Volume 6 Issue 1 (2026) Pages 133-152 P-ISSN:2798-2696 E-ISSIN:2798-0057
JOSTECH
Journal of Science and Technology
Website: https://ejournal.uinib.ac.id/jurnal/index.php/jostech Email: jostech@uinib.ac.id

Comparison of Kernel Functions in Support Vector Machine (SVM) Method for Phishing Website Detection

Muhammad Farhan Said[1], Norma Muhtar ✉[2], Bahriddin Abapihi[3], Arman[4], Kabil Djafar[5], Herdi Budiman[6]
Mathematics, Universitas Halu Oleo, Indonesia[1, 2, 3, 4, 5, 6]
email: muhfarhan292@gmail.com [1], norma.muhtar@uho.ac.id [2], bahriddinabapihi@yahoo.com [3], arman.mtmk@uho.ac.id [4], kabildjafar@gmail.com [5], herdi.budiman@uho.ac.id [6]

Received 07 Feb 2026, Accepted 31 Maret 2026, Published 31 Maret 2026

Abstrak

Serangan *phishing* masih menjadi ancaman signifikan dalam keamanan siber, dengan laporan dari Badan Siber dan Sandi Negara (BSSN) yang menunjukkan peningkatan insiden siber dalam beberapa tahun terakhir. Meskipun metode *machine learning* telah banyak digunakan untuk mendeteksi *phishing*, masih terbatas penelitian yang secara sistematis membandingkan kinerja berbagai fungsi kernel dalam kerangka *Support Vector Machine* (SVM), terutama yang dikombinasikan dengan teknik seleksi fitur. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja empat fungsi kernel SVM, yaitu linear, polinomial, *radial basis function* (RBF), dan sigmoid, dalam mendeteksi *website phishing*. *Dataset* yang digunakan dalam penelitian ini diperoleh dari Hannousse dan Yahiouche (2021), yang terdiri dari 9.587 data setelah tahap praproses, serta merepresentasikan permasalahan klasifikasi biner yang seimbang. Untuk meningkatkan efisiensi model, dilakukan seleksi fitur berbasis SHAP yang mereduksi jumlah fitur dari 87 menjadi 10 fitur paling relevan. Evaluasi model dilakukan menggunakan *stratified 5-fold cross-validation* dengan metrik performa berupa *F1-score* sebagai metrik utama. Parameter kernel ditentukan melalui *grid search* untuk memperoleh konfigurasi terbaik pada masing-masing kernel. Hasil eksperimen menunjukkan bahwa kernel RBF memperoleh nilai F1-score tertinggi sebesar 94,83%, diikuti secara sangat dekat oleh kernel polinomial sebesar 94,80%, kernel linear sebesar 92,94%, dan kernel sigmoid sebesar 90,71%. Hasil ini menunjukkan bahwa kernel RBF dan polinomial memiliki kinerja yang sebanding dan lebih unggul dibandingkan kernel linear dan sigmoid dalam tugas deteksi website phishing. Secara keseluruhan, penelitian ini menegaskan pentingnya pemilihan kernel dan penyesuaian parameter dalam model SVM untuk deteksi *phishing*. Temuan ini memberikan implikasi praktis dalam pengembangan sistem deteksi phishing yang lebih efektif serta mendukung pengambilan keputusan dalam pemilihan kernel pada aplikasi keamanan siber di dunia nyata.

Kata Kunci: *deteksi phising; Support vector machine; Fungsi kernel; Machine learning; Keamanan siber; Klasifikasi website.*

Abstract

Phishing attacks remain a significant cybersecurity threat, with reports from Indonesia's National Cyber and Crypto Agency (BSSN) indicating a substantial increase in cyber incidents in recent years. Despite the widespread use of machine learning methods for phishing detection, limited studies have systematically compared the performance of different kernel functions within the Support Vector Machine (SVM) framework, particularly when combined with feature selection techniques. This study aims to evaluate and compare the performance

of four SVM kernel functions—linear, polynomial, radial basis function (RBF), and sigmoid—in detecting phishing websites. The dataset used in this research was obtained from Hannousse and Yahiouche (2021), consisting of 9,587 instances after preprocessing, representing a balanced binary classification problem. To improve model efficiency, SHAP-based feature selection was applied to reduce the original 87 features to 10 most relevant features. Model evaluation was conducted using stratified 5-fold cross-validation, with performance measured using F1-score as the main metric. Kernel parameters were explored through grid search to identify the best-performing configuration for each kernel. The experimental results show that the RBF kernel achieved the highest F1-score of 94.83%, followed closely by the polynomial kernel at 94.80%, the linear kernel at 92.94%, and the sigmoid kernel at 90.71%. These results indicate that the RBF and polynomial kernels provide comparable performance, both outperforming the linear and sigmoid kernels in phishing website detection tasks. Overall, this study highlights the importance of kernel selection and parameter tuning in SVM-based phishing detection. The findings provide practical guidance for developing more effective phishing detection systems and support informed kernel selection in real-world cybersecurity applications.
Keywords: *Phishing detection; Support vector machine; Kernel function; Machine learning; Cybersecurity; Website classification.*

✉ Corresponding author

INTRODUCTION

1. Background

Phishing remains one of the most persistent cyber threats because it exploits human trust to obtain sensitive information such as usernames, passwords, and financial data. In Indonesia, phishing incidents have risen sharply, with BSSN reporting a 70% increase in 2024 compared to the previous year. Common tactics include fake "gift giveaway" campaigns, malware-based attacks, and scams exploiting family emergencies. A Universitas Gadjah Mada study showed that 36.9% of respondents faced phishing attempts, followed by malware (33.8%) and emergency scams (26.5%). These attacks cause not only financial losses but also emotional distress for victims [1][2].

The increasing number of phishing incidents in Indonesia indicates the need for an effective detection method capable of handling complex website data. Phishing websites are generally represented by a set of numerical features extracted from URLs and webpage characteristics, such as URL length, number of symbols, and domain-related attributes [3]. Furthermore, the relationships among these features are often complex and non-linear, making traditional linear classification approaches less effective [4]. In this context, Support Vector Machine (SVM) is considered an appropriate method due to its strong capability in binary classification and its effectiveness in handling high-dimensional data. Additionally, the use of kernel functions enables SVM to model non-linear decision boundaries, allowing better separation between phishing and legitimate websites [5].

Support Vector Machine (SVM) is among the most effective machine learning methods for phishing detection, capable of classifying complex data such as email text or URLs. By maximizing the margin between classes, SVM can recognize phishing patterns even with limited samples, showing resilience against unstructured data and evolving threats [6][7]. The kernel function plays a vital role by mapping data into higher-dimensional spaces for better

separation. Support Vector Machines utilize various kernel functions to transform data into higher-dimensional spaces, with the most prevalent being Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid kernels. The kernel choice directly influences detection accuracy, with poor selection leading to underfitting or overfitting [8][9][10]. Therefore, SVM is selected in this study as a suitable approach for phishing website detection, particularly for datasets characterized by complex feature relationships.

Research in phishing detection using machine learning has advanced significantly, yet most previous studies focus on broad comparisons between various algorithms rather than the deep optimization of a specific method. For instance, Kolla et al. (2022) compared SVM with other techniques such as Random Forest and Decision Tree but did not conduct an in-depth investigation into the internal optimization of the SVM algorithm itself. Marcello and Putri (2025) and Wahyudi et al. (2022) demonstrated that feature selection techniques could significantly enhance accuracy, but the kernel variations tested in their research remained limited. Furthermore, Rumini et al. (2023) restricted their comparative analysis to RBF and Linear kernels, leaving the potential of other functions like Polynomial and Sigmoid unanalyzed in handling the complexity of phishing website features. Safi and Singh (2023), in their systematic literature review, also emphasized that while high accuracy is frequently reported, the selection of an appropriate mapping function (kernel) remains a crucial factor that is often under-explored in experimental frameworks [11][12][13][14][15].

To date, no study has comprehensively compared the four primary SVM kernel functions—Linear, Polynomial, RBF, and Sigmoid—simultaneously using rigorous parameter optimization via Grid Search. The novelty of this research lies in the systematic evaluation of the performance and mathematical behavior of these four kernels on a dataset of 9,587 records to identify the most adaptive function for distinguishing between phishing and legitimate data patterns. By addressing this research gap, this study not only aims to achieve superior detection performance but also provides a detailed technical justification for the effectiveness of each kernel against complex cybersecurity feature characteristics, which has not been explicitly articulated in prior literature.

To address these identified gaps, this study systematically evaluates and compares the performance of various SVM kernel functions in classifying phishing websites. The primary objective is to enhance detection accuracy and provide comprehensive technical insights into how kernel selection influences the model's ability to handle complex cyber threats, ultimately establishing a more robust framework for phishing mitigation.

2. Problem Statement

The problem statements in this study are as follows:

1. How do the linear, polynomial, radial basis function (RBF), and sigmoid kernel functions perform in SVM-based phishing website detection, measured using F1-score as the main metric on the selected phishing website dataset?
2. Which SVM kernel function provides the most appropriate performance for phishing website detection under the same preprocessing, feature selection, and evaluation framework?

3. Research Objectives

The objectives of this study are as follows:

1. To evaluate and compare the performance of the linear, polynomial, RBF, and sigmoid kernel functions in SVM-based phishing website detection using F1-score.
2. To identify the kernel function that performs best for phishing website detection on the selected dataset under the same preprocessing, feature selection, and evaluation conditions.
3. To provide practical guidance for kernel selection in SVM-based phishing website detection models.

4. Theoretical Background

Support Vector Machine

SVM is a machine learning classification method developed by Vladimir Vapnik that predicts classes based on patterns learned during training. Classification is performed using a decision boundary (hyperplane) that separates positive and negative opinion classes. Intuitively, a good hyperplane maximizes the margin, which is the distance between the closest training data points of each class and the hyperplane. Generally, a larger margin results in lower generalization error [16].

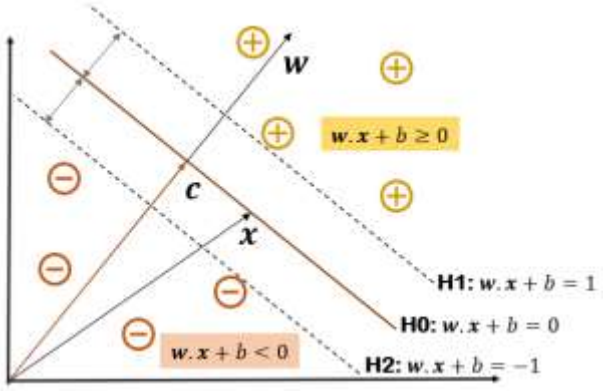

Figure 1. A hyperplane that maximizes the distance between two classes (Adapted from [17]**)**

Let a dataset $S = \{(x_i, t_i)\}_{i=1}^{N}$ be given, where each sample (x_i, t_i) consists of a data vector x_i represented by m features $(x_{1i}, x_{2i}, \ldots, x_{mi})$, and a target variable t_i, which is categorical. A linear SVM model can be represented by the following equation:

$$y(\boldsymbol{x_i}) = \boldsymbol{w^T x_i} + b \quad (1)$$

where $y(\boldsymbol{x_i})$ serves as the prediction of t_i, $\boldsymbol{w}$ is the weight vector (model parameters), $\boldsymbol{x_i}$ is the data variable, and b is the bias term.

Using the quadratic optimization technique, the resulting SVM classification function is defined as follows:

$$f(\boldsymbol{u}) = sign(\boldsymbol{w}.\boldsymbol{u} + b) \quad (2)$$

where $\boldsymbol{u}$ is an unknown data vector [18].

Kernel Function

Non-linear SVM is an extension designed to address classification problems where the data cannot be separated linearly. In non-linear cases, it is not necessary to explicitly construct the transformation; instead, a kernel function approach is used [19]. The kernel function operates by transforming the original input data into a higher-dimensional feature space,

where the data becomes linearly separable. This process is known as the kernel trick [8]. The kernel trick can be formulated as follows:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{\phi}(\boldsymbol{x_i}).\boldsymbol{\phi}(\boldsymbol{x_j}) \quad (3)$$

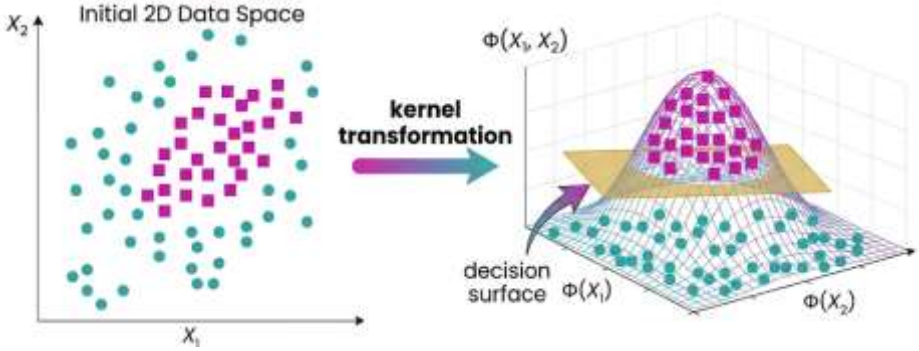

Figure 2. Hyperplane for non-linearly separable classes (Adapted from [17])

The mapping process can be notated as $\phi: \mathbb{R}^p \rightarrow \mathbb{R}^q$ where p and q represent the dimensions. The transformation ϕ is implicitly defined by the kernel function since the explicit form of the transformation is usually unknown. The classification of data can then be determined using the following equation:

$$f(\boldsymbol{\phi}(\boldsymbol{x})) = sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x_i}, \boldsymbol{x_j}) + b\right) \quad (4)$$

Common kernel functions used in the SVM method include the following:

1) Linear Kernel

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i}.\boldsymbol{x_j} \quad (5)$$

2) Polynomial Kernel

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i}.\boldsymbol{x_j} + 1)^h \quad (6)$$

3) Gaussian Radial Basis Function (RBF) Kernel

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = exp\left\{-\gamma \left\|\boldsymbol{x_i} - \boldsymbol{x_j}\right\|^2\right\} \quad (7)$$

4) Sigmoid Kernel

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = tanh\,(\kappa \boldsymbol{x_i}.\boldsymbol{x_j} + \delta) \quad (8)$$

where h, γ, κ, and δ are kernel parameters. For the Polynomial kernel, the configuration follows the non-homogeneous form defined as $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{x_i}.\boldsymbol{x_j} + c)^h$. In this study, the constant term c was set to 1, adhering to the foundational formulation proposed by Vapnik (1995), the pioneer of Support Vector Machines, to ensure a robust mapping in higher-dimensional space. The kernel constant c remained fixed at 1 to maintain consistency with standard SVM theoretical frameworks [20].

The polynomial kernel and the RBF kernel always satisfy Mercer's condition, whereas the sigmoid kernel only satisfies Mercer's condition for certain values of κ and δ [20]. Suitable values for the sigmoid kernel parameters are $\kappa = \frac{1}{d}$ and $\delta = 1$ where d is the dimensionality of the data [21].

K-Fold Cross Validation

In k-fold cross-validation, the data is split into k equal parts; each fold is used once for testing and the remaining folds for training, ensuring every instance is tested exactly once to

reduce bias. A commonly used variation is stratified k-fold cross-validation, which preserves the original class distribution in each fold. To ensure the robustness and generalizability of the model, this study employed Stratified 5-Fold Cross-Validation. In this procedure, the dataset was partitioned into five equal-sized folds, where each fold maintained the original class distribution of 'phishing' and 'legitimate' websites. The choice of $k = 5$ was strategically selected to strike an optimal balance between computational efficiency and the bias-variance tradeoff. Given the dataset size and the extensive parameter tuning performed via Grid Search, 5-fold cross-validation provides a sufficiently low-variance estimate of the model's performance while preventing excessive computational overhead [22][23]. This configuration is consistent with established practices in machine learning research for cybersecurity, ensuring that the evaluation is not biased by a single train-test split.

Confusion Matrix

In this study, the performance of the SVM model is evaluated using a confusion matrix tailored for a binary classification task, where websites are categorized as either 'phishing' (positive class) or 'legitimate' (negative class). The matrix provides a detailed breakdown of the model's predictions: True Positive (TP) represents phishing sites correctly identified as phishing; True Negative (TN) represents legitimate sites correctly identified as legitimate; False Positive (FP) occurs when a legitimate site is incorrectly flagged as phishing; and False Negative (FN) occurs when a phishing site is missed and classified as legitimate. [24]. Classification accuracy can be assessed by calculating accuracy, precision, recall, and specificity, using the following formulas [9]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (9)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (11)$$

$$Specificity = \frac{TN}{TN+FP} \times 100\% \quad (12)$$

However, precision and recall alone may not fully reflect model performance. Thus, the F1-score is often used as a single metric that balances both, offering a more comprehensive measure of accuracy. It is defined as follows [25]:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (13)$$

In addition to the F1-score, the Receiver Operating Characteristic Area Under Curve (ROC AUC) serves as a vital evaluation metric for binary classification, particularly when dealing with imbalanced datasets. The ROC curve provides a visual assessment of a classifier's ability to distinguish between classes by plotting the True Positive Rate (sensitivity) against the False Positive Rate across various threshold values. While the F1-score balances precision and recall to represent overall performance, ROC AUC specifically emphasizes the model's effectiveness in identifying the minority class, which is often the primary focus in specialized classification tasks. A perfect classifier achieves an AUC of 1.0, whereas a score of 0.5 indicates performance no better than random guessing. Furthermore, empirical studies demonstrate that advanced optimization frameworks, such as Optuna, can significantly enhance ROC AUC scores, making it a robust benchmark for comparing the generalization capabilities of different machine learning models [23].

Z-Score

In general, differing attribute value ranges can lead to bias in the data. To prevent this, the data can be normalized or standardized, ensuring that all features are treated with equal weight. One commonly used technique is z-score normalization. Z-score normalization utilizes the mean and standard deviation, making it more robust to outliers and to new values that are smaller than the minimum or greater than the maximum. Z-score normalization can be calculated using the following equation [17][26].

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{14}$$

where:

- z_{ij} = Z-score normalized value of the i-th Z-score normalized value of the j-th feature.
- x_{ij} = Original value of the i-th data point on the j-th feature.
- μ_j = Mean of the j-th feature.
- σ_j = Standard deviation of the j-th feature.

The features were normalized using the Z-score method (standardization) to ensure that all variables contribute equally to the distance calculations in the SVM model. Unlike Min-Max normalization, Z-score is more robust to outliers and prevents features with larger numerical ranges from dominating the model [14][26]. This is particularly important for distance-based classifiers like SVM, where the convergence of the optimization process is highly dependent on the scale of the input data.

5. Related Works

The development of machine learning techniques has significantly enhanced the efficacy of phishing website detection. A systematic literature review conducted by Safi and Singh (2023) revealed that machine learning is the most prevalent approach in this domain, with datasets frequently sourced from PhishTank and Alexa. While various algorithms such as Random Forest, Decision Tree, and XGBoost are often compared, Support Vector Machine (SVM) remains a cornerstone due to its robust classification capabilities. Kolla et al. (2022) positioned SVM among several high-performing techniques, although noting that its performance is highly dependent on proper configuration and comparison with other ensemble methods [15][11].

Recent studies have emphasized that SVM's performance is not only a result of the algorithm itself but also of optimized feature selection and kernel choices. Marcello and Putri (2025) demonstrated that implementing Recursive Feature Elimination (RFE) with SVM can improve accuracy by up to 20%, highlighting the necessity of refining the feature set before classification. Furthermore, the choice of kernel function plays a critical role in handling the non-linear nature of phishing data. Rumini et al. (2023) compared Linear and Radial Basis Function (RBF) kernels, finding that the RBF kernel achieved a superior accuracy of 96.42% compared to 92.58% for the Linear kernel. Similarly, Wahyudi et al. (2022) explored the Polynomial kernel, achieving an accuracy of 85.71% through rigorous parameter optimization, such as adjusting the degree and penalty values. Despite these efforts, there remains a research gap in providing a simultaneous and comprehensive comparison of all primary kernels—Linear, RBF, Polynomial, and Sigmoid—within a single experimental framework to determine the most adaptive function for modern phishing datasets [12][14][13].

The following table summarizes the key methodologies and findings of the primary studies discussed:

Table 1. Summary Table of Related Works

Author (Year)	Methods	Dataset	Results
Marcello & Putri (2025)	SVM and Recursive Feature Elimination (RFE)	Website phising dataset from from Mendeley data	RFE improved SVM accuracy to 96.09%, an increase of up to 20% in specific tests.
Rumini et al. (2023)	SVM (Linear and RBF Kernels)	Web Phising dataset from UCI Machine Learning Repository	RBF kernel (96.42%) significantly outperformed Linear kernel (92.58%).
Kolla et al. (2022)	DT, RF, MLP, SVM, XGBoost	Website phising dataset from University of New Brunswick	Random Forest was the best-performing model among the compared techniques.
Safi & Singh (2023)	SLR (80 papers)	Various website phising dataset from PhishTank and Alexa from 80 papers	Identified ML as the most used approach (57 studies) and PhishTank as the primary data source.
Wahyudi et al. (2022)	SVM (Polynomial), DT, KNN	Website phising dataset from Kaggle Datasets	Polynomial kernel (degree 9) achieved 85.71% accuracy through parameter optimization.

RESEARCH METHODS

This study utilizes a dataset compiled by Hannousse and Yahiouche (2021), sourced from Mendeley Data [27]. The dataset contains 11,430 entries—5,715 phishing websites and 5,715 legitimate ones—ensuring a balanced class distribution. It consists of 89 columns: the first contains the website URL, the next 87 represent features used for phishing detection, and the final column indicates the website status (phishing or legitimate). The data, collected up to May 2020, captures common characteristics of phishing websites. The 87 features within this dataset encompass fundamental structural and content-based characteristics, such as URL length, SSL certificate status, and domain age, that remain core indicators in modern phishing detection. This study acknowledges the temporal constraint as a potential limitation; however, the primary objective is to provide a rigorous comparative analysis of SVM kernel behaviors rather than to develop a real-time detection tool. The findings offer a significant baseline for understanding kernel adaptability, which remains generalizable to contemporary patterns that rely on these established feature sets.

This study utilizes the top 10 most significant features as identified through the SHAP (SHapley Additive exPlanations) analysis conducted by Shafin (2024) [28]. This decision is methodologically justified as Shafin (2024) utilized the identical dataset (Hannousse & Yahiouche, 2021) consisting of 11,430 entries with the same feature space and target distribution as used in this research. The adoption of these specific features serves as an external validation of feature importance, leveraging Shafin's rigorous explainability framework which confirmed the stability and global importance of these predictors across multiple machine learning models. By utilizing these pre-validated features, this study maintains a high degree of experimental consistency with state-of-the-art benchmarks while

focusing its independent contribution on the systematic comparison and hyperparameter optimization of the four primary SVM kernel functions.

The features of websites used in this research are as follows:

a. nb_www (X_1): Indicates how many times the keyword 'www' appears in the URL.
b. longest_word_path (X_2): Measures the length of the longest word in the URL path.
c. phish_hints (X_3): Shows the number of sensitive words in the URL that may indicate phishing.
d. nb_hyperlinks (X_4): Indicates the total number of hyperlinks on the webpage.
e. ratio_extHyperlinks (X_5): Measures the ratio of external hyperlinks to the total number of hyperlinks on the webpage.
f. safe_anchor (X_6): Calculates the percentage of unsafe anchors out of the total number of anchor links on the webpage.
g. domain_age (X_7): Shows the age of the domain in days.
h. ratio_extRedirection (X_8): Measures the ratio of external redirections to the total number of redirections on the webpage.
i. google_index (X_9): Checks whether a given URL is indexed by the Google search engine.
j. page_rank ($\boldsymbol{X_{10}}$): Measures the PageRank score of website on a scale from **0** to **10**.

The research procedure follows these steps:

1) A literature review was conducted to identify the research gap, understand the characteristics of phishing website detection, and examine previous studies related to SVM, kernel functions, and feature selection. This step was necessary to ensure that the study was positioned clearly within the existing body of knowledge.
2) The dataset was collected from Mendeley Data and consisted of 11,430 website instances, with 5,715 phishing websites and 5,715 legitimate websites. A balanced dataset was selected to minimize class bias and to allow a fair comparison among kernel functions.
3) Feature selection was applied to retain the 10 most relevant features, based on SHAP analysis from prior research. This step was used to reduce dimensionality, remove less informative variables, improve computational efficiency, and focus the model on the most influential features for phishing detection.
4) Data preprocessing was performed through data labeling and cleaning. The target variable was encoded as -1 for legitimate and 1 for phisihing websites, invalid values in the domain_age feature were removed, and the URL attribute was excluded because it was not used as an input feature.
5) The dataset was evaluated using stratified 5-fold cross-validation to preserve the proportion of phishing and legitimate websites in each fold. This method was chosen because it provides a more stable and reliable performance estimate than a single train-test split, while also ensuring that each class is represented consistently across all folds.
6) Z-score normalization was then applied to the selected numerical features to ensure that all variables were on a comparable scale and to prevent features with larger magnitudes from dominating the model.
7) An SVM model was designed using four kernel functions: linear, polynomial, RBF, and sigmoid. These kernels were selected because they represent different decision boundary

characteristics, ranging from linear separation to non-linear mapping, thereby allowing a comprehensive comparison of SVM performance in phishing website detection.

8) Hyperparameter tuning was performed using grid search for each SVM kernel to identify the best parameter combination under the same evaluation framework. This step was necessary because SVM performance is highly sensitive to parameter settings such as $h, \gamma, \kappa, \delta$, and C.
9) The model was trained and tested on each fold in turn so that every subset of the data served as both training data and testing data once. This procedure helped reduce overfitting and ensured that the reported performance reflected the model's generalization ability.
10) The F1-score was calculated for each fold and then accumulated to evaluate the balance between precision and recall. This metric was selected as the main metric because phishing detection requires both high detection ability and a low false-alarm rate.
11) The final F1-score results were compared across all kernels to determine which kernel function provided the most effective performance for the selected dataset and feature set.

The research flowchart showed bellow:

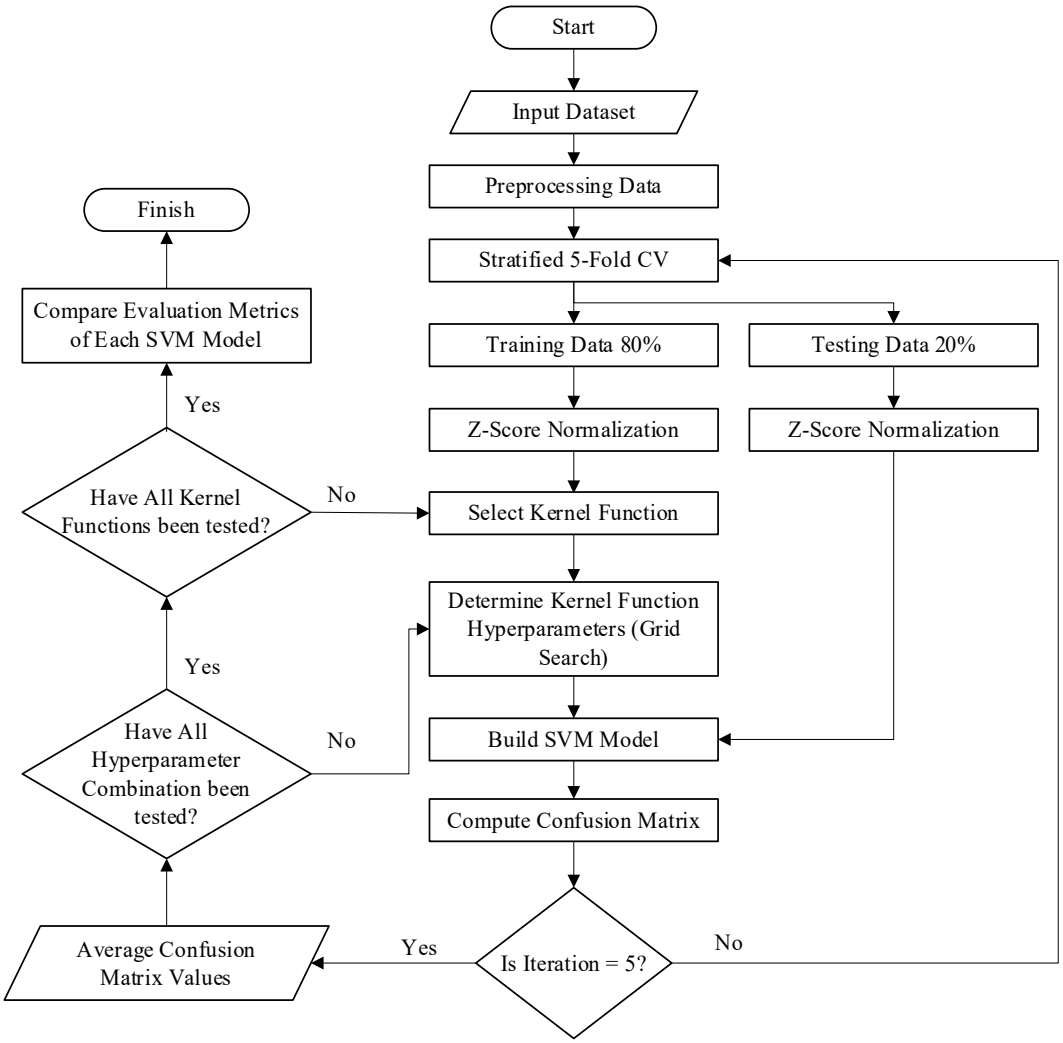

Figure 3. Research Flowchart

RESULTS AND DISCUSSION

1. Preprocessing Data

The preprocessing data process in this study comprises data labeling, cleaning, partitioning, and transformation. Labeling was applied by converting the status column from string labels ('phishing', 'legitimate') to numeric format, with -1 representing legitimate (negative class) and 1 representing phishing (positive class), ensuring compatibility with SVM input requirements. Then, data cleaning aims to improve dataset quality by removing invalid entries. Specifically, the domain_age (X_7) feature contains error values (-1 and -2) indicating extraction issues or missing data; rows with such values were excluded, reducing the dataset from 11,430 to 9,587 entries (4,715 phishing and 4,872 legitimate websites).

A systematic analysis was conducted across all 87 original features to identify missing, invalid, or outlier values. The results confirmed that invalid entries were exclusively localized within the domain_age (X_7) feature, while the other nine selected features were verified to be complete and free of inconsistencies. The data cleaning process involved the removal of 1,843 entries (approximately 16% of the raw dataset). Imputation was considered but ultimately rejected because domain_age (X_7) is a critical temporal feature; substituting such a high volume of missing values with mean or median estimates could introduce significant synthetic bias and weaken the model's ability to learn authentic phishing patterns.

Furthermore, an analysis of the removed entries revealed that they were proportionately distributed between both classes, thus maintaining the original class balance. Although a slight difference of 157 instances was introduced, this ratio (~0.97:1) is considered a near-balanced distribution in machine learning literature, where the minority class still represents nearly half of the total data. Consequently, intensive rebalancing strategies such as SMOTE or undersampling were not applied to avoid the risk of introducing synthetic bias or losing valuable information. To further mitigate any potential impact of this marginal imbalance, this study utilized Stratified 5-Fold Cross-Validation to ensure consistent class proportions across all folds and employed the F1-score as the primary evaluation metric, as it provides a robust measure of performance by balancing precision and recall for the positive class [22][23].

In this approach, the data was partitioned into five disjoint subsets (folds) of approximately equal size, while maintaining the original class proportions. In each of the five iterations, four folds were used for model training, and the remaining fold was reserved for testing. To address differences in feature scale—e.g., domain_age values in the thousands and ratio_extHyperlinks in the 0-1 range—standardization was applied to the traning data using Z-score normalization (mean $= 0$, standard deviation $= 1$). Then the mean and standard deviation from the training data used to normalize the traning data. This step prevents features with larger magnitudes from dominating the model. The iteration process was repeated five times, ensuring that every data folds was used for testing exactly once. No separate holdout set was used, as the cross-validation results provide a comprehensive estimate of the model's performance across the entire dataset.

Table 2. Stratified 5-Fold Cross Validation Data Partitioning

Fold	Positive Class	Negative Class	Total
1	975	943	1918
2	975	943	1918
3	974	943	1917
4	974	943	1917
5	974	943	1917

2. SVM Implementation

SVM classification was performed using four kernel functions: linear, polynomial, radial basis function (RBF), and sigmoid. The first kernel function employed was the linear kernel with a regularization parameter value of $C = 1$. The dataset used in this process was previously partitioned using Stratified 5-Fold Cross Validation. In the first iteration, folds 1, 2, 3, and 4 were used as the training data, while fold 5 was reserved as the testing data. This approach ensures that each data subset is used for both training and testing in a balanced manner throughout the cross-validation process.

The classification function obtained based on **Equation (2)** is as follows:

$$f(\boldsymbol{u}) = sign(0.6999u_1 - 0.9876u_2 - 0.9969u_3 + 1.1216u_4 - 0.2769u_5 + 0.2384u_6 + 0.2926u_7 + 0.0782u_8 - 0.9928u_9 + 0.8494u_{10} + 0.0588) \quad (15)$$

If the value inside the sign function is less than 0, then $f(\boldsymbol{u}) = -1$. Conversely, if the value inside the sign function is greater than or equal to 0, then $f(\boldsymbol{u}) = 1$. Using the **Equation (15)**, the classification prediction results for the testing data are obtained as follows:

Table 3. Testing Data Classification Prediction for the First Iteration of Linear Kernel SVM with $C = 1$

Testing Data	t_i	$f(z_i)$	Classification Result
$\boldsymbol{z_{7671}}$	1	1	True
$\boldsymbol{z_{7672}}$	1	1	True
$\boldsymbol{z_{7673}}$	-1	-1	True
$\boldsymbol{z_{7674}}$	1	1	True
$\boldsymbol{z_{7675}}$	-1	-1	True
$\boldsymbol{z_{7676}}$	1	1	True
...	...	...	...
$\boldsymbol{z_{9587}}$	1	1	True

where $\boldsymbol{z_{7671}} - \boldsymbol{z_{9587}}$ are normalized testing data from fold 5.

Based on the classification prediction results in **Table 3**, the confusion matrix obtained is as follows:

Table 4. Confusion Matrix for the First Iteration of Linear Kernel with $C = 1$

Confussion Matrix		Actual Class	
		Positif	Negatif
Prediction Class	Positif	$TP = 892$	$FP = 52$
	Negatif	$FN = 83$	$TN = 891$

Based on the results presented in **Table 4**, the precision and recall values can be calculated using **Equations (10)** and **Equations (11)**, Respectively.

$$Precision = \frac{TP}{TP + FP} \times 100\%$$
$$= \frac{892}{892 + 52} \times 100\%$$
$$= 94.49\%$$
$$Recall = \frac{TP}{TP + FN} \times 100\%$$
$$= \frac{892}{892 + 83} \times 100\%$$
$$= 91.49\%$$

Next, using **Equation (13)**, the F1-score is calculated as follows:

$$F1 - Score = \frac{2 \times (94.49\%) \times (91.49\%)}{(94.49\%) + (91.49\%)}$$
$$= 92.97\%$$

Thus, the F1-score obtained from the SVM model using the linear kernel function with a regularization parameter of $C = 1$ is 92.97% in the first iteration.

Following the detailed calculation of the first iteration, the comprehensive results for all five iterations are presented in the table bellow. The table includes the performance metrics for each fold, followed by the calculated mean and standard deviation.

Table 5. Value of Performance Metrics for Each Iteration of Linear Kernel with $C = 1$

Iteration	Precision	Recall	F1-Score
Iteration 1	94.49%	91.49%	92.97%
Iteration 2	94.03%	92.00%	93.00%
Iteration 3	93.59%	91.48%	92.53%
Iteration 4	92.86%	92.20%	92.53%
Iteration 5	94.15%	92.47%	93.23%
Mean ± SD	93.82% ± 0.63%	91.93% ± 0.34%	92.85% ± 0.28%

As shown in

Table 5, the small standard deviation across iterations indicates that the model is highly stable and does not suffer from significant performance fluctuations.

The value of the parameter C can be varied to generate an SVM model that achieves a higher F1-score. In this study, the values of C used are $10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}$, and 10^{3}; variations of the polynomial degrees h used are $1, 2, 3, 4$, and 5; and variations of the gamma parameter γ used in this study are $10^{-2}, 10^{-1}, 1, 10^{1}$, and 10^{2}. Meanwhile, the Sigmoid kernel parameters κ and δ are set to $\kappa = \frac{1}{d} = \frac{1}{10} = 0.1$ and $\delta = 1$ where d denotes the dimensionality of the data. These values are chosen because the sigmoid kernel only satisfies Mercer's condition under specific settings of κ and δ [20][21]. It is important to note that the Sigmoid kernel parameters were restricted to $\kappa = 0.1$ and $\delta = 1$ to strictly satisfy Mercer's condition. In contrast, the RBF and Polynomial kernels benefited from a wider hyperparameter optimization space. This disparity may place the Sigmoid kernel at a comparative disadvantage, and future studies could explore a broader range of non-Mercer parameters to evaluate their empirical performance in phishing detection. Using the same procedure as

before, the F1-score values for each variation of the regularization parameter C and hyperparameter of each kernels in the SVM model are presented as follows:

Table 6. F1-Score Values of the Linear Kernel SVM with Various Regularization Parameter C

C	F1-Score (%)
10^{-3}	91.09
10^{-2}	92.68
10^{-1}	92.84
1	92.85
10^{1}	92.92
10^{2}	92.92
10^{3}	92.94

Table 7. F1-Score Values of the Polynomial Kernel SVM with Various Regularization Parameter C and Polynomial Degree h Combinations

C	F1-Score (%)				
	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
10^{-3}	91.83	68.44	71.88	69.28	69.84
10^{-2}	91.09	77.84	92.55	76.50	82.18
10^{-1}	92.68	86.02	93.54	87.90	91.01
1	92.88	89.18	93.96	91.56	93.91
10^{1}	92.90	91.56	94.30	93.30	94.80
10^{2}	92.93	91.77	94.58	94.11	94.36
10^{3}	92.91	94.20	94.38	93.81	93.52

Table 8. F1-Score Values of the RBF Kernel SVM with Various Regularization Parameter C and Parameter γ Combinations

C	F1-Score (%)				
	$\gamma = 10^{-2}$	$\gamma = 10^{-1}$	$\gamma = 1$	$\gamma = 10^{1}$	$\gamma = 10^{2}$
10^{-3}	67.39	91.44	67.39	67.39	67.39
10^{-2}	91.33	92.29	79.35	69.38	68.73
10^{-1}	92.35	93.75	93.52	75.49	71.89
1	93.61	94.40	94.64	87.49	83.06
10^{1}	93.81	94.60	94.08	87.67	83.37
10^{2}	94.21	94.83	93.65	87.64	83.37
10^{3}	94.16	94.03	93.14	87.65	83.38

Table 9. F1-Score Values of the Sigmoid Kernel SVM with Various Regularization Parameter C

C	F1-Score (%)
10^{-3}	90.71
10^{-2}	87.15
10^{-1}	79.54
1	77.58
10^{1}	77.40
10^{2}	77.40
10^{3}	77.42

Based on the experimental results presented in **Tables 6** through **9**, several significant trends emerge regarding the influence of the penalty parameter C, the polynomial degree (h), and the RBF kernel coefficient γ on phishing detection performance. The grid search process reveals that each kernel function responds differently to hyperparameter configurations,

reflecting how the complexity of the phishing feature space—reduced via SHAP-based selection—is mapped by each specific function.

Consistent trend in the Linear, Polynomial, and RBF kernels is the performance improvement correlated with the increase in the value of C. Theoretically, C governs the trade-off between maximizing the margin width and minimizing classification errors on the training data. In the Linear kernel (

Table 6), performance surges from an F1-Score of 91.09% at $C = 0.001$ to a steady state of 92.94% at $C = 1000$. This suggests that the phishing dataset, though optimized through feature selection, still requires a relatively high penalty to suppress misclassifications near the decision boundary. Hyperparameter optimization is crucial; a low C value tends to cause underfitting by allowing a margin that is too wide and ignores critical feature details, whereas an optimal C allows the model to construct a more precise hyperplane for separating legitimate and phishing sites [25].

In contrast to the other kernels, the Sigmoid kernel (**Table 9**) exhibits an inverse and suboptimal trend. As C increases from 0.001 to 100, the F1-Score consistently decreases, only showing a slight, non-optimal recovery at $C = 1000$. The decline in performance at higher C values indicates that a stricter penalty forces the model to over-fit to a feature space mapping that does not naturally separate phishing and legitimate sites, leading to poor generalization. Sigmoid kernel is highly sensitive to parameter configurations and can exhibit unstable performance compared to other kernels. Furthermore, since the Sigmoid kernel effectively functions as a two-layer perceptron, its failure suggests that the underlying distribution of phishing features does not align with the hyperbolic tangent activation logic, unlike the more robust mapping provided by the RBF kernel [29].

Table 7 demonstrates that the performance of the Polynomial kernel is significantly influenced by the interaction between the penalty parameter C and the polynomial degree h. Unlike the Linear or RBF kernels, the Polynomial kernel does not exhibit a monotonic trend; instead, the F1-score fluctuates across different degree configurations. The peak performance, an F1-score of 94.80%, is achieved at $h = 5$ and $C = 10$. The non-linear and inconsistent trend across various degrees suggests that the feature space mapping of this phishing dataset is highly sensitive to the specific dimensionality of the polynomial expansion. Mathematically, a higher-degree kernel allows the model to capture high-order, non-linear correlations between features such as "URL length" and "domain age". The polynomial kernel is often more difficult to tune than the RBF kernel because its numerical stability and generalization ability can vary wildly depending on the combination of the hyperparameter [30]. The fact that $h = 5$ yielded the best result indicates that the phishing features in this study possess complex, high-dimensional interactions that require a more sophisticated decision boundary than simpler low-degree models can provide.

As shown in **Table 8**, the RBF kernel achieved the highest performance in this study, peaking at an F1-score of 94.83% ($C = 100$, $\gamma = 0.1$). The results indicate that $\gamma = 0.1$ is the most stable configuration, providing an optimal Gaussian width to capture the general distribution of phishing features. However, the model exhibits high sensitivity to specific parameter combinations. At the lowest penalty ($C = 0.001$), the F1-score stagnates at 67.39% for almost all γ values, showing resilience only at $\gamma = 0.1$ (91.44%). This suggests that a weak penalty requires a precisely tuned kernel width to prevent the model from underfitting or failing to capture the underlying structure of the phishing data. Furthermore, while

performance varies across different C configurations, it consistently degrades when $\gamma \geq 10$ compared to lower γ settings. This trend is a clear indicator of overfitting, where high γ values excessively localize the influence of training points, resulting in a complex decision boundary that fails to generalize to unseen data. These findings reinforce that the RBF kernel's superiority is strictly contingent on the synergy between C and γ, as improper calibration leads the model to either fail in learning or lose its predictive power [31][32].

The following is the comparison of the best F1-scores and other metrics achieved by SVM models using each kernel function for phishing website detection.

Table 10. Comparison of the Best F1-Scores and Other Metrics Achieved by SVM Models Using Each Kernel Function

Kernel	Best Parameters	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Linear	{'C': 1000}	92.91%	94.01%	91.91%	92.94%	97.46%
Poly	{'C': 10, 'degree': 5}	94.68%	94.23%	95.38%	94.80%	97.78%
RBF	{'C': 100, 'gamma': 0.1}	94.75%	95.01%	94.64%	94.83%	98.04%
Sigmoid	{'C': 0.001, 'gamma': 0.1, 'coef0': 1}	90.20%	87.45%	94.23%	90.71%	96.34%

As presented in **Table 10**, the RBF model achieved the highest F1-Score of 94.83% and an AUC-ROC of 97.46%, indicating exceptional precision and robustness in distinguishing between phishing and legitimate websites. The Polynomial kernel followed as the close second best performer with an F1-Score of 94.80%, while the Linear kernel maintained competitive performance at 92.94%. Conversely, the Sigmoid kernel exhibited the lowest performance with an F1-Score of 90.71%, suggesting that its mapping function is less suited for the specific high-dimensional characteristics of this phishing dataset. The superior AUC-ROC score of the RBF model (approaching 1.0) further validates its excellent discriminatory power, consistent with the theoretical framework discussed in the literature review.

To further evaluate the discriminatory power of each SVM kernel in distinguishing between phishing and legitimate websites, a comparative analysis of the Receiver Operating Characteristic (ROC) curves is illustrated in figure bellow.

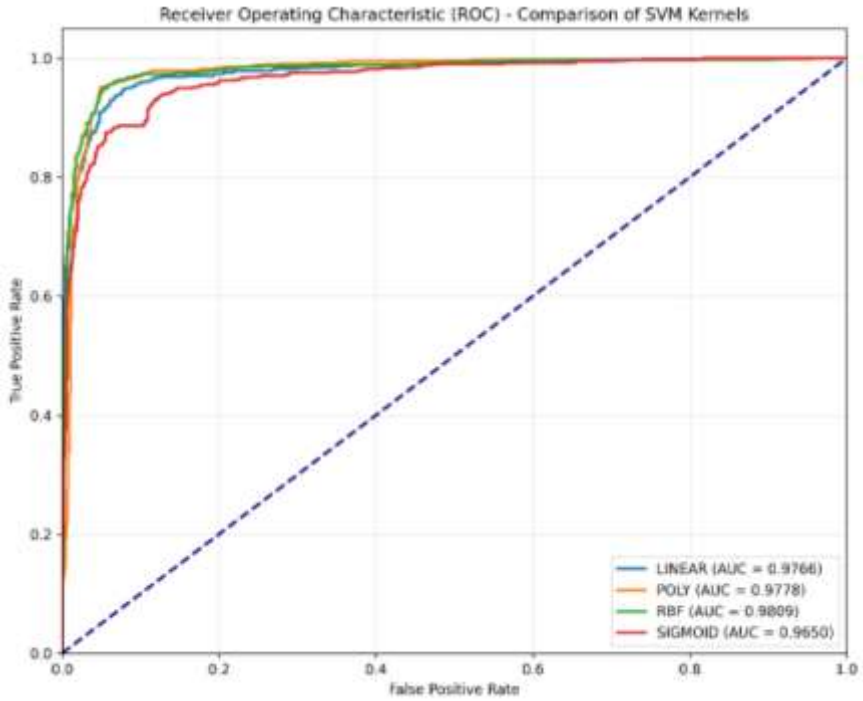

Figure 4. The ROC Comparison of SVM Kernels

As depicted in **Figure 4**, both the RBF and Polynomial kernels exhibit the most optimal performance, with their curves positioned closest to the top-left corner, achieving nearly identical AUC scores. In contrast, the Sigmoid kernel displays the least favorable curve with the lowest AUC value. The consistency between the visual ROC curve positioning and the numerical metrics in the preceding table confirms that both Gaussian and high-degree polynomial mappings are highly effective in capturing the complex feature patterns of the phishing dataset.

The superior performance of the RBF kernel (F1-score of 94.83%) and the Polynomial kernel (F1-score of 94.80%) is mathematically rooted in their ability to map input features into higher-dimensional spaces, providing the necessary flexibility to capture intricate, non-linear correlations within phishing data. While the RBF kernel utilizes an infinite-dimensional mapping, the success of the 5th-degree Polynomial kernel indicates that the relationships among the 10 selected features possess high-order interactions that can be effectively distinguished through polynomial expansion. These findings align with Rumini et al. (2023), who identified the RBF kernel's superiority over the Linear kernel, but this study further highlights that a well-tuned Polynomial kernel can achieve comparable competitive results. Conversely, the Sigmoid kernel exhibited the lowest performance because its hyperbolic tangent mapping—behaving similarly to a two-layer perceptron—failed to align with the underlying distribution of phishing features. Even when configured to meet Mercer's condition, the Sigmoid mapping resulted in a less stable optimization surface, making it ill-suited for this specific classification task.

The practical implications of these findings suggest that for real-world phishing detection systems, both RBF and Polynomial kernels provide a reliable balance between high accuracy and robustness. Given that both achieved F1-scores above 94.8% using only 10 features, security developers have multiple robust options to implement lightweight yet effective filtering layers. While RBF is often preferred for its stability, the Polynomial kernel stands as a potent alternative for capturing complex feature interactions, emphasizing that the

choice of a non-linear kernel is far more critical than the specific function used, as long as it avoids the pitfalls observed in the Sigmoid model.

CONCLUSIONS

This study provides a comprehensive comparison of four SVM kernel functions (Linear, Polynomial, RBF, and Sigmoid) for phishing website detection. The experimental results demonstrate that the RBF and Polynomial kernels achieved comparable and superior performance, with F1-scores of 94.83% and 94.80%, respectively. Given the negligible difference of 0.03% between these two models, both are considered equally effective for capturing the high-dimensional, non-linear patterns inherent in phishing characteristics. In contrast, the Linear kernel (92.94%) and Sigmoid kernel (90.71%) showed lower performance, with the latter exhibiting significant instability under varying penalty parameters.

For practical implementation, this study recommends the use of non-linear kernels (RBF or Polynomial) for phishing detection systems. Practitioners are encouraged to prioritize C values in the range of 10 to 100 and γ values around 0.1 for RBF to avoid the risk of overfitting observed at higher γ levels. Furthermore, the ability of these models to maintain high accuracy using only 10 features suggests that lightweight, real-world security filters can be developed to minimize computational latency while maintaining robust defense.

Despite the promising results, this study acknowledges several limitations. First, the dataset used consists of samples collected up to May 2020, which may not fully represent the most recent evolution of phishing tactics. Second, the feature selection was limited to the top 10 features, potentially excluding nuanced variables that could further improve detection. Additionally, the Sigmoid kernel was tested under a more constrained parameter range, which may have limited its competitive potential.

Future research should focus on evaluating these kernel functions against more recent and diverse phishing datasets to ensure generalizability. Moreover, the integration of automated hyperparameter optimization techniques, such as Bayesian Optimization, and the exploration of hybrid ensemble methods combining multiple kernels could provide further improvements in detection reliability and adaptive security.

REFERENCES

[1] A. Farhansyah, "Mengatasi ancaman phishing di era digital: tantangan dan upaya di Indonesia," KBR. Accessed: Jan. 19, 2025. [Online]. Available: https://kbr.id/ragam/mengatasi-ancaman-phishing-di-era-digital-tantangan-dan-upaya-di-indonesia

[2] T. G. Laksana and S. Mulyani, "Pengetahuan dasar identifikasi dini deteksi serangan kejahatan siber untuk mencegah pembobolan data perusahaan," *J. Jukim*, vol. 3, no. 1, pp. 109–122, 2024, doi: https://doi.org/10.56127/jukim.v3i01.

[3] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Human-centric Comput. Inf. Sci.*, vol. 7, no. 17, pp. 1–13, 2017, doi: 10.1186/s13673-017-0098-1.

[4] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection via feature engineering : Linear and nonlinear space transformation methods," *Inf. Syst.*, vol. 91, p. 101494, 2020, doi: 10.1016/j.is.2020.101494.

[5] K. Shivani, "Detection of phishing website using SVM dan light," *DATA Sci. IOT Manag. Syst.*, vol. 4, no. 4, pp. 51–55, 2025, doi: https://doi.org/10.64751/.

[6] D. Aksu and A. Abdulwakil, "Detecting phishing websites using support vector machine algorithm," *Press. Procedia*, vol. 5, no. 1, pp. 139–142, 2017, doi: 10.17261/Pressacademia.2017.582.

[7] D. Tao, D. Jinran, X. Aidong, X. Chuanmao, L. Boyu, and H. Chao, "Phishing detection system based on the SVM active learning algorithm," in *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, Bangalore: IEEE, 2023, pp. 1–3. doi: https://doi.org/10.1109/SMARTGENCON60755.2023.10442623.

[8] L. Hamel, *Knowledge Discovery with Support Vector Machines*. New Jersey: John Wiley & Sons, Inc, 2009. doi: 10.1002/9780470503065.

[9] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass confusion matrix reduction method and its application on net promoter score classification problem," *Technologies*, vol. 9, no. 4, pp. 81–103, 2021, doi: https://doi.org/10.1145/3453892.3461323.

[10] Z. Lin and L. Yan, "A support vector machine classifier based on a new kernel function model for hyperspectral data," *GIScience Remote Sens.*, vol. 53, no. 1, pp. 85–101, 2016, doi: 10.1080/15481603.2015.1114199.

[11] J. Kolla, S. Praneeth, M. S. Baig, and G. R. Karri, "A comparison study of machine learning techniques for phishing detection," *J. Bus. Inf. Syst.*, vol. 4, no. 1, pp. 21–33, 2022, doi: 10.36067/jbis.v4i1.120.

[12] F. S. Marcello and F. P. Putri, "Improved SVM for website phishing detection through recursive feature elimination," vol. 16, no. 2, pp. 5–9, 2025, doi: https://doi.org/10.31937/ti.v16i2.3744.

[13] 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. 2, 2022, doi: https://doi.org/10.56873/jitu.5.1.4836.

[14] Rumini, Norhikmah, A. Mustofa, and S. Pradana, "Perbandingan pengujian deteksi phising menggunakan metode SVM dengan kernel RBF dan linear," vol. 12, no. September, pp. 754–759, 2023.

[15] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 590–611, 2023, doi: 10.1016/j.jksuci.2023.01.004.

[16] V. I. Santoso, G. Virginia, and Y. Lukito, "Penerapan sentiment analysis pada hasil evaluasi dosen dengan metode support vector machine," *J. Transform.*, vol. 14, no. 2, pp. 79–83, 2017, doi: https://doi.org/10.26623/transformatika.v14i2.439.

[17] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd ed. Birmingham: Packt Publishing Ltd, 2015.

[18] T. Wahyono and Y. Heryadi, *Machine Learning (Konsep dan Implementasinya)*. Yogyakarta: Penerbit Gava Media, 2020.

[19] H. J. Dikkers, "Support vector machines in ordinal classification: An application to corporate credit scoring," *Neural Netw. World*, vol. 15, no. 6, pp. 491–507, 2005.

[20] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer, 2000. doi: 10.1007/978-1-4757-3264-1.

[21] B. Ghojogh, A. L. I. Ghodsi, U. Ca, and M. Crowley, "Reproducing kernel Hilbert Space, Mercer's Theorem, Eigenfunctions, Nystrom Method, and use of kernels in machine learning: tutorial and survey," 2021, doi: https://doi.org/10.48550/arXiv.2106.08443.

[22] Y. Widyaningsih, G. P. Arum, and K. Prawira, "Application of K-fold cross validation in determining the best negative binomial regression model," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 15, no. 2, pp. 315–322, 2021, doi: https://doi.org/10.30598/barekengvol15iss2pp315-322.

[23] J. Allen, H. Liu, S. Iqbal, and D. Zheng, "Deep learning-based photoplethysmography classification for peripheral arterial disease detection : a proof-of-concept study,"

Physiol. Meas., vol. 42, no. 5, p. 054002, 2021, doi: 10.1088/1361-6579/abf9f3.

[24] F. Hilmiyah, "Prediksi Kinerja Mahasiswa Menggunakan Support Vector Machine Untuk Pengelola Program Studi Di Perguruan Tinggi (Studi Kasus : Program Studi Magister Statistika ITS)," M.MT thesis, DEPARTEMEN MANAJEMEN TEKNOLOGI, INSTITUT TEKNOLOGI SEPULUH NOPEMBER, Surabaya, Indonesia, 2017.

[25] M. Imani and H. R. Arabnia, "Hyperparameter optimization and combined data sampling techniques in machine learning for customer churn prediction : A comparative analysis," *Technologies*, vol. 11, no. 6, pp. 1–26, 2023, doi: https://doi.org/10.3390/technologies11060167.

[26] N. Safitri, D. Kusnandar, and S. Martha, "Implementasi algoritma K-nearest neighbor dengan normalisasi Z-score dalam klasifikasi penerima bantuan sosial Desa Serunai," *Bul. Ilm. Math. Stat. dan Ter.*, vol. 13, no. 1, pp. 99–106, 2024, doi: https://doi.org/10.26418/bbimst.v13i1.74063.

[27] A. Hannousse and S. Yahiouche, "Web page phishing detection," Mendeley Data. Accessed: Nov. 12, 2024. [Online]. Available: https://data.mendeley.com/datasets/c2gw7fy2j4/3

[28] S. S. Shafin, "An explainable feature selection framework for web phishing detection with machine learning," *Data Sci. Manag.*, 2024, doi: 10.1016/j.dsm.2024.08.004.

[29] H. Lin and C. Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," *Neural Comput.*, vol. 3, no. 1–32, p. 16, 2003.

[30] H. Chih-Wei, C. Chih-Chung, and L. Chih-Jen, "A practical guide to support vector classification," Taipei, 2003.

[31] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. London: MIT Press, 2018.

[32] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Comput.*, vol. 15, no. 7, pp. 1667–1689, 2003, doi: https://doi.org/10.1162/089976603321891855.