Hyperparameter Tuning of SVM using Metaheuristics: An Empirical Analysis

Sunday O. Oladejo
School for Data Science and Computational Thinking
Stellenbosch University
South Africa
sunday@sun.ac.za

Lateef A. Akinyemi
Centre for Augmented Intelligence and Data Science
University of South Africa
South Africa
akinyla@unisa.ac.za

Stephen O. Ekwe
Dept. of Electrical, Electronic and Comp. Engineering
Cape Peninsula University of Technology
South Africa
ekwes@cput.ac.za

Taiwo G. Omomule
Dept of Mathematics and Statistics
University of Guelph
Ontario, Canada
taiwoo@uoguelph.ca

Oluwaseyi P. Babalola
Dept. of Electrical, Electronic and Comp. Engineering
Cape Peninsula University of Technology
South Africa
babalolao@cput.ac.za

2025 IEEE 3rd Wireless Africa Conference (WAC) | 979-8-3315-1758-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/WAC63911.2025.10992610

***Abstract*—The Support Vector Machine (SVM) performance is highly dependent on the selection of optimal hyperparameters. Conventional optimisation methods, including grid search, gradient descent, randomized search, and various experimental approaches, have been extensively utilized in the literature to identify optimal hyperparameters. This paper provides an empirical analysis of SVM's hyperparameter tuning by use of metaheuristics. We compare 11 widely known metaheuristics, such as the Genetic Algorithm (GA), Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), and Differential Evolution (DE), and employ three common datasets, namely Iris, Breast Cancer Wisconsin, and MNIST, due to their unique characteristics. Empirical analysis takes into account factors including cross-validation, number of agents, noise, and mislabelled data insertion, with emphasis on performance metrics such as accuracy, number of support vectors, and computational time. Our findings reveal that the PSO outperforms other metaheuristics, followed by the DE and ACO. In addition, we conclude that the ACO has the least computational time.**

***Index Terms*—SVM, hyperparameters, metaheuristics, optimisation, support vectors.**

I. Introduction

The Support Vector Machine (SVM), a kernel-based machine learning (ML) model [1], is a widely employed ML algorithm for classification (and sometimes for regression analysis) [2], [3]. Its popularity can be attributed to its simplicity; generalisation and discriminative capabilities; high stability and performance accuracy; statistical and theoretical foundations; and ease of deployment. It is premised on the transformation of the training dataset into a higher dimensionality with an optimal separation boundary (i.e., hyperplanes) between the respective classes.

One of the SVM's challenges is the choice and parameter tuning of the kernels [4], [5]. An incorrect choice of kernel and hyperparameter tuning may adversely affect SVM's performance. To this end, nature-inspired metaheuristics have been proposed for the training or tuning of SVM's hyperparameters; that said, these metaheuristics may not guarantee the global best solution (i.e., the global optima).

II. SVMs: A General Overview

This section covers an overview, mathematical formulation, kernel trick, and hyperparameter tuning of SVMs.

A. The Hard-Margin SVM Optimisation Problem

Let $\mathcal{G}$ be a dataset comprising n m-dimension input vectors $\mathbf{x}_i$ each associated with a class label y_i. The dataset $\mathcal{G}$ is characterised by

$$\mathcal{G} = \left\{ (\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{-1, 1\} \right\}_{i=1}^{n} \tag{1}$$

The hyperplane is defined by [4]

$$\mathbf{w}^T \mathbf{x} + b = 0 \ , \tag{2}$$

where $\mathbf{w} = (a, -1)$. The SVM classifier predicts using $\mathbf{w}^T\mathbf{x} + b$, and comparing this prediction to the actual class y_i yields the functional margin given as

$$\alpha_i = y_i(\mathbf{w}^T \mathbf{x} + b) \tag{3}$$

979-8-3315-1758-8/25/$31.00 ©2025 IEEE
Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.

Due to the scalability weakness of the functional margin, (3) is normalised by the norm of $\mathbf{w}$, resulting in the geometric margin $\widehat{\alpha}_i$, which remains invariant under scaling. The optimisation of the geometric margin written as a minimisation problem is given as [4]

$$\min_{b,w} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{4}$$

s.t.

$$y_i(\mathbf{w}^T\mathbf{x} + b) \geq 1, \quad 1, 2, 3, \ldots, n$$

Moreover, Lagrangian multipliers are employed to solve the SVM optimisation in (4). The Lagrangian of (4), formulated as its dual problem, is defined as

$$\mathcal{L}(\mathbf{w}, b, \gamma) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\gamma_i[\ y_i(\mathbf{w}^T\mathbf{x} + b) - 1] \tag{5}$$

where γ_i represents the Lagrangian multiplier. In this context, γ_i corresponds to the SV. The Lagrangian primal problem of (5) is given as

$$\min_{b,w}\max_{\gamma} \quad \mathcal{L}(\mathbf{w}, b, \gamma) \tag{6}$$

s.t.

$$\gamma_i \geq 0, \quad 1, \ldots, n$$

When solving, the derivative of the Lagrangian function is taken with respect to $\mathbf{w}$ and b [4]

$$\frac{\partial\mathcal{L}(\mathbf{w}, b, \gamma)}{\partial\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\gamma_i y_i \mathbf{x}_i = 0 \tag{7}$$

$$\frac{\partial\mathcal{L}(\mathbf{w}, b, \gamma)}{\partial b} = -\sum_{i=1}^{n}\gamma_i y_i = 0 \tag{8}$$

From (7), $\mathbf{w}$ is defined as

$$\mathbf{w} = \sum_{i=1}^{n}\gamma_i y_i \mathbf{x}_i \tag{9}$$

Since $\|\mathbf{w}\|^2$ in (5) is equivalent to $\mathbf{w}.\mathbf{w}$, and with (9), we can rewrite (5) as

$$\mathcal{L}(\mathbf{w}, b, \gamma) = \frac{1}{2}\left(\sum_{i=1}^{n}\gamma_i y_i \mathbf{x}_i\right).\left(\sum_{j=1}^{n}\gamma_j y_j \mathbf{x}_j\right) - \sum_{i=1}^{n}\gamma_i\left[y_i\left(\left(\sum_{j=1}^{n}\gamma_j y_j \mathbf{x}_j\right).\mathbf{x}_i + b\right) - 1\right] \tag{10}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j y_i y_j \mathbf{x}_i.\mathbf{x}_j - \sum_{i=1}^{n}\gamma_i y_i\left(\left(\sum_{j=1}^{n}\gamma_j y_j \mathbf{x}_j\right).\mathbf{x}_i + b\right) + \sum_{i=1}^{n}\gamma_i \tag{11}$$

$$= \sum_{i=1}^{n}\gamma_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j y_i y_j \mathbf{x}_i.\mathbf{x}_j - b\sum_{i=1}^{n}\gamma_i y_i \tag{12}$$

From (8), we have $\sum_{i=1}^{n}\gamma_i y_i = 0$. Therefore, (5) is rewritten as the Wolfe dual Lagrangian problem, which is given as

$$\max_{\gamma} \quad \sum_{i=1}^{n}\gamma_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j y_i y_j \mathbf{x}_i.\mathbf{x}_j \tag{13}$$

s.t.

$$\gamma_i \geq 0, \quad 1, \ldots, n$$

$$\sum_{i=1}^{n}\gamma_i y_i = 0$$

This represents the hard margin SVM optimisation problem, which is suitable for linearly separable data [4]. However, it does not accommodate non-linearly separable data.

B. The Soft Margin SVM Optimisation Problem

Real-world datasets can be both linearly separable and non-linearly separable. Unlike hard-margin SVMs, which aim for zero errors, soft-margin SVMs are designed to allow for a few classification errors. This is achieved by adding a slack variable (typically denoted as β_i) to the objective function in (4). Moreover, to control the soft margin classifier, a new variable, C, is introduced. When $C = 0$, the soft margin SVM reduces to a hard margin SVM.

To this end, the soft margin SVM optimisation is formulated as

$$\min_{b,w,\beta} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\beta_i \tag{14}$$

s.t.

$$y_i(\mathbf{w}^T\mathbf{x} + b) \geq 1 - \beta_i, \quad i = 1, \ldots, n$$
$$\beta_i \geq 0, \quad i = 1, \ldots, n$$

A detailed Lagrangian solution for both l_1-norm and l_2-norm soft margin SVM optimisation problems is presented in [4] as

$$= \sum_{i=1}^{n}\gamma_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j y_i y_j \mathbf{x}_i.\mathbf{x}_j - \frac{1}{4C}\sum_{i=1}^{n}\gamma_i^2 \tag{15}$$

C. The Kernel Trick

When SVMs classify non-linearly separable data, the data is first transformed into a higher-dimensional space and then fed into the SVM as inputs such that $\mathbf{x} \in \mathbb{R}^m \rightarrow \Phi(\mathbf{x})$ [6], where $\mathbb{R}^m \rightarrow \mathbb{R}^k$, that is $\mathbb{R}^m$ is transformed into $\mathbb{R}^k$ on the condition that $k \gg m$. Determining the appropriate dimensionality for transformation is challenging, particularly due to the computational complexity associated with large feature datasets. Introducing a kernel function in (13) alleviates

Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.

this complexity by computing the dot product result directly, as required in (13), without explicitly transforming the data.

D. Hyperparameter Tuning

Hyperparameters in ML algorithms are parameters not directly learned during training but crucial for controlling the model's flexibility and ability to fit data. In SVMs, these include: (i) kernel type, (ii) regularisation strength C, (iii) RBF width α, and (iv) polynomial kernel degree d.

Traditional optimisation methods like gradient descent, grid search, randomized search, and experimental approaches are commonly used to optimise SVM parameters. However, these methods often struggle to find the global optimum. Hence, this study focuses extensively on applying metaheuristic algorithms to enhance SVM performance.

III. Metaheuristics: A Brief Overview

Optimisation techniques are classified as either deterministic or stochastic [7]. Deterministic techniques yield the same results in every run, while stochastic techniques, like metaheuristics, can produce different results in each run.

Metaheuristics are popular due to their easy implementation, simple structure, fast convergence, low computational resource usage, flexibility, robustness, and ability to avoid local optima [8].

In this work, metaheuristics employed in the empirical analysis include the Genetic Algorithm (GA) [9]–[11], Artificial Bee Colony (ABC) [12], Grey Wolf Optimiser (GWO) [13], Differential Evolution (DE) [14], Teaching Learning Based Optimisation (TLBO) [15], Sine-Cosine Algorithm (SCA) [16], Ant Colony Optimisation (ACO) [17], Biogeography-based Optimiser (BBO) [18], Real-Coded Simulated Annealing (RCSA), Harmony Search (HS) [19], and Particle Swarm Optimisation (PSO) [20]. To apply metaheuristics and other optimisation tools in determining the SVs in SVM classifiers, several strategies have been adopted owing to the large dimension of the datasets.

IV. Empirical Analysis of Metaheuristic-based SVM Training

This section presents an empirical analysis of SVM performance, considering cross-validation, number of agents, noise level, and mislabeled targets. The simulations were conducted in a Python Jupyter Notebook on a computer with an Intel Xeon E5-2630 v2 CPU at 2.60GHz (2 processors), 64GB RAM, and an AMD Radeon Graphics Processor. SVM performance was evaluated using benchmark datasets: (i) Breast Cancer Wisconsin (Diagnostic) dataset (BCW) [21], (ii) Iris dataset [22], and (iii) Modified National Institute of Standards and Technology (MNIST) dataset [23].

A. Impact of the number of agents employed by the metaheuristics

The behavioural patterns of agents greatly impact the performance of metaheuristics owing to their interaction with the search space environment and among themselves, consequently impacting the SVM in this case.

1) Analysis of the Iris dataset classification: Figure 1(a) shows that increasing the number of agents in a metaheuristic from 10 to 50, with a maximum of 10 iterations for fairness, enhances SVM accuracy by improving the chances of finding optimal hyperparameters. In Fig. 1(b), SVM accuracy tuned by TLBO, DE, and PSO outperforms other metaheuristics. Figure 1(b) indicates that ACO, TLBO, DE, and PSO yield fewer support vectors, signaling better performance, while RCSA, GWO, and ABC result in more support vectors, indicating lower performance. Additionally, Fig. 1(c) demonstrates that increasing the number of agents leads to longer computational times, with RCSA being the slowest and ACO the quickest to achieve convergence.

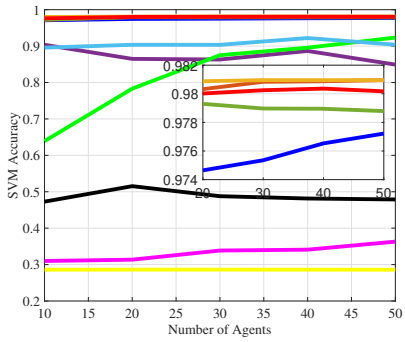
2) Analysis of the Breast Cancer Wisconsin dataset classification: In Figs. 1(d)-1(f), the study focuses on metaheuristic-tuned SVMs applied to the Breast Cancer Wisconsin dataset, considering varying numbers of agents (ranging from 10 to 50) and a maximum of 100 iterations for metaheuristics. Figure 1(d) indicates that as the number of metaheuristic agents increases, SVM accuracy improves, with GA, PSO, and DE outperforming other metaheuristics. Figure 1(e) reveals that the number of support vectors decreases with an increase in the number of agents, primarily due to the ability of more agents to locate optimal hyperparameters. Lastly, in Fig. 1(f), the computational time increases as the number of agents grows, with ACO, HS, and PSO showing better performance, while ABC and TLBO are less efficient in terms of computational time.

3) Analysis of the MNIST dataset classification: Figure 1(g) highlights that the GA, PSO, and DE outperform other metaheuristics in terms of SVM classification accuracy, while GWO, BBO, and HS show the least accuracy. In Fig. 1(h), it is noted that the number of support vectors decreases as the number of agents increases, with PSO, DE, and ACO performing better than other metaheuristics. Figure 1(i) focuses on the computational time of the metaheuristics, where ACO, HS, and PSO are the most efficient, while GWO, BBO, and DE require more time for optimisation. Overall, as observed in Figs. 1(c), 1(f), and 1(i), the computational time of metaheuristics increases with the dataset's dimensionality, considering both the number of data instances and features. According to Occam's razor, among competing entities, in this case, metaheuristics, one should choose the 'simpler' one. In this case, the most appropriate algorithm to use is the one that performs well in terms of the metrics. In Table I, we combine the best metaheuristics for each of the three metrics respectively when applied to the datasets investigated. The simple frequency plot of Table I shows that PSO is more suitable for hyperparameter optimisation for the examined datasets.

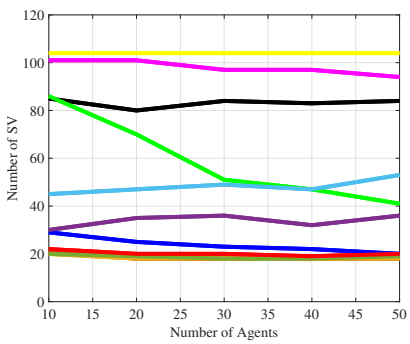
B. Impact of noise on the SVM's classification accuracy

In this subsection, we examine the impact of data noise on the effectiveness of metaheuristics in optimising the SVM's hyperparameters. To this end, a random Gaussian noise is introduced to the attributes of the Iris dataset.

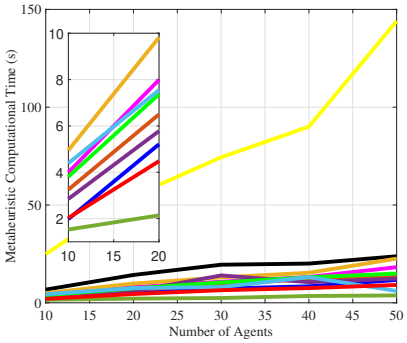
Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.



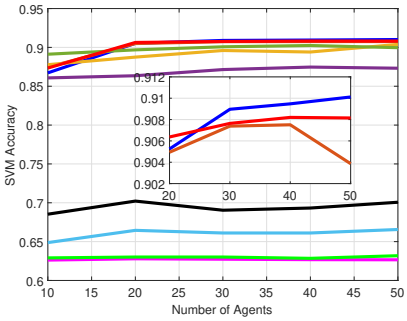
(a) Iris: Impact on the class. accuracy.



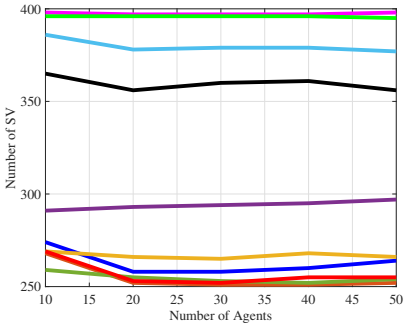
(b) Iris: Impact on the number of SVs.



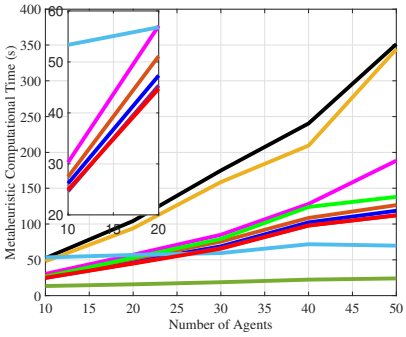
(c) Iris: Impact on the computational time.



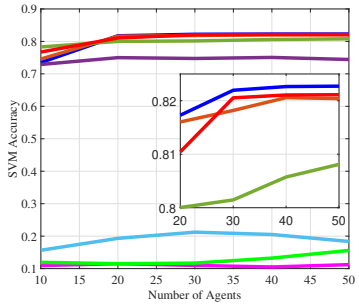
(d) Breast cancer: Impact on the class. accuracy.



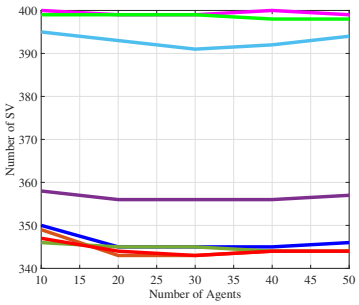
(e) Breast cancer: Impact on the number of SVs.



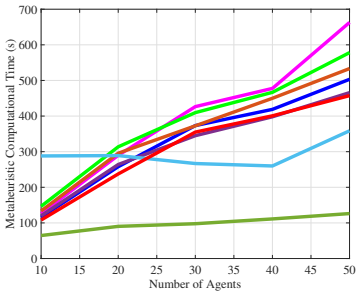
(f) Breast cancer: Impact on the computational time.



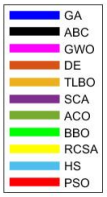
(g) MNIST: Impact on the class. accuracy.



(h) MNIST: Impact on the number of SVs.



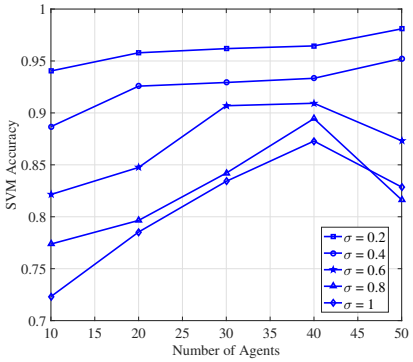
(i) MNIST: Impact on the computational time.



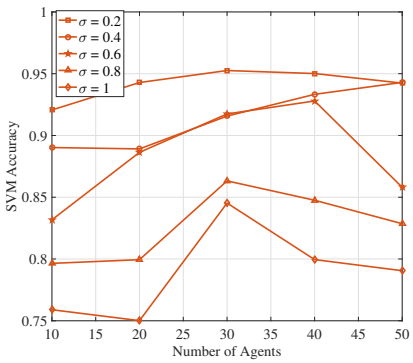
(j) Legend.

Fig. 1. Impact of the number of agents on the performance of the SVM when the Iris, Breast Cancer Wisconsin, and MNIST's datasets are employed.

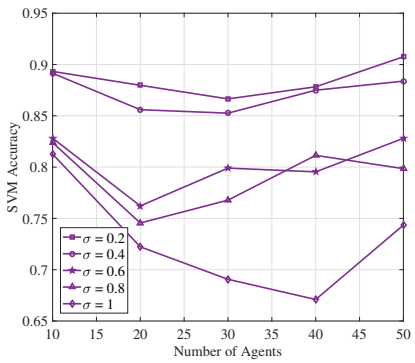
Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.



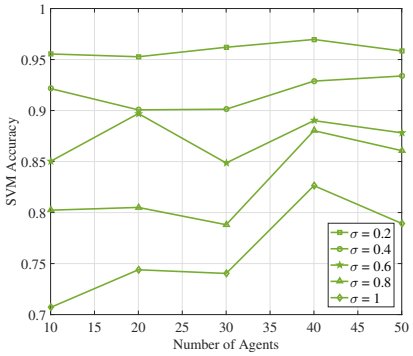
(a) Impact on SVM's accuracy tuned by GA.



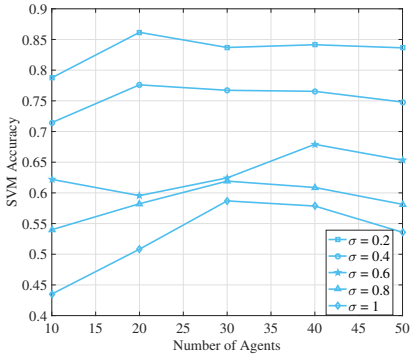
(b) Impact on SVM's accuracy tuned by DE.



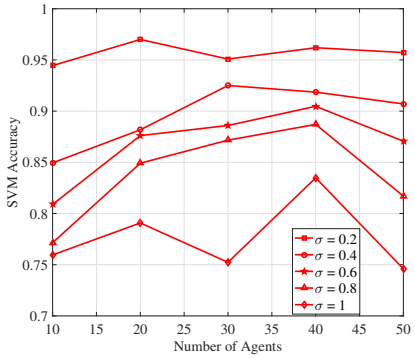
(c) Impact on SVM's accuracy tuned by SCA.



(d) Impact on SVM's accuracy tuned by ACO.

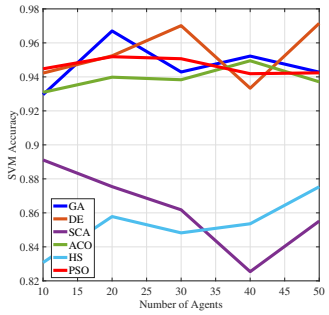


(e) Impact on SVM's accuracy tuned by HS.

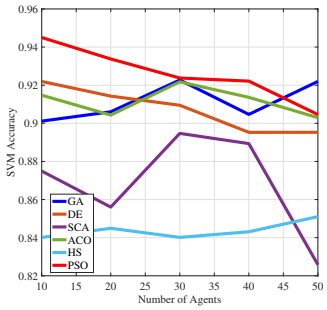


(f) Impact on SVM's accuracy tuned by PSO.

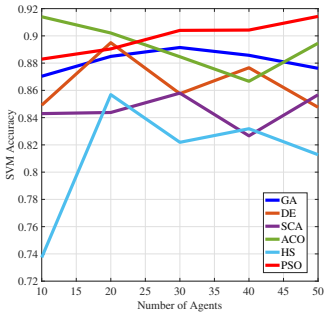
Fig. 2. The impact of varying noise levels in the Iris dataset on the SVM's accuracy with hyperparameters optimised by selected metaheuristics.



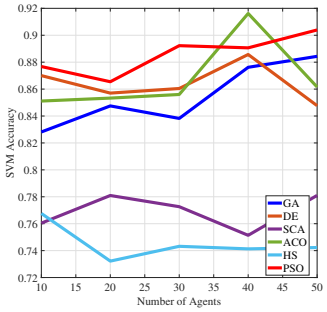
(a) When mislabelled target=5.



(b) When mislabelled target=10.



(c) When mislabelled target=15.



(d) When mislabelled target=20.

Fig. 3. Impact of the mislabelled targets on the classification accuracy of SVM with varying number of mislabelled data settings considering the Iris dataset.

Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.

Gaussian noise with a mean, μ, of 0 and standard deviation, σ, varied between 0 and 1, is introduced into the Iris dataset.

Figure 2 illustrates the SVM's accuracy when tuned by the selected metaheuristics (i.e., GA, DE, SCA, ACO, HS, and PSO) in classifying the Iris dataset in the presence of random Gaussian noise. The selected metaheuristics were chosen based on their performance in the preceding Subsection IV-A.

In Figs. 2(a) - 2(f), we observe a similar trend of a decrease in the SVM's accuracy as the noise level increases (i.e., σ) for all scenarios of metaheuristic agents.

C. Effect of mislabelled targets

We examine the impact of mislabelled targets when the SVM's hyperparameters are optimised by metaheuristics. In the present analysis, the Iris dataset is engaged in the analysis discussed in this subsection; selected metaheuristics, namely GA, DE, SCA, ACO, HS, and PSO, were employed in optimising the hyperparameters of the SVM. Figures 3(a) - 3(d) present the behavioural patterns of the respective metaheuristics for various agent and mislabelled target sizes. The mislabelled target was varied, ranging from 5 to 20 mislabelled targets.

A careful inspection of Figs. 3(a) - 3(d) shows that as the mislabelled target data increases, the SVM accuracy decreases. A simple way of looking at this is to select a specific metaheuristic and then check the accuracy for the different mislabelled target. We observe that PSO, GA, and ACO performed better than others.

V. Conclusion

In this paper, a detailed review of SVM's hyperparameter optimisation and metaheuristics is presented. Specifically, we focus on empirical analysis of tuning the SVM's hyperparameter using metaheuristics. We compared 11 metaheuristics, namely GA, ABC, GWO, DE, TLBO, SCA, ACO, BBO, HS, RCSA, and PSO, in optimising the hyperparameters of the SVMs. We examined the impact of cross-validation techniques, number of agents, noise levels, and mislabelled data on the SVM's classification accuracy, number of SVs, and computational time. Further, we extensively discussed the research challenges associated with the use of SVMs.

TABLE I
RANKING OF THE BEST THREE METAHEURISTICS BASED ON THEIR PERFORMANCE.

Dataset	Accuracy			No. of SVs			Comp. Time(s)		
	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
Iris	TLBO	DE	**PSO**	DE	ACO	TLBO	ACO	**PSO**	GA
Breast Cancer Wisconsin	GA	**PSO**	DE	ACO	DE	**PSO**	ACO	HS	**PSO**
MNIST	GA	DE	**PSO**	**PSO**	DE	ACO	ACO	HS	**PSO**

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, USA: Springer Science & Business Media, 1999.

[2] L. A. Akinyemi, O. P. Oshinuga, O. A. Eshilokun, S. O. Oladejo, and S. O. Ekwe, "Development of phylum chordata sound recognition system using machine learning and deep learning techniques: A case study of catfish," in *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2023, pp. 1–6.

[3] L. A. Akinyemi, O. P. Oshinuga, S. O. Ekwe, and S. O. Oladejo, "Enhancing chronic kidney disease prediction through data preprocessing optimization and machine learning techniques," in *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2023, pp. 1–6.

[4] 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, pp. 189–215, 2020.

[5] S. O. Oladejo, S. O. Ekwe, A. T. Ajibare, L. A. Akinyemi, and S. Mirjalili, "Tuning svms' hyperparameters using the whale optimization algorithm," in *Handbook of Whale Optimization Algorithm*. Elsevier, 2024, pp. 495–521.

[6] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[7] M. H. Davis and I. Karatzas, "A deterministic approach to optimal stopping," *Probability, Statistics and Optimisation (ed. FP Kelly). NewYork Chichester: John Wiley & Sons Ltd*, pp. 455–466, 1994.

[8] S. O. Oladejo, S. O. Ekwe, L. A. Akinyemi, and S. A. Mirjalili, "The deep sleep optimiser: A human-based metaheuristic approach," *IEEE Access*, vol. 11, pp. 83 639–83 665, 2023.

[9] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, Massachusetts, England: MIT Press, 1992.

[10] S. O. Ekwe, S. O. Oladejo, L. A. Akinyemi, and N. Ventura, "A socially-inspired energy-efficient resource allocation algorithm for future wireless network," in *2020 16th International Computer Engineering Conference (ICENCO)*. IEEE, 2020, pp. 168–173.

[11] S. O. Ekwe, L. A. Akinyemi, S. O. Oladejo, and N. Ventura, "Social-aware joint uplink and downlink resource allocation scheme using genetic algorithm," in *2021 IEEE AFRICON*. IEEE, 2021, pp. 1–6.

[12] D. Karaboga, "An idea based on honey bee swarm for numerical optimisation," Technical report-TR06, Erciyes University, Engineering Faculty, Kayseri, Turkey, Tech. Rep., 2005.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimiser," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[14] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimisation over continuous spaces," *Journal of Global Optimisation*, vol. 11, no. 4, pp. 341–359, 1997.

[15] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimisation: A novel method for constrained mechanical design optimisation problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[16] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimisation problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[17] M. Dorigo, "Optimisation, learning and natural algorithms," *Ph. D. Thesis, Politecnico di Milano*, 1992.

[18] D. Simon, "Biogeography-based optimisation," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[19] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimisation algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimisation," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[21] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimisation Methods and Software*, vol. 1, no. 1, pp. 23–34, 1992.

[22] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[23] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

Authorized licensed use limited to: Cape Peninsula University of Technology. Downloaded on November 18,2026 at 08:35:42 UTC from IEEE Xplore. Restrictions apply.