الجمهوريــة الجزائريــة الديمقراطيــة الشعبيــة
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université MUSTAPHA Stambouli
Mascara
Faculté des Sciences Exactes
Département d'Informatique

جامعة مصطفى اسطمبولي
معسكر

THESE DE DOCTORAT de 3ème cycle

Spécialité : Informatique
Option : Intelligence Artificielle

Intitulée

Vers un modèle de classification et de détection pour le diagnostic médical basés sur les algorithmes d'apprentissage.

Présentée par : **DJELLAL SERANDI Mohamed**

Le Lundi **29/06/2026 à 9h00**

À la Faculté des Sciences Exactes
Devant le jury :

Président	Benhaoua Mohammed kamal	PR	Université de Mascara
Directeur de thèse	Boufera Fatma	PR	Université de Mascara
Co-Directeur	Houari Amina	MCA	Université de Mascara
Examinateur	Medber Meriem	MCA	Université de Mascara
Examinateur	Nahnouh Chakib	MCA	Université de Chlef

Année Universitaire : **2025/2026**

الجمهوريــة الجزائريــة الديمقراطيــة الشعبيــة
People's Democratic Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry of Higher Education and Scientific Research

Mustapha Stambouli University
Mascara
Faculty of Exact Sciences
Department of Computer Science

جامعة مصطفى اسطمبولي
معسكر

DOCTORAL THESIS (Third Cycle)

Field: Computer Science
Speciality: Artificial Intelligence

Entitled

Towards a Classification and Detection Model for Medical Diagnosis Based on Machine Learning Algorithms

Presented by: **DJELLAL SERANDI Mohamed**

On **Monday, 29/06/2026 at 09:00**

By the jury :

Chairman	Benhaoua Mohammed kamal	PR	University of Mascara
Supervisor	Boufera Fatma	PR	University of Mascara
Co-Supervisor	Houari Amina	MCA	University of Mascara
Examiner	Medber Meriem	MCA	University of Mascara
Examiner	Nahnouh Chakib	MCA	University of Chlef

University year : **2025/2026**

Contents

List of Figures

List of Tables

I would like to dedicate this thesis to

My beloved parents

Your unwavering love, encouragement, and sacrifices have been the guiding light of my journey. This work is dedicated to you, the pillars of my strength. Your support has fueled my aspirations and shaped my path. I am forever grateful for the love and values you have instilled in me. This achievement is a reflection of the lessons learned from your wisdom. Thank you for being my constant source of inspiration.

My dear wife, and your family

Your patience, understanding, and unconditional love have been my greatest comfort and motivation. You stood by me through every challenge, offering encouragement and faith when I needed it most. This accomplishment is as much yours as it is mine. Thank you for being my unwavering partner and for filling my life with warmth and balance.

My precious children — Tariq, Abdelmalik, Aniss, Ibrahim, Marwa, and Zakaria

You are the joy of my life and the reason behind every effort I make. Your laughter and innocence have been a constant reminder of what truly matters. May this achievement inspire you to pursue your dreams with determination, integrity, and faith.

My dear siblings — Bakhta, Maamar, Chrifa, Djillali, Fatiha, Lalia, Anissa, Mhamed, and Aboubakr

In every step of my journey, your closeness has been a source of joy and strength. This dedication is a testament to the bond we share and the cherished memories that have enriched my life. Abdalwahab, for your guidance and support; Houcine, for your encouragement; my colleagues at work, for their flexibility and help — each of you has left an indelible mark on my heart. This accomplishment is as much yours as it is mine. Thank you for being my companions on this beautiful journey.

Acknowledgements

I would like to express my heartfelt appreciation to Mrs. BOUFERA Fatma for her steadfast support, valuable guidance, and remarkable expertise throughout my academic journey. Her commitment to excellence, insightful feedback, and constant encouragement have been vital to the successful completion of this thesis. I am deeply grateful for her dedication, mentorship, and the significant role she has played in my professional and personal development.

I would like to express my sincere appreciation to Mrs. HOUARI Amina for her invaluable mentorship and continuous support during the development of this thesis. Her expertise, constructive feedback, and commitment to academic excellence have played a decisive role in shaping the outcome of this work. I am truly thankful for her guidance and for fostering an environment that encouraged both learning and growth.

I wish to extend my heartfelt gratitude to Mr. FLITTI Farid for his exceptional guidance, constant encouragement, and insightful advice throughout the course of this thesis. His remarkable expertise and genuine dedication have been a source of continuous inspiration. I am profoundly grateful for his patience, availability, and his invaluable contributions that greatly enriched the quality of this research.

Abstract

High-throughput DNA microarray technology produces high-dimensional gene expression data characterized by a large number of irrelevant and redundant genes, which exacerbates the curse of dimensionality and negatively impacts disease classification performance. Consequently, effective feature selection is a critical step in gene expression analysis.

This thesis investigates bio-inspired and deep learning–based feature selection strategies for gene expression data to improve classification performance through comparative analyses between single-objective and multi-objective Differential Evolution frameworks, Differential Evolution and autoencoder-based feature learning approaches, and Differential Evolution and the Slime Mould Algorithm, highlighting the benefits of multi-objective optimization in balancing the mean squared residual (MSR) and feature subset compactness; moreover, a novel clustering-based multi-objective Differential Evolution framework (CBDE) is proposed to enhance search efficiency and identify compact and biologically meaningful gene subsets.

The proposed methods are evaluated on multiple benchmark microarray datasets using state-of-the-art classifiers, including Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). Experimental results demonstrate that the application of feature selection significantly improves convergence behavior and classification performance, while promoting a diverse set of optimal solutions. Moreover, the proposed approaches consistently achieve effective dimensionality reduction and robust performance across different datasets.

Keywords: Microarray data, Feature selection, Multi-objective , Differential evolution, single-objective, Autoencoder, Slime Mould Algorithm.

Résumé

La technologie de microarrays à haut débit génère des données d'expression génique de très grande dimension, souvent encombrées par de nombreux gènes non pertinents ou redondants. Cette situation accentue la malédiction de la dimensionnalité et affecte négativement la performance des méthodes de classification des maladies. Ainsi, la sélection efficace de caractéristiques constitue une étape cruciale dans l'analyse des données d'expression génique.

Cette thèse s'intéresse à des stratégies de sélection de caractéristiques basées à la fois sur l'inspiration biologique et sur l'apprentissage profond, dans le but d'améliorer la performance des classificateurs. Elle propose des analyses comparatives entre des cadres d'évolution différentielle à objectif unique et multi-objectifs, entre l'évolution différentielle et l'apprentissage de caractéristiques par autoencodeur, ainsi qu'entre l'évolution différentielle et l'algorithme Slime Mould. Ces analyses mettent en évidence les avantages de l'optimisation multi-objectifs pour équilibrer l'erreur quadratique moyenne (MSR) et la compacité des sous-ensembles de gènes. Par ailleurs, un nouveau cadre multi-objectifs basé sur le clustering et l'évolution différentielle (CBDE) est proposé afin d'améliorer l'efficacité de la recherche et d'identifier des sous-ensembles de gènes à la fois compacts et biologiquement significatifs.

Les méthodes proposées sont évaluées sur plusieurs ensembles de données microarrays de référence à l'aide de classificateurs performants, tels que le Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF) et la régression logistique (LR). Les résultats expérimentaux montrent que l'application de la sélection de caractéristiques améliore significativement la convergence des algorithmes et la performance des classificateurs, tout en favorisant un ensemble diversifié de solutions optimales. De plus, les approches proposées assurent une réduction efficace de la dimensionnalité et une robustesse des performances sur différents ensembles de données.

Mots-clés : Données de microarrays, Sélection de caractéristiques, Multi-objectifs, Évolution différentielle, Mono-objectif, Autoencodeur, Algorithme Slime Mould.

General Introduction

Context and Motivation

The rapid expansion of microarray-based gene expression technologies has generated datasets of unprecedented scale and complexity. These datasets typically comprise tens of thousands of genes measured across relatively few samples, creating a pronounced imbalance that gives rise to the well-known curse of dimensionality [1]. As dimensionality increases, the volume of data required to derive reliable statistical or machine-learning inferences grows exponentially, often exceeding what is realistically attainable in biomedical research. This imbalance compromises the robustness and generalization capacity of predictive models, making them highly sensitive to noise and prone to overfitting.

Big data challenges are particularly evident in genomics. The swift advancement of high-throughput genomic technologies has enabled large-scale data acquisition, with microarray analysis producing vast volumes of gene expression data. Such datasets are typically organized into matrices, with genes along rows and experimental conditions or samples along columns. This rich information provides valuable insights into genetic mechanisms and their implications across diverse domains, including evolutionary biology, environmental studies, and biomedical research. Nevertheless, the size, complexity, and inherent noisiness of genomic data require sophisticated computational and analytical methods capable of uncovering meaningful patterns without being misled by redundancy or irrelevant features.

These limitations have a direct influence on disease classification, where the goal is to discriminate between healthy and pathological states or among disease subtypes. When the feature space is excessively large relative to the number of samples, conventional classification algorithms tend to produce unstable decision boundaries, resulting in reduced predictive performance. Consequently, the accuracy obtained during training often overestimates the true performance on unseen data, limiting the practical applicability of such models. Addressing the curse of dimensionality is therefore essential to improve prediction accuracy, enhance model reliability, and

enable the identification of biologically relevant genes that can support diagnostic and therapeutic decision-making.

To mitigate these challenges, Feature Selection (FS) [2] has emerged as a fundamental preprocessing step. FS aims to identify a compact yet biologically meaningful subset of genes while preserving the discriminative power of the dataset. Effective FS not only enhances classification performance and reduces computational cost, but also promotes interpretability—a critical aspect in biomedical research where understanding the molecular drivers of a disease is as important as achieving high predictive performance [3]. Moreover, despite the richness of high-throughput datasets, the informative genes can remain obscured by noise, redundancy, and irrelevant features if rigorous FS techniques are not applied [4].

Selecting an optimal subset of features is an NP-hard combinatorial problem, requiring significant computational resources. Therefore, choosing an appropriate search strategy is crucial for ensuring both the effectiveness (i.e., identifying high-quality feature subsets) and efficiency (i.e., achieving results within reasonable time constraints) of the FS process [5]. Traditional exhaustive or deterministic methods quickly become intractable when dealing with thousands of genes, motivating the adoption of heuristic and metaheuristic approaches.

In recent years, nature-inspired algorithms have gained prominence in microarray feature selection [6]. Their effectiveness stems from the collective behavior of simple agents following self-organizing principles, enabling them to efficiently explore large search spaces while avoiding local optima. Well-known nature-inspired methods include Differential Evolution (DE) [7], Artificial Bee Colony (ABC) [8], Particle Swarm Optimization (PSO) [9], Ant Colony Optimization (ACO) [10], Slime Mould Algorithm (SMA) [11], and various hybrid strategies. These algorithms have demonstrated significant potential in selecting informative gene subsets, improving predictive accuracy, and supporting biological interpretability.

In general, FS involves four main steps: subset generation, subset evaluation, stopping criteria, and result confirmation. During subset generation, candidate feature subsets are produced by the selected search strategy. An evaluation criterion then assesses the quality of each candidate, and the search continues iteratively until a predefined stopping condition is met. Finally, the optimal subset is validated on independent test data to confirm its effectiveness [12]. The choice of evaluation metric, search algorithm, and stopping criteria critically influences both the stability and predictive power of the selected features.

Despite the growing adoption of nature-inspired algorithms, several challenges remain. Many algorithms struggle with convergence in high-dimensional, noisy

datasets or fail to ensure stable selection across multiple runs. Furthermore, balancing exploration (searching broadly across the feature space) and exploitation (refining promising subsets) remains a central challenge, particularly when the number of genes greatly exceeds the number of samples. These limitations highlight the need for novel or hybrid nature-inspired feature selection approaches that can enhance robustness, stability, and biological relevance.

Contributions and publications

To overcome these limitations, this thesis aims to enhance the performance, stability, and generalizability of classification models for gene expression data by developing feature selection approaches based on Differential Evolution, specifically tailored to the characteristics and constraints of such high-dimensional datasets.

1. **DeMoFs:** A Multi-objective Feature Selection Framework Using Binary Differential Evolution (BDE) for Breast Cancer Microarray Data. DeMoFs implements a novel multi-objective feature selection strategy based on Differential Evolution, adapted to a binary search space for high-dimensional gene expression datasets. It exploits the global optimization capabilities of DE to effectively balance model performance and interpretability while minimizing redundancy and noise. Through an enhanced mutation operator and multi-objective evaluation criteria, DeMoFs aims to identify compact, biologically relevant gene subsets that improve classification accuracy and facilitate a better understanding of disease-related genetic mechanisms.

    - **Djellal Serandi, M.**, Houari, A., Boufera, F., Flitti, F. (2025). DeMoFs: Multi-objective Feature Selection Using Binary Differential Evolution (BDE) Optimization on Breast Cancer Gene Expression Data. In: Seddari, N., Redjimi, M. (eds) Modeling, Simulation and Computer Technology. ICMSCT 2024. Communications in Computer and Information Science, vol 2606. Springer, Cham. https://doi.org/10.1007/978-3-032-01922-6_22

2. **FsMo – FsSo:** A Comparative Framework for Multi-Objective and Single-Objective Feature Selection Using Binary Differential Evolution on CNS Gene Expression Data. FsMo – FsSo leverages the global optimization strengths of BDE to tackle the challenges of high-dimensional CNS gene expression

data. The framework implements both multi-objective (FsMo) and single-objective (FsSo) feature selection strategies, enabling a systematic comparison of their effectiveness in identifying informative gene subsets. FsMo – FsSo is designed to improve classification performance, model interpretability, and data quality while reducing dimensionality. Experimental results show that the multi-objective approach (FsMo) outperforms both the single-objective method (FsSo) and the original dataset, confirming its ability to extract compact, biologically meaningful gene subsets.

- **Djellal Serandi, M.**, Boufera F., Houari A., Flitti F. "FsMo - FsSo: A Comparative Study of Multi-Objective and Single-Objective Feature Selection Using Binary Differential Evolution on the CNS Gene Expression Data " 2025, The first international Conference on Artificial Intelligence, Smart Technologies and Communications (AISTC'25) Chlef, Algeria, 2025.

3. **FsMo – FsAe:** A Comparative Framework for Multi-Objective Binary Differential Evolution and Autoencoder-Based Feature Selection on Lung Gene Expression Data. FsMo – FsAe combines the global search capabilities of BDE with the representation power of autoencoders to handle high-dimensional gene expression datasets effectively. The framework simultaneously optimizes the Mean Squared Residual (MSR) while limiting the number of selected genes in the multi-objective DE approach and compares it with an autoencoder-based selection method. FsMo – FsAe is designed to efficiently identify compact, biologically relevant gene subsets, enhancing classification performance, model interpretability, and overall data quality. Experimental results demonstrate that the multi-objective approach (FsMo) outperforms both the autoencoder-based method and the original dataset, highlighting its effectiveness in high-dimensional genomic settings.

   - **Djellal Serandi, M.**, Boufera F., Houari A., Flitti F. "FsMo - FsAe: A Comparative Study of Multi-Objective Using Binary Differential Evolution and Autoencoder-Based Feature Selection on the Lung Gene Expression Data " 2025, The second international Conference on Artificial Theories and Applications (ICAITA25) Mascara, Algeria, 2025.

4. **DeMoFs – Lung:** A Multi-objective Feature Selection Framework Using BDE on Lung Cancer Gene Expression Data. DeMoFs-Lung utilizes the robust global search capabilities of Differential Evolution (DE) to address the

challenges of high-dimensional lung cancer datasets. The framework simultaneously optimizes the MSR while constraining the number of selected genes, aiming to enhance both the predictive performance and interpretability of classification models. By adapting DE to a binary search space and integrating a multi-objective evaluation strategy, DeMoFs-Lung efficiently extracts compact, biologically meaningful gene subsets suitable for lung cancer microarray data.

- **Djellal Serandi, M.**, Boufera F., Houari A., Flitti F. "DeMoFs-Lung: Multi-objective Feature Selection using binary Differential Evolution (BDE) Optimization on Lung Cancer Gene Expression Data" 2025, First National Conference on: "Artificial Intelligence and Socio-Economic Applications AISEA25" Biskra, 5 and 6 November, 2025

5. **FsSMA – FsDE:** A Comparative Feature Selection Framework Using Slime Mould Algorithm and Differential Evolution for Lymphoma Gene Expression Data. FsSMA – FsDE applies the complementary strengths of the Slime Mould Algorithm (SMA) and Differential Evolution (DE) to explore high-dimensional gene expression spaces. The framework is designed to identify the most informative genes, reduce dimensionality, and enhance both classification performance and model interpretability. DE is formulated as a single-objective optimization approach to maximize classification accuracy, while SMA provides an alternative exploration mechanism, enabling a comprehensive comparison of their respective capacities to extract biologically meaningful gene subsets.

   - **Djellal Serandi, M.**, Boufera F., Houari A., Flitti F. "FsSMA – FsDE: Comparative Study of Slime Mould Algorithm and Differential Evolution for Feature Selection on Lymphoma Gene Expression Data" 2025, The first international Conference on Applied Artificial Intelligence and Emerging Technologies (AAIET'2025) Djelfa, Algeria, 2025.

6. **DeFs – CBDE:** A Multi-Objective Feature Selection Framework Using Clustering-Guided Binary Differential Evolution for Microarray Gene Expression Data. DeFs-CBDE leverages the strengths of Differential Evolution (DE) enhanced with a novel Clustering Binary Differential Evolution (CBDE) mutation operator to efficiently explore high-dimensional gene expression spaces. The framework is designed to simultaneously optimize the MSR and constrain the number of selected genes, aiming to improve classification performance and

model interpretability. By integrating the clustering-guided mutation mechanism, DeFs-CBDE adapts DE for more effective feature selection, enabling the identification of compact, biologically meaningful gene subsets suitable for microarray datasets.

- **Djellal Serandi, M.**, Boufera F., Houari A., Flitti F. DeFs-CBDE: Clustering-guided binary mutation in multi-objective differential evolution for microarray gene selection. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2025, vol. 25, no. 6, pp. 1185–1196. doi: 10.17586/2226-1494-2025-25-6-1185-1196

Thesis Organization

After presenting the general introduction in the first chapter, the thesis proceeds with the following chapters, organized as follows:

- **Chapter 1:** focuses on enhancing classification performance in high-dimensional gene expression data, and then presents a thorough review of feature selection methods, discussing their strengths, weaknesses, and impact on predictive modeling.
- **Chapter 2:** presents a detailed exploration of feature selection techniques and optimization strategies applied to high-dimensional gene expression datasets. The chapter begins with a discussion of Differential Evolution (DE) and its adaptation to both single-objective (FsSo) and multi-objective (FsMo) feature selection frameworks, highlighting their ability to efficiently search large and complex gene spaces. It then introduces autoencoder-based feature selection methods, emphasizing their capacity to extract meaningful latent representations from high-dimensional data. Finally, the chapter covers the Slime Mould Algorithm (SMA) and its application to gene selection, showcasing how its bio-inspired exploration mechanisms contribute to identifying informative and biologically relevant gene subsets. Throughout the chapter, the focus is on explaining the principles, algorithmic details, and practical applications of each method in improving classification performance and model interpretability.
- **Chapter 3:** presents a comprehensive analysis of the performance and characteristics of various feature selection approaches applied to high-dimensional

gene expression data. The chapter provides a systematic comparison between single-objective (FsSo) and multi-objective (FsMo) Differential Evolution frameworks, highlighting the benefits of multi-objective optimization in improving classification performance and identifying biologically meaningful gene subsets. It further examines the integration of autoencoder-based feature selection (FsAe) with multi-objective DE, evaluating the advantages of combining dimensionality reduction with multi-objective optimization. Finally, the chapter explores the application of the Slime Mould Algorithm (FsSMA) alongside multi-objective DE (FsMo), emphasizing their complementary strengths in gene selection and their impact on enhancing model interpretability and predictive accuracy. Building on this comparative analysis, Chapter 4 delves into multi-objective Differential Evolution and presents the DeFs-CBDE framework, detailing its innovative Clustering Binary Differential Evolution (CBDE) mutation operator, the simultaneous optimization of MSR and gene subset size, and demonstrating its effectiveness in selecting biologically informative genes while enhancing classification performance and model interpretability.

- **Chapter 4:** investigates multi-objective Differential Evolution in depth, providing a detailed discussion of its underlying principles, operational mechanisms, and practical strengths. The chapter further introduces the DeFs-CBDE framework, highlighting its novel Clustering Binary Differential Evolution (CBDE) mutation operator, which enables the simultaneous optimization of MSR and gene subset size. Through extensive experiments, the chapter demonstrates how this framework efficiently identifies informative, biologically relevant, and functionally significant genes from high-dimensional microarray datasets. By combining dimensionality reduction, multi-objective optimization, and clustering-based search strategies, the approach not only enhances classification accuracy but also improves the biological interpretability, robustness, and practical relevance of predictive models.

Chapter 1

State of the Art

1.1 Introduction

Disease classification using gene expression and genetic data has become a cornerstone of modern computational genomics, supporting crucial tasks such as distinguishing disease states, identifying molecular subtypes, and guiding targeted therapies. Despite their value, these datasets are characterized by high dimensionality, small sample sizes, and substantial redundancy, which undermine the performance, generalization, and interpretability of predictive models.

Feature selection has therefore become an essential component in the analysis of high-dimensional gene expression data, where the objective is to identify compact and informative subsets of genes capable of enhancing prediction accuracy and supporting biological interpretation. This chapter begins by defining the fundamentals of feature selection, presenting its categories, objectives, and evaluation measures.

The chapter then focuses on recent advances in single-objective and multi-objective optimization for feature selection, with a particular emphasis on Differential Evolution, swarm intelligence, and hybrid bio-inspired techniques. Through this examination, we aim to clarify the motivations for adopting optimization-based feature selection and to contextualize the contributions proposed in this thesis within the current state of the art.

1.2 Microarray data

The Microarray experiment generates data that is organized and stored as a large ($N \times M$) matrix [13]. Each Microarray data matrix is composed of samples depicted in rows and genes (features) illustrated in columns, As seen in Table 1.1, an illustrative example of a microarray data matrix structure is presented, where S_i represents the

samples, with each sample i corresponding to an observation unit. Each feature G_j refers to a gene or genetic marker, and each entry x_{ij} indicates the expression level of gene G_j in sample S_i. In this context, S denotes the total number of samples, G represents the total number of genes (features), and x_{ij} specifies the measured gene expression value at the intersection of sample S_i and gene G_j.

Table 1.1: A microarray data matrix structure example.

	G_1	G_2	$\cdots$	G_M
S_1	x_{11}	x_{12}	$\cdots$	x_{1M}
S_2	x_{21}	x_{22}	$\cdots$	x_{2M}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
S_N	x_{N1}	x_{N2}	$\cdots$	x_{NM}

1.3 Feature selection

Feature selection plays a critical role in the analysis of high-dimensional genomic datasets, where the objective is to identify small subsets of informative genes capable of improving disease classification accuracy while preserving biological interpretability. Conventional feature selection methods often rely on deterministic or gradient-based optimization processes, which may be inefficient or ineffective when dealing with complex, nonlinear, and noisy data distributions commonly encountered in gene expression analysis.

Bio-inspired optimization algorithms (BIA) constitute a major class of metaheuristic approaches grounded in principles derived from biological evolution and collective intelligence. These algorithms aim to construct robust and efficient optimization frameworks by abstracting mechanisms observed in natural systems [14]. Within this family, two fundamental paradigms are commonly distinguished: Evolutionary Algorithms (EA) and Swarm Intelligence (SI), which differ in their sources of inspiration and search dynamics.

Evolutionary Algorithms are inspired by Darwinian evolution and operate through population-based search processes driven by natural selection, reproduction, mutation, and crossover. By iteratively evolving candidate solutions, EA demonstrate strong capabilities in exploring complex, high-dimensional, nonlinear, and multimodal search spaces. Consequently, they have been extensively adopted for feature selection in gene expression data, where the number of features vastly exceeds the number of samples. Feature selection techniques are generally classified into filter, wrapper, and embedded methods, among which wrapper and embedded approaches

benefit significantly from EA due to their ability to evaluate feature subsets in direct interaction with learning models [15], thus achieving an effective trade-off between predictive performance, computational cost, and robustness.

In contrast, Swarm Intelligence algorithms are inspired by the self-organized and decentralized behaviors exhibited by social organisms—such as ant colonies, bird flocks, bee swarms, and fish schools—where simple agents cooperate locally to produce emergent global intelligence [14]. Through mechanisms of information sharing and collective adaptation, SI algorithms enhance scalability and stability, making them particularly suitable for large-scale and dynamic optimization problems.

Overall, bio-inspired metaheuristics provide powerful and flexible strategies for identifying optimal or near-optimal feature subsets in high-dimensional biomedical datasets. By leveraging either evolutionary adaptation or collective cooperation, EA and SI offer complementary strengths that contribute to improved efficiency, robustness, and biological relevance in feature selection tasks [16].

1.3.1 Evolutionary Algorithms

The evolutionary paradigm constitutes the most established and theoretically grounded class of bio-inspired algorithms. Foundational approaches such as Genetic Algorithms (GA), Genetic Programming (GP), Evolution Strategies (ES), and Differential Evolution (DE) are widely recognized as methodological cornerstones in this field. Their core operators—including selection, mutation, crossover, self-adaptation, and vector differencing—introduced substantive algorithmic innovations rather than purely metaphorical inspirations. Among them, DE remains a prominent baseline due to its simplicity and effectiveness, while ES is distinguished by its strong mathematical foundations in self-adaptation theory. These classical evolutionary models continue to underpin both theoretical developments and practical applications, in contrast to later evolution-inspired methods that largely reframe similar mechanisms using alternative metaphors [17].

1.3.2 Swarm Intelligence Algorithms

Swarm Intelligence represents a second well-established paradigm within bio-inspired algorithms. Particle Swarm Optimization (PSO) formalizes cognitive and social learning mechanisms through velocity-based position updates, while Ant Colony Optimization (ACO) introduces stigmergic communication as an effective framework for solving discrete optimization problems. Both approaches remain extensively benchmarked and are conceptually distinct within the swarm intelligence literature.

In addition, the Artificial Bee Colony (ABC) algorithm continues to be recognized as a representative and validated swarm-based optimization model [17].

1.4 Related Work

In recent years, bio-inspired optimization techniques, including Evolutionary Algorithms (EA) and Swarm Intelligence (SI) methods, have become increasingly prevalent in microarray feature selection due to their ability to efficiently explore high-dimensional search spaces, maintain solution diversity, and avoid local optima, with well-known approaches such as Differential Evolution (DE), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), and various hybrid methods [6].

1.4.1 DE approaches for feature selection:

Differential evolution (DE) is a population-based metaheuristic search algorithm that optimizes a problem by iteratively improving a candidate solution based on an evolutionary process [18].

Hashmi et al. [19] proposed a two-step method: (1) feature extraction and ranking with 6 well-known filter methods namely Information Gain, Correlation, and Relief among others, and (2) DE optimization of the selected features subset. The method was applied to four cancer datasets: Brain, Breast, Lung, and CNS.

Hassan et al. [20] integrated the Multi-Variant Differential Evolution (MVDE) algorithm, which utilizes multi-mutation and crossover integration within an adaptive scale feature selection framework. Their results demonstrated that MVDE outperforms several metaheuristics, including standard DE, Cat Swarm Optimization (CSO), Whale Optimization Algorithm (WOA), and Sine Cosine Algorithm (SCA), in terms of both classification accuracy and feature selection efficiency.

Zhang et al. [21] introduced a multi-objective approach named MOFS-BDE (Multi-Objective Feature Selection based on Binary Differential Evolution with Self-Learning). This framework defines three novel operators: a probability-based binary mutation to enhance convergence, and two self-learning mechanisms—One-bit Purifying Search and Self-learning Purify Search—to refine the search process.

Serandi et al. [22] addressed the challenge of high dimensionality in microarray gene expression data, where thousands of genes are combined with a limited number of samples, leading to the ”curse of dimensionality”. To tackle this, they proposed a multi-objective feature selection method based on DE that simultaneously optimizes

the number of selected genes and the Mean Squared Residual (MSR). A core innovation of this work is the novel Clustering-Based Binary Differential Evolution (CBDE) mutation operator. Extensively validated on four datasets (Brain, Breast, Lung, and CNS), the method demonstrated significant improvements in classification accuracy and selection efficiency compared to classical DE and other state-of-the-art feature selection methods.

1.4.2 PSO approaches for feature selection:

Particle Swarm Optimization (PSO) is a stochastic optimization method inspired by social behavior and swarm intelligence. Unlike traditional deterministic methods, PSO explores the search space by simulating the movement of particles toward an optimum [9].

Xue et al. [23] proposed a multi-objective PSO (MOPSO) algorithm for feature selection, aimed at simultaneously maximizing classification accuracy and minimizing the number of selected features. Their approach demonstrated exceptional performance in dimensionality reduction while maintaining high predictive accuracy.

Ahadzadeh et al. [24] used an algorithm called simple, fast, and efficient (SFE) to search a vast area of high-dimensional data, eliminating irrelevant characteristics from the search space in the first phase and identifying significant and pertinent features. Next, the smaller search space is searched using the PSO method.

Ludwig [25] proposed a guided Particle Swarm Optimization PSO–based feature selection method for high-dimensional gene expression data. The approach incorporates multiple filter-based techniques to guide the particle search process, effectively eliminating irrelevant features and reducing the search space in the early stages. This guided mechanism mitigates the premature convergence problem commonly encountered in conventional PSO-based feature selection methods. The proposed algorithm was evaluated on very high-dimensional genomic datasets containing up to 44,909 features and demonstrated superior performance compared to several state-of-the-art feature selection approaches in terms of classification accuracy and computational efficiency.

Paul et al. [26] constructed a Multi-objective Particle Swarm Optimization (MOPSO) algorithm that is employed with three stage filtering to dynamically select features within the incoming groups. This approach tries to eliminate redundancy and at the same time maximize relevance of the data with respect to class labels. It also incorporates PSO's evolutionary search features so as to outperform current approaches in many multi-label datasets, using its effectiveness in dynamic feature streams.

Han et al. [27] suggested a Multi-Objective Particle Swarm Optimization with Adaptive Strategies (MOPSO-ASFS) for feature selection. The approach adds an adaptive penalty of Penalty Boundary Interaction for selection pressure reduction to improve diversity and convergence. Also, an adaptive leading particle selection technique combines feature frequency and anti-mutation to improve swarm diversity. The experiments results from 14 UCI datasets and 6 gene expression datasets show evidence that MOPSO-ASFS outperforms the classical multi-objective feature selection algorithms, most especially on the multi-dimensional regions.

1.4.3 ACO approaches for feature selection:

Early in the 1990s, Marco Dorigo and colleagues introduced the first Ant Colony Optimization (ACO) techniques. These algorithms were inspired by the collective behavior of social ant colonies [9], specifically their ability to find optimal paths through stigmergic communication.

Ma et al. [28] designed a two-stage hybrid ACO algorithm aimed at high-dimensional feature selection. In the first stage, the interval-based approach determines the optimum feature subset size. The second stage chooses relevant features by employing ACO with performance-based classification guidance. The results of the experiments carried out show better accuracy and lower computational cost when compared to other ACO techniques.

Meenachi et al. [29] proposed a metaheuristic search-based feature selection approach is proposed for cancer classification using microarray gene expression data. This study regards the usage of ACO, Genetic Algorithm (GA), and Tabu Search for selecting the most relevant gene subsets with the proper application of the global/local search strategy.

Yilmaz et al. [30] introduced an adapted ACO-based method to address the specific challenges of high-dimensional data. By integrating a heuristic distance directly into the probability function, the approach eliminates the need for predefined sub-attribute sets. It iteratively constructs a frequency-ordered list to determine feature importance. Validated against fifteen competing algorithms, the results show that this frequency-ordered construction significantly enhances classification performance, while convergence analysis confirms its suitability for complex genomic datasets.

Santhakumar et al. [31] proposed a hybrid approach combining Ant Colony Optimization (ACO) and Ant Lion Optimizer (ALO) for gene selection and classification on microarray datasets, improving classification accuracy and reducing computational complexity compared to conventional methods.

1.4.4 ABC approaches for feature selection:

In 2005, Derviş Karaboga introduced the Artificial Bee Colony (ABC) algorithm, an optimization technique inspired by the intelligent foraging behavior of honeybee swarms. Under the ABC paradigm, the colony consists of three distinct types of bees: employed bees (workers), observers, and scouts [9].

Aziz et al. [32] presented a new nature inspired framework for gene selection and classification of cancer that applies the Cuckoo Search Algorithm (CSA) to the onlooker phase of the ABC algorithm. The strategy is accomplished in two steps. The first step employs the Independent Component Analysis (ICA) for dimensionality reduction Naïve Bayes classification while the second step does gene selection with a modified ABC algorithm. According to the experiment results, the framework4 improves the Naïve Bayes classifier's performance with less genes.

Wang et al. [33] proposed a self adaptive level based learning artificial bee sclony (SLLABC) algorithm designed for high dimensional feature selection. The standard ABC algorithm is improved by using a level-based learning mechanism, in which bees are ranked according to their fitness and imitate those in upper levels to enhance exploitation. This structure is simple, yet effective because it solves the issues of slow convergence and lack of exploitation ABC has. It is shown through experiments that SLLABC significantly improves classification accuracy and feature reduction in comparison to traditional ABC feature selection approaches.

Xu et al. [34] proposed an enhanced version of the Artificial Bee Colony (ABC) algorithm for feature selection by introducing a deduplication and differential analysis mechanism to increase population diversity and accelerate convergence. This improvement is particularly useful for feature selection tasks on complex, high-dimensional datasets. The approach reduces the number of selected features while improving classification accuracy compared to several competing methods.

Gulande et al. [35] developed a hybrid algorithm combining the Artificial Bee Colony (ABC) and Firefly Algorithm for gene expression data classification. This swarm intelligence–based approach was validated on microarray datasets and demonstrated superior performance in selecting relevant genes and in classification compared to several standard techniques, particularly in terms of accuracy for cancer detection and classification.

1.4.5 Autoencoder-Based approaches for feature selection

Using deep learning paradigms for feature selection, autoencoder-based approaches employ representation learning to capture intrinsic structures in high-dimensional

gene expression data while reducing redundancy and noise. Many recent studies have emphasized unsupervised and semi-supervised autoencoder models that either perform indirect gene selection through latent representations or explicitly integrate feature selection mechanisms into the network architecture, achieving effective dimensionality reduction without significant loss of classification performance [36].

Balın et al. [37] proposed the Concrete Autoencoder, a differentiable feature selection framework that enables the direct selection of original input features while preserving reconstruction capability. The method was successfully applied to gene expression datasets and demonstrated competitive performance compared to traditional feature selection techniques.

Wang et al. [38] introduced an adaptive autoencoder-based feature selection approach with redundancy control, where sparsity and reconstruction objectives are jointly optimized to eliminate irrelevant and redundant genes. Experimental results on multiple microarray cancer datasets showed improved classification accuracy with a reduced number of selected genes.

Li et al. [39] proposed a graph-regularized autoencoder for unsupervised feature selection, incorporating gene relationship structures into the learning process. By preserving local manifold information, the method achieved enhanced stability and robustness in selecting biologically meaningful gene subsets.

Hassanieh et al. [40] presented a selective deep autoencoder model designed specifically for unsupervised feature selection. The proposed architecture embeds feature selection constraints directly into the encoding process, enabling the extraction of compact and informative gene subsets, with promising results on cancer gene expression datasets.

Zhang et al. [41] further extended selective autoencoder frameworks by introducing deeper architectures and improved optimization strategies for unsupervised feature selection. Their experiments confirmed that selective deep autoencoders can achieve performance comparable to or better than evolutionary and swarm-based feature selection methods on high-dimensional biological datasets.

Despite these advances, most autoencoder-based feature selection approaches primarily rely on reconstruction-driven objectives and lack explicit global search mechanisms. Consequently, hybrid frameworks combining autoencoders with evolutionary or swarm intelligence algorithms have emerged as a promising direction to enhance global optimization capability and gene subset optimality in microarray data analysis.

1.4.6 Hybrid approaches:

Using a series of metaheuristic approaches to the choice of features, use different adaptation strategies to promote search efficiency and increase classification exhibition. Many studies have emphasized hybrid methods that add filters and the wrapper technique to limit the facility of the highest and reduce dimensions effectively without compromising the accuracy. A selection of these methods is referred to at the end [42].

Dashtban et al. [43], presented a hybrid approach using the Fisher test and traditional Bat Algorithm with sophisticated formulations, effective multi-objective operator and new local search strategies that use social teaching ideas in manipulating random walks. The accuracy of the selected most was used for four separate uses classifiers which include SVM, KNN, NB and DT.

Li et al. [44], the study successfully addressed these difficulties the absence of label information in cluster features by presenting a unique informal-visited functional choice technique based on an extended binary difference development algorithm. In addition, it combines convenience choice with grouping adaptation.

Lamba et al. [45], the paper introduces a hybrid gene selection model that combines filter and wrapper methods for feature selection. This approach enhances the identification of relevant genes crucial for breast cancer classification, thereby improving the model's accuracy. It illustrates how deep neural networks are used to classify the molecular subtypes of breast cancer.

Li et al. [46] propose a bi-objective evolutionary feature selection algorithm that reduces the size of the search space by applying an improved Fisher score to eliminate irrelevant and redundant features, leading to more efficient optimization and competitive classification performance on high-dimensional datasets.

The authors in [47] propose a hybrid gene selection strategy (MFI-RFPA) that integrates multiple filter methods—namely Information Gain Ratio (IGR), Relief-F, and Chi-squared—to eliminate irrelevant and noisy genes, followed by a recursive Flower Pollination Algorithm to identify a compact gene subset with improved tumor classification performance.

The authors in [48] propose an evolutionary multitasking–based feature selection approach that decomposes the original high-dimensional search space into multiple related low-dimensional tasks using different filtering methods, thereby reducing the search space of the competitive swarm optimizer (CSO) and enhancing feature selection performance through effective knowledge transfer.

Cheng et al. [49] propose a variable-granularity multi-objective evolutionary algorithm that groups features using clustering techniques such as K-means, thereby

reducing the search space and enabling efficient feature selection in high-dimensional classification problems.

Ali et al. [50], suggest a feature selection strategy based on a hybrid filter-genetic algorithm to enhance the classification of cancer in high-dimensional microarray datasets. The method first uses filter-based techniques Information Gain(IG), Gain Ratio(IGR), and Chi-Squared(CS). Then there is a genetic algorithm (GA) Classification is used to adapt further, selected properties and increase accuracy. The approach was tested on four microarray datasets (brain, breast, lung, and CNS) and performed better than traditionally convenience choice methods.

Xu et al. [51], introduce a novel feature filter and group evolution hybrid feature selection algorithm (FG-HFS) aimed at addressing the limitations of existing methods. It employs spectral clustering to group features, filters redundant features using symmetric uncertainty, and utilizes a group evolution multi-objective genetic algorithm for optimal feature selection. The effectiveness of the FG-HFS algorithm is validated through experiments on five datasets from different fields, demonstrating its practical application value in gene expression data analysis.

Table 1.2: Summary of Feature Selection Methods for Gene Expression Data

Author(s)	**Year**	**Method**	**Dataset**	**Advantages**	**Limitations**
Hashmi et al. [19]	2024	Hybrid Filter + Differential Evolution	Brain, Breast, Lung, CNS microarray datasets	Effective dimensionality reduction; improved classification accuracy; robust hybrid framework	Performance depends on filter quality; increased preprocessing complexity
Hassan et al. [20]	2020	Multi-Variant Differential Evolution (MVDE)	Gene expression datasets	Enhanced exploration and exploitation; superior accuracy compared to DE, CSO, and WOA	Higher computational cost; sensitive parameter tuning
Zhang et al. [21]	2020	Multi-objective Binary DE (MOFS-BDE)	Benchmark feature selection datasets	Simultaneous optimization of accuracy and subset size; effective binary operators	Scalability limitations for very high-dimensional data
Djellal Serandi et al. [22]	2025	Clustering-Based Multi-objective DE	Cancer microarray datasets	Balanced trade-off between MSR and gene reduction for improved classification; guided mutation strategy.	Computationally expensive
Xue et al. [23]	2013	Multi-objective Particle Swarm Optimization	Gene expression microarray data	Good balance between accuracy and number of selected genes; maintains solution diversity	Risk of premature convergence; parameter-dependent

Continued on next page

Table 1.2 – continued from previous page

Author(s)	**Year**	**Method**	**Dataset / Application**	**Advantages**	**Limitations**
Ahadzadeh et al. [24]	2023	Statistical Filter Ensemble + PSO	High-dimensional datasets	Reduced search space; improved convergence speed	Filter stage may discard weak but relevant genes
Ludwig [25]	2025	Guided Particle Swarm Optimization	Large-scale genomic datasets	Early elimination of irrelevant features; reduced stagnation	Over-reliance on filter guidance; limited exploration
Paul et al. [26]	2021	Multi-objective PSO (MOPSO)	Multi-label classification datasets	Adaptive redundancy reduction; dynamic feature selection	Primarily designed for streaming data; limited microarray validation
Han et al. [27]	2021	Adaptive MOPSO (MOPSO-ASFS)	UCI and gene expression datasets	Adaptive diversity control; improved convergence stability	Increased algorithmic complexity
Ma et al. [28]	2021	Two-stage Ant Colony Optimization	High-dimensional datasets	Efficient subset size estimation; reduced computational cost	Sensitive to heuristic parameter design
Meenachi et al. [29]	2021	ACO, Genetic Algorithm, Tabu Search	Cancer microarray datasets	Strong global and local search capability	Single-objective formulation only

Continued on next page

Table 1.2 – continued from previous page

Author(s)	Year	Method	Dataset / Application	Advantages	Limitations
Yilmaz et al. [30]	2024	Adapted Ant Colony Optimization	High-dimensional classification problems	Improved convergence; reduced dependency on attribute grouping	Limited biological interpretability
Aziz et al. [32]	2022	ICA + Artificial Bee Colony + Crow Search	Cancer gene expression data	Significant gene reduction; improved Naïve Bayes performance	Multi-stage framework increases computational complexity
Wang et al. [33]	2022	Self-learning ABC (SLLABC)	High-dimensional datasets	Improved exploitation ability; faster convergence	May still get trapped in local optima
Xu et al. [34]	2025	Improved Artificial Bee Colony	Large-scale high-dimensional datasets	Enhanced population diversity; improved classification accuracy	High computational overhead
Balın et al. [37]	2019	Concrete Autoencoder	Gene expression datasets	Direct and differentiable feature selection; end-to-end learning	Training instability; reconstruction-driven objective
Wang et al. [38]	2022	Adaptive Sparse Autoencoder	Cancer microarray datasets	Redundancy control; sparse and stable feature selection	Lacks explicit global optimization

Continued on next page

Table 1.2 – continued from previous page

Author(s)	**Year**	**Method**	**Dataset / Application**	**Advantages**	**Limitations**
Li et al. [39]	2022	Graph-Regularized Autoencoder	Unsupervised gene selection	Preserves gene-gene relationships; robust latent structure	Graph construction complexity
Hassanieh et al. [40]	2024	Selective Deep Autoencoder	Cancer gene expression datasets	Explicit selection constraints; compact gene subsets	Architecture sensitive; high training cost
Li et al. [47]	2023	Multi-filter + RFPA	Tumor classification	Effective noise elimination; compact gene subsets	Sequential stages increase runtime
Xu et al. [51]	2024	Feature-Grouping Hybrid FS (FG-HFS)	Gene expression datasets	Grouped evolutionary search; redundancy reduction	Strongly dependent on clustering quality

1.5 Conclusion

Throughout this chapter, we have highlighted the importance of feature selection as a fundamental strategy for addressing the challenges posed by high-dimensional biological datasets. By examining the various categories of feature selection methods—namely filter, wrapper, and embedded approaches—we provided essential insights into how informative gene subsets can be identified while simultaneously reducing redundancy and noise.

The review of state-of-the-art approaches emphasized the dominance of bio-inspired metaheuristics. While techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC) have demonstrated promising results, Differential Evolution (DE) represents a distinct evolutionary paradigm that offers unique advantages in terms of simplicity, robustness, and efficiency. Despite its strong potential, our investigation reveals that DE remains underexplored in the specific context of microarray gene expression analysis.

Building on this background, the next chapter will introduce our first major contribution: a novel framework based on Binary Differential Evolution (BDE) for feature selection. By adapting the DE operators to discrete search spaces and integrating multi-objective optimization, the proposed approach aims to achieve an adaptive and robust gene selection process. This sets the stage for the subsequent development of more advanced clustering-guided and hybrid deep learning models, offering a comprehensive enhancement to current genomic data analysis methodologies.

After reviewing the different approaches to feature selection, it appears that evolutionary algorithms remain among the most widely adopted optimization strategies for handling high-dimensional and complex genomic data. Their ability to cope with noise and incomplete data, combined with flexibility, parallelism, and their non-deterministic search capability, make evolutionary algorithms particularly well suited for identifying relevant subsets of genes and achieving competitive performance across various benchmark datasets. Techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC) have demonstrated promising results in feature selection tasks. In contrast, Differential Evolution (DE) represents a distinct evolutionary optimization paradigm that offers additional advantages, including simplicity, robustness, efficiency, and versatility across continuous and discrete search spaces. Despite its strong potential, DE has been underexplored in the context of microarray gene expression feature selection, and our investigation suggests that, with appropriate adaptations, this algorithm can be effectively applied to FS for genomic data.

Chapter 2

Study of Feature Selection Methods on Gene Expression Data

2.1 Introduction

Gene expression datasets are widely used in bioinformatics to understand biological processes, identify disease biomarkers, and classify disease subtypes. However, these datasets often contain tens of thousands of genes (features) but only a limited number of samples, which poses challenges for machine learning algorithms due to the curse of dimensionality, noise, and redundancy.

Feature Selection (FS) is a crucial preprocessing step aimed at reducing the number of irrelevant or redundant genes while preserving the informative ones. By selecting a subset of meaningful genes, FS improves classifier performance, reduces computational cost, and enhances interpretability, which is essential in clinical applications.

This chapter focuses on the study of existing feature selection methods for gene expression data. It categorizes FS methods into classical statistical approaches, wrapper and embedded methods, and bio-inspired optimization techniques, including recent deep learning approaches. The goal is to provide a comprehensive overview that highlights strengths, weaknesses, and practical applicability.

2.2 Background

2.2.1 Characteristics of Gene Expression Data

Microarray data are pivotal for cancer detection at the transcriptomic level. In such datasets, only a small fraction of genes are typically relevant to a specific pathology.

Identifying this optimal subset among thousands of candidates is a combinatorial optimization challenge, classified as an NP-hard problem [52], which justifies the use of metaheuristic search strategies.

2.2.2 Curse of Dimensionality

High dimensionality negatively impacts distance-based models, such as KNN and clustering algorithms, by creating a sparse search space. In this environment, the contrast between distances diminishes until all points appear nearly equidistant. This lack of structure reduces statistical power and promotes overfitting; however, feature selection mitigates these issues by pruning the search space, thus enhancing both model reliability and interpretability [53].

2.2.3 Microarray feature selection main concept

The fundamental concepts of microarray feature selection can be categorized into three principal components, as illustrated in Figure 2.1.

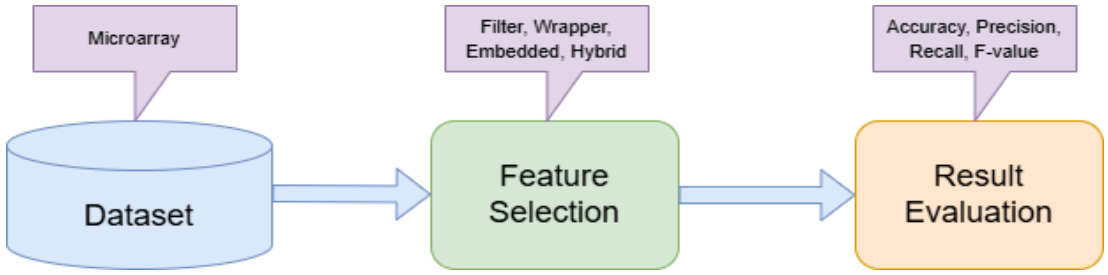

Figure 2.1: Microarray feature selection main concept.

2.2.4 Goals of Feature Selection

The primary objectives of applying feature selection to microarray gene expression data extend beyond mere data reduction; they aim to optimize the overall predictive and analytical quality of the model through the following goals:

- **Dimensionality Reduction:** This objective focuses on pruning the high-dimensional search space by filtering out noise and irrelevant genes. By retaining only the most informative variables, the model mitigates the ”curse of dimensionality” and simplifies the underlying data structure.
- **Classification Performance Enhancement:** A key goal of FS is to improve the discriminative power of the learning model. This is achieved by maximizing

metrics such as *accuracy*, *F-measure*, and *robustness*, ensuring the classifier generalizes well to unseen samples.

- **Biological Interpretability:** In clinical contexts, FS serves as a discovery tool. It identifies a compact set of genes that can act as potential *disease biomarkers*, allowing biologists to gain meaningful insights into the molecular mechanisms of specific pathologies.
- **Computational Efficiency:** By reducing the number of input features, FS significantly decreases memory requirements and execution time during both the training and inference phases, which is crucial for large-scale genomic studies.

2.3 Categories of Feature Selection Methods

2.3.1 Filter Methods

Filter-based feature selection methods operate by ranking variables according to their intrinsic statistical relevance to the target learning task. This process serves as an independent preprocessing step; the highest-ranked features are selected and subsequently provided as input to machine learning algorithms [54].

A defining characteristic of filter methods is that the evaluation is performed independently of any specific classifier. While these techniques often ignore complex dependencies among features, they offer significant advantages, including high computational efficiency and a lower risk of overfitting, particularly on high-dimensional datasets. Filter techniques typically rely on various statistical criteria—such as distance measures, information-theoretic metrics, or correlation-based scores—to evaluate the relationship between input features and output variables [55].

Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores.

Figure 2.2 provides an overview of the general principle and workflow of Filter feature selection methods.

Advantages:

- **Low Computational Complexity:** Filter methods are computationally inexpensive as they rely on direct statistical calculations rather than iterative model training, making them significantly faster than wrapper or embedded approaches.

Figure 2.2: Overview of Filter methods [54].

- **High Scalability:** Due to their efficiency, these methods scale exceptionally well to ultra-high-dimensional datasets, which is essential for processing thousands of genes in microarray analysis.
- **Independence from the Learning Algorithm:** Since the selection process is decoupled from the classifier, the resulting feature subsets are more general and less prone to overfitting specific model biases.

Limitations:

- **Inability to Capture Feature Interactions:** Most filter techniques are univariate, meaning they evaluate genes individually and fail to account for complex synergies or dependencies between groups of features.
- **Potential Selection of Redundant Features:** These methods may select multiple features that are highly correlated with the target class but also with each other, leading to a redundant and sub-optimal gene subset.

2.3.2 Wrapper Methods

Wrapper-based approaches integrate the feature selection process directly with the learning algorithm, treating the classifier as a ”black box” to evaluate the quality of feature subsets. Unlike filter methods, wrappers utilize the predictive performance of a specific model to guide the search, resulting in an iterative and adaptive selection process.

By employing sophisticated search techniques—such as recursive feature elimination, sequential selection strategies, and bio-inspired metaheuristics like Genetic Algorithms—these methods can capture complex dependencies between features. Consequently, they often achieve higher classification accuracy in real-world applications by identifying the most synergistic gene subsets. However, their reliance

on repeated model training and cross-validation procedures leads to a significant computational overhead. This high cost often limits their scalability, particularly when dealing with the high-dimensional nature of large-scale genomic datasets [56].

Figure 2.3 provides an overview of the general principle and workflow of wrapper feature selection methods.

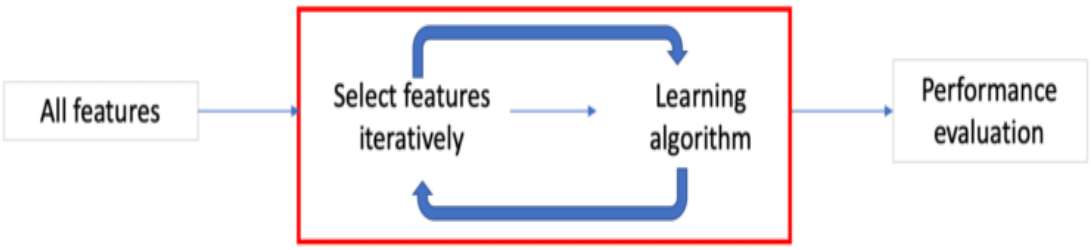

Figure 2.3: Overview of wrapper methods [54].

Advantages:

- **Capture of Feature Interactions:** Unlike univariate filters, wrappers evaluate feature subsets as a whole, allowing them to identify complex dependencies and synergistic relationships between genes that individually might appear insignificant.
- **Superior Predictive Accuracy:** By optimizing the feature subset specifically for a given learning algorithm, these methods generally yield higher classification accuracy and better-tuned models for specific biological contexts.

Limitations:

- **High Computational Cost:** The requirement to train and evaluate the learning model for every candidate subset makes wrappers computationally intensive, often requiring significant hardware resources.
- **Increased Risk of Overfitting:** Because the selection is tightly coupled with the classifier's performance, there is a high risk of "learning the noise" within the data, especially when dealing with the small sample sizes typical of microarray datasets.
- **Limited Scalability:** The combinatorial explosion of the search space in high-dimensional genomic data often makes exhaustive or complex wrapper searches intractable without the use of heuristic or metaheuristic optimization.

2.3.3 Embedded Methods

Embedded feature selection methods incorporate the selection process directly into the model training phase, effectively bridging the gap between filter and wrapper approaches. By performing feature selection and model construction simultaneously, these techniques treat selection as an integral part of the learning mechanism rather than an independent or external step.

A significant advantage of embedded methods is their ability to capture feature interactions and dependencies—a capability shared with wrappers—while maintaining higher computational efficiency and being generally less susceptible to overfitting. This is achieved because the selection criteria are built into the algorithm's objective function. Moreover, unlike filter-based methods that evaluate features independently, embedded and wrapper approaches rely on subset evaluation strategies capable of capturing complex synergies between genes, which is critical for genomic data analysis [55].

Figure 2.4 provides an overview of the general principle and workflow of embedded feature selection methods.

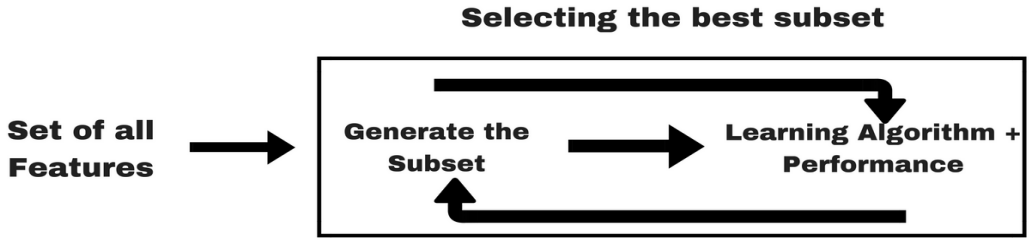

Figure 2.4: Overview of embedded methods [54].

Advantages:

- **Balanced Trade-off:** Embedded methods provide an optimal compromise between filter and wrapper approaches by capturing feature dependencies while maintaining much higher computational efficiency.
- **Reduced Computational Overhead:** Unlike wrapper-based methods that require costly external loops for subset evaluation, embedded techniques perform selection during a single training process, significantly lowering execution time.
- **Improved Robustness:** By incorporating regularization techniques directly into the objective function, these methods are generally less susceptible to overfitting than pure wrapper approaches.

Limitations:

- **Model Specificity:** There is a strong, intrinsic dependence on the underlying learning algorithm; the features selected are specifically optimized for that model's internal structure and may not generalize well to other classifiers.
- **Reduced Flexibility:** These methods lack the modularity of filters or wrappers, as they are inherently tied to specific algorithms (e.g., Lasso or Tree-based models) and cannot be easily adapted to different problem settings or architectures.

2.4 Bio-inspired and Metaheuristic Approaches

2.4.1 Evolutionary Algorithms (EA)

Evolutionary Algorithms (EA) constitute a class of nature-inspired stochastic optimization techniques that operate by maintaining a population of candidate solutions. These algorithms typically begin with a randomly generated population which evolves over successive generations through mechanisms such as selection, crossover, and mutation.

The most prominent representative of this group is the Genetic Algorithm (GA), originally proposed by John Holland in the 1960s and inspired by Darwin's theory of biological evolution. Due to its versatility, GA has attracted significant research interest, leading to numerous variants and its successful application across a wide range of real-world problems. Beyond GA, this category encompasses diverse methodologies including Genetic Programming, Evolution Strategies, and Differential Evolution (DE), as well as hybrid approaches like the Memetic Algorithm [57]. Other contemporary metaheuristics, such as the Flower Pollination Algorithm [58], also follow these evolutionary principles.

These methods are particularly well-suited for feature selection in high-dimensional genomic datasets because they offer:

- **Global Search Capabilities:** These algorithms possess the ability to efficiently explore vast, discontinuous, and high-dimensional search spaces. This is particularly crucial for gene expression data, where the number of possible feature combinations is exponentially large.
- **Evasion of Local Optima:** Through stochastic mechanisms like mutation and crossover, these methods maintain high population diversity. This ro-

bustness helps the search process avoid premature convergence, allowing it to bypass sub-optimal local basins in favor of the global optimum.

- **Resilience to Noise and Non-linearity:** EAs are highly capable of handling non-linear, non-convex, and noisy objective functions. Unlike traditional gradient-based methods, they do not require derivative information, making them ideal for the complex fitness landscapes of biological datasets.
- **Multi-objective Optimization Support:** They provide natural support for multi-objective frameworks. This allows researchers to simultaneously optimize conflicting goals, such as maximizing classification accuracy while minimizing the number of selected genes (subset size).

2.4.2 Swarm Intelligence (SI)

Algorithms in this category are inspired by the social interactions and collective behaviors observed in natural systems, such as bird flocking, fish schooling, and insect foraging. Over the past two decades, the field of Swarm Intelligence (SI) has seen a proliferation of metaheuristic algorithms, with novel methods continuing to emerge. Research in this domain generally focuses on two paths: the development of variants for established algorithms to improve their convergence properties, and the creation of hybrid approaches that combine multiple techniques to balance exploration and exploitation.

The most prominent representative of this group is Particle Swarm Optimization (PSO), originally developed by Kennedy and Eberhart in 1995. Inspired by the synchronized movement of bird flocks searching for optimal food sources, PSO models candidate solutions as "particles" that navigate the search space by following both their personal best experience and the collective experience of the swarm. Due to its strong mathematical foundation and computational efficiency, PSO has remained a cornerstone for solving complex, non-linear optimization problems.

Beyond PSO, other notable algorithms in this category have gained significant traction. These include Cuckoo Search, the Grey Wolf Optimizer, and the Krill Herd Algorithm [59], which mimics the herding behavior of krill individuals in response to environmental pressures. More recently, the Slime Mould Algorithm (SMA) [11], based on the oscillations and foraging patterns of Physarum polycephalum, has shown great promise in navigating complex search spaces. These swarm-based approaches are highly effective for feature selection as they maintain a critical balance between global exploration and local refinement through social information sharing.

Key Characteristics of SI Algorithms:

- **Decentralized and Distributed Control:** SI algorithms operate without a central authority; control is distributed across the entire population, enhancing system robustness and fault tolerance.
- **Simple Local Behaviors:** Individual agents follow localized rules based on their immediate surroundings and neighbor interactions, facilitating low-cost computational cycles.
- **Emergent Collective Intelligence:** Sophisticated global patterns and "intelligent" problem-solving behaviors emerge from the simple, local interactions of self-organized agents.
- **Balance of Exploration and Exploitation:** These methods effectively manage the trade-off between broad global search (exploration) and targeted local refinement (exploitation) to efficiently locate optimal feature subsets.

2.5 Deep Learning Approaches

2.5.1 Autoencoder-based FS

An autoencoder (AE) is a deep neural network (DNN) model designed to learn compact data representations by reconstructing its input at the output layer. It functions as an unsupervised learning algorithm, meaning that no labeled targets are required during training. Instead, the model encodes the input data into a latent representation within the hidden layers and then attempts to reconstruct the original input from this representation.

Autoencoders were first introduced by Rumelhart et al. (1985) [60] as a neural network architecture specifically trained to reconstruct its own input. The primary objective of an AE is to learn an "informative" and compressed representation of the data in an unsupervised manner.

By forcing the network to pass the information through a constrained "bottleneck" layer, the model is compelled to identify the most significant patterns within the data. These learned representations—often referred to as embeddings—are highly effective for downstream tasks, such as:

- **Clustering:** Grouping similar gene expression profiles based on their latent features to identify distinct biological subtypes.
- **Dimensionality Reduction:** Reducing thousands of genes into a few dozen representative components while preserving essential non-linear structures.

- **Data Denoising:** Filtering out stochastic noise inherent in microarray or RNA-seq experiments to recover the underlying biological signal.

As illustrated in Figure. 2.5, the architecture consists of three main components: the encoder, which maps the input x to the latent space w; the latent space w, which serves as a bottleneck; and the decoder, which reconstructs the output $\hat{x}$ from the latent representation.

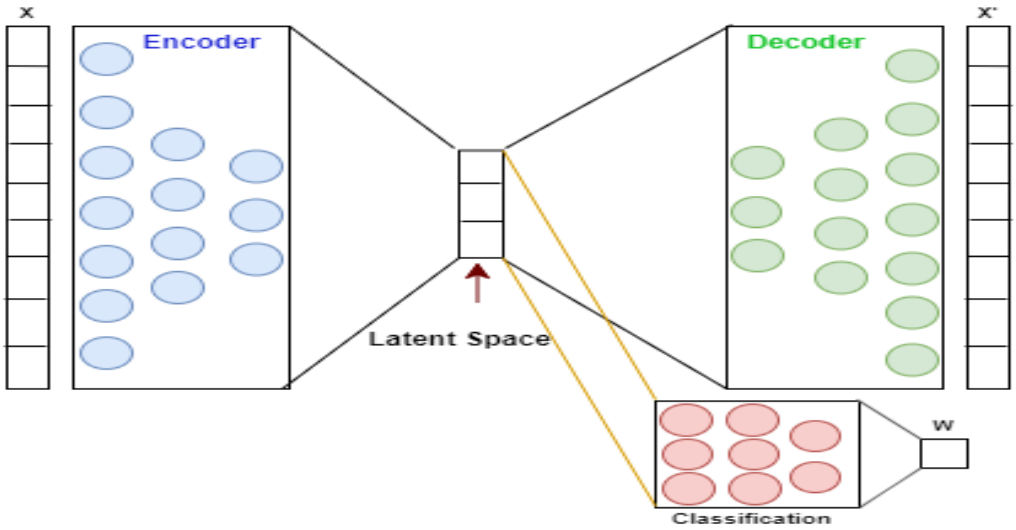

Figure 2.5: AE Architecture

The main components of an autoencoder (AE) can be described as follows:

- **Encoder:** The encoder processes the input data, which may consist of various forms of structured information, through a sequence of neural network layers equipped with nonlinear activation functions such as Rectified Linear Unit (ReLU) or Exponential Linear Unit (ELU). Its primary objective is to map the high-dimensional input space into a lower-dimensional latent representation that efficiently preserves the most relevant features.

- **Bottleneck / Latent Space:** This component represents the compressed knowledge of the original input data. It typically contains one or more layers with a reduced number of neurons compared to the input layer, thereby creating a bottleneck effect. This constraint forces the model to retain only the most informative features while discarding less significant ones.

- **Decoder:** The decoder reconstructs the original input from the compressed latent representation. Its architecture generally mirrors that of the encoder in reverse order and aims to produce an output that closely approximates the

original input, although perfect reconstruction may not always be achieved due to noise or other influencing factors.

The overall process can be summarized as:

Input → Encoding → Bottleneck / Latent Space → Decoding → Output.

During training, the AE learns optimal parameters by minimizing the reconstruction error between the input and the reconstructed output using an appropriate loss function. A key advantage of AE learning lies in its ability to jointly perform feature extraction and data compression, leading to efficient processing and the capture of complex patterns, particularly in high-dimensional datasets.

The different categories of feature selection methods and computational intelligence approaches discussed in this section are summarized in Figure. 2.6, which provides a unified overview of bio-inspired algorithms and deep learning-based techniques.

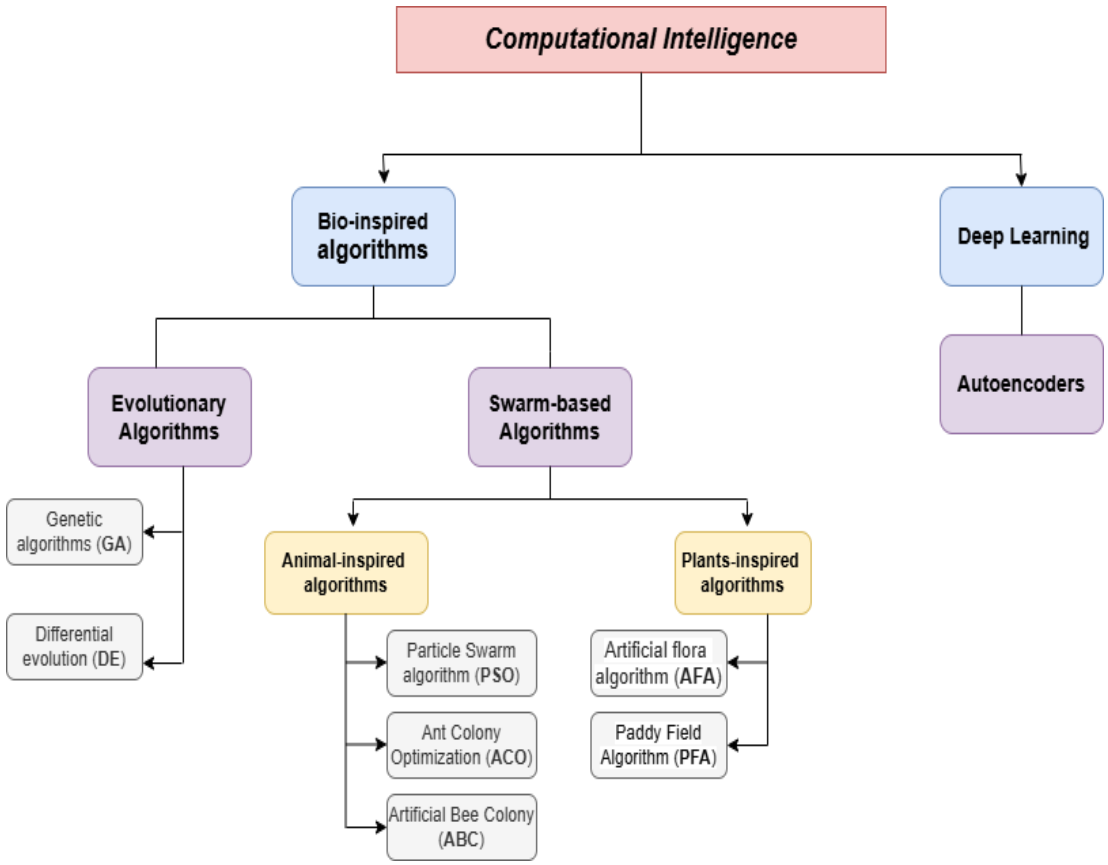

Figure 2.6: Taxonomy of Computational Intelligence Approaches for Feature Selection.

Once the optimal feature subset is identified using feature selection approaches,

machine learning classification models are commonly employed to evaluate the effectiveness of the selected features under supervised or unsupervised learning paradigms.

2.6 Machine Learning

Machine learning (ML) is a rapidly evolving field that focuses on enabling machines to learn from data and perform tasks that traditionally require human intelligence. By integrating principles from statistics and computer science, ML allows computational systems to automatically identify patterns, extract meaningful information, and make predictions or decisions with minimal human intervention. As a subfield of artificial intelligence, ML relies on a variety of models and algorithms trained on historical data to support intelligent applications. The increasing availability of large-scale data has further accelerated the development and adoption of ML techniques. Broadly, ML methods are classified into three main categories: supervised learning, unsupervised learning, and reinforcement learning [61].

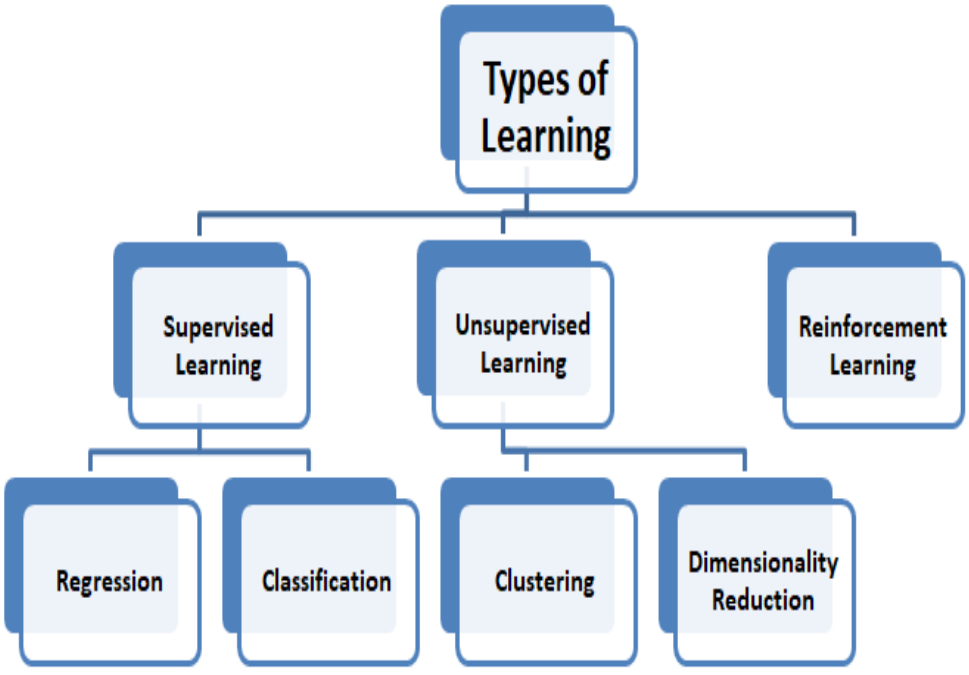

Figure 2.7: An Overview of Supervised, Unsupervised, and Reinforcement Learning in Machine Learning[62].

2.6.1 Supervised Learning

Machine learning represents each data instance through a set of features that can be binary, categorical, or continuous. When labeled data are available, supervised

learning is applied by training models on labeled instances and evaluating them on unseen data. This process typically involves data collection, dataset splitting, preprocessing, and model training to learn underlying patterns in the data [63].

2.6.1.1 Support vector machine (SVM)

Support Vector Machines (SVM) are a widely used method for classification tasks. They work by finding a hyperplane, or a set of hyperplanes, that maximizes the margin between classes in high-dimensional feature spaces. Originally designed for binary classification, SVMs have since been extended to handle multi-class problems through the use of multiple hyperplanes [64].

2.6.1.2 Naïve Bayes (NB)

Naïve Bayes is a classification-oriented machine learning approach that relies on conditional probability to determine the likelihood that an instance belongs to a particular class. Based on these probabilities, a network structure, often referred to as a Bayesian network, can be constructed to model the relationships between features and classes [65].

2.6.1.3 K-nearest neighbor (KNN)

The nearest neighbor algorithm, commonly known as K-nearest neighbor (KNN), is an instance-based learning method. Predictions for test instances are made by identifying the most similar objects in the training dataset. Consequently, the accuracy of the results generally improves with the size of the training dataset [66].

2.6.1.4 Decision trees (DT)

Decision tree learning involves constructing a model that maps observations to target values. Trees with discrete target values are called classification trees, where internal nodes represent features and leaf nodes represent class labels. Trees with continuous target values are known as regression trees. Therefore, decision trees can be applied to both classification and regression tasks [67].

2.6.1.5 Neural networks (NN)

The artificial neural network (ANN) model is inspired by the biological structure of neurons in the human brain. Like the brain, it consists of interconnected nodes (neurons) organized into an input layer, one or more hidden layers, and an output

layer. Each neuron in a layer is typically connected to every neuron in the subsequent layer, forming a fully connected network [68].

2.6.2 Unsupervised Learning

Unsupervised learning is a branch of machine learning in which models identify hidden structures or patterns within unlabeled data. The model is trained solely on the features of the unlabeled dataset and is then used to make inferences on new data. This type of learning is typically applied to clustering and dimensionality reduction tasks [69].

2.6.2.1 K-means clustering

Clustering algorithms group objects with similar features into distinct clusters, organizing data such that elements within the same cluster are more similar to each other than to those in other clusters. The K-means algorithm, a centroid-based method, partitions data into K clusters, positioning the cluster centroid at the mean of the points within that cluster [66].

2.6.2.2 Principal component analysis (PCA)

Principal Component Analysis (PCA) is a fast and easily implementable method for dimensionality reduction. It transforms high-dimensional data into a lower-dimensional space by retaining the most important features while discarding less informative ones. PCA emphasizes features that capture distinct information, thereby reducing data dimensionality, though its transformation is inherently linear [70].

2.6.3 Reinforcement Learning

Reinforcement learning is commonly applied in robotics, automation, gaming, navigation, and various other domains. It is a dynamic learning approach in which an agent learns to achieve a desired outcome without explicit guidance on proximity to the target. This paradigm is based on a trial-and-error strategy, selecting actions that maximize positive feedback and is considered a combination of supervised and unsupervised learning principles. The learning process begins with random actions, followed by feedback that reinforces beneficial behaviors. Reinforcement learning fundamentally relies on two concepts: exploration through trial and error, and delayed outcomes. In this framework, the agent generates a behavior based on its current state and input, which then leads to an action that modifies the environment [71].

2.6.4 Evaluation Metrics

The evaluation of the experimental results is carried out using commonly adopted performance metrics, namely precision, recall, F-measure, and accuracy. These metrics are computed based on the following quantities: true positives (TP), which denote correctly classified positive samples; false negatives (FN), representing positive samples that are misclassified; true negatives (TN), denoting correctly classified negative samples; and false positives (FP), representing negative samples that are misclassified.

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \tag{2.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100 \tag{2.2}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100 \tag{2.3}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{2.4}$$

2.7 Conclusion

This chapter introduced the essential concepts related to gene expression analysis and feature selection in high-dimensional microarray datasets. The main challenges associated with such data, including high dimensionality, redundancy, and noise, were discussed, highlighting the importance of feature selection for improving classification performance and interpretability.

Both classical feature selection methods and bio-inspired approaches were reviewed, with particular attention given to evolutionary algorithms, swarm intelligence techniques, and multi-objective optimization frameworks. These methods provide effective mechanisms for exploring complex search spaces and addressing the inherent trade-off between classification accuracy and the number of selected genes.

In the next chapter, a comprehensive and systematic review of existing feature selection approaches is presented, encompassing single-objective, multi-objective, and deep learning–based methods, including autoencoder-driven strategies. This review not only provides a detailed comparison of these methodologies in terms of optimization objectives, search mechanisms, and computational complexity, but also

critically analyzes their strengths and inherent limitations when applied to high-dimensional gene expression data. Through this comparative analysis, the chapter seeks to identify the most effective and scalable gene selection strategies, highlight unresolved challenges such as redundancy handling and small-sample bias, and ultimately establish a clear motivation and scientific foundation for the proposed methodology.

Chapter 3

Comparison of Feature Selection Approaches on Gene Expression Data

3.1 Introduction

The selection of informative genes from high-dimensional microarray datasets remains a critical step for improving the accuracy of disease classification and reducing computational complexity. Various feature selection approaches have been proposed, ranging from single-objective to multi-objective optimization, as well as bio-inspired algorithms and deep learning-based techniques.

This chapter presents a series of comparative studies integrating our previous works, aiming to analyze the effectiveness of different feature selection strategies. These studies illustrate the trade-offs between single-objective and multi-objective optimization, the role of autoencoder-based feature selection, and the performance of bio-inspired algorithms such as Slime Mould Algorithm (SMA) and Differential Evolution (DE).

The objective is to provide a systematic understanding of these approaches and to guide the selection of the most suitable methodology for gene selection in various contexts.

3.2 Multi-Objective vs Single-Objective Feature Selection (FsMo – FsSo)

3.2.1 Approaches

This study presents a comparative analysis of single-objective and multi-objective feature selection techniques for high-dimensional gene expression data. The Binary Differential Evolution (BDE) algorithm is employed, evaluating three approaches: the original dataset without feature selection, single-objective feature selection (FsSo), which optimizes the Mean Squared Residue (MSR) [72], and multi-objective feature selection (FsMo), which simultaneously optimizes both the MSR and the size of the selected feature subset. Experimental results show that the FsMo approach outperforms both FsSo and the original dataset in terms of classification accuracy, while providing a more compact and stable feature selection.

This work proposes two dimensionality reduction algorithms, FsMo and FsSo, based on Differential Evolution (DE). They follow the same principle except for the fitness function. FsMo is based on two objective functions, while FsSo uses a single objective function. Therefore, in FsMo, a non-dominated sorting genetic algorithm (NSGA-II) [73] is applied.

3.2.2 Illustration of FsMo and FsSo algorithms

FsMo and FsSo are based on a binary variant of the Differential Evolution algorithm (BDE). Since feature selection (FS) is inherently binary, population members in BDE are initialized as binary vectors. For a dataset with N features (number of genes), each solution in the DE algorithm consists of N components, where each component represents a feature, As shown in Figure 3.1, avalue of 1 indicates that a feature is selected, while 0 means it is not.

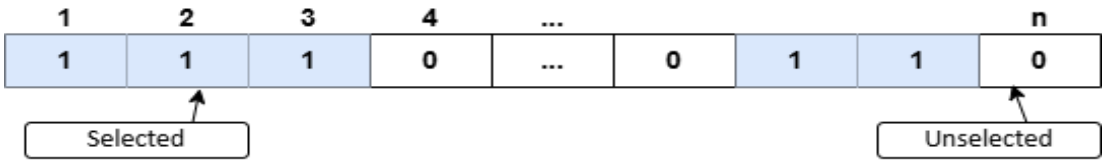

Figure 3.1: Representation of BDE for FS solution.

3.2.3 Dataset and Experimental Setup

The dataset is divided into training and testing subsets. The FsSo and FsMo feature selection methods are applied exclusively to the training set. The selected genes identified by the algorithms on the training set are then used to classify the samples in the testing set without rerunning the selection process. Figure 3.2 illustrates this experimental protocol, highlighting the workflow from data partitioning to feature selection and subsequent classification.

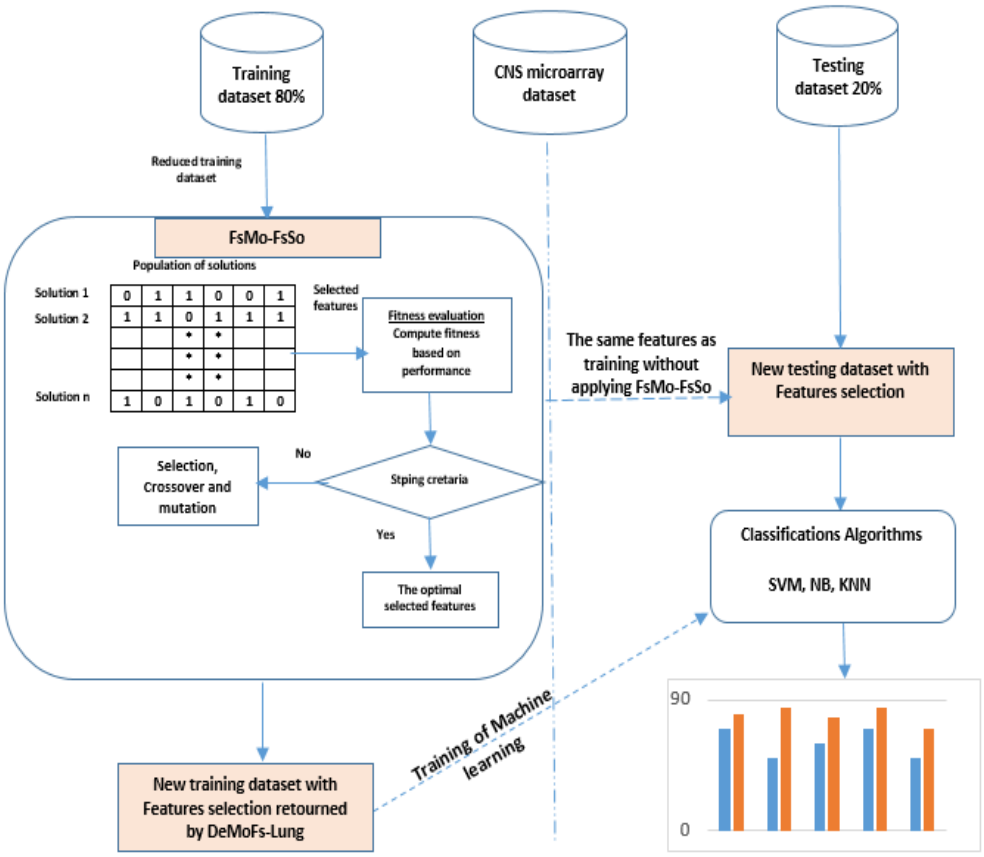

Figure 3.2: Illustration of FsMo and FsSo scheme.

The central nervous system (CNS) dataset contains high-dimensional gene expression profiles with relatively few samples. As illustrated in Table 3.1

Table 3.1: Description of the high-dimensional CNS microarray dataset

Dataset	#Features	#Samples	#Classes
CNS	7,129	60	2

The quality of the selected features was evaluated using SVM, KNN, and NB classifiers. The population size was set to 20 and the number of iterations to 50. The dataset was split into 80% for training and 20% for testing. After applying the

feature selection algorithms, most solutions achieved a reduction of approximately 50% of the original features.

3.2.4 Results and Discussion

Table 3.2: Comparison of classifiers using all features, FsSo, and FsMo in terms of accuracy (%)

Classifier	**All Features**	**FsSo**	**FsMo**
SVM	65.00	77.78	83.33
NB	61.67	83.33	88.89
KNN	61.67	66.67	83.33

The results shown in Table 3.2 and Figure 3.3 clearly demonstrate the effectiveness of the proposed feature selection methods. The multi-objective BDE (FsMo) consistently outperforms both the single-objective approach (FsSo) and the baseline using all features across all classifiers. Specifically, FsMo achieved the highest accuracy with SVM (83.33%), NB (88.89%), and KNN (83.33%), demonstrating its ability to enhance classification performance in high-dimensional gene expression data.

Although FsSo occasionally selected slightly smaller subsets of genes, it did not consistently translate into superior classification accuracy. In contrast, FsMo effectively balances the trade-off between minimizing the MSR and minimizing the number of selected features. This balance ensures that the selected gene subsets are both compact and informative, which is crucial for high-dimensional datasets such as CNS microarrays.

Overall, these findings highlight the advantages of multi-objective optimization in feature selection. FsMo provides more robust and stable feature selections across different classifiers, reducing the risk of overfitting while maintaining high predictive performance. This underscores the importance of considering multiple objectives simultaneously, rather than optimizing a single metric, when dealing with complex biological datasets.

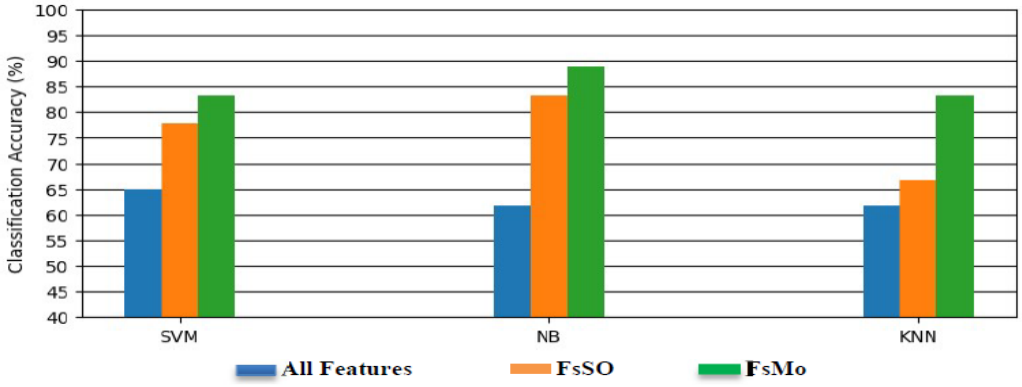

Figure 3.3: Accuracy Plot for FsMo and FsSo.

3.3 Multi-Objective BDE vs Autoencoder-Based Feature Selection (FsMo – FsAe)

3.3.1 Approaches

This study presents a comparative analysis of feature selection techniques for high-dimensional lung gene expression data. The Binary Differential Evolution (BDE) algorithm is employed to evaluate three configurations: the original dataset without feature selection, an autoencoder-based feature selection method (FsAe), which reduces dimensionality by learning latent representations followed by feature selection, and multi-objective feature selection (FsMo), which simultaneously optimizes the Mean Squared Residue (MSR) and the size of the selected feature subset [72]. Experimental results show that the FsMo approach consistently outperforms both FsAe and the original dataset in terms of classification accuracy, while providing a more compact and stable feature selection.

3.3.2 Illustration of FsMo and FsSo algorithms

For the FsMo approach, the dataset was divided into training and testing subsets, with 80% of the samples used for training and 20% for testing. The FsMo method was applied to the training set, and the selected genes were subsequently used to classify the test set without re-running the algorithm. To ensure a fair comparison, the autoencoder was configured to compress the dataset into a latent space of the same size as the feature subset returned by FsMo. This setup allows an equitable evaluation of both methods during the classification phase, as illustrated in Figure 3.4.

The performance of both FsMo and the autoencoder-based feature selection

(FsAe) was evaluated using real lung DNA microarray data illustrated in Table 3.3. Classification was performed on the datasets generated by the feature selection methods as well as on the original unfiltered dataset. The experimental results indicate that FsMo consistently outperforms FsAe and the original dataset in terms of classification accuracy, demonstrating the advantages of multi-objective optimization in high-dimensional spaces.

Furthermore, the methods were compared qualitatively in terms of methodology, computational complexity, memory usage, interpretability, and hyperparameter tuning. FsMo, based on a multi-objective Binary Differential Evolution (BDE) algorithm, offers higher interpretability and allows direct control over the number of selected features, while FsAe relies on a learned latent representation through an autoencoder, which can require more memory and network tuning but captures nonlinear feature interactions effectively. This comparison highlights the trade-offs between evolutionary multi-objective optimization and representation learning-based feature selection.

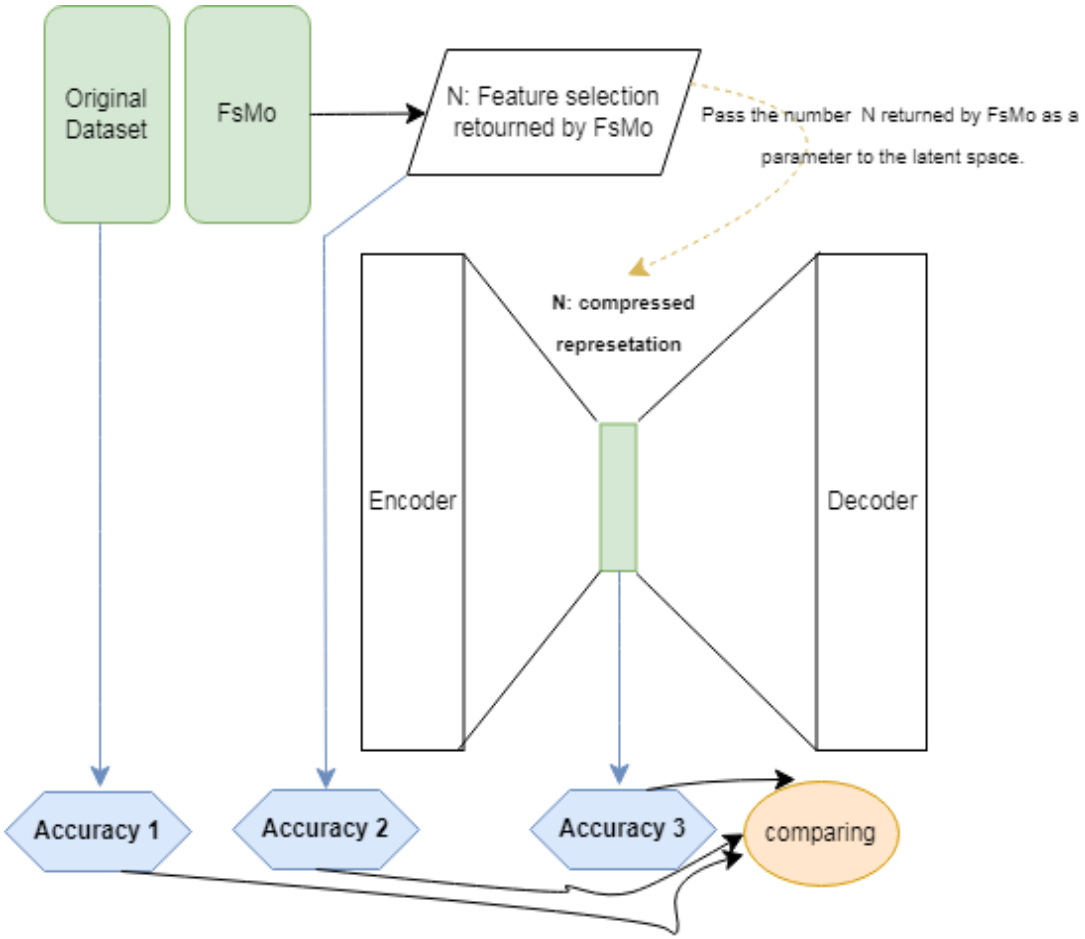

Figure 3.4: Illustration of FsMo and autoencoders FsAe.

3.3.3 Dataset and Experimental Setup

Table 3.3: Description of the high-dimensional Lung microarray dataset

Dataset	#Features	#Samples	#Classes
Lung	12,600	203	5

The Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbors (KNN) classifiers were used to evaluate the quality of the selected features. The population was fixed at 20 and the number of iterations at 50. After applying the FsMo algorithm, most of the solutions reduced the dataset by approximately 50%.

The multi-objective feature selection method (FsMo) resulted in a subset of 6,287 genes, corresponding to an approximate reduction of 50.1% from the original dataset. This subset size was subsequently used to define the dimensionality of the latent space for the autoencoder, ensuring a fair comparison between the two approaches. Given that optimizing an autoencoder requires empirical tuning, several architectures were evaluated, and the configuration presented in Table 2 was selected as it provided the best performance in terms of reconstruction accuracy and feature representation.

Table 3.4: Summary of the Autoencoder architecture with input size 12,600 and latent dimension 6,287

Layer (type)	Output Shape	Parameters
InputLayer (input layer)	(None, 12600)	0
Dense (dense)	(None, 8192)	103,227,392
Dropout (dropout)	(None, 8192)	0
Dense (dense 1)	(None, 6400)	52,435,200
Dropout (dropout 1)	(None, 6400)	0
Dense (dense 2)	(None, 6287)	40,243,087
Dense (dense 3)	(None, 6400)	40,243,200
Dense (dense 4)	(None, 8192)	52,436,992
Dense (dense 5)	(None, 12600)	103,231,800
Total parameters		391,817,671 (1.46 GB)
Trainable parameters		391,817,671 (1.46 GB)
Non-trainable parameters		0 (0.00 B)

Explanation of Column 3 (Param #). The values in Column 3 correspond to the total number of trainable parameters (weights and biases) for each layer of the autoencoder. For a Dense layer, the parameter count is computed as:

$$\text{Parameters} = (\text{Number of input units} \times \text{Number of output units}) + \text{Number of output units (biases)}.$$

For example, the first Dense layer has 12,600 input units and 8,192 output units:

$$(12,600 \times 8,192) + 8,192 = 103,227,392,$$

which matches the value reported in Table 3.4. Dropout layers do not contain any parameters, hence their count is 0. The total number of parameters at the bottom (391,817,671 $\approx$ 1.46 GB) represents the sum of all weights and biases across the network, reflecting the model's overall size and complexity.

3.3.4 Results and Discussion

A classification phase was subsequently conducted on the reduced datasets in order to compare the predictive performance of the models before and after the feature selection and dimensionality reduction processes, namely using all features, FsAe, and FsMo. The comparative results are reported in Table 3.5 and illustrated in Figure 3.5.

The results clearly demonstrate that applying feature selection significantly enhances classification performance. In particular, the SVM classifier exhibits a substantial improvement in accuracy, increasing from 78.82% when using all features to 92.68% with the autoencoder-based approach (FsAe), and reaching 97.56% with the multi-objective feature selection method (FsMo). This highlights the effectiveness of FsMo in identifying highly discriminative gene subsets.

For the Naïve Bayes (NB) classifier, the FsAe approach leads to a decrease in accuracy (82.92%) compared to the original dataset (90.15%), suggesting that the latent representations learned by the autoencoder may not fully preserve class-relevant probabilistic information. In contrast, FsMo yields a marked improvement, achieving an accuracy of 96.72%, which confirms the robustness of multi-objective optimization in preserving discriminative features.

Similarly, the K-Nearest Neighbors (KNN) classifier shows consistent gains across both feature selection methods. The accuracy increases from 92.61% with all features to 95.12% using FsAe, and further to 96.72% with FsMo. These results indicate that reducing feature redundancy while maintaining neighborhood structure benefits distance-based classifiers.

Overall, FsMo consistently achieves the highest accuracy across all classifiers, demonstrating its ability to effectively balance feature reduction and classification

performance. These findings confirm the superiority of multi-objective feature selection over both the original high-dimensional representation and autoencoder-based dimensionality reduction for lung gene expression data.

Table 3.5: Comparison of classifiers using all features, FsAe, and FsMo in terms of accuracy (%)

Classifier	**All Features**	**FsAe**	**FsMo**
SVM	78.82	92.68	97.56
NB	90.15	82.92	96.72
KNN	92.61	95.12	96.72

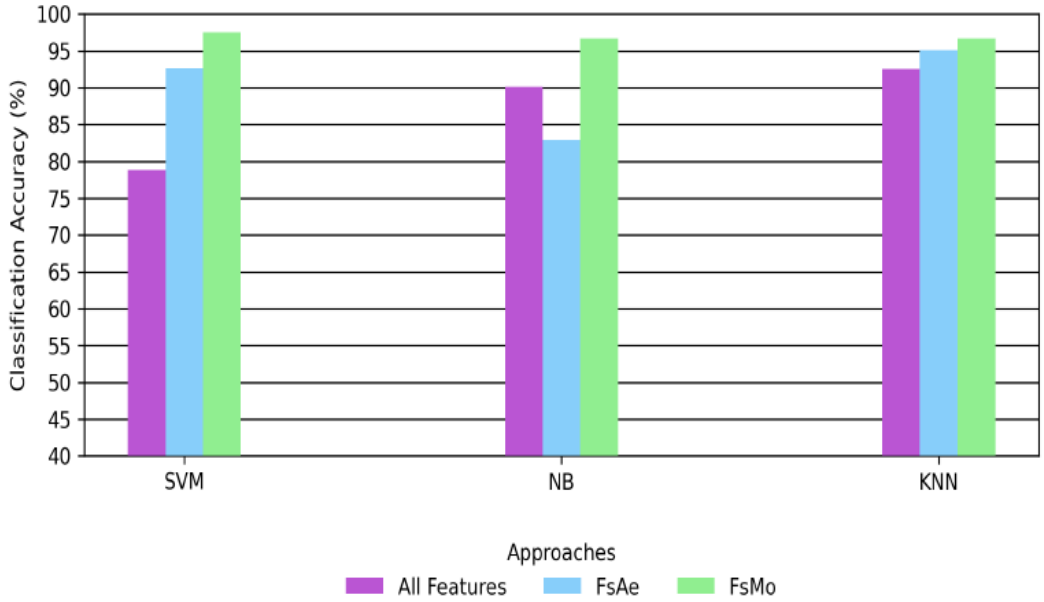

Figure 3.5: Accuracy Plot for FsMo and autoencoders FsAe.

A comprehensive comparison between the autoencoder-based feature selection method (FsAe) and the multi-objective feature selection approach (FsMo) is provided in Table 3.6. This comparison examines key aspects including methodological principles, computational complexity, memory requirements, interpretability, and hyperparameter sensitivity, thereby offering a clear insight into the strengths and limitations of each approach.

Table 3.6: Comparison between FsAe and FsMo

Criterion	**FsAe**	**FsMo**
Method Type	Deep learning (neural network–based)	Evolutionary algorithm (stochastic search)
Execution Complexity	Moderate: $O(n \times d \times$ epochs), where n is the number of samples and d the number of genes	High: $O($generations $\times$ population size $\times$ evaluation time)
Memory Usage	Medium to high (due to neural network parameters)	Low to medium
Interpretability	Low (black-box model)	Medium to high (explicit feature subsets)
Hyperparameters	Network architecture, learning rate, dropout, etc.	Population size, mutation and crossover rates, number of generations

3.4 Slime Mould Algorithm vs Differential Evolution (FsSMA – FsDE)

3.4.1 Approaches

In this study, a comparative analysis is conducted to evaluate the performance of two evolutionary feature selection methods, namely the Slime Mould Algorithm (FsSMA) and Differential Evolution (FsDE), on the Lymphoma gene expression dataset. FsSMA is characterized by a strong exploration capability that enables an extensive search of the high-dimensional feature space, whereas FsDE emphasizes exploitation through efficient refinement of candidate solutions. This complementary behavior makes their comparison particularly relevant for understanding the trade-off between exploration and exploitation in gene selection problems. Experimental results highlight the effectiveness of both approaches in reducing dimensionality and improving classification performance, while revealing distinct strengths that can be leveraged for the design of hybrid feature selection strategies.

3.4.2 Slime Mould Algorithm (SMA)

The Slime Mould Algorithm (SMA) is a stochastic metaheuristic optimization technique introduced in 2020, inspired by the oscillatory foraging behavior observed in slime mould organisms in nature [11]. As a relatively recent optimization approach, its potential has so far been explored in a limited number of studies. SMA mimics

the dynamic oscillation patterns exhibited by slime moulds during food searching, using adaptive weight mechanisms to effectively balance exploration and exploitation throughout the optimization process. This balance is achieved through three fundamental phases: approaching the food source, wrapping around it, and oscillating in its vicinity to refine candidate solutions. The overall workflow of the SMA is illustrated in Figure 2 and formally described in Algorithm 2, providing a clear overview of its operational principles.

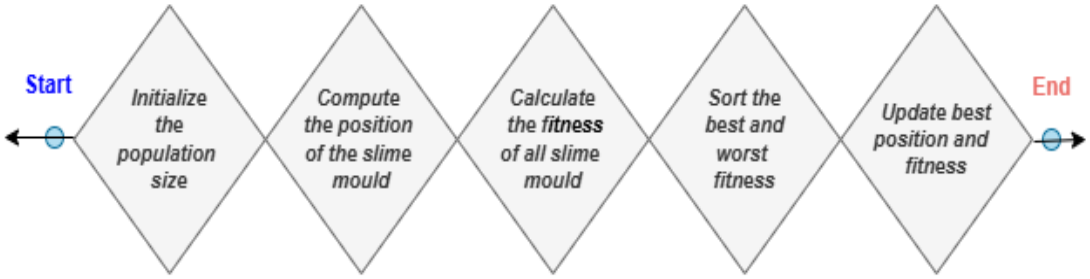

Figure 3.6: Phases of SMA.

$$\mathbf{x}^* = \begin{cases} \text{rand}(U_b - L_b) + L_b, & \text{if rand} < 0.03, \\ \mathbf{x}_b(t) + \mathbf{v}_b\big(W\mathbf{x}_A(t) - \mathbf{x}_B(t)\big), & \text{if } r < p, \\ \mathbf{v}_c\,\mathbf{x}(t), & \text{if } r \geq p. \end{cases} \tag{3.1}$$

$$\begin{cases} \overrightarrow{W(\text{SmellIndex}(i))} = \begin{cases} 1 + r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & \text{if condition}, \\ 1 - r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & \text{otherwise}. \end{cases} \\ \text{SmellIndex} = \text{sort}(S), \\ \overrightarrow{v_b} = [-a, a], \\ a = \text{arctanh}\left(-\left(\dfrac{t}{\text{Max\_t}}\right) + 1\right), \\ p = \tanh\big|S(i) - DF\big| \end{cases} \tag{3.2}$$

where $\vec{v}_b$ denotes the vibration parameter and t represents the current iteration. X denotes the position of the slime mould, while W represents its adaptive weight. DF refers to the best fitness value obtained across all iterations. $S(i)$ ranks the top half of the population according to their fitness values, and r is a random number

uniformly distributed in the interval $[0, 1]$. The terms bF and wF correspond to the best and worst fitness values, respectively. *SmellIndex* denotes the sequence of fitness values sorted in ascending order, as required for minimization problems. Lb and Ub represent the lower and upper bounds of the search space, respectively, and both *rand* and r are uniformly distributed in $[0, 1]$. The parameters $\vec{v_b}$ and W jointly contribute to balancing the exploration and exploitation capabilities of the algorithm.

Algorithm 1 Slime Mould Algorithm (SMA)

Input: Population size n, maximum iterations Max_iter
Output: Best solution X_b

Begin
Initialize slime mould positions X_i, $i = 1, 2, \ldots, n$;
Set iteration counter $t \leftarrow 1$;

while $t \leq Max\_iter$ **do**
  Evaluate fitness of all individuals;
  Identify the best fitness X_b;
  Compute weights W using Eq. (3.1);
  for each individual i in the population **do**
    Update parameters p, vb, and vc;
    Update positions using Eq. (3.2);
  end for
  $t \leftarrow t + 1$;
end while

Return the best solution X_b;

End

3.4.3 Binary representation of features

In feature selection tasks, where each candidate solution represents a subset of selected features, a binary representation is essential. Each solution is encoded as a binary string, in which a value of '1' indicates a selected feature, while '0' denotes a discarded one. However, classical Differential Evolution (DE) and the Slime Mould Algorithm (SMA) are originally designed for continuous optimization problems and therefore require adaptation to operate in binary search spaces.

This limitation has led to the development of Binary Differential Evolution (BDE) [19] and the Binary Slime Mould Algorithm (BSMA) [3]. These binary variants rely on transfer functions, such as sigmoid or V-shaped functions, to map

continuous position updates into probabilistic binary decisions. Through this transformation, the algorithms are able to effectively explore and exploit the discrete solution space associated with feature selection.

Within the proposed hybrid SMA–DE framework, the integration of BDE and BSMA enables a complementary balance between exploration and exploitation. While BSMA promotes diversity by exploring a wide range of feature subsets, BDE focuses on refining promising solutions through effective exploitation. This synergy enhances the algorithm's capability to identify compact and informative subsets of features from high-dimensional gene expression data. Moreover, the adopted binary coding scheme provides a clear and concise representation of selected features, facilitating both interpretability and computational efficiency.

Illustrative Example of Binary Representation

An illustrative example of feature selection using a binary representation is provided below. Consider a dataset represented by the matrix $M(4, 4)$:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

Suppose the binary feature selection vector is

$$\mathbf{x} = (1, 0, 1, 0),$$

where each element corresponds to a column (feature) of M. Applying $\mathbf{x}$ to M selects only the columns for which $x_i = 1$, namely G_1 and G_3, and discards the columns where $x_i = 0$ (G_2 and G_4).

The resulting feature-selected matrix M_{FS} is then:

$$M_{\mathrm{FS}} = \begin{bmatrix} m_{11} & m_{13} \\ m_{21} & m_{23} \\ m_{31} & m_{33} \\ m_{41} & m_{43} \end{bmatrix}.$$

This example illustrates how the binary vector effectively encodes the selection of informative features while discarding irrelevant ones.

3.4.4 Illustration of FsSMA and FsDE algorithms

As illustrated in Figure 3.7, the dataset is initially preprocessed and divided into training and testing sets. The training set is subsequently used to apply the proposed FsSMA and FsDE algorithms, which identify an optimal subset of features, thereby generating a reduced training subset. The same selected features are then extracted from the test set to construct the corresponding reduced testing subset. These feature-selected training and testing subsets are finally provided to various classifiers to comprehensively evaluate the effectiveness and generalization capability of the selected features. This pipeline ensures a consistent and fair assessment of the performance improvements achieved by FsSMA and FsDE compared to the original dataset.

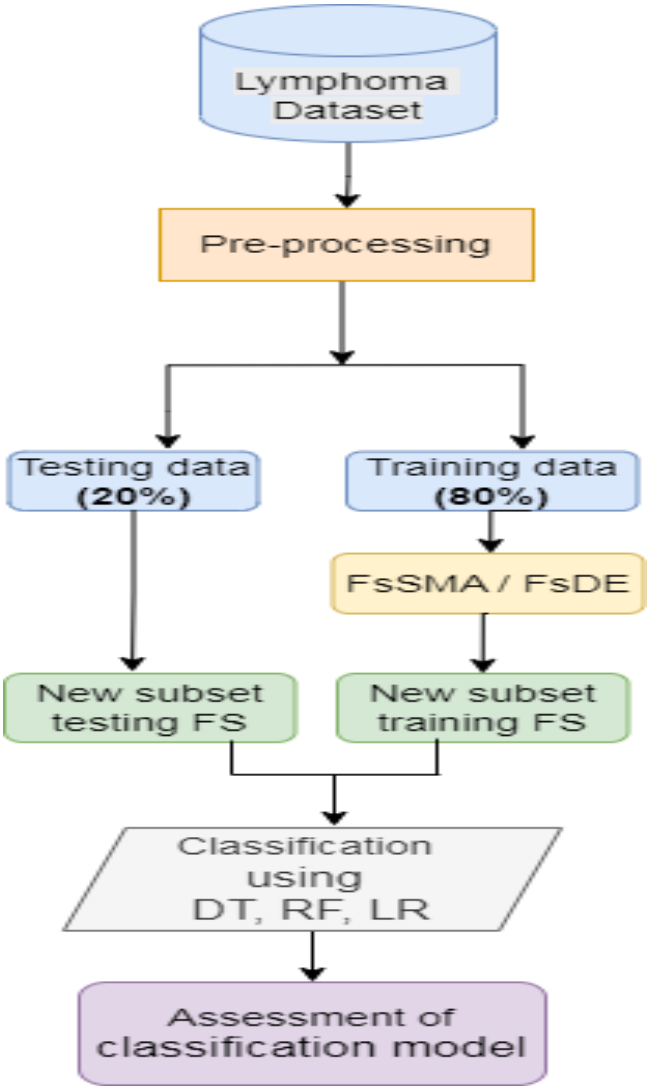

Figure 3.7: Illustration of FsSMA and FsDE scheme.

3.4.5 Dataset and Experimental Setup

Table 3.7: Description of the high-dimensional Lymphoma microarray dataset used in this study

Dataset	#Features	#Samples	#Classes
Lymphoma	4,027	66	3

The Lymphoma microarray dataset is publicly available from several repositories, including Kaggle, UCI, Mendeley Data, and the Global Health Observatory Data Repository. In this study, the dataset was obtained from Mendeley Data [74] in its high-dimensional biomedical format. It consists of 66 samples characterized by 4,027 gene expression features, which are categorized into three classes: 46 samples corresponding to diffuse large B-cell lymphoma (DLBCL), 11 samples to chronic lymphocytic leukemia (CLL), and 9 samples to follicular lymphoma (FL).

The FsSMA and FsDE algorithms were employed to select optimal subsets of features. The experimental results obtained on the Lymphoma dataset are presented in Table 3.7. The classification algorithms applied were Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). As shown in Table 3.8, the accuracy achieved using the FsSMA algorithm surpasses that obtained with FsDE and without feature selection.

3.4.6 Results and Discussion

The results presented in Table 3.8 and Figure 3.8 clearly highlight the effectiveness of the FsSMA algorithm on the high-dimensional Lymphoma dataset. Across all three classifiers Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). FsSMA consistently achieves superior accuracy compared to both the baseline (without feature selection) and the FsDE approach.

Specifically, for DT, FsSMA provides a modest improvement over FsDE, reaching 91% versus 90%, indicating a better selection of informative features. More strikingly, FsSMA achieves perfect classification accuracy (100%) for both RF and LR, outperforming FsDE (93.47% and 100%, respectively) and demonstrating its robustness in capturing the most relevant gene subsets.

These outcomes suggest that FsSMA excels at balancing exploration and exploitation in the search space, effectively identifying compact and informative feature subsets that enhance model performance. Overall, the results confirm that FsSMA is a highly reliable and efficient feature selection strategy for high-dimensional gene

expression datasets, capable of improving predictive accuracy across diverse classification models.

Table 3.8: Classification accuracy for the Lymphoma dataset (%)

Dataset	Algorithm	Without FS	FsDE	FsSMA
Lymphoma	DT	85.71	90.00	91.00
	RF	92.85	93.47	100.00
	LR	82.42	100.00	100.00

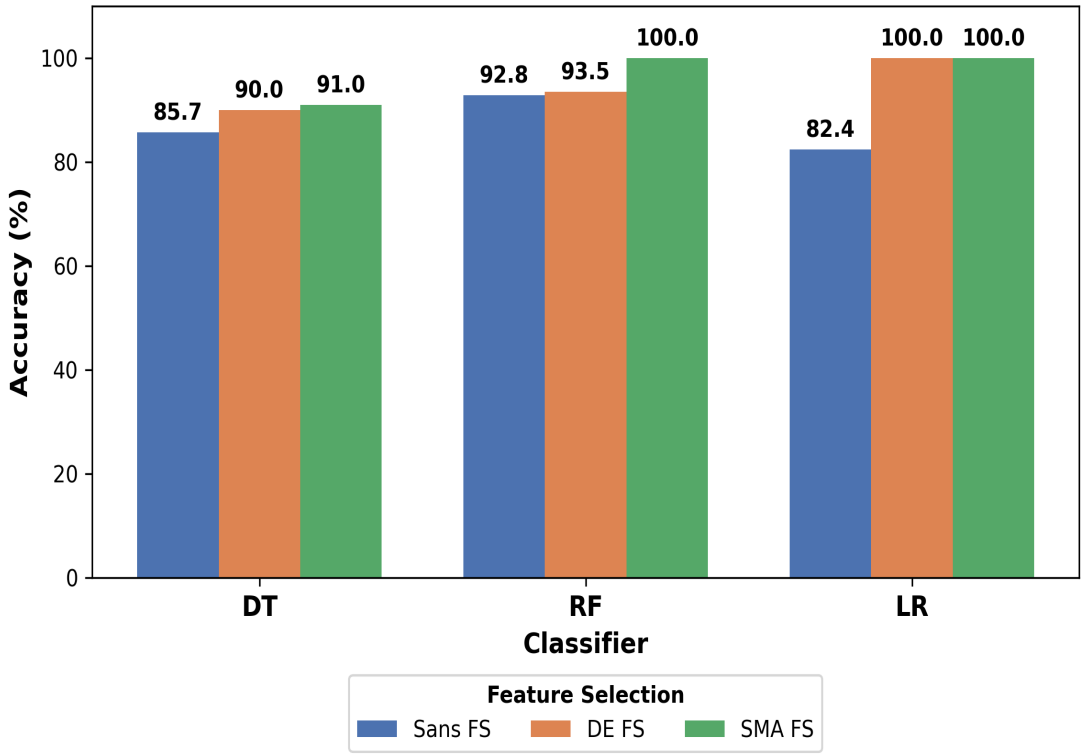

Figure 3.8: Lymphoma Dataset Accuracy.

3.5 Summary and Insights

The three comparative studies presented in this chapter provide several insights:

- Single-objective approaches serve as useful baselines but often fail to balance accuracy and feature subset size.
- Multi-objective optimization offers a robust framework for jointly optimizing classification performance and feature reduction.

- Deep learning-based approaches such as autoencoders capture complex non-linear patterns, enhancing feature selection, particularly in high-dimensional datasets.
- Bio-inspired algorithms like SMA and DE demonstrate complementary exploration and exploitation capabilities, suggesting potential benefits from hybrid strategies.

These studies underscore the necessity of systematic comparisons to identify the most suitable feature selection approach. The next chapter will present further experiments and comparative analyses across multiple datasets to determine the best-performing methodology in terms of accuracy, stability, and computational efficiency.

Chapter 4

DeFs-CBDE: Clustering-guided binary mutation in multi-objective differential evolution for microarray gene selection

4.1 Introduction

This chapter explores a dedicated optimization framework for microarray gene selection, termed DeFs-CBDE: Clustering-guided binary mutation in multi-objective differential evolution for microarray gene selection, which is built upon Multi-Objective Differential Evolution (MODE). Recognizing the importance of MODE in handling conflicting objectives inherent to feature selection, this chapter presents its core principles and adaptation to high-dimensional gene expression data, which simultaneously optimizes the MSR and the size of the selected feature subset.

In particular, the chapter focuses on binary Differential Evolution as the foundation of the proposed method, enabling effective modeling of gene selection problems. Within this framework, a novel clustering-guided binary mutation operator, referred to as CBDE, is introduced to enhance search efficiency, population diversity, and convergence toward high-quality gene subsets.

Finally, the effectiveness of DeFs-CBDE is systematically evaluated through comprehensive comparisons with state-of-the-art feature selection approaches on multiple benchmark microarray datasets.

4.2 Review of differential evolution

In recent years, Differential Evolution has emerged as a widely used method for continuous optimization, demonstrating effective applications in numerous scientific and engineering domains.

Differential Evolution's primary strength is its capacity to create new candidate solutions by exploiting directional information present within the population. Proposed by Storn and Price [75], DE is a straightforward and efficient heuristic for global optimization over continuous domains, and their study highlighted its capability in tackling diverse optimization challenges.

In a comparative study conducted by [76], Differential Evolution (DE) was assessed alongside Genetic Algorithms (GAs) for solving combinatorial optimization problems. DE was rigorously evaluated based on convergence speed, solution quality, and computational efficiency, demonstrating superior robustness and yielding more reliable solutions than GAs. In a similar study, [77] compared DE with Particle Swarm Optimization (PSO) for constrained optimization tasks. The results highlighted that DE not only exhibited greater repeatability but also required fewer iterations to ensure candidate solutions remained within the feasible search space.

Differential Evolution (DE) is a widely recognized optimization strategy for continuous, real-parameter numerical problems. Despite its simplicity, it offers powerful capabilities for global search by employing a greedy selection criterion, which ensures that only the fittest candidate solutions are carried forward to the next generation.

In contrast to conventional Genetic Algorithms, Differential Evolution (DE) exploits distance and directional information encoded in unit vectors to guide reproduction. The reproduction process begins with a mutation step that generates a trial vector, which is then combined with the parent vector through a crossover operation to produce a single offspring. The magnitude of each mutation is computed as a weighted difference between randomly selected population members. A detailed explanation of the main algorithmic steps of DE is provided, as illustrated in Figure 4.1.

4.2.1 Algorithm of Differential Evolution

A candidate solution in Differential Evolution is represented by a vector $\mathbf{x} \in \mathbb{R}^D$, where D denotes the dimensionality of the optimization problem. The objective function to be minimized is defined as $f : \mathbb{R}^D \to \mathbb{R}$. The standard Differential Evolution algorithm follows the *DE/rand/1/bin* strategy, whose pseudocode is presented below [18]. The DE algorithm consists of five main steps: initialization, mutation,

crossover, selection, and termination [7](cf. Figure 4.1).

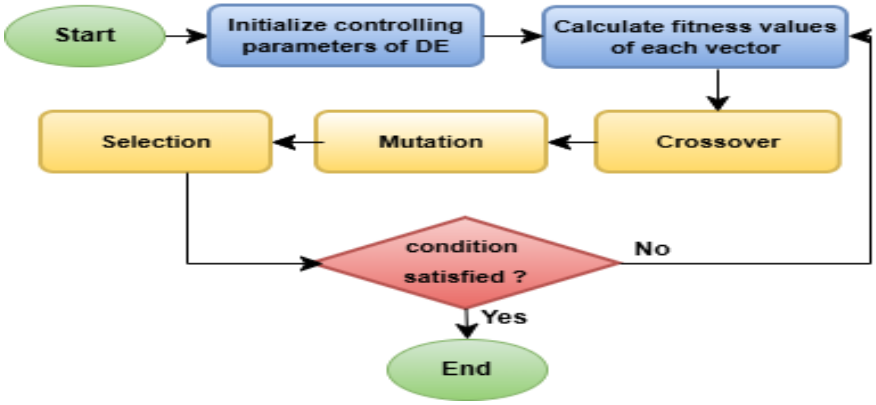

Figure 4.1: Phases of differential evolution

1. **Initialization:** Differential Evolution (DE) begins by generating an initial population of candidate solutions, with the population size determined by the user.

2. **Mutation:** In this step, new candidate solutions are generated by perturbing existing ones. Specifically, the difference between two randomly selected individuals is calculated, scaled by a factor, and added to a third individual, resulting in a modified solution.

3. **Crossover:** Components from the mutated solution are randomly combined with components from the original solution to create a trial solution, ensuring diversity in the population.

4. **Selection:** The trial solution is then compared to the original individual. If the trial solution demonstrates better performance, it replaces the original individual in the population.

5. **Termination:** These steps are repeated for a predetermined number of generations or until a convergence criterion is met.

Georgioudakis and Plevris [18] conducted a comparative study on various Differential Evolution (DE) variants for constrained structural optimization. Their work provides valuable insights into the effectiveness of DE in addressing complex optimization problems and complements the investigation of the standard DE approach presented in this study. The standard DE method, based on the DE/rand/1/bin scheme, is summarized in the following algorithm:

Differential Evolution (DE) is a widely recognized optimization technique, valued for its capability to address non-convex, multi-modal, and nonlinear problems while providing rapid convergence, effective local search, and implementation simplicity. Its versatility and simplicity, combined with the ability to explore large solution spaces, make DE particularly suitable for complex, high-dimensional optimization tasks involving both continuous and discrete variables. Additionally, DE mitigates the risk of becoming trapped in local optima by maintaining a population of candidate solutions, which enhances its convergence behavior. The algorithm can achieve faster convergence with fewer evaluations when control parameters are carefully chosen, reinforcing its utility as an optimization strategy. Nevertheless, DE exhibits certain limitations, including sensitivity to parameter settings, challenges in adapting to dynamic environments, and reduced performance on noisy or irregular landscapes. Moreover, DE faces difficulties in maintaining scalability in higher-dimensional spaces; as the number of dimensions increases, its efficiency and performance may deteriorate, complicating the exploration of complex solution spaces [78].

4.2.2 Mutation operator

In Differential Evolution, the mutation operator generates both a difference vector and a target vector using selected components. The target vector represents the parent whose direction is prioritized in forming the unit vector, while the difference vector captures the differences between two or more parents. The result of the mutation process is the unit vector, which is subsequently applied to the current central parent—the parent under consideration but excluded from the mutation—to create a trial vector for the crossover operation. Formally, for each target vector x_i $(i = 1, 2, \ldots, m)$, a mutant vector is generated as follows:

$$h_i(i+1) = x_{r1} + F\,(x_{r2} - x_{r3}) \tag{4.1}$$

4.2.3 Crossover operator

After the mutation phase, the resulting mutant vector undergoes a crossover with the target vector $X_i(g)$ to generate the trial vector $U_i(t+1)$, a step that has received comparatively less attention in the literature. The crossover operator is as crucial as the mutation operator in Differential Evolution. Two main types of crossover are commonly employed: binomial crossover and exponential crossover.

4.2.3.1 Binomial crossover

The binomial crossover is a uniform crossover scheme based on a binomial distribution. The conditional statement in the algorithm ensures that at least one component of the trial vector is inherited from the mutant vector.

Algorithm 2 Binomial Crossover

```
1: function BinomialCr(x_i, y_i)
2:     j ← rand_i({1, ..., n})
3:     for i ← 1 to n do
4:         if rand_i(0, 1) ≤ CR or i = j then
5:             u_i ← h_i
6:         else
7:             u_i ← x_i
8:         end if
9:     end for
10:    return u_i
11: end function
```

4.2.3.2 Exponential crossover

The exponential crossover mimics a two-point crossover in which the first cut point is selected randomly from the set $\{0, \ldots, n\}$, and the second cut point is determined such that L consecutive components are inherited from the mutant vector.

Algorithm 3 Exponential Crossover

```
1: function ExponentialCr(x_i, h_i)
2:     j ← rand_i({1, ..., n})
3:     i ← j
4:     u_i ← x_i
5:     L ← 0
6:     repeat
7:         u_i ← h_i
8:         i ← i + 1
9:         L ← L + 1
10:    until rand_i(0, 1) ≥ CR or L = n
11:    return u_i
12: end function
```

Zaharie [79] investigated the impact of different crossover strategies on the performance and behavior of Differential Evolution in various optimization tasks. The experimental results indicated that binomial crossover consistently outperformed other strategies on standard benchmark problems widely used in the research community. Consequently, the present study adopts the binomial crossover approach.

4.2.4 Multi-objective Differential Evolution

As the name suggests, a multi-objective optimization method simultaneously considers more than one objective function. In many real-world decision-making problems, multiple, often conflicting objectives must be addressed, making single-objective approaches inadequate. Multi-objective optimization, also referred to as vector optimization, seeks to optimize a vector of objectives rather than a single scalar value. Unlike single-objective optimization, Multi-Objective Differential Evolution (MODE) is designed to optimize two or more competing aspects represented by separate fitness functions. Employing a single-objective DE in such scenarios would require heuristic aggregation of the different objectives into a scalar fitness function, potentially biasing the search. In contrast, MODE generates a set of Pareto-optimal solutions [80], each of which represents a trade-off among the conflicting objectives. A solution is considered Pareto-optimal if no other solution exists that improves at least one objective without degrading any of the others, ensuring that the set provides diverse, high-quality options for decision-making.

4.2.4.1 Pareto-Based Evaluation

Individuals are evaluated according to a Pareto-based ranking strategy [81]. Solutions that are not dominated by any other individual are assigned the first rank, corresponding to the highest fitness level, and are temporarily removed from the selection process. Subsequently, non-dominated solutions from the remaining population are assigned the second rank, and this procedure is repeated iteratively until all individuals are ranked.

To preserve a diverse set of Pareto-optimal solutions, a fitness sharing mechanism [81] is commonly employed in Multi-Objective Differential Evolution (MODE), enabling the maintenance of multiple well-distributed optimal solutions in the population (cf Figure 4.2).

4.2.4.2 Non-dominated Sorting Genetic Algorithm II

Deb et al. introduced a fitness sharing mechanism in NSGA-II [73], in which individuals within the same Pareto rank are differentiated based on the density of their surrounding solutions. Individuals located in less crowded regions are penalized to a lesser extent, thereby promoting population diversity.

To estimate the local density within each rank, a *crowding distance* metric is employed. For a given individual, this measure is computed as the sum of the normalized distances along each objective dimension between the two neighboring

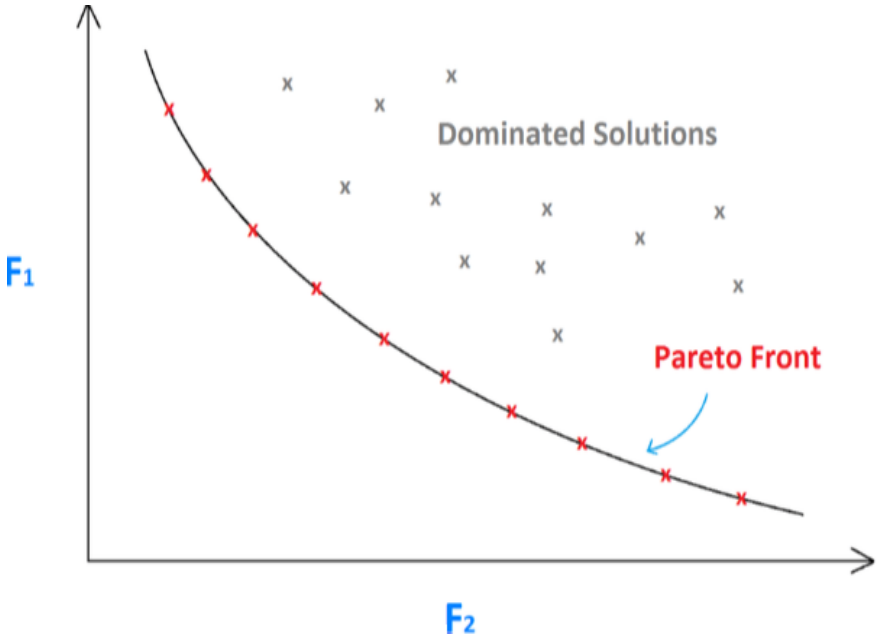

Figure 4.2: Pareto Optimality Visualization in 2D Objective Space [82].

individuals in the same rank that define the smallest enclosing interval for that individual.

After examining Multi-Objective Differential Evolution, Pareto-based evaluation, and the NSGA-II algorithm, our study naturally progresses toward another important variant of Differential Evolution, namely Binary Differential Evolution. This aspect is particularly significant, as binary DE constitutes a fundamental component of the proposed approach.

4.2.5 Binary Differential Evolution

The method uses Binary Differential Evolution(BDE) to perform efficient FS in a binary search space where each gene is represented by a bit. A variation of DE called BDE was created to address optimization issues with binary variables [83]. The aim of BDE is to effectively search discrete solution spaces where decision variables are binary. In BDE, the crossover and mutation operators have been adapted to work with binary variables. For every potential solution, a binary string of size n (genes) is used. Every bit in the string corresponds to a feature: a value of 1 means that the feature has been selected, whereas a value of 0 means that it has not. as illustrated in Figure 4.3

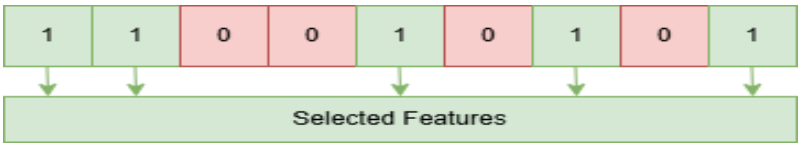

Figure 4.3: Representation of binary string encoding for FS solution.

4.2.6 Clustering-Based Binary Differential Evolution (CBDE) mutation operator

The proposed Clustering-Based Binary Differential Evolution (CBDE) operator is inspired by the Biclustering Binary Differential Evolution (BBDE) [83] strategy but introduces key improvements to better suit gene clustering for feature selection. Unlike BBDE, which simultaneously handles row and column structures, CBDE eliminates biclustering constraints and focuses exclusively on clustering genes into compact and discriminative subsets, a requirement more relevant to genomic feature selection.

Furthermore, the CBDE operator is designed to enhance the mutation process by prioritizing solutions with better fitness scores, thereby ensuring a more effective exploration of the search space. Specifically, three vectors $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, and $\mathbf{x}_{r3}$ are randomly selected and ranked according to their fitness values, where the best vector $\mathbf{x}_{r1}$ strongly influences the mutation process, while the worst vector $\mathbf{x}_{r3}$ is suppressed. This strategy biases the mutation toward high-quality solutions and increases the probability of generating better offspring.

After ranking, the mutation operator combines the selected vectors based on the chosen strategy, such as difference-based perturbation. A scaling factor F (initially set to 0.65) controls gene insertion or removal by comparing a randomly generated value $rand_i \in [0, 1]$, where i denotes the position of the current bit in the individual. This mechanism prevents solutions from degenerating into trivial cases (e.g., all zeros or all ones), thereby preserving population diversity and enabling efficient gene clustering.

The CBDE method performs gene insertion or removal when the generated random value $rand_i$ is less than F; otherwise, the gene remains unchanged.

The proposed CBDE operator introduces several innovations, including a clustering-oriented design for compact and non-redundant gene subsets, a fitness-guided ranking mechanism for knowledge-driven exploration, an adaptive dual insertion-removal strategy to preserve diversity, a perturbation tailored to gene clustering for biological interpretability and empirical validation, showing statistically significant improvements over conventional binary DE approaches.

$$\text{CBDE} = \text{GENEINSERTION}(\mathbf{x}_{r1}, \text{GENEREMOVAL}(\mathbf{x}_{r2}, \mathbf{x}_{r3}, F), F). \tag{4.2}$$

The new individual is evaluated using the fitness function.

$$F_i = \text{normrnd}(\text{mean}(F), 0.1), \tag{4.3}$$

$$CR_i = \text{normrnd}(\text{mean}(CR), 0.1). \tag{4.4}$$

F_i and CR_i represent the adapted versions of the mutation factor (F) and crossover probability (CR) for individual i, adjusting the mutation intensity and crossover rate respectively only when the individual shows no improvement, thereby ensuring effective self-adaptation and enhancing the algorithm's exploration capability. If its score improves, the values of F and CR remain unchanged; otherwise, they are slightly adjusted using predefined Eqs. (4.3) and (4.4).

A summary of the CBDE mutation operation, using the adapted gene insertion and removal mechanisms tailored for clustering tasks is as follows:

$$\text{GENEINSERTION}(x, y, F) = \begin{cases} 1, & \text{if } x_i = 1 \text{ or } (y_i = 1 \text{ and } rand_i < F) \\ x_i, & \text{otherwise} \end{cases} \tag{4.5}$$

$$\text{GENEREMOVAL}(x, y, F) = \begin{cases} 0, & \text{if } x_i = 0 \text{ or } (y_i = 1 \text{ and } rand_i < F) \\ x_i, & \text{otherwise} \end{cases} \tag{4.6}$$

4.2.7 Illustrative example of the CBDE mutation operator

Consider the gene expression matrix $M \in \mathbb{R}^{5\times5}$, where rows represent genes and columns represent samples, and the initial population P, consisting of four individuals represented by binary vectors (5 bits, one per gene):

$$M = \begin{pmatrix} 42 & 15 & 36 & 71 & 24 \\ 33 & 58 & 17 & 39 & 65 \\ 13 & 45 & 92 & 12 & 26 \\ 57 & 81 & 40 & 59 & 31 \\ 21 & 14 & 29 & 76 & 54 \end{pmatrix}$$

$$P = \begin{pmatrix} a: & 1 & 0 & 1 & 0 & 1 \\ b: & 0 & 1 & 1 & 0 & 0 \\ c: & 1 & 1 & 0 & 1 & 0 \\ d: & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

We apply the CBDE (b, c, d) mutation using Eq. (4.2) with scaling factor $F = 0.65$ and a random vector:

$$rand = [0.8, 0.9, 0.2, 0.5, 0.6].$$

Step 1: Gene Removal. We compute GENEREMOVAL(c, d, F) using Eq. (4.6). Applying it to $c = [1, 1, 0, 1, 0]$ and $d = [0, 0, 1, 1, 1]$, we obtain $x' = [1, 1, 0, 0, 0]$.

If the resulting fitness does not improve, Eqs. (4.3) and (4.4) are applied to adjust F and CR.

Step 2: Gene Insertion. We then compute GENEINSERTION(b, x', F) using Eq. (4.5). Applying it to $b = [0, 1, 1, 0, 0]$ and $x' = [1, 1, 0, 0, 0]$, we obtain

$$\text{CBDE}(b, c, d) = [1, 1, 1, 0, 0].$$

Interpretation: The resulting solution selects genes 1, 2, and 3. Therefore, the gene cluster constructed is:

$$\begin{pmatrix} 42 & 15 & 36 & 71 & 24 \\ 33 & 58 & 17 & 39 & 65 \\ 13 & 45 & 92 & 12 & 26 \end{pmatrix}.$$

This cluster is then evaluated using the fitness function which combines the number of selected genes and the MSR.

4.2.8 Fitness evaluation

The algorithm evaluates the quality of each solution using two objective functions: the size of the selected features and the Mean Squared Residual (MSR), aiming to guide the search toward solutions that balance minimal feature size with high classification accuracy.

In this study, the MSR [72] is used as the second objective function to assess the coherence of gene clusters by measuring the similarity of gene expression patterns across all samples; a lower MSR indicates more consistent and biologically relevant gene groups. The MSR for a cluster C that consists of I rows and J columns is

defined as follows:

$$\mathrm{MSR}(C) = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \left(c_{ij} - c_{iJ} - c_{Ij} + c_{IJ}\right)^2, \tag{4.7}$$

where c_{ij} is the expression value of gene i under condition j;

$$c_{iJ} = \frac{1}{|J|} \sum_{j \in J} c_{ij}$$

is the average expression of gene i across all conditions;

$$c_{Ij} = \frac{1}{|I|} \sum_{i \in I} c_{ij}$$

is the average expression of all genes under condition j; and

$$c_{IJ} = \frac{1}{|I| \times |J|} \sum_{i \in I} \sum_{j \in J} c_{ij}$$

is the overall average expression within the cluster.

4.2.9 Example

Consider a gene cluster C composed of 2 genes and their expression values across 3 samples (conditions) as follows:

$$C = \begin{pmatrix} 10 & 15 & 30 \\ 25 & 20 & 40 \end{pmatrix}.$$

We compute the MSR as defined in Eq. (4.7). For example, when $i = 1$, $j = 1$, we have:

$$c_{11} = 10, \quad c_{IJ} \approx 23.333, \quad c_{iJ} \approx 18.333, \quad c_{Ij} = 17.5.$$

We then substitute these into the MSR equation and iterate for all values in the matrix:

$$\mathrm{MSR}(C) = \frac{1}{6} \sum_{i=1}^{2} \sum_{j=1}^{3} \left(c_{ij} - c_{iJ} - c_{Ij} + c_{IJ}\right)^2.$$

After plugging in all values and performing the calculations:

$$\mathrm{MSR}(C) \approx 22.22.$$

4.2.10 Selection process

After evaluating fitness, the algorithm chooses the best individuals to create the next generation based on dominance (NSGA-II), ensures diversity and convergence.

4.2.11 Iteration and convergence

The process is repeated (mutation, crossover, fitness evaluation, and selection) the predetermined number of iterations or until the convergence criteria are completed.

4.3 Experiments and Evaluation

4.3.1 Experimental Settings

To get the best results, the study suggested DeFs-CBDE feature selection method optimal parameters were chosen by trial and error. The parameter settings for the suggested DeFs-CBDE approach on each experimental dataset are displayed in Table 4.2.

4.3.2 Description of considered datasets

In this work, we evaluated the effectiveness of the proposed DeFs-CBDE feature selection strategy on four high-dimensional carcinogenic microarray datasets, namely Brain, Breast, Lung, and Central Nervous System (CNS) cancers, as summarized in Table 4.1. These datasets were selected because they are well-curated, publicly available, and have been widely adopted as benchmark datasets in previous studies on gene expression–based feature selection and cancer classification. Their extensive use in the literature enables fair and meaningful comparisons with existing methods while ensuring validation under standard experimental conditions.

The Brain tumor dataset [84] contains 42 microarray samples characterized by 5,597 genes and is organized into five distinct classes: medulloblastomas, malignant gliomas, atypical teratoid/rhabdoid tumors, primitive neuroectodermal tumors, and normal human cerebella. This dataset is particularly challenging due to its multi-class structure and the high dimensionality relative to the number of samples.

The Breast cancer dataset [85] comprises 97 samples with 24,481 gene expression features. It is a binary classification problem, where 46 samples correspond to patients who developed distant metastases within five years, while 51 samples remained metastasis-free for at least five years. The large number of genes combined

with a limited number of samples makes this dataset a representative example of the "small-sample, high-dimensional" setting.

The Lung cancer dataset [86] includes 203 samples described by 12,600 genes and involves five classes: normal lung tissue (17 samples), adenocarcinoma (139 samples), small cell lung cancer (6 samples), squamous cell carcinoma (21 samples), and pulmonary carcinoid (20 samples). The presence of class imbalance and multiple tumor subtypes further increases the classification complexity of this dataset.

Finally, the CNS cancer dataset [86] consists of 60 samples with 7,129 genes and is divided into two classes, where 21 samples correspond to long-term cancer survivors and 39 samples represent treatment failures. This dataset is commonly used to evaluate the robustness of feature selection methods in binary classification scenarios involving survival outcomes.

Overall, these four datasets cover a wide range of classification settings, including binary and multi-class problems, varying levels of class imbalance, and extreme dimensionality, making them highly suitable for a comprehensive evaluation of the proposed DeFs-CBDE approach.

Table 4.1: Description of the high-dimensional microarray datasets used in this study

Dataset	#Features	#Samples	#Classes
Brain	5,597	42	5
Breast	24,481	97	2
Lung	12,600	203	5
CNS	7,129	60	2

The 5-fold cross-validation method is employed to divide the dataset into training and testing sets. Each algorithm is executed independently 20 times to ensure statistical reliability. All experiments were conducted on a Windows 10 PC with a 2.40 GHz Xeon E5620 8-core processor and 40 GB of RAM, using the parameter settings shown in Table 4.2.

Table 4.2: Hyperparameters used in the proposed DeFs-CBDE

Hyperparameter	Value
Initial F (scaling factor)	0.65
Crossover rate (CR)	0.6
Number of generations	50
Population size (individuals)	20

4.3.3 Experimental results and discussion

4.3.4 Architecture of the proposed method

As illustrated in Figure 4.4, the dataset is first subjected to a preprocessing stage, including normalization and noise removal, to ensure data consistency and improve learning stability. The preprocessed data are then partitioned into training and testing subsets. The training subset is used to apply the proposed DeFs-CBDE feature selection algorithm, which identifies a compact and informative subset of features, resulting in a reduced training set with feature selection.

To ensure a fair and unbiased evaluation, the same selected feature indices obtained from the training phase are subsequently applied to the testing data without re-running the DeFs-CBDE algorithm, thereby generating the corresponding feature-selected testing set. This strategy prevents information leakage from the test set and preserves the integrity of the evaluation process.

Finally, the newly generated training and testing subsets are fed into multiple classifiers to assess the discriminative capability of the selected features. Classification performance is evaluated using standard metrics such as accuracy, precision, recall, F-measure, and specificity, allowing a comprehensive assessment of the effectiveness and robustness of the proposed DeFs-CBDE framework.

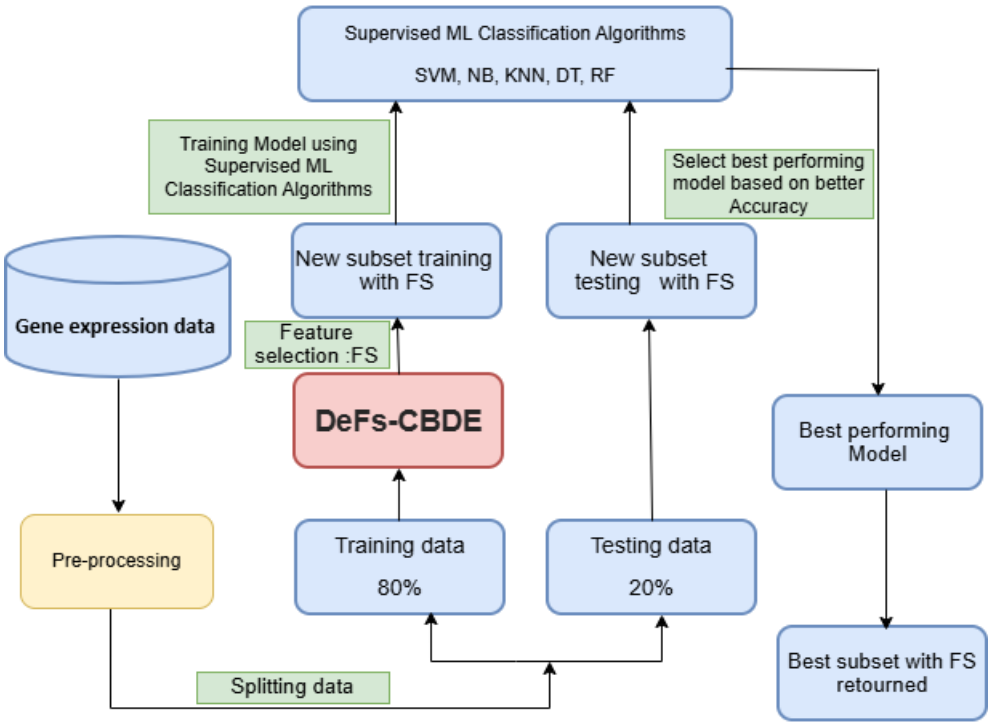

Figure 4.4: Illustration of DeFs-CBDE scheme

4.3.5 DeFs-CBDE Algorithm Description

By incorporating Differential Evolution (DE) into the feature selection procedure, the suggested method described in Algorithm 4 guides the search for the best feature subsets. The DeFs-CBDE (Clustering-Based Differential Evolution for Feature Selection) algorithm is a multi-objective optimization framework designed to identify the most discriminative genes in high-dimensional microarray data. The core logic of the algorithm is structured as follows:

4.3.5.1 Initialization and Objective Functions

The algorithm begins by generating an initial population P of size p, where each individual represents a binary vector indicating the presence or absence of specific genes. Unlike standard single-objective DE, this approach manages two conflicting objectives simultaneously:

- **MSR Minimization (f_1):** Reducing the MSR to improve classification accuracy and model generalization.
- **Gene Reduction (f_2):** Limiting the number of selected genes to achieve a parsimonious model and suppress redundant or noisy features.

4.3.5.2 The Evolutionary Loop and CBDE Mutation

For each generation, the algorithm explores the search space using a specialized **Clustering-Based Differential Evolution (CBDE)** operator. Three distinct solutions (x, y, z) are selected from the population and ranked based on their fitness to guide the mutation toward promising regions.

The core of the CBDE operator (Lines 23–31 of Algorithm 4) utilizes two dedicated functions:

- **GENEINSERTION:** Decisions are made to add potentially informative genes to the subset based on the scale factor F.
- **GENEREMOVAL:** This function prunes redundant or non-informative features, maintaining the sparsity required for high-dimensional gene expression analysis.

A trial vector is subsequently created through binomial crossover using probability C, combining the current solution with the mutated vector to maintain genetic diversity.

4.3.5.3 Parameter Adaptation and Multi-Objective Management

The algorithm features an adaptive mechanism (Lines 13–14) where the crossover probability C and mutation factor F are updated in each iteration based on the success of the offspring. This allows the search to transition from global exploration to local exploitation dynamically.

To manage the multi-objective nature of the problem, the algorithm integrates the **NSGA-II** framework:

- **Non-Dominated Sorting:** Solutions are organized into Pareto fronts, ensuring that the final output provides a range of optimal trade-offs between accuracy and subset size.
- **Crowding Distance:** This metric is employed to maintain a diverse set of solutions along the Pareto front, preventing the algorithm from converging prematurely to a single sub-optimal region.

4.3.5.4 Standard Deviation

Karl Pearson first developed the idea of standard deviation in 1983. The most practical and often used measure of dispersion among populations is the standard deviation. The Greek letter sigma (σ) is used to represent it. The standard deviation considers the value of each observation, just like the mean deviation does [87]. The steps to calculate standard deviation can be summarized as follows:

Given data: $x_1, x_2, x_3, \ldots, x_n$

Step 1: Mean (average)

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Step 2: Deviation from the mean

$$x_i - \mu$$

Step 3: Squared deviation

$$(x_i - \mu)^2$$

Step 4: Variance

Population variance:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Sample variance:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Step 5: Standard deviation

$$\sigma = \sqrt{\sigma^2}, \qquad s = \sqrt{s^2}$$

Step 6: Mean ± Standard Deviation

The final result is commonly reported as:

$$\mu \pm \sigma$$

For sample data, it is expressed as:

$$\bar{x} \pm s$$

Example

Given data: $x = \{2, 4, 6, 8\}$

$$\mu = \frac{2+4+6+8}{4} = 5$$

$$\sigma = \sqrt{\frac{9+1+1+9}{4}} = \sqrt{5} \approx 2.24$$

$$s = \sqrt{\frac{9+1+1+9}{3}} = \sqrt{6.67} \approx 2.58$$

Example representation

$$\mu \pm \sigma = 5 \pm 2.24$$

Algorithm 4 The pseudocode of DeFs-CBDE approach

Require: A partial training matrix of size $n \times m$ (n genes and m samples), Initial population size p, initial crossover probability C, initial factor F

Ensure: A matrix of size $s \times m$ (the dataset with s optimally selected features)

Initialize population P by generating p solutions, each gene random in $[0, 1]$
Compute fitness values (f_1, f_2) for each solution in P
Initialize array C and array F for all individuals in P using initial C and F
while termination criteria not met **do**
    for each solution i in P **do**
        Select three randomly chosen solutions (x, y, z) from P ranked from best to worst based on fitness, ensuring $x \neq y \neq z \neq i$
        Generate a mutant vector using CBDE operator:
            $\text{CBDE}(x, y, z, F_i)$
        Create a trial vector by performing binomial crossover between the mutant vector and the current solution i using crossover probability C
        Evaluate the fitness of the trial vector using f_1 and f_2
        if the trial vector is better than the current solution **then**
            Replace the current solution i with the trial vector
        else
            $C_i \leftarrow \text{normrnd}(\text{mean}(C), 0.1)$
            $F_i \leftarrow \text{normrnd}(\text{mean}(F), 0.1)$
        end if
    end for
    Merge current population P with offspring population
    Perform Non-Dominated Sorting using NSGA-II based on Pareto dominance
    Calculate crowding distance for solutions within each rank and select top p solutions for the next generation based on rank and crowding distance
end while
return the final set of selected features
function $\text{CBDE}(x, y, z, F)$
    $w = \text{GENEINSERTION}(x, \text{GENEREMOVAL}(y, z, F), F)$
end function
function $\text{GENEINSERTION}(x, y, F)$
    return if $x_i = 1$ **or** $(y_i = 1$ **and** $rand_i < F)$ **then** 1; **else** x_i
end function
function $\text{GENEREMOVAL}(x, y, F)$
    return if $x_i = 0$ **or** $(y_i = 1$ **and** $rand_i < F)$ **then** 0; **else** x_i
end function

4.3.6 Evaluation of the suggested DeFs-CBDE Performance

The performance of machine learning models using features selected by the proposed DeFs-CBDE method was compared to the FS techniques presented in [50]. Figure. 4.5 shows the classification results using all features, features selected by individual filter methods (Chi-squared (CS), Information Gain (IG), and Gain Ratio (IGR)), hybrid filter-GA methods (CS-GA, IG-GA, and IGR-GA) from [50], and those selected by DeFs-CBDE.

For the Brain dataset, Figure. 4.5(a) and Table 4.3 demonstrate the superiority of the proposed DeFs-CBDE method. It achieves 100 % accuracy, recall, precision, and F-measure with SVM, KNN, and RF, outperforming IG-GA, IGR-GA, and CS-GA (97.62 %–100 %). For NB and DT, DeFs-CBDE reaches 85.71 % accuracy, exceeding IG (61.90 %) and IGR (64.29 %) for DT, while remaining competitive for NB. The low standard deviations across runs (e.g., 97.48 ± 3.14 for SVM and 95.63 ± 5.40 for KNN) further confirm the robustness and stability of the proposed feature selection strategy.

Table 4.3: Performance comparison of classifiers on the Brain dataset

Metric	All	IG	IG-GA	IGR	IGR-GA	CS	CS-GA	DeFs-CBDE (Best)	DeFs-CBDE (Mean ± Std)
SVM									
Accuracy	69.05	73.81	85.71	78.57	97.62	83.33	97.62	**100**	97.48 ± 3.14
Recall	58.00	63.00	75.00	78.00	97.50	83.00	97.50	**100**	96.25 ± 3.76
Precision	42.28	64.05	68.57	66.52	98.18	88.41	98.18	**100**	96.58 ± 3.87
F-measure	48.91	63.52	71.64	71.80	97.84	85.62	97.84	**100**	96.98 ± 3.45
NB									
Accuracy	69.05	88.10	92.86	85.71	95.24	83.33	95.24	85.71	82.81 ± 2.14
Recall	60.50	80.50	88.00	78.00	93.00	76.50	93.00	85.71	82.17 ± 2.13
Precision	58.69	90.91	94.55	88.72	95.96	88.77	95.96	78.57	74.59 ± 2.32
F-measure	59.58	85.39	91.16	83.02	94.46	82.18	94.46	80.95	77.12 ± 2.25
KNN									
Accuracy	78.57	80.95	92.86	83.33	97.62	80.95	97.62	**100**	95.63 ± 5.40
Recall	75.50	80.50	92.50	83.00	97.50	80.50	97.50	**100**	97.90 ± 2.58
Precision	86.36	84.33	94.85	88.33	98.18	87.05	98.18	**100**	92.07 ± 4.54
F-measure	80.57	82.37	93.66	85.58	97.84	83.65	97.84	**100**	97.31 ± 2.74
DT									
Accuracy	50.00	61.90	85.71	64.29	88.10	69.05	85.71	85.71	83.46 ± 2.52
Recall	48.50	65.00	87.00	61.50	89.50	69.00	83.50	85.71	83.97 ± 2.49
Precision	53.06	61.73	87.22	63.83	88.50	69.89	88.01	78.57	75.99 ± 2.74
F-measure	50.68	63.32	87.11	62.64	89.00	69.44	85.70	80.95	77.62 ± 3.09
RF									
Accuracy	78.57	90.48	100	92.86	100	88.10	100	**100**	96.02 ± 4.06
Recall	76.50	88.00	100	93.00	100	88.00	100	**100**	96.35 ± 3.57
Precision	80.21	92.73	100	94.36	100	91.33	100	**100**	94.41 ± 4.52
F-measure	78.31	90.30	100	93.68	100	89.63	100	**100**	95.35 ± 4.18

For the breast dataset, Figure. 4.5(b) and Table 4.4 highlight the effectiveness of the proposed DeFs-CBDE approach across different classifiers. With SVM, DeFs-CBDE achieves the best performance, reaching 86.67 % in accuracy, recall, precision, and F-measure, clearly outperforming IG (74.23 %), IGR (69.07 %), and their GA-based variants. For RF, it attains 93.33 % accuracy, which is comparable to IGR-GA (93.81 %) and higher than IG-GA (89.69 %), while maintaining low variability (91.59 ± 1.42). In the case of DT, DeFs-CBDE achieves 84.85 % accuracy, representing a substantial improvement over the original feature set (57.73 %) and remaining competitive with IGR-GA (90.72 %). Although DeFs-CBDE shows lower performance with KNN (76.00 %) compared to IG-GA (89.69 %) and IGR-GA (86.60 %), it still outperforms most filter-based approaches. For NB, the proposed method yields 76.00 % accuracy, exceeding IG (55.67 %) and IGR (54.64 %), confirming the robustness and generalization ability of DeFs-CBDE for breast cancer gene selection.

Table 4.4: Performance comparison of classifiers on the Breast dataset

Metric	All	IG	IG-GA	IGR	IGR-GA	CS	CS-GA	DeFs-CBDE (Best)	DeFs-CBDE (Mean ± Std)
SVM									
Accuracy	52.58	74.23	84.54	69.07	82.47	73.20	82.47	86.67	84.35 ± 1.92
Recall	50.00	73.89	84.34	68.67	82.27	73.34	82.48	86.67	84.20 ± 2.03
Precision	26.29	74.41	84.69	69.21	82.60	73.30	82.43	86.67	84.14 ± 2.01
F-measure	34.46	74.15	84.51	68.94	82.43	73.32	82.45	86.67	83.96 ± 2.14
NB									
Accuracy	48.45	55.67	57.73	54.64	62.89	72.16	79.38	76.00	73.10 ± 1.91
Recall	46.93	53.47	55.54	52.71	60.87	72.04	79.33	70.00	67.99 ± 1.72
Precision	45.32	62.94	70.63	56.23	79.31	72.09	79.33	70.09	68.27 ± 1.43
F-measure	46.11	57.82	62.18	54.41	68.88	72.06	79.33	69.97	67.91 ± 1.50
KNN									
Accuracy	55.67	71.13	89.69	64.95	86.60	72.16	84.54	76.00	60.74 ± 1.65
Recall	54.43	70.52	89.77	63.58	85.98	71.50	84.44	63.33	60.74 ± 1.65
Precision	55.78	71.93	89.67	69.30	88.89	73.25	84.53	63.56	61.49 ± 1.38
F-measure	55.10	71.22	89.72	66.32	87.41	72.36	84.48	63.37	60.58 ± 1.66
DT									
Accuracy	57.73	67.01	86.60	60.82	90.72	69.07	84.54	84.85	84.85 ± 1.47
Recall	57.25	66.71	86.30	60.51	90.43	68.88	84.34	90.00	86.89 ± 1.92
Precision	57.52	66.97	87.09	60.67	91.31	69.00	84.69	86.67	83.97 ± 1.79
F-measure	57.38	66.84	86.69	60.59	90.87	68.94	84.51	86.79	83.69 ± 1.95
RF									
Accuracy	63.92	86.60	89.69	87.63	93.81	81.44	85.57	93.33	91.59 ± 1.42
Recall	63.55	86.51	89.66	87.49	93.80	81.29	85.64	94.17	91.58 ± 1.65
Precision	63.85	86.60	89.66	87.71	93.80	81.48	85.54	93.33	90.47 ± 1.80
F-measure	63.70	86.55	89.66	87.60	93.80	81.38	85.59	93.33	89.83 ± 1.91

For the lung dataset, Figure. 4.5(c) and Table 4.5 show that DeFs-CBDE delivers strong and stable performance across all classifiers. With SVM, the method achieves 97.56 % in accuracy, recall, precision, and F-measure, surpassing IG-GA (94.09 %) and IGR-GA (94.58 %) while maintaining low variability (96.88 ± 0.45). Similarly, DT attains 97.56 % accuracy, outperforming the original feature set (84.73 %) and competing filter-based methods. For NB and KNN, DeFs-CBDE achieves solid results with 96.72 % accuracy, improving over IG (90.15 % for NB) and IGR (89.66 % for KNN), confirming the method's reliability even with classifiers sensitive to feature distributions. Although RF reaches 91.80 %, slightly below some other methods, it remains competitive overall, demonstrating the robustness and versatility of DeFs-CBDE in selecting highly discriminative genes across different classifiers.

Table 4.5: Performance comparison of classifiers on the Lung dataset

Metric	All	IG	IG-GA	IGR	IGR-GA	CS	CS-GA	DeFs-CBDE (Best)	DeFs-CBDE (Mean ± Std)
SVM									
Accuracy	78.82	92.12	94.09	83.25	94.58	84.24	95.07	97.56	96.88 ± 0.45
Recall	41.13	71.60	75.68	52.80	86.47	55.15	90.03	97.56	96.62 ± 0.56
Precision	75.27	75.72	76.41	54.47	98.53	54.90	98.66	98.37	97.33 ± 0.57
F-measure	53.19	73.60	76.04	53.62	92.11	55.02	94.15	97.79	96.75 ± 0.61
NB									
Accuracy	90.15	95.07	98.52	93.60	97.54	92.12	97.04	96.72	96.06 ± 0.45
Recall	79.07	88.50	97.73	93.64	97.44	93.21	97.30	96.72	95.94 ± 0.45
Precision	88.21	94.36	98.54	88.90	93.92	84.92	93.05	95.23	94.39 ± 0.54
F-measure	83.39	91.34	98.13	91.21	95.65	88.87	95.13	95.08	94.27 ± 0.53
KNN									
Accuracy	92.61	92.61	97.04	89.66	96.06	92.12	95.57	96.72	96.27 ± 0.33
Recall	80.73	87.91	94.87	73.98	92.74	80.36	91.79	96.72	96.16 ± 0.35
Precision	95.22	89.10	95.40	93.10	97.78	95.09	97.65	97.20	96.57 ± 0.40
F-measure	87.38	88.50	95.13	82.45	95.19	87.11	94.63	96.81	96.24 ± 0.35
DT									
Accuracy	84.73	86.70	96.55	84.73	96.06	85.22	96.55	97.56	97.27 ± 0.21
Recall	69.07	73.69	94.68	69.07	92.20	72.40	93.15	97.56	97.20 ± 0.25
Precision	84.15	80.63	94.35	84.15	95.21	85.92	96.21	97.63	97.33 ± 0.22
F-measure	75.87	77.00	94.51	75.87	93.68	78.58	94.66	97.17	96.90 ± 0.23
RF									
Accuracy	83.74	93.60	96.06	91.13	95.57	92.61	96.06	91.80	91.51 ± 0.22
Recall	59.10	85.60	90.58	78.46	92.01	83.47	92.97	91.80	91.42 ± 0.24
Precision	93.54	97.15	97.86	94.45	97.73	96.82	97.86	92.73	92.34 ± 0.20
F-measure	72.43	91.01	94.08	85.72	94.78	89.65	95.35	90.65	90.28 ± 0.21

For the CNS dataset, Figure. 4.5(d) and Table 4.6 illustrate that DeFs-CBDE demonstrates competitive performance across all classifiers. With SVM, it achieves 91.67 % accuracy, recall, precision, and F-measure, surpassing most filter-based methods such as IG-GA (86.67 %) and CS-GA (83.33 %), while maintaining very low variability (91.37 ± 0.16). For NB, DeFs-CBDE reaches 88.89 % accuracy, outperforming IG (61.67 %) and IGR-GA (88.33 %), confirming its robustness even for probabilistic classifiers. Although slightly lower with KNN (83.33 %), DT (72.22 %), and RF (77.78 %), the method maintains balanced recall, precision, and F-measure values, demonstrating its reliability across classifiers with different characteristics. Overall, these results confirm the stability and generalization ability of DeFs-CBDE in feature selection for CNS gene expression data.

Table 4.6: Performance comparison of classifiers on the CNS dataset

Metric	All	IG	IG-GA	IGR	IGR-GA	CS	CS-GA	DeFs-CBDE (Best)	DeFs-CBDE (Mean ± Std)
SVM									
Accuracy	65	65	86.67	65	65	65	83.33	91.67	91.37 ± 0.16
Recall	50	50	82.05	50	50	50	77.29	91.67	91.36 ± 0.17
Precision	32.50	32.50	88.89	32.50	32.50	32.50	86.58	92.59	92.31 ± 0.13
F-measure	39.39	39.39	85.33	39.39	39.39	39.39	81.67	91.32	91.06 ± 0.12
NB									
Accuracy	61.67	75	90	78.33	88.33	70	83.33	88.89	88.59 ± 0.14
Recall	59.52	74.18	87.91	75.64	87.73	69.23	81.68	88.89	88.52 ± 0.14
Precision	59.03	72.92	89.86	76.25	86.96	68.00	81.68	92.59	92.19 ± 0.13
F-measure	59.27	73.54	88.87	75.94	87.34	68.61	81.68	89.57	89.18 ± 0.14
KNN									
Accuracy	61.67	75	93.33	68.33	83.33	75	88.33	83.33	83.00 ± 0.17
Recall	54.03	71.98	91.58	61.36	78.39	69.78	84.43	83.33	82.92 ± 0.15
Precision	55.12	72.50	93.71	64.44	84.44	73.01	90.06	86.46	85.89 ± 0.19
F-measure	54.57	72.24	92.63	62.86	81.30	71.36	87.15	82.43	82.00 ± 0.17
DT									
Accuracy	58.33	70	93.33	68.33	93.33	61.67	88.33	72.22	71.72 ± 0.21
Recall	54.76	67.03	91.58	63.55	91.58	54.03	84.43	72.22	71.62 ± 0.23
Precision	54.67	67.03	93.71	64.68	93.71	55.12	90.06	79.17	78.49 ± 0.21
F-measure	54.71	67.03	92.63	64.11	92.63	54.57	87.15	72.49	71.86 ± 0.21
RF									
Accuracy	53.33	80	91.67	80	90	83.33	88.33	77.78	77.24 ± 0.22
Recall	45.42	75.82	89.19	72.53	85.71	78.39	85.53	77.78	77.05 ± 0.26
Precision	44.44	78.93	92.46	72.53	93.33	84.44	88.49	82.72	81.94 ± 0.29
F-measure	44.92	77.34	90.80	72.53	89.36	81.30	86.98	68.06	66.94 ± 0.61

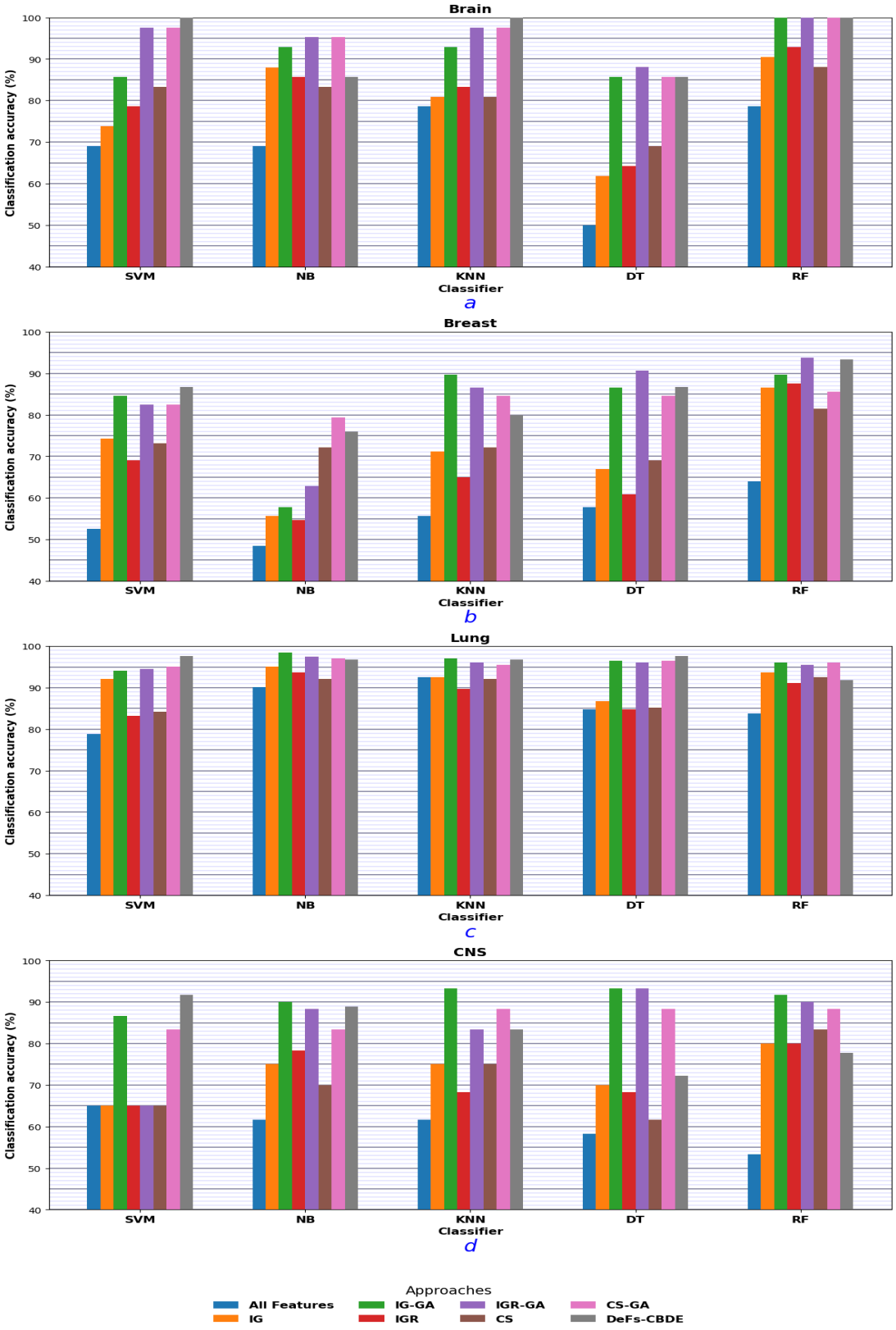

Figure 4.5: Comparison of classifier accuracies on the data sets: Brain (a); Breast (b); Lung (c); CNS (d) for DeFs-CBDE method

4.3.7 Comparison of DeFs-CBDE with existing works

In addition to comparisons with several state-of-the-art FS algorithms [50], the proposed DeFs-CBDE was evaluated against other existing works [24, 88, 89]. As shown in Figure 4.6 and summarized in Table 4.7, DeFs-CBDE consistently achieves high performance across all four datasets. On the Brain dataset, it reaches 100 % accuracy with SVM, KNN, and RF, while NB attains 85.71 %, outperforming the baseline of 50–78.57 %. For the Breast dataset, it achieves 93.33 % with RF and 86.67 % with SVM, maintaining competitive results with NB (76 %) and KNN (80 %). On the Lung dataset, top accuracies are 97.56 % with SVM and DT, while NB and KNN achieve 96.72 %, and RF reaches 91.80 %. Finally, on the CNS dataset, SVM and NB achieve 91.67 % and 88.89 %, respectively, with KNN, DT, and RF ranging from 77.78 % to 83.33 %. These results highlight the robustness and reliability of DeFs-CBDE across classifiers and datasets, confirming its effectiveness for high-dimensional gene expression feature selection.

Table 4.7: Evaluation of DeFs-CBDE through comparison with other existing feature selection methods in terms of best classification accuracy

Dataset	Classifier	Original dataset	S. Prajapati et al.	H. Hamla and K. Ghanem	Ahadzadeh et al.	**DeFs-CBDE**
Brain	SVM	69.05	/	/	/	**100**
	NB	69.05	/	/	/	85.71
	KNN	78.57	/	/	/	**100**
	DT	50.00	/	/	/	85.71
	RF	78.57	/	/	/	**100**
Breast	SVM	52.58	/	100	/	86.67
	NB	48.45	/	69.36	/	76.00
	KNN	55.67	/	85.95	94.94	80.00
	DT	57.73	70	72.61	/	86.67
	RF	63.92	75	84.09	/	93.33
Lung	SVM	78.82	/	100	/	97.56
	NB	90.15	/	92.12	/	96.72
	KNN	92.61	/	92.56	99.02	96.72
	DT	84.73	88.51	91.72	/	97.56
	RF	83.74	87.80	91.72	/	91.80
CNS	SVM	65.00	/	100	/	91.67
	NB	61.67	/	85.50	/	88.89
	KNN	61.67	/	90.33	96.66	83.33
	DT	58.33	83.33	77.83	/	72.22
	RF	53.33	66.66	86.33	/	77.78

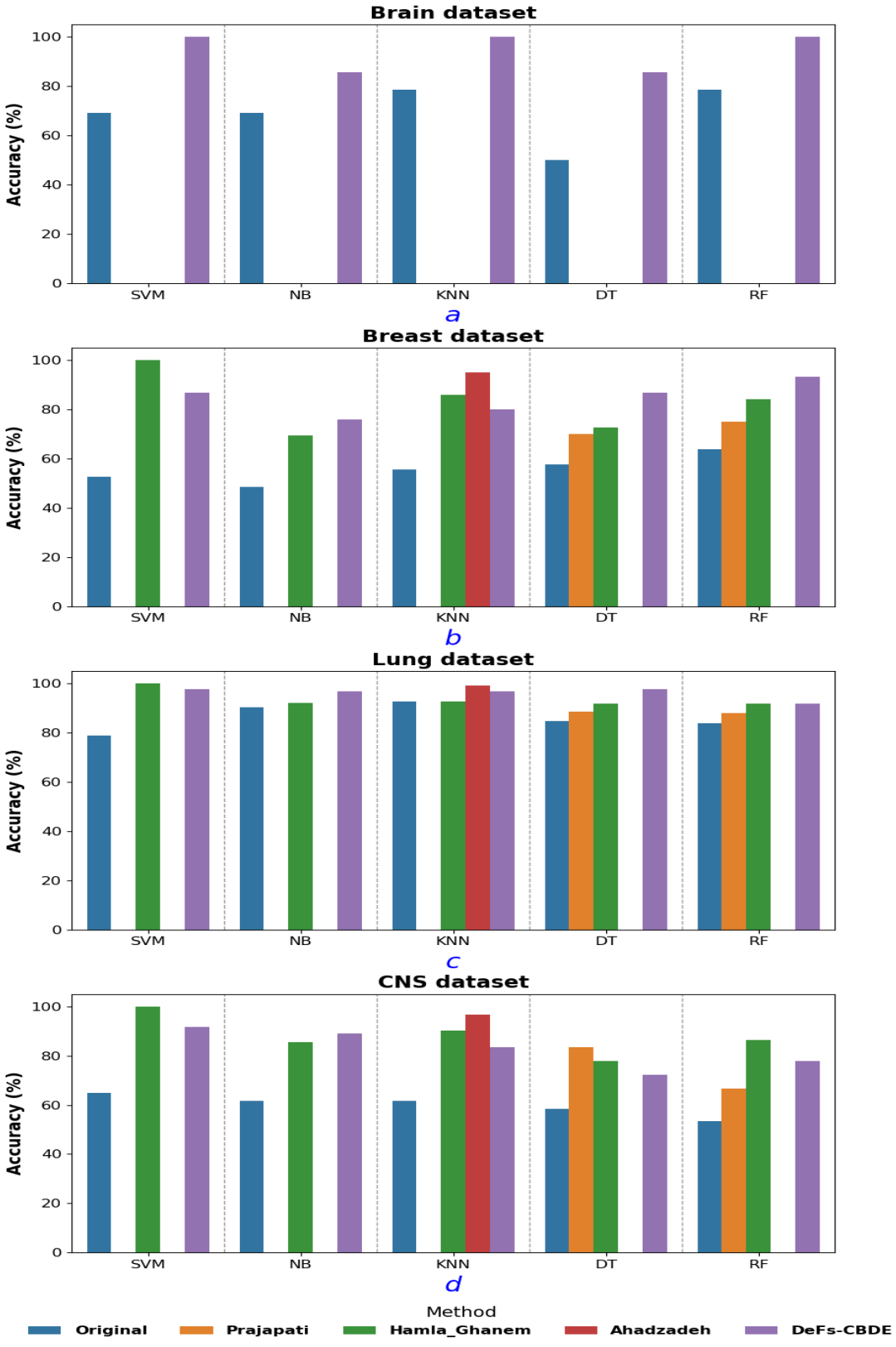

Figure 4.6: Comparison of classifier accuracies on the data sets: Brain (a); Breast (b); Lung (c); CNS (d) with existing works

4.3.8 Computational Complexity Analysis

The computational complexity of the proposed DeFs-CBDE algorithm is analyzed based on the main parameters of Algorithm 1. Let T denote the number of iterations (generations), p the population size, n the total number of genes (features), m the number of samples, s the number of selected features ($s \leq n$), and T_{fitness} the time complexity required to compute the fitness functions (f_1, f_2).

4.3.8.1 Initialization Phase

During the initialization phase (lines 1–3 of Algorithm 1), the algorithm generates an initial population of p individuals, each represented by a binary vector of length n. This step requires $O(p \times n)$ operations. Subsequently, the fitness values (f_1, f_2) are computed for all individuals, leading to a complexity of $O(p \times T_{\text{fitness}})$. The initialization of the adaptive parameters C_i and F_i for all individuals incurs a negligible cost of $O(p)$. Therefore, the total initialization complexity is given by:

$$O(p \times n + p \times T_{\text{fitness}}).$$

4.3.8.2 Main Evolutionary Loop

The main loop of the algorithm (lines 4–21) is executed for T iterations. In each iteration, the algorithm processes all p individuals in the population.

For each individual, three solutions (x, y, z) are randomly selected and ranked according to fitness, which can be performed in $O(p)$ time. The CBDE mutation operator, consisting of the *GENEINSERTION* and *GENEREMOVAL* functions, applies bitwise operations over n genes, resulting in a complexity of $O(n)$. The subsequent binomial crossover also requires $O(n)$ operations.

The fitness evaluation of the trial solution involves two objectives. The first objective f_1, which measures the number of selected features, can be computed in $O(n)$ time. The second objective f_2 is based on the Mean Squared Residue (MSR), which is calculated over a submatrix of size $s \times m$, resulting in a complexity of $O(s \times m)$. In addition, the classification-based evaluation introduces a classifier-dependent cost T_{clf}. Consequently, the overall fitness evaluation cost is:

$$T_{\text{fitness}} = O(s \times m + T_{\text{clf}}).$$

Updating the adaptive parameters C_i and F_i (lines 11–16) requires constant time

$O(1)$. Therefore, the overall cost per individual is:

$$O(n + T_{\text{fitness}}).$$

Thus, the computational cost per iteration is:

$$O(p \times (n + T_{\text{fitness}})).$$

4.3.8.3 Multi-objective Selection Phase

After generating the offspring population, the algorithm merges the parent and offspring populations (line 18), which incurs a cost of $O(p)$. The non-dominated sorting procedure based on NSGA-II (line 19) requires pairwise comparisons between individuals, leading to a complexity of $O(p^2)$. The computation of the crowding distance (line 20) requires $O(p \log p)$ operations, which is dominated by the non-dominated sorting step.

4.3.8.4 Overall Time Complexity

By combining all the above components, the overall time complexity of the DeFs-CBDE algorithm over T iterations can be expressed as:

$$O\Big(T \times \big(p \times (n + T_{\text{fitness}}) + p^2\big)\Big),$$

where $T_{\text{fitness}} = O(s \times m + T_{\text{clf}})$.

4.3.8.5 Memory Complexity

The memory complexity of the algorithm is mainly dominated by the storage of the population and the gene expression dataset. Storing the population requires $O(p \times n)$ memory, while storing the dataset requires $O(n \times m)$. Therefore, the overall memory complexity is given by:

$$O(p \times n + n \times m).$$

According to this analysis, the main factors influencing computational effort are the population size (p), the number of iterations (T), and the number of genes (n). By striking an effective balance among these variables, our method can handle high-dimensional data at a computationally affordable cost.

4.3.9 Biological Relevance of Selected Features (FS) — Lung Dataset

Figure. 4.7, a shows the functional enrichment results obtained from the genes selected by the FS method applied to the Lung dataset. The analysis, performed with the g:Profiler tool [90], highlights a strong biological significance of the retained genes. Indeed, among the selected features, about 70 % are mapped to significantly enriched pathways (p-value < 0.05 after multiple testing correction). The most enriched categories include cell cycle regulation, cell proliferation, and immune response, which are all closely related to lung cancer mechanisms.

Additionally, several enriched terms from Gene Ontology (GO) were identified, particularly within Biological Process and Molecular Function. For instance, genes involved in apoptosis regulation and intracellular signaling pathways are significantly represented (adjusted p-value $\approx 10^{-5}$). These findings demonstrate that the FS-selected genes are not random but rather capture key biological mechanisms of lung cancer, thus providing strong biological validation for the feature selection process, as shown in Table 4.8.

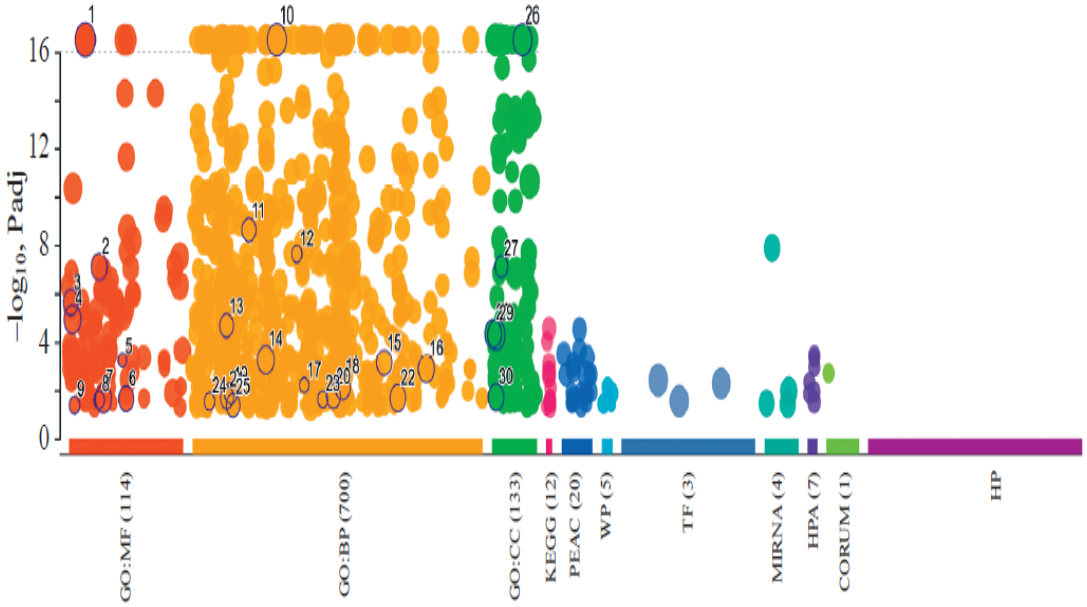

Figure 4.7: Functional enrichment analysis of FS-selected genes on the Lung dataset using, g:Profiler

Table 4.8: Enrichment results of FS-selected genes (Lung dataset, g:Profiler).

ID	Source	Term ID	Term Name	P_{adj}
1	GO:MF	GO:0005515	protein binding	4×10^{-70}
2	GO:MF	GO:0016301	kinase activity	8.162×10^{-8}
3	GO:MF	GO:0001216	DNA-binding transcription activator activity	2.369×10^{-6}
4	GO:MF	GO:0003690	double-stranded DNA binding	1.165×10^{-5}
5	GO:MF	GO:0035005	1-phosphatidylinositol-4-phosphate 3-kinase activity	5.772×10^{-4}
6	GO:MF	GO:0042578	phosphoric ester hydrolase activity	2.255×10^{-2}
7	GO:MF	GO:0017111	ribonucleoside triphosphate phosphatase activity	2.316×10^{-2}
8	GO:MF	GO:0016307	1-phosphatidylinositol-3-kinase activity	2.535×10^{-2}
9	GO:MF	GO:0004016	adenylate cyclase activity	4.119×10^{-2}
10	GO:BP	GO:0032501	multicellular organismal process	1.097×10^{-89}
11	GO:BP	GO:0016054	organic acid catabolic process	2.380×10^{-9}
12	GO:BP	GO:0019318	phosphatidylinositol-3-phosphate biosynthetic process	2.387×10^{-8}
13	GO:BP	GO:0007601	visual perception	2.105×10^{-5}
14	GO:BP	GO:0022414	reproductive process	5.497×10^{-4}
15	GO:BP	GO:0072521	purine-containing compound metabolic process	7.213×10^{-4}
16	GO:BP	GO:0019313	carbohydrate derivative metabolic process	1.307×10^{-3}
17	GO:BP	GO:0042354	L-fucose metabolic process	6.232×10^{-3}
18	GO:BP	GO:0007049	cell division	7.033×10^{-3}
19	GO:BP	GO:0009308	amine metabolic process	1.328×10^{-2}
20	GO:BP	GO:0048545	rhythmic process	1.773×10^{-2}
21	GO:BP	GO:0007612	cell recognition	2.023×10^{-2}
22	GO:BP	GO:0000793	supramolecular fiber organization	2.120×10^{-2}
23	GO:BP	GO:0046068	cGMP metabolic process	2.396×10^{-2}
24	GO:BP	GO:0003085	regulation of systemic arterial blood pressure by hormone	2.491×10^{-2}
25	GO:BP	GO:0009749	response to glucose	4.196×10^{-2}
26	GO:CC	GO:0005813	cell periphery	6.517×10^{-57}
27	GO:CC	GO:0022626	cytosolic ribosome	7.834×10^{-8}

We now provide concrete examples of top-ranked genes selected by CBDE and highlight their known involvement in lung cancer. For example:

- **Epidermal Growth Factor Receptor (EGFR):** frequently mutated in non-small cell lung cancer (NSCLC) and a target of tyrosine kinase inhibitors [91].
- **Kirsten Rat Sarcoma Viral Oncogene:** one of the most common driver mutations in lung adenocarcinoma, associated with poor prognosis [92].
- **Tumor Protein p53:** the most frequently mutated tumor suppressor gene in lung cancer, playing a central role in cell cycle regulation and apoptosis [92].
- **Anaplastic Lymphoma Kinase (ALK):** gene rearrangements involving ALK are important biomarkers guiding targeted therapy in lung adenocarcinoma [91].
- **Mucin 1:** overexpressed in lung cancer and associated with tumor progression and metastasis [93].

Furthermore, the functional enrichment analysis performed on the broader set of genes selected by CBDE (e.g., pathways such as *p53 signaling pathway* and *EGFR tyrosine kinase inhibitor resistance*) provides additional statistical support that our method captures not only individual key players but also biologically coherent pathways and processes critically involved in lung carcinogenesis.

4.4 Conclusion

In this chapter, we proposed a novel and effective feature selection strategy based on Multi-Objective Differential Evolution, termed DeFs-CBDE. A key contribution is the Clustering-Based Binary Differential Evolution (CBDE) mutation operator, which integrates gene insertion and removal to guide the search toward high-quality feature subsets. The method was evaluated using five classifiers—SVM, NB, KNN, DT, and RF—on high-dimensional gene expression datasets (Brain, Breast, Lung, and CNS). Comparisons with traditional filter-based methods (IG, IGR, CS), hybrid filter-GA approaches (IG-GA, IGR-GA, CS-GA), and other state-of-the-art feature selection techniques from recent studies demonstrate that DeFs-CBDE consistently achieves higher or highly competitive classification accuracies. Experimental results highlight its robustness, stability across classifiers, and the ability to reduce irrelevant features while retaining biologically meaningful genes. Despite increased

computational cost and the need for careful parameter tuning, the proposed approach offers reliable performance and practical applicability for high-dimensional biomedical data, outperforming existing methods and confirming its potential as a versatile tool in gene expression analysis.

General Conclusion

The results of this thesis open several directions for future research. This section highlights a number of promising avenues for further investigation, including the following:

In the growing field of bioinformatics, where handling massive volumes of high-dimensional biological data remains a major challenge, advanced analytical techniques have become essential. Feature Selection (FS) plays a crucial role in gene expression analysis by identifying the most informative genes while reducing dimensionality and noise. This thesis, which falls within the broad domain of feature selection methods, aims to propose effective solutions to directly address the challenges posed by large-scale microarray datasets. The primary objective is to enhance analytical efficiency, robustness, and interpretability, enabling more accurate and reliable analysis of gene expression data and contributing to the advancement of feature selection techniques in bioinformatics.

This thesis is organized into six chapters. It begins with a general overview of bioinformatics and the major challenges associated with high-dimensional gene expression data, with particular emphasis on the curse of dimensionality and its impact on classification and predictive modeling. The second chapter provides a comprehensive review of existing feature selection methods, including traditional techniques and state-of-the-art metaheuristic and swarm intelligence–based approaches. This review establishes a solid foundation for the proposed methods introduced in the subsequent chapters.

The core of this thesis focuses on the development of novel and effective feature selection strategies designed to address the challenges posed by high-dimensional microarray datasets. Each proposed approach targets specific limitations of existing feature selection techniques, contributing to a deeper understanding of how optimization-based methods can enhance robustness, efficiency, and biological relevance. The final chapter summarizes the main findings of the thesis and highlights the potential impact of the proposed methods on future research in feature selection for bioinformatics.

The contributions of this thesis are distinguished by the introduction of innovative feature selection methodologies, each carefully designed to address key challenges in high-dimensional genomic data analysis. The first major contribution exploits the strong global search capability and optimization efficiency of Differential Evolution (DE) for feature selection. Recognizing that classical DE operators are inherently designed for continuous search spaces, this work introduces adapted mechanisms suitable for binary feature encoding. The proposed DE-based feature selection framework efficiently explores large combinatorial search spaces and identifies compact, high-quality gene subsets, laying the groundwork for further advancements in evolutionary feature selection methods.

Experimental results demonstrate the effectiveness of employing differential evolution within a binary search space for global optimization. The proposed approach successfully selects informative gene subsets that improve classification performance while maintaining reasonable computational cost. This contribution represents an important step toward extending differential evolution techniques to complex feature selection problems in bioinformatics and provides a solid basis for future investigations.

The second major contribution of this thesis introduces a multi-objective feature selection framework, which extends the DE-based approach by simultaneously optimizing conflicting objectives such as classification accuracy and subset size. This multi-objective formulation enhances solution diversity and stability while offering a more comprehensive view of the trade-offs inherent in feature selection. The proposed method demonstrates strong performance across multiple benchmark gene expression datasets, yielding compact, discriminative, and biologically meaningful gene subsets. These results highlight the effectiveness of integrating multi-objective optimization principles into feature selection and represent a significant advancement in the field.

The third contribution explores a hybrid feature selection approach that combines deep learning–based data representation with swarm intelligence optimization. In this framework, a denoising autoencoder is employed as a preprocessing stage to reduce noise and extract meaningful representations from gene expression data, followed by a swarm intelligence algorithm adapted for feature selection. Experimental evaluations on standard microarray datasets demonstrate improved robustness to noise, enhanced classification accuracy, and greater stability compared to existing methods. This hybrid strategy underscores the potential of combining learning-based representations with evolutionary optimization for high-dimensional feature selection.

Future Research Directions

Several promising avenues for future research emerge from this work. One important direction involves extending the proposed feature selection frameworks to other rapidly evolving domains such as healthcare analytics, cybersecurity, text mining, and financial data analysis. Although the methods were primarily designed for gene expression data, their flexibility and strong optimization capabilities make them suitable for a wide range of high-dimensional data mining tasks, including anomaly detection and pattern recognition in heterogeneous datasets.

Another promising direction for future research lies in the development of hybrid feature selection frameworks that combine the complementary strengths of multiple metaheuristic algorithms. In particular, integrating Particle Swarm Optimization (PSO) with Differential Evolution (DE) or Slime Mould Algorithm (SMA) with DE could further enhance the balance between global exploration and local exploitation, leading to more accurate and stable gene subset selection.

Beyond these combinations, future studies may investigate other hybrid strategies, such as Genetic Algorithms coupled with Differential Evolution (GA–DE), Ant Colony Optimization with DE (ACO–DE), PSO combined with SMA, or the integration of deep learning models with Differential Evolution (DL–DE), where neural networks guide or adapt the evolutionary search process. Such hybridization schemes have strong potential to exploit diverse search behaviors, enhance representation learning, and improve convergence reliability, while reinforcing solution quality and mitigating premature convergence.

In addition, incorporating adaptive or self-tuning mechanisms within hybrid models—where control parameters and search strategies dynamically evolve during the optimization process and are validated across a wide range of heterogeneous gene expression datasets—could significantly reduce algorithmic complexity and eliminate the need for manual parameter tuning. Evaluating such models on diverse datasets of varying dimensionality and sample sizes is essential to improve the effectiveness, robustness, and generalization capability of feature selection approaches.

Finally, future work should emphasize computational efficiency and scalability by optimizing convergence speed, reducing redundant evaluations, and exploiting parallel or distributed implementations of hybrid algorithms. These improvements would enable faster and more robust feature selection, facilitating large-scale biological data analysis and strengthening the practical applicability of feature selection techniques in bioinformatics and precision medicine.

Bibliography

[1] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 306–307, 1979. doi:10.1109/TPAMI.1979.4766926.

[2] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Information fusion*, vol. 52, pp. 1–12, 2019. doi:10.1016/j.inffus.2018.11.008.

[3] R. Ghiasi and A. Malekjafarian, "Feature subset selection in structural health monitoring data using an advanced binary slime mould algorithm," *Journal of Structural Integrity and Maintenance*, vol. 8, no. 4, pp. 209–225, 2023. doi:10.1080/24705314.2023.2230398.

[4] A. Mahmoud and E. Takaoka, "An enhanced machine learning approach with stacking ensemble learner for accurate liver cancer diagnosis using feature selection and gene expression data," *Healthcare Analytics*, vol. 7, p. 100373, 2025. doi:10.1016/j.health.2024.100373.

[5] F. Saberi-Movahed, M. Rostami, K. Berahmand, S. Karami, P. Tiwari, M. Oussalah, and S. S. Band, "Dual regularized unsupervised feature selection based on matrix factorization and minimum redundancy with application in gene selection," *Knowledge-Based Systems*, vol. 256, p. 109884, 2022. doi:10.1016/j.knosys.2022.109884.

[6] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104210, 2021. doi:10.1016/j.engappai.2021.104210.

[7] **Serandi, Mohamed Djellal**, A. Houari, F. Boufera, and F. Flitti, "Demofs: Multi-objective feature selection using binary differential evolution (bde) optimization on breast cancer gene expression data," in *Modeling, Simulation and*

Computer Technology. ICMSCT 2024, vol. 2606 of *Communications in Computer and Information Science*, (Cham), Springer, 2025. doi:10.1007/978-3-032-01922-6$_2$2.

[8] A. Khoirunnisa, D. Adytia, M. M. Deris, *et al.*, "Comprehensive review of hybrid feature selection methods for microarray-based cancer detection," *IEEE Access*, 2025. doi:10.1109/ACCESS.2025.3564706.

[9] A. Yaqoob, R. M. Aziz, N. K. Verma, P. Lalwani, A. Makrariya, and P. Kumar, "A review on nature-inspired algorithms for cancer disease prediction and classification," *Mathematics*, vol. 11, no. 5, p. 1081, 2023. doi:10.3390/math11051081.

[10] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," *Handbook of metaheuristics*, pp. 311–351, 2018. doi:10.1007/978-3-319-91086-4$_1$0.

[11] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future generation computer systems*, vol. 111, pp. 300–323, 2020. doi:10.1016/j.future.2020.03.055.

[12] K. Tadist, S. Najah, N. S. Nikolov, F. Mrabti, and A. Zahi, "Feature selection methods and genomic big data: a systematic review," *Journal of Big Data*, vol. 6, no. 1, pp. 1–24, 2019. doi:10.1186/s40537-019-0241-0.

[13] B. Hosseini and K. Kiani, "A big data driven distributed density based hesitant fuzzy clustering using apache spark with application to gene expression microarray," *Engineering Applications of Artificial Intelligence*, vol. 79, pp. 100–113, 2019. doi:10.1016/j.engappai.2019.01.006.

[14] T. H. Pham and B. Raahemi, "Bio-inspired feature selection algorithms with their applications: a systematic literature review," *IEEE Access*, vol. 11, pp. 43733–43758, 2023. doi:10.1109/ACCESS.2023.3272556.

[15] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, and L. Gu, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Applied Soft Computing*, vol. 74, pp. 634–642, 2019. doi:10.1016/j.asoc.2018.10.036.

[16] S. Zhang, C. K. Lee, H. K. Chan, K. L. Choy, and Z. Wu, "Swarm intelligence applied in green logistics: A literature review," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 154–169, 2015. doi:10.1016/j.engappai.2014.09.007.

[17] S. B. B. POLOCK, A. DUTTA, and S. DAS, "A review on influx of bio-inspired algorithms: Critique and improvement needs," *J. ACM*, vol. 37, no. 4, 2025. doi:10.48550/arXiv.2506.04238.

[18] M. Georgioudakis and V. Plevris, "A comparative study of differential evolution variants in constrained structural optimization," *Frontiers in Built Environment*, vol. 6, p. 102, 2020. doi:10.3389/fbuil.2020.00102.

[19] A. Hashmi, W. Ali, A. Abulfaraj, F. Binzagr, and E. Alkayal, "Enhancing cancerous gene selection and classification for high-dimensional microarray data using a novel hybrid filter and differential evolutionary feature selection," *Cancers*, vol. 16, no. 23, p. 3913, 2024. doi:10.3390/cancers16233913.

[20] S. Hassan, A. M. Hemeida, S. Alkhalaf, A.-A. Mohamed, and T. Senjyu, "Multi-variant differential evolution algorithm for feature selection," *Scientific Reports*, vol. 10, no. 1, p. 17261, 2020. doi:10.1038/s41598-020-74228-0.

[21] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, and X.-y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67–85, 2020. doi:10.1016/j.ins.2019.08.040.

[22] **Serandi, Mohamed Djellal**, F. Boufera, A. Houari, and F. Flitti, "Defs-cbde: Clustering-guided binary mutation in multi-objective differential evolution for microarray gene selection," *Scientific and Technical Journal of Information Technology, Mechanics and Optics*, vol. 25, no. 6, pp. 1185–1196, 2025. doi:10.17586/2226-1494-2025-25-6-1185-1196.

[23] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, "Multi-objective evolutionary algorithms for filter based feature selection in classification," *International Journal on Artificial Intelligence Tools*, vol. 22, no. 04, p. 1350024, 2013. doi:10.1142/S0218213013500243.

[24] 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," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1896–1911, 2023. doi:10.1109/TEVC.2023.3238420.

[25] S. A. Ludwig, "Guided particle swarm optimization for feature selection: Application to cancer genome data," *Algorithms*, vol. 18, no. 4, p. 220, 2025. doi:10.3390/a18040220.

[26] D. Paul, A. Jain, S. Saha, and J. Mathew, "Multi-objective pso based online feature selection for multi-label classification," *Knowledge-Based Systems*, vol. 222, p. 106966, 2021. doi:10.1016/j.knosys.2021.106966.

[27] F. Han, W.-T. Chen, Q.-H. Ling, and H. Han, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm and Evolutionary Computation*, vol. 62, p. 100847, 2021. doi:10.1016/j.swevo.2021.100847.

[28] W. Ma, X. Zhou, H. Zhu, L. Li, and L. Jiao, "A two-stage hybrid ant colony optimization for high-dimensional feature selection," *Pattern Recognition*, vol. 116, p. 107933, 2021. doi:10.1016/j.patcog.2021.107933.

[29] L. Meenachi and S. Ramakrishnan, "Metaheuristic search based feature selection methods for classification of cancer," *Pattern Recognition*, vol. 119, p. 108079, 2021. doi:10.1016/j.patcog.2021.108079.

[30] D. Yilmaz Eroglu and U. Akcan, "An adapted ant colony optimization for feature selection," *Applied Artificial Intelligence*, vol. 38, no. 1, p. 2335098, 2024. doi:10.1080/08839514.2024.2335098.

[31] D. Santhakumar, G. Rajaram, R. Elankavi, J. Viswanath, I. Govindharaj, and J. Raja, "Enhanced leukemia prediction using hybrid ant colony and ant lion optimization for gene selection and classification," *MethodsX*, vol. 14, p. 103239, 2025. doi:10.1016/j.mex.2025.103239.

[32] R. M. Aziz, "Application of nature inspired soft computing techniques for gene selection: a novel frame work for classification of cancer," *Soft Computing*, vol. 26, no. 22, pp. 12179–12196, 2022. doi:10.1007/s00500-022-07032-9.

[33] J. Wang, Y. Zhang, M. Hong, H. He, and S. Huang, "A self-adaptive level-based learning artificial bee colony algorithm for feature selection on high-dimensional classification," *Soft Computing*, vol. 26, no. 18, pp. 9665–9687, 2022. doi:10.1007/s00500-022-06826-1.

[34] Z. Xu, B. Zhang, J. Yu, L. Yang, F. Yuan, and S. Gao, "An improved artificial bee colony algorithm with differential analysis and deduplication for feature selection," *International Journal of Computational Intelligence Systems*, 2025. doi:10.1007/s44196-025-01022-z.

[35] P. Gulande and R. Awale, "Microarray gene expression classification: An efficient feature selection using hybrid swarm intelligence algo-

rithm.," *Computer Systems Science & Engineering*, vol. 48, no. 4, 2024. doi:10.32604/csse.2024.046123.

[36] R. Fakoor *et al.*, "Using deep learning to enhance cancer diagnosis and classification," *ICML Workshop on Deep Learning*, 2013.

[37] M. Balın, A. Abid, and J. Zou, "Concrete autoencoders: Differentiable feature selection and reconstruction," in *ICML*, 2019.

[38] J. Wang *et al.*, "Unsupervised feature selection via adaptive autoencoder with redundancy control," *Neural Networks*, 2022. doi:10.1016/j.neunet.2022.03.004.

[39] Y. Li *et al.*, "Graph-regularized autoencoder for unsupervised feature selection," *Knowledge-Based Systems*, 2022.

[40] W. Hassanieh and A. Chehade, "Selective deep autoencoder for unsupervised feature selection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 12322–12330, 2024. doi:10.1609/aaai.v38i11.29123.

[41] X. Zhang *et al.*, "Selective deep autoencoder for unsupervised feature selection," in *AAAI*, 2024. doi:10.1609/aaai.v38i11.29123.

[42] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112–123, 2014. doi:10.1016/j.engappai.2014.03.007.

[43] M. Dashtban, M. Balafar, and P. Suravajhala, "Gene selection for tumor classification using a novel bio-inspired multi-objective approach," *Genomics*, vol. 110, no. 1, pp. 10–17, 2018. doi:10.1016/j.ygeno.2017.07.010.

[44] T. Li and H. Dong, "Unsupervised feature selection and clustering optimization based on improved differential evolution," *IEEE Access*, vol. 7, pp. 140438–140450, 2019. doi:10.1109/ACCESS.2019.2937739.

[45] M. Lamba, G. Munjal, and Y. Gigras, "A hybrid gene selection model for molecular breast cancer classification using a deep neural network," *International Journal of Applied Pattern Recognition*, vol. 6, no. 3, pp. 195–216, 2021. doi:10.1504/IJAPR.2021.117203.

[46] T. Li, Z.-H. Zhan, J.-C. Xu, Q. Yang, and Y.-Y. Ma, "A binary individual search strategy-based bi-objective evolutionary algorithm for high-dimensional feature selection," *Information Sciences*, vol. 610, pp. 651–673, 2022. doi:10.1016/j.ins.2022.07.183.

[47] M. Li, L. Ke, L. Wang, S. Deng, and X. Yu, "A novel hybrid gene selection for tumor identification by combining multifilter integration and a recursive flower pollination search algorithm," *Knowledge-Based Systems*, vol. 262, p. 110250, 2023. doi:10.1016/j.knosys.2022.110250.

[48] L. Li, M. Xuan, Q. Lin, M. Jiang, Z. Ming, and K. C. Tan, "An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 4, pp. 802–816, 2023. doi:10.1109/TEVC.2023.3254155.

[49] F. Cheng, J. Cui, Q. Wang, and L. Zhang, "A variable granularity search-based multiobjective feature selection algorithm for high-dimensional data classification," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 266–280, 2022. doi:10.1109/TEVC.2022.3160458.

[50] W. Ali and F. Saeed, "Hybrid filter and genetic algorithm-based feature selection for improving cancer classification in high-dimensional microarray data," *Processes*, vol. 11, no. 2, p. 562, 2023. doi:10.3390/pr11020562.

[51] Z. Xu, F. Yang, C. Tang, H. Wang, S. Wang, J. Sun, and Y. Zhang, "Fg-hfs: A feature filter and group evolution hybrid feature selection algorithm for high-dimensional gene expression data," *Expert Systems with Applications*, vol. 245, p. 123069, 2024. doi:10.1016/j.eswa.2023.123069.

[52] Z.-Z. Li, F.-L. Wang, F. Qin, Y. B. Yusoff, and A. M. Zain, "Feature selection of gene expression data using a modified artificial fish swarm algorithm with population variation," *IEEE Access*, vol. 12, pp. 72688–72706, 2024. doi:10.1109/ACCESS.2024.3402652.

[53] S. Al-Azani, O. S. Alkhnbashi, E. Ramadan, and M. Alfarraj, "Gene expression-based cancer classification for handling the class imbalance problem and curse of dimensionality," *International Journal of Molecular Sciences*, vol. 25, no. 4, p. 2102, 2024. doi:10.3390/ijms25042102.

[54] T. M. Le, L. Van Tran, and S. V. T. Dao, "A feature selection approach for fall detection using various machine learning classifiers," *IEEE Access*, vol. 9, pp. 115895–115908, 2021. doi: 10.1109/ACCESS.2021.3105581.

[55] M. Z. Ali, A. Abdullah, A. M. Zaki, F. H. Rizk, M. M. Eid, and E. M. El-Kenway, "Advances and challenges in feature selection methods: a comprehensive review," *J. Artif. Intell. Metaheuristics*, vol. 7, no. 1, pp. 67–77, 2024. doi:10.54216/JAIM.070105.

[56] G. Chen and J. Chen, "A novel wrapper method for feature selection and its applications," *Neurocomputing*, vol. 159, pp. 219–226, 2015. doi:10.1016/j.neucom.2015.01.070.

[57] P. Moscato, C. Cotta, and A. Mendes, "Memetic algorithms," in *New optimization techniques in engineering*, pp. 53–85, Springer, 2004. doi:10.1007/978-3-540-39930-8_3.

[58] H. Yapici and N. Cetinkaya, "A new meta-heuristic optimizer: Pathfinder algorithm," *Applied soft computing*, vol. 78, pp. 545–568, 2019. doi:10.1016/j.asoc.2019.03.012.

[59] 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, 2022. doi:10.1007/s00521-022-07705-4.

[60] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., University of California, 1985. doi:10.21236/ADA164453.

[61] S. Rezazadeh, "Review of machine learning," *TMP Universal Journal of Research and Review Archives*, vol. 4, no. 2s, 2025. doi:10.69557/ujrra.v4i2s.190.

[62] I. Sarker, "Machine learning: algorithms, real-world applications and research directions. sn comput sci 2: 160," 2021. doi: 10.1007/s42979-021-00592-x.

[63] I. Muhammad, "Supervised machine learning approaches: A survey," *ICTACT Journal on Soft Computing*, 2015. doi:10.21917/ijsc.2015.0133.

[64] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, 2004. doi:10.1198/016214504000000098.

[65] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Proceedings of the 22nd international conference on Machine learning*, pp. 529–536, 2005. doi:10.1145/1102351.1102418.

[66] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," in *2018 International conference on computing, power and communication technologies (GUCON)*, pp. 347–352, IEEE, 2018. doi:10.1109/GUCON.2018.8675097.

[67] A.-D. Dana and A. Alashqur, "Using decision tree classification to assist in the prediction of alzheimer's disease," in *2014 6th international conference on computer science and information technology (CSIT)*, pp. 122–126, IEEE, 2014. doi:10.1109/CSIT.2014.6805989.

[68] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[69] D. Pandey, K. Niwaria, and B. Chourasia, "Machine learning algorithms: a review," *Mach. Learn*, vol. 6, no. 2, 2019.

[70] R. S. Sutton, "Introduction: The challenge of reinforcement learning," in *Reinforcement learning*, pp. 1–3, Springer, 1992. doi:10.1007/978-1-4615-3618-5_1.

[71] M. L. Littman, A. W. Moore, *et al.*, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, no. 1, pp. 237–285, 1996.

[72] M. D. Noronha, R. Henriques, S. C. Madeira, and L. E. Zárate, "Impact of metrics on biclustering solution and quality: a review," *Pattern Recognition*, vol. 127, p. 108612, 2022. doi:10.1016/j.patcog.2022.108612.

[73] D. Kalyanmoy, "A fast and elitist multi-objective genetic algorithm: Nsga-ii," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[74] A. Dabba, A. Tari, S. Meftali, and R. Mokhtari, "Gene selection and classification of microarray data method based on mutual information and moth flame algorithm.," *Expert Systems with Applications*, vol. 166, p. 114012, 2021. doi:10.1016/j.eswa.2020.114012.

[75] R. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces,"

[76] B. Hegerty, C.-C. Hung, and K. Kasprak, "A comparative study on differential evolution and genetic algorithms for some combinatorial problems," in *Proceedings of 8th Mexican international conference on artificial intelligence*, vol. 9, p. 13, 2009.

[77] M. Iwan, R. Akmeliawati, T. Faisal, and H. M. Al-Assadi, "Performance comparison of differential evolution and particle swarm optimization in constrained optimization," *Procedia Engineering*, vol. 41, pp. 1323–1328, 2012. doi:10.1016/j.proeng.2012.07.317.

[78] Z. Yang, K. Tang, and X. Yao, "Scalability of generalized adaptive differential evolution for large-scale continuous optimization," *Soft Computing*, vol. 15, no. 11, pp. 2141–2155, 2011. doi:10.1007/s00500-010-0643-6.

[79] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms," *Applied soft computing*, vol. 9, no. 3, pp. 1126–1138, 2009. doi:10.1016/j.asoc.2009.02.012.

[80] K. Deb, "Multi-objective optimisation using evolutionary algorithms: an introduction," in *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34, Springer, 2011. doi:10.1007/978-0-85729-652-8_1.

[81] D. Goldberg, "Genetic algorithms for selection, optimization, and machine learning," *Reading: Addison-Wesley*, 1989.

[82] Y. Charfaoui, *Algorithmes de Bi-regroupement pour l'Analyse des Données Complexes.* PhD thesis, 2024.

[83] Y. Charfaoui, A. Houari, and F. Boufera, "Amodebic: An adaptive multi-objective differential evolution biclustering algorithm of microarray data using a biclustering binary mutation operator," *Expert Systems with Applications*, vol. 238, p. 121863, 2024. doi:10.1016/j.eswa.2023.121863.

[84] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, *et al.*, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, no. 6870, pp. 436–442, 2002. doi:10.1038/415436a.

[85] L. J. Van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen, *et al.*, "Gene expression profiling predicts clinical outcome of breast cancer," *nature*, vol. 415, no. 6871, pp. 530–536, 2002. doi:10.1038/415530a.

[86] G. Zhao and Y. Wu, "Feature subset selection for cancer classification using weight local modularity," *Scientific Reports*, vol. 6, no. 1, p. 34759, 2016. doi:10.1038/srep34759.

[87] M. S. H. Khan, "Standard deviation," in *International Encyclopedia of Statistical Science* (M. Lovric, ed.), pp. 2413–2415, Springer, Berlin, Heidelberg, 2025. doi:10.1007/978-3-662-69359-$9_5$93.

[88] S. Prajapati, H. Das, and M. K. Gourisaria, "Feature selection using differential evolution for microarray data classification," *Discover Internet of Things*, vol. 3, no. 1, p. 12, 2023. doi:10.1007/s43926-023-00042-5.

[89] H. Hamla and K. Ghanem, "A hybrid feature selection based on fisher score and svm-rfe for microarray data," *Informatica*, vol. 48, no. 1, 2024. doi:10.31449/inf.v48i1.4759.

[90] L. Kolberg, U. Raudvere, I. Kuzmin, P. Adler, J. Vilo, and H. Peterson, "g: Profiler—interoperable web service for functional enrichment analysis and gene identifier mapping (2023 update)," *Nucleic acids research*, vol. 51, no. W1, pp. W207–W212, 2023. doi:10.1093/nar/gkad347.

[91] R. S. Herbst, D. Morgensztern, and C. Boshoff, "The biology and management of non-small cell lung cancer," *Nature*, vol. 553, no. 7689, pp. 446–454, 2018. doi:10.1038/nature25183.

[92] F. Skoulidis and J. V. Heymach, "Co-occurring genomic alterations in non-small-cell lung cancer biology and therapy," *Nature Reviews Cancer*, vol. 19, no. 9, pp. 495–509, 2019. doi:10.1038/s41568-019-0179-8.

[93] S. Nath and P. Mukherjee, "Muc1: a multifaceted oncoprotein with a key role in cancer progression," *Trends in molecular medicine*, vol. 20, no. 6, pp. 332–342, 2014. doi:10.1016/j.molmed.2014.02.007.