International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-13, Issue-3, May 2025
DOI: https://doi.org/10.55524/ijircst.2025.13.3.26
Article ID IRP-1651, Pages 188-196
www.ijircst.org

Autoencoder-Based Adaptive Multi-Objective Particle Swarm Optimization for Gene Selection

Sumet Mehta[1], Fei Han[2], Muhammad Sohail[3], Arfan Nagra[4], and Qinghua Ling[5]

[1] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, Jiangsu, China
[2] R&D, Star Engineers India Private Limited, Pune, 411019, Maharashtra, India
[3] Department of Computer Software Engineering, Military College of Signals, NUST, Islamabad 46000, Pakistan
[4] School of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[5] School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

Correspondence should be addressed to Sumet Mehta; 1000006362@ujs.edu.cn

Received 29 April 2025; Revised 14 May 2025; Accepted 29 May 2025

ABSTRACT- In gene expression analysis, selecting informative genes is essential for uncovering biological mechanisms and identifying potential biomarkers. However, conventional gene selection methods often struggle with scalability and parameter tuning, limiting their effectiveness in large-scale datasets and algorithmic optimization. To overcome these challenges, we propose Autoencoder-based Adaptive Multi-Objective Particle Swarm Optimization for Gene Selection (AAMOPSO). Our approach incorporates an autoencoder-based preprocessing step to enhance scalability by learning a compressed representation of gene expression data, reducing dimensionality while retaining critical features. Additionally, we introduce an Adaptive Parameter Tuning mechanism within the Multi-Objective Particle Swarm Optimization (MOPSO) framework, dynamically adjusting algorithm parameters based on real-time performance metrics. Extensive experiments on four benchmark microarray datasets demonstrate that AAMOPSO consistently outperforms existing state-of-the-art methods in classification accuracy and the compactness of selected gene subsets.

KEYWORDS- Microarray gene selection, multi-objective optimization, particle swarm optimization, autoencoder.

I. INTRODUCTION

High-throughput genomic technologies, such as microarrays and RNA sequencing, have revolutionized biomedical research by enabling the simultaneous measurement of gene expression levels across tens of thousands of genes [1]. These technologies generate vast amounts of high-dimensional data, offering unprecedented opportunities to understand the molecular mechanisms underlying complex diseases and identify diagnostic and prognostic biomarkers [2]. However, analyzing such high-dimensional data is non-trivial. One of the most critical steps in this process is gene selection, that is to identify a subset of informative genes most relevant to a specific phenotype, such as disease presence, progression, or treatment response [3]. Gene selection enhances predictive model accuracy, improves interpretability, reduces computational overhead, mitigates the curse of dimensionality, and facilitates biologically meaningful insights [4]. Nevertheless, gene selection remains challenging due to the high feature-to-sample ratio in genomic datasets, increasing the risk of overfitting and rendering many traditional machine learning methods ineffective [5].

Traditional gene selection approaches fall into three categories: filter, wrapper, and embedded methods. Filter methods, such as mutual information and correlation-based techniques, are computationally efficient but often ignore feature dependencies and classifier-specific performance [6]. Wrapper methods evaluate gene subsets based on classifier performance, yielding better results at the cost of high computational complexity [7]. Embedded methods, like LASSO and decision tree-based feature importance, integrate selection within model training but still struggle with ultra-high dimensionality [8]. As dimensionality increases, even advanced optimization-based methods face significant challenges. Among metaheuristic techniques, Multi-Objective Particle Swarm Optimization (MOPSO) has gained attention for its ability to optimize conflicting objectives, such as classification accuracy and the number of selected genes by identifying Pareto-optimal solutions (POS) [9].

MOPSO, inspired by swarm intelligence, explores the feature space via particles guided by their personal best positions and the swarm's best solutions [10][11]. This mechanism naturally suits multi-objective optimization but critically depends on parameter settings: inertia weight, cognitive and social coefficients, and swarm size [12]. Poor parameter choices can lead to premature convergence, stagnation, and failure to find meaningful solutions [13]. Worse, optimal parameters vary across datasets, requiring manual tuning; a time-consuming and suboptimal process that limits MOPSO's practical usability in gene selection [14].

Another major bottleneck is scalability. As gene expression data dimensionality grows, the search space expands exponentially, making efficient exploration difficult [15]. Parallel or distributed MOPSO implementations improve runtime but do not fundamentally resolve high-dimensional

search challenges [16]. Moreover, these approaches still rely on fixed parameters, inheriting the same limitations.

To address these challenges, we propose Autoencoder-based Adaptive Multi-Objective Particle Swarm Optimization (AAMOPSO), integrating unsupervised deep learning for dimensionality reduction and adaptive MOPSO parameter tuning. First, an autoencoder compresses gene expression data into a lower-dimensional latent space, preserving essential features while reducing noise [17]. Unlike filter methods, AAMOPSO conducts selection on the original gene set, ensuring biological interpretability. Second, a dynamic parameter adjustment mechanism adapts MOPSO's inertia weight, learning coefficients, and swarm size based on real-time metrics like convergence rate, swarm diversity, and Pareto front spread [18]. This eliminates manual tuning and enhances robustness across datasets.

By combining these innovations, AAMOPSO overcomes key limitations of existing methods: (1) standard MOPSO struggles with high dimensionality and static parameters, (2) parallel MOPSO improves speed but not search efficiency, and (3) autoencoder-based methods often fail to link feature transformation with selection. AAMOPSO bridges this gap, enabling scalable, interpretable, and high-performing gene selection for precision medicine.

The paper is structured as follows: Section 2 reviews related work, Section 3 details AAMOPSO's methodology, Section 4 describes experiments and results, and Section 5 concludes with future directions.

II. RELATED WORK

Recent advances in multi-objective particle swarm optimization (MOPSO) have demonstrated significant potential for gene selection in high-dimensional biomedical datasets. Several studies have explored enhanced MOPSO variants to address the challenges of feature selection, classification accuracy, and biological interpretability.

For instance, Azadifar and Ahmadi proposed a graph-based many-objective PSO algorithm for medical diagnosis, incorporating gene interaction networks to improve selection robustness [19]. Their approach leveraged topological properties of biological networks but faced scalability limitations with ultra-high-dimensional data. Similarly, Rostami et al. integrated MOPSO with node centrality measures, enhancing feature relevance by considering both statistical significance and biological network importance [20]. While effective, their method relied heavily on prior biological knowledge, which may not always be available.

More recent works have focused on adaptive and neighborhood-preserving strategies. Mehta et al. introduced an adaptive neighborhood-preserving MOPSO (ANPMOPSO) that maintains local feature structures while optimizing classification performance [14]. Their method improved stability in gene selection but required extensive parameter tuning. In another study, Mehta et al. presented MORPSO_ECD+ELM, a unified framework combining MOPSO with an extreme learning machine for simultaneous gene selection and cancer classification [13]. This approach achieved competitive accuracy but struggled with interpretability due to the black-box nature of ELMs.

Despite these advancements, critical gaps remain: (1) most MOPSO-based methods rely on static parameter configurations, limiting adaptability across diverse datasets; (2) high-dimensional gene expression data still pose computational challenges; and (3) few approaches effectively balance feature reduction with biological interpretability. Our proposed AAMOPSO gene selection method addresses these limitations by integrating unsupervised deep learning for dimensionality reduction and dynamic parameter adaptation, enabling more efficient and robust gene selection without sacrificing biological relevance. This highlights the need for an adaptive, scalable, and interpretable MOPSO framework, positioning AAMOPSO as a novel solution that bridges deep learning and evolutionary optimization for improved biomarker discovery.

III. METHODOLOGY

The proposed AAMOPSO method is designed to enhance gene selection by addressing two major challenges: high dimensionality and parameter sensitivity. Initially, a deep autoencoder is employed to learn a compressed and noise-reduced representation of the gene expression data, effectively reducing dimensionality while retaining critical biological features. This compressed data is then used within a MOPSO framework to optimize both classification accuracy and the number of selected genes. To improve convergence and solution quality, an adaptive parameter tuning strategy is integrated into the MOPSO process, allowing dynamic adjustment of inertia weight and acceleration coefficients based on the swarm's performance. This combined strategy ensures more efficient exploration and exploitation of the search space, leading to the identification of compact and highly informative gene subsets.

A. Autoencoder Preprocessing

An autoencoder is a type of artificial neural network used for unsupervised learning of efficient codings. It consists of an encoder network that maps the input data to a lower-dimensional representation (encoding), and a decoder network that reconstructs the original input from the encoded representation. The autoencoder learns to encode the input data into a lower-dimensional latent space and then decode it back to the original input. This process enables the autoencoder to capture the most important features or patterns in the data while reducing its dimensionality. In our AAMOPSO for gene selection, the autoencoder can be trained using raw gene expression data. Once trained, it can be used to preprocess the gene expression data by encoding it into a lower-dimensional representation.

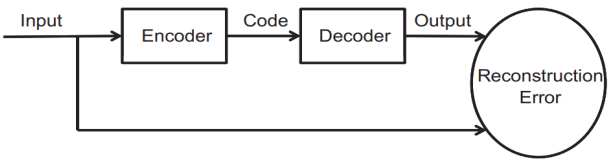

Figure 1: The Visualization Description of and Autoencoder

The pre-processing step involves passing the raw gene expression data through the trained autoencoder to obtain the encoded representation, which contains the essential features of the input data in a more compact form.

Mathematically, the encoding process of an autoencoder can be represented as follows:

$$Encoded_{data} = \sigma(W_{encoder} \cdot Raw_{data} + b_{encoder}) \quad (1)$$

where Raw_{data} is the raw gene expression data matrix, $W_{encoder}$ and $b_{encoder}$ are the weights and biases of the encoder network respectively. σ is the activation function (e.g., sigmoid, ReLU) applied elementwise to the output. The resulting encoded data matrix represents a lower-dimensional representation of the original gene expression data, capturing the most salient features or patterns in the data. This lower-dimensional representation obtained through autoencoder preprocessing used as input to the AAMOPSO algorithm for gene selection, enabling more efficient exploration of the solution space and addressing scalability issues.

Algorithm 1: Autoencoder Preprocessing for Gene Expression Data

Input:
- Raw_{data}, $hidden_{\text{units}}$, $activation_{func}$, $learning_{rate}$, $epochs$

Output:
- $Encoded_{data}$

Initialization:
1. Initialize autoencoder neural network with one hidden layer.
   - Set input size to number of genes in Raw_{data}.
   - Set hidden layer size to $hidden_{\text{units}}$.
   - Initialize weights and biases randomly.
   - Choose activation function ($activation_{func}$).
   - Set $learning_{rate}$ and $epochs$ for training.

Training:
2. Train autoencoder:
   for epoch = 1 to epochs **do**:
      for each sample in Raw_{data} **do**:
         Forward pass:
         - Compute $encoded_{output}$ using encoder network.
         Backpropagation:
         - Update weights and biases using reconstruction error gradient.

Encoding:
3. Encode gene expression data:
   for each sample in Raw_{data} **do**:
      - Compute $Encoded_{data}$ using trained encoder network.

Output:
4. Return $Encoded_{data}$

End Algorithm

B. Initialization

In the initialization step of the AAMOPSO for gene selection, the MOPSO algorithm is initialized with various parameters essential for the optimization process. These parameters include the swarm size (N), which determines the number of particles or candidate solutions in the population, and the maximum number of generations (G_{max}), defining the termination criterion for the optimization process. Additionally, parameters such as the inertia weight (w), cognitive learning factor (c_1), and social learning factor (c_2), are set to control the particles' movement and exploration-exploitation balance.

Furthermore, the positions and velocities of each particle within the search space are randomly initialized. The position of a particle represents a potential solution or gene subset, while the velocity influences the particle's movement towards promising regions of the search space. Random initialization ensures diversity in the initial population, allowing for exploration across different regions of the solution space. This diversity is crucial for the optimization process to avoid premature convergence to suboptimal solutions and facilitate the discovery of a diverse set of Pareto-optimal solutions during the optimization process. Overall, this initialization step sets the stage for the subsequent optimization process in AAMOPSO, providing the foundation for effective gene selection.

C. Fitness Evaluation

In gene selection tasks, objective functions play a crucial role in quantifying the quality of selected gene subsets. In the proposed AAMOPSO for gene selection, two objective functions are defined to guide the optimization process. Firstly, $(f_1(Acc))$, representing the classification accuracy of the gene subset using a classifier such as Support Vector Machine or Random Forest, serves as a measure of the predictive performance of the selected genes. This objective function aims to maximize the accuracy of the classification model built upon the selected genes, ensuring that the chosen subset contributes effectively to the predictive power of the model. Secondly, $(f_2(NS))$, representing the number of genes in the subset, serves as a minimization objective. The goal of this objective function is to minimize the number of genes selected while maintaining high classification accuracy. By minimizing the gene subset size, the algorithm aims to identify a compact and informative set of genes, facilitating biological interpretation and reducing computational complexity.

To assign fitness values to each particle we combine these two objective functions, AAMOPSO seeks to strike a balance between maximizing classification accuracy and minimizing the number of selected genes, ultimately leading to the discovery of robust and parsimonious gene subsets with high predictive performance. Therefore, two objective functions are:

$$Fitness = \{(f_1(Acc)), -(f_2(NS))\} \quad (2)$$

Here, $(f_2(NS))$ is multiplied by -1 to obtain the fitness value for the second objective. This negative multiplication is done to minimize the gene subset size, as smaller gene subsets are generally preferred to avoid overfitting and improve computational efficiency. Therefore, the fitness evaluation is crucial for guiding the optimization process towards identifying gene subsets that balance classification accuracy and size effectively.

D. Adaptive Parameter Initialization

In the Adaptive Parameter Initialization step of the AAMOPSO for gene section method, adaptive control mechanisms are established to dynamically adjust the algorithm's parameters based on the observed performance metrics. This adaptation aims to optimize the convergence rate (CR), diversity (D), and spread of the Pareto front (SP) throughout the optimization process. Initially, the adaptive parameters are set to their respective initial values: CR_{initial}, D_{initial}, and SP_{initial}. To set these initial values, we need to calculate them using appropriate metrics. Here's how each parameter can be calculated:

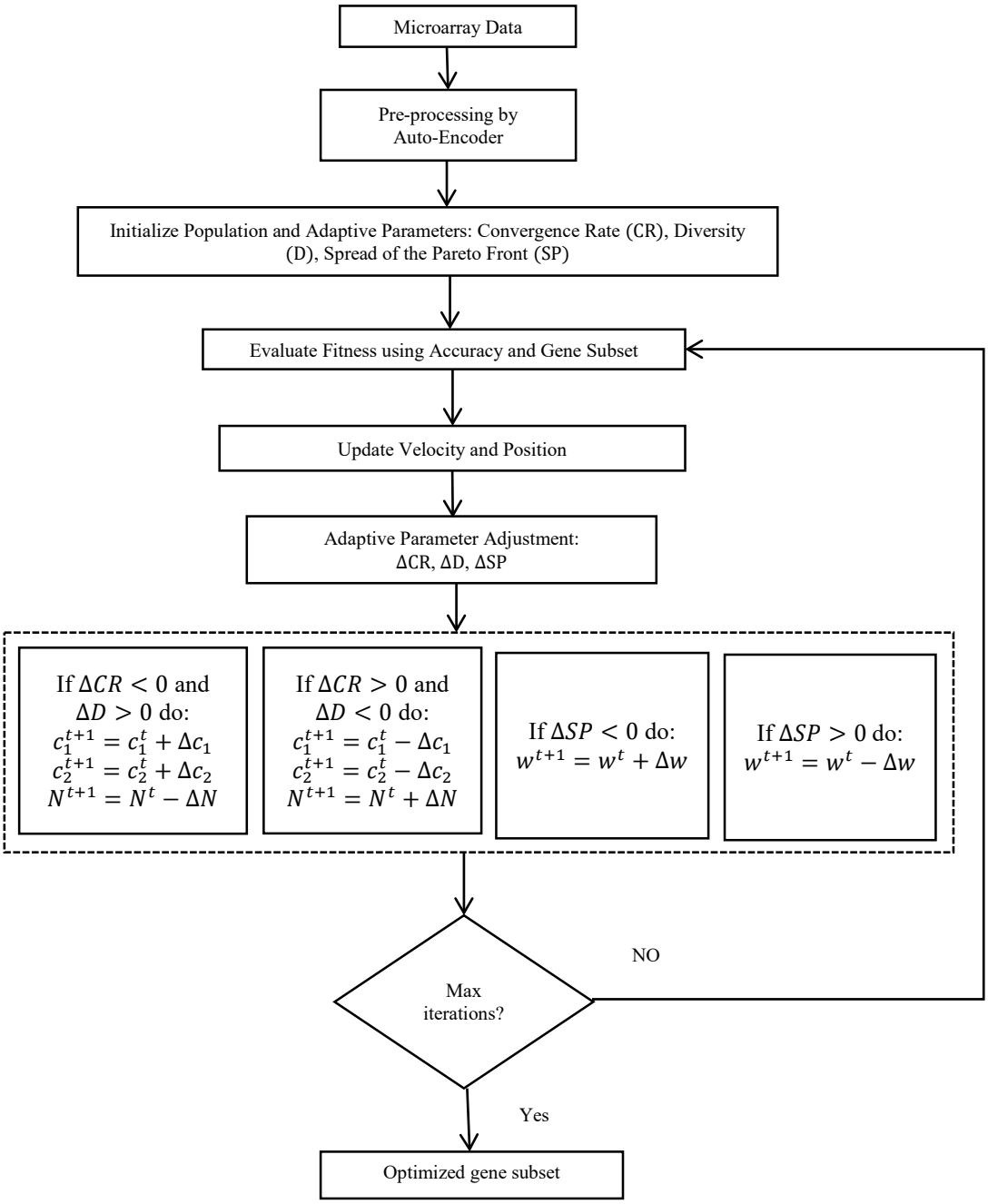

Figure 2: The Flowchart of the Proposed AAMOPSO Gene Selection Method

- **Convergence Rate (CR):** CR_{initial} represents the convergence rate at the beginning of the optimization process. Convergence rate measures how quickly the algorithm converges towards the Pareto front, indicating its efficiency in finding optimal or near-optimal solutions. In our AAMOPSO method, we calculate the convergence rate by computing the change in the hypervolume (HV) of the Pareto front over successive generations:

$$CR = \frac{HV_{t-1} - HV_t}{HV_{t-1}} \quad (3)$$

where HV_{t-1} and HV_t represent the hypervolume of the Pareto front at generations $t-1$ and t respectively. Therefore, the initial hypervolume can be computed based on the initial population of solutions.

- **Diversity (D)**: D_{initial} denotes the diversity of solutions in the initial population. Diversity measures the spread or variety of solutions in the population, indicating how well the algorithm explores the search space. In our AAMOPSO method, we calculate the diversity is by computing the average pairwise distance between solutions in the population:

$$D = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} |S_i - S_j| \quad (4)$$

where N is the population size, and $|S_i, S_j|$ is the Euclidean distance between solutions S_i and S_j.

- **Spread of Pareto Front (SP):** SP_{initial} indicates the spread or uniformity of solutions along the Pareto front initially. The spread of the Pareto front measures how evenly solutions are distributed along the front, indicating the extent of coverage of the objective space. In our AAMOPSO method, we calculate the spread by

computing the average Euclidean distance between adjacent solutions along the Pareto front:

$$SP = \frac{1}{M-1}\sum_{i=1}^{M-1}|P_i - P_{i+1}| \quad (5)$$

where M is the number of solutions on the Pareto front, and P_i and P_{i+1} are adjacent solutions on the front. Once these initial values are computed, they serve as the starting points for the adaptive parameter adjustment process. Throughout the optimization loop, the algorithm dynamically adjusts the parameters based on changes in $CR, D, and\ SP$ to ensure efficient exploration and convergence towards high-quality solutions in the gene selection task.

E. Optimization Loop

In the Optimization Loop of the AAMOPSO for Gene Selection method, the algorithm iterates through multiple generations $t = (1,2,\dots,G_{max})$. For each particle i, the velocity and position are updated using the MOPSO equations. The velocity update equation is given by:

$$vel_{i,j}^{k+1} = wv_{i,j}^{k} + r_1c_1\left(Pb_{i,j} - pos_{i,j}^{k}\right) + r_2c_2\left(Gb_j - pos_{i,j}^{k}\right) \quad (6)$$

where ω is the inertia weight, c_1 and c_2 are cognitive and social learning factors respectively, r_1 and r_2 are random values between 0 and 1, $Pb_{i,j}$ is the personal best position of particle i for dimension j, and Gb_j is the global best position for dimension j. The position update as stated below is then used to update the position of each particle.

$$pos_{i,j}^{k+1} = pos_{i,j}^{k} + vel_{i,j}^{k+1} \quad (7)$$

After updating the position, boundary constraints are applied to ensure that particles remain within the search space. The fitness of each particle is then evaluated using pre-processed gene expression data. Next, the personal best and global best positions are updated based on the fitness of the particles. Finally, the convergence rate, diversity, and spread of the Pareto front are updated using the pre-processed data. These metrics provide insights into the convergence, diversity, and spread of solutions in the population, guiding the adaptive parameter adjustment process in subsequent iterations of the algorithm.

F. Adaptive Parameter Adjustment

To address the challenge of parameter tuning in the traditional MOPSO, we introduced an Adaptive Parameter Adjustment mechanism in our AAMOPSO method for gene selection. This mechanism dynamically adjusts the algorithm's parameters based on the observed performance metrics, aiming to optimize convergence, diversity, and overall efficiency in gene selection tasks. The adaptive adjustment starts by monitoring key performance metrics such as convergence rate (CR), diversity (D), and spread of the Pareto front (SP) over successive generations. These metrics are calculated using the preprocessed gene expression data obtained from the autoencoder. The Adaptive Parameter Adjustment mechanism involves defining adaptive rules to adjust parameters such as swarm size, inertia weight, cognitive, and social learning factors dynamically. For instance, if the convergence rate is slow and diversity is high, indicating suboptimal exploration, the mechanism may increase cognitive and social learning factors while decreasing swarm size to encourage exploration. Conversely, if the convergence rate is rapid but diversity is low, indicating premature convergence, the mechanism may decrease cognitive and social learning factors while increasing swarm size to promote exploration. Additionally, adjustments are made based on the spread of the Pareto front to balance exploration and exploitation. The detailed Adaptive Parameter Adjustment mechanism is shown as:

Step 1: *Monitoring Performance Metrics:* Calculate convergence rate (CR), diversity (D), and spread of the Pareto front (SP) at each generation using preprocessed gene expression data obtained from the autoencoder.

Step 2: *Calculate Changes in Performance Metrics:* Compute changes in performance metrics from the initial values:

$$\Delta CR = \frac{CR - CR_{\text{initial}}}{CR_{\text{initial}}} \quad (8)$$

$$\Delta D = \frac{D - D_{\text{initial}}}{D_{\text{initial}}} \quad (9)$$

$$\Delta SP = \frac{SP - SP_{\text{initial}}}{SP_{\text{initial}}} \quad (10)$$

Step 3: *Adaptive Parameter Adjustment:* Based on the changes in performance metrics, dynamically adjust the algorithm's parameters using adaptive rules:

- If $\Delta CR < 0$ and $\Delta D > 0$, increase cognitive and social learning factors (c_1) and (c_2), decrease swarm size (N) such as:

$$c_1^{t+1} = c_1^{t} + \Delta c_1 \quad (11)$$

$$c_2^{t+1} = c_2^{t} + \Delta c_2 \quad (12)$$

$$N^{t+1} = N^{t} - \Delta N \quad (13)$$

- If $\Delta CR > 0$ and $\Delta D < 0$, decrease cognitive and social learning factors (c_1) and (c_2), increase swarm size (N) such as:

$$c_1^{t+1} = c_1^{t} - \Delta c_1 \quad (14)$$

$$c_2^{t+1} = c_2^{t} - \Delta c_2 \quad (15)$$

$$N^{t+1} = N^{t} + \Delta N \quad (16)$$

- Adjust inertia weight (w) based on changes in spread of Pareto front (SP). If $\Delta SP < 0$, increase w, such as:

$$w^{t+1} = w^{t} + \Delta w \quad (17)$$

- If $\Delta SP > 0$, decrease w, such as:

$$w^{t+1} = w^{t} - \Delta w \quad (18)$$

Step 4: *Update Adaptive Parameters:* Update the initial values of performance metrics for the next iteration:

$$CR_{\text{initial}} = CR \quad (19)$$

$$D_{\text{initial}} = D \quad (20)$$

$$SP_{\text{initial}} = SP \quad (21)$$

By dynamically adapting parameters based on the optimization process's performance, the AAMOPSO method effectively addresses the challenge of parameter tuning in gene selection, leading to improved optimization performance and robustness.

G. Termination

At this step, the termination condition is checked. If the number of iterations is not more than the maximum allowed iteration, fitness function calculation, particle position, and velocity updating and mutation operation will be repeated. Otherwise, these steps will be stopped, and the global best position reported as the final feature set. Therefore, by following this stepwise working of the AAMOPSO method, gene selection can be efficiently performed while addressing scalability and parameter tuning issues using autoencoder-based preprocessing and adaptive parameter adjustment.

IV. RESULTS AND DISCUSSIONS

A. Experimental Setup

The performance of the proposed AAMOPSO method was evaluated using four microarray datasets available at: https://github.com/Pengeace/MGRFE-GaRFE, summarized in Table 1. All datasets were pre-processed to remove missing values and normalized using min-max scaling to ensure consistency across experiments.

Table 1: Dataset Description

No.	Dataset	Instances	Genes	Classes
1	Brain Tumor (BT)	90	5920	2
2	Leukaemia (LK)	72	5327	2
3	Lung Cancer (LCN)	203	12600	5
5	Breast Cancer (BRC)	104	22,283	2

The effectiveness of the proposed AAMOPSO method was compared against state-of-the-art metaheuristic algorithms commonly used for gene selection include: MORPSO_ECD [13], ANPMOPSO [14], MPSONC [20], MaPSOGS [19], and traditional MOPSO [21]. These comparisons were conducted on desktop with Intel Core i7 processor, 2.4 GHz with 16GB RAM by using Matlab 2018a mathematical development environment as the execution platform. All algorithms utilized a population size of 200 and 50 iterations. In MaPSOGS, MPSONC, MORPSO_ECD, ANPMOPSO, and MOPSO, both $c_1$and c_2 were set to 2.05 and w was set to 0.7298 [13]
[14]. Other parameters were configured according to corresponding references [19][20][21]. To ensure fair comparison and reliable evaluation, these experimental settings were applied consistently across all datasets. To enhance reliable results, each algorithm was run ten times for each microarray dataset. The average classification accuracy and number of selected genes across these ten runs were recorded and compared across algorithms.

B. Comparison of the Number of Selected Genes

Table 2 compares the average number of genes selected by AAMOPSO and five other multi-objective gene selection methods across all four microarray datasets. AAMOPSO consistently selects the smallest number of genes across all datasets, with as few as 8–17 genes, while still achieving superior classification accuracy (as shown in Figure 3.). In contrast, traditional methods like MOPSO and MPSONC tend to select significantly larger subsets (e.g., 44 genes for MOPSO in LCN), indicating possible redundancy and less efficient selection.

These results highlight AAMOPSO's strength in producing compact, non-redundant gene subsets that maintain or even improve classification performance. The integration of autoencoder-based feature compression and adaptive optimization helps reduce dimensionality effectively, making AAMOPSO particularly suitable for high-dimensional biomedical datasets where minimal gene sets are preferred for cost-effective diagnostic applications.

Table 2: The number of selected genes by different multi-objective gene selection methods.

Dataset	MOPSO	MPSONC	MaPSOGS	ANPMOPSO	MORPSO_ECD	AAMOPSO
BT	30	18	21	15	12	**10**
LK	27	16	18	10	**6**	8
LCN	44	32	27	18	14	**12**
BRC	35	24	29	25	22	**17**

C. Classification Performance Evaluation Using SVM Classifie

This experiment evaluates the performance of AAMOPSO gene selection method by comparing it with five gene selection methods on four microarray datasets (BRC, LNC, LK, BT). An SVM classifier was used consistently across all methods to measure classification accuracy based on varying numbers of selected genes.

The results depicted in Figure 3 demonstrate that the proposed AAMOPSO method consistently outperforms existing multi-objective gene selection approaches across all four datasets. AAMOPSO achieves higher classification accuracy with fewer selected genes, indicating its effectiveness in identifying compact yet highly informative gene subsets. Notably, in datasets like BRC and BT, AAMOPSO reaches near-peak accuracy (~95%) with only 15 genes, outperforming competitors such as MORPSO_ECD, ANPMOPSO, and MPSONC. This early convergence reflects the model's ability to extract discriminative features efficiently, aided by the integration of autoencoder-based dimensionality reduction.

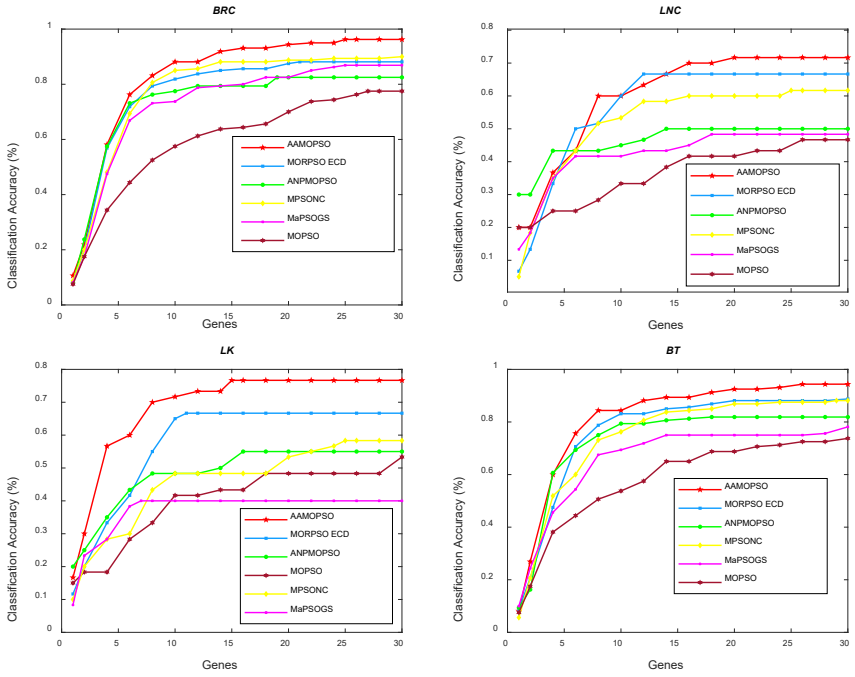

Figure 3: The Classification Accuracy vs. Number of Genes Results on Four Microarray Datasets for all Comparative Gene Selection Methods

In contrast, conventional methods like MOPSO and MaPSOGS exhibit slower performance gains and lower overall accuracy, particularly with smaller gene subsets, suggesting less effective exploration and selection strategies. Across all datasets, AAMOPSO maintains a clear margin of improvement, especially in the more complex LNC and LK datasets, highlighting its robustness. Overall, these results validate the superiority of AAMOPSO in achieving high classification performance with minimal gene subsets, making it a powerful tool for gene selection in high-dimensional biomedical data.

D. Functional and Biological Insights of the Selected Genes

Table 3 presents the top five most frequently selected genes by the proposed AAMOPSO algorithm across four benchmark gene expression datasets: LK, BT, LCN, and BRC. These genes were consistently identified across multiple runs, indicating their strong relevance and discriminative power in classifying cancer types. For instance, in the LK dataset, genes such as X13934 and Z79881 were frequently selected, suggesting their potential as biomarkers for leukemia. Similarly, in the BT dataset, genes like H75832 and 5189 appeared prominently, while in LCN and BRC, genes such as 39799, 1310_at, and NP_053056 stood out for their high selection frequency.

This consistent selection across different runs demonstrates AAMOPSO's robustness in identifying biologically meaningful and informative gene subsets. The results further support the method's effectiveness not just in achieving high classification accuracy, but also in uncovering potentially significant genetic markers for further biomedical research and validation.

Table 3: The top five frequently selected genes by AAMOPSO on all datasets

LK		BT		LCN		BRC	
Gene no.	Gene name	Gene no.	Gene name	Gene no.	Gene name	Gene no.	Gene name
4040	X13934	4213	U49817	**3188**	**39799**	346	NP_005749
1722	M37891	**5189**	**H75832**	265	42170	1842	NP_001885
2222	M73138	1715	R93186	2570	34884	**850**	**NP_002658**
129	L23852	6322	Y10317	**435**	**1310_at**	**1749**	**NP_053056**
5121	**Z79881**	5901	N80358	4207	33012_at	669	LIC561578

E. Statistical Significance of Classification Performance

Table 4 presents the p-values obtained using a one-tailed t-test to statistically compare the classification accuracy of each algorithm across four datasets: BT, LK, LCN, and BRC. The p-values indicate the statistical significance of improvements over a baseline (typically lower-performing algorithms), with lower values representing stronger evidence of performance difference. Across all datasets, AAMOPSO achieves the lowest or near-lowest p-values, such as 5.18E-10 (BT), 3.02E-05 (LK), and 2.71E-12 (BRC), highlighting the high statistical significance of its superior classification performance. While some methods like MORPSO_ECD and ANPMOPSO also yield competitive p-values, AAMOPSO consistently demonstrates better or comparable statistical reliability.

Table 4: P-Value Obtained Using Statistical Testing (One-Tailed T-Test) for Each Algorithm

Dataset	MOPSO	MPSONC	MaPSOGS	ANPMOPSO	MORPSO_ECD	AAMOPSO
BT	3.57E-02	1.33E-05	1.72E-07	3.25E-08	2.23E-04	**5.18E-10**
LK	3.68E-06	1.46E-05	2.19E-07	**4.89E-12**	2.19E-06	3.02E-05
LCN	3.12E-02	1.24E-08	2.41E-03	3.35E-03	**2.88E-10**	3.71E-09
BRC	3.22E-04	1.88E-08	2.23E-04	2.25E-04	1.78E-06	**2.71E-12**

These results confirm that the improvements offered by AAMOPSO are not due to random variation but are statistically meaningful, reinforcing its robustness and reliability for gene selection in high-dimensional cancer datasets.

V. CONCLUSIONS

In this study, we proposed AAMOPSO—an Autoencoder-based Adaptive Multi-Objective Particle Swarm Optimization approach—for effective gene selection in high-dimensional gene expression data. By integrating a deep autoencoder for unsupervised dimensionality reduction and an adaptive parameter tuning mechanism within the MOPSO framework, AAMOPSO addresses the core challenges of scalability and dynamic search control in gene selection tasks. Extensive experiments conducted on four benchmark cancer datasets (BRC, LCN, LK, BT) demonstrate that AAMOPSO consistently outperforms state-of-the-art methods in terms of classification accuracy, achieving competitive results using significantly fewer genes. Statistical analysis using one-tailed t-tests further confirms the superiority and reliability of AAMOPSO's performance.

Moreover, AAMOPSO's ability to frequently identify biologically meaningful genes across datasets underlines its potential for real-world biomedical applications such as biomarker discovery. The compact and highly discriminative gene subsets selected by AAMOPSO not only improve model interpretability but also reduce downstream analysis costs. Overall, AAMOPSO offers a robust, scalable, and accurate solution for gene selection, paving the way for enhanced genomic data analysis and precision medicine.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Yang et al., "TSPLASSO: A Two-Stage Prior LASSO Algorithm for Gene Selection Using Omics Data," IEEE Journal of Biomedical and Health Informatics, vol. 28, no. 1, pp. 526-537, 2023. Available from: https://doi.org/10.1109/JBHI.2023.3326485

[2] W. DeGroat et al., "Discovering Biomarkers Associated and Predicting Cardiovascular Disease With High Accuracy Using a Novel Nexus of Machine Learning Techniques for Precision Medicine," Scientific Reports, vol. 14, no. 1, 2024. Available from: https://doi.org/10.1038/s41598-023-50600-8

[3] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," Biometrics, vol. 44, no. 3, pp. 837-845, 1988. Available from: https://pubmed.ncbi.nlm.nih.gov/3203132/

[4] S. Kar, K. D. Sharma, and M. Maitra, "Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique," Expert Systems with Applications, vol. 42, no. 1, pp. 612-627, 2015. Available from: https://doi.org/10.1016/j.eswa.2014.08.014

[5] H. Zhang, Z. Zhu, H. Li, and S. He, "Network Biomarker Detection from Gene Co-expression Network Using Gaussian Mixture Model Clustering," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2023. Available from: https://doi.org/10.1109/TCBB.2023.3297388

[6] F. Han et al., "A hybrid gene selection method based on gene scoring strategy and improved particle swarm optimization," BMC Bioinformatics, vol. 20, pp. 1-13, 2019. Available from: https://doi.org/10.1186/s12859-019-2773-x

[7] F. Han et al., "A gene selection method for microarray data based on binary PSO encoding gene-to-class sensitivity information," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 14, no. 1, pp. 85-96, 2015. Available from: https://doi.org/10.1109/TCBB.2015.2465906

[8] Y. Xiong et al., "An efficient gene selection method for microarray data based on LASSO and BPSO," BMC Bioinformatics, vol. 20, pp. 1-13, 2019. Available from: https://doi.org/10.1186/s12859-019-3228-0

[9] F. Han et al., "A Feature Selection Method Based on Feature-Label Correlation Information and Self-Adaptive MOPSO," Neural Processing Letters, vol. 56, no. 2, 2024. Available from: https://doi.org/10.1007/s11063-024-11553-9

[10] C. S. R. Annavarapu, S. Dara, and H. Banka, "Cancer microarray data feature selection using multi-objective binary particle swarm optimization algorithm," EXCLI Journal, vol. 15, p. 460, 2016. Available from: https://doi.org/10.17179/excli2016-481

[11] F. Han et al., "Multi-objective particle swarm optimization with adaptive strategies for feature selection," Swarm and Evolutionary Computation, vol. 62, 2021. Available from: https://doi.org/10.1016/j.swevo.2021.100847

[12] H. Bakır et al., "Dynamic switched crowding-based multi-objective particle swarm optimization algorithm for solving multi-objective AC-DC optimal power flow problem," Applied Soft Computing, vol. 166, 2024. Available from: https://doi.org/10.1016/j.asoc.2024.112155

[13] S. Mehta et al., "MORPSO_ECD+ ELM: A Unified Framework for Gene Selection and Cancer Classification," 2025. Available from: https://doi.org/10.1109/JBHI.2025.3526825

[14] S. Mehta et al., "Gene selection based on adaptive neighborhood-preserving multi-objective particle swarm optimization," 2025. Available from: https://doi.org/10.7717/peerj-cs.2872

[15] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," Science, vol. 290, no. 5500, pp. 2319-2323, 2000. Available from: https://www.science.org/doi/10.1126/science.290.5500.2319

[16] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," IEEE Transactions on Evolutionary Computation, vol. 6, no. 5, pp. 443-462, 2002. Available from: https://doi.org/10.1109/TEVC.2002.800880

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504-507, 2006. Available from: https://www.science.org/doi/10.1126/science.1127647

[18] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 19-31, 2011. Available from: https://doi.org/10.1016/j.swevo.2011.02.001

[19] S. Azadifar and A. Ahmadi, "A graph-based gene selection method for medical diagnosis problems using a many-objective PSO algorithm," BMC Medical Informatics and Decision Making, vol. 21, no. 1, pp. 1-16, 2021. Available from: https://doi.org/10.1186/s12911-021-01696-3

[20] M. Rostami et al., "Integration of multi-objective PSO based feature selection and node centrality for medical datasets," Genomics, vol. 112, no. 6, pp. 4370-4384, 2020. Available from: https://doi.org/10.1016/j.ygeno.2020.07.027

[21] M. R. Rahimi et al., "Classification of cancer cells and gene selection based on microarray data using MOPSO algorithm," Journal of Cancer Research and Clinical Oncology, vol. 149, no. 16, pp. 15171-15184, 2023. Available from: https://doi.org/10.1007/s00432-023-05308-7