ILKOM Jurnal Ilmiah Vol. 18, No. 2, August 2026, pp. 268-280
Accredited 2nd by KEMDIKTISAINTEK No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

Phishing Email Detection Using SVM with RBF Kernel Based on Manhattan Distance

Asrianda [a,1,*]; Sujacka Retno [a,2]; Beno Jange [b,3]; Mansur [c,4]
[a] *Universitas Malikussaleh, Jl. Cot Tengku Nie, Reuleut, Aceh Utara, 24351, Indonesia*
[b] *STMIK Dharmapala Riau, Jl. Samanhudi No 13, Senapelan, Pekanbaru, 28155, Indonesia*
[c] *Politeknik Negeri Bengkalis, Jl. Bathin Alam, Sungai Alam, Bengkalis, 28712, Indonesia*
[1] *asrianda@unimal.ac.id;* [2] *sujacka@unimal.ac.id;* [3] *beno.jange@lecturer.stmikdharmalpalariau.ac.id;* [4] *mansur@polbeng.ac.id;*
** Corresponding author*

Article history: Received Juni 26, 2025; Revised September 12, 2025; Accepted May 07, 2026; Available online August 10, 2026

Abstract

This study examines the performance of a Support Vector Machine (SVM) model with a Radial Basis Function (RBF) kernel for phishing email detection using Euclidean and Manhattan distance measures. The dataset consists of 3,600 email samples, including 2,400 legitimate emails and 1,200 phishing instances. The features are designed to capture both linguistic and structural characteristics of emails, including word count, vocabulary diversity, stop word usage, number of links and domains, presence of email addresses, spelling errors, and urgency-related terms. The experiments were conducted using two train-test split ratios, 80:20 and 70:30, combined with hyperparameter tuning of C and gamma across 15 iterations. The findings indicate that the Manhattan distance consistently outperforms the Euclidean distance, particularly in terms of recall and F1-score, which are critical for detecting the minority class. The model achieved a best accuracy of 78.33%, accompanied by noticeable improvements in recall and F1-score. These results suggest that the choice of distance function within the RBF kernel plays a crucial role in enhancing model sensitivity and generalization when dealing with imbalanced data. Furthermore, the iterative hyperparameter tuning process contributes significantly to improving both performance and model stability. Overall, the SVM-RBF approach with Manhattan distance provides an effective and reliable framework for phishing email detection in machine learning applications.

Keywords: SVM; RBF Kernel; Phishing Email Detection; Hyperparameter Tuning; Manhattan Distance

Introduction

Phishing emails represent a serious threat in the digital era, often executed through social engineering techniques in which attackers impersonate trusted entities. These emails are designed to appear legitimate in order to obtain sensitive information such as account credentials, credit card details, or to install malware on the victim's device [1]. The techniques used have become increasingly sophisticated, incorporating psychological elements such as impersonating coworkers or internal members of an organization. Research shows that 72% of participants were deceived by phishing emails that appeared to come from their internal IT department [2]. In critical sectors like healthcare, the response rate to phishing emails remains high despite regular training. This indicates that conventional mitigation strategies are no longer sufficient, especially in the face of increasingly complex and adaptive phishing attacks [3].

Although user training and spam filters have been widely implemented, phishing attacks remain highly successful due to the attackers' ability to mimic official communication styles [4]. This situation has worsened with the emergence of artificial intelligence, which can now generate highly realistic and contextual phishing emails [5]. As a result, recent studies have increasingly focused on machine learning approaches for automatically detecting phishing emails by analyzing textual features, metadata, and suspicious links [6], [7]. The applied approaches demonstrate improved performance in enhancing the accuracy of distinguishing between legitimate and phishing emails, particularly in situations where traditional methods struggle to address the increasing complexity of phishing techniques.

Support Vector Machine (SVM) is widely employed in classification tasks due to its capability to handle high-dimensional data and its robustness against overfitting through the principle of structural risk minimization [8]. As a

result, SVM has been applied in various fields, including network intrusion detection [9], disease classification [10], [11], and demand-supply prediction in smart grids [12]. The effectiveness of SVM largely depends on the appropriate selection of kernels and parameters, as their combination determines the model's ability to map nonlinear data into higher-dimensional space [13]. This directly affects the classification accuracy and the model's generalization to new data. Without proper parameter tuning, the model may suffer from overfitting or underfitting, ultimately reducing its ability to accurately classify unseen data.

The Radial Basis Function (RBF) kernel has the ability to project data into a high-dimensional space, making it effective for distinguishing non-linear classes. The conventional RBF kernel based on Euclidean distance assumes that all features contribute uniformly to the distance computation. In the context of phishing email detection, the data is inherently heterogeneous, comprising diverse linguistic and structural features such as textual patterns, URLs, and metadata, which often exhibit different scales and distributions. As a consequence, Euclidean distance may not adequately represent the differences between samples under such conditions [14], [15]. This limitation can reduce the model's ability to capture subtle distinctions between legitimate and phishing emails. Therefore, alternative distance measures that are more robust to feature variability, such as Manhattan distance, are considered more appropriate for representing differences in heterogeneous data environments.

The effectiveness of SVM is highly influenced by the proper selection of kernel functions and parameters. The RBF kernel is often chosen due to its ability to project data into high-dimensional space, which is useful for distinguishing non-linear classes. The RBF kernel based on Euclidean distance may not optimally [16] represent data with heterogeneous feature distributions. The assumption of uniform feature contributions is often inadequate, limiting its ability to accurately capture differences between samples. In contrast, Manhattan distance provides a more effective representation by accounting for absolute differences across features [17], making it more robust in handling non-uniform data variability.

Therefore, developing an RBF kernel based on Manhattan distance is worth further investigation. Several studies have shown that replacing Euclidean distance with Manhattan distance can yield more stable classification performance, especially in non-Gaussian data distributions with heterogeneous feature spreads [18], [19]. The complex structure of phishing email data requires adaptive approaches in distance computation and feature selection to improve model accuracy [20], [21]. This study proposes the integration of Manhattan distance into the RBF kernel within the SVM framework to enhance the representation of heterogeneous data. The proposed approach is designed to improve the model's capability in identifying complex patterns in phishing emails that closely resemble legitimate communication. The findings of this study contribute to the development of a more accurate and stable classification method for phishing email detection.

Method

The stages carried out in this study using the SVM algorithm with an RBF kernel are illustrated in **Figure 1**.

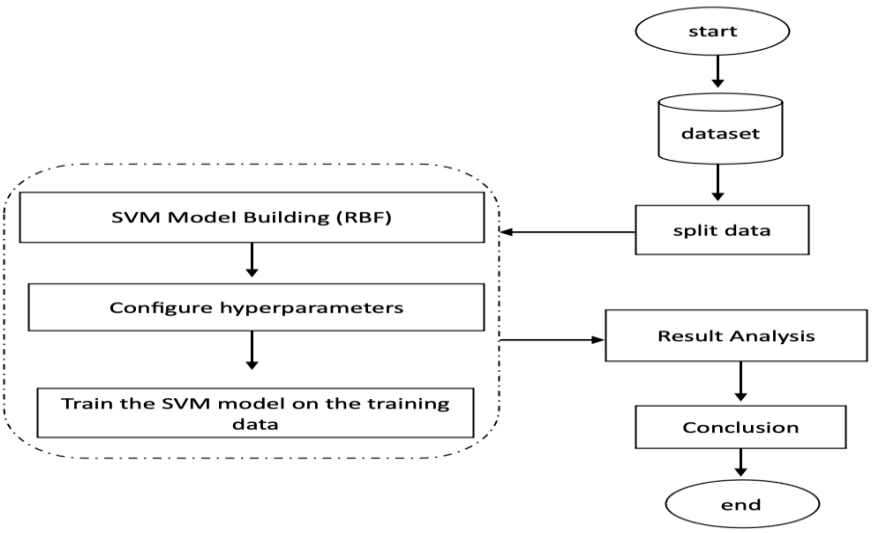

Figure 1. Framework of SVM Algorithm with RBF Kernel

As shown in **Figure 1**, the process begins with dataset selection, followed by pre-processing and splitting the data into training and testing sets. The SVM model with an RBF kernel [22] is then selected and configured by tuning its hyperparameters. The model is trained using the prepared training data, and finally, its performance is evaluated using classification metrics to assess its effectiveness and accuracy.

A. Support Vector Machine (SVM)

SVM works by finding the optimal hyperplane that separates the data into two distinct classes [23]. The main objective is to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors [24], [25]. In an n-dimensional space, the hyperplane is defined as:

$$f(x) = w^T x + b = 0 \tag{1}$$

The vector *x* represents the input features of the data being classified, while *w* is the weight vector that determines the direction and orientation of the separating hyperplane [26], [27]. The term *b* is the bias or intercept, which sets the position of the hyperplane relative to the origin. Together, *w* and *b* form the decision function, denoted as *f(x)*, which decides on which side of the hyperplane a data point lies. If *f(x)* is positive, the data point is classified into one class; if negative, it is classified into the other. Prediction class:

$$predict(x) = \begin{cases} +1, \ if \ w^T x + b \geq 0 \\ -1, \ if \ w^T x + b < 0 \end{cases} \tag{2}$$

The goal of SVM is to maximize the margin between classes, which is equivalent to minimizing $\frac{1}{2}\|w\|^2$, subject to the constraint [28]:

$y_i \in (w^T x_i + b) \geq 1$, for all i

$y_i \in \{-1, +1\}$: represents the class labels.

When the data is not perfectly linearly separable, a soft margin approach is used by introducing [29] a penalty through slack variables ξ. The optimization problem becomes:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{3}$$

subject to:

$$y_i(w^T x_i + b) \geq -\xi_i, \quad \xi_i \geq 0 \tag{4}$$

Here, y_i is the class label for the *i*-th data point, x_i is the input feature vector, *w* is the weight vector that defines the hyperplane direction, and *b* is the bias that positions the hyperplane. The variable ξ_i, known as the slack variable, allows for small violations of the margin constraint. The larger the value of ξ_i, the greater the violation. However, the constraint $\xi_i \geq 0$ ensures that such violations are controlled to maintain good generalization performance.

In the case of using a kernel (non-linear SVM), the dual problem is formulated as:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{5}$$

subject to:

$$0 \leq \alpha_i \leq C, \ \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{6}$$

Equation (5) represents the objective function in the dual form of the SVM algorithm, aiming to find the optimal values for the Lagrange multipliers α_i. These α_i values help determine the support vectors that define the optimal separating hyperplane. The first term $\sum_{i=1}^{n} \alpha_i$ reflects the total contribution from the training data. The second term $\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$ serves as a penalty term, accounting for the relationships between data points based on their class labels and the kernel function $K(x_i, x_j)$. The kernel function enables the mapping of data into a higher-dimensional space, making it linearly separable. The goal of this optimization is to maximize the margin between the two classes and minimize classification errors, resulting in a model with strong generalization capabilities.

The kernel function must satisfy the conditions stated by Mercer's theorem [30], [31]. Several commonly used kernel functions include:

1. Linear Kernel:

$$K(x_i, x_j) = (x_i \ . \ x_j) \tag{7}$$

2. Polynomial Kernel:

$$K(x_i, x_j) = (x_i . x_j + 1)^p \tag{8}$$

3. Gaussian Kernel:

$$K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{9}$$

4. RBF Kernel:

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \tag{10}$$

5. Sigmoid Kernel:

$$K(x_i, x_j) = tanh(\eta x_i . x_j + v) \tag{11}$$

To simplify the Radial Basis Function (RBF) kernel:

$$K(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{12}$$

The squared Euclidean distance $\|x_i - x_j\|^2$ can be expanded as:

$$\|x_i - x_j\|^2 = (x_i - x_j)^T (x_i - x_j) \tag{13}$$

Scalar expansion results in:

$$\|x_i - x_j\|^2 = x_i^T x_j - 2x_i^T x_j + x_j^T x_j \tag{14}$$

Substituting into the RBF formula gives:

$$K(x_i, x_j) = exp\left(-\gamma\left(x_i^T x_i - 2x_i^T x_j + x_j^T x_j\right)\right) \tag{15}$$

To replace the Euclidean distance with Manhattan distance $\|x_i - x_j\|_1$, the formula becomes:

$$\|x_i - x_j\|_1 = \sum_{k=1}^{n} |x_{ik} - x_{jk}| \tag{16}$$

There are two feature vectors, x_i and x_j, each representing data points in a d-dimensional space [32], [33]. The vector x_i consists of components [x_{i1}, x_{i2}, .. , x_{in}], while x_j consists of components [x_{j1}, x_{j2}, .. , x_{jn}]. Each element in the vector represents a feature value in a specific dimension, where n denotes the total number of dimensions or features used in the data representation. This vector representation forms the basis for measuring distances between data points, such as in the calculation of Manhattan [34], [35] or Euclidean distances in machine learning algorithms.

Suppose there are two 3-dimensional vectors:

$$x_i = [x_{i1}, x_{i2}, x_{i3}], \ \ x_j = [x_{j1}, x_{j2}, x_{j3}] \tag{17}$$

Then the Manhattan distance is calculated as:

$$\|x_i - x_j\|_1 = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + |x_{i3} - x_{j3}| \tag{18}$$

For a general case in d dimensions:

$$\|x_i - x_{ju}\|_1 = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + .. \ + |x_{id} - x_{jd}| \tag{19}$$

The RBF kernel can be modified using Manhattan distance as follows:

$$K(x_i, x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|_1\right) \quad (20)$$

Explicitly:

$$K(x_i, x_j) = exp\left(-\gamma \sum_{k=1}^{n} |x_{ik} - x_{jk}|\right) \quad (21)$$

This modified Radial Basis Function (RBF) kernel uses the Manhattan distance, where $\left\| x_i - x_j \right\|_1$ represents the sum of the absolute differences between the corresponding components of the two vectors. This modification offers a more flexible kernel for handling data with varying characteristics.

B. Dataset

In developing an accurate phishing email detection system, the availability and use of a representative dataset play a crucial role. The dataset serves as the primary source for understanding the characteristics and patterns of both legitimate and malicious emails. Through dataset evaluation, various features that distinguish phishing emails from non-phishing ones can be identified and utilized. During the training of a classification model, selecting a dataset with sufficient quantity and clearly labeled data is a critical first step.

num_words	num_unique_words	num_stopwords	num_links	num_unique_domains	num_email_addresses	num_spelling_errors	num_urgent_keywords	label
140	94	52	0	0	0	0	0	0
5	5	1	0	0	0	0	0	0
34	32	15	0	0	0	0	0	0
6	6	2	0	0	0	0	0	0
9	9	2	0	0	0	0	0	0

Figure 2. Phising Email Dataset

As shown in **Figure 2**, the phishing email dataset used in this study consists of 3,600 samples. Of the total, 2,400 samples are labeled as 0, indicating non-phishing emails, while 1,200 samples are labeled as 1, indicating phishing emails. This distribution reveals a significant class imbalance, with the number of legitimate emails being much higher than phishing emails. Therefore, it is essential that any evaluation or development of the detection model takes this imbalance into account to ensure the model remains accurate in identifying the relatively fewer phishing instances.

C. Preprocessing

Data preprocessing was conducted to ensure data quality and feature consistency prior to model training. The process includes data cleaning, handling missing values, and feature normalization. Records containing null values were removed to prevent bias and inconsistencies during the learning process. Numerical data were transformed through normalization to ensure that all features are on a comparable scale. This step is essential, as the SVM model with an RBF kernel is sensitive to variations in feature scales. Therefore, each attribute was adjusted to contribute proportionally during the classification process.

No data balancing technique, such as Synthetic Minority Oversampling Technique (SMOTE), was applied. This decision was made to preserve the original data distribution, allowing the model to be evaluated under realistic class imbalance conditions commonly found in phishing email detection. Through this preprocessing stage, the dataset becomes cleaner, more structured, and suitable for subsequent model training and evaluation.

D. Hyperparameter Configuration

The performance of the SVM model is strongly influenced by the selection of hyperparameters, particularly the C and gamma parameters [36]. These parameters play a crucial role in defining the decision boundary and the generalization capability of the model. The parameter C controls the trade-off between maximizing the margin and minimizing classification errors [37]. A large value of C tends to produce a model with high accuracy on training data but increases the risk of overfitting. In contrast, a smaller C value allows for a wider margin, improving generalization while potentially increasing classification errors.

The gamma parameter determines the influence of individual training samples within the RBF kernel [38]. A high gamma value leads to more complex and localized decision boundaries, whereas a low gamma value results in smoother and more generalized decision boundaries. Therefore, appropriate hyperparameter selection is essential to achieve optimal model performance, particularly when dealing with heterogeneous data.

The tuning process was conducted iteratively by evaluating various combinations of C and gamma values to identify the optimal parameter configuration. This approach enables the model to adapt to non-uniform data distributions, thereby improving both accuracy and performance stability. The tuning procedure was carried out over 15 iterations, with different parameter values applied in each iteration. The optimal configuration was selected based on the highest evaluation performance achieved.

E. Experimental Setup

This study aims to evaluate the performance of the Support Vector Machine (SVM) model with a Radial Basis Function (RBF) kernel using two distance metrics, namely Euclidean and Manhattan. The data preprocessing stage has been described in the previous section. The evaluation was conducted using two data split scenarios, namely 80:20 and 70:30. Model performance was assessed using accuracy, precision, recall, and F1-score metrics. In addition, a confusion matrix was employed to analyze classification errors, providing a comprehensive understanding of the model's capability in detecting phishing emails.

Results and Discussion

The performance of the Support Vector Machine (SVM) model with an RBF kernel was evaluated using both Euclidean and Manhattan distance functions. The evaluation employed several metrics, including accuracy, precision, recall, and F1-score. The main objective was to assess how effectively the choice of distance function could improve the model's ability to detect phishing emails, particularly within an imbalanced dataset. The differences in metric values between the Euclidean and Manhattan distances provide insight into how the choice of distance impacts the model's accuracy in identifying phishing data.

Table 1. Performance Metrics of SVM RBF with Different Data Splits

	Split Data 80:20				Split Data 70:30			
	Euclidean		Manhattan		Euclidean		Manhattan	
Accuracy	0.7653		0.7833		0.7472		0.7583	
class	0	1	0	1	0	1	0	1
Precision	0.75	0.83	0.78	0.81	0.75	0.75	0.77	0.71
recall	0.96	0.38	0.95	0.46	0.94	0.36	0.91	0.46
F1-score	0.85	0.52	0.85	0.59	0.83	0.49	0.83	0.56

In **Table 1**, the evaluation was conducted using two data splitting strategies: 80:20 and 70:30. Based on the 80:20 split, the model achieved an accuracy of 76.53% with Euclidean distance, which improved to 78.33% when using Manhattan distance. For the 70:30 split, accuracy dropped slightly to 74.72% with Euclidean and 75.83% with Manhattan. The evaluation metrics show that Manhattan distance provides more consistent results than Euclidean, especially in terms of precision and recall for the phishing class (class 1). In the 80:20 split, the precision for class 1 reached 0.83 using Manhattan, compared to 0.80 with Euclidean. Recall also improved from 0.38 with Euclidean to 0.46 with Manhattan.

The F1-score for the phishing class increased with Manhattan distance in both data splits—from 0.52 to 0.59 (80:20), and from 0.49 to 0.56 (70:30). These results indicate that using Manhattan distance in the RBF kernel provides a better balance between precision and recall, which is especially important for detecting phishing emails, which are the minority class. For all experiments, the SVM used a regularization parameter of $C = 2.9774$ and a kernel parameter $\gamma = 0.7444$, both determined through hyperparameter tuning to achieve optimal performance.

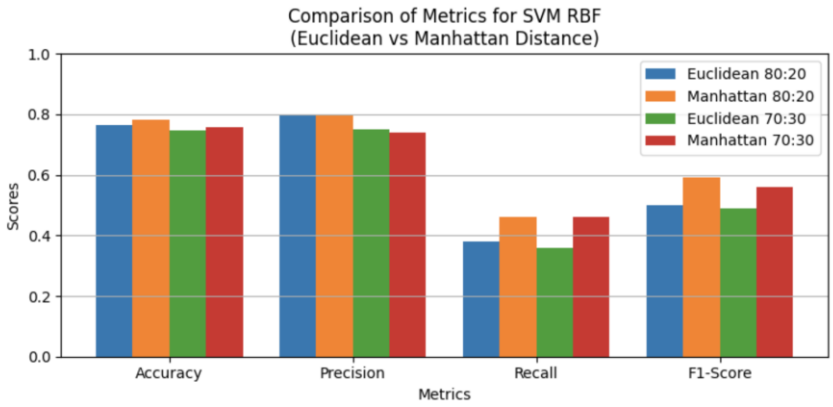

Figure 3. Evaluation Metrics of the SVM RBF Model

Figure 3 illustrates that the use of Manhattan distance consistently delivers superior performance compared to Euclidean distance. This is particularly evident in the recall and F1-score metrics, which are directly related to the model's sensitivity in detecting the phishing class. In the 80:20 data split scenario, the use of Manhattan distance increased the recall from 0.38 (Euclidean) to 0.46, and the F1-score from 0.52 to 0.59. A similar improvement was observed in the 70:30 split scenario. Although the differences in accuracy and precision between the two distance measures were relatively small, the Manhattan-based approach still achieved high overall accuracy. These results suggest that using Manhattan distance in the RBF kernel can enhance the model's generalization ability for minority class data without significantly compromising overall performance. The findings highlight the importance of selecting an appropriate distance function to optimize the performance of SVM models, particularly for phishing email detection where class imbalance is a critical factor.

The results shown in **Figure 3** demonstrate a consistent improvement in model performance when using Manhattan distance compared to Euclidean distance. This improvement is not limited to a single scenario but appears consistently across both tested data split ratios: 80:20 and 70:30. This phenomenon suggests that the model's sensitivity to the minority class is more strongly influenced by the choice of distance function than by the data split proportion. With higher recall and F1-score values, the Manhattan-based approach proves to be more adaptive in handling imbalanced data conditions. This has important implications, highlighting that selecting the appropriate distance function is a strategic factor for improving model effectiveness in real-world applications.

Table 2. Performance Metrics of SVM-RBF Models

Iteration	Split Data 80:20				Split Data 70:30			
	C	gamma	Accuracy		C	gamma	Accuracy	
			Euclidean	Manhattan			Euclidean	Manhattan
1	7.1922	0.0075	0.6750	0.6958	0.5856	0.6472	0.6972	0.7407
2	0.6341	0.3691	0.6861	0.7361	3.3226	0.8169	0.7537	0.7546
3	7.9057	0.3183	0.7472	0.7611	2.6065	0.5805	0.7417	0.7574
4	5.5255	0.8680	0.7583	0.7667	1.3985	0.6297	0.7259	0.7481
5	2.3705	0.3845	0.7306	0.7508	7.0974	0.3096	0.7509	0.7657
6	0.6774	0.8266	0.7125	0.7486	2.9774	0.7444	0.7472	0.7583
7	3.6923	0.9235	0.7528	0.7611	8.9520	0.1523	0.7426	0.7602
8	7.5335	0.4232	0.7556	0.7625	9.4134	0.3318	0.7537	0.7546
9	3.3884	0.0390	0.6889	0.7069	3.7803	0.3957	0.7463	0.7611
10	0.8703	0.6994	0.7167	0.7542	6.6046	0.9254	0.7565	0.7509
11	9.0032	0.5026	0.7569	0.7625	4.2953	0.3670	0.7454	0.7611
12	3.5370	0.7433	0.7514	0.7625	0.1600	0.4131	0.6889	0.6843
13	2.2237	0.5936	0.7417	0.7556	4.3606	0.9853	0.7556	0.7500
14	0.4410	0.0311	0.6750	0.6764	0.6061	0.6209	0.6981	0.7417
15	6.5639	0.5196	0.7542	0.7667	9.9152	0.6637	0.7583	0.7500

Table 2 presents the variations in the regularization parameter (C) and the kernel parameter (gamma), and their significant impact on model accuracy using both Euclidean and Manhattan distance functions. The evaluation was

carried out over 15 iterations using two different data split scenarios. Overall, the use of Manhattan distance showed consistently stable and generally higher accuracy compared to Euclidean distance in most iterations. In the first scenario (80:20 split), the highest accuracy achieved using Manhattan distance was 0.7667 in iterations 4 and 5, while the highest accuracy for Euclidean distance was 0.7583 in iteration 4. A similar pattern was observed in the second scenario (70:30 split), where Manhattan distance reached an accuracy of 0.7657 in iteration 5, outperforming Euclidean distance, which peaked at 0.7583 in iteration 15.

These results indicate that the optimal combination of C and gamma parameters is not fixed, but depends on the characteristics of the dataset and the distance function used. The consistent performance of Manhattan distance across various parameter settings suggests its adaptability to different data conditions. When dealing with class-imbalanced classification tasks, selecting the appropriate parameters and distance function is crucial for optimizing SVM RBF model performance. The sensitivity of these parameters to model accuracy, as shown in the variation across iterations, highlights the importance of hyperparameter tuning. Although the values of C and gamma differed in each test, the relatively stable performance of Manhattan distance indicates that it is a robust approach under changing parameter configurations.

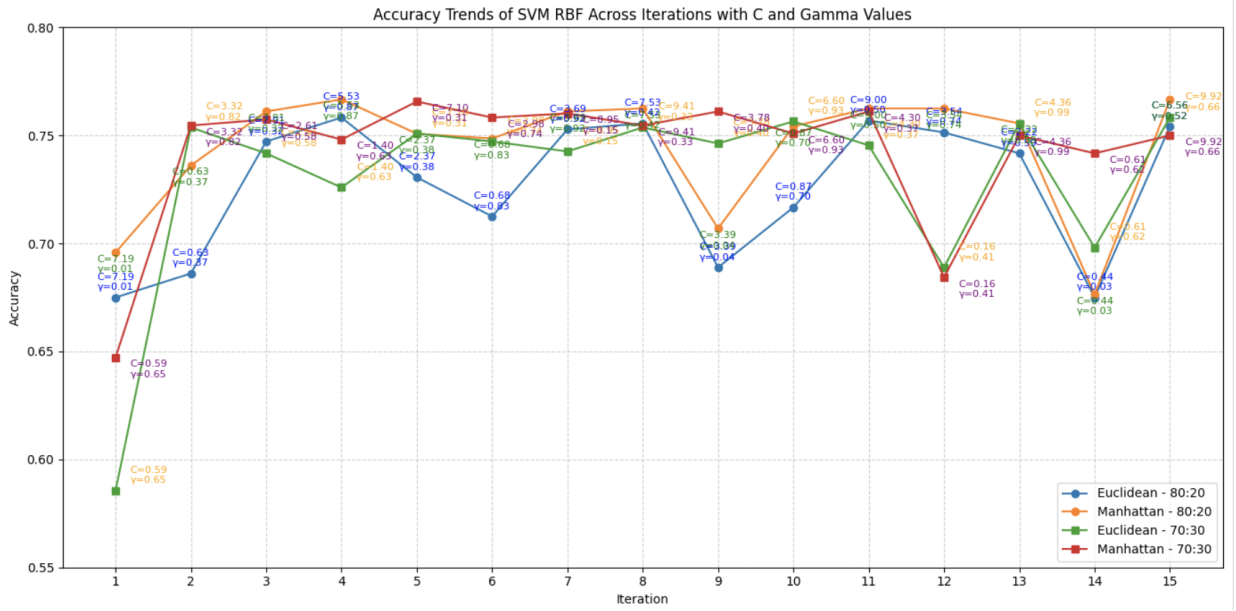

Figure 4. Evaluation Metrics of the SVM RBF Model

Figure 4 presents the accuracy trends of the SVM model with a Radial Basis Function (RBF) kernel across 15 different iterations, each using varied combinations of the C and gamma parameters. The visualization shows that parameter selection has a significant impact on model performance, with noticeable fluctuations in accuracy across the iterations. In both data split proportions (80:20 and 70:30), Manhattan distance consistently yielded slightly higher accuracy compared to Euclidean distance. In iterations 11 and 15, the highest accuracy was achieved with parameter configurations that balanced regularization strength and kernel sensitivity to data variation.

The difference in results between the 80:20 and 70:30 data splits remained relatively stable, with better accuracy patterns observed in the 80:20 ratio. This suggests that having a larger training dataset contributes to the model's ability to form a more reliable decision margin. The success of Manhattan distance is linked to its sensitivity to absolute differences between relevant features, which is particularly beneficial in phishing email detection tasks where feature distributions are non-uniform. Overall, the graph emphasizes the importance of proper parameter optimization and distance function selection in improving classification accuracy and model stability.

1. Error Analysis and Confusion Matrix

Error analysis was conducted using a confusion matrix to examine the model's performance in classifying phishing and non-phishing emails. The results indicate that the model based on Manhattan distance produces fewer misclassifications in the phishing class compared to the Euclidean-based approach.

Table 3. Confusion Matrix Comparison (70:30 Split)

Method	TN (0→0)	FP (0→1)	FN (1→0)	TP (1→1)
Euclidean	676	44	229	131
Manhattan	654	66	195	165

Table 3 presents the performance of the SVM model using Euclidean and Manhattan distance metrics under the 70:30 data split scenario, with hyperparameters set to $C = 2.9774$and $\gamma = 0.7444$. The results show that the Manhattan distance produces fewer false negatives and a higher number of true positives compared to the Euclidean approach, indicating an improved capability in detecting phishing emails. Although a slight increase in false positives is observed, this condition remains acceptable, as failing to detect threats poses a greater risk. These findings suggest that incorporating Manhattan distance into the RBF kernel enhances the model's sensitivity toward the minority class without compromising overall performance stability.

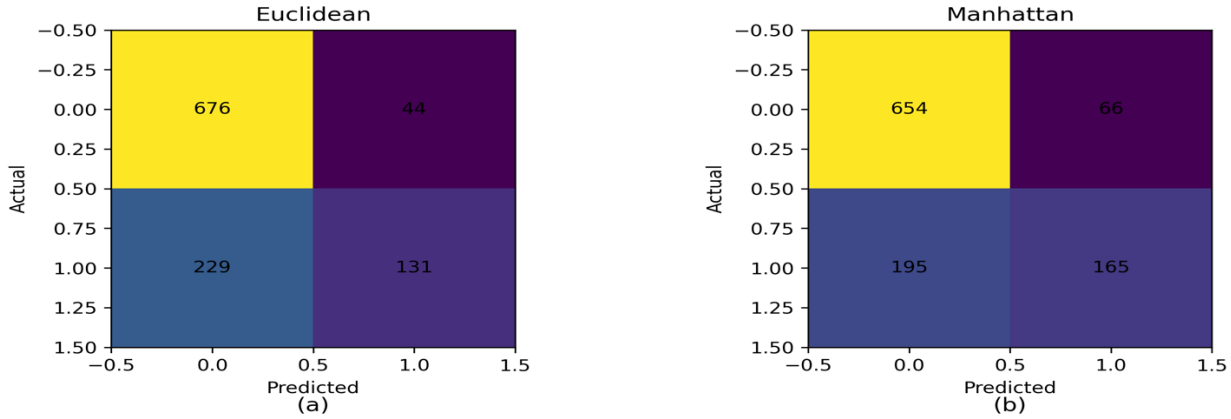

Figure 5. Confusion Matrix of SVM using (a) Euclidean and (b) Manhattan

Figure 5 illustrates the difference in prediction distribution between the two approaches. The model based on Manhattan distance demonstrates an improved ability to capture patterns in the phishing class, as indicated by the increase in correctly classified instances. This suggests that the model becomes more responsive to the characteristics of imbalanced data, thereby reducing critical detection errors.

A shift in misclassification is observed in the non-phishing class; however, the improved identification of phishing emails contributes more substantially to the overall model performance. This indicates that the Manhattan distance tends to prioritize sensitivity toward the security-relevant class. The pattern shown in Figure 5 further supports that the choice of distance metric plays an important role in enhancing model effectiveness when dealing with complex data distributions.

Table 4. Confusion Matrix Comparison (80:20 Split)

Method	TN (0→0)	FP (0→1)	FN (1→0)	TP (1→1)
Euclidean	448	32	150	90
Manhattan	430	50	135	105

The evaluation under the 80:20 data split scenario reveals distinct classification patterns between Euclidean and Manhattan distance metrics, as presented in Table 4. The Manhattan-based approach yields a higher number of correctly classified instances in the phishing class, reflected by an increase in true positives alongside a reduction in classification errors for that class. This indicates an improved ability of the model to capture the characteristics of phishing emails during training.

Although an increase in misclassification is observed in the non-phishing class, this shift does not substantially degrade the overall model performance. On the contrary, the enhanced capability in identifying phishing emails contributes more significantly to the effectiveness of the detection system. These findings suggest that the use of Manhattan distance provides a more adaptive response to imbalanced data distributions, thereby improving the model's sensitivity toward classes associated with higher risk.

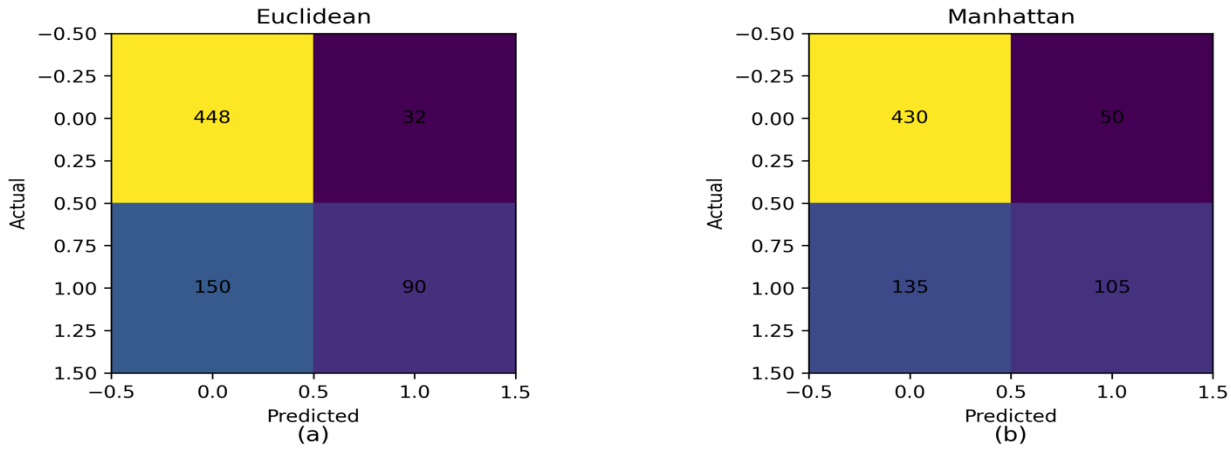

Figure 6. Confusion Matrix of SVM Model (a) Euclidean Distance and (b) Manhattan Distance

The visualization in **Figure 6** (a) and (b) highlights changes in classification outcomes between the two approaches. The Manhattan distance demonstrates an improved ability to identify instances within the phishing class, as reflected by the increase in correctly classified cases and the reduction of critical misclassifications. This pattern suggests that the model becomes more sensitive to the characteristics of imbalanced data distributions.

A shift in prediction distribution is also observed in the non-phishing class, indicating a trade-off in the form of increased misclassification for that class. However, this change does not substantially affect the overall performance, as the improved detection of phishing emails contributes more significantly to the model's effectiveness. These findings indicate that the Manhattan distance tends to guide the model toward greater responsiveness to higher-risk classes.

2. Statistical Validation

Statistical validation was conducted using the Wilcoxon signed-rank test [39] to assess the performance differences between the Euclidean and Manhattan models. Incorporating statistical validation is an essential component in evaluating classification models, as it enhances the reliability and credibility of the reported results.

Table 5. Iterative Performance Comparison of SVM Models (80:20 Split)

Iteration	Euclidean	Manhattan	Difference
1	0.6750	0.6958	0.0208
2	0.6861	0.7361	0.0500
3	0.7472	0.7611	0.0139
4	0.7583	0.7667	0.0084
5	0.7306	0.7508	0.0202
6	0.7125	0.7486	0.0361
7	0.7528	0.7611	0.0083
8	0.7556	0.7625	0.0069
9	0.6889	0.7069	0.0180
10	0.7167	0.7542	0.0375
11	0.7569	0.7625	0.0056
12	0.7514	0.7625	0.0111
13	0.7417	0.7556	0.0139
14	0.6750	0.6764	0.0014
15	0.7542	0.7667	0.0125

The model performance was evaluated over 15 iterations under the 80:20 data split scenario, as presented in **Table 5**, with varying parameter configurations. The results show that the Manhattan-based approach consistently achieves higher accuracy than the Euclidean method across all iterations. The observed performance differences range from 0.0014 to 0.0500, with an average improvement of 0.0176, indicating a stable enhancement in each trial. Furthermore, the Manhattan approach demonstrates superior performance in all iterations, reflecting a high level of consistency.

Table 6. Wilcoxon Signed-Rank Test Results

Metric	Value
Test Type	Wilcoxon Signed-Rank
Number of Iterations (n)	15
Test Statistic (W)	0.0000
p-value	0.000653
Significance Level (α)	0.05
Result	Significant
Interpretation	Manhattan outperforms Euclidean

To ensure that the observed performance differences were not due to random variation, the Wilcoxon signed-rank test was applied to the accuracy results, as presented in Table 6. The test produced a statistic of 0.0000 with a p-value of 0.000653, which is below the predefined significance level ($\alpha = 0.05$). This indicates that the difference between the two evaluated approaches is statistically significant.

The performance improvement obtained from the Manhattan-based approach exhibits a consistent pattern across all iterations. Incorporating Manhattan distance into the RBF kernel not only enhances accuracy but also contributes to improved stability and adaptability under varying data conditions. These findings highlight that the choice of distance metric plays a crucial role in improving model performance, particularly when dealing with complex and imbalanced data characteristics.

Conclusion

The evaluation results indicate that the Manhattan distance consistently delivers better performance than the Euclidean approach across various testing scenarios. The model's ability to detect phishing emails is reflected in the improvement of recall and F1-score, along with a reduction in false negative instances, which are critical in security-related contexts. The iterative evaluation further shows that the Manhattan-based approach produces stable and consistent performance, achieving higher accuracy across all 15 experimental iterations. These findings are supported by the Wilcoxon signed-rank test, which confirms that the performance differences between the two approaches are statistically significant and not due to random variation. Overall, the results highlight that the choice of distance metric within the RBF kernel plays an important role in enhancing model sensitivity, particularly when dealing with imbalanced data.

From a practical perspective, the proposed approach provides a reliable framework for phishing email detection, where minimizing undetected threats is essential. Future research may consider exploring ensemble techniques or hybrid approaches, as well as evaluating the model on larger and more diverse datasets to further improve its generalization capability.

References

[1] W. J. Gordon *et al.*, "Assessment of Employee Susceptibility to Phishing Attacks at US Health Care Institutions," *JAMA Netw. Open*, vol. 2, no. 3, pp. 2–10, 2019, doi: 10.1001/jamanetworkopen.2019.0393.

[2] T. Lin *et al.*, "Susceptibility to spear-phishing emails: Effects of internet user demographics and email content," *ACM Transactions on Computer-Human Interaction*, vol. 26, no. 5, 2019, doi: 10.1145/3336141.

[3] W. Priestman, T. Anstis, I. G. Sebire, S. Sridharan, and N. J. Sebire, "Phishing in healthcare organisations: Threats, mitigation and approaches," *BMJ Health Care Inform.*, vol. 26, no. 1, pp. 1–6, 2019, doi: 10.1136/bmjhci-2019-100031.

[4] P. K. K. Loh, A. Z. Y. Lee, and V. Balachandran, "Towards a Hybrid Security Framework for Phishing Awareness Education and Defense," *Future Internet*, vol. 16, no. 3, 2024, doi: 10.3390/fi16030086.

[5] C. S. Eze and L. Shamir, "Analysis and Prevention of AI-Based Phishing Email Attacks," *Electronics (Switzerland)*, vol. 13, no. 10, 2024, doi: 10.3390/electronics13101839.

[6] R. Brindha, S. Nandagopal, H. Azath, V. Sathana, G. P. Joshi, and S. W. Kim, "Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification," *Computers, Materials and Continua*, vol. 74, no. 3, pp. 5901–5914, 2023, doi: 10.32604/cmc.2023.030784.

[7] E. S. Gualberto, R. T. De Sousa, T. P. B. De Vieira, J. P. C. L. Da Costa, and C. G. Duque, "From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection," *IEEE Access*, vol. 8, pp. 76368–76385, 2020, doi: 10.1109/ACCESS.2020.2989126.

[8] A. Abdiansah and R. Wardoyo, "Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, no. 3, pp. 28–34, 2015, doi: 10.5120/ijca2015906480.

[9] D. Wang and G. Xu, "Research on the detection of network intrusion prevention with SVM based optimization algorithm," *Informatica (Slovenia)*, vol. 44, no. 2, pp. 269–273, 2020, doi: 10.31449/inf.v44i2.3195.

[10] A. M. Elshewey, M. Y. Shams, N. El-Rashidy, A. M. Elhady, S. M. Shohieb, and Z. Tarek, "Bayesian Optimization with Support Vector Machine Model for Parkinson Disease Classification," *Sensors*, vol. 23, no. 4, pp. 1–21, 2023, doi: 10.3390/s23042085.

[11] A. Asrianda, H. Mawengkang, P. Sihombing, and M. K. M. Nasution, "Evaluating the Impact of Model Complexity on the Accuracy of ID3 and Modified ID3: A Case Study of the Max_Depth Parameter," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 5, pp. 3707–3718, Oct. 2025, doi: 10.52436/1.jutif.2025.6.5.4864.

[12] E. Cebekhulu, A. J. Onumanyi, and S. J. Isaac, "Performance Analysis of Machine Learning Algorithms for Energy Demand–Supply Prediction in Smart Grids," *Sustainability (Switzerland)*, vol. 14, no. 5, 2022, doi: 10.3390/su14052546.

[13] A. J. Wathen and S. Zhu, "On spectral distribution of kernel matrices related to radial basis functions," *Numer. Algorithms*, vol. 70, no. 4, pp. 709–726, 2015, doi: 10.1007/s11075-015-9970-0.

[14] K. Shi, H. Qin, C. Sima, S. Li, L. Shen, and Q. Ma, "Dynamic Barycenter Averaging Kernel in RBF Networks for Time Series Classification," *IEEE Access*, vol. 7, pp. 47564–47576, 2019, doi: 10.1109/ACCESS.2019.2910017.

[15] T. C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, 2011, doi: 10.1016/j.engappai.2010.09.007.

[16] S. Seo and W. Kim, "Graph-free kernel discriminant analysis for noise-resilient label spreading," *Knowl. Based. Syst.*, vol. 335, Feb. 2026, doi: 10.1016/j.knosys.2025.115214.

[17] T. Strauss and M. J. Von Maltitz, "Generalising ward's method for use with manhattan distances," *PLoS One*, vol. 12, no. 1, Jan. 2017, doi: 10.1371/journal.pone.0168288.

[18] K. S. Chong and N. Shah, "Comparison of Naive Bayes and SVM Classification in Grid-Search Hyperparameter Tuned and Non-Hyperparameter Tuned Healthcare Stock Market Sentiment Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, pp. 90–94, 2022, doi: 10.14569/IJACSA.2022.0131213.

[19] J. Liu and E. Zio, "SVM hyperparameters tuning for recursive multi-step-ahead prediction," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3749–3763, 2017, doi: 10.1007/s00521-016-2272-1.

[20] Z. Yu, Y. Wang, and Y. Wang, "A Support Vector Machine and Particle Swarm Optimization Based Model for Cemented Tailings Backfill Materials Strength Prediction," *Materials*, vol. 15, no. 6, 2022, doi: 10.3390/ma15062128.

[21] S. Sarkar and K. Mali, "Monkey king evolution (MKE)-GA-SVM model for subtype classification of breast cancer," *Digit. Health*, vol. 10, 2024, doi: 10.1177/20552076241297002.

[22] G. Sharma, A. Panwar, I. Nasiruddin, and R. C. Bansal, "Non-linear LS-SVM with RBF-kernel-based approach for AGC of multi-area energy systems," *IET Generation, Transmission and Distribution*, vol. 12, no. 14, pp. 3510–3517, 2018, doi: 10.1049/iet-gtd.2017.1402.

[23] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "A novel face detection approach using local binary pattern histogram and support vector machine," *2018 International Conference on Advanced Systems and Electric Technologies, IC_ASET 2018*, pp. 28–33, 2018, doi: 10.1109/ASET.2018.8379829.

[24] K. Suksut, K. Kerdprasop, and N. Kerdprasop, "Support Vector Machine with Restarting Genetic Algorithm for Classifying Imbalanced Data," *International Journal of Future Computer and Communication*, vol. 6, no. 3, pp. 92–96, 2017, doi: 10.18178/ijfcc.2017.6.3.496.

[25] C. CORTES and V. VAPNIK, "Support-Vector Networks CORINNA," *J. Phys. Conf. Ser.*, vol. 20, no. 1, pp. 273–297, 1995, doi: 10.1088/1742-6596/628/1/012073.

[26] S. Chen and C. J. Harris, "Design of the optimal separating hyperplane for the decision feedback equalizer using support vector machines," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 5, no. 3, pp. 2701–2704, 2000, doi: 10.1109/ICASSP.2000.861043.

[27] S. Abe, "Are twin hyperplanes necessary?," *Pattern Recognit. Lett.*, vol. 116, pp. 218–224, 2018, doi: 10.1016/j.patrec.2018.10.032.

[28] P. Danenas and G. Garsva, "Selection of Support Vector Machines based classifiers for credit risk domain," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 3194–3204, 2015, doi: 10.1016/j.eswa.2014.12.001.

[29] S. Balasundaram, D. Gupta, and S. C. Prasad, "A new approach for training Lagrangian twin support vector machine via unconstrained convex minimization," *Applied Intelligence*, vol. 46, no. 1, pp. 124–134, 2017, doi: 10.1007/s10489-016-0809-8.

[30] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.

[31] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, no. xxxx, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.

[32] J. Jäger and R. V. Krems, "Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines," *Nat. Commun.*, vol. 14, no. 1, pp. 1–7, 2023, doi: 10.1038/s41467-023-36144-5.

[33] A. Gangopadhyay, O. Chatterjee, and S. Chakrabartty, "Extended Polynomial Growth Transforms for Design and Training of Generalized," pp. 1–14, 2017.

[34] T. Strauss and M. J. Von Maltitz, "Generalising ward's method for use with manhattan distances," *PLoS One*, vol. 12, no. 1, pp. 1–21, 2017, doi: 10.1371/journal.pone.0168288.

[35] B. R. de Oliveira *et al.*, "Selection of Soybean Genotypes under Drought and Saline Stress Conditions Using Manhattan Distance and TOPSIS," *Plants*, vol. 11, no. 21, Nov. 2022, doi: 10.3390/plants11212827.

[36] N. Yu, W. Xu, and K. L. Yu, "Research on Regional Logistics Demand Forecast Based on Improved Support Vector Machine: A Case Study of Qingdao City under the New Free Trade Zone Strategy," *IEEE Access*, vol. 8, pp. 9551–9564, 2020, doi: 10.1109/ACCESS.2019.2963540.

[37] A. Rizwan, N. Iqbal, R. Ahmad, and D. H. Kim, "Wr-svm model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification," *Applied Sciences (Switzerland)*, vol. 11, no. 10, May 2021, doi: 10.3390/app11104657.

[38] L. W. Rizkallah, "Optimizing SVM hyperparameters for satellite imagery classification using metaheuristic and statistical techniques," *Int. J. Data Sci. Anal.*, vol. 20, no. 5, pp. 4945–4962, Oct. 2025, doi: 10.1007/s41060-025-00762-7.

[39] A. Rizwan, N. Iqbal, R. Ahmad, and D. H. Kim, "Wr-svm model based on the margin radius approach for solving the minimum enclosing ball problem in support vector machine classification," *Applied Sciences (Switzerland)*, vol. 11, no. 10, May 2021, doi: 10.3390/app11104657.