Jurnal Teknik Informatika (JUTIF) Vol. 6, No. 2, April 2025, Page. 691-708
P-ISSN: 2723-3863 https://jutif.if.unsoed.ac.id
E-ISSN: 2723-3871 DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4344

Comparison of SVM and Gradient Boosting with PCA for Website Phising Detection

Nur Aini Syam*[1], Nurhikma Arifin[2], Wawan Firgiawan[3], Muhammad Furqan Rasyid[4]

[1,2,3]Informatics, Universitas Sulawesi Barat, Indonesia
[4]Information Science, Nara Institute of Science and Technology, Japan

Email: [1]nurainisyam822@gmail.com

Received: Jan 22, 2025; Revised: Feb 18, 2025; Accepted: Feb 20, 2025; Published: Apr 26, 2025

Abstract

The increasing use of the internet has led to a rise in phishing attacks, posing a threat to user data security. This study compares the performance of the Support Vector Machine (SVM) and Gradient Boosting algorithms, integrated with Principal Component Analysis (PCA) for dimensionality reduction, in classifying phishing websites. The dataset consists of 11,054 samples classified into two categories: phishing (1) and non-phishing (-1), with three data partition scenarios for training and testing: 70:30, 80:20, and 90:10. Experimental results indicate that SVM outperforms Gradient Boosting in terms of accuracy and recall, particularly in detecting phishing websites. In the 80:20 and 70:30 data partition scenarios, the SVM model achieved an accuracy of 96% to 97% and had a higher recall for phishing websites, making it more sensitive to phishing detection. However, Gradient Boosting demonstrated consistent performance with an accuracy of around 94%, providing a balanced result between precision and recall for both classes. Therefore, the SVM model is superior for phishing detection tasks requiring high sensitivity to phishing websites, while Gradient Boosting remains a viable alternative when a more balanced performance between phishing and non-phishing sites is needed. The study concludes that both algorithms can be effectively used for phishing detection, with potential improvements through further experiments and hyperparameter tuning.

Keyword : *Classification, Gradient Boosting, Phishing Website, Principal Component Analysis, Support Vector Machine.*

1. INTRODUCTION

The swift progress of techology has made it more challenging for people to separate themselves from the internet and digital tools [1]. According to the Central Statistics Agency (BPS), 62.10% of Indonesia's population had accessed the internet by 2021 [2]. Meanwhile, the Indonesian Internet Service Providers Association (APJII) reported that during the 2023-2024 period, Indonesia had approximately 221.56 million internet users, with a penetration rate of 79.5% of the total population [3]. However, the increasing number of internet users also raises risks to user data security. User data becomes vulnerable to theft by malicious actors through various threats, one of which is phishing [4]. Phishing is a method that manipulates targets to acquire sensitive data, such as username, passwords, or credit card details, by pretending to be a trustworthy entity legitimateentities. Typically, phishing perpetrators direct victims to fake websites via URL links [4],[5].

According to the Kaspersky Network Report, phishing attacks emerged as the primary cybersecurity threat in Indonesia in 2021, with 1.6 million attacks detected. Phishing also ranked as the most significant threat across Southeast Asia, with Indonesia topping the list. Several factors contributing to the rise of phishing attacks in Indonesia include the growing number of internet users, low awareness and cybersecurity practices, weak infrastructure, and the large portion of internet users who remain poorly educated about phishing practices [6].

Various methods have been designed to recognize and combat phishing attacks, including blacklisting and feature extraction. However, these approaches have limitations in addressing new and dynamic phishing attacks. As a solution, machine learning-based techniques have proven effective in detecting phishing websites with high accuracy [7],[8].

The research conducted by Dr. M. Prasad and Ansifa Kouser M has implemented a Gradient Boosting Classifier-based approach to effectively detect phishing websites. This approach focuses on significant aspects of the URL, aiming to provide a real-time phishing detection solution with high accuracy and a low false positive rate. The results of the study show that the applied system achieved a detection accuracy of 97%, with a very low false positive rate. This approach has proven effective in detecting phishing websites in real-time by utilizing URL characteristics to distinguish between legitimate sites and phishing sites [9]. The same research was conducted by Kamal Omari using the Gradient Boosting model to detect phishing domains. With an accuracy of 0.972, this model demonstrates a high level of precision in its predictions. An F1_score of 0.969 indicates a good balance between precision and recall, with a recall value of 0.970 emphasizing the model's ability to accurately identify phishing. Additionally, a precision of 0.968 reflects a low false positive rate, and the ROC-AUC score of 0.996 demonstrates the model's resilience in distinguishing between phishing and non-phishing sites [10]. In addition, Rizal Dwi Prayogo, et al., used the gradient-boosting method with hyperparameter optimization on three phishing website datasets. The proposed approach successfully achieved high accuracy rates of 97.45%, 99.16%, and 97.85% on the UCI (2015), Mendeley (2018), and Mendeley (2020) datasets. The three most influential features in phishing detection were length_url, directory_length, and time_domain_activation [11].

Phishing website detection has also been studied using other algorithms, such as Support Vector Machine (SVM), as demonstrated by Wahyudi Diki, et al., whose tests showed the best accuracy of 85.71% using a polynomial kernel SVM with degree 9 and C 2.5 [12]. Emmanuel Song Shombot, et al., predicted phishing attacks by comparing the polynomial and radial basis function (RBF) of the support vector machine (SVM). The results showed that the polynomial kernel performed better with an accuracy of 84.5% compared to 82.6% for RBF [13]. The same algorithm was used by Sagnik Anupam, et al., to classify phishing websites using Support Vector Machine along with four nature-inspired optimization algorithms, namely Bat Algorithm, Firefly Algorithm, Grey Wolf Optimiser (GWO), and Whale Optimization Algorithm. The research found that the GWO algorithm provided the best performance [14].

Based on related research, the utilization of machine learning algorithms such as Support Vector Machine (SVM) and Gradient Boosting is highly relevant for improving phishing website detection performance. These algorithms offer different approaches to data classification and have the potential to achieve optimal accuracy. However, one of the main challenges in applying machine learning is the high data dimensionality, which can reduce model efficiency and accuracy. To address this issue, Principal Component Analysis (PCA) can be used as a dimensionality reduction technique. PCA simplifies the data while retaining relevant information, thereby speeding up model training without sacrificing performance. The application of PCA is also expected to reduce overfitting, which often occurs in machine learning models, especially when dealing with large datasets that have correlated features. By reducing data dimensionality, PCA can help improve model generalization and produce more stable and reliable predictions, as evidenced in previous research [15],[16].

Although various studies have examined the use of machine learning in phishing detection, research specifically evaluating the application of PCA to models such as Gradient Boosting and SVM

DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4344

remains limited. Most previous studies have focused solely on models without dimensionality reduction, thus failing to explicitly analyze the impact of PCA on improving efficiency and accuracy. Therefore, this study aims to compare the performance of Gradient Boosting and SVM in detecting phishing websites with the application of PCA. The evaluation is conducted to measure the effect of dimensionality reduction on accuracy, processing efficiency, and the model's ability to avoid overfitting.

Thus, this study is expected to make a significant contribution to the development of a more optimal and reliable phishing detection system, as well as provide new insights into how PCA can enhance the performance of machine learning models in detecting phishing threats.

2. METHOD

This study utilizes the Principal Component Analysis (PCA) dimensionality reduction method to simplify the dataset without losing significant information and employs Gradient Boosting and Support Vector Machine (SVM) algorithms for the classification process. The dimensionality reduction process aims to enhance computational efficiency while maintaining data quality for further analysis. The results of this dimensionality reduction will be used as input for the Gradient Boosting and Support Vector Machine (SVM) algorithms in classification. This research was conducted using software such as Python with libraries including Scikit-Learn, Pandas, and NumPy for data processing, as well as Matplotlib and Seaborn for result visualization. The model training process was carried out using Jupyter Notebook or Google Colab to support interactive experiments. In terms of hardware, the experiments were conducted on a computer with Windows 10 Pro 64-bit (10.0, Build 19045) as the operating system, an Intel(R) Core (TM) i5-4300U CPU @ 1.90GHz 2.49GHz processor, and 4.00 GB of installed RAM.

The methodological steps employed are illustrated in Figure 1.

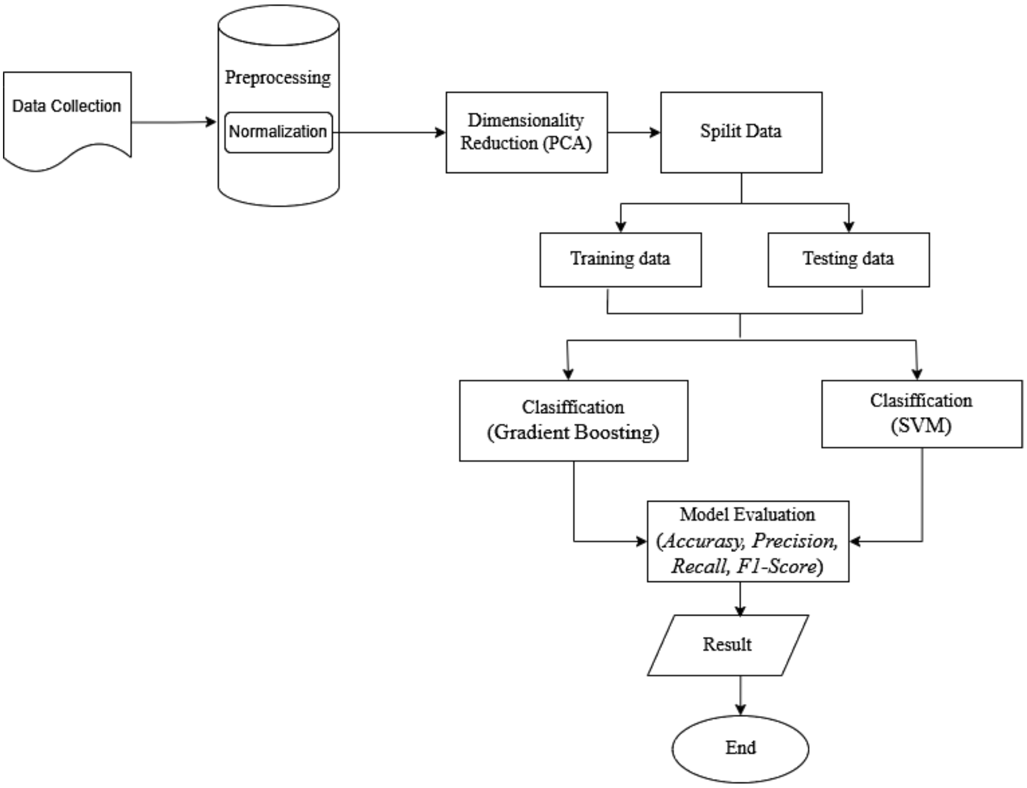
Data Collection
Preprocessing
Normalization
Dimensionality Reduction (PCA)
Spilit Data
Training data
Testing data
Clasiffication (Gradient Boosting)
Clasiffication (SVM)
Model Evaluation (*Accurasy, Precision, Recall, F1-Score*)
Result
End

Figure 1. Methodological Steps Used in This Study

The detailed explanation of the methodological steps used is as follows:

2.1. Data Collection

The dataset used in this study was sourced from https://www.kaggle.com/. It consists of 32 attributes with a total of 11,054 data entries and is classified into two classes: 1 and -1, where 1 represents phishing and -1 represents non-phishing. The dataset is stored in Excel format, specifically as a CSV file. The attributes in the data, as shown in Table 1, are categorized as 0 (suspicious), 1 (phishing), and -1 (non-phishing).

Table 1. Data Attribute

Attribute Name	Category Explanation
Using the ip address	1 domain part if it has an ip address -1 if it doesn't use an ip
Long Url	1 if length >75. 0 if length ≥ 54 and ≤ 75. -1 if length <54.
ShortURL	1 using shortener service -1 not using shortener service
URL's Having "@" Symbol	1 url has @ symbol after domain name -1 does not use
Redirecting "//"	1 if the last occurrence position of "//" >7 -1 last occurrence position of "//" ≥ 6 ≤7
PrefixSuffix (-)	1 if the domain name part includes the (-) symbol -1 if not using
Sub Domains	1 if the point in the domain=3. 0 if the point in the domain=2. -1 if the point in the domain=1.
HTTPS	1 if not using https, trusted issuer and certificate age ≥1 year. 0 if using https and untrusted issuer -1 if using https
Domain Registration Length	1 if domain expired ≤ 1 year -1 if domain expired >1 year
Favicon	1 if the favicon is loaded from an external domain -1 if it is loaded from the same favicon
Using Non- Standard Port	1 if the port is a non standard port -1 if the port is standar
HTTPS Domain URL	1 if using https in the domain part of the url -1 if not
Request URL	1 if >61%. 0 if ≥22% and ≤61%. -1 if request url <22%.
URL of Anchor	1 if >67%. 0 if ≥31% and ≤67%. -1 if anchor url<31%.
LinksIn Script Tags	1 if >81%. 0 if ≥17% and ≤81%. -1if number of links <17%
Server Form Handler (SFH)	1 if sfh is "about blank" or empty 0 if sfh refers to a different domain. -1 otherwise
Submitting Information to Email	1 if using "email" or "mailto:" to send user information. -1 otherwise
Abnormal URL	1 if hostname is not included in url -1 if included
Website Forwarding	0 if diverted ≥2 and ≤4. 1 if diverted >5
Status Bar Customization	1 if using one mouse over to change the status bar. -1 if not
Disabling Right Click	1 if right click is disabled. -1 fi not

Attribute Name	Category Explanation
Using pop-up Window	1 if the pop-up window contains a text field -1 if not
IFrame Redirection	1 if using iframe. -1 if not using Ifrem
Age of Domain	-1 if domain age ≥ 6 months. 1 if < 6 months
DNS Record	1 if you don't have a dns record for the domain. -1 if you have a dns record
Website Traffic	-1 if website rank <100.000. 0 if website rank >100.000. 1 if not rank
PageRank	1 if pagerank ≤0,2. -1 if >0,2
Google Index	-1 if the web page is indexed by google. 1 if not
Links Pointing to Page	-1 if the link leads to a web page =0. 0 if the link leads to a web page >0 ≤2. 1 if >2
Statistical- Reports Based Feature	1 if host belongs to top phishing ip or domain -1 otherwise
Class	1 and -1 where 1 indicates a safe website and -1 phishing website.

2.2. Preprocessing

The preprocessing stage plays a significant role in improving the performance of machine learning algorithms [17]. Data preprocessing is expected to produce an optimal data condition for further processing [18]. There are various data preprocessing techniques, one of which is data normalization [17]. This study employs data normalization as part of the preprocessing stage. Among the many normalization methods, the Z-Score method is commonly used. Z-Score normalization, also known as standardization, is a statistical technique used to standardize the data scale so that it has a mean of 0 and a standard deviation of 1 [19]. Its primary goal is to enhance data comparability, particularly when there are significant scale differences between data attributes [20]. This method is highly beneficial in various data analysis and machine learning applications, where model performance can be affected by scale differences among attributes [21].

$$z = \frac{X - \mu}{\sigma} \tag{1}$$

where Z are Z score for each data value, X are data value to be transformed, μ are mean of the data and σ are standard deviation of the data.

The Standardization steps used in this study is as follows:

$$Data = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

a. Calculate the Mean (μ) for each feature (column):

$$\mu_1 = \frac{0 + 1 - 1}{3} = 0$$

$$\mu_2 = \frac{1 + 0 - 1}{3} = 0$$

$$\mu_3 = \frac{-1 - 1 + 0}{3} = 0$$

b. Calculate the Standard Deviation (σ) for each feature (column):

$$\sigma_1 = \sqrt{\frac{(0-0)^2 + (1-0)^2 + (-1-0)^2}{3}} = \sqrt{\frac{2}{3}} \approx 0.8165$$

$$\sigma_2 = \sqrt{\frac{(1-0)^2 + (0-0)^2 + (-1-0)^2}{3}} = \sqrt{\frac{2}{3}} \approx 0.8165$$

$$\sigma_3 = \sqrt{\frac{(-1-0)^2 + (1-0)^2 + (-0-0)^2}{3}} = \sqrt{\frac{2}{3}} \approx 0.8165$$

c. Apply Standardization using the Z-Score formula in equation 1:

$$z_1 = \frac{0-0}{0.8165} = 0$$
$$z_2 = \frac{1-0}{0.8165} = 1.2247$$
$$z_3 = \frac{-1-0}{0.8165} = -1.2247$$

d. Data After Standardization:

$$Standardized\ Data = \begin{bmatrix} 0 & 1.2247 & -1.2247 \\ 1.2247 & 0 & 1.2247 \\ -1.2247 & -1.2247 & 0 \end{bmatrix}$$

2.3. *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) is a statistical method used to reduce the dimensionality of complex multivariate data. This method works by transforming the existing variables into a new set of variables that are uncorrelated, known as principal components (PCs). The primary goal of PCA is to identify the direction or linear combination of the original variables that has the largest variance [22],[23]. Below is the formula for PCA [21].

$$C = \frac{1}{m-1} X^T X \qquad (2)$$

where ***C*** are covariance matrix (dimension m×n), ***X*** are standardized data matrix (dimension m×n), and $\boldsymbol{X^T}$ are transpose of the data matrix.

The results of PCA reduction in this study are shown in Figure 2.

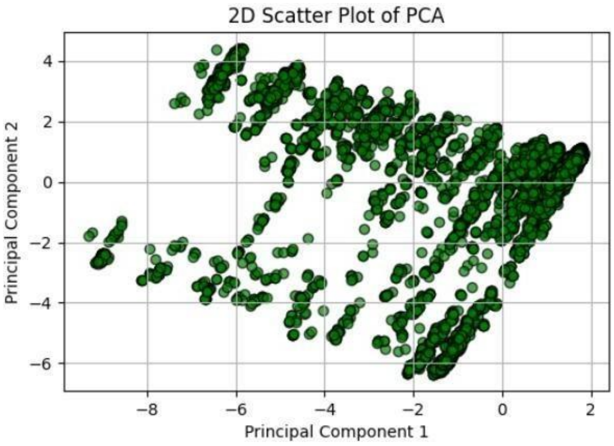

Figure 2. 2D Scatter Plot of PCA

Vol. 6, No. 2, April 2025, Page. 691-708
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.2.4344

Based on Figure 1, the plot showing data in two main dimensions, Principal Component 1 (PC1) on the X-axis and Principal Component 2 (PC2) on the Y-axis, aims to reduce the data's dimensions for easier understanding of its patterns and distribution. Areas with denser points indicate higher data concentration, while patterns or clusters reveal the underlying structure or grouping within the data. PCA helps simplify the visualization of complex data, retaining important information, and making analysis easier, so patterns that were previously hard to detect become clearer and more comprehensible. After applying PCA, the reduced data will be analyzed using the Gradient Boosting and Support Vector Machine (SVM) algorithms for classification. These techniques will help to further improve the classification accuracy and computational efficiency by using the transformed data from PCA as input.

2.4. Spilitting data

After preprocessing and dimensionality reduction using PCA, the data is divided into two main parts: training data and testing data. The training data is used to train the model to recognize patterns based on the reduced features, while the testing data is used to evaluate the model's performance with unseen data. This division ensures that the model learns effectively and is fairly tested to assess its accuracy in making predictions.

The data is divided into three ratios, as shown in Table 2 below.

Table 2. Spilitting Data

	Training data testing data composition
1	70:30
2	80:20
3	90:10

Based on Table 2, the data is divided into three partitioning schemes with different ratios between training and testing data. The first scheme uses a 90:10 ratio, where 90% of the total data is used to train the model, while the remaining 10% is used to evaluate its performance. The second scheme splits the data with an 80:20 ratio, meaning 80% of the data is allocated for training and 20% for testing. Meanwhile, the third scheme adopts a 70:30 ratio, with 70% of the data used for training and the remaining 30% for testing. This data partitioning with various ratios aims to assess how the proportion of training and testing data affects the model's accuracy and generalization in detecting phishing websites.

2.5. Classification using Gradient Boosting and Support Vector Machine

In this study, the classification methods Gradient Boosting and Support Vector Machine will be compared based on their results. The following describes how the algorithms work:

2.5.1. Gradient Boosting

Gradient Boosting (GB) is a highly effective ensemble method in machine learning, designed to build predictive models incrementally by reducing errors at each step. This technique enables the optimization of differentiable loss functions, offering high flexibility for various problem types. The core concept of Gradient Boosting lies in progressively combining multiple simple models, such as decision trees, to create a more robust and accurate predictive model. Here are the steps in building a Gradient Boosting Machine (GBM) [24]:

a. Initialization of the Initial Model: Begin with a simple model that makes an initial estimation of the target values. This initial prediction is often the average of the target values:

$$F(x) = \arg\min \Sigma\, i \;\; 1^{N} L(y_i, c) \qquad (3)$$

where L is the loss function, y_i are the actual target values, and c is a constant.

b. Error Identification: Evaluate the performance of the initial model by computing the errors (differences) between the target and predicted values:

$$rim = y_i - F_{m-1}(x_i) \qquad (4)$$

where are the residuals (errors) at iteration m, y_i are the actual values, and $F_{m-1}(x_i)$ are the predictions from the model at the previous iteration.

c. Training a New Model: create a new simple model $(h_m(x))^n$ to predict the identified errors. Fit this new model to the residuals and determine the optimal weight p_m for the latest model:

$$p_m = \text{argmin}\, p \sum_{i=1}^{N} L\,(y_i, F_{m-1}(x_i) + p h_m(x_i)) \qquad (5)$$

d. Update the original model: Refine the original model by incorporating the predictions of the new model.

$$F_m(x) = F_{m-1}(x) + p_m h_m(x) \qquad (6)$$

e. Repeat the process: continuously repeat the cycle of calculating errors, building new models to predict those errors, and updating the model. Each iteration progressively improves the accuracy of the model.

In this study, the Gradient Boosting model was used with parameters max_depth = 4 and learning_rate = 0.7. The max_depth parameter determines the maximum depth of each tree in the ensemble, where a value of 4 is used to prevent overfitting while maintaining sufficient complexity to capture patterns in the data. Meanwhile, the learning_rate controls how much each tree contributes to the final model, with a value of 0.7 chosen to accelerate convergence without losing generalization capability. These parameter choices aim to achieve a balance between accuracy and model efficiency.

2.5.2. Support Vector Machine

Support Vector Machine (SVM) is a machine learning method used to find a classification function that separates data into two distinct classes [25]. SVM is designed to handle classification problems due to its superior ability to generalize data compared to previous methods. This technique offers several advantages, such as a model that explicitly relies on a subset of specific data points and the use of support vectors, which makes the model easier to interpret [26].

In this study, the RBF kernel (Radial Basis Function) is used. The RBF kernel is a type of kernel function used in Support Vector Machine (SVM) to transform data from the original input space to a higher-dimensional space, where data that cannot be separated linearly in the input space can be separated linearly in the higher-dimensional space. The RBF kernel works by calculating the distance between data points and the center of the function, where the closer the data points are to the center, the larger the kernel value. The advantage of the RBF kernel is its ability to handle complex data with non-linear distributions, making it widely used in various SVM applications. RBF kernel function is [13]:

$$k(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \qquad (7)$$

where $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are input vectors, ||.|| denotes the Euclidean distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, and γ is a hyperparameter that controls the width of the kernel.

This kernel function transforms the input data into a higher-dimensional feature space,where it can be separated by a linear decision boundary. The RBF kernel is commonly employed in support vector machines because of its capacity to manage complex,nonlinear decision boundaries in the data.

In this study, the SVM model with the RBF kernel was used with parameters C = 1 and gamma = 0.1 The C parameter controls the penalty for classification errors,where a higher value can improve accuracy but risks overfitting.Meanwhile,gamma determines the range of influence of each sample,where a higher value makes the model more sensitive to patterns in the data.These parameter choices aim to achieve a balance between accuracy and model generalization.

2.6. Model Evaluation

The model in this study is evaluated using accuracy,precision,recall,and F1-score metrics,which are presented in the form of a confusion matrix.These metrics are used to measure the model's performance in detecting phishing websites.The confusion matrix is a table that compares the model's prediction results with the actual data,as shown in Table 3.

Table 3. *Confusion Matrix*

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Based on the confusion matrix above,the evaluation metrics are calculated using the following formulas [27]:

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{8}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{9}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{10}$$

$$F1 - Score = \frac{2 * \mathrm{Persisi+Recall}}{\mathrm{Persisi+Recall}} \tag{11}$$

3. RESULT

Phishing website classification testing was conducted using three data splitting scenarios: 70:30, 80:20, and 90:10 from a total of 11,054 data points. This approach examines the impact of the training-to-testing ratio on model performance. The testing employs Gradient Boosting with PCA and SVM with PCA to compare their effectiveness in detecting phishing websites with dimensionality reduction.

3.1.1. Gradient Boosting

Pengujian Testing the Gradient Boosting method with PCA reduction was conducted using a learning rate of 0.7 and a max_depth of 4. The learning rate of 0.7 regulates the contribution of each decision tree to the overall model at each iteration, aiming to balance the speed of learning and accuracy. Meanwhile, the max_depth of 4 limits the depth of the decision trees to prevent excessive complexity, thereby avoiding overfitting while still allowing the model to capture relevant patterns in the data. These parameters were chosen to ensure the model learns effectively without losing its generalization capability. The testing results using the Gradient Boosting algorithm with PCA reduction under the 70:30, 80:20, and 90:10 scenarios are presented in Tables 4–9.

Table 4. Confusion Matrix Results 70:30

Class		Prediction Labels	
		-1	1
True Label	-1	1332	123
	1	87	1775

Table 4 presents the test results using the 70:30 data split scenario. Out of a total of 3,317 samples, the model successfully classified most of the data correctly. For the non-phishing class (-1), out of 1.455 tested samples, 1.332 were correctly classified, while 123 were misclassified as phishing (1). Meanwhile, for the phishing class (1), out of 1,862 tested samples, 1.175 were correctly identified, while 87 were misclassified as non-phishing (-1). These results indicate that the model has high accuracy, although some misclassifications still occur.

Table 5. Accuracy Results of Gradient Boosting with PCA 70:30

Class	Precision	Recall	F1-Score	Support
-1	0.94	0.92	0.93	1455
1	0.94	0.95	0.94	1862
Accuracy			0.94	3317
Macro Avg	0.94	0.94	0.94	3317
Weighted Avg	0.94	0.94	0.94	3317

Table 5 presents the accuracy results of the Gradient Boosting model with PCA using a 70:30 data split. Based on the results in Table 4, the model achieved a precision of 0.94 for both classes (-1 and 1), reflecting high accuracy in predicting each class. The recall for class -1 was 0.92, while for class 1, it was 0.95, indicating that the model is more effective in detecting data from class 1 compared to class -1. This suggests that the model tends to recognize positive samples better than negative ones. The F1-Score, which balances precision and recall, showed excellent results, with 0.93 for class -1 and 0.94 for class 1. The overall accuracy of the model reached 94%, meaning that out of a total of 3,317 samples, 3,118 samples were correctly classified. Additionally, the macro and weighted averages for precision, recall, and F1-Score also achieved a score of 0.94. As shown in Table 4, this confirms that Gradient Boosting with PCA provides consistent and balanced performance across all classes.

Table 6. Confusion Matrix Results 80:20

Class		Prediction Labels	
		-1	1
True Label	-1	879	79
	1	55	1180

Based on Table 6, the testing results for the 80:20 data split scenario show that a total of 2,211 samples were tested. For the non-phishing class (-1), out of 976 tested samples, 879 were correctly classified, while 79 samples were misidentified as phishing (1). Meanwhile, for the phishing class (1), out of 1,235 tested samples, 1,180 were correctly classified, while 55 samples were misclassified as non-phishing (-1). These results indicate that the model has a good level of accuracy in classifying both classes.

Table 7. Accuracy Results of Gradient Boosting with PCA 80:20

Class	Precision	Recall	F1-Score	Support
-1	0.94	0.92	0.93	976
1	0.94	0.96	0.95	1235
Accuracy			0.94	2211
Macro Avg	0.94	0.94	0.94	2211
Weighted Avg	0.94	0.94	0.94	2211

As shown in Table 7, the accuracy results of the Gradient Boosting model with PCA using the 80:20 data split demonstrate stable performance. The model achieved a precision of 0.94 for both classes (-1 and 1), with a recall of 0.92 for class -1 and 0.96 for class 1. This indicates that the model is capable of correctly identifying most samples in each category. Additionally, the F1-Score, which represents the balance between precision and recall, reached 0.93 for class -1 and 0.95 for class 1. Overall, the model achieved an accuracy of 94%, with both macro and weighted average scores of approximately 0.94 for all metrics. As displayed in Table 6, these results suggest that the model performs consistently across all classes without exhibiting significant bias, despite differences in the number of samples.

Table 8. Confusion Matrix Results 90:10

Class		Prediction Labels	
		-1	1
True Label	-1	463	39
	1	26	578

As shown in Table 8, the testing results for the 90:10 data split scenario indicate that the model performs quite well. Out of a total of 1,106 tested samples, for the non-phishing class (-1), 502 samples were tested, with 463 correctly classified, while 39 samples were misclassified as phishing (1). Meanwhile, for the phishing class (1), out of 604 tested samples, 578 were correctly classified, while 26 samples were misidentified as non-phishing (-1). These results indicate that although some classification errors exist, the model consistently demonstrates high accuracy in detecting both classes.

Table 9. Accuracy Results of Gradient Boosting with PCA 90:10

Class	Precision	Recall	F1-Score	Support
-1	0.95	0.92	0.93	502
1	0.94	0.96	0.95	604
Accuracy			0.94	1106
Macro Avg	0.94	0.94	0.94	1106
Weighted Avg	0.94	0.94	0.94	1106

Table 9 presents the accuracy results of the Gradient Boosting model with PCA for the 90:10 data split scenario. As shown in Table 8, the model achieved a precision of 0.95 for the non-phishing class (-1) and 0.94 for the phishing class (1), demonstrating its ability to classify both categories effectively. The recall obtained was 0.92 for class -1 and 0.96 for class 1, as stated in Table 8, indicating the model's effectiveness in identifying phishing sites. Meanwhile, the F1-Score, which reflects the balance between precision and recall, was recorded at 0.93 for class -1 and 0.95 for class 1. The overall accuracy of the model reached 94%, with both the macro and weighted average scores also at 0.94, as summarized in

Table 8. These results indicate that the model maintains stable performance across both classes without significant bias.

3.1.2. Support Vector Machine

Phishing website classification using the Support Vector Machine (SVM) algorithm with PCA reduction has been performed using the Radial Basis Function (RBF) kernel with parameters gamma = 0.1 and C = 1. The C = 1 parameter indicates that the model balances between minimizing errors on the training data and maintaining a wide margin, thus reducing the risk of overfitting. Meanwhile, gamma = 0.1, which is relatively small, provides flexibility by giving more influence to data points that are farther away, helping the model recognize broader patterns in the data. The test results with data splitting scenarios of 70:30, 80:20, and 90:10 arepresented in Tables 10-14, reflecting the model's performance under different data configurations.

Table 10. Confusion Matrix Results 70:30

Class		Prediction Labels	
		-1	1
True Label	-1	1379	76
	1	49	1813

As shown in Table 10, the confusion matrix results for the 70:30 data split scenario indicate that the model performs well in classifying phishing and non-phishing websites. Out of a total of 3,317 tested samples, for the non-phishing class (-1), 1,455 samples were tested, with 1,379 samples correctly classified, while 76 samples were misclassified as phishing (1). Meanwhile, for the phishing class (1), out of 1,862 tested samples, 1,813 samples were correctly classified, whereas 49 samples were misclassified as non-phishing (-1). Although some misclassifications remain, these results demonstrate that the model is capable of accurately identifying most samples.

Table 11. Accuracy Results of SVM with PCA 70:30

Class	Precision	Recall	F1-Score	Support
-1	0.97	0.95	0.96	1455
1	0.96	0.97	0.97	1862
Accuracy			0.96	3317
Macro Avg	0.96	0.96	0.96	3317
Weighted Avg	0.96	0.96	0.96	3317

As shown in Table 11, the test results of the Support Vector Machine (SVM) model with PCA in the 70:30 data split scenario demonstrate good performance. The model achieved a precision of 0.97 for the non-phishing class (-1) and 0.96 for the phishing class (1), indicating its ability to classify both classes accurately. In terms of recall, the model obtained a score of 0.95 for class -1 and 0.97 for class 1, signifying its effectiveness in detecting phishing websites. The F1-score reached 0.96 for class -1 and 0.97 for class 1, reflecting a balance between precision and recall. Overall, the model achieved an accuracy of 96% with 3,317 tested samples. Additionally, the macro and weighted averages for all metrics were 0.96. These results indicate that the SVM model with PCA can detect phishing websites with high accuracy and stable performance.

Table 12. Confusion Matrix Results 80:20

	Class	Predicted Labels	
		-1	1
True Label	-1	930	46
	1	31	1204

Based on Table 12, the confusion matrix results for the 80:20 data split show that the model performs well in classifying samples. In the non-phishing class (-1), 976 samples were tested, with 930 samples correctly identified, while 46 samples were incorrectly categorized as phishing (class 1). Meanwhile, in the phishing class (1), 1,235 samples were tested, with 1,204 samples correctly classified, while 31 samples were misclassified as non-phishing (-1). These results indicate that the model can classify most samples with reasonable accuracy, although some misclassifications still occur in detecting the non-phishing class.

Table 13. Accuracy Results of SVM with PCA 80:20

Class	Precision	Recall	F1-Score	Support
-1	0.97	0.95	0.96	976
1	0.96	0.97	0.97	1235
Accuracy			0.97	2211
Macro Avg	0.97	0.96	0.96	2211
Weighted Avg	0.97	0.97	0.97	2211

Table 13 presents the accuracy results of SVM with PCA using an 80:20 data split. The model achieved a precision of 0.97 for the non-phishing class (-1) and 0.96 for the phishing class (1), demonstrating high accuracy in identifying both classes. The recall results obtained were 0.95 for the non-phishing class (-1) and 0.97 for the phishing class (1), indicating the model's effectiveness in detecting phishing incidents. The F1-score achieved was 0.96 for the non-phishing class (-1) and 0.97 for the phishing class (1), showing a good balance between precision and recall. The overall accuracy was 0.97, with consistent performance across all metrics, as reflected in the macro and weighted averages.

Table 14. Confusion Matrix Results 90:10

Class		Prediction Labels	
		-1	1
True Label	-1	477	25
	1	17	587

Based on Table 14, which presents the confusion matrix results for the 90:10 data split, the model demonstrates strong performance. For the non-phishing class (-1), out of 502 tested samples, the model correctly predicted 477 instances, while 25 samples were misclassified as phishing (1). For the phishing class (1), out of 604 tested samples, the model correctly predicted 587 instances, with 17 misclassified as non-phishing (-1). These results indicate that the model performs well in distinguishing phishing and non-phishing websites, with a relatively low number of misclassifications.

Table 15. Accuracy Results of SVM with PCA 90:10

Class	Precision	Recall	F1-Score	Support
-1	0.97	0.95	0.96	502
1	0.96	0.95	0.96	604
Accuracy			0.96	1106
Macro Avg	0.96	0.96	0.96	1106
Weighted Avg	0.97	0.96	0.96	1106

Based on Table 15, which presents the accuracy results of SVM with PCA for the 90:10 data split, the model achieved a precision of 0.97 for class -1 (non-phishing) and 0.96 for class 1 (phishing), demonstrating strong performance in predicting both classes. The recall for both classes is 0.95, indicating that the model is effective in identifying phishing websites while maintaining a good balance. The F1-score for both classes is 0.96, highlighting the balance between precision and recall. The overall model accuracy is 0.96, with the macro and weighted averages for precision, recall, and F1-score also at 0.96, demonstrating consistent performance across the dataset.

4. DISCUSSIONS

In this section, we discuss the results obtained from testing the Gradient Boosting and Support Vector Machine (SVM) algorithms for phishing website classification, using Principal Component Analysis (PCA) as a dimensionality reduction technique. The performance of each algorithm is analyzed under various data split scenarios (70:30, 80:20, and 90:10), and key metrics, including precision, recall, F1-score, and accuracy, are considered to evaluate their effectiveness in detecting phishing websites.

The results of this study are in line with several previous studies that also applied Gradient Boosting and SVM in phishing detection. Research conducted by Dr. M. Prasad and Ansifa Kouser M used a Gradient Boosting Classifier based on URL features and achieved 97% accuracy with a low false positive rate. Similar results were also found in the study by Kamal Omari, who applied Gradient Boosting to detect phishing domains with 97.2% accuracy and an F1-score of 0.969, demonstrating a good balance between precision and recall. Additionally, research by Rizal Dwi Prayogo et al. optimized Gradient Boosting with hyperparameter tuning and achieved 99.16% accuracy on the Mendeley (2018) dataset, highlighting the effectiveness of model optimization in improving phishing detection accuracy.

On the other hand, other studies have shown that the SVM algorithm is also used in detecting phishing sites, although with more varied performance. For example, the study by Wahyudi Diki et al. used SVM with a polynomial kernel and achieved an accuracy of 85.71%, which is still lower compared to other studies. Emmanuel Song Shombot et al. compared polynomial and RBF kernels for SVM, with results of 84.5% for polynomial and 82.6% for RBF. Meanwhile, Sagnik Anupam et al. combined SVM with optimization algorithms such as the Gray Wolf Optimizer (GWO), which showed improved performance compared to conventional approaches.

Compared to previous studies, the SVM model in this research achieved an accuracy of up to 97%, which is equivalent to studies that used Gradient Boosting with hyperparameter optimization. However, the Gradient Boosting model in this study only reached 94%, which is still lower than some other studies. This discrepancy is due to differences in the dataset, the features used, and the hyperparameter optimization methods that have not been fully applied in this research.

Based on the experimental results, SVM demonstrated the best performance with the highest accuracy, especially in detecting phishing websites. This makes it a preferred model in scenarios with high sensitivity to phishing threats. However, this model requires more computational power and is sensitive to parameter selection, such as the kernel and C values, which can affect its performance stability. Meanwhile, Gradient Boosting remains a strong alternative, with a stable accuracy of 94%, but it has a lower recall for the non-phishing class. This can lead to some safe websites being misclassified as phishing. Nevertheless, this model is more flexible and can be further improved through hyperparameter tuning techniques.

This research has a significant impact on cybersecurity, especially in combating increasingly complex phishing attacks. By implementing a machine learning-based approach, phishing detection systems can automatically identify malicious websites and protect users from cyber threats. The SVM model has proven to be more sensitive in detecting phishing sites, making it suitable for implementation in browsers, email security systems, or firewalls to prevent users from accessing suspicious websites. Meanwhile, the Gradient Boosting model demonstrates more stable performance, making it a good choice for real-time security systems that require a balance between accuracy and detection speed. However, since phishing techniques continue to evolve and become harder to detect, this machine learning-based approach must be continuously updated with the latest datasets to remain effective in addressing new threats.

The model developed in this study can be implemented in cloud-based phishing detection systems or browser plugins to alert users when they visit suspicious websites. To enhance model performance, future research can utilize larger and more diverse datasets to improve generalization. Additionally, deep learning approaches such as CNN or LSTM can be employed to handle more complex features of phishing websites. Further hyperparameter optimization, particularly for Gradient Boosting, could also be explored to achieve higher accuracy, similar to previous studies. Moreover, an ensemble learning approach could be a promising solution by combining the strengths of SVM and Gradient Boosting into a more robust system.

5. CONCLUSION

Both Gradient Boosting and SVM with PCA are effective algorithms for detecting phishing websites. Based on the experimental results, both methods can classify phishing websites accurately. However, SVM demonstrates superior performance, particularly in terms of accuracy and recall, making it more sensitive in detecting phishing sites. The SVM model achieved higher accuracy in the 80:20 and 70:30 data splits, ranging from 96% to 97%, and exhibited better recall rates for phishing sites. Nevertheless, Gradient Boosting remained stable, with an accuracy of approximately 94%, offering a balance between precision and recall for both classes. Therefore, SVM can be the primary choice in scenarios requiring high sensitivity to phishing detection, while Gradient Boosting remains a solid alternative when a balance between phishing and non-phishing detection is needed. This study highlights the importance of selecting the appropriate algorithm based on specific phishing detection needs. For future research, further exploration can be conducted by incorporating additional features, optimizing hyperparameters, or utilizing deep learning models such as CNN to enhance accuracy and robustness in real-world phishing detection systems.

ACKNOWLEDGEMENT

We would like to express our gratitude to Mrs. Nurhikma Arifin, S. Kom., M.T., and Mr. Wawan Firgiawan, S.T., M.T., as the first and second supervisors, for their guidance, mentorship, and valuable feedback throughout the research process and the writing of this article. We also extend our thanks to the Faculty of Engineering and Universitas Sulawesi Barat for their support and contribution to the success of this research.

REFERENCES

[1] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, and A. Setiawan, “Deteksi website phishing menggunakan teknik filter pada model machine learning,” Information System Journal (INFOS), vol. 6, no. 1, pp. 39–43, May 2023
[2] Badan Pusat Statistik (BPS), “Statistik Telekomunikasi Indonesia 2021.” [Online]. Available: https://www.bps.go.id/id/publication/2022/09/07/bcc820e694c537ed3ec131b9/statistiktelekomunikasi-indonesia-2021.html.
[3] Asosiasi Penyelenggara Jasa Internet Indonesia (Apjii), “Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang.” [Online]. Available: Https://Apjii.Or.Id/Berita/D/Apjii-Jumlah-Pengguna-Internet-Indonesia-Tembus-221-Juta-Orang.
[4] M. F. Al Rifqi, M. Dina, M. Nknababan, and S. Aisyah, “Infokum dilisensikan di bawah lisensi

internasional Creative Commons Attribution-Noncommercial 4.0 (CC BY-NC 4.0)," [Online]. Available: http://infor.seaninstitute.org/index.php/infokum/index

[5] A. Mustopa, "Sistemasi: Jurnal Sistem Informasi perbandingan logistic regression dan random forest menggunakan correlation-based feature selection untuk deteksi website phishing," [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[6] R. P. Ramadhan and T. Desyani, "Implementasi algoritma J48 untuk identifikasi website phishing," *Teknik dan Multimedia*, vol. 1, no. 2, 2023.

[7] L. R. Kalabarige, R. S. Rao, A. R. Pais, and L. A. Gabralla, "A boosting-based hybrid feature selection and multi-layer stacked ensemble learning model to detect phishing websites," *IEEE Access*, vol. 11, pp. 71180–71193, 2023, doi: 10.1109/ACCESS.2023.3293649.

[8] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or not phishing? A survey on the detection of phishing websites," *IEEE Access*, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[9] M. Prasad and A. K. M, "Phishing website prediction using gradient boosting classifier," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 7, pp. 1329–1335, Jul. 2023, doi: 10.22214/IJRASET.2023.54854.

[10] K. Omari, "Phishing detection using gradient boosting classifier," *Procedia Comput. Sci.*, vol. 230, pp. 120–127, 2023, doi: 10.1016/j.procs.2023.12.067.

[11] R. D. Prayogo, A. R. Alfisyahrin, W. Gambetta, and S. A. Karimah, "An explainable machine learning-based phishing website detection using gradient boosting," in *Proc. Int. Conf. Inf. Technol. Res. Innov. (ICITRI)*, Jakarta, Indonesia, 2024, pp. 76–81, doi: 10.1109/ICITRI62858.2024.10698870.

[12] D. Wahyudi, M. Niswar, and A. A. P. Alimuddin, "Website phishing detection application using support vector machine (SVM)," *J. Inf. Technol. Its Util.*, vol. 5, no. 1, pp. 18–24, Jun. 2022, doi: 10.56873/jitu.5.1.4836.

[13] E. S. Shombot, G. Dusserre, R. Bestak, and N. B. Ahmed, "An application for predicting phishing attacks: A case of implementing a support vector machine learning model," *Cyber Security Appl.*, vol. 2, p. 100036, 2024, doi: 10.1016/j.csa.2024.100036.

[14] S. Anupam and A. K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms," *Telecommun. Syst.*, vol. 76, no. 1, pp. 17–32, Jan. 2021, doi: 10.1007/s11235-020-00739-w.

[15] E. A. Winanto, Y. Novianto, S. Sharipuddin, I. S. Wijaya, and P. A. Jusia, "Peningkatan performa deteksi serangan menggunakan metode PCA dan random forest," *J. Teknol. Inf. Dan Ilmu Komputer. *, vol. 11, no. 2, pp. 285–290, Apr. 2024, doi: 10.25126/jtiik.20241127678.

[16] D. Tuapattinaya, A. Wibowo, and Computer Science Department, Bina Nusantara University, Jakarta, Indonesia, "Phishing website detection using neural network and PCA based on feature selection," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 1150–1152, Mar. 2020, doi: 10.35940/ijrte. D 4532.038620.

[17] I. M. Karo Karo and Hendriyana, "Klasifikasi penderita diabetes menggunakan algoritma machine learning dan Z-score," *Jurnal Teknol. Terpadu*, vol. 8, no. 2, pp. 94–99, Feb. 2022, doi: 10.1007/S11235-020-00739-W.

[18] B. M. Akbar, A. T. Akbar, and R. Husaini, "Analisis sentimen dan emosi vaksin Sinovac pada Twitter menggunakan Naïve Bayes dan valence shifter," *Jurnal Teknol. Terpadu*, vol. 7, no. 2, pp. 83–92, Dec. 2021, ISSN: 2477-0043, ISSN Online: 2460-7908.

[19] I. Sabilirrasyad, A. Muliawan, M. Hermansyah, N. A. Prasetyo, and A. Wahid, "Unveiling X/Twitter's Sentiment Landscape: A Python Crawler That Maps Opinion Using Advanced Search," vol. 1, no. 1, [Online]. Available: https://twitter.com/search-advanced?lang=en.

[20] H. 1, T. Wahyuningsih , and E. Rahwanto, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer." http://archive.ics.uci.edu/ml.

[21] J. Badriyah, N. Ramadhani, A. Muliawan, K. R. Ummah, A. Amrullah, and P. Korespondensi, "Jurnal Restikom: Riset Teknik Informatika dan Komputer Penerapan Dimensi Reduksi Pada Machine Learning Dalam Klasifikasi Kanker Payudara Berdasarkan Parameter Medis," vol. 6, no. 3, pp. 526–533, 2024. Available: https://restikom.nusaputra.ac.id

[22] A. K. Gárate-Escamila, A. Hajjam El Hassani, and E. Andrès, “Classification models for heart disease prediction using feature selection and PCA,” Inform Med Unlocked, vol. 19, Jan. 2020, doi: 10.1016/j.imu.2020.100330.

[23] Z. Susanti, P. Sirait, and E. S. Panjaitan, “Peningkatan Kinerja Random Forest Melalui Seleksi Fitur Secara Pca Untuk Mendeteksi Penyakit Diabetes Tahap Awal,” Jurnal Sains dan Teknologi, vol. 4, no. 3, pp. 51–56, doi: 10.55338/saintek. v5i1.1093.

[24] G. S. M. Khamis, Z. M. S. Mohammed, S. M. Alanazi, A. F. A. Mahmoud, F. A. Abdalla, and S. A. Bkheet, “Prediction of Myocardial Infarction Complications using Gradient Boosting,” Eng. Technol. Appl. Sci. Res., vol. 14, no. 6, pp. 18550–18556, Dec. 2024, doi: 10.48084/etasr.9076.

[25] J. Smith et al., “Placeholder Text: A Study,” Penerapan Metode Support Vector Machine Learning Dalam Klasifikasi Bunga Iris" vol. 3, Jul. 2021, doi: 10.10/X.

[26] N. Huda Ovirianti, M. Zarlis, and H. Mawengkang, “Support Vector Machine Using AClassification Algorithm,” Jurnal dan Penelitian Teknik Informatika, vol. 6, no. 3, 2022, doi: 10.33395/sinkron. v 7i3.

[27] A. F. Nugraha, R. F. A. Aziza, and Y. Pristyanto, “Penerapan metode Stacking dan Random Forest untuk meningkatkan kinerja klasifikasi pada proses deteksi web phishing,” Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan, vol. 7, no. 1, pp. 39-44, Jun. 2022.