

IMPROVING HEART DISEASE PREDICTION THROUGH PSO-OPTIMIZED SUPPORT VECTOR MACHINES USING LINEAR AND GAUSSIAN KERNELS

Pirapong Inthapong¹, Narongdech Dungkratoke¹, Nara Samattapapong², and Niwatchai Namvichaisirikul^{3,*}

Received: October 09, 2024; Revised: December 02, 2024; Accepted: February 03, 2025

Abstract

This study examines the performance of support vector machines (SVMs) utilizing two distinct kernels, linear and Gaussian, optimized through two variants of the particle swarm optimization (PSO) algorithm: classical and adaptive. The focus is on classifying heart disease data in medical diagnostics, where precision and efficiency are vital. PSO, an evolutionary optimization technique, is employed to fine-tune SVM parameters, thereby enhancing the models' classification accuracy. The heart disease dataset used includes multiple features representing various health indicators, necessitating a robust classification methodology. Performance is evaluated using accuracy, precision, recall, and F1-score metrics. The results reveal how different PSO variants influence SVM classification, providing a comparative analysis between the linear and Gaussian kernels. The findings indicate that PSO-optimized SVMs can substantially improve heart disease detection and diagnosis, with each kernel demonstrating unique strengths, depending on the optimization strategy employed.

Keywords: Heart disease prediction; Machine learning in healthcare; Particle swarm optimization; Support vector machine

¹ Department of Interdisciplinary Science and Internationalization, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. E-mail: pirapong@g.sut.ac.th

² School of Industrial Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. E-mail: nara@g.sut.ac.th

³ School of Family and Community Medicine, Institute of Medicine, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. E-mail: niwatchai@sut.ac.th

* Corresponding author

DOI: <https://doi.org/10.55766/sujst6984>

Suranaree J. Sci. Technol. 32(1):030274(1-8)

Introduction

Heart disease is a significant global health concern and remains one of the leading causes of death worldwide (Vaduganathan *et al.*, 2022). Early and accurate diagnosis of heart disease is crucial for effective treatment and the prevention of severe outcomes. In recent years, machine learning techniques have become valuable tools for improving the accuracy of medical diagnoses, offering the potential to analyze complex datasets and identify patterns that may be difficult to detect through traditional methods (Ahsan and Siddique, 2022; Alkayyali *et al.*, 2023). Among these techniques, support vector machines (SVMs) have emerged as robust and reliable classifiers, particularly when dealing with high-dimensional and intricate data. SVMs operate by finding an optimal hyperplane that separates data points into different classes, making them highly effective in tasks such as heart disease prediction (Sarraf *et al.*, 2022). However, the performance of SVMs is strongly influenced by the selection of hyperparameters and kernel functions, which can impact their ability to generalize effectively on unseen data (Tharwat, A. and Gabel, 2020).

Particle swarm optimization (PSO) (Shami *et al.*, 2022) is an evolutionary optimization algorithm inspired by the collective behavior of social organisms such as birds or fish. PSO is well-suited for optimizing complex problems, as it can efficiently search for optimal parameter values by simulating the movement of particles in a solution space (Nayak *et al.*, 2023). Over the past decade, PSO and its variants have been proposed, developed, and employed in a variety of scientific and engineering applications, including optimization in machine learning (Wang Wang, 2022), network routing (Yelure and Sonavane, 2013), and power system operations (Latif *et al.*, 2022). The main advantages that make PSO appealing are its simplicity, flexibility, and ability to handle non-linear and high-dimensional optimization problems. Additionally, PSO offers fast convergence and is highly effective in escaping local optima, making it particularly useful for solving complex, multi-objective optimization tasks (Damaj *et al.*, 2020; Kouziokas, 2023).

This study focuses on applying PSO to fine-tune critical SVM parameters, including the regularization parameter (C) and the kernel parameter (γ) for Gaussian SVMs (Ebarts and Steinwart, 2013), with the aim of achieving greater classification accuracy. By optimizing both linear and Gaussian kernels with two distinct PSO

variants, this research seeks to evaluate how these kernels perform on heart disease datasets, which are often composed of multiple features and complex health indicators. The combination of PSO and SVM offers a promising approach to addressing the challenges of hyperparameter tuning, potentially enhancing the precision and efficiency of heart disease detection.

The primary objective of this work is to evaluate how PSO-optimized SVMs can improve heart disease prediction by analyzing the performance based on critical metrics such as accuracy, precision, recall, and F1-score (Vujović, 2021). The study also aims to explore the distinct advantages and trade-offs of each kernel in the context of medical diagnostics, with the findings suggesting that integrating PSO with SVM significantly enhances classification accuracy. Each kernel exhibits unique strengths, depending on the optimization strategy, highlighting the potential of PSO-optimized SVMs in contributing to more reliable and precise heart disease diagnosis. By improving early detection capabilities, this approach may help reduce the impact of heart disease on patients, leading to better treatment outcomes and lower mortality rates.

Methodology

Support Vector Machine Classifiers

Support vector machines (SVMs) are powerful tools for classification, finding the optimal hyperplane that maximizes the margin between two classes. Given a dataset $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \{-1, 1\}$ are labels, the SVM finds a hyperplane in the following form:

$$f(x) = w^T x + b = 0 \quad (1)$$

The goal is to maximize the margin subject to $y_i (w^T x_i + b) \geq 1$ for all i . The optimization problem for a hard-margin SVM is expressed as

$$\min_{w, b} \frac{1}{2} \|w\|^2 \quad (2)$$

In cases where data are not linearly separable, a soft-margin SVM introduces slack variables ξ_i , leading to the form expressed below:

$$y_i (w^T x_i + b) \geq 1 - \xi_i \forall_i, \quad \min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (3)$$

Where $C > 0$ controls the trade-off between margin size and classification errors.

Linear SVM

For linearly separable data, the optimization problem is

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ subject to } y_i(w^T x_i + b) \geq 1 \quad (4)$$

This can be solved using the dual formulation as shown in the following form:

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

subject to $\sum_{i=1}^n \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$. The solution $w = \sum_{i=1}^n \alpha_i y_i x_i$ depends on the support vectors.

Gaussian Non-linear SVM

For non-linear data, the Gaussian RBF kernel defined as

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (6)$$

maps data into a higher-dimensional space, allowing the SVM to find a linear boundary in that space. The dual problem becomes the following form:

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (7)$$

with similar constraints as the linear case. The decision function is as follows:

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x_i, x) + b \quad (8)$$

Parameters C (regularization) and γ (kernel width) need to be tuned for optimal performance, which is the focus of this work. To achieve this goal, three strategies are deployed: the well-known Grid Search, the classical PSO, and the Adaptive PSO, the last two being briefly introduced in the following section.

Mathematical Formulation of PSO

This section provides an overview of the mathematical formulation of PSO, covering both the classical version and adaptive strategies.

Classical Particle Swarm Optimization (CPSO)

The CPSO algorithm involves a set of particles (solutions) moving in the search space to find the optimal solution. The position of each particle is updated based on its velocity, influenced by the particle's personal best position ($P_i(t)$) and the

global best position ($G(t)$) among all particles. The velocity and position update equations are given, respectively, as

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [P_i(t) - p_i(t)] + c_2 r_2 [G(t) - p_i(t)] \quad (9)$$

$$p_i(t+1) = p_i(t) + v_i(t+1) \quad (10)$$

where $v_i(t)$ and $p_i(t)$ represent the velocity and position of particle i at iteration t , speculatively. $P_i(t)$ is the personal best position of particle i , and $G(t)$ is the global best position. The constants c_1 and c_2 are the cognitive and social learning coefficients, and r_1 and r_2 random numbers uniformly distributed between 0 and 1. The inertia weight ω controls the trade-off between exploration and exploitation in the search space.

Inertia Weight Adaptation Strategies (APSO)

Several strategies have been proposed to adapt the inertia weight ω to improve convergence. One common approach is to set a fixed inertia weight, with $\omega = 0.7298$ being widely adopted in many PSO variants. However, more dynamic approaches decrease the inertia weight linearly over time, as described by the following equation:

$$\omega(t) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T} \cdot t \quad (11)$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum inertia weights, respectively, t is the current iteration, and T is the total number of iterations.

The acceleration coefficients c_1 and c_2 determine the relative importance of the cognitive component (p -best) and the social component (g -best) in the velocity update. In some PSO variants, these coefficients are set to fixed values, typically $c_1 = c_2 = 2.0$. However, to improve the balance between exploration and exploitation over time, time-varying acceleration coefficients have been proposed.

A time-varying acceleration strategy reduces the cognitive component while increasing the social component over the course of the iterations. The equations for the time-varying acceleration coefficients are as follows:

$$c_1(t) = c_1^{\max} - (c_1^{\max} - c_1^{\min}) \frac{t}{T} \quad (12)$$

$$c_2(t) = c_2^{\max} - (c_2^{\max} - c_2^{\min}) \frac{t}{T} \quad (13)$$

where $c_1^{max} = c_2^{max} = 2.5$ and $c_1^{min} = c_2^{min} = 0.5$. This strategy encourages early exploration of the search space by initially prioritizing the cognitive component, but gradually shifts toward exploiting the global best solution as the number of iterations increases.

By adapting both the inertia weight and acceleration coefficients, the classical PSO can maintain a better balance between exploration and exploitation, potentially improving its performance on complex optimization problems. This version is referred to hereafter as “Adaptive PSO (APSO).”

The Dataset and Algorithm

Data Source

This study utilizes a Heart Failure Prediction Dataset containing 918 instances of various patient characteristics and medical data. The dataset consists of 12 features, capturing patient demographics, clinical information, and test results. This list comprises age, gender, chest pain type, resting blood pressure, cholesterol levels, fasting blood glucose, results from a resting electrocardiogram, peak heart rate achieved, incidence of angina induced by exercise, ST depression (commonly called oldpeak), and the ST segment slope. The instances are categorized into two classes: those with heart disease and those without. In the dataset, approximately 55% of the instances belong to the heart disease class, while 45% do not.

Description of Attributes

The dataset consists of 12 attributes, including demographic details, clinical measurements, and test results. These attributes are crucial for predicting the presence of heart disease. The attributes are described in Table 1.

Normalization

To ensure that the attributes in the dataset are comparable and avoid the effects of different measurement scales, two methods of data transformation were employed: normalization and standardization.

Normalization adjusts the values of features to fall within 0 and 1 using the following formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (14)$$

where X_{min} and X_{max} are the minimum and maximum values of the feature, ensuring all values are scaled to the 0 to 1 range.

In contrast, standardization adjusts the feature values to achieve a mean of 0 and a standard deviation of 1. This method is particularly effective when dealing with data that follow a normal distribution or when necessary to ensure all features operate on the same scale while preserving their underlying distribution. The equation for standardization is as follows:

$$X_{standardized} = \frac{X - \mu}{\sigma} \quad (15)$$

here, μ represents the mean of the feature and σ its standard deviation, resulting in features with a zero mean and unit variance.

Encoding Categorical Variables

In order to effectively use categorical variables in machine learning models, these variables need to be transformed into numerical format. This process, known as encoding, allows algorithms to interpret and process categorical data. One commonly used method is Label Encoding, which assigns a corresponding numerical value to each unique category in a feature. For instance, categories such

Table 1: Attributes in the heart failure prediction dataset

No	Attribute	Description
1	Age	Patient's age represented in years.
2	Sex	Patient's gender, denoted by 'M' for male and 'F' for female.
3	ChestPainType	Classification of chest pain, including Typical Angina (TA) and Atypical Angina (ATA).
4	RestingBP	Non-Anginal Pain (NAP), and Asymptomatic (ASY).
5	Cholesterol	Blood pressure measured while at rest, expressed in mm Hg.
6	FastingBS	Concentration of cholesterol in the blood, measured in mg/dl.
7	RestingECG	Indicator for fasting blood sugar, with a value of 1 if exceeding 120 mg/dl and 0 otherwise.
8	MaxHR	Results of a resting ECG, categorized as Normal, ST-T wave abnormality (ST), or LVH.
9	ExerciseAngina	The highest recorded heart rate during testing, presented as a numerical value ranging from 60 to 202.
10	Oldpeak	Indicates whether angina was triggered by exercise ('Y' for Yes, 'N' for No).
11	ST_Slope	The value of ST depression relative to the baseline (rest), measured in millimeters.
12	HeartDisease	The slope of the ST segment during peak exercise, classified as upsloping, flat, or downsloping.

as 'Yes' and 'No' may be encoded as 1 and 0, respectively. This method is suitable for categorical variables with an inherent order or when there are only a few unique categories.

Evaluation Metrics

To evaluate the performance of the classification models, several performance measures are utilized, derived from the confusion matrix: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The following metrics are considered.

Accuracy: Proportion of correctly classified instances over the total number of instances.

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

Precision: Proportion of true positives from all predicted positive instances.

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

Recall: Proportion of true positives from all actual positive instances.

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

F1 Score: The harmonic mean of precision and recall.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (19)$$

Algorithms

In our approach, PSO is utilized to optimize the parameters of the SVM. The corresponding pseudocode is as follows:

Algorithm: SVM with PSO

Result: Best position ($G(t)$)

Initialize the PSO parameters:

- Set the swarm size, ω for Linear Gaussian SVM, T , c_1 , and c_2 .
- Set the bounds for c_1 , c_2 , and ω .
- Set the position and velocity for each swarm: 1D (optimize C for the linear Gaussian SVM model) or 2D (optimize C and γ for the non-linear Gaussian SVM).

For each iteration:

For each particle:

- Update the velocity for each particle following Equation 9.
- Update the position for each particle following C 10.
- Evaluate the fitness function.

If the current fitness is better than the fitness of ($P_i(t)$):

- Update $P_i(t)$.

If $P_i(t)$ is better than $G(t)$:

- Update $G(t)$.

Return $G(t)$

After completing the iterative process with the training set in the algorithm, the optimal parameters are applied to the test set, and the results are subsequently analyzed.

Numerical Experiments and Results

In this experiment, three optimization strategies were employed to determine the optimal parameters for a Support Vector Machine (SVM), including Grid Search, Classical Particle Swarm Optimization (CPSO), and Adaptive Particle Swarm Optimization (APSO). For the Grid Search method, the regularization parameter, C , and Gaussian kernel parameter, γ , were explored within a search space ranging from 0.0001 to 1000, covering more than 10,000 possible values in both one-dimensional (linear SVM) and two-dimensional (non-linear Gaussian SVM) cases. The CPSO and APSO methods each utilized 10 particles and 50 iterations. While CPSO maintained fixed values for the PSO parameters ω , c_1 , and c_2 , set at 0.7298, 2.0000, and 2.0000, respectively, the APSO approach adapted these parameters dynamically during the optimization process. The following sections provide a detailed comparison of the results obtained from these optimization strategies and their effectiveness in improving the performance of the SVM models.

The Linear SVM

In the first case, the effectiveness of the linear SVM optimized by the Grid Search and the two versions of PSO are investigated, and the results are summarized in Table 1. As can be observed, the Grid Search method yielded a relatively low C value of 0.0230, corresponding to an accuracy of 82.07%, a precision of 89.36%, a recall of 78.50%, and an F1-score of 83.58%. In contrast, both CPSO and APSO achieved a much higher optimized C value of 193.4609, leading to improved performance metrics. Specifically, both CPSO and APSO reached an accuracy of 84.24%, precision of 83.79%, recall of 84.45%, and F1-score of 83.99%.

These findings suggest that while the Grid Search was able to achieve relatively high precision, it resulted in a significantly lower recall and overall model performance compared to the PSO-based

strategies. The higher C values obtained through CPSO and APSO indicate that these methods were more effective in balancing the trade-off between margin maximization and classification errors, thus improving the SVM's ability to generalize. Additionally, the similarity in performance between CPSO and APSO implies that both variants of PSO were equally effective in this optimization task, confirming their potential for enhancing SVM-based classification models.

Taking a closer look at the Grid Search, Figure 1 highlights the significant influence of parameter C on the performance of a linear SVM, emphasizing the importance of selecting the right value for C for optimizing model behavior. The C values in this time-consuming task were slightly above the aforementioned optimal value (as indicated in Table 2) to demonstrate their effect on performance. Accordingly, as C increases from 0 to 5, accuracy remains relatively stable between 0.86 and 0.87, while precision fluctuates more notably at lower values, ranging from 0.85 to 0.87, and only stabilizes around 0.86 after C exceeds 2.45. Recall, which consistently stays near 0.89, is less affected by C , though minor dips occur when C is very small. Similarly, the F1-score follows the precision trend, stabilizing at around 0.87 beyond a C value of 2.45. These fluctuations, especially at lower C values, point to the challenge of under-regularization, where the model is overly flexible, resulting in unstable performance. As C increases, performance stabilizes, but it becomes clear that choosing the right value of C is not straightforward. While larger C values seem to help balance classification accuracy and margin maximization, no significant improvements can be observed for very high C values, suggesting a delicate trade-off between

underfitting and overfitting. This underscores the difficulty in pinpointing the most appropriate C value, as the best choice depends on the specific data and the balance between minimizing errors and maximizing the margin. This complexity highlights the crucial role played by optimization algorithms like PSO. By automating the search for the optimal C , PSO can help to efficiently navigate this parameter tuning process, achieving a better balance between competing performance metrics.

The Non-linear Gaussian SVM

The optimization of the regularization parameter C and the Gaussian kernel parameter γ for the non-linear Gaussian SVM was performed using the three strategies and performance metrics detailed in **Error! Reference source not found..** As can be observed, the Grid Search approach resulted

Table 2. Performance comparison for the linear Gaussian SVM model

Strategies	Grid Search	CPSO	APSO
C of SVM	0.0230	193.4609	193.4609
Accuracy	0.8207	0.8424	0.8424
Precision	0.8936	0.8379	0.8379
Recall	0.7850	0.8445	0.8445
F1-Score	0.8358	0.8399	0.8399

Table 3 Performance comparison for the non-linear Gaussian SVM model

Strategies	Grid Search	CPSO	APSO
C of SVM	0.8111	332.8363	400.6134
Gamma	0.0658	0.00001	0.0043
Accuracy	0.8804	0.8424	0.9022
Precision	0.9048	0.8980	0.9406
Recall	0.8879	0.8224	0.8879
F1-Score	0.8962	0.8585	0.9135

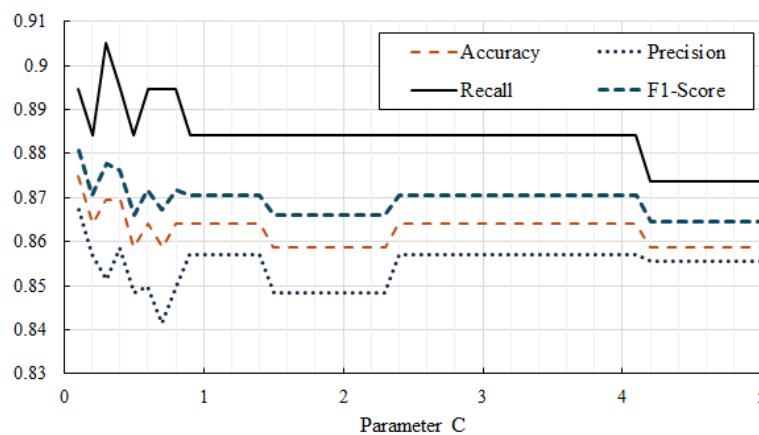


Figure 1. Accuracy, precision, recall, and F1-score variation with parameter C in the linear SVM classification model

in $C = 0.8111$ and $\gamma = 0.0658$ with an accuracy of 88.04%, precision of 90.48%, recall of 88.79%, and F1-score of 89.62%. The CPSO, however, identified $C = 332.8363$ and $\gamma = 0.00001$ yielding lower accuracy and recall of 84.24% and 82.24%, respectively, but still exhibited high precision of 89.80% and an F1-score of 85.85%. The APSO method achieved the best performance, optimizing $C = 400.6134$ and $\gamma = 0.0043$, leading to an accuracy of 90.22%, precision of 94.06%, recall of 88.79%, and an F1-score of 91.35%.

To further validate the effectiveness of our method, we compared the results obtained from Mohan *et al.* (2019) and Nasiruddin *et al.* (2024). The findings revealed that APSO demonstrated superior accuracy at 0.9022 compared to 0.8841 and 0.8847, respectively. APSO also achieved a higher precision of 0.9406, indicating improved positive prediction accuracy. Despite the recall being slightly lower than that reported by Mohan *et al.* (2019) and Nasiruddin *et al.* (2024), APSO maintained a competitive performance at 0.8879. Notably, APSO excelled with the highest F1-Score of 0.9135, highlighting its balanced and effective overall performance.

These results suggest that while the Grid Search produced a well-balanced model with relatively high accuracy and recall, APSO provided the best overall performance across all metrics, particularly in terms of precision and F1-score. This indicates that APSO effectively balances the trade-off between C and γ , improving the classification performance of the non-linear Gaussian SVM. The CPSO method, although producing slightly lower results compared to the other methods, demonstrated its potential to achieve competitive precision values. Overall, the APSO method proved to be the most effective in optimizing the non-linear SVM model.

Conclusions

The findings of this study highlight the importance of Adaptive Particle Swarm Optimization (APSO) in improving the performance of Support Vector Machines (SVMs) for heart disease prediction. Compared to Grid Search and classical PSO (CPSO), APSO achieved superior accuracy, precision, recall, and F1-score by dynamically adjusting key parameters, such as C and γ . This adaptability allowed for a more finely tuned model that better balances minimizing false positives and maximizing true positives, which is crucial in medical diagnostics. Beyond heart disease, the potential of APSO extends to managing non-communicable diseases (NCDs) like diabetes

and cancer, where early diagnosis and personalized treatment can be enhanced by machine learning models. For healthcare professionals, such AI-driven solutions reduce the workload of data analysis and empower faster, more accurate decision-making, ultimately improving patient care and job satisfaction. The integration of APSO with machine learning models not only improves healthcare quality but also enhances the careers of doctors and nurses by enabling more effective and fulfilling work.

Acknowledgment

This research has been generously supported by the Health Systems Research Institute (HSRI), Thailand. The authors sincerely appreciate their financial and institutional support, which was instrumental in the successful completion of this study.

References

- Ahsan, M.M. and Siddique, Z. (2022). Machine learning-based heart disease diagnosis: A systematic literature review. *Artificial Intelligence in Medicine*, 128:102289. <https://doi.org/10.1016/j.artmed.2022.102289>
- Alkayyali, Z.K, Idris, S.A., and Abu-Naser, S.S. (2023). A systematic literature review of deep and machine learning algorithms in cardiovascular diseases diagnosis. *Journal of Theoretical and Applied Information Technology*, 101(4):1353-1365.
- Damaj, I., Elshafei, M., El-Abd, M., and Aydin, M.E. (2020). An analytical framework for high-speed hardware particle swarm optimization. *Microprocessors and Microsystems*, 72:102949. <https://doi.org/10.1016/j.micpro.2019.102949>
- Eberts, M. and Steinwart, I. (2013). Optimal regression rates for SVMs using Gaussian kernels. 7:1-42 <https://doi.org/10.1214/12-EJS760>
- Kouziokas, G. (2023). *Swarm Intelligence and Evolutionary Computation: Theory, Advances and Applications in Machine Learning and Deep Learning*. 1st ed. CRC Press, 218p. <https://doi.org/10.1201/9781003247746>
- Latif, S., Irshad, S., Kamarposhti, M.A., Shokouhandeh, H., Colak, I., and Eguchi, K. (2022) Intelligent design of multi-machine power system stabilizers (PSSs) using improved particle swarm optimization. *Electronics*, 11(6):946. <https://doi.org/10.3390/electronics11060946>
- Mohan, S., Thirumalai, C., and Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*, 7:81542-81554. <https://doi.org/10.1109/ACCESS.2019.2923707>
- Nasiruddin, M., Dutta, S., Sikder, R., Islam, M.R., Mukaddim, A.A., and Hider, M.A. (2024). Predicting heart failure survival with machine learning: assessing my risk. *Journal of Computer Science and Technology Studies*, 6(3):42-55. <https://doi.org/10.32996/jcsts.2024.6.3.5>
- Nayak, J., Swapnarekha, H., Naik, B., Dhiman, G., and Vimal, S. (2023). 25 years of particle swarm optimization: Flourishing voyage of two decades. *Archives of Computational Methods in Engineering*, 30(3):1663-725. <https://doi.org/10.1007/s11831-022-09849-x>

- Sarra, R.R., Dinar, A.M., Mohammed, M.A., and Abdulkareem, K.H. (2022). Enhanced heart disease prediction based on machine learning and χ^2 statistical optimal feature selection model, *Designs*, 6(5):87. <https://doi.org/10.3390/designs6050087>
- Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A., and Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *Ieee Access*, 10:10031-61. <https://doi.org/10.1109/ACCESS.2022.3142859>
- Tharwat, A. and Gabel, T. (2020). Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. *Neural Computing and Applications*, 32(11):6925-6938. <https://doi.org/10.1007/s00521-019-04159-z>
- Vaduganathan, M., Mensah, G.A., Turco, J.V., Fuster, V., and Roth, G.A. (2022). The global burden of cardiovascular diseases and risk: a compass for future health. *Journal of the American College of Cardiology*, 80(25):2361-2371. <https://doi.org/10.1016/j.jacc.2022.11.005>
- Vujović, Ž. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):599-606. <https://doi.org/10.14569/IJACSA.2021.0120670>
- Wang, F., Wang, X., and Sun, S. (2022). A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Information Sciences*, 602:298-312. <https://doi.org/10.1016/j.ins.2022.04.053>
- Yelure, B. and Sonavane, S. (2020). Particle swarm optimization based routing method for vehicular ad-hoc network. In: 2020 International Conference On Communication And Signal Processing (ICCSP), IEEE, Chennai, India, p. 1573-1578. <https://doi.org/10.1109/ICCSP48568.2020.9182305>