A particle swarm optimisation maximum-margin classifier

1st Sizalobuhle Ncube
Department of Industrial Engineering
Stellenbosch University
Stellenbosch, South Africa
28816110@sun.ac.za

2nd Jean-Pierre van Zyl
Division of Computer Science
Stellenbosch University
Stellenbosch, South Africa
vanzylj@sun.ac.za

3rd Andries Petrus Engelbrecht
Division of Computer Science
Department of Industrial Engineering
Stellenbosch University
Stellenbosch, South Africa
engel@sun.ac.za

***Abstract*—This paper proposes a particle swarm optimisation maximum-margin classifier (PSO-MMC) that borrows core principles from support vector machines (SVMs). The task of identifying an optimal linear separating hyperplane is formulated as a constrained optimisation problem defined exclusively over training instances that form Tomek links. For non-linearly separable data, kernel mappings are incorporated in a manner analogous to the SVM kernel formulation. Prior to Tomek link identification, noise is removed from the training data using established filtering techniques. Experimental evaluation is conducted against classical SVMs, classification trees and k-nearest neighbour (KNN) across multiple benchmark data sets. The results show that the proposed classifier achieves improved classification accuracy and reduced optimisation complexity, particularly under conditions of class overlap and feature redundancy. The findings indicate that restricting optimisation to boundary-critical instances provides an efficient and scalable alternative model for maximum-margin classification.**

***Index Terms*—particle swarm optimisation, maximum margin classifier, non-linearly separable classification, tomek links**

I. Introduction

A central objective of modern computational science is the development of intelligent systems that produce accurate, interpretable, and scalable decisions. Margin-based classification, particularly SVMs, creates a maximal separation between the decision boundary and the closest instances from opposing classes in the feature space [8]. Training of SVMs is formulated as a constrained quadratic optimisation problem grounded in statistical learning theory [25]. Kernel-based extensions enable the construction of linear decision boundaries in transformed feature spaces and allow effective classification of nonlinearly separable data. Training complexity increases for large, noisy, or high-dimensional datasets due to reliance on quadratic programming (QP) solvers and optimisation over the complete set of training instances.

Metaheuristic optimisation techniques provide gradient-free alternatives for classifier training in complex, non-convex search spaces [10], [12]. Particle swarm optimisation (PSO) has been applied successfully to SVM training as a population-based alternative to conventional QP solvers [15]. Existing PSO-SVM approaches typically perform optimisation over the complete training set or employ random instance selection strategies [16]. The strategies increase computational cost, slow convergence, and allocate optimisation effort to interior or redundant instances that do not influence margin placement [3]. As a result, efficiency is reduced and generalisation performance is degraded [20].

In practical decision-making environments, datasets often have class overlap, noisy instances, and redundant or correlated features. Classifiers that do not account for these conditions produce inconsistent predictions and poor generalisation, which is particularly problematic when outcomes influence automated decisions, risk assessment, or resource allocation. Consequently, classification models must maintain reliable performance, ensure computational efficiency, and demonstrate resilience to adverse data conditions. Training strategies should prioritise informative instances and avoid unnecessary optimisation over samples that do not contribute meaningfully to decision boundary formation.

The paper proposes a PSO-MMC that restricts optimisation to boundary-defining instances identified through Tomek links. Tomek links isolate borderline samples located near class boundaries. The samples directly influence the position of the separating hyperplane. The restriction of the optimisation to informative boundary instances reduces the search space and preserves the key information that defines the margin. The remainder of the paper is organised as follows: Section 2 reviews the foundational concepts of SVMs, PSO, and Tomek links. The methodology of the proposed model is presented in Section 3. Section 4 details the datasets, preprocessing steps, baseline models, evaluation metrics, and implementation specifics. Section 5 reports the experimental results, while Section 6 discusses these results and outlines directions for future work.

II. Background

This section outlines the foundational concepts central to the study.

A. Particle Swarm Optimisation

PSO is a population-based stochastic optimisation algorithm inspired by the social behaviour of birds flocking [13]. Each particle represents a candidate solution defined by a position vector, $\mathbf{x}_i(t) \in \mathbb{R}^d$, and a velocity vector, $\mathbf{v}_i(t) \in \mathbb{R}^d$, where d is the dimensionality of the search space and t denotes

the iteration index. Positions are randomly initialised within predefined bounds, while initial velocities are set to zero [11] as

$$\mathbf{v}_i(0) = \mathbf{0}, \quad \mathbf{x}_i(0) \sim \mathcal{U}(\mathbf{x}_{\min}, \mathbf{x}_{\max}). \tag{1}$$

The quality of the solution represented by each particle is evaluated using an objective function, $f : \mathbb{R}^d \to \mathbb{R}$. The personal best (pbest) position, denoted by, $\mathbf{y}_i(t)$, is updated whenever the current position yields a better objective value than the stored pbest. Assuming a minimisation objective, the personal best position is updated as $\mathbf{y}_i(t) = \mathbf{x}_i(t)$ if $f(\mathbf{x}_i(t)) \leq f(\mathbf{y}_i(t))$.

Particles incorporate information from the neighbourhood best position, denoted by $\hat{\mathbf{y}}_i(t)$, to guide the movement. The best position of the neighbourhood corresponds to the best solution identified within the defined neighbourhood up to iteration t. Under a fully connected topology, the neighbourhood consists of all particles in the swarm, whereas under local topologies, the neighbourhood is limited to a subset of particles.

Particle velocities are updated using a combination of three components, *i.e.,* inertia, cognitive, and social contributions. The velocity update equation is

$$\begin{aligned} v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1ij}(t) \left[y_{ij}(t) - x_{ij}(t)\right] \\ + c_2 r_{2ij}(t) \left[\hat{y}_{ij}(t) - x_{ij}(t)\right], \end{aligned} \tag{2}$$

where c_1 and c_2 are acceleration coefficients that control cognitive and social behaviour respectively, ω is the inertia weight that regulates the influence of prior velocity and $r_{1ij}(t), r_{2ij}(t) \sim \mathcal{U}(0,1)$ introduce stochasticity. The integration of personal and neighbourhood knowledge enables PSO to balance exploration of the search space with exploitation of high-quality regions and drive convergence towards optimal or near-optimal solutions [14].

B. Support Vector Machines

The main objective of a SVM is to identify a hyperplane that maximises the margin of separation between instances of different classes [1]. Given a training dataset of n labelled samples, with each input vector $\mathbf{x}_i \in \mathbb{R}^d$ and corresponding class label $y_i \in \{-1, +1\}$, the goal of a classifier is to discover a linear boundary in the input space [1]. A hyperplane, defined by a linear equation that includes a normal vector $\mathbf{w}$ and a scalar bias term b, represents the set of points $\mathbf{x} \in \mathbb{R}^d$. The hyperplane is described by the equation

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0. \tag{3}$$

For linearly separable data, the optimisation problem is

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

subject to

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, n \tag{5}$$

which yields a unique optimal separating hyperplane. Using Lagrange duality with non-negative multipliers, the dual optimisation problem is obtained as

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^\top \mathbf{x}_j), \tag{6}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \text{and} \quad \alpha_i \geq 0. \tag{7}$$

The optimal weight vector is

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i. \tag{8}$$

After reconstructing the weight vector, the bias term b is computed using a training sample that satisfies the support vector condition. For a support vector $(\mathbf{x}_i, y_i)$ with $\alpha_i > 0$, the bias term is calculated as $b = y_i - \mathbf{w}^\top \mathbf{x}_i$. Only samples with $\alpha_i > 0$ contribute to the solution and constitute the set of support vectors. To accommodate non-separable and noisy datasets, slack variables $\xi_i \geq 0$ are introduced [2]. The resulting soft-margin primal optimisation problem is then defined as

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i, \tag{9}$$

subject to $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$. The regularisation parameter $C > 0$ controls the trade-off between margin width and classification error, which leads to the dual constraints $0 \leq \alpha_i \leq C$. Many real-world problems have nonlinear class boundaries. SVMs addresses the limitation through kernel functions, which implicitly map the input space to a higher-dimensional feature space. In the dual formulation, the inner product is replaced by a kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j), \tag{10}$$

where $\phi(\cdot)$ denotes a mapping from the input space to the feature space [19]. The substitution, referred to as the kernel trick, reformulates the optimisation problem without explicit computation of $\phi(\cdot)$. The reformulation preserves computational tractability and convexity, and permits nonlinear separation in the original input space. The dual optimisation problem becomes

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{11}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. The corresponding decision function is

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{12}$$

Commonly used kernels include the polynomial kernel, which captures feature interactions up to a given degree, the radial basis function (RBF) kernel, which measures similarity based on distance in the feature space, and the sigmoid kernel, which behaves similarly to neural network activation functions.

C. Tomek Links

Tomek links are defined using a reciprocal nearest neighbour relationship between instances that belong to opposite classes [22]. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote a data instance. A pair $(\mathbf{x}_i, \mathbf{x}_j)$ forms a Tomek link if $\mathbf{x}_j$ is the nearest neighbour of $\mathbf{x}_i$, $\mathbf{x}_i$ is the nearest neighbour of $\mathbf{x}_j$, and $y_i \neq y_j$ [22]. Such pairs may correspond to samples located near the class boundary, noisy instance, or mislabelled instances. In margin-based classification, only samples near the decision boundary, *i.e.,* potential support vectors, significantly influence hyperplane optimisation. By training exclusively on Tomek link instances, the algorithm focuses on the critical, boundary-adjacent samples while also identifying instances that may represent noise [5]. The approach reduces problem dimensionality, sharpens the decision boundary, and improves both stability and generalisation, consistent with the core principles of maximum-margin classifiers and instance selection theory [25].

III. Particle Swarm Optimisation Maximum-Margin Classifier

Consider a binary classification problem with a training set $\{(\mathbf{x}_k, y_k)\}_{k=1}^{n_T}$, where $\mathbf{x}_k \in \mathbb{R}^d$ denotes a feature vector, n_T denotes the number of training instances after Tomek link selection, and $y_k \in \{-1, +1\}$ denotes the corresponding class label. All features are normalised to the interval $[0, 1]$ to ensure numerical stability during optimisation. The classifier seeks a separating hyperplane that maximises the margin between classes. In a kernel-based formulation, the decision function is defined as

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{k=1}^{n_T} \alpha_k y_k K(\mathbf{x}, \mathbf{x}_k) + b\right), \tag{13}$$

where $\alpha_k \geq 0$ are Lagrange multipliers, b is the bias term, and $K(\cdot, \cdot)$ denotes a kernel function. The training process is formulated as an optimisation problem that maximises the margin subject to a penalty term associated with margin violations. The objective function is given by

$$\begin{aligned} f(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{k=1}^{n_T} \sum_{j=1}^{n_T} \alpha_k \alpha_j y_k y_j K(\mathbf{x}_k, \mathbf{x}_j) \\ + C \sum_{k=1}^{n_T} \max\big(0,\ 1 - y_k\, f(\mathbf{x}_k)\big). \end{aligned} \tag{14}$$

where $C > 0$ controls the trade-off between margin maximisation and classification error. PSO is employed to minimise $f(\boldsymbol{\alpha})$ in a derivative-free manner. Each particle i represents a candidate solution. The solution is defined by a position vector $\mathbf{x}_i(t) = (\boldsymbol{\alpha}_i) \in \mathbb{R}^{n_T}$, where $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,n_T})$ denotes the vector of Lagrange multipliers associated with particle i. The quality of each particle is evaluated using the objective function in Equation (14).

Particle velocities and positions are updated using Equation (2). The swarm iteratively converges toward the optimal multiplier vector $\boldsymbol{\alpha}^*$ that minimises $f(\boldsymbol{\alpha})$. Convergence is declared when the best global objective value remains unchanged for a fixed number of iterations. Alternatively, convergence is declared when a preset maximum iteration count is reached.

Non-negativity constraints on the multipliers $\alpha_k \geq 0$ are enforced by projecting each updated multiplier onto the feasible set $[0, \infty)$ after every optimisation step, such that any negative value is set to zero.

he bias term b is computed using a training instance that satisfies the support vector condition. For a support vector $(\mathbf{x}_i, y_i)$ with $\alpha_i > 0$, the bias term is obtained as

$$b = y_i - \sum_{k=1}^{n_T} \alpha_k y_k K(\mathbf{x}_k, \mathbf{x}_i). \tag{15}$$

The decision function in Equation (13) defines the separating hyperplane that maximises the margin and minimises classification error.

IV. Experimental Setup

This section describes the experimental setup used to assess the performance of the proposed PSO-MMC.

A. Datasets

The proposed PSO-MMC was evaluated on 11 binary classification tasks, four synthetic datasets (i.e., datasets 1 to 4) and seven real-world benchmark datasets obtained from the UCI machine learning repository and Kaggle. All datasets are summarised in Table I. All datasets were preprocessed through the removal of instances with missing values, the standardisation of features, and the encoding of class labels as $y_i \in \{-1, +1\}$.

TABLE I
Summary of datasets used in evaluation.

Dataset	Instances	Features
Dataset 1 (linear)	1000	10
Dataset 2 (noisy linear)	1000	10
Dataset 3 (moons)	1000	10
Dataset 4 (circles)	1000	10
Banknote	1372	4
Blood Transfusion	748	4
Breast Cancer	569	30
Heart Disease	303	13
ILPD	583	10
Pima Diabetes	768	9
Titanic	891	11

B. Noise filtering and Tomek links selection

Noise filtering was performed using synthetic minority over-sampling technique-edited nearest neighbours (SMOTE-ENN) [7], [26], which combines synthetic minority over-sampling technique (SMOTE) with edited nearest neighbour (ENN) cleaning. SMOTE generates synthetic minority instances via interpolation between each minority sample and its nearest minority neighbours. ENN then removes any instance whose label disagrees with the majority of the k nearest neighbours. The SMOTE-ENN class from the `imbalanced-learn` library was used for implementation [1]. Five nearest neighbours

[1]The `imbalanced-learn` library is available at: https://imbalanced-learn.org

were used for oversampling, and three neighbours were used for the ENN cleaning step. The sampling strategy was adjusted according to the degree of class imbalance in each data set [19].

Following noise filtering, Tomek links were identified using a reciprocal nearest neighbour criterion. A KNN search with $k = 1$ was performed using the Euclidean distance metric over the filtered training set. For each instance $\mathbf{x}_i \in X$, its nearest neighbour $\mathbf{x}_j$ was identified. A Tomek link was defined between the pair $(\mathbf{x}_i, \mathbf{x}_j)$ if the two instances belonged to different classes and if $\mathbf{x}_i$ was the nearest neighbour of $\mathbf{x}_j$ and $\mathbf{x}_j$ was the nearest neighbour of $\mathbf{x}_i$ [22]. Only instances that participated in at least one Tomek link were retained, forming the refined training subset $(X_{TL}, \mathbf{y}_{TL})$.

C. Algorithm Tuning

All classifiers, *i.e.*, PSO-MMC, SVM, KNN, and decision trees, were independently tuned per data set using Sobol quasi-random sampling [21] in 128 configurations. A swarm size of 30 particles and 300 iterations per configuration were used. The best-performing parameter set, as determined by cross-validation accuracy on the training set, was selected for final evaluation using a 70/30 train–test split. The PSO parameters satisfied the stability condition $\omega > \frac{1}{2}(c_1 + c_2) - 1$ [23]. Hyperparameters for PSO, SVM, KNN, and decision trees were selected from predefined ranges summarised in Table II. All results report averages over 30 independent runs. The SVM was trained using the sequential minimal optimisation (SMO) algorithm [17], as implemented in the LIBSVM library [6]. [2]

TABLE II
PARAMETER RANGES

Algorithm	Parameter	Range
PSO	c_1	$[0.0, 4.0]$
	c_2	$[0.0, 4.0]$
	ω	$[0.1, 1.0]$
	C	$[10^{-2}, 10^{2}]$
SVM	d	$[2, 3, 4, 5]$
	c_0	$[-1, 1]$
	C	$[10^{-2}, 10^{2}]$
	γ	$[0.0, 1.0]$
Decision Trees	max_depth	$[3, 5, 10, 20, 30]$
	min_samples_split	$[2, 20]$
	min_samples_leaf	$[1, 10]$
KNN	k	$[1, 8]$

D. Performance Measures

Classifier performance was evaluated using accuracy, F1-score, balanced accuracy (BA), and the normalised F_1-measure, F_ι [24]. The latter two metrics were included to account for class imbalance. BA was computed as the average of the true positive rate (TPR) and true negative rate (TNR). The normalised F_ι was defined in terms of error rates rather than raw counts, *i.e.*

$$F_\iota = \frac{2 \cdot \mathrm{TPR}}{2 \cdot \mathrm{TPR} + \mathrm{FNR} + \mathrm{FPR}}. \tag{16}$$

[2]The `scikit-learn` library, available at: https://scikit-learn.org

To assess optimisation stability, the following measures were computed: support vector consistency, *i.e.* the average Jaccard index across support vector sets obtained from 30 independent runs [4], [18]; and decision function variance, *i.e.* the variance of the decision function, as defined in Equation (13), computed over all test instances and averaged across all experimental runs [9]. Low decision function variance and high support vector consistency indicated robust and reproducible decision boundaries.

Auxiliary metrics included model complexity, i.e., measured by support vector count for SVMs and total training time. The proposed PSO-based classifier was compared against three baselines classifiers: standard SVMs, KNN, and decision trees. All classifiers were trained and evaluated under identical preprocessing and experimental conditions.

E. Statistical Comparison Procedure

A non-parametric statistical framework was employed to evaluate differences in performance between the selected models. The Wilcoxon signed-rank test was selected due to the paired nature of the observations and the absence of an assumption of normality. For each dataset, each model was evaluated over 30 independent runs. The differences in BA were computed across the 30 runs for each dataset. The Wilcoxon signed-rank test was then applied to determine whether the median of the paired differences deviates significantly from zero.

To control the family-wise error rate arising from multiple pairwise comparisons, the Holm correction procedure was applied. The significance level was set to $\alpha = 0.05$. Adjusted p-values below the threshold indicate statistically significant differences in performance. The procedure ensures that the reported differences between models are statistically valid and not attributable to random variation within each dataset.

V. RESULTS

The performance metrics were averaged across all 11 datasets for each classifier, kernel, and preprocessing configuration. Table III shows that the PSO-MMC consistently outperformed all baselines. The RBF kernel on Tomek link subsets achieved the highest scores of 95.82% accuracy.

SVM in Table VI, achieved strong, but consistently lower performance. Under the same RBF Tomek configuration, SVM obtained 90.87% accuracy and 89.93% F1-score. The results indicated that the PSO-MMC produced more effective separating hyperplanes than conventional QP under identical data conditions.

In contrast, the instance-based methods, KNN and decision trees showed performance degradation on Tomek link subsets, results are presented in Table V. While KNN maintained moderate accuracy 79.77%, decision trees dropped sharply to 69.02% when trained on Tomek Link subsets. The results indicate a reliance on interior data points, unlike margin-based optimisers.

Table VII presents the BA in percent for the evaluated models across the 11 datasets. The results show that PSO with

a RBF kernel, trained on the Tomek links subset, achieved higher performance compared to the baseline classifiers. The application of the Wilcoxon signed-rank test with Holm correction at $\alpha = 0.05$ confirmed that the observed differences were not due to random variation.

Table VI shows the pairwise statistical comparisons. The results show that PSO-MMC achieved statistically significant improvements over all baseline models. The statistically significant improvements confirmed consistent performance gains across datasets. In contrast, the comparisons between SVM, KNN, and the decision tree were not statistically significant. The absence of statistical significance indicated similar performance levels among the baseline classifiers under the same conditions.

TABLE III
THE AVERAGE PERFORMANCE ACROSS ALL DATASETS FOR THE PSO-BASED MAXIMUM-MARGIN CLASSIFIER

Configuration	Accuracy (%)	F1-score (%)	Balanced Accuracy (%)	F_L-Measure (%)
Linear Before Filtering	83.53 ± 2.61	82.42 ± 2.83	82.30 ± 2.54	75.97 ± 2.91
Linear After Filtering	86.57 ± 2.00	85.63 ± 2.11	86.83 ± 1.92	78.79 ± 2.21
Linear Tomek Links Subset	88.99 ± 1.61	88.10 ± 1.73	89.27 ± 1.55	80.81 ± 1.83
Polynomial Before Filtering	86.47 ± 2.71	85.50 ± 2.92	85.20 ± 2.61	78.72 ± 2.81
Polynomial After Filtering	89.69 ± 2.11	88.91 ± 2.23	90.06 ± 2.00	81.73 ± 2.31
Polynomial Tomek Links Subset	92.88 ± 1.61	92.08 ± 1.75	93.45 ± 1.55	84.78 ± 1.83
RBF Before Filtering	89.13 ± 2.31	88.30 ± 2.44	89.84 ± 2.11	81.63 ± 2.56
RBF After Filtering	92.59 ± 1.76	91.86 ± 1.88	92.76 ± 1.66	84.23 ± 1.95
RBF Tomek Links Subset	**95.82** ± 1.21	**95.13** ± 1.33	**96.37** ± 1.00	**87.26** ± 0.91

TABLE IV
THE AVERAGE PERFORMANCE ACROSS ALL DATASETS FOR SUPPORT VECTOR MACHINE

Configuration	Accuracy (%)	F1-score (%)	Balanced Accuracy (%)	F_L-Measure (%)
Linear Before Filtering	78.53 ± 2.61	77.26 ± 2.83	78.65 ± 2.51	74.52 ± 2.33
Linear After Filtering	82.37 ± 2.11	81.23 ± 2.24	82.98 ± 2.01	77.93 ± 1.90
Linear Tomek Links Subset	84.21 ± 1.72	83.01 ± 1.83	85.31 ± 1.62	79.79 ± 1.53
Polynomial Before Filtering	80.12 ± 2.91	79.14 ± 3.00	80.41 ± 2.84	77.60 ± 2.61
Polynomial After Filtering	84.85 ± 2.22	83.92 ± 2.31	85.37 ± 2.10	81.21 ± 2.00
Polynomial Tomek Links Subset	87.87 ± 1.83	86.91 ± 1.92	88.55 ± 1.71	83.98 ± 1.61
RBF Before Filtering	83.42 ± 2.41	82.39 ± 2.60	83.92 ± 2.33	80.81 ± 2.11
RBF After Filtering	87.94 ± 1.92	87.01 ± 2.01	88.72 ± 1.08	84.47 ± 1.75
RBF Tomek Links Subset	**90.87 ± 1.31**	**89.93 ± 1.40**	**91.68 ± 1.21**	**87.39 ± 1.14**

TABLE V
THE AVERAGE PERFORMANCE ACROSS ALL DATASETS FOR DECISION TREES AND K-NEAREST NEIGHBOURS

Configuration	Accuracy (%)	F1-score (%)	Balanced Accuracy (%)	F_L-Measure (%)
KNN Before Filtering	**84.58 ± 3.33**	**83.83 ± 3.87**	**85.29 ± 2.01**	**86.65 ± 1.65**
KNN After Filtering	77.72 ± 3.51	77.24 ± 3.91	77.49 ± 3.66	79.39 ± 3.81
KNN Tomek Links Subset	79.77 ± 3.21	79.13 ± 3.28	79.76 ± 3.24	88.34 ± 1.87
Decision Tree Before Filtering	**83.42 ± 2.79**	**80.69 ± 2.89**	**81.38 ± 2.85**	**84.86 ± 1.96**
Decision Tree After Filtering	77.89 ± 3.54	77.13 ± 3.19	77.51 ± 3.25	80.75 ± 3.01
Decision Tree Tomek Links Subset	69.02 ± 4.66	68.69 ± 4.89	65.90 ± 5.25	90.39 ± 5.45

TABLE VI
WILCOXON SIGNED-RANK TEST WITH HOLM CORRECTION FOR PAIRWISE MODEL COMPARISON

Comparison	Statistic (W)	Adjusted p-value (Holm)	Significant
PSOMMC vs KNN	0.0	0.005859	YES
PSOMMC vs Decision Tree	0.0	0.004883	YES
PSOMMC vs SVM	3.0	0.019531	YES
SVM vs KNN	5.0	0.029297	YES
KNN vs Decision Tree	10.0	0.083984	NO
SVM vs Decision Tree	12.0	0.067383	NO

TABLE VII
BALANCED ACCURACY (%) ACROSS DATASETS

Dataset	PSOMMC RBF Tomek Links Subset	SVM RBF Tomek Links Subset	KNN After Filtering	Decision trees After Filtering
Indian Liver Patient	**99.04 ± 1.20**†	89.85 ± 2.09	79.90 ± 2.05	90.64 ± 1.76
Titanic	**96.25 ± 0.95**†	89.70 ± 1.15	80.15 ± 2.05	83.06 ± 2.60
Blood Transfusion Service	**95.41 ± 3.04**†	85.31 ± 1.52	88.84 ± 1.79	90.89 ± 1.65
Banknote Authentication	**99.93 ± 0.10**†	99.69 ± 0.31	95.45 ± 2.75	96.54 ± 2.11
Breast Cancer	**98.83 ± 0.88**†	98.35 ± 0.81	85.50 ± 1.00	87.57 ± 1.43
Pima Indians Diabetes	**91.34 ± 2.41**†	86.41 ± 2.74	89.50 ± 1.01	90.57 ± 1.43
Heart Disease	93.58 ± 1.19	**94.09 ± 1.14**	84.10 ± 3.55	85.05 ± 3.25
Dataset 1	**100.00 ± 0.00**†	96.92 ± 0.75	92.15 ± 0.85	93.65 ± 1.06
Data Set 2	**96.06 ± 2.73**†	89.37 ± 2.38	85.11 ± 2.19	86.39 ± 2.21
Data Set 3	**99.31 ± 1.65**†	92.72 ± 2.74	85.73 ± 3.05	86.95 ± 2.01
Data Set 4	**90.32 ± 2.51**†	85.32 ± 3.01	82.59 ± 3.49	76.45 ± 3.09

† Statistically significant at $\alpha = 0.05$ (Wilcoxon signed-rank test with Holm correction).

Table VIII presents the average training times of the PSO-MMC across the 11 datasets, before and after Tomek link instance selection. Training without filtering required higher computational time across all datasets. Application of Tomek links reduced the number of training instances. The reduction led to shorter training times for all datasets. The percentage reduction in training time remained consistent across datasets. The results demonstrate that Tomek link filtering reduces computational time without modifying the optimisation procedure of the PSO-MMC.

TABLE VIII
AVERAGE PARTICLE SWARM OPTIMISATION MAXIMUM-MARGIN CLASSIFIER TRAINING TIME (S) ACROSS DATASETS

Dataset	Average Runtime Before Filtering (s)	Average Runtime After Tomek Links (s)	Reduction (%)
Indian Liver Patient	152.43 ± 14.72	76.31 ± 7.91	49.95 ± 3.12
Titanic	179.63 ± 19.84	92.12 ± 9.71	48.70 ± 3.42
Banknote	123.54 ± 12.31	71.84 ± 7.93	41.90 ± 2.57
Breast Cancer	198.72 ± 24.92	92.43 ± 11.81	53.50 ± 3.63
Pima Diabetes	141.79 ± 14.64	69.71 ± 7.83	50.80 ± 3.04
Blood Transfusion	191.79 ± 28.63	79.71 ± 9.33	50.80 ± 3.00
Heart Disease	132.21 ± 12.11	64.91 ± 6.82	50.92 ± 3.01
Dataset 1	94.72 ± 9.78	54.32 ± 5.23	42.72 ± 2.12
Dataset 2	108.61 ± 12.11	61.23 ± 6.94	43.67 ± 2.41
Dataset 3	146.53 ± 15.23	77.84 ± 7.92	46.91 ± 2.58
Dataset 4	159.33 ± 17.63	91.71 ± 9.81	42.54 ± 2.91

VI. DISCUSSION

This section examines the main observations derived from the empirical results presented in Table VII and Table VI.

A. Particle swarm optimisation as an alternative to quadratic programming-based optimisation

The PSO-MMC exceeded the performance of the SVM across datasets. The largest performance differences were observed under the RBF kernel with Tomek link selection. The largest performance differences were observed on the Indian liver patient dataset (9.19%), the Titanic dataset (6.55%), and dataset 3 (6.59%) with performance differences ranging from 6.55% to 9.19%. The Wilcoxon signed-rank test with Holm correction confirmed statistical significance for the comparison between the PSO-MMC and the SVM, with an adjusted p-value of 0.019531, as reported in Table VI. The results demonstrate that PSO-MMC produced a separating hyperplanes with improved generalisation performance compared to SVM.

B. Impact of Tomek link selection on margin-based classifiers

Tomek link selection improved the performance of margin-based classifiers, particularly for PSO-MMC. PSO-MMC achieved the highest BA when trained on Tomek link subsets. For example, a BA of 99.93% for the banknote authentication dataset and 98.83% for the breast cancer dataset were obtained under the RBF kernel. Similar improvements were observed for SVM. However, PSO-MMC maintained large BA across all datasets. In contrast, the KNN classifier and the decision tree showed reduced performance when trained on Tomek link subsets. The KNN classifier achieved 79.90% BA on the Indian liver patient dataset and 80.15% on the Titanic dataset. The decision tree achieved 76.45% on dataset 4. The results indicate that training only on critical instances reduces the effectiveness of classifiers that rely on the global data distribution.

C. Effectiveness of kernel-based optimisation for nonlinear problems

The RBF kernel produced large BA compared to the linear kernel on nonlinear datasets. The PSO-MMC achieved 96.06% on dataset 2 and 99.31% on dataset 3 using the RBF kernel. The linear kernel achieved 100.00% on dataset 1, which corresponds to a linearly separable dataset. The results demonstrate that kernel transformations improved classification performance for nonlinear data and maintained strong performance on linear data. The global search mechanism of PSO facilitates improved adaptation to complex kernel-induced feature spaces compared to gradient-based methods.

D. Stability compared with conventional classifiers

The results in Table VII reveal contrasting behaviour for KNN and decision tree classifiers. Unlike margin-based methods, the classifiers showed performance degradation after filtering and Tomek link restriction. BA declined, particularly for decision trees trained exclusively on Tomek link subsets. The results indicate that Tomek link selection was most suitable for margin-based classifiers. Instance-based and tree-based classifiers depend on global data density or hierarchical partitioning and therefore suffered when training data were restricted to boundary samples.

VII. Conclusion

The goal of this study was to develop a scalable and efficient maximum-margin classifier. The PSO-MMC was developed and trained on a reduced training set composed exclusively of Tomek link instances, while kernel mappings were employed to handle nonlinear decision boundaries. The main findings are as follows. The restriction of optimisation to Tomek link instances improved computational efficiency and classification performance. Kernel-based extensions enabled effective handling of nonlinearly separable data. The proposed classifier demonstrated higher stability and accuracy compared to baseline models under challenging conditions. The results confirm that the combined use of PSO, Tomek links selection, and kernel methods provides a reliable and scalable approach to maximum-margin classification in practical machine learning applications.

Future studies should prioritise evaluation on dynamic, real-world datasets. Application to streaming, evolving, and highly imbalanced data distributions would provide insight into PSO-MMC adaptability under operational conditions. Integration with incremental learning strategies could further extend applicability to non-stationary environments. The findings establish a strong foundation for the continued development of optimisation-driven margin-based classification methods in practical machine learning settings

References

[1] Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2009.

[2] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.

[3] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[4] Ioan Buciu, Constantine Kotropoulos, and Ioannis Pitas. Demonstrating the stability of support vector machines for classification. *Signal Processing*, 86(9):2364–2380, 2006.

[5] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] V. Dhar and H. Yu. On the stability of machine learning models: Measuring model and outcome variance. *Journal of Investment Management*, 18(2):9–22, 2020.

[10] R. Eberhart and J. Kennedy. *Particle Swarm Optimization*. Institute of Electrical and Electronics Engineers, Perth, Australia, 1995.

[11] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005.

[12] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[13] J. Kennedy and R. Eberhart. Particle swarm optimisation. *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[14] Salman A Khan and Andries P Engelbrecht. A fuzzy particle swarm optimisation algorithm for computer communication network topology design. *Applied Intelligence*, 36:161–177, 2012.

[15] Shih-Wei Lin, Kuo-Ching Ying, Shih-Chieh Chen, and Zne-Jung Lee. Particle swarm optimisation for parameter determination and feature selection of support vector machines. *Expert systems with applications*, 35(4):1817–1824, 2008.

[16] Vitthal Manekar and Kalyani Waghmare. Intrusion detection system using support vector machine (svm) and particle swarm optimisation (pso). *International Journal of Advanced Computer Research*, 4(3):808, 2014.

[17] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, Redmond, WA, USA, 1998.

[18] Raimundo Real and Juan M Vargas. The probabilistic basis of jaccard's index of similarity. *Systematic Biology*, 45(3):380–385, 1996.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularisation, Optimisation, and Beyond*. MIT Press, 2002.

[20] Liming Shen, Huiling Chen, Zhe Yu, Wenchang Kang, Bingyu Zhang, Huaizhong Li, Bo Yang, and Dayou Liu. Evolving support vector machines using fruit fly optimisation for medical data classification. *Knowledge-Based Systems*, 96:61–75, 2016.

[21] I. M. Sobol. Distribution of points in a cube and approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

[22] I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.

[23] F. van den Bergh and A. P. Engelbrecht. A study of particle swarm optimisation particle trajectories. *Information Sciences*, 176(8):937–971, 2006.

[24] Jean-Pierre van Zyl and Andries Petrus Engelbrecht. Analysis of classification metric behaviour under class imbalance. *Egyptian Informatics Journal*, 31:100711, 2025.

[25] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[26] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972.