-0.633345	0.765973	-1.0	-1.0	-1.0	-1.0	1.0
ShortURL	11054.0	0.738737	0.674024	-1.0	1.0	1.0	1.0	1.0
Symbol@	11054.0	0.700561	0.713625	-1.0	1.0	1.0	1.0	1.0
Redirecting//	11054.0	0.741632	0.670837	-1.0	1.0	1.0	1.0	1.0
PrefixSuffix-	11054.0	-0.734938	0.678165	-1.0	-1.0	-1.0	-1.0	1.0
SubDomains	11054.0	0.064049	0.817492	-1.0	-1.0	0.0	1.0	1.0
HTTPS	11054.0	0.251040	0.911856	-1.0	-1.0	1.0	1.0	1.0
DomainRegLen	11054.0	-0.336711	0.941651	-1.0	-1.0	-1.0	1.0	1.0
Favicon	11054.0	0.628551	0.777804	-1.0	1.0	1.0	1.0	1.0
NonStdPort	11054.0	0.728243	0.685350	-1.0	1.0	1.0	1.0	1.0
HTTPSDomainURL	11054.0	0.675231	0.737640	-1.0	1.0	1.0	1.0	1.0
RequestURL	11054.0	0.186720	0.982458	-1.0	-1.0	1.0	1.0	1.0
AnchorURL	11054.0	-0.076443	0.715116	-1.0	-1.0	0.0	0.0	1.0
LinksInScriptTags	11054.0	-0.118238	0.763933	-1.0	-1.0	0.0	0.0	1.0
ServerFormHandler	11054.0	-0.595712	0.759168	-1.0	-1.0	-1.0	-1.0	1.0
InfoEmail	11054.0	0.635788	0.771899	-1.0	1.0	1.0	1.0	1.0
AbnormalURL	11054.0	0.705446	0.708796	-1.0	1.0	1.0	1.0	1.0
WebsiteForwarding	11054.0	0.115705	0.319885	0.0	0.0	0.0	0.0	1.0
StatusBarCust	11054.0	0.762077	0.647516	-1.0	1.0	1.0	1.0	1.0
DisableRightClick	11054.0	0.913877	0.406009	-1.0	1.0	1.0	1.0	1.0
UsingPopupWindow	11054.0	0.613353	0.789845	-1.0	1.0	1.0	1.0	1.0
IframeRedirection	11054.0	0.816899	0.576807	-1.0	1.0	1.0	1.0	1.0
AgeofDomain	11054.0	0.061335	0.998162	-1.0	-1.0	1.0	1.0	1.0
DNSRecording	11054.0	0.377239	0.926158	-1.0	-1.0	1.0	1.0	1.0
WebsiteTraffic	11054.0	0.287407	0.827680	-1.0	0.0	1.0	1.0	1.0
PageRank	11054.0	-0.483626	0.875314	-1.0	-1.0	-1.0	1.0	1.0
GoogleIndex	11054.0	0.721549	0.692395	-1.0	1.0	1.0	1.0	1.0
LinksPointingToPage	11054.0	0.343948	0.569936	-1.0	0.0	0.0	1.0	1.0
StatsReport	11054.0	0.719739	0.694276	-1.0	1.0	1.0	1.0	1.0
class	11054.0	0.113986	0.993527	-1.0	-1.0	1.0	1.0	1.0

Fig. 2. Description of the Dataset

B. Data Visualization

Based on the results of data visualization using heatmap correlation, it can be seen that the relationship between features in the phishing URL dataset is in the form of correlation values. Heatmaps are used to determine the level of relationships between variables so that they can help understand data patterns before the machine learning model training process is carried out. In Figure. 3, it can be seen that lighter colors indicate a higher correlation, while dark colors indicate a lower

relationship between features. From the visualization results, it can be seen that most of the features have low to moderate correlation, which shows that each feature provides different information and does not have high multicollinearity in the dataset. Some features show a positive correlation to phishing classification labels, such as URL features related to domains, redirects, and website traffic, while others have a negative correlation to class. The results of this visualization show that the dataset has a good enough distribution of features to be used in the classification and performance analysis process of various phishing URL detection algorithms.

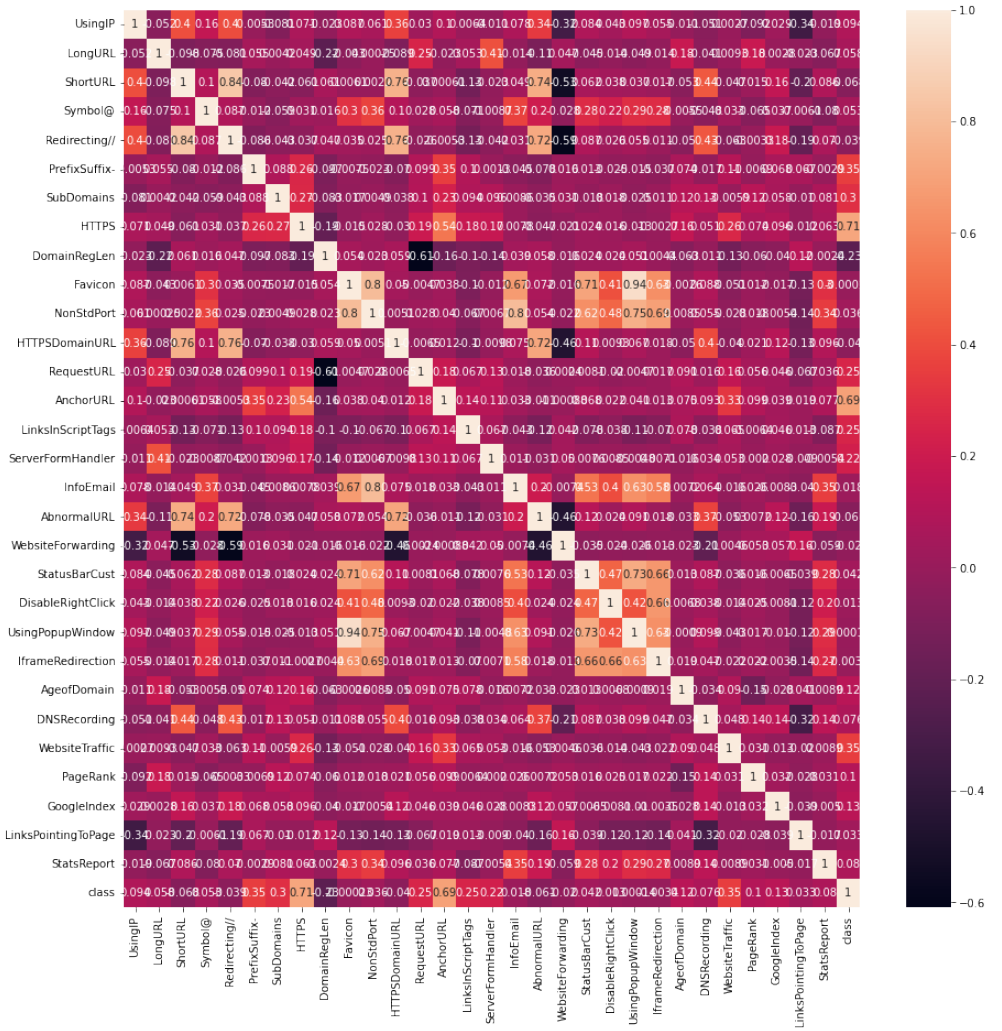

Fig. 3. Data Visualization

C. Model Comparison

The model comparison stage was carried out to compare the performance of all classification algorithms used in phishing URL detection research. Comparisons were made using several evaluation metrics, namely accuracy, precision, recall, and F1-score, to determine the ability of each model to accurately identify phishing and legitimate URLs. Through this process, it is possible to determine the algorithm that has the best performance and the most effective model to be applied to the phishing detection system based on Machine Learning and Cybersecurity. The results obtained can be seen in the table. 2.

Table. 2. Model Comparison Results

No	*Model*	*Accuracy*	*F1_Score*	*Recall*	*Precision*
1	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
2	CatBoost Classifier	0.972	0.975	0.994	0.989
3	Multi Layer Perceptron	0.971	0.974	0.992	0.985
4	XGBoost Classifier	0.969	0.973	0.993	0.984
5	Random Forest	0.967	0.970	0.992	0.991
6	Support Vector Machine	0.964	0.968	0.980	0.965
7	Decision Tree	0.961	0.965	0.991	0.993
8	K-Nearest Neighbours	0.956	0.961	0.991	0.989
9	Logistic Regression	0.934	0.941	0.943	0.927
10	Naïve Bayes Classifier	0.605	0.454	0.292	0.997

Table 2 shows that the results of the classification model test indicate that the Gradient Boosting algorithm produced the best performance with an accuracy value of 0.974, an F1-score of 0.977, a recall of 0.994, and a precision of 0.986. These results show that the model is capable of detecting phishing URLs with a very high level of accuracy and sensitivity. In addition, CatBoost also showed excellent performance with an accuracy of 0.972, an F1-score of 0.975, a recall of 0.994, and a precision of 0.989. The Perceptron Multilayer model obtained an accuracy of 0.971 with an F1-score of 0.974, recall of 0.992, and a precision of 0.985, indicating that the deep learning approach is able to effectively recognize phishing patterns in the URL dataset. Furthermore, Extreme Gradient Boosting obtained an accuracy of 0.969, an F1-score of 0.973, a recall of 0.993, and a precision of 0.984. Almost similar results were also obtained by Random Forest with an accuracy of 0.967, an F1-score of 0.970, a recall of 0.992, and a precision of 0.991. Meanwhile, the Support Vector Machine obtained an accuracy of 0.964, an F1-score of 0.969, a recall of 0.980, and a precision of 0.985. The Decision Tree model also showed good performance with an accuracy of 0.961, an F1-score of 0.965, a recall of 0.991, and a precision of 0.993. In the K-Nearest Neighbor algorithm, an accuracy of 0.956, an F1-score of 0.961, a recall of 0.991, and a precision of 0.989 were obtained. Meanwhile, Logistic Regression produced an accuracy of 0.934 with an F1-score of 0.941, a recall of 0.943, and a precision of 0.927. The lowest results were obtained by Naive Bayes with an accuracy of 0.605, an F1-score of 0.454, a recall of 0.292, and a precision of 0.997. The low performance of Naive Bayes indicates that the assumption of independence between features is not in accordance with the characteristics of URL phishing datasets that have complex relationships between variables.

D. Best Model Results

Based on the results of the performance evaluation of all classification algorithms, it was found that some models showed excellent ability to detect phishing URLs. The comparison process is carried out using accuracy, precision, recall, and F1-score metrics to determine the model with the most optimal performance. The test results showed that the ensemble learning and boosting-based algorithm provided more stable and accurate classification results than other methods, so the model with the best performance was chosen as the best model in this study. For the best model results, **see** Figure. 4.

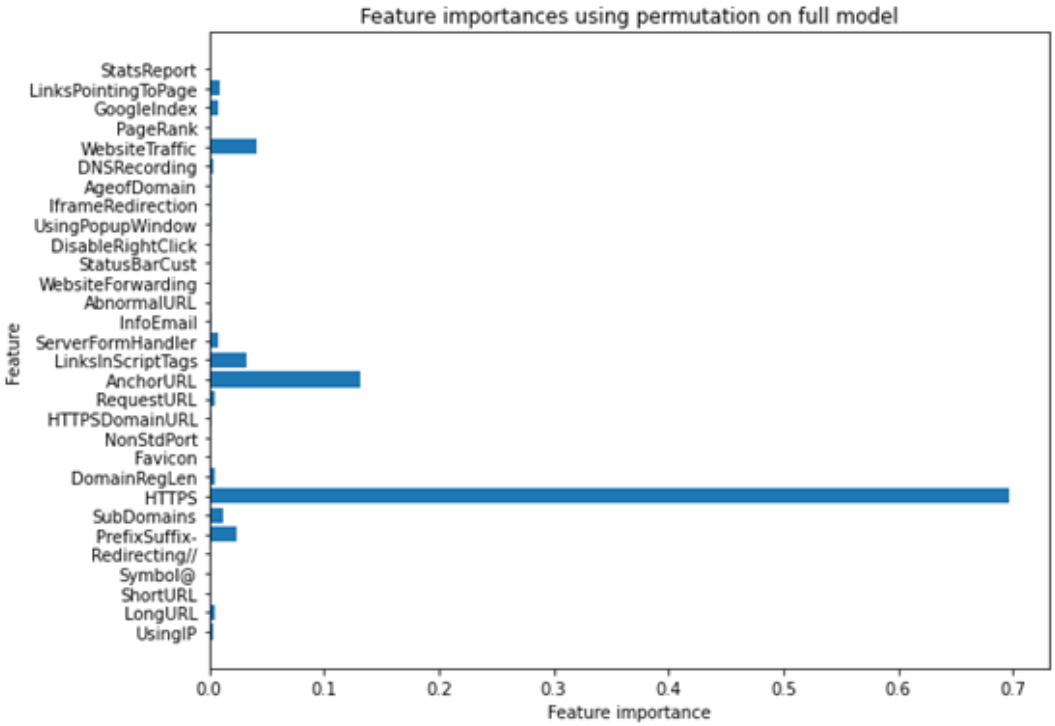

Fig. 4. Feature Importance Full Model

Figure. 4 shows that the best model results visualization results with the Gradient Boosting model have the best performance compared to other algorithms in detecting phishing URLs. This is shown by the highest evaluation scores on several test metrics such as accuracy, recall, precision, and F1-score. The horizontal bar graph clearly shows the comparison of performance between algorithms, where ensemble learning-based models have a better predominance of results than traditional classification methods. In addition to Gradient Boosting, the CatBoost, Extreme Gradient Boosting, and Random Forest algorithms also show high and stable performance. Meanwhile, some models, such as Naive Bayes, have much lower performance than others. The results of this visualization show that the boosting and ensemble learning methods are more effective in recognizing complex patterns in the URL phishing dataset, so that they are able to produce a more accurate and consistent classification level.

IV. Conclusion

Based on the results of the research that has been conducted, it can be concluded that the application of Machine Learning and Deep Learning can detect phishing URLs with a high level of accuracy. The results of the dataset analysis show that several features, such as HTTPS, AnchorURL, and WebsiteTraffic, have an important influence on the process of classifying phishing and legitimate URLs because these features are able to represent the security characteristics and behavior patterns of the website used in phishing attacks. In addition, the results of visualization and correlation analysis show that the dataset has good data quality without missing values or outliers, thus supporting the classification model training process optimally. Based on the results of the model performance evaluation, the Gradient Boosting algorithm became the best model with an accuracy value of 0.974, an F1-score of 0.977, a recall of 0.994, and a precision of 0.986. These results show that the model is able to classify phishing URLs very well with a low error rate. The high recall value indicates that most phishing URLs are detected correctly, which can reduce the risk of accessing malicious websites and minimize the possibility of malware and theft of user data. Thus, the Gradient Boosting algorithm can be used as an effective method for the development of artificial intelligence-based phishing detection systems in modern cybersecurity environments.

References

[1] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," *IEEE Access*, vol. 11, no. March, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[2] M. U. Javeed, S. M. Aslam, H. A. Sadiqa, A. Raza, M. M. Iqbal, and M. Akram, "Phishing Website URL Detection Using a Hybrid Machine Learning Approach," *J. Comput. Biomed. Informatics*, vol. 9, no. 1, 2025.

[3] I. Kara, M. Ok, and A. Ozaday, "Characteristics of Understanding URLs and Domain Names Features: The Detection of Phishing Websites with Machine Learning Methods," *IEEE Access*, vol. 10, no. December, pp. 124420–124428, 2022, doi: 10.1109/ACCESS.2022.3223111.

[4] S. Ariyadasa, S. Fernando, and S. Fernando, "Combining Long-Term Recurrent Convolutional and Graph Convolutional Networks to Detect Phishing Sites Using URL and HTML," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3196018.

[5] S. H. Ahammad *et al.*, "Phishing URL detection using machine learning methods," *Adv. Eng. Softw.*, vol. 173, no. August, p. 103288, 2022, doi: 10.1016/j.advengsoft.2022.103288.

[6] M. M. Alani and H. Tawfik, "PhishNot: A Cloud-Based Machine-Learning Approach to Phishing URL Detection," *Comput. Networks*, vol. 218, no. January, p. 109407, 2022, doi: 10.1016/j.comnet.2022.109407.

[7] S. R. Abdul Samad *et al.*, "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," *Electron.*, vol. 12, no. 7, 2023, doi: 10.3390/electronics12071642.

[8] 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.

[9] S. Alnemari, "Detecting Phishing Domains Using Machine Learning," *Early Writings on India*, pp. 124–134, 2018, doi: 10.4324/9781315232140-14.

[10] S. Asiri, Y. Xiao, S. Alzahrani, and T. Li, "PhishingRTDS: A real-time detection system for phishing attacks using a Deep Learning model," *Comput. Secur.*, vol. 141, p. 103843, 2024, doi: https://doi.org/10.1016/j.cose.2024.103843.

[11] P. M. Bhagat, S. Waghmare, M. Waje, R. Patil, K. Joshi, and M. Bakuli, "Feature Classification and Extreme Learning Machine Based Detection of Phishing Websites," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, 2023, doi: 10.17762/ijritcc.v11i8s.7182.

[12] U. Zara, K. Ayyub, H. Ullah Khan, A. Daud, T. Alsahfi, and S. Gulzar Ahmad, "Phishing Website Detection Using Deep Learning Models," *IEEE Access*, vol. 12, no. November, pp. 167072–167087, 2024, doi: 10.1109/ACCESS.2024.3486462.

[13] Y. Wei and Y. Sekiya, "Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3224781.

[14] E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, "A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators," *Sensors*, vol. 23, no. 9, 2023, doi: 10.3390/s23094403.

[15] Z. Muthi'ah, Oktalia Triananda Lovita, Oktalia Triananda Lovita, and Mohd Iqbal Muttaqin, "Deep Learning Based Classification of TB and Normal Chest X-rays Using a Custom CNN with Minimal Epoch Training," *J. Inotera*, vol. 10, no. 2, pp. 272–279, 2025, doi: 10.31572/inotera.vol10.iss2.2025.id503.

[16] M. N. Yakubu and A. M. Abubakar, "Applying machine learning approach to predict students' performance in higher educational institutions," *Kybernetes*, vol. 51, no. 2, pp. 916–934, 2021, doi: 10.1108/K-12-2020-0865.

[17] C. Thapa *et al.*, "Evaluation of Federated Learning in Phishing Email Detection," *Sensors*, vol. 23, no. 9, 2023, doi: 10.3390/s23094346.

[18] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, "An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Syst. Appl.*, vol. 165, p. 113863, 2021, doi: https://doi.org/10.1016/j.eswa.2020.113863.

[19] Jupron and Sutrisno, "Analysis of Heart Disease Using the Random Forest Method," *J. Inotera*, vol. 10, no. 1, pp. 167–174, 2025, doi: 10.31572/inotera.vol10.iss1.2025.id452.

[20] M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Inf. Secur.*, vol. 17, no. 3, pp. 423–440, 2023, doi: 10.1049/ise2.12106.

[21] F. Nthurima, A. Mutua, and W. Stephen Titus, "Detecting Phishing Emails Using Random Forest and AdaBoost Classifier Model," *Open J. Inf. Technol.*, vol. 6, no. 2, 2023, doi: 10.32591/coas.ojit.0602.03123n.

[22] 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, 2023, doi: 10.3390/s23198070.