Optimized Dimensionality Reduction Using Metaheuristic and Class Separability

Eman Abdulazeem Ahmed[1], Malek Alzaqebah[2], Sana Jawarneh[3]
Department of Mathematics-College of Science, Imam Abdulrahman Bin Faisal University, Saudi Arabia[1, 2]
Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Saudi Arabia[1, 2]
Computer Science Department-Community College Dammam, Imam Abdulrahman Bin Faisal University, Saudi Arabia[3]

***Abstract*—The high dimensionality of modern datasets presents significant challenges for machine learning, including increased computational cost, model complexity, and risk of overfitting. This study introduces a metaheuristic framework for optimized dimensionality reduction to identify the highly discriminative feature subsets. The proposed method (KDR-PSO) combines a Particle Swarm Optimization (PSO) algorithm with the K-Nearest Neighbors Distance Ratio (KDR) as a filter-based objective function. This metric quantitatively assesses class separability within a feature subspace by computing the ratio of the average distance from a sample to neighbors in other classes versus those in its own class. Maximizing this ratio with a penalty for model size, KDR-PSO automates the discovery of parsimonious feature sets that maximize inter-class discrimination. The method is computationally efficient, naturally lending itself to multi-class classification and avoiding the prohibitive cost associated with classifier-in-the-loop wrappers. Experimental results on benchmark gene expression and image datasets show that KDR-PSO can achieve better dimensionality reduction compared to baselines and other algorithms, such as winning a better or at least similar performing models with decreased features. This approach is a strong and pragmatic technique to improve the model interpretability and generalizability for high-dimensional regions.**

***Keywords*—Dimensionality reduction; Particle Swarm Optimization; metaheuristics; K-Nearest Neighbors; class separability; high-dimensional data**

I. Introduction

The curse of dimensionality is a well-known problem in modern machine learning, especially in fields such as bioinformatics, computer vision, and text mining, where datasets often contain thousands of features with comparatively few samples. The main objective of the Feature selection (FS) is to select the most informative subset of features, which leads to models that are simpler, faster to train, less prone to overfitting, and more interpretable [1].

FS techniques are basically categorized into three types, which are filter, wrapper, and embedded methods [2]. Filter methods, such as correlation, mutual information, and variance-based selection, are computationally active but ignore feature interdependencies or classifier behavior [3]. Wrapper methods—such as recursive feature elimination (RFE) or evolutionary search guided by classifier accuracy— can accomplish higher predictive performance, but with high cost for training the repeated model, which becomes unreasonable in high-dimensional datasets [4,5,6]. Embedded techniques, such as Least Absolute Shrinkage and Selection Operator (LASSO), integrate feature selection into model training but depend basically on model choice [7].

In the last several years, metaheuristic optimization has become important in feature selection for its ability to examine large search spaces effectively [8, 9]. One of these metaheuristic optimizations is a PSO, a population-based metaheuristic, that has been successfully adapted for wrapper-based feature selection [10] and its binary variants (BPSO) that have been widely selected due to their simple execution and powerful global search abilities [7,11]. The requirement for classifier retraining via cross-validation for every fitness evaluation, which results in a high computational cost, is a significant drawback of conventional PSO-based wrappers [8].

New studies have tried to address this drawback by hybridization and adaptive mechanisms [1], but getting an optimal trade-off between search quality and computational efficiency remains challenging. To control this limitation, we suggest KDR-BPSO, a hybrid metaheuristic framework that maintains the exploration power of BPSO while deleting the classifier-in-the-loop dependency. The method introduces a KDR as a filter-based, model-free objective function that measures class separability directly inside a subspace of features. Specifically, KDR calculates the ratio of the mean distance between samples and their nearest neighbors of other classes to the mean distance within their own class—motivating subsets that maximize inter-class distance and minimize intra-class variability. By optimizing this ratio by a dimensionality penalty, KDR-BPSO automatically finds dense and discriminative feature subsets without repeated classifier training.

This approach extends across the gap between filter efficiency and wrapper accuracy, introducing a computationally scalable, multi-class-capable, and interpretable solution for high-dimensional feature selection. Empirical studies on gene expression and image datasets demonstrate that KDR-BPSO gains better and superior performance compared to traditional filter, wrapper, and embedded methods while basically reducing feature dimensionality and runtime [7, 11, 12]. The proposed framework gives a robust, efficient, and generalizable tool for dimensionality reduction in modern data-intensive applications. The KDR metric directly measures the inherent separability of classes in the selected feature space. By leading the PSO swarm to maximize KDR, the proposed method efficiently gets the optimal feature subsets, that are optimal for distance-based classifiers (like KNN itself) and useful for other classifiers, without the computational cost of inner model training. This lines up with some of the recent efforts in joining distance-based separability metrics with metaheuristic feature selection techniques to enhance performance on high-dimensional problems [1, 5, 10].

The proposed feature selection methodology combines Binary Particle Swarm Optimization with a novel K-nearest neighbor Distance Ratio fitness function. The framework is designed to select the optimal feature subsets that maximize the classification performance simultaneously, improve geometric separability, and keep feature parsimony. The proposed approach faces limitations of conventional filter and wrapper methods by combining both statistical and geometric criteria into a unified optimization framework.

The basic innovation reflected in the multi-objective fitness function that balances three critical sides of feature quality: predictive accuracy measured through cross-validation, class separability measured via distance ratios, and subset compactness enforced through regularization. This evaluation confirms that the selected features obtain high classification performance and show strong discriminative characteristics in the reduced feature space.

The rest of this study proceeds as follows: Section II reviews the related work. Section III describes the complete PSO-KDR framework, including an overview of the Particle Swarm Optimization (PSO) algorithm and the corresponding implementation procedure, followed by Section IV that presents a description of the datasets used and the experimental results and analysis. Finally, Section V concludes the study.

II. Related Work

Recent Dimensionality reduction and Metaheuristic-based feature selection have been widely studied over the past years. Particle Swarm Optimization (PSO) was one of the most frequently used search strategies. Modern systematic reviews emphasized both understanding of the field and its common limitations. A review contained more than a hundred studies on metaheuristic feature selection and showed that most studies depended on wrapper or hybrid filter–wrapper strategies connected to classifier performance, which can be computationally expensive, as introduced by Al-Shalif et al. [13]. By contrast, Akinola et al. [14] showed that filter-based metaheuristic methods, while more efficient, have been comparatively neglected, particularly in multiclass and high-dimensional scenarios. Bassi et al. [15] noted that existing filter-based approaches typically depend on traditional relevance measures such as mutual information or chi-square statistics rather than directly capturing class separability.

By PSO-driven feature selection, present methods mainly optimize inexplicit distance heuristics or classifier-oriented objectives. Shafipour et al. [16] presented a PSO-based feature selection method that used a particle distance ranking in a multi-objective framework to equate the classification performance and the subset size, but class separability in their approach is addressed indirectly through classification accuracy rather than through an explicit geometric measure. Particle Swarm Optimization (PSO) was used to improve distance metrics for KNN classification, where feature weights were modified to reduce classification errors introduced by Jurecek in [17]. Although effective, this strategy focuses on metric learning instead of selecting a compact feature subset and still relies on classifier evaluation. Recent work, such as Boolean Operator–based Particle Swarm Optimization (BOPSO) feature selection study for emotion analysis introduced by Sharma et al. [18], displayed improvement in the optimization performance, but it continued depending on classifier-based objectives, which constrained computational efficiency and scalability.

For all these studies, direct KNN-based inter-/intra-class distance ratio objectives with integrated feature-count penalties are extremely absent. The proposed KDR-PSO is located as a filter-based, PSO-driven dimensionality reduction method that immediately optimizes class separability through a K-Nearest Neighbors Distance Ratio (KDR). Different from wrapper and hybrid approaches, KDR-PSO rates the candidate subsets using only distance statistics derived from the data, preventing classifier-in-the-loop training. Additionally, it merges a direct subset-size penalty into a single scalar objective, supporting the most close-fisted feature sets without needing post hoc Pareto-front selection.

The proposed work handles a lack of PSO-driven methods that optimize a KNN-based distance–ratio separability measure and feature-set compactness via a purely filter-based framework, mainly for scalable multi-class and high-dimensional problems.

III. Proposed PSO with KDR for Feature Selection

The proposed framework integrates the PSO algorithm with KDR as a fitness function. The goal of the framework is to find the best feature subsets that improve separability and maximize classification performance. The fitness function proposed in this work balances three key aspects of good feature selection: keeping the feature set small, ensuring classes are well separated based on distance ratios, and achieving high prediction accuracy using cross-validation to achieve effective classification and distinguish between different classes.

Fig. 1 illustrates the overall workflow of the proposed KDR-PSO framework for feature selection. Starting from high-dimensional input data, the dataset is first preprocessed and then passed to a PSO mechanism that iteratively explores candidate feature subsets. Each particle, representing a binary feature selection vector, is evaluated using a hybrid fitness function that integrates the KDR, classification accuracy, and a penalty term to control subset size. Guided by personal and global best solutions, the swarm progressively refines the search until convergence, yielding a compact and highly discriminative feature subset that is subsequently used for final classification and performance evaluation.

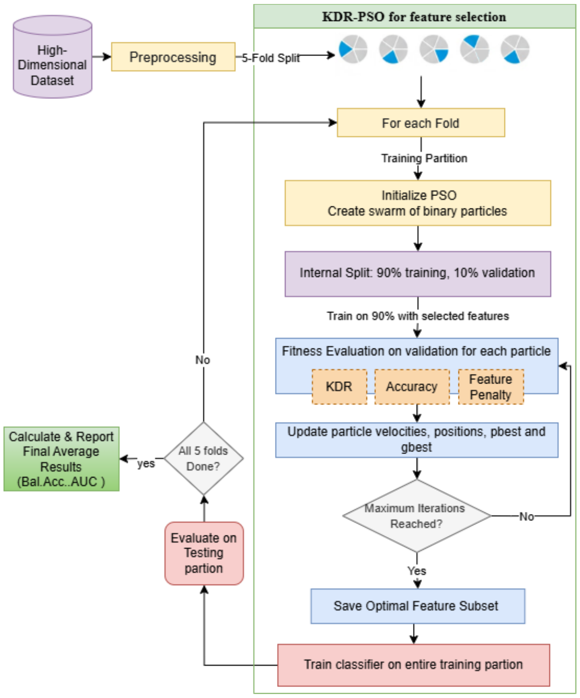

Fig. 1. Proposed KDR-PSO framework.

A. Particle Swarm Optimization

The PSO algorithm begins by giving each particle a random starting point and some initial velocity. As the process goes on, every particle learns from what it has tried before and from the results of nearby particles. Step by step, they move closer to a good solution. In PSO, each particle stands for one possible set of features in the search space, and its position is written as a binary vector $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, where D is the total number of features in the dataset, and $x_i \in \{0,1\}$ indicates whether the j^{th} feature is selected (1) or not selected (0). The velocity of particle i is denoted as $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, which determines the probability of changing the feature selection state.

A predefined maximum velocity (v_{max}) restricts the velocity of particles within a certain range $v_{id}^t \in [-v_{max}, v_{max}]$ to maintain stability. Each particle memorizes its best previous position as personal best (*pbest*), and the best position found by the entire swarm is referred to as global best (*gbest*). The velocity and position of each particle are updated according to the following Eq. (1) and Eq. (2) [19]:

$$v_{id}^{(t+1)} = w \times v_{id}^t + c_1 \times r_{1i} \times (p_{id} - x_{id}^t) + c_2 \times r_{2i} \times (p_{gd} - x_{id}^t) \quad (1)$$

$$x_{id}^{(t+1)} = \begin{cases} 1, & \text{if rand}() < \frac{1}{1+e^{-v_{id}^{(t+1)}}} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where, t is the iteration number, w is the inertia weight (decreasing linearly from 0.9 to 0.4), c_1 and c_2 are cognitive and social acceleration constants, respectively, r_{1i} and r_{2i} are random numbers uniformly distributed in [0, 1], p_{id} and p_{gd} represent the d^{th} elements of *pbest* and *gbest*, respectively.

In this study, each particle encodes a binary vector representing feature inclusion or exclusion. If the dataset contains *N* features, then the position vector length is *N*. Each bit in the vector takes a value of 1 (feature selected) or 0 (feature not selected). For instance, in a dataset with five features, a possible particle representation could be: x_i= [1, 0, 1, 0, 0]. This means that the first and third features are selected while the others are excluded. The PSO algorithm continuously updates these binary values to maximize the objective fitness function. Algorithm 1 shows the pseudo-code for PSO algorithm for feature selection.

Algorithm 1: Pseudo-code for PSO Algorithm for feature selection.

```
// Initialize swarm
For i = 1 to swarm_size:
    xi = random binary vector with ≥ 2 ones
    vi = random velocity vector
    pbesti = xi  // personal best
    pbest_fitnessi = -∞
End For
// Evaluate fitness based on equation 4 and update the best
For iter = 1 to max_iter:
    For each particle i:
        fitness = evaluate_fitness(xi)
        If fitness > pbest_fitnessi:
            pbesti = xi
            pbest_fitnessi = fitness
        End If
        If fitness > gbest_fitness
            gbest = xi
            gbest_fitness = fitness
        End If
    End For
    For each particle i:
        Update vi //Update velocity based on equation (1)
        Update xi //Update position using sigmoid equation (2)
// Ensure the minimum number of selected features
        While sum(xi) < 2:
            xi[random_index] = 1
        End While
    End For
End For
output gbest
```

B. Proposed PSO with KNN Distance Ratio (PSO_KDR)

Using classification accuracy as the sole objective function in feature selection often leads to overfitting and poor generalization, as it does not explicitly account for the structural separability of data classes. A feature subset may achieve high accuracy on the training data, but fail to maintain clear class boundaries in the feature space. To overcome this, the proposed method uses the K-Nearest Neighbor Distance Ratio as an additional discriminative criterion within PSO. By comparing the average distances between neighbors of the same class and neighbors of other classes, the KDR calculates the relative distance between classes. The algorithm looks for feature subsets that enhance class separability while achieving good prediction performance by combining accuracy and KDR in a hybrid fitness formulation.

The improved fitness design enables the PSO to find compact yet discriminative feature subsets, avoid local optima, and better explore the search space. The suggested method thus enhances the stability and robustness of the classification of certain features across several datasets. For each sample i, let:

$\bar{d}_{\text{same}}(i)$ is the average distance to K-Nearest Neighbors that belong to the same class.

$\bar{d}_{\text{diff}}(i)$ is the average distance to K-Nearest Neighbors from different classes.

The KDR is calculated as explained by Eq. (3):

$$KDR(S) = \frac{1}{n}\sum_{i=1}^{n} \frac{\bar{d}_{diff}(i)}{\bar{d}_{same}(i)+\epsilon} \quad (3)$$

where, n is the total number of samples and $\epsilon = 10^{-8}$ is a small constant to prevent division by zero. The overall fitness function is calculated as in Eq. (4):

$$F(S) = \beta \times \text{Accuracy}(S) + \alpha \times KDR_{norm}(S) - \lambda \times \frac{|S|}{d} (4)$$

where, *Accuracy(S)* is the classification accuracy evaluated using 5-fold KNN, $KDR_{\text{norm}}(S)$ is the normalized KDR value, $|S|$ is the number of selected features, $\beta = 0.95, \alpha = 0.05$ are weighting factors, and λ=0.001 controls the penalty for subset size.

IV. Experimental Results

A. Datasets and Experimental Setup

Experiments were conducted on ten available datasets to evaluate the proposed method, as summarized in Table I. These datasets are from different application domains, including biomedical, financial, environmental, and computer vision, to achieve an evaluation across different real-world applications.

TABLE I SUMMARY OF DATASETS USED

Dataset Name	Domain	Samples	Features	Classes	*Reference*
Climate_model	Climate	540	18	2	[20]
Credit	Finance	690	14	2	[21]
Heart	Medical	270	13	2	[22]
Ionosphere	Physics	351	34	2	[23]
Leukemia	Gene Expression	72	7,129	2	[24]
QSAR	Chemistry	1,055	41	2	[25]
Satellite	Remote Sensing	6,435	36	6	[26]
Segment	Computer Vision	2,310	19	7	[27]
Splice	Bioinformatics	3,190	62	3	[28]
Vehicle	Object Recognition	846	18	4	[29]

The datasets used in this study vary widely in both size and structure — from small medical datasets with only a few samples to large, high-dimensional gene expression datasets containing thousands of features. This diversity shows the effectiveness of the proposed feature selection method. All experiments were performed on a desktop computer equipped with an Intel Core i7 processor, 16 GB of RAM, and a 64-bit Windows operating system. The proposed KDR-PSO algorithm, along with all comparative methods, was implemented using the Python programming language.

The KNN with k=5 was employed, and performance assessed using 5-fold stratified cross-validation. The parameters of the PSO-KDR algorithm were determined through an iterative sensitivity analysis: initial parameter ranges were adopted from the literature, then testing of multiple values across preliminary runs to identify the optimal configuration. The finalized parameters, fixed for all experiments, are as follows:

- swarm size= 20
- maximum iterations = 40.
- w (inertia weight) from 0.9 to 0.4 decreased each iteration.
- c1=c2=2 (acceleration coefficients)
- α=0.05 and β=0.95 (The fitness function weighting factors)
- λ=0.001 (small penalty factor).

B. Results and Analysis

The Results performed using 5-fold cross-validation on ten publicly available datasets. The performance comparison includes Balanced Accuracy (Bal. Acc), F1-Score, Precision, and AUC (Area Under the Curve). Also, the number of selected features (Features). For each metric, the mean ± standard deviation is presented. The best results for each metric are highlighted in bold. The comparison includes four methods:

1) *Baseline*: KNN using all features without selection.

2) *Filter (IG)*: Features selected based on Information Gain ranking, independently of the classifier.

3) *PSO*: Particle Swarm Optimization for feature selection.

4) *PSO-KDR*: The proposed method combining PSO with the KDR-based fitness function.

As shown in Table II, the PSO-KDR outperforms other methods across most tested datasets in terms of classification accuracy and feature selection, where PSO-KDR using Leukemia dataset achieves Bal. Acc of 0.962 ± 0.006, overperforming PSO (0.920 ± 0.025) and the filter-based approach (0.882 ± 0.033), while maintaining a compact subset of features. In addition, the Heart and Credit datasets, PSO-KDR demonstrates superior F1-scores and AUC values.

In terms of feature selection, PSO-KDR identifies smaller feature subsets compared to the baseline, where in the Climate_model dataset, it reduces the feature count from 18 to around 7 while improving the AUC from 0.797 to 0.881. PSO-KDR maintains high performance stability across all datasets, as indicated by its relatively low standard deviations.

A t-test conducted between PSO-KDR and the PSO across all datasets, the results show a statistically significant improvement in Bal. Accuracy with $p < 0.05$ for 8 out of 10 datasets. The results show that PSO-KDR achieves trade-off between feature selection and classification performance, indicating its adaptability to diverse datasets.

Fig. 2 illustrates the convergence behavior of PSO-KDR compared to PSO method across three datasets (Leukemia, Heart, and Climate_model).

TABLE II COMPREHENSIVE PERFORMANCE COMPARISON OF FEATURE SELECTION METHODS

Dataset	*Method*	*Bal. Accuracy*	*F1-Score*	*Precision*	*AUC*	*#Features*
Climate_model	Baseline	0.538 ± 0.011	0.545 ± 0.018	0.700 ± 0.091	0.797 ± 0.018	18 ± 0
	Filter (IG)	0.635 ± 0.060	0.667 ± 0.067	0.778 ± 0.059	0.835 ± 0.046	5 ± 2
	PSO	0.680 ± 0.031	0.733 ± 0.030	0.902 ± 0.046	0.869 ± 0.031	8 ± 1
	PSO-KDR	**0.705 ± 0.017**	**0.767 ± 0.016**	**0.944 ± 0.023**	**0.881 ± 0.021**	7 ± 1
Credit	Baseline	0.672 ± 0.012	0.673 ± 0.012	0.685 ± 0.013	0.728 ± 0.012	14 ± 0
	Filter (IG)	0.777 ± 0.059	0.775 ± 0.058	0.784 ± 0.059	0.845 ± 0.064	4 ± 2
	PSO	0.858 ± 0.010	0.857 ± 0.010	0.859 ± 0.008	0.891 ± 0.009	4 ± 0
	PSO-KDR	**0.864 ± 0.003**	**0.863 ± 0.003**	**0.865 ± 0.004**	**0.892 ± 0.011**	4 ± 1
Heart	Baseline	0.646 ± 0.020	0.646 ± 0.021	0.650 ± 0.020	0.691 ± 0.021	13 ± 0
	Filter (IG)	0.765 ± 0.051	0.765 ± 0.054	0.779 ± 0.048	0.833 ± 0.047	3 ± 1
	PSO	0.823 ± 0.013	0.825 ± 0.013	0.834 ± 0.011	0.865 ± 0.017	5 ± 1
	PSO-KDR	**0.842 ± 0.008**	**0.844 ± 0.007**	**0.849 ± 0.007**	**0.875 ± 0.016**	5 ± 2
Ionosphere	Baseline	0.774 ± 0.011	0.794 ± 0.011	0.869 ± 0.005	0.910 ± 0.006	34 ± 0
	Filter (IG)	0.828 ± 0.025	0.847 ± 0.023	0.896 ± 0.012	0.928 ± 0.010	9 ± 4
	PSO	**0.854 ± 0.012**	**0.872 ± 0.010**	**0.911 ± 0.005**	**0.928 ± 0.007**	10 ± 3
	PSO-KDR	0.852 ± 0.015	0.869 ± 0.014	0.908 ± 0.008	0.926 ± 0.010	9 ± 2
Leukemia	Baseline	0.840 ± 0.025	0.859 ± 0.026	0.927 ± 0.010	0.976 ± 0.017	2938 ± 0
	Filter (IG)	0.882 ± 0.033	0.899 ± 0.031	0.941 ± 0.018	0.982 ± 0.017	852 ± 350
	PSO	0.920 ± 0.025	0.934 ± 0.022	0.960 ± 0.014	0.985 ± 0.013	1431 ± 20
	PSO-KDR	**0.962 ± 0.006**	**0.969 ± 0.005**	**0.981 ± 0.003**	**0.993 ± 0.009**	1455 ± 33
QSAR	Baseline	0.797 ± 0.008	0.791 ± 0.008	0.787 ± 0.008	0.865 ± 0.007	41 ± 0
	Filter (IG)	0.783 ± 0.020	0.782 ± 0.020	0.784 ± 0.020	0.855 ± 0.018	11 ± 5
	PSO	0.836 ± 0.011	0.835 ± 0.009	0.836 ± 0.008	0.899 ± 0.008	18 ± 3
	PSO-KDR	**0.846 ± 0.007**	**0.845 ± 0.005**	**0.845 ± 0.006**	**0.900 ± 0.006**	20 ± 3
Satellite	Baseline	0.886 ± 0.002	0.888 ± 0.002	0.890 ± 0.002	0.978 ± 0.001	36 ± 0
	Filter (IG)	0.855 ± 0.020	0.859 ± 0.018	0.865 ± 0.016	0.963 ± 0.008	10 ± 4
	PSO	0.887 ± 0.003	0.890 ± 0.003	0.894 ± 0.003	0.977 ± 0.001	22 ± 2
	PSO-KDR	**0.889 ± 0.002**	**0.893 ± 0.002**	**0.897 ± 0.002**	**0.977 ± 0.002**	20 ± 2
Segment	Baseline	0.933 ± 0.004	0.932 ± 0.004	0.933 ± 0.004	0.992 ± 0.001	19 ± 0
	Filter (IG)	0.870 ± 0.032	0.869 ± 0.033	0.871 ± 0.032	0.972 ± 0.009	5 ± 2
	PSO	0.962 ± 0.002	0.961 ± 0.002	0.962 ± 0.002	0.994 ± 0.001	8 ± 1
	PSO-KDR	**0.962 ± 0.002**	**0.962 ± 0.002**	**0.962 ± 0.002**	**0.993 ± 0.001**	7 ± 1
Splice	Baseline	0.701 ± 0.007	0.679 ± 0.008	0.753 ± 0.009	0.807 ± 0.006	60 ± 0
	Filter (IG)	0.795 ± 0.022	0.787 ± 0.025	0.819 ± 0.015	0.882 ± 0.019	17 ± 7
	PSO	0.797 ± 0.004	0.790 ± 0.004	0.817 ± 0.010	0.876 ± 0.013	24 ± 3
	PSO-KDR	**0.801 ± 0.009**	**0.794 ± 0.010**	**0.823 ± 0.011**	**0.882 ± 0.013**	24 ± 4
Vehicle	Baseline	0.643 ± 0.012	0.633 ± 0.011	0.629 ± 0.012	0.857 ± 0.008	18 ± 0
	Filter (IG)	0.555 ± 0.044	0.550 ± 0.042	0.549 ± 0.040	0.795 ± 0.031	5 ± 2
	PSO	0.717 ± 0.011	0.711 ± 0.010	0.708 ± 0.010	0.898 ± 0.008	10 ± 1
	PSO-KDR	**0.731 ± 0.009**	**0.725 ± 0.008**	**0.724 ± 0.008**	**0.902 ± 0.005**	10 ± 1

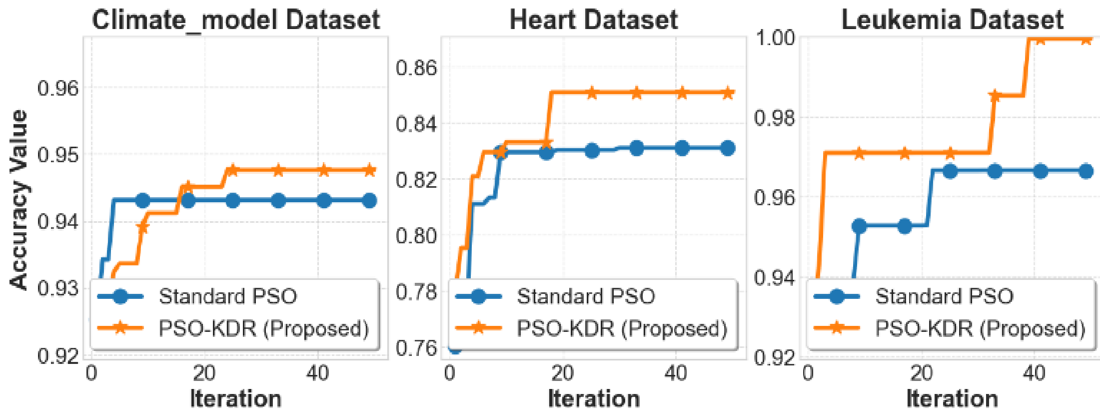

Fig. 2. Convergence curves of standard PSO vs. PSO-KDR.

The convergence curves show that PSO-KDR achieves faster and smoother fitness improvement over iterations that due to the guidance provided by the KDR that effectively balances global exploration and local refinement during the optimization process.

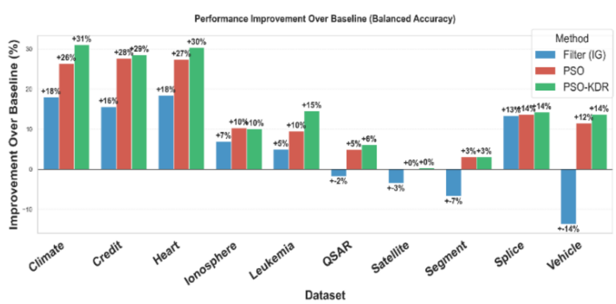

Fig. 3. Comparison of improvement over baseline.

As illustrated in Fig. 3, the comprehensive performance analysis across ten datasets shows the efficacy of various feature selection methods. Across the most tested datasets, all three evaluated feature selection techniques showed positive performance gains, with the most pronounced improvements observed in the Climate Model, Credit, and Heart datasets. The maximum performance enhancement reached +31.0% by PSO-KDR on the Climate_Model dataset.

PSO and its enhanced variant, PSO-KDR, show consistent superiority compared to conventional filter methods that achieved the maximum performance improvement across four datasets (Climate_model, Heart, Leukemia, and Vehicle), and the IG filter method exhibited inconsistent behavior, with performance degradation observed in certain datasets, such as Vehicle and Segment.

Fig. 4 shows the analysis of selected feature subset sizes across ten datasets. To handle high-dimensional datasets, the logarithmic scale is used. The minimal subsets across most of the datasets were selected by using the IG filter method, but its independence from classifier behavior sometimes caused performance unreliability, as shown in datasets such as Segment and Vehicle, which feature elimination causes less accuracy. The PSO and PSO-KDR wrapper methods gave more balanced feature selection behavior, keeping slightly larger but more discriminative subsets to maximize classification accuracy. This drift is usually evident in QSAR and Satellite, where keeping a moderate number of features gave higher results. The average computation time of the proposed KDR-PSO method ranged from approximately 3 to 8 seconds for low- and medium-dimensional datasets and from 2 to 6 minutes for high-dimensional gene expression datasets, depending on the number of features and optimization iterations.

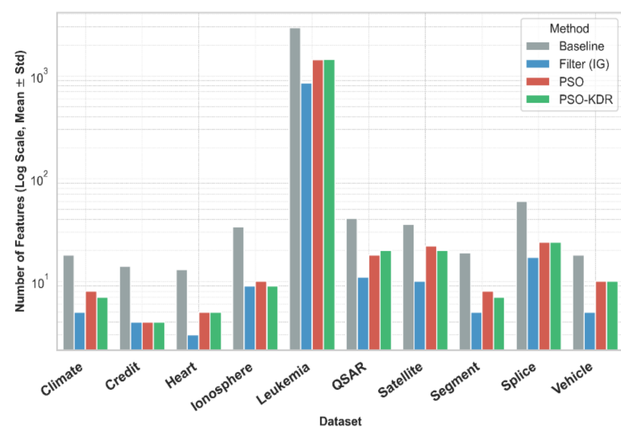

Fig. 4. Feature reduction comparison (Log scale).

PSO-KDR outperforms baseline, filter, and standard PSO methods by selecting compact, relevant feature subsets while improving classification performance. This shows that the KDR-based fitness function effectively balances feature relevance and predictive accuracy, especially on high-dimensional datasets. However, although effective, the proposed KDR-PSO method has some limitations. As a population-based optimization approach, its computational cost increases with data dimensionality and the number of PSO iterations.

V. Conclusion

This study proposed KDR-PSO, a new hybrid feature selection method that merges the global search capability of Binary Particle Swarm Optimization (BPSO) with a filter-based KNN Distance Ratio (KDR) objective function. The obtained results showed reaching the optimal balance between computational efficiency and feature selection performance in the proposed algorithm KDR-PSO, that keeping high accuracy and clarity across various datasets.

In terms of classification accuracy and feature reduction, the proposed KDR-PSO method showed excellent performance, as shown by the values of the evaluation metrics F1-score, AUC, and precision. KDR-PSO joined filter-based techniques with a wrapper framework so it can handle strongly the computational limitations related to the wrapper techniques and make it practical for high-dimensional data. Furthermore, the proposed KDR-PSO got significant feature proximity while maintaining or improving the classification performance and selecting the minimum number of features. The logarithmic-scale comparisons of feature counts over datasets revealed and emphasized the wide variation in feature dimensions. The basic contribution for this work is building KNN Distance Ratio multi-Objective, that works as an effective mechanism for PSO search phases, accelerating in superior and low-cost computations for effective feature subset selection.

This study introduced successful implementation of KDR-PSO, that will be strong basis for a lot of future work. The expected research can include studying alternative distance metrics, expanding adaptive penalty terms, and executing the method to emerging domains that need efficient feature selection for high-dimensional data. This work introduces valuable advancement in terms of the feature selection literature, covering the gap between the strengths of filter and wrapper methods.

References

[1] M. Alzaqebah and E. a. E. Ahmed, “Accelerated fuzzy min–max neural network and arithmetic optimization algorithm for optimizing hyper-boxes and feature selection,” Neural Computing and Applications, vol. 36, no. 4, pp. 1553–1568, Nov. 2023.

[2] C. Kuzudisli, B. Bakir-Gungor, N. Bulut, B. Qaqish, and M. Yousef, “Review of feature selection approaches based on grouping of features,” PeerJ, vol. 11, p. e15666, Jul. 2023.

[3] D. Patel and A. K. Saxena, “Feature Selection in High Dimension Datasets using Incremental Feature Clustering,” Indian Journal of Science and Technology, vol. 17, no. 32, pp. 3318–3326, Aug. 2024.

[4] M. Alzaqebah, N. Alrefai, E. a. E. Ahmed, S. Jawarneh, and M. K. Alsmadi, “Neighborhood search methods with Moth Optimization algorithm as a wrapper method for feature selection problems,” International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering, vol. 10, no. 4, p. 3672, Mar. 2020.

[5] M. Alzaqebah et al., “Hybrid feature selection method based on particle swarm optimization and adaptive local search method,” International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering, vol. 11, no. 3, p. 2414, Feb. 2021.

[6] F. Wang et al., “Navigating the microarray landscape: a comprehensive review of feature selection techniques and their applications,” Frontiers in Big Data, vol. 8, p. 1624507, Jul. 2025.

[7] L. Qu, W. He, J. Li, H. Zhang, C. Yang, and B. Xie, “Explicit and size-adaptive PSO-based feature selection for classification,” Swarm and Evolutionary Computation, vol. 77, p. 101249, Jan. 2023.

[8] A.-D. Li, B. Xue, and M. Zhang, “Multi-objective particle swarm optimization for key quality feature selection in complex manufacturing processes,” Information Sciences, vol. 641, p. 119062, May 2023.

[9] F. Ayeche and A. Alti, “Efficient feature selection in high dimensional data based on enhanced binary chimp optimization algorithms and machine learning,” Human-Centric Intelligent Systems, vol. 3, no. 4, pp. 558–587, Nov. 2023.

[10] M. K. Alsmadi et al., “An Enhanced particle swarm optimization for ITC2021 sports Timetabling,” Computers, Materials & Continua/Computers, Materials & Continua (Print), vol. 72, no. 1, pp. 1995–2014, Jan. 2022.

[11] K. Chen, W. Wang, F. Zhang, J. Liang, and K. Yu, “Correlation-Guided particle swarm optimization approach for feature selection in fault diagnosis,” IEEE/CAA Journal of Automatica Sinica, vol. 12, no. 11, pp. 2329–2341, May 2025.

[12] B. Ahadzadeh, M. Abdar, F. Safara, A. Khosravi, M. B. Menhaj, and P. N. Suganthan, “SFE: A simple, fast and efficient feature selection Algorithm for High-Dimensional Data,” arXiv (Cornell University), Mar. 2023.

[13] S. A. Al-Shalif et al., “A systematic literature review on meta-heuristic based feature selection techniques for text classification,” PeerJ Computer Science, vol. 10, p. e2084, Jun. 2024.

[14] O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, and L. Abualigah, “Multiclass feature selection with metaheuristic optimization algorithms: a review,” Neural Computing and Applications, vol. 34, no. 22, pp. 19751–19790, Aug. 2022.

[15] D. Bassi and H. Singh, “A comparative analysis of metaheuristic feature selection methods in software vulnerability prediction,” e-Informatica Software Engineering Journal, vol. 19, no. 1, p. 250103, Nov. 2024.

[16] M. Shafipour, A. Rashno, and S. Fadaei, “Particle distance rank feature selection by particle swarm optimization,” Expert Systems With Applications, vol. 185, p. 115620, Jul. 2021.

[17] M. Jureček and R. Lórencz, "Distance Metric Learning using Particle Swarm Optimization to Improve Static Malware Detection," in Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020), Prague, Czech Republic, 2020, pp. 725-732.

[18] H. D. Sharma et al., “Sentiment classification via improved feature selection using Boolean operator-based particle swarm optimization,” Scientific Reports, vol. 15, no. 1, p. 38923, Nov. 2025.

[19] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. ICNN'95-Int. Conf. Neural Networks, Nov. 1995, pp. 1942–1948.

[20] D. D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, and Y. Zhang, “Failure analysis of parameter-induced simulation crashes in climate models,” Geosci. Model Dev., vol. 6, no. 4, pp. 1157–1171, Jul. 2013.

[21] R. Quinlan, “StatLog (Australian Credit Approval),” UC Irvine. Jan. 01, 1987.

[22] A. J. W. Steinbrunn, “Heart disease,” UC Irvine. Jan. 01, 1989.

[23] V. S. S. Wing, “Ionosphere,” UC Irvine. Jan. 01, 1989.

[24] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ... & Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 286(5439), 531-537.

[25] K. M. T. Ringsted, “QSAR biodegradation,” UC Irvine. Jan. 01, 2013.

[26] A. Srinivasan, “Statlog (Landsat Satellite),” UC Irvine. Jan. 01, 1993.

[27] “Image segmentation,” UC Irvine. Jan. 01, 1990.

[28] UC Irvine, "Molecular Biology (Splice-junction Gene Sequences)," Jan. 01, 1991.

[29] P. Mowforth and B. Shepherd, "Statlog (Vehicle Silhouettes) [Dataset]," UCI Machine Learning Repository, 1990s (approx).