IRAQI Academic Scientific Journals

KHAZAYIN OF ECONOMIC
AND ADMINISTRATIVE SCIENCES (2025) (02) (02): P(121-136)

خزائن للعلوم الاقتصادية والإدارية
KHAZAYIN OF ECONOMIC AND ADMINISTRATIVE SCIENCES
ISSN: 2960-1363 (Print)
ISSN: 3007-9020 (Online)

A Particle Swarm Optimization Approach to Fine-Tuning Machine Learning Models for Diabetes Prediction

Shaima Muhealddin Rahman [1], Dr. Mohammad Mahmood Faqe Hussien [2]

[1] *University of Sulaimani, Sulaimani, Iraq*
[2] *University of Sulaimani, Sulaimani, Iraq*
[1]Shaima.rahman@univsul.edu.iq ، [2]Mohammad.faqe@univsul.edu.iq

Abstract: Diabetes has been calculated as one of the greatest common health problems that several people are at risk of growing. Diabetes classification is a difficult work that needs accurate and effective Machine Learning models to help clinical choice-making. This paper examines the optimization of Support Vector Machine models utilizing Particle Swarm Optimization to find the model hyperparameter to increase predictive performance. Different test sizes 0.2, 0.3, and 0.4 were assessed, and the models were tested using two kernel functions: Radial Basis Function and polynomial. The results display that the Support Vector Machine-Particle Swarm Optimization hybrid method substantially enhances accuracy, Areia Under Curve, and decreases Mean Squared Error compared to the classical Support Vector Machine model. The Support Vector Machine-Particle Swarm optimized model, utilizing the kernel polynomial, earns a steady accuracy of 0.99 over all test sizes, surpassing the classical Support Vector Machine with the Radial Basis Function kernel, which had a decrease in accuracy ranging from 0.84 to 0.90. However, the Areia Under Curve values in the optimized model were steadily high, achieving 1.000 at a test size of 0.4, showing the model's higher classification ability. These results recommend that PSO is an efficient optimization technique for optimizing SVM parameters, leading to enhanced performance in diabetes classification.

Keywords: Machine Learning, Particle Swarm Optimization, Support Vector Machine, Diabetes, classification.

DOI: 10.69938/Keas.25020210

نهج تحسين أسراب الجسيمات لضبط نماذج التعلم الآلي بدقة للتنبؤ بمرض السكري

شهيما محي الدين رحمن [1]، د. محمد محمود فقي حسين [2]
[1] جامعة السليمانية، السليمانية، العراق
[2] جامعة السليمانية، السليمانية، العراق
[1]Shaima.rahman@univsul.edu.iq ، [2]Mohammad.faqe@univsul.edu.iq

الملخص: لقد تم تصنيف مرض السكري كواحد من أكثر المشاكل الصحية شيوعًا والتي يكون العديد من الأشخاص معرضين لخطر الإصابة بها. يعد تصنيف مرض السكري مهمة صعبة تتطلب نماذج تعلُّم آلي دقيقة وفعالة للمساعدة في اتخاذ القرارات السريرية. تبحث هذه الورقة في تحسين نماذج آلة الدعم المتجه الداعم باستخدام تحسين سرب الجسيمات للعثور على المعلمات الفائقة للنموذج لزيادة الأداء التنبئي. تم تقييم أحجام الاختبار المختلفة 0.2 و 0.3 و 0.4، وتم اختبار النماذج باستخدام دالتين أساسيتين: دالة الأساس الشعاعي والمتعددة الحدود. تظهر النتائج أن طريقة التحسين الهجينة لآلة دعم المتجهات - سرب الجسيمات تعمل على تحسين الدقة

والمساحة تحت المنحنى بشكل كبير وتقلل من متوسط مربع الخطأ مقارنة بنموذج آلة دعم المتجهات الكلاسيكي. حصل نموذج آلة دعم المتجهات - سرب الجسيمات المحسن، الذي يستخدم متعدد الحدود الأساسي، على دقة ثابتة تبلغ 0.99 على جميع أحجام الاختبار، متجاوزًا نموذج آلة دعم المتجهات الكلاسيكية مع نواة دالة الأساس الشعاعي، والتي شهدت انخفاضًا في الدقة يتراوح من 0.84 إلى 0.90. ومع ذلك، كانت قيم AUC في النموذج المحسن مرتفعة ، حيث وصلت إلى 1.000 عند حجم اختبار 0.4، مما يدل على قدرة التصنيف الأعلى للنموذج. تشير هذه النتائج إلى أن PSO هي تقنية تحسين فعالة لتحسين معلماتSVM ، مما يؤدي إلى تحسين الأداء في تصنيف مرض السكري.

الكلمات المفتاحية: التعلم الآلي، تحسين أسراب الجسيمات، آلة الدعم المتجه، مرض السكري، التصنيف.

Corresponding Author :Shaima.rahman@univsul.edu.iq

1 Introduction

One of the most common health problems that several people in both improved and growing countries are at risk for is diabetes. Diabetes is a situation in which the body is incapable of digesting glucose, or blood sugar. As an outcome, blood glucose levels increase to risky high levels. The body cannot create enough insulin in this state. Another possibility is that the body is unable to respond to the insulin that is made. Serious issues such as nerve damage, heart attacks, kidney failure, and stroke may happen in people with this health issue. Statistics from 2017 display that 8.8% of people worldwide had diabetes in 2015. By 2030, that number is expected to increase to a percentage between 11.3% and 12.2% (Jibril, Haruna & Jiangsheng, 2023).
Through the use of machine learning (ML), tools may be skilled to process data additional efficiently. Sometimes, after observing the data, we are incapable of interpreting the explanation that has existed taken from it. In that state, we use machine learning. Numerous businesses use ML to extract related data; the aim of ML is to study from that data. This procedure needs large data sets and is resolved by several mathematicians and programmers utilizing a range of methods. (Mahesh, B., 2020).
Support Vector Machine (SVM) is lone of the most favoured cutting-edge ML techniques. SVM is model-supervised learning that analyses data for classification and regression utilizing linked learning methods. By indirectly mapping their entries into high-dimensional attribute spaces, a method famous as kernel deception, SVM is able to effectively do non-linear classification in supplement to linear classification. In nature, it generates limits among the classes. The edges are produced to minimize the classification error by growing the largest possible space among the margins and the classes (Saputra, Dharmawan, and Irmayani, 2022).
Selecting the optimal option from a scope of options is the aim of optimization, a significant element of decision-making in several different domains. The accurate method and the metaheuristic method are the two main methods utilized to resolve optimization issues. Accurate methods, like dynamic programming and branch-and-bound, confirm the best outcomes but are mathematically expensive for more complex problems. Metaheuristics, Alternatively, are more efficient for big, complex problems. algorithms community-situated like algorithms genetic or algorithms particle swarm optimization (Rahman & Rashid, 2021).
The essential ideas of the particle swarm optimization algorithm (PSO) are drawn from the research of artificial life and hunter-gathering behavior. Imagine an environment in which a group of birds are randomly searching for food. There is only one food source in the area, and while all birds are informed of its location, they are uninformed of their length from it. For the birds to discover food in

the easiest and most efficient method possible, they must search the area where the closest bird is to the food (Zhou & Liao, 2013).

The aim of this paper is to create a model optimized for ML for diabetes forecasting by improving the parameters of classical SVM by PSO to decrease the dimensionality of attributes and select the optimal attributes that will increase the model's accuracy. The created results will be compared with the results of classical SVM to confirm the ability of the optimized SVM.

2 Support Vector Machine (SVM)

SVM goals are to discover the best hyperplane. A hyperplane is defined by the Equation 1 that divides two data point classes. The space is divided into two areas by this hyperplane: the half-space that is positive (for the first class, ω+) as well as the negative half-space (for the second class, ω−). SVM creates two margin planes, H1 and H2, and maximizes the margin between the nearest samples by improving the values of b, which represent the bias or threshold, and w, which represent a weight vector. (Wang, 2005; Tharwat, 2019). Support vectors are the data items that are nearest to the choice outer layer or hyperplane. Hyperplane: In two-spatial spaces, this hyperplane is a line separating a plane into two parts where every class lies on one of two sides. Margin: By identifying the hyperplane that optimizes the margin between the two classes, an SVM may do classification (Khalil, Faqi & Haider, 2022).

$$\mathrm{w}^T x_i + b = 0 \qquad \dots(1)$$
$$H_1 \rightarrow \mathrm{w}^T x_i + b = +1 \quad for \quad y_i = +1 \qquad \dots(2)$$
$$H_2 \rightarrow \mathrm{w}^T x_i + b = -1 \quad for \quad y_i = -1 \qquad \dots(3)$$

Where xi is ith m-dimensional training data, yi is class label 1 or -1 for input xi for pattern classification, and a scalar function approximation output, w stands for a vector weight, b stands for the limit or bias. Where $\mathrm{w}^T x_i + b \geq +1$ is the plane for the positive class and $\mathrm{w}^T x_i + b \leq -1$ represents the plane for the negative class. These two equations are able to be merged as follows.

$$y_i(\mathrm{w}^T x_i + b) - 1 \geq 0 \ \forall \ i = 1,2,\dots,N \qquad \dots(4)$$

The sum of d1 and d2 is shown by the SVM margin (M) as follows.

$$M = d1 + d2 = \frac{2}{||w||} \qquad \dots(5)$$

Where d1 and d2 act for the space since the first and second plane, sequentially, to the hyperplane, and d1 = d2 SVM (Khalil, Faqi & Haider, 2022).

$$\min \frac{1}{2} ||w||^2 \qquad \dots(6)$$
$$y_i(\mathrm{w}^T x_i + b) - 1 \geq 0 \quad \forall \ i = 1,2,\dots,N \qquad \dots(7)$$
$$\text{When } ||w||^2 = \mathrm{w}^T W \qquad \dots(8)$$

Quadratic programming issues are shown by Equation 4, which may be structured into the Lagrange equation by merging the aim function Equation 7 and the constraint Equation 4 as follows.

$$minLp = \frac{||\mathrm{w}||^2}{2} - \sum_i \alpha i(\, y_i(\mathrm{w}^T x_i + b)) - 1$$

$$= \frac{||\mathrm{w}||^2}{2} - \sum_i \alpha i\, y_i(\mathrm{w}^T x_i + b) + \sum_{i=1}^{N} \alpha i \qquad \ldots (9$$

Where LP denotes the primal problem and αi is the Lagrange multiplier for. By separating LP together with regard to w and b and setting the derivatives to zero, the values of b, w, and α that reduce LP in Equation 9 are resolved.

$$\frac{\partial Lp}{\partial w} = 0 \ \rightarrow w = \sum_{i=1}^{N} \alpha i\, y_i x_i \qquad \ldots (10)$$

$$\frac{\partial Lp}{\partial b} = 0 \ \rightarrow w = \sum_{i=1}^{N} \alpha i\, y_i = 0 \qquad \ldots (11)$$

By replacing Equations 10 and 11 into Equations 9, the binary issue is able to be written as follows

$$maxLD = \sum_{i=1}^{N} \alpha i - \frac{1}{2} \sum_{i,j} \alpha i \alpha j y_i y_j \mathrm{x}^T X \qquad \ldots (12)$$

$$\alpha i \geq 0, \sum_{i=1}^{N} \alpha i\, y_i = 0 \forall\ i = 1,2, \ldots, N \qquad \ldots (13)$$

αi the Lagrange multiplier for x_i LP is linear programming where LD stands for LP's binary form. The values of b, w, and α are discovered by resolving Equation 5, 6, and 7. Since the majority of αs in SVM are zeros, scarcity is an ordinary SVM characteristic. Support vectors, which are the samples nearest to the dividing hyper plane, are shown by the non-zero α‟s; as a result, SVs were can earn the maximum width margin (Khalil, Faqi & Haider, 2022).

3 Particles Swarm Optimization (PSO)

PSO takes inspiration from the communal behaviour noticed in bird gathering. In this method, every particle in the swarm symbolizes a possible answer, possessing both a velocity and location that are improved in every iteration. Particles modify their velocity based on their individual best location and the best location found by the swarm as a whole. The location of every particle is then refreshed in line with this new velocity. This repetitive process allows particles to efficiently discover the solution space and gradually move towards the optimal solution (Cho, M. Y., & Hoang, T. T. 2017).

$$s^n[t+1] = w * s^n[t] + c_1 * r_1 * (b^{n,best}[t] - b^n[t]) + c_2 ** r_2 * (b^{l,best}[t] - b^n[t]) \quad \ldots (13)$$

$$b^n[t+1] = b^n[t] + s^n[t+1]; t = 1,2,3, \ldots, T \quad \ldots (14)$$

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{T}\right) * t \quad \ldots (15)$$

Where, n = 1,2, …, N, N is a swarm community number. $s^n(t)$ is the velocity vector in $[t]$ th iteration. $b^n(t)$ shows the nth particle's present position. $b^{n,best}$ is the optimal location of the particle. The optimal swarm location is $b^{l,best}$ (Venkatesan, Saideepika and Sarathambekai, 2024) Below the state of the n-th particle at the t-th iteration, $b^n[t]$ and $s^n[t]$ are the n-th are the n-th position and velocity element. Coefficient positives c_1, c_2, r_1 and r_2 are the random numbers; the scope is 0 to 1, and v is the weight inertial of algorithm PSO. The algorithm PSO may agree with the continuous optimization issue and help multi-point search. Consequently, we receive the algorithm PSO decide the parameters of SVM to enhance the performance of model SVM (Du et.al. 2017).

4 Model Analysis

The achievement of the models trained was then evaluated utilizing a number of metrics, like accuracy, recall, precision, specificity, F1-score, and Area Under Curve (AUC).

4.1 Accuracy

Accuracy evaluations the general accuracy of a model's prediction. The mathematical structure of accuracy is displayed in Equation 16.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FP}} \quad \ldots (16)$$

True Positives (TP): Occurrences where the model exactly determined an outcome positive.
True Negatives (TN): Occurrences where a negative outcome was accurately identified by the model.
False Positives (FP): incidents that are factually unfavourable but incorrectly predicted as good.
False Negatives (FN): Examples that are wrongly categorized as negative but are actually positively expected.

4.2 Precision

Precision represents the ratio of genuine positives within all of the positive model's forecasts. The statistical structure of precision is presented in Equation 17.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \ldots (17)$$

4.3 Recall (also known as sensitivity or true positive rate)

Recall shows the percentage of real positive cases that are real positives. The statistical structure of this measure is displayed under Equation 18.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \ldots (18)$$

4.4 F1-Score

F1-Score is the harmonic means of precision and recall. It offers a balanced measure between the two. The F-score has a maximum value of 1 and a minimum value of 0. It can be understood as a

weighted harmonic means of precision and recall. The statistical structure of F1-Score is presented in Equation 19.

$$\text{F1\_score} = \frac{2 \times \text{ precision } \times \text{ recall}}{\text{precision } + \text{ recall}} \qquad \ldots (19)$$

4.5 Specificity (True Negative Rate)

Particularity measures the ratio of true negatives to all negative state The statistical structure of this metric is as follows. show in Equation 20 (Venkatesan, Saideepika and Sarathambekai, 2024).

$$\text{Specifity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \qquad \ldots (20)$$

4.6 Confusion Matrix of Classical SVM

A confusion matrix, which shows the numbers for TP, TN, FN, and FP, is a metric measure used to characterize how well a classification model performs on datasets with known true values (Jibril, Haruna & Jiangsheng, 2023).

4.7 Receiver Operating Characteristic

Receiver operating characteristic (ROC) is an optical showcase that differences the FP rate with the TP rate. It shows an exchange of sorts among specificity and sensitivity. Charting a ROC produces the AUC Score, which intervals from 0 to 1. An approach with an AUC mark of 1 is seen as perfect, whereas models with an AUC score of 0.5 or below are regarded as useless. The ROC curve is another metric for characterizing a model's performance, and it's especially helpful for binary classifiers. A decent classifier keeps as far off from the spotted line (toward the upper-left corner) as possible in the ROC curve under the model SVM, which most likely displays the curve ROC of an absolutely classifier random (Jibril, A.U., Haruna, K., and Jiangsheng, Z. 2023).

5 Methodology

5.1 Description of the Dataset

To assess the classification precision of the suggested approach, the dataset utilized in this paper was got from the Centre for Diabetes in Sulaymaniyah, from files of patients. The dataset consists of 251 cases that were checked for diabetes. The dataset includes a total of 15 characteristics that were employed in the analysis of diabetes There are 76 negatives containing Class 0 and 175 positives containing Class 1 examples in the sample overall. The following features, as shown in Table 1, one target class feature that indicates whether a test result was positive or negative is the output variable other attribute is a total of 14 for the input variable.

Table 1: Dataset Description

Attribute	Data Type	Possible Values

Gender	**Categorical**	**"Male", "Female"**
Duration of Diabetes Mellitus	**Numeric**	**Continuous (<1 year,1-5-year,5-10 year,>10 year)**
Age	**Numeric**	**Continuous (e.g., 18 to 100 years)**
Smoker	**Categorical**	**"Yes", "No"**
Alcohol	**Categorical**	**"Yes", "No"**
Family History of Diabetes	**Categorical**	**"Yes", "No"**
Hypertension	**Categorical**	**"Yes", "No"**
Heart Disease	**Categorical**	**"Yes", "No"**
Heigh	**Numeric**	**Continuous (e.g., 140 to 190)**
weight	**Numeric**	**Continuous (e.g., 50 to 140)**
BMI (Body Mass Index)	**Numeric**	**Continuous (e.g., 18.5 to 40.0)**
Fasting/Random	**Categorical**	**Fasting/Random**
Blood Sugar (mg/dL)	**Numeric**	**Contain between (90 mg/dL -520 mg/dL)**
HbA1c (%)	**Numeric**	**Continuous (e.g., 4.0% to 15.0%)**
Target Variable (Outcome)	**Categorical**	**"Negative", "Positive"**

5.2 The Suggested Methods

The significant aim of the suggested SVM-PSO method is to select the superior optimal parameters gamma and C for training SVM in order to enhance performance. Figure 2 shows the recommended algorithm's methods and diagram.

In this paper, we assessed the achievement of classical SVM and an optimized SVM utilizing SVM-PSO for diabetes classification. The classical SVM model was applied using two functions kernel, polynomial (poly), and Radial Basis Function (RBF), with default hyperparameter settings, where organize the parameter, the parameter C was set to the 'default' is 1, and the coefficient gamma was set to ‘scale'. To improve the achievement of SVM, the algorithm PSO was used to improve the hyperparameters, looking for optimal values of C and gamma to increase classification accuracy. The dataset was split into separate test sizes: 0.2, 0.3, and 0.4, equivalent to 80% training-20% testing, 70% training-30% testing, and 60% training-40% testing, in order. The models were evaluated based on accuracy, AUC, and MSE to compare their effectiveness.

The technique suggested was carried out and estimated utilizing the language programming Python and the ML tools in the library scikit-learn.

5.3 Optimization SVM Model Utilizing PSO

Under the algorithm, a step-by-step testing method to provide sufficient insight into what requirements to be while optimizing the parameters of SVM utilizing PSO:

Step 1: Download the data. Step 2: Get the data prepared. Step 3: Divide the data into groups for training and testing. Step 4: Set the PSO settings after reading the training set. Step 5: Initialized sets of SVM parameters W, C1, and C2 within the interval of location and velocity. Step 6: Using training data to form SVM. Step 7: Choose the best particle by assessing each one's fitness. Step 8: Set repetition number k=1. Step 9: If k max repetition, next k=k+1 and run to step 7; another run to step 10. Step 10: Best resolution achieved: type the outcome of the perfect result as best. Step 11: Duplicate train the SVM utilizing the optimal parameters and attributes, after using the testing data to find unknown samples. Step 12: End. The flowchart in Figure 2 illustrates these steps.

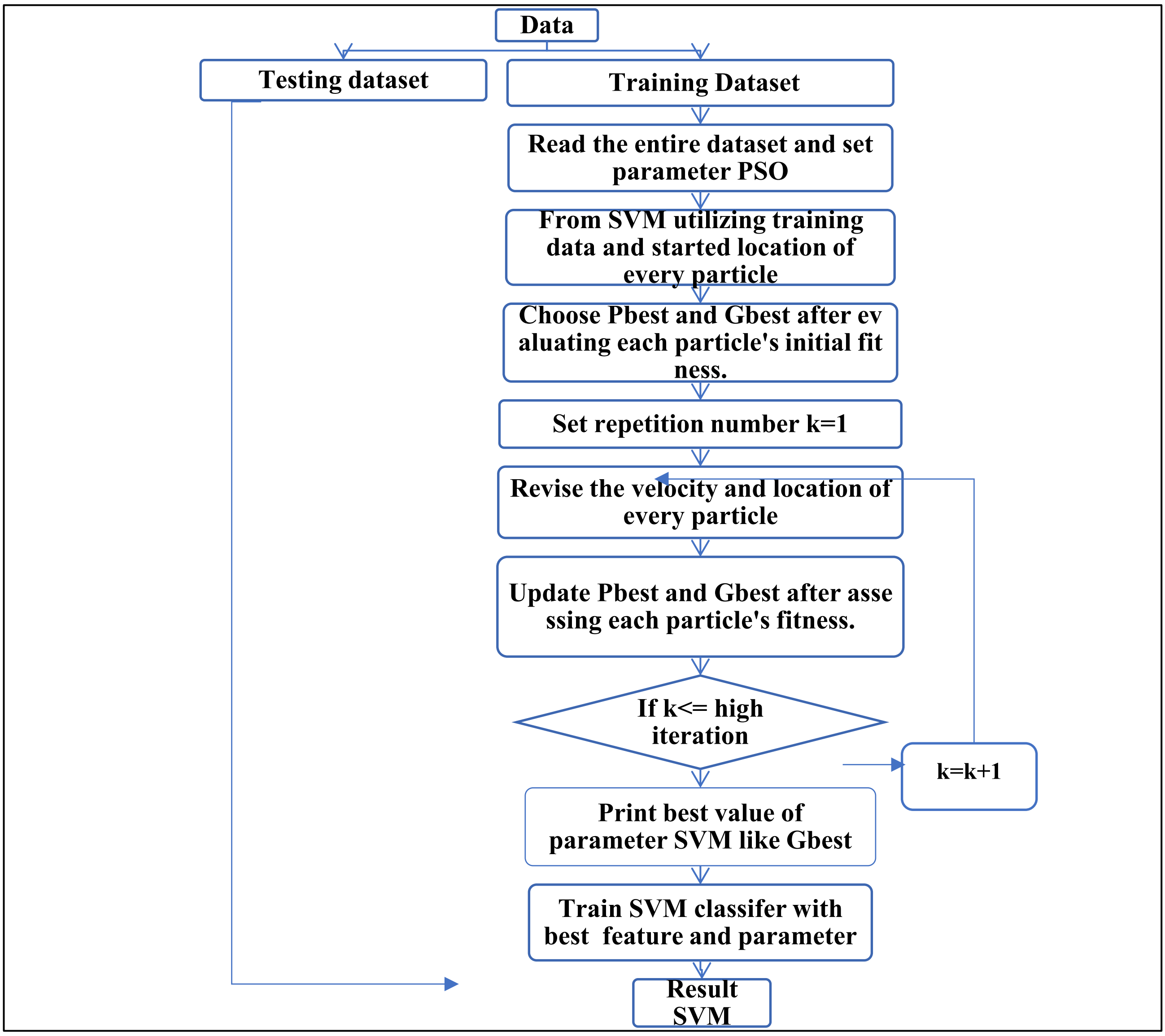

Figure 2: SVM-PSO Flowchart

6 Results and Discussion

6.1 Accuracy of Classical SVM Model Results

The accuracy, AUC, and MSE of the classical SVM are displayed in Table 2. The accuracy of the model, as shown in the table, are 0.84, 0.85, 0.86, and 0.90. Which means that the approach was can to accurately forecast 0.84, 0.85, 0.86, and 0.90 of the varying test size cases. Furthermore, the model has a near-perfect classification performance with an AUC score of 0.9556, 0.9537, 0.9403, 0.9385, 0.9545, and 0.9560, indicating a good capacity to distinguish between the classes. However, there are until now some prediction errors the model's predictions are fairly near to the real values, as indicated by the Mean Squared Error (MSE) of 0.0980, 0.1579, 0.1386, and 0.1485. For each test size, the model generally earns higher accuracy, with values of 0.90 or less. The 'RBF' kernel especially yields a little better accuracy and AUC values compared to the 'poly' kernel, with the highest AUC being 0.9556 for a test size of 0.2. But the MSE is lower for the test size 0.2 compared to the test sizes 0.3 and 0.4. Generally, the 'RBF' kernel gives more stable and little better performance in terms of accuracy and AUC through different test sizes, when the 'poly' kernel displays a marginally decreased but still fixed competitive performance.

Table 2: Accuracy of Classical SVM Model

Test size	C	Gamma	Kernal	Accuracy	AUC	MSE
0.2	**Default=1**	**Scale**	**RBF**	0.90	0.9556	0.0980
			poly	0.90	0.9537	0.0980
0.3	**Default=1**	**Scale**	**RBF**	0.84	0.9403	0.1579
			poly	0.84	0.9385	0.1579
0.4	**Default=1**	**Scale**	**RBF**	0.86	0.9545	0.1386
			poly	0.85	0.9560	0.1485

6.2 Classification Report of Classical SVM Model

Classification Report of classical SVM Model of the test size is 0.2 and kernel RBF the majority of cases are correctly classified by the model, which has an accuracy of 0.90. Class 1 exhibits a superior balance between precision 0.74, recall 1.00, and F1-score 0.85, whereas Class 0 has excellent precision 1.00 but low recall 0.74. The weighted averages precision: 0.93, recall: 0.90, and F1: 0.91 point to improved performance for the larger class, class 1, while the macro average scores precision: 0.93, recall: 0.90, and F1: 0.91 show strong performance.

Table 3: Classification Report of Classical SVM Model (test size=0.2, kernel= RBF)

	precision	recall	f1_score	support
0	0.74	1.00	0.85	14
1	1.00	0.86	0.93	37
accuracy			0.90	51
marco avg	0.87	0.93	0.89	51
weight avg	0.93	0.90	0.91	51

6.3 Confusion Matrix of Classical SVM Model

In Figure 3, the underside-right value 32 shows the number of TP is class 1, showing that 32 cases were exactly classified as positive. The uppermost-left value 14 shows the number of true negatives is class 0, which means 14 cases were exactly classified as negative. The uppermost-right value 0 shows FP, meaning no negative situations were incorrectly classified as positive, and the underside-left value 5 shows FN, where 5 positive situations were wrongly classified as negative. In general, the model exactly determines 14 negative and 32 positive cases, displaying a slightly strong performance.

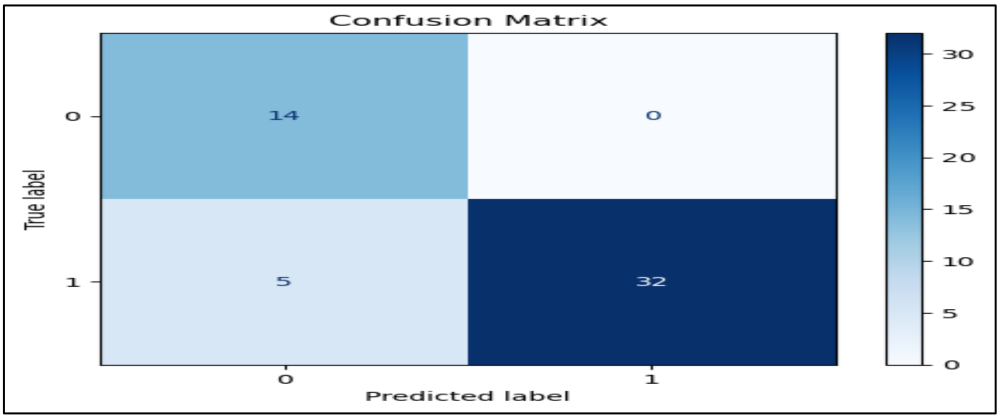

Figure 3: Classical SVM Confusion Matrix (test size=0.2, kernel= RBF)

6.4 ROC Curve of Classical SVM Model

The ROC curve evaluates the performance of SVM classifier. In Figure 6, the orange curve shows the classifier's performance over different thresholds, with the true positive score plotted opposed to the false positive ratio. As shown in the Figure 6, the high AUC value of 0.9556 indicates excellent classification performance, as it is close to the ideal value of 1.0. The dashed random guessing is represented by the diagonal line. As presented in the figure, the SVM's curve stays far above the diagonal line, which reflects the strong discriminatory force among the classes negative and positive in the dataset.

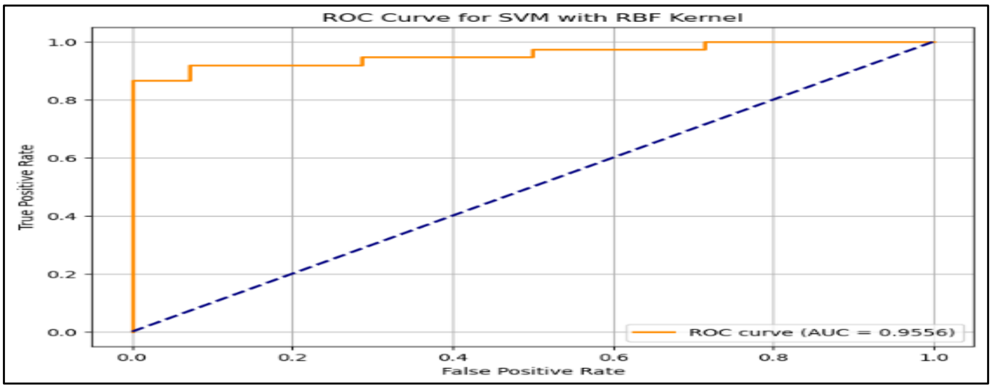

Figure 4: Classical SVM ROC curve (test size=0.2, kernel=RBF)

7 SVM Classification with PSO (SVM-PSO)

In this step, PSO was used to specify the C and gamma parameters of SVM. The parameters of PSO, C1 (cognitive component) and C2 (social component), were set so that C1 and C2 are the same. It is equal to 2. C1: This constant regulates how much the particle is attracted to its own best-known position. C2: This constant regulates how much the particle is attracted to the swarm's best position. And uses varying weights of 0.2, 0.4, 0.6, 0.8, and 1. Weight commonly indicates the inertia weight, which is a key parameter utilized to balance the exploitation and exploration skills of the algorithm. By PSO hyperparameter C and gamma identify The C and Gamma parameters are then entered as input parameters to determine the accuracy of the SVM classifier.

In Table 4, the data collection is split into 0.2, with PSO utilized to optimize the hyperparameters of the model SVM. The outcomes display that the kernel poly continuously outperforms the kernel RBF in conditions of AUC and accuracy. The kernel poly succeeds in a remarkable AUC of 0.998 and accuracy of 0.98 over all values weight, with the smallest MSE 0.0196 between (y-test) actual and (y-pred) predicted outcomes. However, the kernel RBF achieves a little lower AUC values ranging between 0.971 and 1.000, accuracy 0.96 with a MSE higher 0.0392. These
results show that the kernel poly is better possible for the dataset under this split ratio, providing more robust results classification. The consistent performance over varying weights 0.2, 0.4, 0.6, 0.8, and 1 shows the effectiveness of PSO in finding optimal hyperparameters like gamma 0.001 and C 10,000 for both kernels.

Table 4: SVM-PSO (test size=0.2)

	SVM-PSO Optimal Parameters			SVM with Optimal Parameters			Result		
weight	**Particles**	**Iteration**	**C**	**Gamma**	**MSE test set**	**Kernal**	**Accuracy**	**MSE y-test y-pred**	**AUC**
0.2	40	150	16.8668	0.001	0.059201	**RBF**	0.88	0.1176	0.969
						poly	0.98	0.0196	0.998
0.4	40	150	10000	0.001	0.060428	**RBF**	0.96	0.0392	0.971
						poly	0.98	0.0196	0.998
0.6	40	150	10000	0.001	0.060428	**RBF**	0.96	0.0392	0.971
						poly	0.98	0.0196	0.998
0.8	40	150	10000	0.001	0.060428	**RBF**	0.96	0.0392	1.000
						poly	0.98	0.0196	0.998
1	40	150	10000	0.001	0.060428	**RBF**	0.96	0.0392	0.971
						poly	0.98	0.0196	0.998

In Table 4 the kernel poly consistently earns an AUC of 0.998 and an accuracy of 0.98 over all values weight, creating it a reliable choice. However, the kernel RBF displays developments in accuracy with increasing weight, but its MSE decreases significantly when the weight is set to 0.4 and higher than. In view of the exchange between accuracy, MSE, and AUC, a weight of 0.4 is Suggested as it provides a balanced performance with lower error, high accuracy, and improved generalization over the dataset.

7.1 Classification Report of SVM-PSO

In table 5, the majority of cases are correctly classified by the model, which has an accuracy of 0.98. Class 1 exhibits a superior balance between precision 0.96 recall 1.00 and F1-score 0.98, whereas Class 0 has excellent precision 1.00 but low recall 0.96 and F1-score 0.98. The weighted averages precision: 0.98, recall: 0.98, and F1: 0.98 point to improved achievement for the larger class, class 1, while the macro average scores precision: 0.98, recall: 0.98, and F1: 0.98 show strong performance.

Table 5: Classification Report of SVM-PSO (test size=0.2, weight =0.4, kernel=poly)

	Precision	Recall	F1_score	Support
0	1.00	0.96	0.98	24
1	0.96	1.00	0.98	27
accuracy			0.98	51
marco avg	0.98	0.98	0.98	51
weight avg	0.98	0.98	0.98	51

7.2 Confusion Matrix of SVM-PSO

In Figure 3, The underside-right value 27 shows the number of TP is class 1, showing that 27 cases were exactly classified as positive. the uppermost-left value 23 shows the number of TN is class 0, which means 23 cases were exactly classified as negative. The uppermost-right value 1 shows FP, meaning no negative situations were incorrectly classified as positive, and the underside-left value 0

shows FN, where 1 positive case were wrongly classified as negative. In general, the model exactly determines 23 negative and 27 positive cases, displaying strong performance.

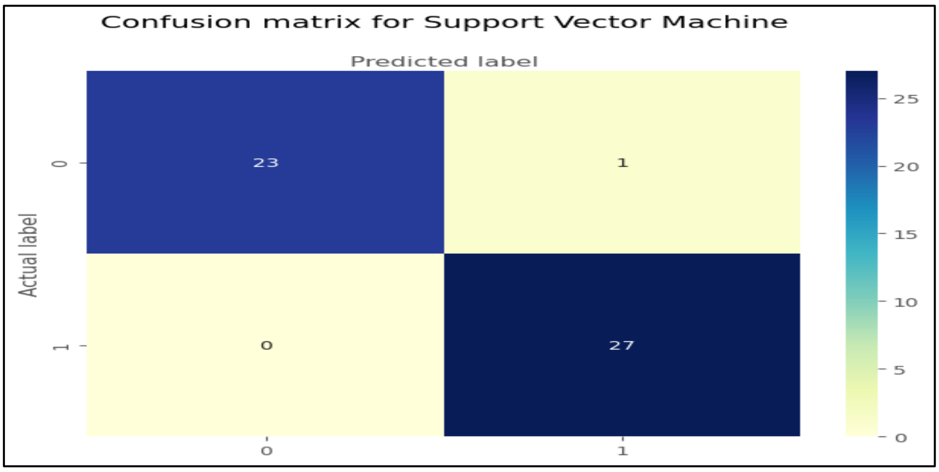

Figure5: Confusion Matrix of SVM_PSO for (test size=0.2, w=0.4, kernel=poly)

7.3 ROC Curve for SVM-PSO

The ROC curve evaluates the performance of the SVM classifier. In Figure 6, the orange curve shows the classifier's performance over different thresholds, with the true positive score plotted opposed to the false positive ratio. As shown in Figure 6, the high AUC value of 0.998 indicates excellent classification performance, as it is close to the ideal value of 1.0. The dashed random guessing is represented by the diagonal line. As presented in the figure, the SVM's curve stays far above the diagonal line, which reflects the strong discriminatory force among the classes positive and negative in the dataset.

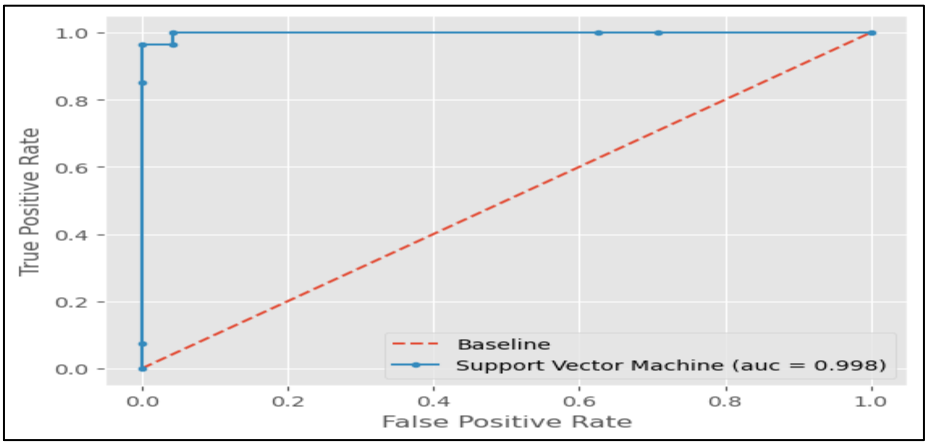

Figure 6: Curve ROC for SVM-PSO (test size=0.2, w=0.4, kernel=poly)

In Table 6, the dataset is split into 0.3, with a similar process of PSO-based hyperparameter optimization. The kernel poly again outperforms the kernel RBF, achieving a higher AUC of 0.999, an accuracy of 0.99, and the smallest MSE of 0.0132 over all values weight. The kernel RBF earns an AUC of 0.983 and an accuracy of 0.97, with a MSE higher of 0.0263. Different from Table 6, there is a small variation in the parameter C for lower weights; C is 13.2482 and weight is 0.2. However, the model still earns great performance. These results indicate that kernel Poly keeps its excellence under a ratio of 0.3 split, providing improved generalization on the data set test. The consistency across varying weights 0.2 to 1 again displays the robustness of the algorithm PSO.

Table 6: SVM-PSO (test size=0.3)

	SVM-PSO Optimal Parameters			SVM with Optimal Parameters			Result		
Weight	**Particles**	**Iteration**	**C**	**Gamma**	**MSE test set**	**Kernal**	**Accuracy**	**MSE y-test, y-pred**	**AUC**
0.2	40	150	13.2482	0.001	0.046940	**RBF**	0.89	0.1053	0.967
						Poly	0.99	0.0132	0.999
0.4	40	150	10000	0.001	0.044923	**RBF**	0.97	0.0263	0.983
						poly	0.99	0.0132	0.999
0.6	40	150	8567.24	0.001	0.044923	**RBF**	0.97	0.0263	0.983
						poly	0.99	0.0132	0.999
0.8	40	150	10000	0.001	0.044923	**RBF**	0.97	0.0263	0.983
						poly	0.99	0.0132	0.999
1	40	150	10000	0.001	0.044923	**RBF**	0.97	0.0263	0.983
						poly	0.99	0.0132	0.999

In table 6 for the test size of 0.3, the kernel poly delivers the maximum with an AUC of 0.999, accuracy of 0.99, and a steadily low MSE of 0.0132 over all values weight, which shows the robustness of the model. The kernel RBF achieves an accuracy of 0.97, but its MSE is slightly higher at kernel. To ensure steadiness and keep lower MSE, a kernel poly is considered optimal, as it offers an excellent balance over all values weight improves the model's forecasting performance without effect accuracy.

Table 7: Classification Report of SVM-PSO (test size=0.3, all weight, kernel =poly)

	Precision	**Recall**	**F1_score**	**Support**
0	1.00	0.96	0.98	26
1	0.98	1.00	0.99	50
accuracy			0.99	76
marco avg	0.99	0.98	0.99	76
weight avg	0.99	0.99	0.99	76

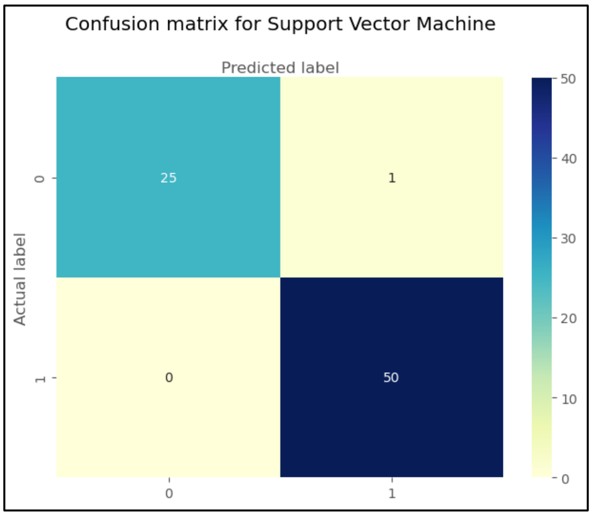

Figure7: Confusion Matrix of SVM-PSO (test size=0.3, all weight, kernel =poly)

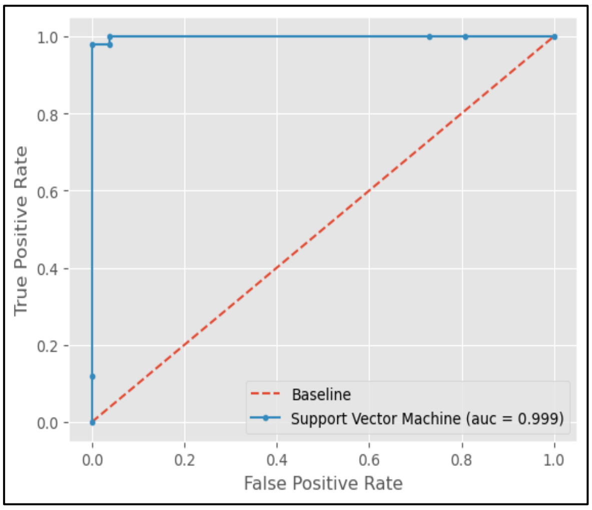

Figure8: ROC Curve for of SVM-PSO (test size=0.3, all weight, kernel =poly)

Table 8 represents the results of applying the algorithm PSO to optimize the hyperparameters of model SVM with a test size 0.4. The table displays how different values of weight affect the model's SVM performance with both RBF and Poly kernels. For weight 0.2, the optimized C value is 19.2276, leading to a kernel RBF with an AUC of 0.980 and an accuracy of 0.91, whereas the kernel poly earns an AUC of 1.000 and an accuracy of 0.99, with a significantly small test MSE of 0.0009. For weights ranging from 0.4 to 1.0, the optimal C stabilizes at 10000, in all weights, gamma stabilizes at 0.001, and the kernel poly continues to earn good performance compared to the kernel RBF. The consistent trend observed is that the poly results in the kernel RBF in terms of accuracy, smaller MSE, and higher AUC over all tested varies weights, making it the best choice for this test size

Table 8: SVM-PSO (test size=0.4)

	SVM-PSO Optimal Parameters			**SVM with Optimal Parameters**			**Result**		
Weight	**Particles**	**Iteration**	**C**	**Gamma**	**MSE test set**	**Kernal**	**Accuracy**	**MSE y-test, y-pred**	**AUC**
0.2	40	150	19.2276	0.001	0.035363	**RBF**	0.91	0.0891	0.980
						poly	0.99	0.0099	1.000
0.4	40	150	10000	0.001	0.048321	**RBF**	0.98	0.0198	0.987
						poly	0.99	0.0099	1.000
0.6	40	150	10000	0.001	0.048321	**RBF**	0.98	0.0198	0.987
						poly	0.99	0.0099	1.000
0.8	40	150	10000	0.001	0.048321	**RBF**	0.98	0.0198	0.987
						poly	0.99	0.0099	1.000
1	40	150	10000	0.001	0.048321	**RBF**	0.98	0.0198	0.987
						poly	0.99	0.0099	1.000

For the test size of 0.4, the kernel poly earns an AUC of 1.000, an accuracy of 0.99, and the smallest MSE over all values weight, highlighting its higher performance. The kernel RBF displays a consistent accuracy of 0.98 with a visible improvement in MSE for a weight over all values weight except of weight is 0.2. earns an AUC of 0.980, an accuracy of 0.91, and the bigger MSE is 0.0891. Between these kernels, a poly of is recommended as it provides optimal accuracy and the minimal MSE over all values weight, making certain the most stable and dependable performance when utilizing the 0.4 test size proportion.

Table 9: Classification Report of SVM-PSO (test size=0.4, all weight, kernel =poly)

	Precision	Recall	F1_score	Support
0	1.00	0.97	0.98	29
1	0.98	1.00	0.99	72
accuracy			0.99	101
marco avg	0.99	0.98	0.99	101
weight avg	0.99	0.99	0.99	101

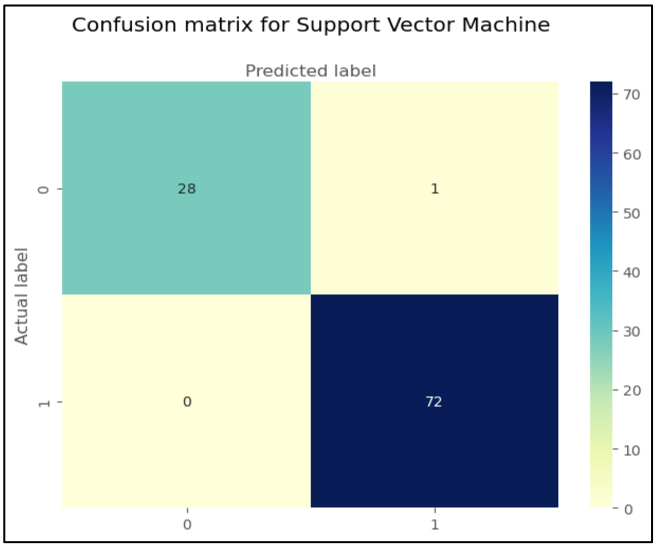

Figure 9: Confusion Matrix of SVM-PSO (test size=0.4, all weight, kernel =poly)

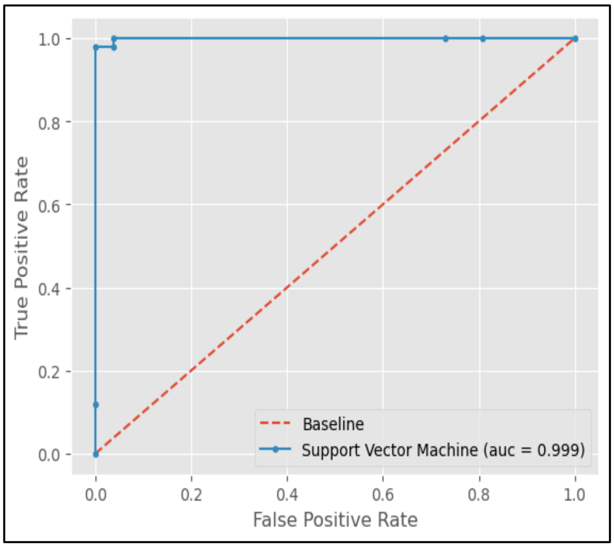

Figure10: ROC Curve for of SVM-PSO (test size=0.4, all weight, kernel =poly)

Comparing all test sizes of 0.2, 0.3, and 0.4, the performance of the kernel poly remains reliably high over different data splits. Kernel poly in the 0.4 test size case has an AUC of 1.000 and an accuracy of 0.99, while the kernel RBF struggles to reach similar accuracy levels. As the test size increases, the MSE of the kernel RBF tends to decrease a little, but it remains significantly higher than the kernel Poly. The smaller test size of 0.2 in Table 4 provides a little better RBF performance in the bigger weight compared to the smaller weight, recommending that the model generalizes greater with more training data. But the kernel poly displays a slight difference in performance over variance test sizes, showing its robustness and efficiency. In general, over-all test sizes, the kernel Poly consistently earns better generalization performance, decreases error rates, and has increased predictive accuracy, making it the most possible choice for this classification problem.

7.4 Performance comparison of SVM-PSO

Table 6: Performance comparison of SVM-PSO.

Split	SVM Model Results				SVM-PSO Result			
Test size	**kernel**	**Accuracy**	**AUC**	**MSE**	**kernel**	**Accuracy**	**AUC**	**MSE**
0.2	RBF	0.90	0.9556	0.0980	**poly**	0.98	0.998	0.0196
0.3	RBF	0.84	0.9403	0.1579	**poly**	0.99	0.999	0.0132
0.4	RBF	0.86	0.9545	0.1386	**poly**	0.99	1.000	0.0099

8. Conclusions and Recommendation

8.1 Conclusions

The results of this paper show that optimizing SVM models utilizing PSO substantially improves diabetes classification accuracy and decreases error ratio compared to classical SVM models. The optimized SVM-PSO approach with a kernel poly steadily surpassed the classical SVM model with an RBF kernel in all test-size situations. Especially, the SVM-PSO model earns an accuracy of 0.99 over all test sizes, with an AUC value of 1.000 at a test size of 0.4, showing superior forecasting capacity and generalization. Additionally, the decrease in MSE values highlights the enhanced dependability of the optimized model. These results highlight the capability of PSO as a powerful technique for optimizing ML algorithms in medical analysis, especially for diabetes classification.

8.2 Recommendations

1. The authors recommend other researchers to use the proposed SVM-PSO for forecasting different datasets in other areas.

2.It is additionally suggested that in future work the classical model requirements be improved with other techniques, including genetic algorithms and artificial neural networks, to examine which technique improvement requirements to improve the model best.

3. Future work may focus on merging extra optimization techniques and investigating deep learning models to further improve prediction accuracy and model understandability.

References:

1. Cho, M. Y., & Hoang, T. T. (2017). Feature selection and parameters optimization of SVM using particle swarm optimization for fault classification in power distribution systems. *Computational intelligence and neuroscience*, *2017*(1), 4135465. https://doi.org/10.1155/2017/4135465.
2. Du, J., Liu, Y., Yu, Y., & Yan, W. (2017). A prediction of precipitation data based on support vector machine and particle swarm optimization (PSO-SVM) algorithms. *Algorithms*, *10*(2), 57. https://doi.org/10.3390/a10020057
   a. http://doi/ org/10.31272/JAE.43.2020.126.24.
3. Jibril, A. U., Haruna, K., & Jiangsheng, Z. (2023). Feature Selection and Parameter Optimization of Support Vector Machine (Svm) and Logistic Regression (Lr) Algorithms Using Particle Swarm Optimization (Pso) In Prediction of Diabetes. *Journal of Computer Science*, *11*(1), 21-47. https://doi.org/10.15640/jcsit.v11n1a3.
4. Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, *9*(1), 381-386. DOI: 10.21275/ART20203995.
5. Rahman, C. M., & Rashid, T. A. (2021). A new evolutionary algorithm: Learner performance based behavior algorithm. *Egyptian Informatics Journal*, *22*(2), 213-223. https://doi.org/10.1016/j.eij.2020.08.003.
6. Saputra, D., Dharmawan, W. S., & Irmayani, W. (2022). Performance comparison of the SVM and SVM-PSO algorithms for heart disease prediction. *International Journal of Advances in Data and Information Systems*, *3*(2), 74-86. https://doi.org/10.25008/ijadis.v3i2.1243
7. Tharwat, A., & Hassanien, A. E. (2019). Optimizing support vector machine parameters using bat optimization algorithm. *Machine learning paradigms: theory and application*, 351-374. https://doi.org/10.1007/978-3-030-02357-7_17.
8. Venkatesan, D. S., Saideepika, G., & Sarathambekai, S. (2024). A Comparative Study of Swarm-Optimized Machine Learning Models for Diabetic Kidney Disease Prediction. https://doi.org/10.20944/preprints202410.1528.v1.
9. Wang, L. (Ed.). (2005). *Support vector machines: theory and applications* (Vol. 177). Springer Science & Business Media.
10. Zhou, F. H., & Liao, Z. Z. (2013). A particle swarm optimization algorithm. *Applied Mechanics and Materials*, *303*, 1369-1372. https://doi.org/10.4028/www.scientific.net/AMM.303-306.1369
11. بديعة رحمن خليل, أ. م. د. محمد محمود فقي, & أ. م. د. سوزان صابر حيدر. (2022). Classifying Patients with Myocardial Infarction and Heart Failure by Using SVM and KNN Learning Techniques.