



REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

**ENHANCED BIO-INSPIRED MANTARAY FOR
GENES SELECTION OF ALZHEIMER'S DISEASE
USING MICROARRAY DATASETS**

Zahraa Khalid Ahmed AHMED

Doctor of Philosophy

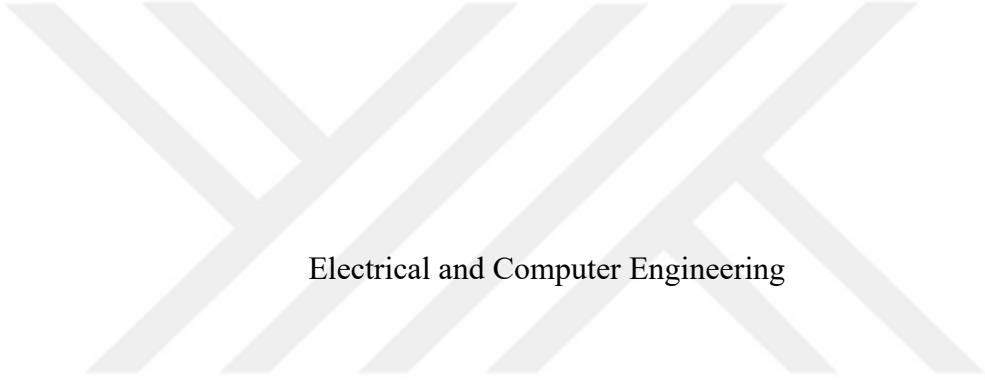
Supervisor

Asst. Prof. Dr. Mesut ÇEVİK

İstanbul, 2025

ENHANCED BIO-INSPIRED MANTARAY FOR GENES SELECTION OF ALZHEIMER'S DISEASE USING MICROARRAY DATASETS

Zahraa Khalid Ahmed AHMED



Electrical and Computer Engineering

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2025

The dissertation titled ENHANCED BIO-INSPIRED MANTARAY FOR GENES SELECTION OF ALZHEIMER’S DISEASE USING MICROARRAY DATASETS prepared by ZAHRA KHALID AHMED AHMED and submitted on 04/07/2025 has been **accepted unanimously** for the degree of PhD in Electrical and Computer Engineering.

Asst. Prof. Dr. Mesut ÇEVİK
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Mesut ÇEVİK	Department of Electrical- Electronics Engineering, Altınbaş University	
Prof. Dr. Osman Nuri UÇAN	Department of Electrical- Electronics Engineering, Altınbaş University	
Asst. Prof. Dr. Hameed Mutlag Farhan FARHAN	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Gülay ÇİÇEK	Department of Electronics and Automation, İstanbul Beykent University	
Assoc. Prof. Dr. Abdurrahim AKGÜNDOĞDU	Department of Electrical- Electronics Engineering, İstanbul University-Cerrahpaşa	

I hereby declare that this dissertation meets all format and submission requirements for a PhD dissertation

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the above mentioned rules and conduct.

Zahraa Khalid Ahmed AHMED

Signature

XXXXXXXXXX

DEDICATION

To the one whose name I carry with pride.

To whose memory will never leave me.

My dear father, may God have mercy on him.

To the one who was my support and strength, my dearest brother, may God have mercy on him.

To everything I have in existence after God and His Messenger, to the fountain of patience, optimism, and hope, my dear mother.

To those who showed me what is more beautiful than life, my sisters, brothers, and all of their children. To all of them, I present my research.

PREFACE

I thank God Almighty, who has blessed me with reason, religion, health, and determination, and I praise Him for enabling me to complete this scientific study.

Additionally, out of loyalty, appreciation, and gratitude, I sincerely thank my honorable supervisor, Assistant Professor Mesut Çevik, for his support in supervising this study and for his guidance and contributions. Finally, I would like to thank everyone who supported me, including my family, colleagues, and friends.



ABSTRACT

ENHANCED BIO-INSPIRED MANTARAY FOR GENES SELECTION OF ALZHEIMER'S DISEASE USING MICROARRAY DATASETS

AHMED, Zahraa Khalid Ahmed

Ph.D., Electrical and Computer Engineering, Altınbas University,

Supervisor: Asst. Prof. Mesut ÇEVİK

Date: 07/2025

Pages:149

Alzheimer's disease (AD), a prominent neurodegenerative condition, poses significant challenges in its early diagnosis and treatment due to its intricate pathophysiology characterized by the presence of amyloid plaques and neurofibrillary tangles. The recent developments in microarray technology, machine learning, artificial intelligence, and data mining analysis techniques hold considerable potential for comprehending the genetic foundations of AD. Nevertheless, the prediction of AD using machine learning faces several challenges surrounding the analysis of genes expression datasets. Summarized in the “*curse of dimensionality*” caused by the high dimensional disease's microarray datasets, the accurate prediction of disease can significantly suffer from overfitting, bias, and computational demands. To address this challenge, genes selection methods are applied to alleviate the effect of Alzheimer high dimensional genes expression datasets and elevate the overall machine performance.

From critically reviewing the limitations of current genes selection methods associated with AD microarray datasets, this thesis presented a novel bio-inspired wrapper using Mant Ray Foraging Optimization Algorithm MRFO, designed to efficiently explore the vast and large search space of genes expression datasets. The proposed improvements to MRFO (EMRFOGS), including Sign Random Mutation, Best Rank approach, and Hybrid Boundary Function, are not only aimed to reduce the number of genes, but also to increase the disease prediction accuracy. Rigorous tests were performed across multiple machine learning models and six different microarray gene expression datasets to thoroughly evaluate the effectiveness of the proposed approach. Key performance metrics such as

classification accuracy, cross-validation mean accuracy, recall, precision, F1-measure, and AUC curve are used to validate the effectiveness of the proposed method.

Evident from the results, EMRFOGS approach effectively reduced the dimensional space of microarray genes expression datasets and significantly increased the prediction of Alzheimer disease and other performance metrics through efficient identification of relevant genes. While performance improvements varied from one classifier to another given the dimensional differences and statistical relationship distribution among features in each of the six tested datasets. The proposed approach proved to be efficient for clinical applications to provide precise prediction necessary for Alzheimer disease patients. Overall, EMRFOGS performance analysis demonstrated strong and balance exploration-exploitation in navigating complex features space and adaptability to different optimization scenarios. In addition, convergence rate across evaluated classifiers utilizing six genes expression datasets was also significant.

Keywords: Alzheimer's Disease, Genes Selection, Microarray Analysis, Manta Ray Foraging Optimizer (MRFO), Wrapper Features Selection.

TABLE OF CONTENTS

	<u>Pages</u>
ABSTRACT	vii
LIST OF TABLES	xiii
LIST OF FIGURES	xv
ABBREVIATIONS	xviii
1. INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 MACHINE LEARNING AND DATA MINING TECHNIQUES IN MOLECULAR- BASED DIAGNOSIS AND TREATMENT	5
1.3 PROBLEM STATEMENT	5
1.4 GENE SELECTION IN BIOMEDICAL APPLICATIONS	7
1.5 AIM AND OBJECTIVES	9
1.6 THESIS OUTLINE	10
2. LITERATURE REVIEW	11
2.1 INTRODUCTION	11
2.2 GENES SELECTION	11
2.3 MANTARAY OPTIMIZATION ALGORITHM	32
3. THEORETICAL BACKGROUND	42
3.1 INTRODUCTION	42
3.2 ALZHEIMER DISEASE	42
3.2.1 Types of Alzheimer Disease	43
3.3 GENES AND GENE EXPRESSION	44
3.4 ALZHEIMER'S DISEASE AND GENES	46
3.5 MICROARRAY TECHNOLOGY	47

3.5.1 Alzheimer's Disease and Genes Expression	49
3.5.2 Genes Selection	50
3.6 FEATURE SELECTION	51
3.6.1 Techniques of Features Selection	52
3.6.1.1 Filter methods	52
3.6.1.2 Wrapper methods	53
3.6.2 Wrapper Features Selection Techniques Compared to Filter Techniques	54
3.6.3 Weaknesses of Wrapper Features Selection Techniques	55
3.7 MACHINE LEARNING AND CLASSIFICATIONS	56
3.7.1 Support Vector Machine (SVM)	58
3.7.2 Decision Tree	60
3.7.3 Random Forest	62
3.7.4 XGBoost (Extreme Gradient Boosting)	64
3.7.5 Naive Bayes	66
3.7.6 ML Performance Evaluation Measurements	67
3.7.6.1 Evaluation metrics	68
3.7.6.2 Area under the curve	70
3.8 OPTIMIZATION AND NATURE INSPIRED OPTIMIZATION ALGORITHM	71
3.8.1 Mantaray Foraging Optimization Algorithm	72
3.9 TRANSFORMATION FUNCTIONS	76
3.9.1 S-Shaped and V-Shaped Functions	77
4. METHODOLOGY	79
4.1 INTRODUCTION	79
4.2 MOTIVATION	79
4.3 PROPOSED GENE SELECTION MODEL OF ALZHEIMER DISEASE	80

4.3.1 Data Gathering and Preprocessing	80
4.3.2 Selection Model.....	80
4.3.3 Performance Metrics	81
4.3.4 Evaluation.....	82
4.4 SOLUTION REPRESENTATION	82
4.5 ALZHEIMER MICROARRAY GSE DATASETS.....	85
4.6 MRFO OPTIMIZATION ALGORITHM	87
4.6.1 Best Rank	90
4.6.2 Sign Random Mutation	92
4.6.3 Hybrid Boundary Function.....	94
5. TESTS AND RESULTS	96
5.1 INTRODUCTION	96
5.2 EXPERIMENTAL SETTINGS	96
5.3 RESULTS AND DISCUSSION	97
5.3.1 GSE1297	97
5.3.1.1 GSE1297 analysis.....	100
5.3.2 GSE5281	100
5.3.2.1 GSE5281 analysis.....	103
5.3.3 GSE33000	104
5.3.3.1 GSE33000 analysis.....	108
5.3.4 GSE44768	109
5.3.4.1 GSE44768 analysis.....	113
5.3.5 GSE44770	113
5.3.5.1 GSE44770 analysis.....	117
5.3.6 GSE132903	118

5.3.6.1 GSE132903 analysis.....	121
6. CONCLUSION AND FUTURE WORKS.....	122
6.1 CONCLUSION.....	122
6.2 FUTURE WORKS.....	123
REFERENCES	124



LIST OF TABLES

	<u>Pages</u>
Table 2.1: SDA Performance Metrics for the Overlapping Gene Nodes and Pathways between Frontal Cortex and Cerebellum on a Hippocampal Testing Dataset [34].	13
Table 2.2: Details of AD Dataset [35].	14
Table 2.3: The Results of Average Accuracy and A Specified Gene	14
Table 2.4: Performance Evaluation [37].	16
Table 2.5: Summary of the Four Datasets Integrated in the Present Study into One Dataset of 1157 Cases, Each Described by 39,280 Genes [38].	17
Table 2.6: Datasets and Selected Samples Used in the Study [43].	24
Table 2.7: Testing and Training Accuracy for 2000 Genes [44].	27
Table 2.8: Performance Evaluation of the 9 Genes found only by PCA [44].	29
Table 2.9: Performance Evaluation of the Newly Found Genes by PCA and the Overlapped set [44].	30
Table 3.1: AUC Values and Their Interpretation.	71
Table 4.1: S-Shaped and V-Shaped Transformation Functions.	83
Table 4.2: Alzheimer GSE Dataset Dimension.	85
Table 5.1: GSE1297 Performance Measurements before Genes Selection.	97
Table 5.2: GSE1297 Performance Measurements after Genes Selection.	97
Table 5.3: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE1297 with 16379 Features.	99
Table 5.4: GSE5281 Performance Measurements before Genes Selection.	101
Table 5.5: GSE5281 Performance Measurements after Genes Selection.	101
Table 5.6: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE5281 with 54674 Features.	103
Table 5.7: GSE33000 Performance Measurements before Genes Selection.	105
Table 5.8: GSE33000 Performance Measurements after Genes Selection.	106

Table 5.9: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE33000 with 39279 Features.	107
Table 5.10: GSE44768 Performance Measurements Before Genes Selection.	110
Table 5.11: GSE44768 Performance Measurements Before Genes Selection.	111
Table 5.12: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE44768 with 39279 Features.	113
Table 5.13: GSE44770 Performance Measurements before Genes Selection.	114
Table 5.14: GSE44770 Performance Measurements after Genes Selection.	115
Table 5.15: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE44770 with 39279 Features.	116
Table 5.16: GSE132903 Performance Measurements before Genes Selection.	118
Table 5.17: GSE132903 Performance Measurements after Genes Selection.	119
Table 5.18: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE132903 with 42178 Features.	119

LIST OF FIGURES

	<u>Pages</u>
Figure 1.1: Astrocytes of Human Brain with Alzheimer Disease [2]	1
Figure 1.2: PET Scans of Three Brains with Different Conditions: Normal, Mild, And Alzheimer Disease	2
Figure 1.3: Steps of Creating Microarray Chips.	3
Figure 1.4: Features Selection Techniques.	7
Figure 2.1: Proposed Model Flowchart [33]	12
Figure 2.2: Testing (left) and Training (right) Accuracies of Three Classifiers Based on Features Extracted from Three Feature Selection Methods [44].....	27
Figure 2.3: Classification Accuracy of Selected Genes Identified by Three Feature Selection Methods and Trained by Three Classifiers [44]	28
Figure 2.4: Accuracy of the Model with Genes only Found by Each Feature Selection Method When Trained Upon Three Classifiers; SVM with Gaussian and Linear Kernel and Random Forest [44]	29
Figure 2.5: Proposed Gene Selection Model [45]	31
Figure 3.1: Comparison of a Healthy Brain with a Brain with Alzheimer Disease.	44
Figure 3.2: Transcription vs Translation.....	45
Figure 3.3: The Amyloid Cascade Hypothesis.	47
Figure 3.4: Spotted cDNA-based Microarray.....	48
Figure 3.5: Classification Techniques in Supervised Learning.	57
Figure 3.6: A Decision Tree Example Using a Binary Target Variable Y	61
Figure 3.7: Decision Trees Represented Using a Sample Space View.	62
Figure 3.8: A non-linear Function (blue) and its Approximation Using Piecewise Linear Segments (red).....	76
Figure 3.9: The Sigmoid Function's Elongated 'S'-shaped Curve	77

Figure 3.10: V-shaped Transfer Function	78
Figure 4.1: Proposed Gene Selection Model.	80
Figure 4.2: S-Shaped and V-Shaped Transformation Functions [117].	83
Figure 4.3: Solution Encoding Steps in the Proposed Algorithm.	85
Figure 4.4: Flowchart of MRFO Gene Selection Algorithm.	88
Figure 4.5: MRFO Pseudocode and Proposed Enhancements	90
Figure 4.6: MRFO Chain Foraging [102].	91
Figure 4.7: MRFO Cyclone Foraging [102].	92
Figure 5.1: Convergence of Minimum and Maximum Fitness Values of (a) EMRFOGS-NB, (b) EMRFOGS-XGBoost, on GSE1297.	98
Figure 5.2: Minimum and Maximum Number of Features Selected by EMRFOGS-RF on GSE1297.	99
Figure 5.3: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE5281.	101
Figure 5.4: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-SVM on GSE5281.	102
Figure 5.5: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-XGBoost on GSE5281.	102
Figure 5.6: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE5281.	103
Figure 5.7: ROC for: (a) EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.	104
Figure 5.8: Minimum and Maximum Number of Features Selected by: (A) EMRFOGS-NB, (B) EMRFOGS-SVM on GSE33000.	106
Figure 5.9: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-RF on GSE33000.	107

Figure 5.10: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-XGBoost on GSE33000.	108
Figure 5.11: ROC for: (a) EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.....	109
Figure 5.12: Minimum and Maximum Number of Features Selected by EMRFOGS-NB on GSE44768.....	111
Figure 5.13: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE44768.....	112
Figure 5.14: Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE44768.	112
Figure 5.15: Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE44770.	114
Figure 5.16: Minimum and Maximum Number of Features Selected by EMRFOGS-SVM on GSE44770.....	115
Figure 5.17: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-RF on GSE44770.....	116
Figure 5.18: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-SVM on GSE44770.....	116
Figure 5.19: ROC for: (a)EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.....	117
Figure 5.20: Minimum and Maximum Number of Features Selected by EMRFOGS-NB on GSE132903.....	120
Figure 5.21: Minimum and Maximum Number of Features Selected by EMRFOGS-RF on GSE132903.....	120
Figure 5.22: Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE132903.	121

ABBREVIATIONS

AD	:	Alzheimer Disease
AUC	:	Area Under the Curve
EMRFOGS	:	Enhanced Manta Ray Forging Optimizer for Genese Selection
GEO	:	Gene Expression Omnibus
GSE	:	Genomic Spatial Event
MRFO	:	Manta Ray Foraging Optimizer
NB	:	Naive Bayesain
PET	:	Positron Emission Tomography
RF	:	Random Forest
SML	:	Supervised Machine Learning
SVM	:	Supprot Vector Machine
XGBoost	:	Extreme Gradient Boosting

1. INTRODUCTION

1.1 INTRODUCTION

Alzheimer's disease (AD), also known as senile dementia, is a progressive neurological illness that manifests itself more severely as time passes. Alzheimer's disease (AD), which is the most well-known kind of neurodegenerative illness, has a major negative impact on a patient's ability to carry out regular day-to-day activities. The characteristic pathologic abnormalities of Alzheimer's disease (AD) include the generation of extracellular amyloid plaques by aberrant amyloid beta build-up, the loss of neuronal tissue due to gliosis proliferation, and the development of intracellular neurofibrillary tangles caused by tau hyperphosphorylation [1]. Both of the pathogenesis and the ethology of Alzheimer's disease remain unknown to this time.

The chemical changes that cause astrocytes, a kind of brain cell, to become reactive astrocytes and show a very visible morphological change in response to a stressful situation are referred to as reactive astrocyte morphology, see Figure 1.1 [2]. The term "reactive astrocyte morphology" is used in the field of neurobiology and disease pathology to describe the biochemical and structural alterations that occur in the brain and spinal cord's star-shaped glial cells in reaction to stress, be it an injury, inflammation, or a neurodegenerative illness like Alzheimer's. Astrocytes have several roles in the normal brain, including facilitating neuronal metabolism, controlling the integrity of the blood-brain barrier, and preserving the balance of ions in the extracellular space [3][4].

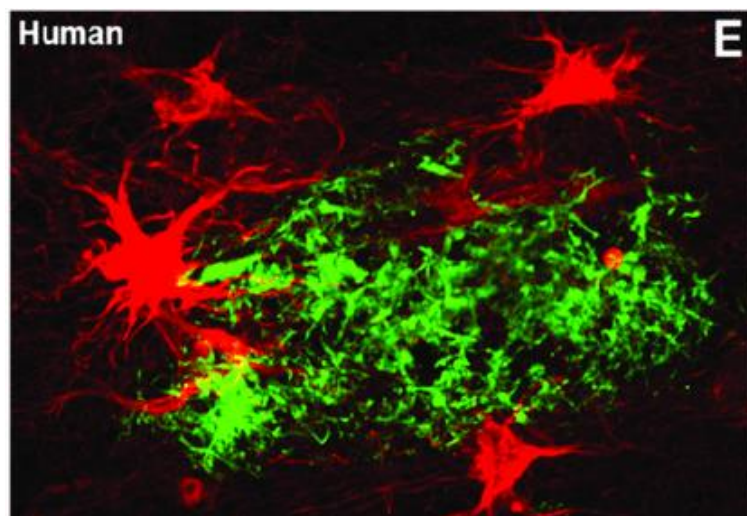


Figure 1.1: Astrocytes of Human Brain with Alzheimer Disease [2].

Despite the high prevalence of Alzheimer's disease, it remains one of the most extensively studied illnesses [5][6]. The reason why neurons in diseased brains have communication issues with astrocytes as well as other neurons remains a mystery [7]. The high level of attention given to reactive astrocytes is due to the dual role they can play: on one hand, they may protect neurons and contain the damage, but on the other hand, chronic or dysregulated astrocyte reactivity can exacerbate neuronal dysfunction and contribute to disease progression. The Positron Emission Tomography (PET) scans of human brain from Science Photo Library in Figure 1.2 shows the progressive decrease of activity in a brain with AD compared to two brains; normal and mild cognitive impairment [8]. Pronounced areas of blue and green indicate significant reduction of brain activity in comparison to normal brain with much uniformly distributed across most areas of the brain. Alzheimer's disease is characterized by the accumulation of amyloid-beta plaques and tau tangles, which interfere with the functioning of neurons. These diseases often begin in places such as the hippocampus and then extend to other parts of the cerebral cortex.

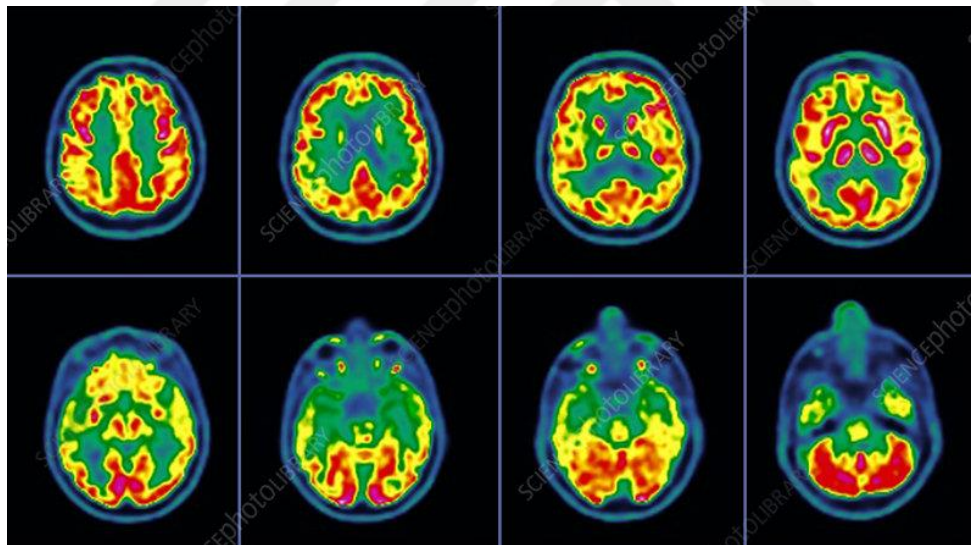


Figure 1.2: PET Scans of Three Brains with Different Conditions: Normal, Mild, And Alzheimer Disease.

The exact mechanisms of how astrocytes become reactive and the consequences of their reactivity are subjects of intense research, partly because of the potential to develop therapies that modulate astrocyte functions to treat Alzheimer's and other neurological diseases [7].

The progress of RNA and DNA microarray technology has fundamentally transformed the area of genomics and molecular biology. Microarrays, or what is commonly referred to as DNA chips or biochips, are compact and solid platforms where DNA sequences are systematically immobilized. Using DNA microarrays, it is now possible to routinely analyze the biological and molecular status of a cell by monitoring the concurrent expression of a large number of genes [9][10].

Such process begins by hybridization, whereby tagged RNA or DNA probes obtained from a sample of interest are permitted to interact with these immobile DNA sequences. Fluorescent dyes are often used as labels, and after any probes that are not properly attached are removed, the microarray is scanned to identify the fluorescent signals. The fluorescence intensity at each point on the array directly reflects the amount of the nucleic acid sequence in the sample, which is directly related to the expression level of the corresponding gene [9][10][11]. Figure 1.3 for instance shows the steps of creating microarray chip for normal cell and another cell with cancer disease.

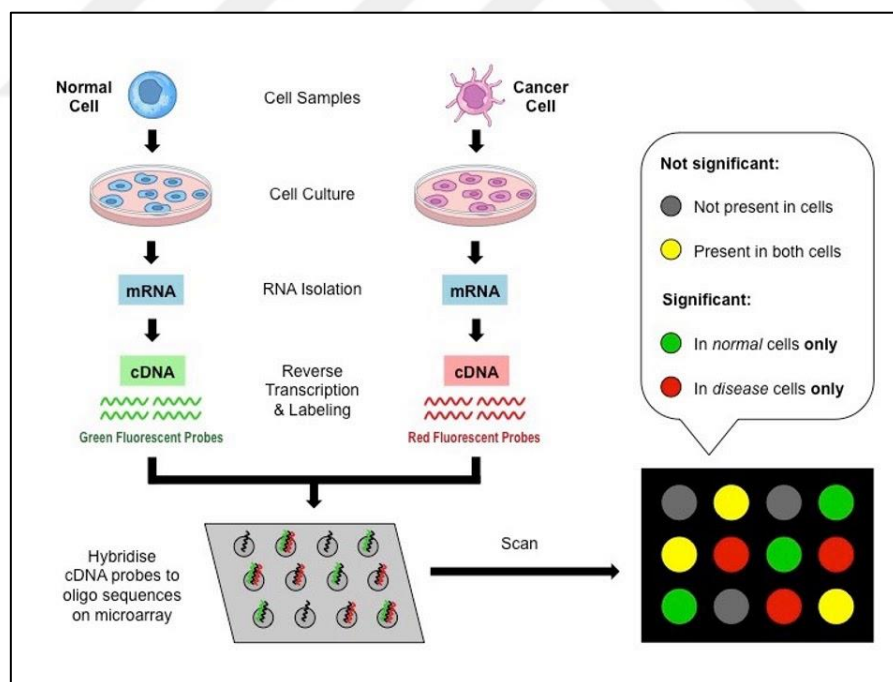


Figure 1.3: Steps of Creating Microarray Chips.

Utilizing such high-throughput method, researchers can assess gene expression profiles, identify co-expressed genes through comparison of expression patterns, and discovers what known as Biomarkers, i.e. genes that serve as the markers for disease valuable in diagnosis

and monitoring treatment efficacy [11]. Information derived from microarrays chips can also be used to categorize biological samples according to how all or a subset of the genes on the microarray express themselves. It is possible to classify organisms using all of their genes or just a few of them. Genes with differential expression and groups of co-expressed genes can be used to make hypotheses about the biological pathways involved in a particular biological process [14]. Additionally, co-expressed gene clusters can be used as a jumping off point to research the regulatory mechanisms underlying the co-expression and deduce the functional relationship of a clustered gene. To classify cancers and other subclasses of malignancies in the context of tumor classification, gene expression patterns have been used as biomarkers [13].

Multiple technologies are used to measure the levels of mRNA expression by different kinds of microarrays; a comprehensive description of these technologies is outside the scope of this study. The analysis of expression data derived from microarrays will be the main focus of this section. Many current initiatives are adapted in this direction. The results of microarray analysis have occasionally found their way into numerous important applications in biology and medicine.

In summary, it is important to keep in mind that genes expression serves two very important tasks: it contributes to the biological mechanism underlying the disease (demonstrating the biological relevance of a gene expression) and it is a crucial element in the development of successful diagnostic prediction frameworks (demonstrating the diagnostic relevance of a gene expression). However, when developing diagnostic framework for diseases using molecular genes expression microarrays, not all genes are necessarily useful or relevant for precise diagnostics process. It is crucial to differentiate the correlation and causation between gene expression levels in microarray chips and disease states. In other words, the presence or absence of gene expression in microarray chip doesn't mean the presence or absence of connection between this particular gene and the disease itself. Therefore, the development of gene-based diagnostic frameworks necessitates research, empirical biological and clinical evidence, and accurate validation techniques [14].

1.2 MACHINE LEARNING AND DATA MINING TECHNIQUES IN MOLECULAR-BASED DIAGNOSIS AND TREATMENT

The adoption of data mining techniques has quickly become one of the most used approaches in the field of microarray analysis. The utilization of machine learning and data mining techniques transformed microarray analysis for molecular diseases diagnosis and treatment [15]. For instance, in oncology, machine learning models can help in identifying oncogenic mutations and predicting patient response to specific therapies, such that enabling tailored and effective treatment plans for different patients. Through use of machine learning algorithms such as supervised learning, clustering techniques, and regression, vast amount of microarrays data can be processed efficiently [17]. With each microarray measuring the expression levels of hundreds and thousands of genes across multiple samples and many patients, machine learning algorithms can identify patterns and discover complex relations among different genes expression levels quickly and much efficiently compared to standard clinical approaches [17].

Supervised Machine Learning (SML) is a major branch of machine learning that involves training various algorithms on labeled datasets. In other words, the correct output of each sample is provided for data sample. SML is then learn or attempt to discover the relationship between each input and the correspondent output. Inputs are comprised of series of values known as features. Supervised learning model iteratively adjust their weights and biases to minimize prediction errors and increase generalization of final model. Unseen data inputs are then used to measure model accuracy. Labeled microarray sampled from number of patients can be used after labeling in gene expression classification-based diagnosis and treatment monitoring. Supervised learning models can predict clinical outcomes based on gene expression data. For instance, these models can predict patient prognosis, response to specific treatments, or disease recurrence. This predictive capability is crucial for personalized medicine, where treatment plans are tailored based on individual genetic profiles [16].

1.3 PROBLEM STATEMENT

Despite the early success of supervised learning and data mining methods, these techniques of microarray analysis still face significant challenges. First and most importantly, is the curse of dimensionality [18][19][20][21]. This phenomenon refers to a series of problems

that arise in machine learning when processing high-dimensional spaces datasets, and becomes particularly significant in the context of microarray genes datasets. To simply summarize it, high-dimensional data become sparse when the volume of dimensional space increase, this sparsity renders it difficult for machine learning algorithms to detect patterns among data points and therefore forming an inaccurate decision. In addition to computational complexity to process high number of dimensions and delay in medical decisions and prompt disease diagnosis and treatments, high number of features/genes compared to only fewer data samples increases the risk of learning model overfitting. This will lead to poor classification performance of unseen test data samples due to weak learning generalization of the training samples. Second, the large number of features/genes in microarray chips are not all typically relevant to the problem at hand, in other words, some genes in the microarray profile are unrelated to the disease status of the tissue. Training of irrelevant data features generally can cause low classification accuracy [22].

To alleviate these challenges, feature selection and dimensionality reduction techniques are essential [23][24][25]. Feature selection involves identifying and retaining only those features/genes from microarray vast features space that are most relevant to the predictive model, thereby improving model accuracy, reducing overfitting, and enhancing model interpretability. In addition, reducing data dimensional space exponentially decrease the computation complexity of microarray datasets, this can lead to efficient processing time and use of complex learning approaches such as ensemble learning to enhance the discovery of complex genes and biomarkers relationships.

In the context of Alzheimer diagnosis for instance, genes selection significantly contributes to specific disease or to the success of treatment. Understanding genes expression and its relation to diseases is crucial for multiple reasons, first, researchers can develop targeted therapies that specifically modulated to these genes. Second, increasing the treatment efficiency through personalized medicine where treatments thereby can be tailored to the genetic makeup for each individual patient. Last, the discovery of genes biomarkers of any disease can enable more accurate and precise detection of diseases. In addition, the identification of relevant genes for a given disease can lead to the creation of low-cost diagnostic assays for a limited number of genes. These assays might then be improved further so that they can be used extensively in clinical settings [26][27][28].

In summary, genes selection has the potential to alleviate the "curse of dimensionality" linked to supervised machine learning and analysis of microarray genes expression datasets. Thereby, paves the way for utilization of various learning algorithms with emphasis on precise classification while improving the overall status of molecular treatments and diagnosis.

1.4 GENE SELECTION IN BIOMEDICAL APPLICATIONS

In the realm of machine learning, there are various methods for effective selection of features for a given dataset. These methods, depicted in Figure 1.4, are categorized into three main sets, namely: filter, wrapper, and embedded [29][30]. Each approach has its set of advantages and disadvantages while their performance varies from one dataset to another and from one classifier to another. For instance, filter based approaches such as information gain and chi-square are known for their computational efficiency and scalability to very large datasets. However, filter-based approaches treat features independently and don't consider the interaction among features. This contribute to weak discovery of complex relationships among features which is substantially important in genes and biomarkers selection for a specific disease or drugs development. Embedded methods on the other hand are considered more accurate when compared to filter-based since these methods integrate feature selection as a part of the algorithmic training process, such as L1 and L2 regularization. Thereby, the subsets of selected features are iteratively tested internally within the learning model [29][30].

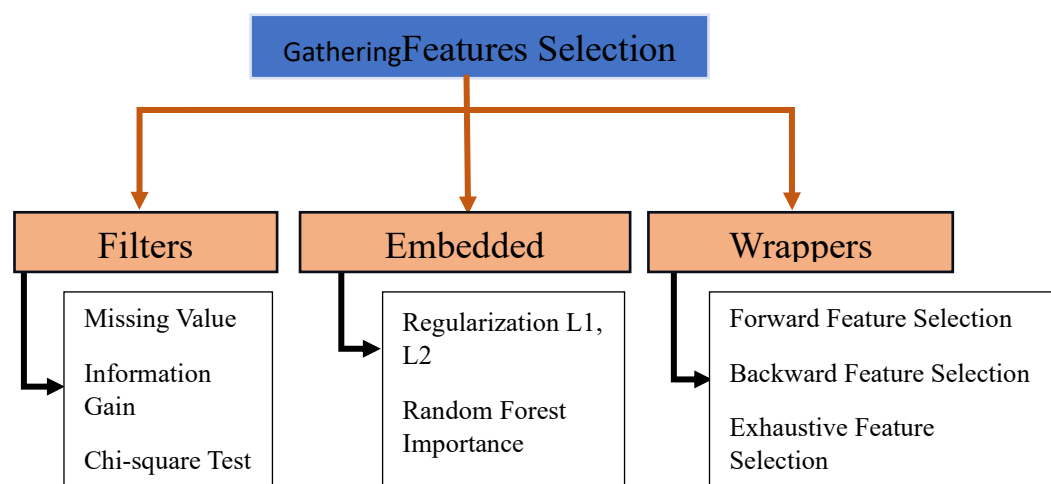


Figure 1.4: Features Selection Techniques.

While these methods are prone to overfitting, the major drawback of embedded approach is the evaluation is tied and limited to the specific model being trained with no support of external learning algorithms. Such that, limit the generalization of accuracy of the final selected features to other types of learning models.

Wrappers, in comparison, are more sophisticated category of features selection techniques. Wrappers are comprised of a search strategy responsible for exploring the features space of possible features and an evaluation strategy to evaluate the selected features according to various metrics such as classification accuracy, cross-validation accuracy, Area Under the Curve (AUC), or a similar criterion. Wrappers in general are models independent, search strategies and learning models can be appropriately selected according to the case at hand [30][31]. A critical challenge faces utilization of wrappers is the computational expenses involved in exploring the vast search space of possible features combinations. Common search strategies like Exhaustive Search, Sequential Forward and Backward Selection, and Recursive Feature Elimination are slow and weak in discovery of optimal subset of features due to iterative and sequential nature of features elimination render them to be computationally infeasible for large microarray datasets [29][30][31].

To address such challenge, optimization algorithms as a search strategy for wrappers features selection are utilized to efficiently and effectively enhance the search process. The exploration and exploitation of optimization algorithms, such as genetic algorithms (GAs), particle swarm optimization (PSO), and ant colony optimization (ACO), can leverage the search process of large microarray datasets through reduction of the time and computational resources required in comparison to aforementioned strategies. Furthermore, effective and strong balance between exploration and exploitation ensures the convergence to optimal or near-optimal subset of features with better generalizability and performance of the utilized machine learning model [31][32]. In conclusion, adoption of optimization algorithms to overcome search strategy challenges in wrapper methods represents has a significant impact on features selection techniques with particular emphasis on scenarios involving large microarrays datasets with extensive features space [32].

To gain more understanding of wrapper-based features/genes selection method, it is essential to describe wrappers three fundamental components:

- a. Selection Strategy: this refers to approach employed to explore the features space and identify the optimal set of features from a larger set. For large space scenarios such as microarray datasets, optimization algorithms are often used to efficiently navigate the search space.
- b. Evaluation Metrics: a list of criteria or objectives used to assess the selection performance. Metrics typically include learning model accuracy, crossover accuracy, AUC, precision, recall, and others. However, evaluation metrics can be similar in the sense they are related to same aspect of model performance such as performance increase and reduction of number of features, or contradictory such as reducing the number of features while increasing classification accuracy.
- c. Evaluation Model: this refers to the learning model used to evaluate the selection of subset of features by selection strategy according to defined evaluation metrics. Evaluation model selection is independent from selection strategy method, such that, promote utilization of complex learning approaches such as ensemble learning to enhance the discovery of complex genes and biomarkers relationships.

1.5 AIM AND OBJECTIVES

This thesis aims to enhance the prediction of Alzheimer disease through enhanced Bio-inspired wrapper-based genes selection method of Alzheimer high-dimensional microarray datasets. Thesis objectives are detailed as follow:

- a. Critically review the current challenges of gene selection in microarray datasets associated with Alzheimer disease.
- b. Design and develop novel wrapper-based gene selection algorithm predicated on bio-inspired optimization algorithm known as Manta Ray Foraging Optimizer (MRFO). MRFO is large-scale optimization algorithm efficient for large features space of microarray datasets of Alzheimer disease.
- c. To investigate the enhancement of MRFO exploration and exploitation to increase the convergence speed, and the search of high-quality optimal solution to improve predication accuracy and diagnosis of Alzheimer disease using microarray genes expression datasets.
- d. To evaluate the enhanced MRFO using different machine learning models to efficiently measure the optimization performance across multiple learning scenarios

given the theoretical and arithmetic differences of each machine learning algorithm. Such analysis can help determine suitable classifier to microarray datasets used in this study.

- e. To validate and test the proposed gene selection optimization algorithm on six different microarray genes expression datasets. Evaluation metrics include classification accuracy, 5-fold cross validation mean accuracy, recall, precision, F1-measure, and AUC curve.

1.6 THESIS OUTLINE

The study of genes selection of Alzheimer microarray expression dataset presented in this thesis is divided into six chapters. Chapter One covers introduction, problem statement, the proposed objectives. Chapter Two thoroughly examines the landscape of literatures related to genes selection. Chapter Three depicts general theoretical background related to Alzheimer disease, methods of features selection, supervised machine learning, Meta Ray optimization algorithm, and encoding transformations functions. While Chapter Four and Five detail the proposed genes selection methodology and delve into thorough examinations and testing of proposed approach. Last, Chapter Six present the conclusion of and further future work remarks.

2. LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, an extensive examination of the existing literature concerning the selection of genes/features associated with Alzheimer's disease is provided. The focus of this review is primarily on studies published between 2020 and 2023. This section consists of two main parts: first, a comprehensive analysis of genes selection methodologies and related research with particular attention to the utilization of machine learning techniques and genes selection methods. Second, a thorough review of recent studies on the Mantaray foraging optimization algorithm is presented, exploring the recent exploration, exploitation, and convergence speed enhancement techniques of the algorithm.

2.2 GENES SELECTION

The study led by Cuihua Zou et al. primarily focused on identifying Differentially Expressed Genes (DEGs) in Alzheimer's disease (AD) by analyzing four datasets. In GSE125583, 12,555 DEGs were identified, with significant upregulation (6,584) and downregulation (5,971). GSE118553, GSE5281, and GSE122063 also revealed a large number of DEGs, providing valuable insights into gene expression changes in AD [33]. The study highlighted 1,855 intersecting DEGs across all four datasets, enriched in key pathways related to RAS, neurotrophin, AMPK signaling, and the cell cycle. Moreover, these DEGs were associated with critical biological processes, particularly memory and learning, shedding light on the molecular mechanisms underlying AD [33].

Utilizing advanced analysis techniques such as Gene Set Enrichment Analysis (GSEA) and Receiver Operating Characteristic (ROC) analysis, the research evaluated diagnostic models. The LASSO model exhibited superior diagnostic performance compared to others, identifying eight feature genes (ATP2B3, BDNF, DVL2, ITGA10, SLC6A12, SMAD4, SST, and TPI1) with potential as biomarkers for AD [33].

The study employed Receiver Operating Characteristic (ROC) analysis to assess the diagnostic performance of four models using two datasets, GSE125583 and GSE118553. In GSE125583, the area AUC values were 0.81 for RF, 0.87 for LASSO, 0.95 for logistic regression, and 0.96 for Gradient Boosting Machine (GBM), indicating good performance for all models. In GSE118553, the AUC values were 0.78 for RF, 0.91 for LASSO, 0.55 for

logistic regression, and 0.67 for GBM. The LASSO model demonstrated the best performance in both datasets compared to the other models, a flowchart of the study is shown in Figure 2.1 below [33].

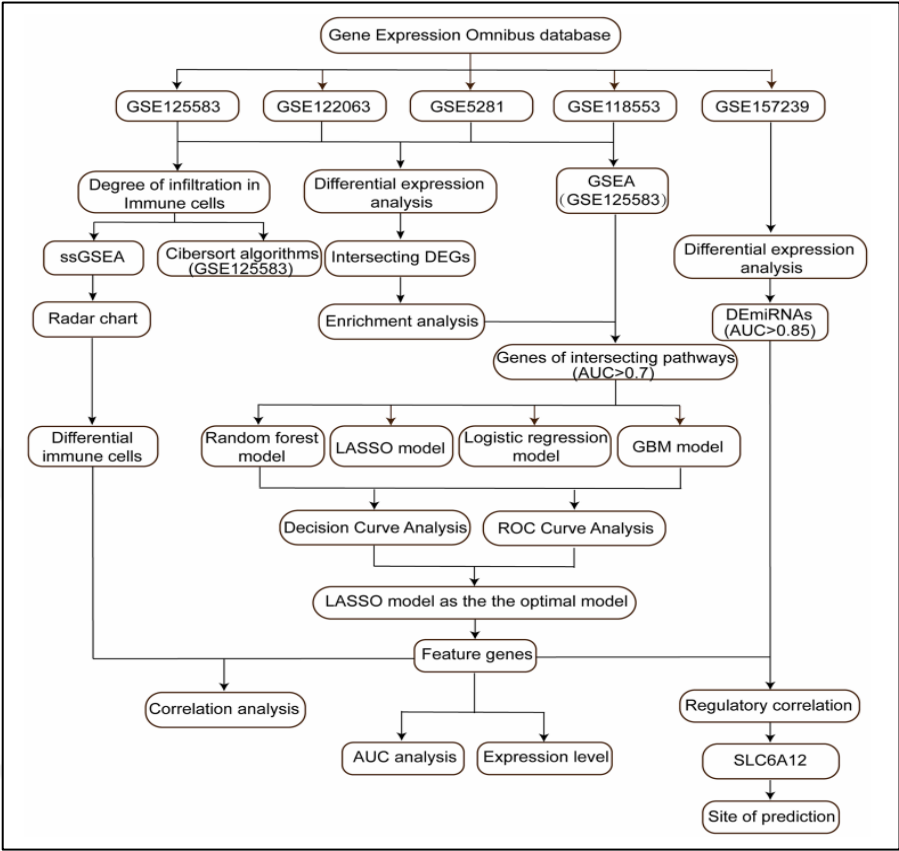


Figure 2.1: Proposed Model Flowchart [33].

Intriguingly, the study also explored the role of microRNAs (miRNAs) in AD, identifying specific DEmiRs and elucidating their regulatory impact on feature genes like ATP2B3, TPI1, SLC6A12, and SMAD4. Notably, SLC6A12 emerged as a pivotal hub gene with elevated expression in AD, regulated by hsa-miR-3176, further highlighting its potential significance in AD pathogenesis [33]. Overall, this research significantly contributes to understanding the molecular landscape of AD and offers promising insights for the development of targeted AD therapies. The study by Caitlin A. Finney et al. involved screening 1010 datasets in GEO, identifying 14 datasets meeting inclusion criteria for meta-analysis. After analyzing metadata and sample sizes, 11 datasets were excluded due to insufficient sample size. Two brain regions, the frontal cortex and cerebellum, were included from three identified datasets: GSE44770, GSE33000, and GSE44768. These datasets comprised samples from AD patients and healthy controls, with all donors being

Caucasian [34]. Unsupervised machine learning using PCA showed distinct clustering between AD patients and healthy controls in the frontal cortex and cerebellum datasets. The top dysregulated genes in AD were identified using PCA. Subsequent unsupervised machine learning using k-means clustering and Gene Ontology (GO) analysis characterized biological functions of gene clusters in both brain regions. Supervised machine learning revealed top signaling pathways for AD in both brain regions, with mitochondrial energy pathways being highly significant. The analysis was also extended to the hippocampus, confirming the generalizability of findings across different brain regions. Overlapping gene nodes and pathways were identified between the frontal cortex and cerebellum, emphasizing the significance of specific genes like ATP5L and ATP5H in AD-related pathways [34].

The study tackled the challenge of identifying novel gene candidates for AD through a meta-analysis of microarray data from the frontal cortex and cerebellum of AD patients. Advanced artificial intelligence, employing both unsupervised and supervised machine learning, was utilized to pinpoint potential genes for future investigation. Table 2.1 below shows the shrinkage discriminant analysis (SDA) models each consisting of the pathways and respective genes [34].

Table 2.1: SDA Performance Metrics for the Overlapping Gene Nodes and Pathways between Frontal Cortex and Cerebellum on a Hippocampal Testing Dataset [34].

Cluster	Specificity	Sensitivity	Accuracy	AUC
Mitochondrial energy, ATP, and oxidative phosphorylation	0.6	1	0.73	0.96
Mitochondrial cellular biosynthesis	0.75	1	0.87	0.96
Signaling	0.67	1	0.80	1

The study highlighted the critical role of mitochondria in AD development and progression, showcasing potential common mechanisms such as dysregulated ATP synthase subunits. These findings offer valuable insights into AD etiology and potential targets for future research and therapeutic interventions [34].

Shaymaa Taha Ahmed and Suhad Malallah Kadhemi showcased various methods that have been employed to predict AD and extract relevant genes, such as NPO, Rhinoceros Search Algorithm, MLPNN, SVM, RF, KNN, IG, PCA, and ANN. The study emphasizes Information Gain (IG) as a common filtering strategy due to its low computational cost but

notes its limitation in overlooking gene interactions. In contrast, NPO, known for considering gene interactions, is presented as an effective gene selection technique [35]. The proposed approach integrates IG for gene ranking, considers expression to reduce redundancy, employs NPO for the final gene set selection, and leverages SVM for accurate classification of AD and non-AD individuals based on compressed genes. NPO's effectiveness lies in its balanced exploration and exploitation capabilities, contributing to efficient solutions through swarm-based exploration of the state space [35].

The study highlights the application of the proposed gene selection technique across five open datasets as shown in Table 2.2, demonstrating the importance of excluding irrelevant genes to enhance classification performance. The evaluation of the model's performance through five-fold cross-validation showcases its robustness and shows the superior performance of the IG-based classification model [35].

Table 2.2: Details of AD Dataset [35].

Datasets	Alzheimer Disease AD	Normal	MCI	Total No.	Age	Features (genes)
GSE63060, GSE63061	245	182	142	569	Range[55 to 91]	16382
GSE2581	87	74	NA	161	Range[60 to 90]	54675
GSE1297	22	9	NA	31	Range[55 to 88]	22283
GSE132903	97	98	NA	195	Range[57 to 92]	42179
GSE33000	310	157	157	624	Range[50 to 90]	39280

The research emphasizes the critical task of distinguishing individuals with AD from healthy subjects through gene expression analysis in both blood and brain samples. The proposed SVM model based on the NPO algorithm demonstrates promise in diagnosing AD, particularly in its early stages, offering valuable clinical contributions for disease diagnosis and progression monitoring. Table 2.3 below shows accuracy of the datasets using the SVM+NPO [35].

Table 2.3: The Results of Average Accuracy and A Specified Gene.

Proposed Model	Datasets	Accuracy (%)	Selected Gene
SVM+NPO	GSE1297	93.0031	250
	GSE5281	96.8944	1115
	GSE63060 and GSE63061	87.0005	30
	GSE132903	92.4124	1100
	GSE33000	92.0961	322

The Metaheuristic Radical Algorithm (MRA) within NPO facilitates radical information exchange among swarm leaders, represented by the best local solutions. NPO's application with only two structural parameters simplifies its use. The research showcasing NPO's versatility, applicable across domains like artificial intelligence and machine learning. Its efficacy in tackling feature selection and modifying neural network weights highlights its potential for advancing machine learning techniques, with prospects of developing multiple objective NPO variants for enhanced capabilities [35].

In a study by Yongxing Lai et al. using the GSE33000 dataset, researchers analyzed expression profiles related to cuproptosis regulators and immune characteristics in Alzheimer's disease. Utilizing 310 Alzheimer's disease samples, they explored molecular clusters based on cuproptosis-related genes and studied immune cell infiltration. The WGCNA algorithm identified cluster-specific differentially expressed genes. The study compared various machine learning models (RF, SVM, GLM, and eXtreme Gradient Boosting) and evaluated their predictive efficiency using receiver operating characteristic (ROC) curves. RF model stood out with the highest AUC at 0.9829, showcasing its exceptional ability to distinguish patients with different clusters. Validation of the 5-gene prediction model (MYT1L, PDE4D, SNAP91, NPTN, and KCNC2) on external datasets consistently showed strong AUC values, with an AUC of 0.8529 in the GSE5281 dataset and 0.8333 in the GSE122063 dataset. These AUC values underscore the effectiveness of the model in distinguishing AD from normal individuals, emphasizing its outstanding pathological diagnostic value [36].

Nivedhitha Mahendran et al. examined two datasets, GSE33000 and GSE44770. The study addressed missing data with KNN imputation, applied log transformation, and used volcano plots to assess gene significance. These plots portray the relationship between p-values and fold changes. Significance was defined by a p-value threshold of 0.05 [37]. Following data preprocessing, the top 250 genes, deemed most relevant for AD, were selected. Ensemble-based feature selection was then employed using the function variation method, which incorporates five filter techniques: signal-to-noise ratio, ReliefF, chi-squared, mutual information, and information gain. The ranked features from each technique were merged into a final ranked list through mean aggregation, forming the basis for AD classification [37].

For classification, the Deep Multi-Layer Perceptron (DMLP) was employed and refined with stopping criteria to ensure convergence and prevent overfitting. Starting with the top 100 features from the ranked list, features were progressively added to the model until no further accuracy improvement was observed, resulting in the selection of 24 final features for AD classification. The DMLP parameters were optimized using the Grid search algorithm, considering factors such as the number of epochs, activation function (ReLU), dropout regularization (0.7), number of neurons in the hidden layer, and the number of hidden layers (3 hidden layers). A convergence-based stopping criterion was introduced to prevent overfitting, where training and testing accuracy were compared after 15 epochs, and training was halted upon convergence [37].

Results were compared against other classification models, including MLP, LDA, SVM, and the proposed ensemble approach. Table 2.4 summarized the best performance percentages achieved across these approaches through five-fold cross-validation. The ensemble approach and the Deep MLP consistently outperformed other models. Notably, the shallow MLP demonstrated superior performance, especially when combined with the proposed feature selection approach. Introducing depth into the shallow MLP further enhanced results [37].

Table 2.4: Performance Evaluation [37].

Feature Selection	Classification Approach	Accuracy	Sensitivity	Specificity	F- score	AU- ROC
mRmR	LDA	0.715	0.721	0.718	0.719	0.725
	SVM	0.747	0.731	0.744	0.735	0.730
	MLP	0.851	0.847	0.846	0.859	0.84
	DMLP	0.912	0.917	0.907	0.904	0.919
SVM-RFE	LDA	0.768	0.761	0.751	0.769	0.75
	SVM	0.794	0.79	0.782	0.786	0.797
	MLP	0.887	0.883	0.876	0.871	0.88
	DMLP	0.921	0.917	0.919	0.928	0.923
LASSO	LDA	0.807	0.817	0.809	0.811	0.81
	SVM	0.827	0.814	0.819	0.824	0.815
	MLP	0.871	0.875	0.88	0.877	0.881
	DMLP	0.937	0.931	0.925	0.933	0.92
Proposed Ensemble Approach	LDA	0.854	0.864	0.869	0.841	0.866
	SVM	0.891	0.897	0.88	0.897	0.892
	MLP	0.958	0.959	0.947	0.949	0.95
	DMLP	0.972	0.979	0.970	0.981	0.987

Modern data technologies, such as microarrays, provide valuable insights into diseases like AD. However, the large volume of data generated presents challenges, such as the High-Dimensional, Low Sample Size (HDLSS) problem. To overcome this challenge, machine learning techniques, including feature selection, are instrumental [37].

In a study led by Aliaa El-Gawady et al., researchers explored a framework designed for predicting AD using a dataset comprising 1157 cases (Table 2.5) and 39,280 genes. This dataset was curated by aggregating data from various smaller public datasets. The dataset was thoughtfully divided into two subsets: 1000 cases earmarked for training and testing, utilizing a 10-fold cross-validation approach repeated 30 times, and a separate set of 157 cases designated for the final validation phase. Throughout the training and testing stage, the framework meticulously identified 1058 genes deemed the most pertinent for AD prediction, SVM model consistently showcasing the highest accuracy. The ultimate validation step, which involved 157 previously unseen cases, yielded remarkable performance metrics: sensitivity (recall) at 0.97, specificity at 0.97, precision at 0.98, kappa index at 0.945, AUC at 0.972, and overall accuracy at 0.975 [38].

Table 2.5: Summary of the Four Datasets Integrated in the Present Study into One Dataset of 1157 Cases, Each Described by 39,280 Genes [38].

Dataset ID	GSE33000	GSE44770	GSE44768	GSE44771
Type	Prefrontal cortex	Prefrontal cortex	Cerebellum	Visual cortex
Number of AD cases	310	129	129	129
Number of normal cases	157	101	101	101
Total number of cases	467	230	230	230

Nonetheless, the need for improved sensitivity to establish a more specific blood signature for AD remained a prominent focus. To address this, the study leveraged three independent datasets, ANM1, ANM2, and ADNI, to distinguish AD from CN. Various gene selection methods, including variational autoencoder, transcription factor analysis, hub gene identification, and convergent functional genomics (CFG), were employed to identify the most informative genes. For the critical task of classification, five distinct machine learning models, encompassing SVM, RF, logistic regression (LR), L1-regularized LR (L1-LR), and DNN, were brought into play. The resulting outcomes were AUC values of 87.4%, 80.4%, and 65.7% for ANM1, ANM2, and ADNI, respectively [38].

This comprehensive analysis revealed 789 genes with differential expression that were shared between blood and brain tissues, highlighting their potential relevance in AD research. To leverage this potential, the scientists utilized the Least Absolute Shrinkage and Selection Operator (LASSO) regression for gene selection. Additionally, they employed logistic ridge regression (RR), SVM, and RF models for classification, resulting in an impressive accuracy of 78.1% in discriminating between AD cases and control subjects [38]. Additionally, the study explored research efforts that capitalized on gene expression data obtained from multiple brain regions to unearth potential diagnostic biomarkers for AD. Another intriguing facet of the study revolved around the integration of gene expression and DNA methylation datasets, culminating in the creation of a multi-omics dataset for AD prediction through the application of a deep neural network (DNN). To extract the most informative features from this complex dataset, PCA was employed and t-stochastic nearest neighbor techniques. Furthermore, the study shed light on the utilization of blood gene expression data from different cohorts. This part of the research entailed the application of recursive feature elimination for gene selection and the use of RF for classifying AD cases. These efforts yielded an AUC of 72.4% and an accuracy of 65.7% [38].

Genes that boasted the highest average values were subsequently cherry-picked, based on user-defined thresholds, and funneled into multiple machine learning models to evaluate their process in classification. The study unearthed a recurring theme: SVM consistently outperformed other models. This trend persisted across three different sets of genes, each selected based on distinct criteria: 700 genes with the highest χ^2 value, 1000 genes with the highest ANOVA (F statistic) value, and 1700 genes with the highest Mutual Information (MI) value. By meticulously conducting pairwise intersections of these gene sets, the researchers uncovered common genes. These common genes, when unified, coalesced into a set of 1058 genes, christened $GU \cap$, which unequivocally proved to be the most relevant for predicting AD when the SVM model was employed [38].

Carol J. Huseby's et al. study emphasizes the need for a simple, affordable, and noninvasive diagnostic tool, such as a blood-based test. The study's goal is to identify a concise set of blood transcripts capable of distinguishing various neurodegenerative diseases, including Alzheimer's, Parkinson's, Huntington's, ALS, Friedreich's ataxia, and

frontotemporal dementia. These findings validate the potential of blood RNA transcriptomics for effectively classifying neurodegenerative diseases and suggest the possibility of using this knowledge to inform future treatment strategies. The study also highlights the use of extensive gene expression platforms, covering over 30,000 probes, enabling both the exclusion of AD and the precise categorization of the underlying dementia type [39]. Here are the key findings:

Transcript Selection: Initially, the study selected 22 transcripts based on empirical data, focusing on three functional mechanisms associated with AD: inflammation, epigenetics, and stress.[39]

Classification Results by Transcript Subgroups:

- a. Inflammation Transcripts: Seven transcripts linked to inflammation displayed subpar performance in distinguishing between different disease cohorts, with AUC values falling below or equal to 70%.
- b. Epigenetics Transcripts: The study found that the six epigenetics-related transcripts also exhibited limited effectiveness in separating disease cases from controls.
- c. Stress Transcripts: Conversely, nine transcripts associated with stress processes demonstrated satisfactory predictive accuracy ($AUC > 70\%$) when applied to ALS1/2-affected and HD-affected cohorts.
- d. Combined Transcripts: By utilizing all 22 transcripts from the three functional categories, the study improved predictive capabilities for specific diseases. It achieved successful classification for AD1, AD2, ALS1, ALS2, and HD, with AUC values ranging from 72% to 80%. However, the combined transcript set did not perform well when applied to PD1, PD2, FTD, or the FRDA group [39].

In conclusion, this study presented the potential of blood RNA for distinguishing neurodegenerative diseases from healthy states. Inconsistent data preparation across cohorts may account for limited transcript overlap in the RF. Notably, two transcript sets effectively discriminated AD across different cohorts, suggesting broader utility. Such disparities could arise from pre-analytical variations or inherent neurodegenerative disease heterogeneity. The study proposes a noninvasive blood RNA test platform for disease differentiation and swift clinical trial subject recruitment. Consistent data collection

protocols and consideration of temporal disease factors are crucial for enhancing understanding and diagnostic accuracy [39].

Nivedhitha Mahendran et al. proposed an intricate feature selection process comprised of three essential elements [40]:

- i. mRmR (Minimum Redundancy and Maximum Relevance): This technique is instrumental in selecting genes that exhibit high relevance to Alzheimer's disease while minimizing redundancy among them. It isolates the most critical features from the dataset, ensuring that they are highly correlated with the target class, Alzheimer's disease in this context [40].
- ii. Wrapper-based Particle Swarm Optimization (WPSO): To further enhance feature selection, the research team employed PSO, a powerful stochastic metaheuristic algorithm inspired by the swarming behavior of birds. In this context, PSO acts as a wrapper method, aiding in overcoming local optima and contributing to more effective feature selection [40].
- iii. Autoencoders: Autoencoders play a crucial role in compressing gene data while preserving essential information. This process reduces the sensitivity of the data to variations during training, which can be particularly advantageous in complex datasets. The autoencoder's ability to minimize information loss is invaluable in feature selection [40].

Following this comprehensive feature selection process, a subset of genes that demonstrated the greatest relevance to Alzheimer's disease was identified. This subset included genes such as CTD-3092A11.2, CHGB, JPX, MAFF, AC004951.6, APLNR, MT1M, SST, PCYOX1L, PRO 1804, and SLC39A12. These genes held the potential to serve as significant biomarkers for Alzheimer's disease, offering a more precise means of diagnosis [40].

With the relevant genes selected, the study progressed to the implementation of an Improved Deep Belief Network (IDBN). This machine learning-based classifier was chosen for its ability to provide more accurate and reliable AD classification. To ensure optimal performance, Bayesian Optimization was employed to fine-tune the IDBN's hyperparameters. The results obtained from this comprehensive approach were highly promising. The proposed feature selection pipeline, combining mRmR, WPSO, and

Autoencoders, when coupled with the IDBN classifier, demonstrated an impressive accuracy of 96.78% in classifying Alzheimer's patients. This performance outshone other models and techniques employed in the study, offering a substantial improvement in Alzheimer's disease diagnosis accuracy [40].

AD is linked to factors such as A β deposition and tau protein hyperphosphorylation, though the exact underlying causes remain elusive. Various hypotheses, including the A β cascade, tau, inflammation, cholinergic, oxidative stress, and glucose hypometabolism theories, have been proposed to explain AD. Longxiu Yang et al. proposed three steps gene analysis technique applied to a dataset consisting of 28,844 genes, focusing on AD classification [41]. Detailed as follow:

- a. Feature Selection and Classifier: Initially, 500 feature genes were selected using mRmR analysis. Among the classifiers constructed, the K-nearest neighbor (KNN) classifier with 14 feature genes achieved the highest Matthews correlation coefficient (MCC). This KNN classifier demonstrated high diagnostic efficacy with a sensitivity of 0.907, specificity of 0.929, accuracy of 0.918, MCC of 0.836, and an area under the curve (AUC) value of 0.935 for classifying AD patients [41].
- b. PCA and Heatmap Analysis: Principal Component Analysis (PCA) was employed to classify patients into normal and AD groups based on the expression of the 14 feature genes. The results clearly separated AD patients from normal subjects. Additionally, a heatmap analysis revealed that these 14 feature genes effectively distinguished between AD patients and healthy individuals [41].
- c. RWR and Enrichment Analysis: Differentially expressed genes (DEGs) from the dataset were compared with the feature genes from the KNN classifier to identify three core genes: HSPB3, AEBP1, and RNU1G2. A protein-protein interaction (PPI) network was constructed using these core genes, and 52 genes closely associated with them were identified. Enrichment analysis indicated that these genes were involved in various biological processes and signaling pathways, including the MAPK signaling pathway, B cell receptor signaling pathway, and T cell receptor signaling pathway. Lastly, expression analysis of the three core genes in AD patients showed that HSPB3 was down-regulated, while AEBP1 and RNU1G2 were up-regulated, suggesting their

involvement in AD pathogenesis, particularly in immunity and cell transportation processes [41].

In summary, the study highlights the potential of machine learning techniques for gene analysis in AD diagnosis. The KNN classifier using 14 feature genes demonstrated strong diagnostic performance, and further analysis revealed key genes associated with AD pathogenesis and immune-related processes. These findings contribute to our understanding of the molecular mechanisms underlying AD. The study revealed three core genes (HSPB3, AEBP1, RNU1G2) associated with AD pathogenesis, with potential roles in protein folding, inflammation, and RNA processing which suggests that these genes may contribute to AD development and could be targets for future research and clinical therapy. The study also emphasized the need for further experimental verification to better understand AD pathogenesis and improve treatment options for AD patients [41].

A study done by Jack Cheng et al. utilized gene expression profiles from 310 AD patients and 157 non-demented controls sourced from the GSE33000 dataset at NCBI GEO. These data were preprocessed, accounting for factors like age and gender. The dataset included 39,279 probes, 13,798 annotated genes, and 9,969 profiled genes, excluding 31 probes mapping to multiple genes [42]. For model verification, another dataset (GSE844229) with 56 postmortem samples reflecting various AD pathology levels was employed. Samples were categorized into control or AD groups based on clinical criteria. Notably, 3,680 genes absent in the testing dataset were artificially introduced with a fixed fold change (FC) set to "1" for all samples during testing. FC was defined as the average of gene expression of AD samples relative to that of control samples. Student's T-test was used to calculate the significance of FC. Non-significant FCs ($p > 0.05$) were neglected. The study developed a workflow utilizing machine learning (ML) to identify genes associated with Alzheimer's disease (AD). Seven different ML models were applied, each producing a list of genes along with their respective predictive weights. These models exhibited average accuracies ranging from 86.30% to 91.22%. Combining the genes identified by the seven models, a list of 1,126 non-redundant genes with weights greater than 0 was obtained. To further refine the selection, genes meeting specific criteria—those with the lowest weight value (0.001) and not present in the global enrichment analysis—were filtered out. This process resulted in a final list of 314 representative genes [42]. The study conducted an analysis of

variance (ANOVA) test to assess the differences in performance among the various ML models. The results did not indicate a significant difference, suggesting that the models perform similarly in this context [42].

To validate the findings on a different dataset (GSE84422), the study applied the trained models. Although nearly one-third of the training genes were missing in the test dataset, GSE84422, the accuracy varied across models (28.57%, 58.93%, 76.79%, 82.14%, 71.43%, 44.64%, and 69.64% for decision tree, random forest, gradient, boosted tree, generalized linear model, linear regression, decision stump, and rule induction, respectively). Models with a larger number of attributes, such as the Generalized Linear Model (GLM), exhibited better performance compared to others. These results highlight both the generalizability and challenges of applying ML models to cross-platform datasets [42].

ML and FC based methods for gene identification were compared. Two strategies were used, one based on FC extremes and the other on FC cutoffs and p-value ranks. Surprisingly, there was minimal overlap (21.3% or 25.5%) between ML-derived genes and conventional methods, indicating substantial differences. ML emphasized genes related to oxidative phosphorylation, including ATP5C1, ATP5G1, NDUFA1, NDUFA4, NDUFA6, NDUFA12, NDUFB1, NDUFB2, NDUFB9, NDUFV1, NDUFV2, and UQCRCF1, which play roles in the electron transport chain (ETC) complexes. This pathway is part of the KEGG Alzheimer's pathway. ML also uncovered co-predictive genes associated with ETC complex genes, revealing their relevance to AD. Genes like ATP5G1, ATN1, CTXN1, and NDUFV were identified as predictors of AD, with functions related to neuronal transcription, cortical neuron signaling, and synaptic transmission. ML-generated decision trees highlighted other potential AD-related genes [42].

Makrina Karaglani et al. used the JADBIO AutoML platform, offering classification algorithms like Random Forests, SVM, Ridge Logistic Regression, and Decision Trees [43]. It incorporates the BBC-CV protocol for hyper-parameter tuning, performance estimation, and handling multiple attempts, also explored multiple types of datasets related to Alzheimer's disease (AD) and its associations with age and sex, identified and used a total of seven different types of datasets [43]:

- a. Two metabolomic datasets (Metabolomic 1: ST000046 and Metabolomic 2: ST000433)
- b. Two proteomic datasets (GSE39087 and GSE29676)
- c. Two microRNA (miRNA) transcriptomic datasets (Transcriptomic 1: GSE46579 and Transcriptomic 2: GSE120584)
- d. Two mRNA transcriptomic datasets (Transcriptomic 3: GSE63060 and Transcriptomic 4: GSE63063)

Three datasets were excluded from further analysis due to low sample numbers, unavailability of processed data, or inclusion of only female samples. The study focused on comparing AD patient samples with age and sex-matched cognitively healthy individual samples detailed in Table 2.6 [43].

Table 2.6: Datasets and Selected Samples Used in the Study [43].

Dataset	Data Type	Alzheimer's Disease Samples	Cognitively Healthy Samples	Features
Metabolomic 1	Metabolites profiles	15	15	3734
Metabolomic 2	Sphingolipid and fatty acid profiles	18	21	25
Proteomic	Protein profiles	25	37	9483
Transcriptomic 1	miRNA profiles	48	22	506
Transcriptomic 2	miRNA profiles	300	289	2566
Transcriptomic 3	mRNA profiles	134	100	38327
Transcriptomic 4	mRNA profiles	126	131	32053

AutoML analysis of omics datasets in an AD study yielded the following predictive biosignatures:

- a. **Proteomic Data:** Analyzing proteomic data from 25 AD patients and 37 healthy individuals produced a protein-based biosignature with seven predictors, achieving an AUC of 0.921.
- b. **miRNA-Transcriptomic Data:** Analysis of miRNA-transcriptomic data from 48 AD patients and 22 healthy individuals generated a miRNA-based predictive biosignature with three predictors, achieving an AUC of 0.975.
- c. **mRNA-Transcriptomic Data:** A large mRNA-transcriptomic dataset from 134 AD patients and 100 healthy individuals led to the creation of six statistically equivalent predictive biosignatures, each with 25 mRNA predictors, achieving a high AUC of 0.846.

- d. **Train-Test Split:** The dataset was split into a 70/30 train-test ratio, with the train dataset including miRNA-transcriptomic data from 232 AD patients and 200 healthy individuals. A predictive biosignature was developed using Ridge Logistic Regression with 25 mRNA predictors, achieving an AUC of 0.812. It was validated on the test dataset with an AUC of 0.807.
- e. **Additional mRNA-Transcriptomic Data:** Another analysis of mRNA-transcriptomic data from 126 AD patients and 131 healthy individuals generated two predictive biosignatures with 25 mRNA predictors, achieving an AUC of 0.726.
- f. **Random Forests Model:** The dataset was split into 70/30 train-test sets, with the train set including mRNA-transcriptomic data from 100 AD patients and 90 healthy individuals. Four statistically equivalent predictive biosignatures were produced using a Classification Random Forests model, each with 25 mRNA predictors and an AUC of 0.728. The test data confirmed model stability with an AUC of 0.814.
- g. **Metabolomic Data:** Analysis of metabolomic data from AD patients and healthy individuals resulted in low-performing biosignatures using Support Vector Machines models.

These biosignatures provide valuable insights into AD prediction and diagnostics using diverse biological data types. In conclusion, using the autoML JADBIO tool, predictive biosignatures were generated from limited multi-omics data, offering potential for minimally invasive blood-based Alzheimer's disease diagnostics, pending clinical validation. This underscores the importance of autoML in biomarker discovery. Among the biosignatures, the miRNA-transcriptomic biosignature stands out with the highest performance (AUC 0.975) and simplicity (three predictors).

Furthermore, this study highlights the significance of reevaluating "old" datasets, challenging the notion that all discoveries have been made. New computational methods like JADBIO and AutoML enable fresh insights, emphasizing the value of reanalyzing existing data in light of evolving scientific knowledge [43].

Shehan Perera's et al. study identifies 14 genes, some validated biologically, and evaluates a machine learning framework using the GSE5281 gene dataset to discover potential Alzheimer's disease risk biomarkers. The research analyzes PCA's impact, utilizes feature selection methods like Random Forest and Extra Tree Classifier to identify top biomarker

genes, and validates them against known risk genes, offering novel insights into Alzheimer's disease [44].

The study's methodology comprises several key steps. Initially, data pre-processing is conducted using the GSE5281 brain dataset. Probe set IDs are mapped to official gene symbols according to the GPL570 annotation table. This dataset encompasses 161 samples, with 87 diagnosed with Alzheimer's disease and 74 from the healthy control group, involving a total of 24,438 unique gene symbols [44]. Next, the study filters out differentially expressed genes, which exhibit significantly distinct expression values between Alzheimer's disease-affected samples and control samples. This is achieved through the t-test, where the p-value for each gene is calculated. If the p-value is less than 0.05, the null hypothesis (equal means of the two sample classes) is rejected in favor of the alternative hypothesis, indicating a difference between Alzheimer's disease and non-Alzheimer's disease samples. Features with p-values less than 0.05 are selected, and their fold change is calculated [44].

Feature selection is then performed using multiple methods. Firstly, Principal Component Analysis (PCA) effectively reduces the number of gene features, resulting in 50 unique genes selected as the highest scored feature in each component. Random Forest and Extra Tree Classifier methods are also employed to rank features by their importance and select the top 50 highest-scoring features in each case. In the classification stage, performance evaluation is carried out on a dataset containing 2,000 genes. To ensure unbiased results, two-thirds of the dataset are used for training, and the remaining one-third for testing. Six classification methods are applied, including Naive Bayes, Decision Trees, Random Forest, Nearest Neighbor, SVM with linear and Gaussian kernels. In the classification phase, five methods were assessed. SVM with a linear kernel achieved the highest accuracy, 93.8% in training and 89.8% in testing. C4.5 and Naive Bayes had lower accuracy. SVM with a linear kernel was chosen for gene selection evaluation. Feature selection using PCA, Random Forest, and Extra Tree Classifier aimed to identify Alzheimer's biomarkers. Experimental results, illustrated in Figure 2.2, demonstrated the significance of selected features in Alzheimer's disease. PCA stood out by identifying a higher number of differentiating genes. These findings underscore the potential importance of the identified

genes in Alzheimer's disease research. Table 2.7 shows the cross validated classification accuracy for all five methods on the dataset [44].

Table 2.7: Testing and Training Accuracy for 2000 Genes [44].

Model	Testing	Training
Naive Bayes	81.63%	83.93%
C4.7 Decision Tree	87.76%	78.57%
Nearst Neighbor	85.71%	86.58%
Random Forest	83.67%	87.48%
SVM + Gaussian Kernel	83.67%	91.06%
SVM + Linear Kernel	89.80%	93.76%

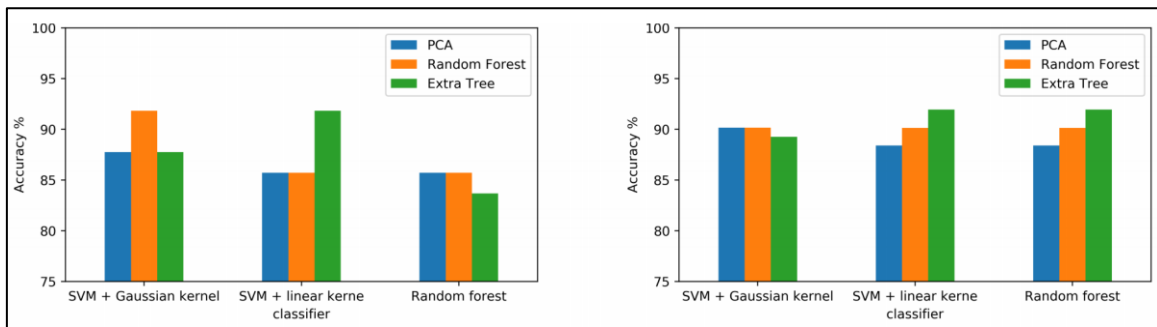


Figure 2.2: Testing (left) and Training (right) Accuracies of Three Classifiers Based on Features Extracted from Three Feature Selection Methods [44].

For small biomarkers sets, correlation values of each gene that are only identified by one feature selection method were calculated, 34 genes selected by PCA, 18 genes by extra tree classifier and 18 genes from random forest, also correlated feature percentages with different correlation values. When correlation value is 0.86, it has 40% correlated features in extra tree and 60% correlated features in random forest whereas the correlation value for PCA is just 0%. This proves that PCA is better in finding new biomarker genes that provide new information. Therefore, the genes identified by PCA may provide a novel insight into molecular processes related to AD, then the other two approaches [44]. The resulting 50 genes of each fold are sorted by their importance based on applying PCA, random forest and the extra tree classifier. Then we select the best 50, 40, 30, 20, 15, 10, 5, 3, 2 and 1 genes for each algorithm and utilize these to compute the classification performance using an SVM with gaussian kernel, linear kernel as well as random forest. As shown in Figure 2.3, the highest accuracy is given when the number of features is in the range of 10-15. This means that the models work better with small sets of genes than with a large number [44].

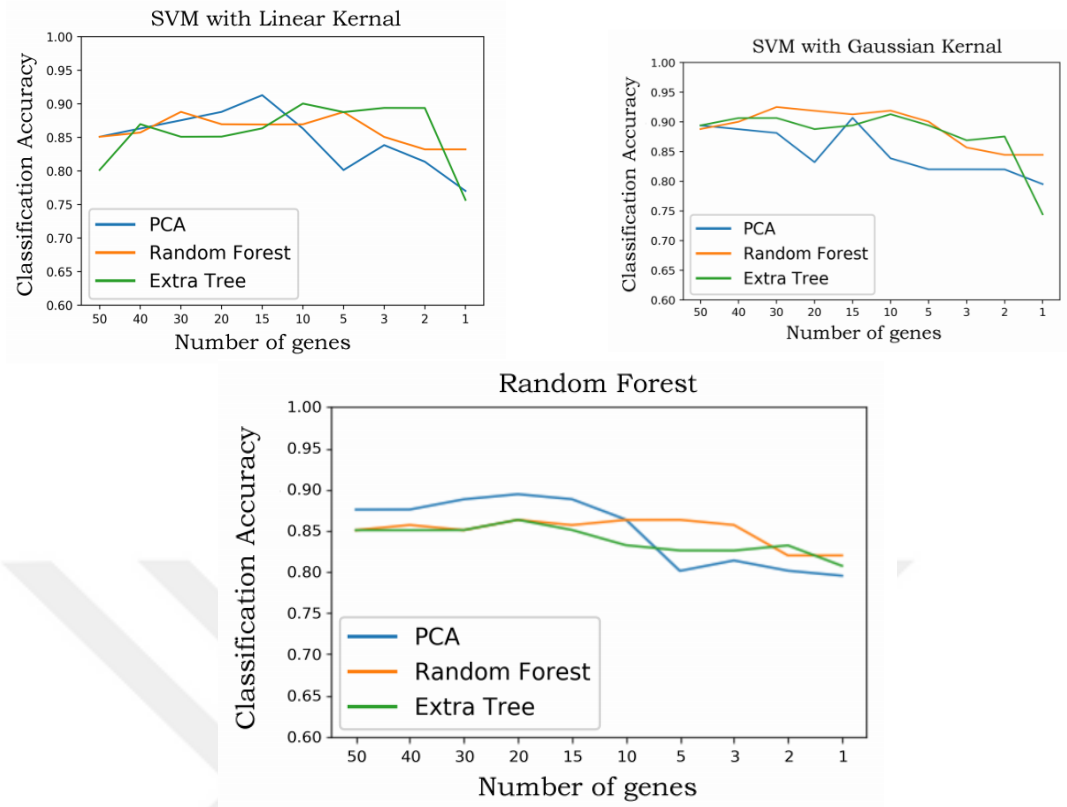


Figure 2.3: Classification Accuracy of Selected Genes Identified by Three Feature Selection Methods and Trained by Three Classifiers [44].

The study employed three feature selection methods, ultimately identifying eight common genes shared among them. These genes formed an overlapped set that demonstrated higher accuracy compared to individual gene selection methods (Figure 2.3). However, the study also found that the classifiers achieved the best accuracy when selecting a smaller number of features, typically ranging from 10 to 15 (Figure 2.4). To determine the optimal number of genes, the study considered correlation coefficients and selected an additional nine genes. Interestingly, three genes, SLC39A12, CTD-3092A11.2, and RP11-271C24.3, were found in both the overlapped set and the newly identified nine genes, prompting their removal from the overlapped set [44].

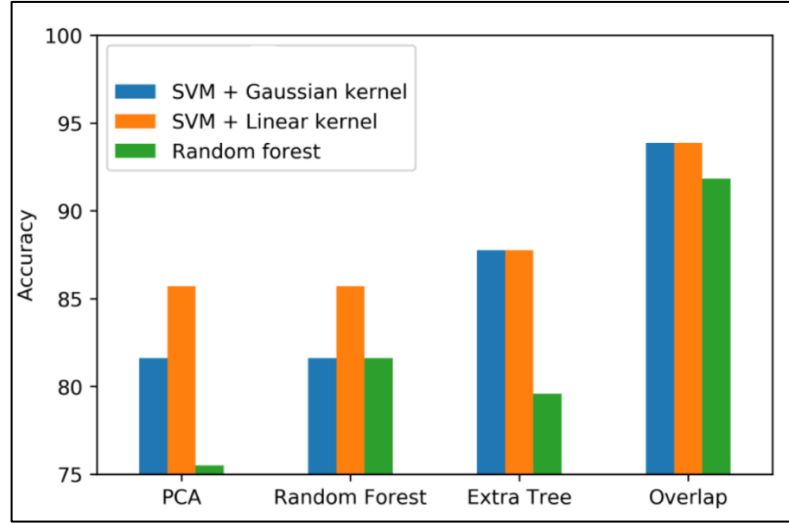


Figure 2.4: Accuracy of the Model with Genes only Found by Each Feature Selection Method When Trained Upon Three Classifiers; SVM with Gaussian and Linear Kernel and Random Forest [44].

The combination of the overlapped genes and the new nine genes resulted in a remarkable accuracy of 93.9% with SVM and 91.4% with the Random Forest classifier (Table 2.8). Moreover, an analysis of the correlation matrix of these selected genes revealed lower correlations compared to the original overlapped set. This strategy aimed to minimize gene redundancy, which is valuable for diagnostic tool design and understanding the results of other gene selection approaches [44].

Table 2.8: Performance Evaluation of the 9 Genes found only by PCA [44].

Model	Accuracy	Sensitivity	Specificity
SVM + Gaussian kernel	93.9%	98.6%	96.6%
SVM + linear kernel	93.9%	98.6%	97.7%
Random Forest	91.84%	95.9%	96.6%

The combined gene set yielded the highest accuracy achieved in the study, with SVM using a Gaussian kernel providing the highest testing accuracy (Table III). In total, this method selected 14 potential biomarker genes for Alzheimer's disease, including AC004951.6, MAFF, SLC39A12, PCYOX1L, CTD-3092A11.2, RP11-271C24.3, PRO1804, PRR34-AS1, SST, CHGB, MT1M, JPX, APLNR, and PPEF1. These genes hold promise for advancing our understanding of Alzheimer's disease and potentially serving as valuable biomarkers [44].

Table 2.9: Performance Evaluation of the Newly Found Genes by PCA and the Overlapped set [44].

Model	Accuracy	Sensitivity	Specificity
SVM + Gaussian kernel	93.9%	97.3%	96.6%
SVM + linear kernel	93.9%	98.6%	97.7%
Random Forest	91.84%	95.8%	94.4%

In summary, this analysis highlights 14 genes, including SST, CHGB, SLC39A12, MT1M, MAFF, and PCYOX1L, as potential biomarkers or contributors to Alzheimer's disease. These genes offer avenues for further research to deepen our understanding of the molecular mechanisms underlying AD and potentially lead to novel therapeutic targets [44].

Patel Hamel et al. conducted a study employing two distinct strategies to develop XGBoost classification models for the diagnosis of Alzheimer's disease (AD) through blood transcriptomic profiling [45]. The initial approach concentrated solely on training the models using samples from AD patients and cognitively healthy individuals. The study adhered to specific criteria for the selection of microarray gene expression datasets. These criteria encompassed the profiling of disorders related to AD, common elderly health disorders, or mental health issues, utilizing RNA extracted from whole blood or its components. The datasets needed to comprise a minimum of ten human subjects and be generated on the Illumina or Affymetrix microarray platform, with a minimum of 20,000 probes present on an expression of 11 different BeadArrays of 1,681 common reliably detected genes in any group (based on gender and diagnosis) [45].

Two distinct types of classification models were formulated based on the aforementioned methodologies. The initial model, termed the "AD vs Healthy Control" classification model, was developed using the only two AD datasets (GSE63060 and E-GEOD-6613) utilizing 160 samples from AD patients along with 127 associated samples from healthy controls sourced from available datasets. On the other hand, the second model, referred to as the "AD vs Mixed Control" classification model, was trained using the same AD and healthy control samples as the first model but supplemented with samples from diverse related disorders and their corresponding healthy controls. This mixed control group aimed to provide a more diverse and representative training set for the model, as shown in Figure 2.5 in the training data section [45].

The performance of these models was subsequently evaluated within an independent testing set that encompassed AD patients, diverse disorders, and healthy subjects, totaling 814 individuals. To account for inherent randomness in the process, a thousand iterations of each classification model were developed, refined, and evaluated. The outcomes of the study affirmed that the "AD vs Healthy Control" models exhibited limited efficacy when applied to a diverse and heterogeneous aging population. Conversely, the "AD vs Mixed Control" models demonstrated enhanced performance, implying their potential in effectively distinguishing AD from other related disorders and healthy subjects [45].

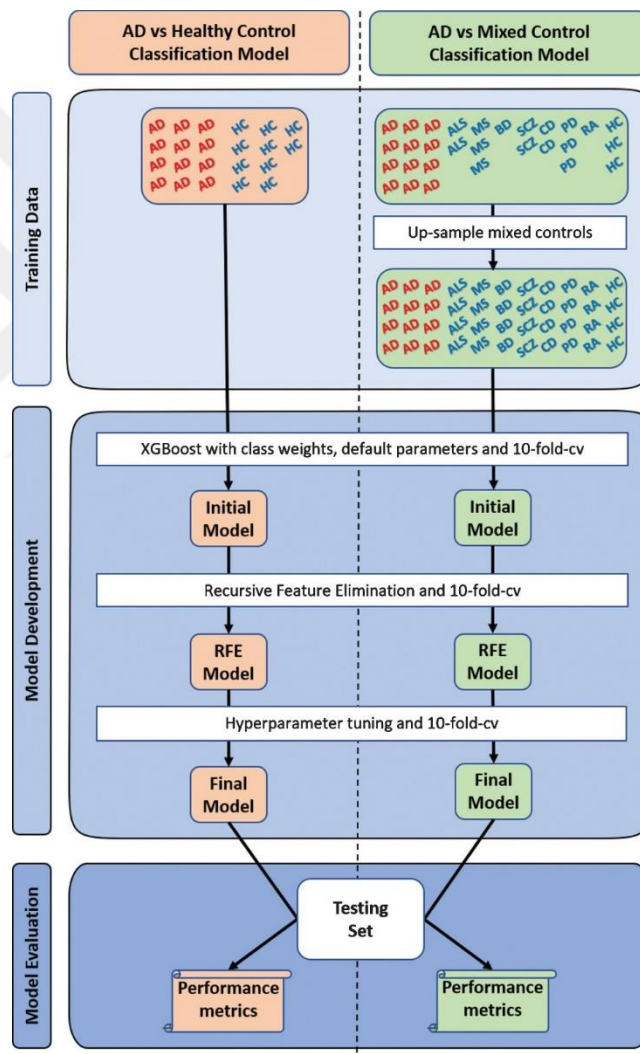


Figure 2.5: Proposed Gene Selection Model [45].

Among the thousand "AD vs Mixed Control" classification models, approximately 89 predictive features (genes) were identified on average, allowing discrimination between

AD and non-AD subjects with an average balanced accuracy of 68%. However, out of the 1,681 genes available, only 800 were selected as predictive features by at least one of the thousand models, and interestingly, 11 genes were consistently chosen by all one thousand models. These specific genes include KDM3B, TH1L, RARA, SPEN, NDUFA1, THYN1, UBR4, BSDC1, LDHB, LPP, and BAG5 [45].

In terms of diagnostic accuracy, the "AD vs Mixed Control" models showcased encouraging results, boasting an average area under the curve (AUC) score of 0.86, which signifies a "very good" diagnostic performance. Nevertheless, the models' clinical utility was found to be more adept at excluding the presence of AD rather than accurately detecting its occurrence (AUC scores were generated using the R package "ROCR" (version 1.07) with the following recommended diagnostic interpretations used: "excellent" (AUC=0.9–1.0), "very good" (AUC=0.8–0.9), "good" (AUC=0.7–0.8), "sufficient" (AUC=0.6–0.7), "bad" (AUC=0.5–0.6), and "test not useful" when AUC value is <0.5) [45].

In summary, the study shed crucial light on the constraints of the conventional approach that exclusively employs AD and healthy control samples for training classification models aimed at identifying blood-based gene expression biomarkers for AD diagnosis. The integration of samples from related disorders and prevalent elderly health conditions led to marked enhancements in model performance, thereby emphasizing the intricacies involved in diagnosing AD within a heterogeneous aging population [45].

2.3 MANTARAY OPTIMIZATION ALGORITHM

Mihailo Micev's et al. study introduces a novel hybrid metaheuristic algorithm that combines two distinct types of existing algorithms, Simulated Annealing (SA) and MRFO. Metaheuristic algorithms fall into two categories: single solution-based and population-based algorithms. Single solution-based algorithms, with SA as a prominent example, iteratively apply generation and replacement procedures on a single solution. In contrast, MRFO represents population-based algorithms that commence with an initial population and iteratively generate new populations, replacing the current population [46].

The hybrid approach in this study merges the SA algorithm with evolutionary algorithms (EAs). This merging can be collaborative or integrative. Collaborative hybrid SA–EA can

take the form of teamwork collaboration, where both algorithms work in parallel, or relay collaboration, where one follows the other. In the EA–SA version, the EA is executed first, and then each individual in the final population is optimized by the SA algorithm. In the relay-collaborative hybrid algorithm SA–EA, the SA algorithm initializes the EA algorithm's population. In integrative hybrid metaheuristics, a specific function of one algorithm is replaced by another algorithm [46].

This study explores the hybridization of the non-evolutionary MRFO algorithm with the SA algorithm. While MRFO doesn't fall into the category of evolutionary algorithms, it operates on a population of solutions, making it a suitable candidate for merging with SA. The chosen strategy is a teamwork relay-collaboration approach, similar to SA–EA. Specifically, the hybrid algorithm uses SA to initialize the population for MRFO, resulting in a high-quality initial solution. This approach demonstrates a significant improvement in convergence speed when compared to other algorithms for solving the same optimization problem [46].

Before the iteration process begins, the manta ray population needs to be initialized. In the original MRFO algorithm, the population is randomly initialized within the search space. The proposed hybrid SA–MRFO algorithm, akin to SA–EA, employs SA for population initialization. This initialization with SA aims to enhance solution accuracy in the initial iterations, thereby greatly accelerating convergence. SA is a single-solution-based algorithm, which means it must be applied to each individual in the initial population of manta rays. In the MRFO algorithm, the population, denoted as P , consists of m manta rays, represented as $P = \{p_1, p_2, \dots, p_m\}$, where p_i ($i = 1, 2, \dots, m$) signifies the position of each individual [46].

Following the population initialization, the foraging strategies, as mentioned earlier, are applied to the entire population. Initially, the chain foraging strategy is executed for each individual based on Eq. (2.1).

$$p_i^d(t+1) = \begin{cases} p_i^d(t) + r(p_{\text{best}}^d - p_i^d(t)) + a(p_{\text{best}}^d - p_i^d(t)), & i = 1 \\ p_i^d(t) + r(p_{i-1}^d - p_i^d(t)) + a(p_{\text{best}}^d - p_i^d(t)), & \forall i \in m, i \neq 1 \end{cases} \quad (2.1)$$

So that,

$$a = 2r\sqrt{|\log(r)|} \quad (2.2)$$

Here, $p_i^d(t)$ represents the position of the i th individual at iteration t in the d th dimension. r is a random number within the range of 0 to 1, a is the weight coefficient, and p_{best} is the best individual discovered thus far [46].

The second foraging strategy, referred to as cyclone foraging, is expressed by the following mathematical equation:

$$p_i^d(t+1) = \begin{cases} p_{\text{best}}^d + r_1(p_{\text{best}}^d - p_i^d(t)) + \beta(p_{\text{best}}^d - p_i^d(t)), & i = 1 \\ p_{\text{best}}^d + r_1(p_{i-1}^d - p_i^d(t)) + \beta(p_{\text{best}}^d - p_i^d(t)), & \forall i \in m, i \neq 1 \end{cases} \quad (2.3)$$

So that,

$$\beta = 2e^{r_1 \left(\frac{t_{\text{max}} - t + 1}{t_{\text{max}}} \right)} \sin(2\pi r_1) \quad (2.4)$$

In this equation, r_1 is a randomly generated number between 0 and 1, t_{max} represents the maximum number of iterations, and β is the weight coefficient. Equation (2.3) reflects the MRFO algorithm's capacity for exploitation, which entails searching the vicinity around the current best solution [46].

In a study by Ehab E. Elattar et al., it is emphasized that MRFO is significantly influenced by the optimal position of plankton food. To address this, the proposed EMRFOGS introduces an adaptive adjustment to the fitness function. This adaptation is based on the dynamic changes in the fitness landscape during each iteration, achieved through the use of a dynamic weighting factor [47]. Consequently, the objective function can be expressed as follows:

$$O = \sum_{i=1}^M \omega_i \frac{F_i}{F_{i\text{max}}} \quad (2.5)$$

$$\omega_1 = \begin{cases} 1 & \text{if } I_t < M_{I_t}/3 \\ 1 - \frac{I_t}{M_{I_t}} & \text{else} \end{cases} \quad (2.6)$$

$$\omega_i = \frac{1 - \omega_1}{M}, i = 2, 3, \dots, M \quad (2.7)$$

Here, ω_i represents the weight factor associated with each specific objective (i) [47]. According to equations (2.2-2.3), the fitness function under consideration dynamically changes within a single simulation operation, leading to increased diversity among individuals with a focus on an adaptive normalized objective. The normalization process, coupled with weight variation, ensures that various objectives are treated equally. To generate Pareto-optimal individuals, the proposed EMRFOGS incorporates an external repository for storing non-dominated solutions, using Pareto dominance to update this repository. In each iteration, the updated individuals are compared with the existing Pareto solutions in the repository to eliminate dominated solutions. As a result, the repository is continuously updated, and if the capacity is exceeded, a removal strategy is applied by selectively removing Pareto candidates in densely populated areas using a roulette wheel selection method [47].

In addressing the Optimal Power Flow (OPF) problem, the balance equations for both AC and MDC power systems are inherently ensured through a sequential AC/DC power flow method. To maintain operational limits for decision variables, they begin within their specified bounds, and if any boundary is breached during iterations, it is randomly regenerated within the appropriate range [47].

In the context of operational limits for dependent variables in AC/MDC power transmission systems, they are incorporated into the objective functions with quadratic penalty terms. This approach ensures that solutions violating the constraints of dependent variables are not selected in the subsequent iteration. Consequently, each new position is compared to its equivalent in the previous iteration in terms of achieved objectives. If the new position does not dominate the prior one, it replaces it. This process maintains diversity and enhances solution quality, resulting in the generation and storage of a set of Pareto optimal solutions [47].

To determine the optimal operating point that strikes a balance, the study employs the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). TOPSIS is a method used to select the most favorable solution (A) from a set of alternatives that consider multiple criteria (X) [47].

Jiaying Feng's et al. study aimed to enhance MRFO by implementing two improvements. The first involved modifying the algorithm using the well-known opposition-based learning (OBL) technique [48]. This adjustment addressed situations where a deviation between the primary solution and the best solution occurred. Such deviations could lead to either further optimization or potential problems when solving certain issues. To account for these scenarios, the OBL technique was applied to enhance individual algorithm performance as follows:

$$\bar{x}_i^d = x_i^{max} + x_i^{min} - x_i^d \quad (2.8)$$

In this context, $\bar{x}_i^d$ represents the opposite location of x_i^d , while x_i^{max} and x_i^{min} indicate the upper and lower constraints, respectively. $\bar{x}_i^d$ is used in place of x_i^d when $\bar{x}_i^d$ performs better [48].

The second modification involves implementing a self-adaptive approach, which adjusts the size of the individual dynamically. Defining the individual size is crucial for optimization. Initially, the primary individual size is transformed as follows:

$$x_i^d(t) = 10 \times N \quad (2.9)$$

Here, N represents the number of parameters. Subsequently, the revised individual size for subsequent iterations is calculated using the following equation:

$$\bar{x}_i^{d+} = \text{round} (x_i^d \times (1 + \delta)) \quad (2.10)$$

Here, δ represents a randomly selected value from a uniform distribution within the range of $[-0.5, 0.5]$. This variable allows for the adjustment of the individual's size [48].

The individual's size increases or decreases based on whether δ is positive or negative. If the size of the updated individual is greater than the previous one ($\bar{x}_i^{d+} > \bar{x}_{i-1}^{d+}$), the current populations transition to the subsequent iteration, and through elitism, the remaining individual is retained. Conversely, when $\bar{x}_i^{d+} < \bar{x}_{i-1}^{d+}$, only the best individuals from the current population proceed to the later iteration, and the remaining populations

are eliminated. If the updated individual size is less than the problem size, it is set equal to the number of problem parameters [48].

In line with previous research, Mohamed Abd Elaziz's et al. study focuses on enhancing MRFO using fractional-order (FO) calculus during the exploitation phase. The Grunwald–Letnikov fractional differ-integral operator, leveraging heredity and non-locality properties, simulates the lasting impact of previous manta ray locations on their future movement directions. The study investigates FO-MRFO's sensitivity to memory length and derivative order through CEC2017, assessing its quality and stability with 30- and 50-dimensional optimization problems from CEC2017. The applicability of the proposed algorithm is demonstrated in multilevel threshold image segmentation optimization [49].

Fractional calculus, an older mathematical branch, has gained renewed attention due to its flexible modeling capabilities. Researchers are increasingly interested because non-integer orders in differential equations offer extra degrees of freedom for describing various physical models. The non-local nature of fractional differ-integral operators makes them suitable for representing systems with memory dependency using fewer model parameters. In contrast to integer-order calculus, where position updates depend solely on the previous iteration, fractional calculus introduces weighted memory dependency, giving more significance to the latest positions over older ones in the position update formula for agents or animals [49].

FO-MRFO aims to boost MRFO performance and apply it to global optimization and image segmentation. Fractional-order is harnessed to enhance MRFO's exploitation capabilities, addressing its tendency for better exploration than exploitation. The fractional-order element supports MRFO by enabling memory sharing among agents regarding their individual positions, ultimately enhancing convergence speed and output quality. Additionally, to further enhance FO-MRFO, modifications were made to the somersault factor (S), now based on the best fitness function achieved and the attained fitness function of each agent [49].

The following equation is utilized by FO-MRFO to update Z_i .

$$\begin{aligned}
Z_i^{t+1} = & \frac{1}{1!} \gamma Z_i^t + \frac{1}{2!} \gamma(1-\gamma) Z_i^{t-1} + \frac{1}{3!} \gamma(1-\gamma)(2-\gamma) Z_i^{t-2} + \frac{1}{4!} \gamma(1-\gamma)(2-\gamma)(3-\gamma) Z_i^{t-3} + \\
& \frac{1}{5!} \gamma(1-\gamma)(2-\gamma)(3-\gamma)(4-\gamma) Z_i^{t-4} + \frac{1}{6!} \gamma(1-\gamma)(2-\gamma)(3-\gamma)(4-\gamma)(5-\gamma) Z_i^{t-5} + \\
& \frac{1}{7!} \gamma(1-\gamma)(2-\gamma)(3-\gamma)(4-\gamma)(5-\gamma)(6-\gamma) Z_i^{t-6} + \\
& \frac{1}{8!} \gamma(1-\gamma)(2-\gamma)(3-\gamma)(4-\gamma)(5-\gamma)(6-\gamma)(7-\gamma) Z_i^{t-7} + \mathbf{S} \times (r_2 Z_{best} - r_3 Z_i^t), i=1,2,...,N
\end{aligned} \tag{2.11}$$

To enhance the equilibrium between incorporating previous terms from memory and factoring in the best agent position achieved, adjustments were made to the somersault factor (S).

$$S = 2 \times \text{atan}(\text{abs}(\text{best}F(i) - \text{global}F)) \tag{2.12}$$

Here, $\text{best}F$ represents the best function of each agent, $\text{global}F$ is the overall best function achieved so far, and atan signifies the inverse tangent function [49].

Mohamed H. Hassan's et al. study discusses MRFO, introduced by Zhao et al. in 2020, which emulates Manta ray behaviors through chaining, cyclone, and somersault strategies. While MRFO resembles various meta-heuristic algorithms, it notably starts with a random initialization step as defined by Eq (2.13) [50].

$$X_{ij}(:) = Lb_{ij} + r(:) \cdot (Hb_{ij} - Lb_{ij}) \forall i \in N_{\text{pop}}, j \in N_{\text{var}} \tag{2.13}$$

Mimicking Manta Ray behavior, the mathematical formula in Eq (2.14) adjusts the positions (Xi, j) within the chaining strategies to optimize food location [50]:

$$\begin{aligned}
X_{i,j}(t+1) = & \begin{cases} X_{i,j}(t) + (X_{best,j}(t) - X_{i,j}(t)) \cdot (r(:) - \lambda) \Delta \forall i=1, j \in N_{\text{var}} \\ X_{i,j}(t) + (X_{i-1,j}(t) - X_{i,j}(t)) \cdot (r(:) - \lambda) \Delta \forall i > 1 : N_{\text{pop}} \end{cases} \\
\varphi = & 2 \cdot r(:) \cdot \sqrt{|\log(r(:))|}
\end{aligned} \tag{2.14}$$

The swarm exhibits a spiral movement akin to cyclones upon detecting food. In the final step of this operator scheme, the latest Manta ray positions are fine-tuned around the best position using the formula presented in is determined as outlined in (2.15),

$$X_{i,j}(t+1) = X_{i,j}(t) + SF \cdot (r_2 X_{best,j} - r_3 X_{i,j}(t)) \Delta \forall i \in N_{\text{pop}} \tag{2.15}$$

$X_{best,j}$ representing the optimal position with abundant food concentration. In the event of low concentration, a random position is set as illustrated in (2.19) [18]:

$$X_{i,j}(t+1) = \begin{cases} X_{best,j} + \Delta X_{i,j} \cdot (r - \beta) \Delta \forall i=1, j \in N_{va} \\ X_{best,j} + r \cdot (x_{i-1,j}(t) - X_{i,j}(t)) + \beta \cdot \Delta X_{i,j} \Delta \forall i>1: N_{pop} \end{cases} \quad (2.16)$$

$$where \Delta X_{i,j} = X_{best,j}(t) - X_{i,j}(t)$$

$$\beta = 2 \cdot \exp\left(r_1 \cdot \frac{T_{max} - t + 1}{T_{max}}\right) \cdot \sin(2\pi r_1) \quad (2.17)$$

$$X_{i,j}(t+1) = \begin{cases} X_{rand} + \Delta X_{i,j} (r - \beta) \Delta \forall i=1, j \in N_{var} \\ X_{best,j} + r \cdot (X_{i-1,j}(t) - X_{i,j}(t)) + \beta \cdot \Delta X_{i,j} \Delta \forall i>1: N_{pop} \end{cases} \quad (2.18)$$

$$where \Delta X_{i,j} = X_{rand} - X_{i,j}(t)$$

$$X_{rand}(:) = Lb + r(:) \cdot (Hb - Lb) \quad (2.19)$$

Metaheuristic algorithms exhibit exploration and exploitation behaviors, granting them an advantage over other approaches. In MRFO, the exploitation cycle is executed if $tTmax$ is less than $rand$, otherwise, search spaces are explored. $Npop$, $Tmax$, and SF represent the MRFO control system crucial for ensuring its optimal performance [50].

The study by Andi Tang et al. identified limitations in Manta Ray Foraging Optimization (MRFO), including random reference point selection in early iterations reducing exploitation capability, chain foraging leading to local optima, and declining population diversity in later iterations. To address these, the study introduced m-MRFO, incorporating an elite search pool (ESP) for better exploitation, adaptive control parameter strategies (ACP) to balance exploration and exploitation, and a distribution estimation strategy (DES) to adjust evolutionary direction based on dominant population information, ultimately aiming to enhance convergence [51]. MRFO involves the emulation of three foraging tactics employed by manta rays:

- a. Chain Foraging: Manta rays create a foraging chain by connecting their heads and tails in a linear formation. MRFO posits that the optimal solution involves a greater density of plankton, the primary food source for manta rays. The initial individual in the chain moves directly towards the food source, while subsequent individuals move both towards the food and towards those positioned ahead of them in the foraging chain [51].
- b. Cyclone Foraging: Manta rays, upon locating plankton in deep waters, organize into lengthy foraging chains and adopt a spiral movement towards the food source. This behavior shares similarities with the Whale Optimization Algorithm (WOA), wherein the spiral movement near food is coupled with a tendency to follow individuals ahead in the chain. In this foraging strategy, food serves as a primary reference point for the spiral movement, facilitating comprehensive exploration of the vicinity near the food source. The integration of this random spiral foraging mechanism primarily emphasizes exploration, enabling MRFO to conduct an extensive global search [51].
- c. Somersault Foraging: During this phase, the food location acts as a central point of rotation. Each individual flips around this pivot point, thereby exploring and searching for a new location [51].

To enhance MRFO, an elite search pool was employed, transitioning effectively from exploration to exploitation using adaptive parameters. Adjusting the evolutionary path was done through a distribution estimation strategy, boosting population diversity and algorithm performance. A thorough explanation of the mathematical model for m-MRFO is presented as follow [51]:

- a. Elite Search Pool Strategy (ESP): Cyclone foraging initially uses random positions in the search space, improving exploration but hindering exploitation. To strike a balance, an elite search pool strategy is introduced, incorporating the top three individuals as shown in Eq. (2.20):

$$x_{esp} = \{X_{esp1}, X_{esp2}, X_{esp3}\} \quad (2.20)$$

- b. X_{esp1} , X_{esp2} , and X_{esp3} represent the top three performing individuals. The reference point is now selected from the best three individuals using the ESP strategy, reducing randomness. To prevent premature convergence, the three individuals are chosen randomly. To balance exploitation and exploration, a new individual is added, composed of random traits from the top three. The ESP strategy's mathematical model is as follows:

$$\begin{aligned} x_{esp} &= \{x_{esp1}, x_{esp2}, x_{esp3}, x_{espr}\} \\ x_{espr} &= r_{13} \cdot x_{esp1} + r_{14} \cdot x_{esp2} + r_{15} \cdot x_{esp3} \end{aligned} \quad (2.21)$$

Here $r \in [0, 1]$, represent randomly generated vectors following a uniform distribution [51].

- c. Adaptive Control Parameter Strategy (ACP): Nonlinear parameter control prevents premature convergence. This study suggests an adaptive control strategy using a mix of sine and cosine functions, represented by the *Coef* mathematical model of parameter as follows:

$$Coef = \sin(0.5\pi \cdot iter / iter_{max})^{(2.5\cos(iter/iter_{max})^3)} \quad (2.22)$$

- d. The new strategy prioritizes early exploration to avoid local optima, shifting to increased exploitation for faster convergence later. The parameter S remains constant in the original MRFO's somersault foraging, limiting effectiveness. Early stages demand a larger S for more exploration, while later stages require a smaller S for precision in exploitation. A linear decreasing strategy for S is proposed with the following mathematical model.

$$S = (Smin - Smax) / iter_{max} \cdot iter + Smax \quad (2.23)$$

Where $Smin$ and $Smax$ represent the minimum and maximum values for parameter S [51].

3. THEORETICAL BACKGROUND

3.1 INTRODUCTION

This chapter delves into the theoretical underpinnings of Alzheimer's Disease (AD) and its biological aspects. It commences by elucidating the nature of AD, delineating its various types, and subsequently delving into a genetic explanation of the disease.

A pivotal focus of the chapter is gene expression, its significance, and the challenges it presents. Additionally, it explores the process of gene selection, elucidating methods like information gain that can be employed for this purpose. Furthermore, the chapter scrutinizes feature selection, expounding on its techniques, limitations, and its application in the context of AD research.

Moreover, the chapter delves into machine learning, explicating different classification methods pertinent to the study of AD. Notably, it examines the Support Vector Machine (SVM) classification model, Random Forest (RF), Decision Tree (DT), XGBoost, and Naive Bayes, and underscoring its role as a classifier in machine learning. Finally, the chapter identifies and discusses contemporary challenges associated with gene selection using nature-inspired algorithms.

3.2 ALZHEIMER DISEASE

Alzheimer's disease (AD), named after the German psychiatrist Alois Alzheimer, represents the most prevalent form of dementia. It is characterized by a gradual and irreversible decline in cognitive function, impacting memory, thinking abilities, and the ability to perform daily tasks independently. The pathology of AD is marked by the accumulation of amyloid-beta peptide ($A\beta$), leading to the formation of neurotic plaques and neurofibrillary tangles, primarily in the medial temporal lobe and neocortical regions of the brain. This accumulation results in the degeneration of brain cells, particularly neurons [52].

Alois Alzheimer first identified these pathological features in the brain of his initial patient, who exhibited symptoms such as memory loss and personality changes before death. Alzheimer termed this condition a severe cerebral cortex disease. Subsequently, Emil

Kraepelin officially named the disorder Alzheimer's disease in the 8th edition of his psychiatry handbook [52].

While AD is a primary cause of cognitive decline, it is crucial to note that other cerebral disorders, intoxications, infections, circulatory and pulmonary abnormalities leading to reduced oxygen supply to the brain, nutritional deficiencies, vitamin B12 deficiency, tumors, and various other factors can also contribute to progressive cognitive impairment [52].

3.2.1 Types of Alzheimer Disease

In recent years, significant strides have been made in comprehending Alzheimer's disease, with ongoing momentum in research. However, the precise causes of Alzheimer's in the majority of cases remain elusive to scientists. It is likely that a combination of age-related brain changes, genetic predisposition, environmental influences, and lifestyle factors contribute to the development of the disease. The significance of each of these factors in influencing the risk of developing Alzheimer's may vary from person to person. The scan below illustrates patient brain with Alzheimer disease.

Despite the similar impact of Alzheimer's disease (AD), it is categorized into two distinct types (early-onset AD (EOAD) and late-onset AD (LOAD)):

- a. EOAD: Early-onset Alzheimer's disease accounts for a small fraction (1-2%) of cases and is associated with autosomal dominant inheritance, reflecting the complex polygenic nature of Alzheimer's disease. When Alzheimer's disease is caused by autosomal dominant variants, it is classified as early-onset familial Alzheimer's disease (FAD), a condition that is comparatively rare and progresses more rapidly.

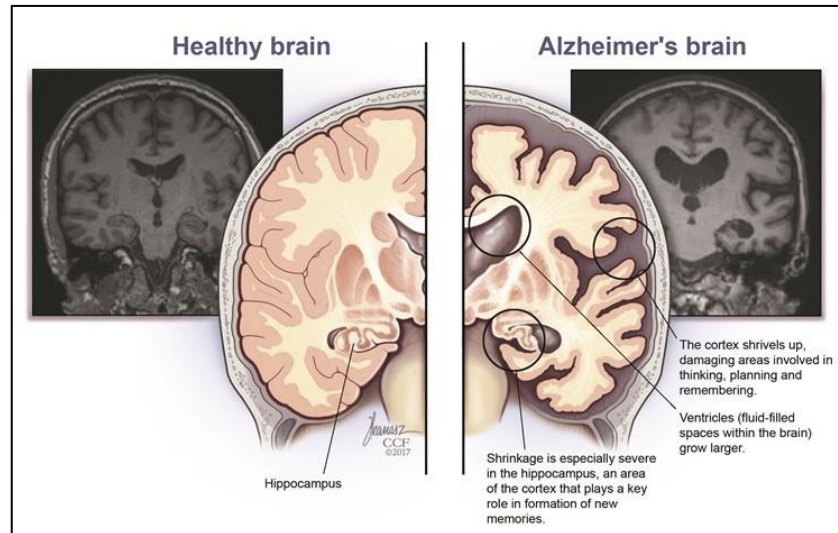


Figure 3.1: Comparison of a Healthy Brain with a Brain with Alzheimer Disease.

Early-onset Alzheimer's is highly heritable, with approximately 90% of cases showing a genetic component. In sporadic Alzheimer's disease, which comprises the majority of cases, less than 5% exhibit an earlier onset. The presence of FAD typically indicates the involvement of multiple affected individuals within one or more generations of a family [53].

- b. **LOAD:** Late-onset Alzheimer's disease (LOAD) is estimated to be about 70% heritable, indicating a strong genetic component. Recent genetic models, as of 2020, have shown the ability to predict Alzheimer's disease with an impressive 90% accuracy rate. It's important to note that while LOAD has a significant genetic influence, most cases of Alzheimer's are not familial and are thus referred to as sporadic Alzheimer's disease. The majority of sporadic Alzheimer's cases are late-onset, typically developing after the age of 65 years [54].

3.3 GENES AND GENE EXPRESSION

Genes, the fundamental units of inheritance, serve as the blueprints for cellular function. Encoded within their DNA sequences lies the information necessary to synthesize proteins and other functional molecules. Gene expression is the sophisticated and highly regulated process that translates the information encoded within genes into functional products, thus playing a pivotal role in the development of the complex biological systems that constitute life.

Gene expression encompasses two main phases: It encompasses two main phases: transcription, where DNA is transcribed to messenger RNA (mRNA), and translation, in which mRNA is decoded by ribosomes to synthesize proteins. This newly coded mRNA then serves as the intermediary, carrying the genetic code to the ribosomes, the protein factories of the cell. Here, during translation, the mRNA is deciphered, and the encoded amino acids are assembled into a specific protein, the functional product of gene expression [55]. Figure 3.2 depicts the transcription and translation of mRNA.

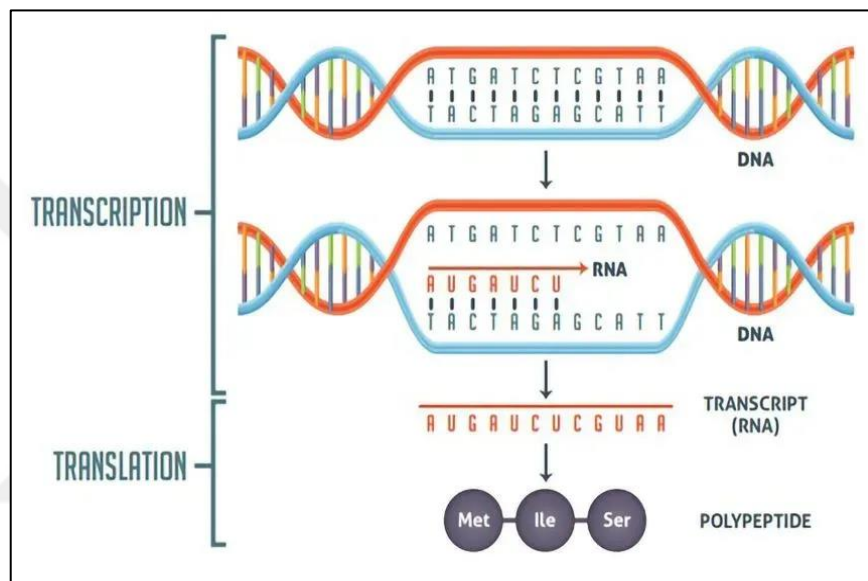


Figure 3.2: Transcription vs Translation.

However, gene expression is not a haphazard process. It is tightly regulated by a complex interplay of factors, ensuring the production of the right proteins in the right amounts at the right time. Promoter regions, specific DNA sequences, act as the conductor, initiating transcription by attracting RNA polymerase, the enzyme responsible for mRNA synthesis. Transcription factors, akin to the individual musicians, bind to these promoter regions, fine-tuning the level of transcription. Additionally, epigenetic modifications, chemical alterations to DNA or its associated proteins, can influence gene expression without altering the underlying DNA sequence itself, adding another layer of control to this intricate symphony [55].

Understanding the intricacies of gene expression is fundamental to various scientific disciplines. In the realm of medicine, deciphering the dysregulation of gene expression in

diseases paves the way for the development of targeted therapies. Biotechnology leverages this knowledge to engineer organisms with desirable traits, while agriculture strives to optimize gene expression for improved crop yields.

It is important to remember that gene expression is a dynamic process. Only a select subset of genes is actively expressed within each cell at a given time, and this repertoire is constantly changing based on the cell's specific needs and environmental cues. Furthermore, external factors such as nutrition, chemicals, and radiation can influence gene expression, highlighting the intricate interplay between genes and the environment. Finally, mutations within genes can disrupt the delicate balance of expression, potentially contributing to the development of various diseases [56].

In conclusion, genes and gene expression form the cornerstone of life, dictating the production of the molecular machinery that governs cellular function and ultimately shaping the observable traits of an organism. By unraveling the intricacies of this complex process, we gain a deeper understanding of the symphony of life and unlock the potential for advancements in various scientific fields.

3.4 ALZHEIMER'S DISEASE AND GENES

Early-onset AD (EOAD) and late-onset AD (LOAD), the two main forms of Alzheimer's disease, have distinct genetic landscapes. While three genes - APP, PSEN1, and PSEN2 - are well-established contributors to EOAD, the genetics of LOAD remain largely enigmatic. The APOE ϵ 4 allele is a known risk factor for both forms, but its presence doesn't fully explain LOAD susceptibility [57]. Recent advancements in next-generation sequencing have identified potential novel EOAD-linked genes like SORL1, TYROBP, and NOTCH3. Additionally, TYROBP variants identified through exome sequencing suggest their potential role in EOAD pathogenesis [57].

The ϵ 4 allele undeniably increases EOAD severity, but its presence in individuals who never develop the disease hints at the involvement of unidentified LOAD genetic and environmental risk factors [57]. Deciphering the missing pieces of the EOAD and LOAD genetic puzzle is crucial for understanding the disease's pathological mechanisms. This knowledge can pave the way for the development of novel therapeutic strategies to alleviate, reverse, and ultimately prevent Alzheimer's disease [57]. This analysis focuses on

currently known susceptibility genes for both forms of the disease, explores potential molecular mechanisms underlying their genetic etiology, and emphasizes the importance of massively parallel sequencing technologies in discovering novel genes associated with Alzheimer's disease [57].

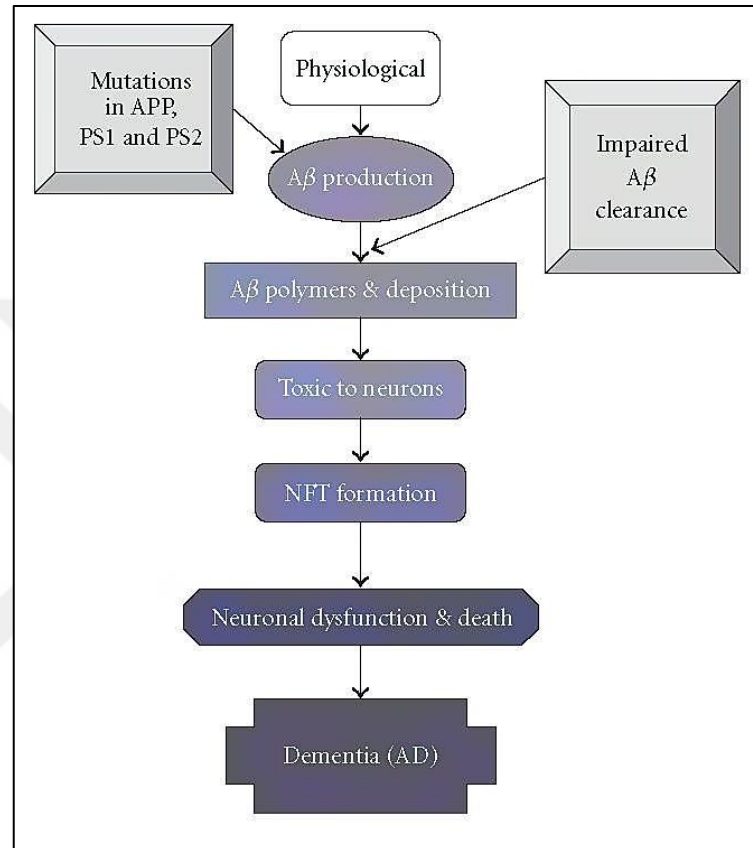


Figure 3.3: The Amyloid Cascade Hypothesis.

3.5 MICROARRAY TECHNOLOGY

DNA microarrays are sophisticated tools used to analyze the expression levels of thousands of genes simultaneously. They offer several advantages, including reproducibility, scalability, and the ability to perform experiments quickly. Microarrays are an extension of hybridization-based methods that have been used for decades in molecular biology to identify and quantify nucleic acids in biological samples. In a typical microarray experiment, RNA is extracted from tissues, labeled, and allowed to hybridize to the array. The intensity of hybridization is then measured to determine the expression level of each gene on the array [58].

There are two main types of DNA microarrays: spotted cDNA microarrays and oligonucleotide chips, as shown in Figure 3.4. Spotted microarrays consist of a solid surface onto which small amounts of single-stranded nucleotide sequences are deposited in a grid-like arrangement. Each spot represents a specific gene and serves as a probe for hybridization. Oligonucleotide chips, on the other hand, synthesize probes directly on the array based on gene sequences. Both types of microarrays can accommodate thousands of probes, allowing for high-density arrays [58].

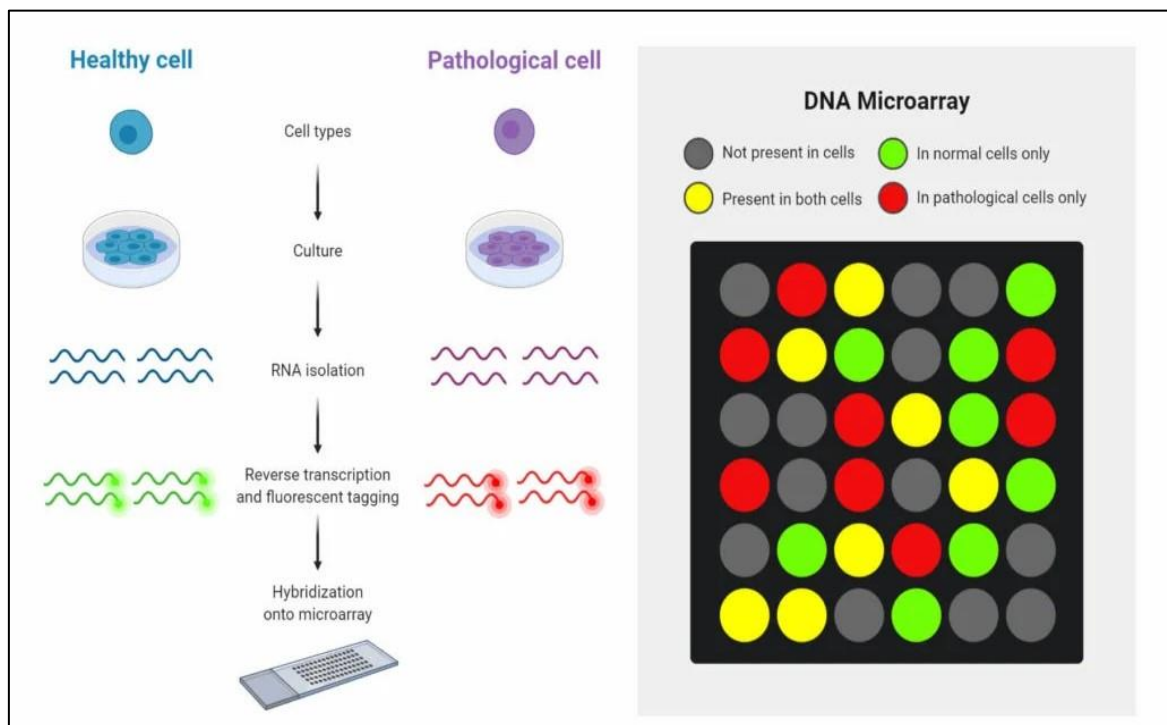


Figure 3.4: Spotted cDNA-based Microarray.

Microarray experiments are used to study relationships between biological samples based on gene expression measurements in different conditions. They can also be used to study the development of biological phenomena over time. Microarray data analysis involves preprocessing steps to normalize and integrate the data, followed by exploratory data analysis and the application of advanced computational methods, such as support vector machines (SVM) and singular value decomposition (SVD), for tasks like clustering, classification, and feature selection [58].

Overall, microarray experiments provide a wealth of data that require advanced computational methods for analysis. These methods range from classic statistical

approaches to machine learning techniques and methods from artificial intelligence. The choice of analysis technique depends on the data and the goals of the experiment. Supervised and unsupervised methods are two broad categories of analysis techniques, each with its own advantages and applications [58].

3.5.1 Alzheimer's Disease and Genes Expression

Alzheimer's disease (AD) is a complex neurodegenerative disorder characterized by progressive cognitive decline and memory loss. While the precise etiology of AD remains incompletely understood, genetic factors are recognized as significant contributors to its pathogenesis. Investigations into gene expression patterns have yielded valuable insights into the underlying molecular mechanisms of AD [59].

AD is characterized by the accumulation of abnormal protein deposits, including amyloid plaques and tau tangles, in the brain. Gene expression studies have revealed dysregulation of genes involved in the production and clearance of these proteins in AD patients. For instance, the amyloid precursor protein (APP) gene, which is processed to produce amyloid-beta ($A\beta$) peptides, is upregulated in AD brains. Similarly, genes related to $A\beta$ clearance, such as the apolipoprotein E (APOE) gene, show altered expression in AD [60]. Beyond genes directly linked to $A\beta$ metabolism, studies have identified dysregulation of genes associated with inflammation, synaptic function, and neuronal signaling in AD brains. These findings underscore the multifaceted nature of AD pathology, implicating diverse pathways and processes within the brain [60].

Recent advancements in gene expression profiling, particularly with RNA sequencing, have enabled more comprehensive and unbiased analyses of gene expression changes in AD. These studies have identified novel genes and pathways relevant to AD pathogenesis, presenting new prospects for therapeutic strategies [61].

In summary, gene expression studies have significantly enhanced our understanding of the molecular underpinnings of AD. By elucidating key genes and pathways involved in the disease, these studies have opened avenues for the development of innovative treatments and interventions for AD [61].

3.5.2 Genes Selection

In gene expression studies, selecting pertinent genes for sample classification is a critical task aimed at identifying the smallest gene subset that can yield optimal predictive performance, particularly for future clinical diagnostic applications. Many gene selection methods rely on univariate gene ranking and arbitrary thresholds to determine the number of genes selected. However, these approaches are limited to two-class problems and employ gene ranking criteria that are unrelated to the classification algorithm [62].

Gene selection represents a critical phase in microarray analysis, a method utilized for the simultaneous examination of the expression of numerous genes. Given the extensive volume of data involved, pinpointing the most pertinent genes for a particular study emerges as a fundamental task. This overview accentuates the pivotal facets of gene selection and its paramount importance in scientific inquiry [63].

Gene selection is significant because it targets pertinent genes, allowing the analysis to concentrate on those with the most substantial impact on the study goal. This focused approach leads to more precise and understandable outcomes. Additionally, it addresses the issue of high dimensionality often present in microarray data, which can cause computational complexities and potentially misleading conclusions if all genes are considered [63].

By narrowing down the genes under scrutiny, gene selection simplifies the analysis and highlights the most informative genes. Moreover, gene selection enhances classification accuracy, particularly in tasks such as disease diagnosis, by excluding irrelevant genes and emphasizing those that significantly contribute to the observed phenotype [64]. Furthermore, gene selection plays a crucial role in identifying potential biomarkers by highlighting genes with notable expression differences between different groups (e.g., healthy individuals and those with a disease). This process aids in the discovery of biomarkers that could be valuable for disease diagnosis, prognosis, or predicting treatment responses [65].

Various approaches are commonly employed for genes or features selection. Filter-based methods rank genes using statistical measures such as t-tests or fold-change, selecting those that surpass a predetermined threshold [66]. Wrapper methods involve using a classification algorithm to assess the performance of different gene subsets, refining the

selection iteratively until optimal performance is achieved [65]. Embedded methods integrate gene selection within the classification algorithm itself, automatically selecting relevant genes during the model building process [67].

The choice of approach depends on factors such as the nature of the research question, the desired level of interpretability, and the available computational resources. No single method is universally superior, and a combination of approaches is often necessary to obtain the most reliable results.

3.6 FEATURE SELECTION

Feature selection, also known as variable or attribute selection, is a fundamental process in machine learning and statistics aimed at identifying a subset of relevant features from a larger set for model construction. The primary objective is to enhance model performance by reducing overfitting, computational complexity, and improving interpretability through the selection of the most informative and discriminative features [68].

In situations where the number of features exceeds the number of samples, a common challenge in high-dimensional datasets, feature selection becomes pivotal. Utilizing all available features can result in issues such as heightened model complexity, diminished generalization performance, and prolonged training times. By pinpointing a subset of relevant features, this process allows models to concentrate on the most pertinent information, leading to more efficient and effective outcomes [68].

There are several approaches to feature selection, including filter methods, wrapper methods, and embedded methods. Filter methods evaluate feature relevance using statistical measures like correlation or information gain, independent of the learning algorithm. Wrapper methods, conversely, utilize a specific learning algorithm to assess different feature subsets and select the one that optimizes model performance. Embedded methods integrate feature selection into the model construction process, with feature selection guided by the learning algorithm [68].

In summary, feature selection is a crucial aspect of machine learning workflows, significantly enhancing model performance and interpretability, particularly in high-dimensional datasets.

3.6.1 Techniques of Features Selection

In the past decade, the significance of employing feature selection (FS) techniques in bioinformatics has transitioned from being merely illustrative to becoming a crucial prerequisite for constructing models. This shift is primarily due to the high-dimensional nature of many modeling tasks in bioinformatics, which encompass sequence analysis, microarray analysis, spectral analyses, and literature mining, leading to the introduction of numerous feature selection techniques in the field.

In the context of classification, techniques for feature selection can be categorized into two main groups based on how they integrate the feature selection process with the construction of the classification model: filter methods and wrapper methods.

3.6.1.1 Filter methods

Filter methods utilize variable ranking techniques as the primary criteria for variable selection through ordering. Ranking methods are preferred due to their simplicity and demonstrated effectiveness in practical applications. A specific ranking criterion is employed to assign scores to variables, and a threshold is then applied to exclude variables scoring below it. These ranking methods are considered filter methods as they are applied prior to classification to eliminate less relevant variables [69]. The following are four of the widely used techniques in literatures:

Correlation based Feature Selection

This method focuses on identifying features that exhibit redundancy. It aims to find features that show high correlation with the target variable while maintaining low correlation among themselves. The selection process involves calculating the correlation coefficient for each pair of features. The feature with the highest correlation coefficient is selected first. The equation for feature selection based on correlation is [69]:

$$M_S = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}} \quad (3.1)$$

Fast Correlation-based Filter (FCBF)

FCBF begins with a complete set of features and employs symmetrical uncertainty to measure feature dependency. It eliminates redundant features through backward selection. FCBF includes an internal stopping criterion to prevent excessive feature elimination. This method is known for its speed compared to other feature selection techniques [69].

Information Gain

This metric, rooted in information theory and entropy, quantifies the reduction in disorder or noise when incorporating features. Features with higher information gain are prioritized. However, it's important to note that Information Gain does not eliminate redundant features. The computation of the information gain regarding X given by Y is determined as follows [69]:

$$IG(X | Y) = H(X) - H(X | Y) \quad (3.2)$$

Chi-squared Statistics

This method assesses the association between two categorical variables. Therefore, numeric variables must be discretized into multiple intervals. The Chi-square statistic is derived in the following equation [69]:

$$\chi^2 = \sum_{i=1}^m \sum_{j=1}^k \left(\frac{A_{ij} - E_{ij}}{E_{ij}} \right)^2 \quad (3.3)$$

3.6.1.2 Wrapper methods

Wrapper methods operate similarly to filter methods but utilize a predetermined classification algorithm instead of an independent measure for subset evaluation. While wrapper methods generally yield superior results to filter methods, they can be computationally intensive, especially with a large number of features. The first two search techniques in wrapper methods, outlined in the following section, are the two most commonly used greedy methods for feature selection [69]:

- a. Sequential Forward Selection (SFS): SFS begins with an empty set and performs well with a small number of features. However, its main drawback is the inability to eliminate features that become insignificant after the addition of other features [69].
- b. Sequential Backward Elimination (SBE): SBE, also known as Sequential Backward Selection (SBS), starts with a full set of features and works best with a large number of features in the dataset [69].
- c. Plus-L-Take-Away-R: This method aims to address the nesting effect seen in SFS and SBE, where once features are selected or removed, they cannot be reversed. The approach allows SFS to use "l" forward and then "r" back-tracking steps of SBS. Selecting suitable (l, r) values is challenging but crucial for good results with moderate computation [69].
- d. Sequential Floating Forward and Backward Selection (SFFS and SFBS): These methods control backtracking without parameter setting, offering a more flexible approach. The number of forward and backtracking steps is dynamically adjusted, making these methods highly effective for feature selection. Unlike the plus-l-take-away-r method, floating methods allow dynamic addition and deletion of feature subsets until an optimal number of feature subsets is achieved [69].

3.6.2 Wrapper Features Selection Techniques Compared to Filter Techniques

Wrapper and filter-based features selection are two distinct approaches used in the domain of machine learning and data mining. Wrappers utilize an optimization algorithm as a search technique to identify a subset of features that offers the optimal performance. Wrappers can be used to increase the performance of machine learning model through reduction of high dimension of dataset, or increase prediction accuracy through removal of irrelevant features. Both of which are evaluated in optimization algorithm fitness function. Wrappers based method offers several advantages over the filter approach, such as [109, 110]:

- a. Higher Accuracy and Model Performance: wrappers assess the accuracy of selected features based on the model prediction outcome, this results in improved accuracy and overall performance of the selected features that are iteratively tailored to the utilized specific model.

- b. **Features Relationship and Interactions:** unlike filter methods, which evaluate the selected subset of features based on their correlation to the target variable, wrappers consider how the combinations of features can together influence the model's prediction such that complex and not easy to discover relationship patterns can be discovered and examined.
- c. **Flexibility in Support of Different Learning Models:** wrapper approach can be applied to a variety of learning models, depending on the case at hand, several models can be tested or an ensemble of models can be used to overcome specific weaknesses or to gain specific strength [111].

3.6.3 Weaknesses of Wrapper Features Selection Techniques

Since wrappers are inherently dependent on utilized learning model, wrapper methods can suffer from several weaknesses compared to filter methods:

- a. **Higher Computational Demands:** Wrapper methods are more computationally intensive because they require training and evaluating the model multiple times for each subset of features. They achieve high classification accuracy only for the specific learning algorithm used. As a result, they lack generality and have higher computational complexity compared to embedded and filter methods. This can be especially challenging with a large number of features or a complex model [70].
- b. **Risk of Overfitting:** Due to their dependency on a particular machine learning algorithm for evaluating feature subsets, wrapper methods are susceptible to overfitting, especially when dealing with small or noisy datasets [70].
- c. **Large Search Space:** The search space in wrapper methods can be extensive, particularly with numerous features. Conducting a comprehensive search of all potential feature subsets may not be practical, resulting in the adoption of heuristic search techniques that might not identify the optimal subset [71].
- d. **Model Bias:** The selection of the machine learning algorithm in wrapper methods can introduce bias into the feature selection process. If the algorithm is not suitable for the dataset or the problem at hand, it may choose features that do not generalize effectively to unseen data [71].

- e. **Limited Generalization:** Because wrapper methods choose features based on a particular machine learning algorithm, the features selected may not generalize effectively to different algorithms or datasets. This limitation can restrict the usefulness of the feature selection outcomes [72].
- f. **Computational Cost:** The iterative training and evaluation process in wrapper methods can be computationally costly, particularly when contrasted with filter methods that assess features without relying on a particular model [72].
- g. **Dependency on Model Performance:** The efficacy of wrapper methods relies significantly on the performance of the underlying machine learning algorithm. If the algorithm is not well-suited to the data or the problem, the results of the feature selection may not be optimal [72].

3.7 MACHINE LEARNING AND CLASSIFICATIONS

Machine-learning (ML) algorithms are known for their ability to model intricate class characteristics and can handle various types of input predictor data without assuming a specific data distribution (i.e., they are nonparametric). Numerous studies have shown that these algorithms often achieve higher accuracy compared to traditional parametric classifiers, especially when dealing with complex data featuring a high-dimensional feature space, meaning there are many predictor variables. The use of machine-learning approaches has become widespread, as demonstrated by their application in operational land-cover mapping [73]. Despite the growing popularity of machine-learning classifiers, parametric methods continue to be commonly employed in research articles and remain a key benchmark for evaluating classification experiments [73].

ML is a broad interdisciplinary field that draws on concepts from computer science, statistics, cognitive science, engineering, optimization theory, and various other branches of mathematics and science. Among its numerous applications, data mining stands out as particularly significant. Machine learning is primarily categorized into two main types: supervised machine learning and unsupervised machine learning [74].

Unsupervised machine learning involves analyzing datasets without labeled responses or predefined outcomes. In unsupervised learning, the desired output is not provided. On the other hand, supervised machine learning techniques aim to establish relationships between

input attributes (independent variables) and a target attribute (dependent variable). Supervised techniques can be further classified into two main categories: classification and regression. In regression, the output variable takes continuous values, while in classification, the output variable takes class labels [74]. Figure 3.5 below illustrates typical classification models of supervised learning techniques.

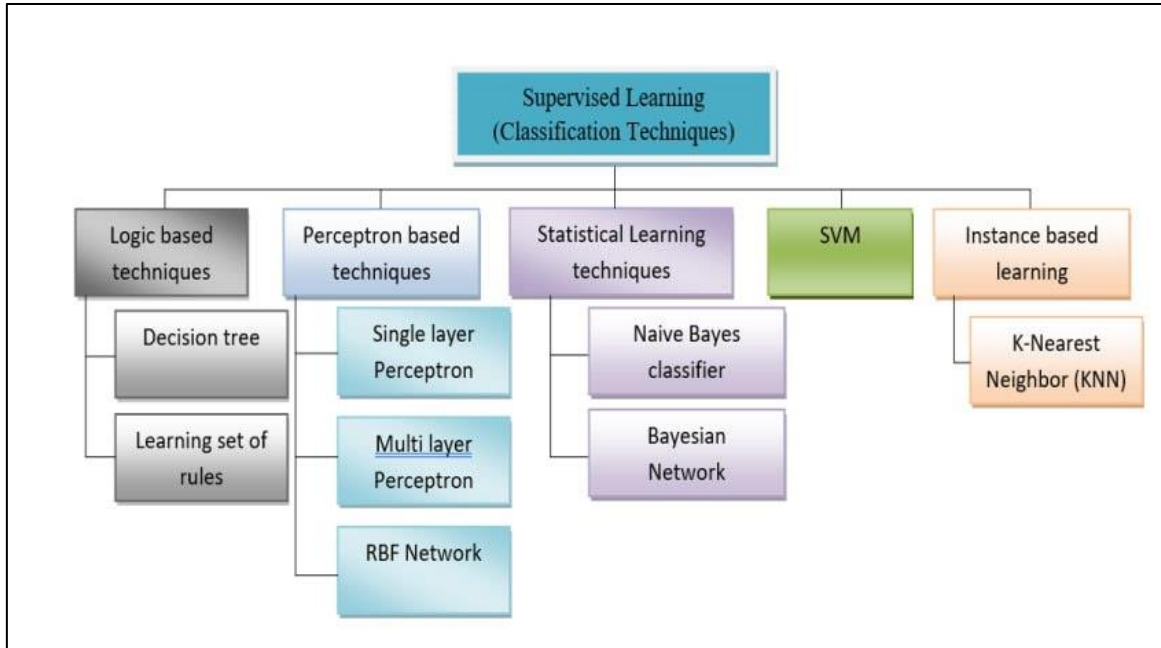


Figure 3.5: Classification Techniques in Supervised Learning.

Classifications are thought of as data mining algorithms used to predict group membership for data instances. While there are various machine learning techniques available, classification is the most widely used. It is a popular task in machine learning, particularly in future planning and knowledge discovery, and is considered one of the most extensively studied problems in the machine learning and data mining fields [74].

- a. **Supervised Machine Learning:** In supervised algorithms, classes are predefined in a finite set determined by humans. This means that a specific portion of the data will be labeled with these classifications. The algorithm's task is to discover patterns and create mathematical models, which are then assessed based on their predictive accuracy relative to the data's variability [75]. It's important to distinguish between two main types of supervised models: classification models (classifiers) and regression models. Regression models map the input space to a real-value range, while classifiers map the input space to predefined classes. There are various methods for representing

classifiers, such as support vector machines, decision trees, probabilistic summaries, and algebraic functions [75]. Classification, along with regression and probability estimation, is among the most studied models and holds significant practical relevance. Progress in classification techniques can have immense benefits, as it has a substantial impact on various areas, both within Data Mining and its applications [75].

- b. **Unsupervised Machine Learning:** In contrast, unsupervised learning algorithms operate without predefined classifications. Their primary objective is to autonomously generate classification labels. These algorithms seek similarities among data points to determine if they can be grouped together to form clusters. Clusters represent families of clustering machine learning techniques. In unsupervised classification (cluster analysis), the machine does not have prior knowledge of how the clusters should be grouped. Cluster analysis has the potential to reveal unexpected relationships, making it a valuable tool for exploring connections among numerous papers [75].

3.7.1 Support Vector Machine (SVM)

Support Vector Machines (SVM) is a robust algorithm with a strong theoretical foundation based on the Vapnik-Chervonenkis theory, renowned for its strong regularization properties that enable generalization to new data. SVM was specifically designed for supervised learning classification problems, where a training set of data with known classifications is provided. SVMs are a group of related supervised learning methods used for both classification and regression tasks, belonging to the generalized linear classification family. A notable characteristic of SVM is its ability to simultaneously minimize the empirical classification error and maximize the geometric margin, earning it the nickname of Maximum Margin Classifiers [76].

SVM operates on the principle of Structural Risk Minimization (SRM), which involves mapping input vectors to a higher-dimensional space where a maximal separating hyperplane is constructed. This hyperplane is positioned to maximize the distance between two parallel hyperplanes, which are built on each side of the separating hyperplane to effectively segregate the data. The kernel function plays a crucial role in this process by elevating the data to a higher-dimensional space, thereby facilitating separation. Kernel functions, including SVM, are utilized for pattern analysis or recognition. In this process, training vectors x_i are transformed into a higher (potentially infinite) dimensional space by

the function Φ . Subsequently, SVM identifies a linear separating hyperplane with the maximum margin in this higher-dimensional space. The parameter $C > 0$ signifies the penalty parameter of the error term. Kernel function can be seen below [76]:

$$K(x_i, x_j) \equiv \Phi(x_i)^T \Phi(x_j) \quad (3.4)$$

The fundamental components and principles of SVM are as follows:

- a. **Hyperplane:** In SVM, the hyperplane acts as the decision boundary segregating data points into distinct classes. In a binary classification scenario, the hyperplane manifests as a line in two dimensions, a plane in three dimensions, and a hyperplane in higher dimensions [77].
- b. **Support Vectors:** These are data points closest to the hyperplane, exerting the most influence on its placement. They are pivotal in determining the optimal hyperplane [77].
- c. **Margin:** The margin denotes the distance between the hyperplane and the nearest data point from either class. SVM strives to maximize this margin as it signifies a clearer separation between classes and enhances generalization to unseen data [78].
- d. **Kernel Trick:** SVM adeptly performs linear classification in high-dimensional spaces by employing the kernel trick. This method transforms input data into a higher-dimensional space, facilitating the identification of a hyperplane that separates classes. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid [79].
- e. **Regularization Parameter (C):** Denoted as C , this parameter governs the balance between maximizing the margin and minimizing the classification error. A higher C value leads to a narrower margin but may result in overfitting, while a lower C value may result in a wider margin but potentially more misclassifications [80].
- f. **Soft Margin SVM:** In cases where data is not linearly separable, a soft margin SVM permits some misclassifications to achieve a better overall fit. The regularization parameter C determines the penalty for misclassifications [81].
- g. **Multiclass Classification:** SVM can be extended to handle multiclass classification using techniques such as one-vs-one or one-vs-rest, where multiple binary classifiers are trained and combined to make predictions for multiple classes [82].

- h. Regression: SVM can also be utilized for regression tasks, predicting a continuous output rather than a discrete class label. In regression, the objective is to fit a function that predicts the output value based on the input features [83].

Overall, SVM stands out as a versatile and robust algorithm for classification and regression tasks, renowned for its prowess in handling high-dimensional data and intricate decision boundaries.

3.7.2 Decision Tree

Data mining, which aims to extract valuable insights from large datasets and present them in understandable visualizations, has become increasingly important. Decision trees, first introduced in the 1960s, stand out as a highly effective tool in this field. They are widely utilized across various disciplines due to their ease of use, clarity, and robustness even when faced with missing data. Decision trees can handle both discrete and continuous variables as either target or independent variables. In recent years, decision tree methodology has gained popularity in medical research, particularly in diagnosing medical conditions based on symptom patterns. This approach can classify different clinical subtypes or determine appropriate therapies for patients with specific conditions [84].

Decision trees offer several practical applications, including variable selection, assessing variable importance, handling missing values, prediction, and data manipulation. They can help in selecting relevant input variables, identifying key variables, dealing with missing data by either classifying them separately or using a predictive model, and predicting outcomes based on historical data. Additionally, decision trees can aid in simplifying overly complex data by collapsing categories or subdividing skewed variables into manageable ranges [84].

Figure 3.6 below shows a basic decision tree model comprising a single binary target variable, Y (0 or 1), and two continuous variables, x_1 and x_2 , ranging from 0 to 1. Decision tree models consist of nodes and branches, and three key stages in constructing tree model which include splitting, stopping, and pruning [84].

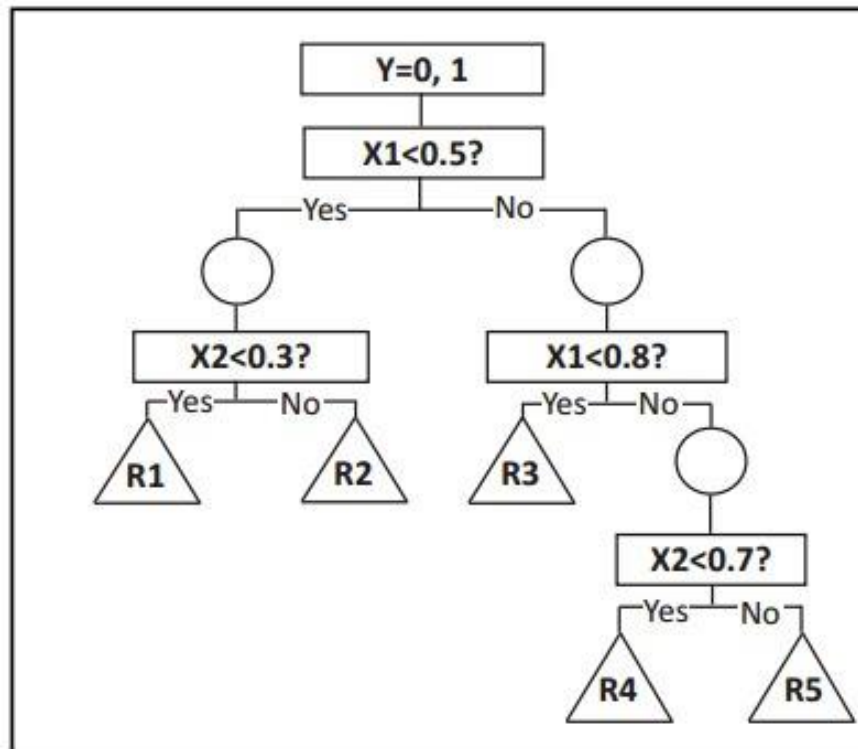


Figure 3.6: A Decision Tree Example Using a Binary Target Variable Y.

Nodes in decision tree are categorized into three types: root nodes, internal nodes, and leaf nodes. Root nodes serve as the starting point, dividing all records into distinct subsets. Internal nodes, or chance nodes, present available choices in the tree, connected to parent and child nodes. Leaf nodes, or end nodes, signify the final outcome after a series of decisions [84]. Branches in a decision tree represent outcomes from root and internal nodes, forming a hierarchy. Each path from the root to a leaf node embodies a decision rule, often expressed as an "if-then" statement, determining outcomes based on specific conditions [84].

With regard to tree operations, splitting is a pivotal step in decision tree construction, where input variables related to the target variable divide parent nodes into purer child nodes. Discrete and continuous variables can both be used for splitting, with the selection of significant input variables being crucial. Various criteria like entropy, Gini index, and information gain assess the purity of child nodes and guide the splitting process. This process continues until predefined homogeneity or stopping criteria are met, sometimes utilizing variables multiple times at different tree levels [34]. Stopping rules on the other

hand prevent overfitting and ensure model robustness. Parameters such as minimum records in a leaf and node, and leaf depth from the root, manage model complexity. Proper parameter setting is vital for balancing model complexity and reliability [84].

With large datasets, trees grow significantly large to the extent of hundreds and thousands of nodes and branches. Decision trees utilized a method known as Pruning where less contributed branches are removed. Pruning can prevent overfitting and improves model computational performance and prediction. There are two types of trees pruning: Pre-pruning aims to prevent non-significant branches, while post-pruning removes branches to improve overall classification accuracy [84].

Decision trees can be visualized as segmented spaces, such as the illustrated example below in Figure 3.7. The sample space is divided into distinct segments, each corresponding to a leaf node, which represents the final outcome of the decision rules. Each record is assigned to a single segment. The goal of decision tree analysis is to find the optimal model for dividing all records into these segments [84].

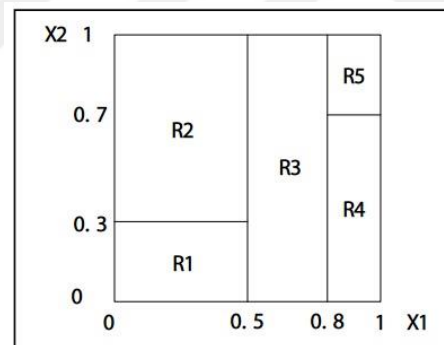


Figure 3.7: Decision Trees Represented Using a Sample Space View.

3.7.3 Random Forest

When analyzing survival data to pinpoint factors influencing mortality, researchers have typically used regression methods such as the Cox model. However, with greater computational power now accessible, more intricate approaches are gaining traction, particularly in the age of big data and machine learning. The Random Forest technique, for instance, is a regression tree method that enhances predictive accuracy through bootstrap aggregation and randomization of predictors [85].

The random forest algorithm, introduced by Breiman in 2001, has become a widely used method for developing prediction models. It combines multiple randomized decision trees, improving predictive accuracy, especially in scenarios with a high number of variables compared to observations. Unlike traditional decision trees, which may offer poor accuracy for complex datasets, random forests construct many trees using random subsets of training data and predictor variables. This approach often provides higher accuracy than a single decision tree model while maintaining the interpretability of relationships between predictors and outcomes [86].

One major advantage of random forests is their ability to handle datasets with a large number of predictor variables. However, in practice, it is often beneficial to minimize the number of predictors to improve efficiency. Variable selection is a common technique used to identify the most important predictors for inclusion in a simplified model. This reduces the burden of data collection and can improve prediction efficiency. Variable selection in the random forest framework is crucial for many applications, such as medical decision support systems or stock market projection models [86].

A random forest consists of a set of tree predictors $h(\mathbf{x}; \theta_k), k = 1, \dots, K$, Where $\mathbf{x}$ denotes the observed input vector (covariate) of length p , associated with the random vector $\mathbf{x}$ and θ_k represents independent and identically distributed (*iid*) random vectors [87].

For regression, the prediction from a random forest is calculated as the simple average over all the trees in the collection: $h(\mathbf{x}) = (1/K) \sum_{k=1}^K h(\mathbf{x}; \theta_k)$ where $K \rightarrow \infty$ the Law of Large Numbers guarantees.

$$E_{\mathbf{X},Y}(Y - \bar{h}(\mathbf{X}))^2 \rightarrow E_{\mathbf{X},Y}(Y - E_{\theta}h(\mathbf{X}; \theta))^2 \quad (3.5)$$

The value PE_f^* on the right side represents the prediction or generalization error for the random forest model. The convergence shown in (3.1) indicates that random forests do not overfit [87].

Now, let's define the average prediction error for an individual tree $h(\mathbf{x}; \theta_k)$ as follows:

$$PE_t^* = E_{\theta}E_{\mathbf{X},Y}(Y - h(\mathbf{X}; \theta))^2. \quad (3.6)$$

Assume for all θ the tree is unbiased, i.e., $EY = E_{\mathbf{X}}h(\mathbf{X}; \theta)$. Then:

$$PE_f^* \leq \bar{\rho} PE_t^* \quad (3.7)$$

where $\bar{\rho}$ represents the weighted correlation between residuals $Y - h(\mathbf{X}; \theta)$ for independent θ, θ' . The equation (3.7) highlights the key factors for accurate random forest regression: low correlation between residuals of different trees in the forest, and low prediction error for individual trees. Additionally, the random forest is expected to reduce the individual tree error PE_f^* by the factor $\bar{\rho}$. This injection of randomization aims to minimize correlation [87].

3.7.4 XGBoost (Extreme Gradient Boosting)

XGBoost is a scalable machine learning system designed for tree boosting, known for its exceptional scalability in various scenarios. It outperforms existing solutions by running more than ten times faster on a single machine and can scale to handle billions of examples in distributed or memory-limited environments [88]. This scalability is achieved through several key optimizations:

- a. A novel tree learning algorithm specifically designed for sparse data.
- b. The use of a weighted quantile sketch procedure, which is theoretically justified and allows for handling instance weights in approximate tree learning.
- c. Utilization of parallel and distributed computing, which accelerates the learning process and enables faster model exploration.
- d. Implementation of out-of-core computation, allowing data scientists to process hundreds of millions of examples even on a desktop machine.

Combining these optimizations results in an end-to-end system that can scale to handle even larger datasets with minimal cluster resources [88].

Here are some key features of XGBoost:

- i. Gradient Boosting: XGBoost uses gradient boosting to iteratively add new models to the ensemble, each correcting errors made by the existing models. It minimizes a

loss function by adding new models that move the predictions closer to the correct target values [89].

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (3.8)$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

- ii. **Regularization:** XGBoost incorporates regularization techniques to prevent overfitting. It includes L1 (Lasso) and L2 (Ridge) regularization terms in the objective function, controlling the complexity of the model [89].

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}, \quad (3.9)$$

- iii. **Tree Pruning:** Unlike traditional decision tree algorithms that grow the full tree and then prune it, XGBoost grows trees level-wise and prunes them backward, stopping the splitting process early when it is not beneficial [90].
- iv. **Cross-validation:** XGBoost supports built-in cross-validation to help users find the optimal number of boosting rounds. It can automatically stop boosting when the performance on a validation dataset stops improving [91].
- v. **Missing Values Handling:** XGBoost has a built-in mechanism to handle missing data. It learns the best direction to go when a value is missing, reducing the need for imputation [91].
- vi. **Parallel and Distributed Computing:** XGBoost is designed for efficiency and can take advantage of parallel and distributed computing resources, making it scalable to large datasets [92].
- vii. **GPU Support:** XGBoost offers GPU acceleration, allowing it to train models faster on compatible hardware [92].

- viii. Custom Objectives and Evaluation Criteria: Users can define custom objectives and evaluation criteria to tailor XGBoost to specific tasks or metrics [93].

Overall, XGBoost is widely used in machine learning competitions and real-world applications due to its speed, scalability, and ability to achieve high performance across a wide range of datasets and tasks.

3.7.5 Naive Bayes

The naive Bayes classifier simplifies learning by assuming that features are independent given the class, a simplification that can still yield competitive performance compared to more complex classifiers. Through simulations, we can systematically analyze classification accuracy across various randomly generated problems, exploring the impact of distribution entropy on classification error. It is observed that low-entropy feature distributions tend to result in improved performance for naive Bayes [94].

Surprisingly, naive Bayes also demonstrates strong performance with certain types of nearly-functional feature dependencies, showing optimal performance in scenarios with completely independent features as well as functionally dependent features. Additionally, the accuracy of naive Bayes is not directly correlated with the degree of feature dependencies, as measured by class-conditional mutual information. Instead, a more reliable predictor of naive Bayes accuracy is the amount of information lost about the class due to the independence assumption [94].

Naive Bayes learning algorithm has several key features:

- a. Computational efficiency: Training time scales linearly with the number of training examples and attributes, while classification time scales linearly with the number of attributes and is unaffected by the number of training examples [95].
- b. Low variance: Naive Bayes has low variance since it does not involve search operations, but this comes at the expense of higher bias [95].
- c. Incremental learning: It operates using estimates of low-order probabilities derived from training data, allowing for easy updates with new training data [95].

- d. Direct prediction of posterior probabilities: It can directly predict the probabilities of different classes given the input features [95].
- e. Robustness to noise: Naive Bayes uses all attributes for all predictions, making it relatively insensitive to noise in the examples to be classified. Additionally, its probabilistic nature makes it robust to noise in the training data [95].
- f. Robustness to missing values: Since it always uses all attributes for all predictions, missing attribute values do not significantly impact performance. It gracefully degrades in performance with missing values, and it is also relatively insensitive to missing attribute values in the training data due to its probabilistic framework [95].

Naive Bayes is a type of Bayesian Network Classifier that utilizes Bayes' rule:

$$P(y | \mathbf{x}) = P(y)P(\mathbf{x} | y)/P(\mathbf{x}) \quad (3.10)$$

This involves assuming that the attributes are conditionally independent given the class. For data with attribute-value pairs, this assumption leads to:

$$P(\mathbf{x} | y) = \prod_{i=1}^n P(x_i | y) \quad (3.11)$$

x_i represents the value of the i^{th} attribute in the $\mathbf{x}$, and n is the total number of attributes.

$$P(\mathbf{x}) = \prod_{i=1}^k P(c_i)P(\mathbf{x} | c_i) \quad (3.12)$$

Where k is the number of classes and c_i is the i^{th} class. Therefore, (3.10) can be computed by normalizing the numerators on the right-hand side of the equation [95].

The classifier that results from this process utilizes a linear model, which is similar to the one used in logistic regression, but differs in how the parameters are selected [95].

3.7.6 ML Performance Evaluation Measurements

An important advancement in machine learning in recent decades has been the realization that a single aggregated measurement is insufficient for accurately assessing algorithm performance. Provost, Fawcett, and Kohavi (1998) emphasized the limitations of relying solely on predictive accuracy in classification and advocated for the use of ROC analysis. They showed how ROC analysis is beneficial in dealing with changing class and cost distributions. The ROC convex hull method, introduced by Provost and Fawcett (2001),

was later found to be equivalent to using isotonic regression to derive calibrated probabilities from a classifier [96].

ROC analysis is based on multi-objective optimization, aiming to delay the choice of trade-offs between different objectives as much as possible. In classification, the optimization objectives are per-class accuracies, and the Pareto front is represented by the ROC convex hull. ROC analysis serves as a conceptual tool, leading to the development of new decision tree splitting criteria and a deeper understanding of performance evaluation metrics. Cost curves, introduced by Drummond and Holte (2006), explicitly represent trade-offs and the loss incurred at specific operating points, providing a complementary view to ROC curves [96].

Both ROC curves and cost curves provide meaningful aggregated performance measures. The area under the ROC curve (AUC) not only estimates the probability that a random positive is ranked higher than a random negative but also has a linear relationship with the expected classification accuracy of a classifier. The area under the cost curve equals the Brier score of a probabilistic classifier, measuring the mean squared residuals compared to the 'ideal' probabilities of 0 and 1 [96].

These techniques signify a departure from solely focusing on decision boundaries to considering the entire range of operating points. Calibration over the whole range of predicted probabilities allows for easy adjustment of the decision threshold when class distributions change. This approach enables more robust and flexible machine learning models, especially in scenarios with evolving class distributions [96].

3.7.6.1 Evaluation metrics

Evaluation can be classified into two types: qualitative evaluation and quantitative evaluation. Qualitative evaluation involves obtaining feedback from users or user groups to assess the satisfaction with the results of an information retrieval system. It emphasizes the users' experiences with the system. In contrast, quantitative evaluation involves quantifying the results in numbers [97].

Quantitative evaluation serves various purposes and may involve different constraints on the amount of data used for training and evaluation. Prioritizing high recall over high

precision, or vice versa, depends on the specific context. During system development, a development set, which can be data for rule development or training material for a machine learning system, is utilized. In machine learning, this set is often referred to as the training set, used to train the machine learning system. A portion of the training set is reserved for error analysis, known as the development test set, which helps adjust the machine learning algorithm based on errors, a process called parameter tuning [97].

Precision and recall are two key metrics for measuring the performance of a retrieval system, their equations can be seen below. Precision calculates the proportion of correct instances retrieved among all retrieved instances, while recall calculates the proportion of correct instances retrieved among all correct instances. Accuracy, on the other hand, is the proportion of true instances retrieved, both positive and negative, among all instances retrieved. It is a weighted arithmetic mean of precision and inverse precision [97].

$$\text{Precision : } P = \frac{tp}{tp+fp} \quad (3.13)$$

$$\text{Recall : } R = \frac{tp}{tp+fn} \quad (3.14)$$

The F-score is a measure of a test's accuracy and is defined as the weighted average of precision and recall, with the weighting determined by the parameter β . Specifically, the F1-score represents the harmonic mean between precision and recall, and it is often referred to simply as the F-score or F-measure. Different variations of the F1-score exist, depending on the weights assigned to precision and recall [97].

$$F\text{-score : } F_{\beta} = (1 + \beta^2) * \frac{P * R}{\beta^2 * P + R} \quad (3.15)$$

When β is set to 1, the standard F-score is calculated, as shown in equation (3.16).

$$F\text{-score : } F_1 = F = 2 * \frac{P * R}{P + R} \quad (3.16)$$

Accuracy is another metric that measures the proportion of correctly retrieved instances, including both true positives and true negatives, among all instances retrieved. It is calculated as a weighted arithmetic mean of precision and its inverse. High accuracy can indicate that a system performs well overall, but low precision suggests that the results may be somewhat scattered. This is akin to hitting a target where both high accuracy and high precision are desirable [97], as illustrated in equation (3.17).

$$\text{Accuracy : } A = \frac{tp+tn}{tp+tn+fp+fn} \quad (3.17)$$

3.7.6.2 Area under the curve

In binary classification, the Area Under the ROC Curve (AUC) is frequently utilized to assess classifier performance. It indicates the likelihood that a randomly selected positive instance will receive a higher classification score than a randomly selected negative instance. AUC is a valuable metric for evaluating classifiers [98].

The ROC curve is a graphical method used to evaluate the performance of a binary diagnostic classification method. It helps determine the optimal cut-off value for classifying test results into distinct categories, such as the presence or absence of a disease. Originally developed for signal detection during World War II, the ROC curve has since been widely used in medicine and other fields [99].

The ROC curve offers several advantages, including a comprehensive visualization of test performance, independence from disease prevalence, and the ability to display sensitivity and specificity for each cut-off value. However, it has limitations, such as not directly showing the optimal cut-off value or the number of samples, and its appearance may be jagged with smaller sample sizes [99].

AUC is a widely used metric to measure the accuracy of diagnostic tests. The AUC represents the ability of a test to distinguish between positive and negative cases. A perfect test would have an AUC of 1.0, indicating perfect discrimination. In contrast, a test with an AUC of 0.5 is no better than random chance (like flipping a coin). For a test to be considered acceptable, the AUC is generally expected to be greater than 0.8 [99].

When comparing the performance of two or more tests, the one with the larger AUC is considered to have better diagnostic performance. The AUC is often reported with a 95% Confidence Interval (CI) to account for the variability in the data. For a test to be statistically significant, the lower limit of the 95% CI of the AUC must be greater than 0.5, as shown in the table below [99].

Table 3.1: AUC Values and Their Interpretation.

Area under the curve (AUC)	Interpretation
$0.9 \leq \text{AUC}$	Excellent
$0.8 \leq \text{AUC} < 0.9$	Good
$0.7 \leq \text{AUC} < 0.8$	Fair
$0.6 \leq \text{AUC} < 0.7$	Poor
$0.5 \leq \text{AUC} < 0.6$	Fail

The CI of the AUC can be estimated using parametric or nonparametric methods. The parametric approach assumes a normal distribution of the data and uses methods like the binormal method. Nonparametric approaches, like the rank-sum test, estimate the AUC of the empirical ROC curve and its variance without assuming a specific distribution. Nonparametric estimates may underestimate the AUC for discrete data, but the bias is generally negligible for large sample sizes. When comparing AUCs, a Z-statistic can be calculated to determine if the difference is statistically significant as shown below in equation (3.18) [99].

$$Z = (A_1 - A_2) / \sqrt{\text{Var}(\text{AUC})} \quad (3.18)$$

In summary, the ROC curve is a statistical tool used to identify the optimal diagnostic method and the most effective cut-off value for optimal diagnostic performance. It is crucial to select the best diagnostic test method and cut-off value using the correct methodology [99].

3.8 OPTIMIZATION AND NATURE INSPIRED OPTIMIZATION ALGORITHM

Optimization in the realms of machine learning and artificial intelligence revolves around identifying the optimal solution from a range of viable choices. Typically, the goal is to either minimize or maximize an objective function, which serves as a measure of how well the solution satisfies specific criteria. In machine learning, optimization is essential for refining models by tweaking their parameters to minimize a loss function. This function evaluates how effectively the model performs on training data, and the optimization algorithm calculates the parameter values that minimize this loss [100].

A variety of optimization algorithms are available, each with its own strengths and weaknesses. Gradient descent, stochastic gradient descent, and Adam are some of the most widely used optimization algorithms in machine learning. Overall, optimization is crucial in machine learning as it enables models to learn from data and enhance their performance over time [100].

Nature-inspired optimization algorithms, like those leveraging swarm intelligence, have become valuable for solving intricate, nonlinear optimization challenges. These algorithms mimic natural processes and are adept at finding global optimization solutions, which sets them apart from traditional gradient-based methods that often converge to local optima [101].

Genetic algorithms are one such example of these algorithms. Known for their simplicity, adaptability, and efficiency, these algorithms find application across optimization, computational intelligence, data mining, and machine learning fields. Over the past few decades, the field of nature-inspired optimization has progressed significantly, introducing new algorithms, variations, and applications [101].

3.8.1 Manta Ray Foraging Optimization Algorithm

Manta rays are fascinating creatures, despite their intimidating appearance. They are among the largest marine animals known. Their bodies are flat and wide, and they have a pair of pectoral fins that allow them to swim gracefully, resembling the flight of birds. Additionally, they possess a pair of cephalic lobes that extend in front of their large, terminal mouths [102].

A new bio-inspired optimization technique called the Manta Ray Foraging Optimization (MRFO) algorithm is introduced by Zhao et al. (2020). The algorithm is designed to provide an innovative approach for solving real-world engineering problems. Inspired by the intelligent behaviors of manta rays, MRFO mimics three distinct foraging strategies observed in manta rays: chain foraging, cyclone foraging, and somersault foraging [102].

These strategies are integrated into an efficient optimization framework for solving various optimization problems. The performance of MRFO is evaluated by comparing it with other state-of-the-art optimizers using benchmark optimization functions and eight real-world

engineering design cases. The results of the comparisons on the benchmark functions demonstrate that MRFO outperforms its competitors by a significant margin [102].

The core processes of MRFO are explained, which entail simulating three foraging behaviors seen in manta rays: chain foraging, cyclone foraging, and somersault foraging. In line with other swarm-based metaheuristic algorithms, MRFO starts by randomly generating initial populations within the search space. These populations are then refined using the three aforementioned strategies. The mathematical representations for these strategies are detailed below [103]:

Chain foraging in manta rays involves forming a line by linking their heads and tails. MRFO views the best solution as a higher concentration of plankton, their target food. While the first individual moves directly toward the food, the others not only move toward the food but also toward those ahead of them in the chain [103]. The mathematical model for chain foraging is outlined below.

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_i^t + \mathbf{r}_1 \cdot (\mathbf{x}_{\text{best}}^t - \mathbf{x}_i^t) \\ + \alpha \cdot (\mathbf{x}_{\text{best}}^t - \mathbf{x}_i^t), i = 1 \\ \mathbf{x}_i^t + \mathbf{r}_2 \cdot (\mathbf{x}_{i-1}^t - \mathbf{x}_i^t) \\ + \alpha \cdot (\mathbf{x}_{\text{best}}^t - \mathbf{x}_i^t), i = 2, 3, \dots NP \end{cases} \quad (3.19)$$

$$\alpha = 2 \cdot \mathbf{r}_3 \cdot \sqrt{|\log(\mathbf{r}_4)|} \quad (3.20)$$

In this context, $\mathbf{x}_i^t$ represents the position of the individual i^{th} at generation t . The vectors $\mathbf{r}_i \in [0,1], i = 1,2,3,4$ are randomly distributed. $\mathbf{x}_{\text{best}}^t$ denotes the plankton with the highest concentration, which is the optimal individual. NP refers to the number of populations, and α is a weight coefficient [103].

Cyclone foraging in manta rays involves forming long foraging chains and spiraling towards food found in deep water. This behavior resembles the spiral movement in the Whale Optimization Algorithm (WOA). However, in cyclone foraging, manta rays also follow the individuals in front of them as they spiral towards the food source [103]. The mathematical model for cyclone foraging is expressed by the following equation:

$$\begin{aligned}
x_i^{t+1} = & \begin{cases} x_{\text{best}}^t + r_5 \cdot (x_{\text{best}}^t - x_i^t) \\ + \beta \cdot (x_{\text{best}}^t - x_i^t), i = 1 \\ x_{\text{best}}^t + r_6 \cdot (x_{i-1}^t - x_i^t) \\ + \beta \cdot (x_{\text{best}}^t - x_i^t), i = 2, 3, \dots, NP \end{cases}
\end{aligned} \quad (3.21)$$

$$\beta = 2 \cdot \exp(r_7 \cdot (iter_{\max} - iter + 1) / iter_{\max}) \cdot \sin(2\pi r_7) \quad (3.22)$$

where $r_i \in [0,1]$, $i = 5,6$ represents uniformly distributed random vectors. β is a weight coefficient. $r_i \in [0,1]$ is a uniformly distributed random number. $iter_{\max}$ and $iter$ denote the maximum number of iterations and the current number of iterations, respectively.

Equation (3.21) uses food as a reference point for spiral foraging, aiding in the thorough exploration of the space near food. To broaden the search range, a randomly generated location in the search space is used as a reference for spiral foraging, enabling all individuals to explore areas distant from their current best position. This random spiral foraging mechanism emphasizes exploration, facilitating a comprehensive global search [103]. The mathematical model is detailed below:

$$x_{\text{rand}} = lb + r_8 \cdot (ub - lb) \quad (3.23)$$

$$\begin{aligned}
x_i^{t+1} = & \begin{cases} x_{\text{rand}} + r_9 \cdot (x_{\text{best}}^t - x_i^t) \\ + \beta \cdot (x_{\text{best}}^t - x_i^t), i = 1 \\ x_{\text{rand}} + r_{10} \cdot (x_{i-1}^t - x_i^t) \\ + \beta \cdot (x_{\text{best}}^t - x_i^t), i = 2, 3, \dots, NP \end{cases}
\end{aligned} \quad (3.24)$$

Here, x_{rand} represents a randomly generated position within the search space. $r_i \in [0,1]$, $i = 8,9,10$ are uniformly distributed random vectors. ub and lb denote the upper and lower bounds of the search space, respectively [103].

Somersault foraging: During this phase, each individual treats the food location as a pivot point, flipping around it to explore new positions [103]. This behavior can be mathematically described as follows:

$$x_i^{t+1} = x_i^t + S \cdot (r_{11} \cdot x_{\text{best}}^t - r_{12} \cdot x_i^t), i = 1, 2, \dots, NP \quad (3.25)$$

where S is the somersault factor determining the range of the manta rays' somersaults, set to $S = 2 \cdot r_{11}$ and r_{12} are two random numbers in $[0,1]$.

MRFO manages its exploration and exploitation strategies by adjusting the ratio ($iter/iter_{max}$). When $(iter/iter_{max}) < rand$, the algorithm emphasizes exploration, randomly selecting food sources as reference points in the search space. Conversely, when $(iter/iter_{max}) \geq rand$, it emphasizes exploitation, using the optimal individual as a reference point. Furthermore, a random number determines whether chain foraging or spiral foraging is employed, followed by somersault foraging [103].

MRFO have several advantages over other metaheuristic algorithms, to detail few:

- a. **Innovative Foraging Strategies:** as mentioned previously, MRFO emulates the foraging behaviors of manta rays, such as chain foraging, cyclone foraging, and somersault foraging. These unique strategies enable MRFO to efficiently explore the search space, potentially yielding better solutions compared to algorithms with more conventional exploration methods [103].
- b. **Balanced Exploration and Exploitation:** as said above, MRFO maintains a balance between exploration and exploitation by adjusting the parameter ($iter/iter_{max}$). This balance allows MRFO to explore the search space effectively while also exploiting promising regions, achieving a good trade-off between exploring new solutions and exploiting known solutions.
- c. **Adaptability to Various Problem Types:** MRFO is highly versatile and adaptable, making it suitable for different types of optimization problems, including non-linear, non-differentiable, and problems with multiple local minima. This flexibility enables MRFO to address a wide range of real-world engineering challenges [104].
- d. **Superior Performance on Complex Problems:** MRFO has demonstrated superior performance in solving complex optimization problems, especially those with numerous decision variables, complex nonlinear constraints, and objective functions with multiple peaks. This capability makes MRFO particularly well-suited for tackling challenging engineering optimization tasks [105].
- e. **Efficiency and Effectiveness:** MRFO's performance has been compared with other state-of-the-art optimizers on benchmark functions and real-world engineering design cases. The results indicate that MRFO outperforms its competitors in terms of convergence speed and solution quality, highlighting its potential as a highly efficient and effective optimization algorithm [102].

In summary, MRFO's innovative foraging strategies, balanced exploration-exploitation mechanism, adaptability to different problem types, and demonstrated efficiency and effectiveness make it a competitive choice for solving complex optimization problems across various fields.

3.9 TRANSFORMATION FUNCTIONS

Transformation functions are essential tools in optimization, simplifying complex mathematical formulations for easier analysis and solution. Particularly useful for handling non-linear terms, these functions can reduce the computational complexity of optimization problems. By applying transformation functions, models can be reformulated to be more compatible with existing solution techniques, like linear programming or convex optimization [106]. Here are some common types of transformation functions and how they are used:

- a. Linearization Functions: These convert non-linear terms in objectives or constraints into linear forms, making them easier to manage [106].
- b. Piecewise Linear Approximations: Non-linear functions can be approximated by piecewise linear functions, dividing the function into segments and approximating each segment with a linear function, simplifying the problem significantly [106], as demonstrated in Figure 3.8 below.

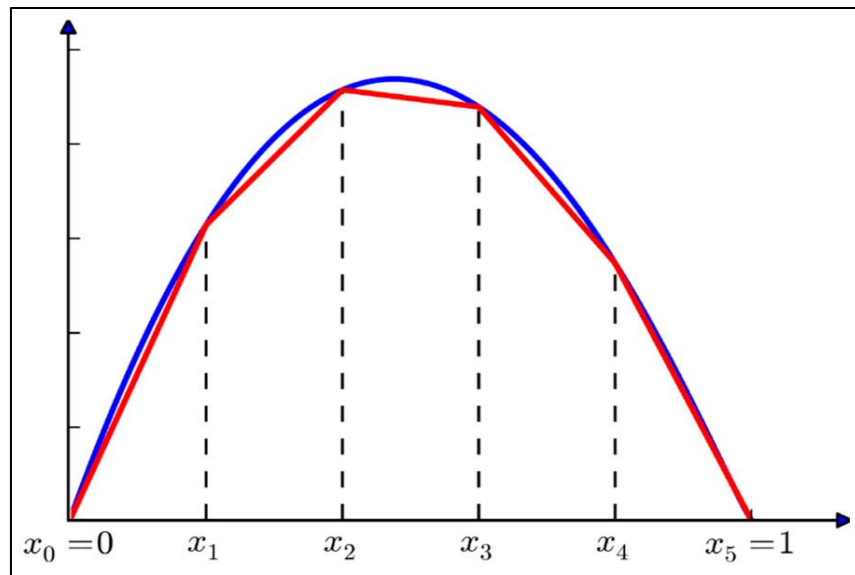


Figure 3.8: A Non-linear Function (blue) and its Approximation Using Piecewise Linear Segments (red).

- c. **Binary Multiplication Functions:** Multiplication involving binary variables can introduce non-linear terms, which can be linearized by introducing additional binary variables and constraints [106].
- d. **Absolute Value Function:** Although non-linear, the absolute value function can be linearized using transformation techniques [106].
- e. **Floor and Ceiling Functions:** These functions round real numbers down (floor) or up (ceiling) to the nearest integer. Transformation functions can represent these in linear form by introducing additional variables and constraints [106].
- f. **Multiple Breakpoint Functions:** Functions with multiple breakpoints, like piecewise linear functions, can be linearized by approximating them with a series of linear segments, each defined over a different interval [106].
- g. Overall, transformation functions simplify complex optimization problems, enabling the use of efficient solution techniques and improving computational efficiency.

3.9.1 S-Shaped and V-Shaped Functions

S-shaped transfer functions utilize variations of sigmoid functions. In these functions, when the velocity is positive, the subsequent position will be either zero or one. If a randomly generated number between 0 and 1 is less than or equal to the velocity, the next position will be one; otherwise, it will be zero. Conversely, if the velocity is negative, the next position will always be zero since the random number is positive [107], as can be seen below in Figure 3.9.

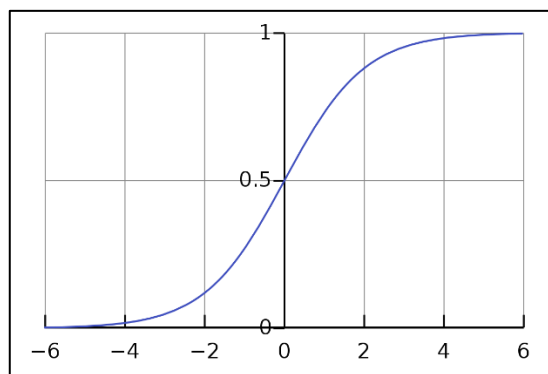


Figure 3.9: The Sigmoid Function's Elongated 'S'-shaped Curve.

V-shaped functions are mathematical expressions that form a shape reminiscent of the letter 'V' when plotted, as can be seen in Figure 3.10 below. These functions usually feature a swift rise or fall in the output as the input varies. While the specific form of a V-shaped

function can differ, it typically includes a sudden change in slope at a specific point, creating the distinctive V-like appearance. These functions are prevalent across different disciplines, like mathematics, economics, and physics, where they can depict phenomena like rapid decline, abrupt transitions, or bifurcations [108].

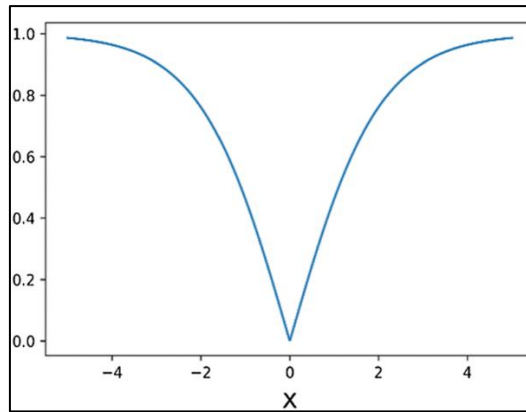


Figure 3.10: V-shaped Transfer Function.

In summary, S-shaped and V-shaped functions are fundamental mathematical concepts with widespread applications across various fields. They are essential tools for understanding and modeling complex systems, playing significant roles in disciplines such as mathematics, economics, biology, and ecology.

4. METHODOLOGY

4.1 INTRODUCTION

The main contribution of this study is to enhance the prediction of Alzheimer's disease (AD) using bio-inspired, wrapper-based gene selection method applied to Alzheimer GSE microarray datasets. The motivation and methodology behind the proposed approach are detailed here. The chapter begins by providing comprehensive description of weakness and contribution made to improve the algorithm exploration and exploitation, known as EMRFOGS. Subsequently, the optimization objective function of EMRFOGS and the solution encoding method using S-Shaped function suitable for transforming discrete values into binary representations are discussed. The details of proposed improvements, Best Rank approach to enhance *Chain* exploration, Sign Random Mutation (SRM) to improve *Cyclone* exploitation, and Hybrid Boundary Function (HBF) to increase convergence, are explained theoretically with examples.

4.2 MOTIVATION

In the light of challenges facing prediction and early diagnosis of Alzheimer disease posed by the curse of dimensionality, advance gene selection method is required. The inherent complexity of high number of genes in GSE Alzheimer datasets necessitate sophisticated gene selection method in order to discover genes relationships, hidden patterns, and explore the large features search space more efficiently. While efficient genes selection methods in general achieves significant reduction of computational burden and mitigate the impact of curse of dimensionality, identifying the optimal subset of genes can increase the classification accuracy of Alzheimer disease by supervised machine learning algorithms. Inspired by the foraging behavior of Manta Rays, the proposed MRFO large-scale optimization algorithm for genes selection can efficiently mitigate the curse of dimensionality and reduce the dataset dimensionality, thereby mitigating overfitting and computational inefficiency faced by supervised learning algorithms. Given the MRFO strong exploration and exploitation, the proposed approach aims to improve the identification of biomarkers, enhance predictive accuracy, and ultimately contribute to the early diagnosis and targeted treatment strategies of Alzheimer's disease. This approach not only addresses the critical need for efficient genes selection and analysis in AD microarrays, but also can be utilized in genes selection of other diseases as well.

4.3 PROPOSED GENE SELECTION MODEL OF ALZHEIMER DISEASE

The process of gene selection using wrapper method based on the MRFO algorithm involves several key steps, see Figure 4.1:

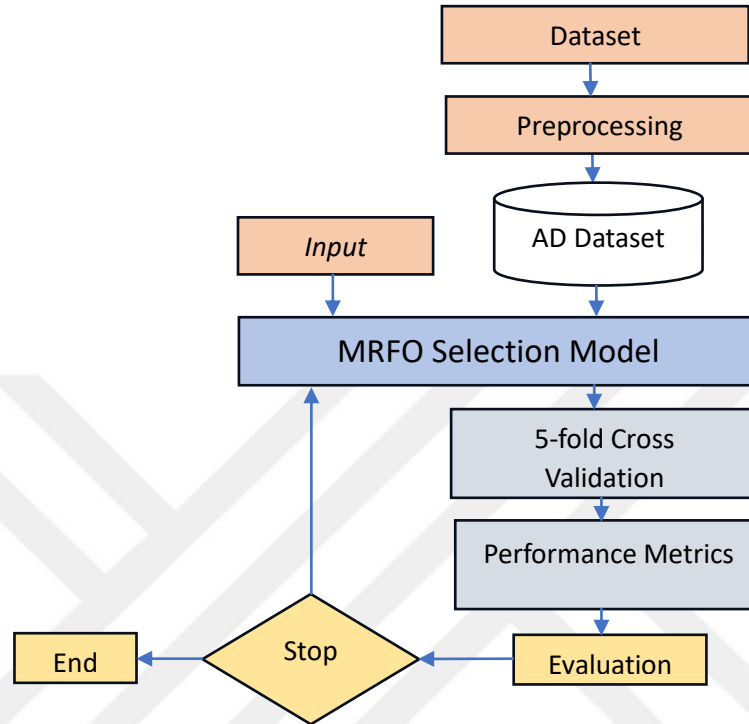


Figure 4.1: Proposed Gene Selection Model.

4.3.1 Data Gathering and Preprocessing

Alzheimer genes microarray datasets are obtained from the National Library of Medicine (NCBI) [112] in csv file format (comma-separated values). Six of the highly referenced and extensively tested datasets were selected from the literatures: GSE1297, GSE5281, GSE33000, GSE63060, GSE63061, and GSE132903, respectively. Detailed description of each dataset will be provided in the later sections. Data processing typically involves – if necessary – values normalization and handling missing data values using various techniques. Upon data inspection using Pandas Library [113], no missing values were found in the datasets.

4.3.2 Selection Model

In this study, wrapper-based genes selection model using Bio-inspired Manta Ray Foraging Optimization (MRFO) algorithm is proposed. MRFO can efficiently explore the high dimension features space and optimally select the subset of genes to enhance the

classification accuracy and prediction of Alzheimer disease. Optimization algorithms, including MRFO, often start with population initialization. Each solution within population is represented as one dimensional vector $\vec{\pi}$ with size equals to problem dimension Dim , where Dim is the number of genes in a given GSE dataset. The population of solutions is iteratively evaluated and performance metrics are calculated at each step. Through optimization iteration, MRFO alternates between exploration and exploitation strategies to identify the local optimal solution, and eventually converges to the global optimal $Global_{best}$.

MRFO algorithm is characterized by its minimal parameters' tuning compared to other optimization algorithms. MRFO primarily requires two inputs parameters: the maximum number of iterations ($MaxIter$) and population size ($PopSize$). The optimization process often continuous until predefined termination condition is met. In the proposed approach, the termination condition is set as $MaxIter$. The output of proposed selection model is $Global_{best}$, final converged solution. The genes identified in the global best $\vec{\pi}$ are then clinically utilized to develop disease diagnosis tool.

4.3.3 Performance Metrics

The set of predefined performance metrics is varied dependent on specific requirements of one optimization problem to another. Commonly used metrics include learning model accuracy, recall, precision, and F1-measure. However, features or genes selection optimization problems utilizes different measures such as mean accuracy of K -fold cross-validation, number of features, and feature selection average weights or scores. In this study, error rate of mean classification accuracy and averaged selected genes are used to assess genes selection. The mean classification accuracy of utilized learning algorithm is calculated based on mean accuracy of 5-fold cross validation. Cross-validation is crucial in features selection for several reasons especially in scenarios where data samples are limited. CV ensures better model performance generalization to unseen data, mitigates overfitting of learning model through training and validation on multiple data splits, and provide an accurate estimation of selected features set performance on different training and testing data splits.

4.3.4 Evaluation

In the context of optimization and machine learning, performance metrics often combined into a single score by what's known as optimization *fitness function*. Such that, the score is used to quantitatively measure how 'fit' is one solution compared to another. In the context of genes selection, fitness score is used to measure how well selected set of genes increases prediction performed of specific disease compared to another set. Performance metrics are weighted and combined into a single score. Weighted sum approach can effectively navigate the trade-offs among different desired performance metrics, given the nature of similar or contradiction properties, weights are assigned to priorities metrics over others to best meet the defined objectives. From reviewing several features selection fitness functions [114][115][116], the following is the formulated fitness function used in the proposed MRFO genes selection approach:

$$fitness = (1 - accuracy_{mean}) + w * \frac{SF}{VF} \quad (4.1)$$

Where $accuracy_{mean}$ is the mean accuracy of 5-fold cross validation, w is features weight, SF is number of selected features, and VF is the number of all features in dataset. MRFO evaluate each $\vec{\pi}$ using the proposed fitness function and iteratively evolve the final $Global_{best}$ through exploration and exploitation.

All of the presented key steps are detailed in the later sections of this chapter.

4.4 SOLUTION REPRESENTATION

Typically, the solution vector $\vec{\pi}$ is composed of an N continuous values that represent an individual solution position among population of solutions for a given optimization problem, where N equals to the Dim of the problem. Dependent on the Dim or the number of variables, solution vectors are evaluated by fitness function and solution values are updated through exploration and exploitation of optimization algorithm. While the structure is typically straightforward, in the context of features selection, different solution structure must be adapted. The vector $\vec{\pi}$ must reflect the gene/feature status. Each continuous value must reflect either "inclusion" or "exclusion" of particular gene from the dataset.

While the solution vector $\vec{\pi}$ values are directly updated by the optimization algorithm to explore potential solutions within search space, a transformation of these values is necessary before the evaluation step. Transformation process refers to the use of binary encoding functions where continuous input values are transformed to binary output values of 0s and 1s. The value of 0 means the exclusion of gene while value of 1 represents the inclusion of specific gene during metrics evaluation. Figure 4.2 depicts the two types of transformation functions, S-Shaped and V-Shaped [117]. Each type consisted of a family of four transformation functions illustrated in Table 4.1 [118][119].

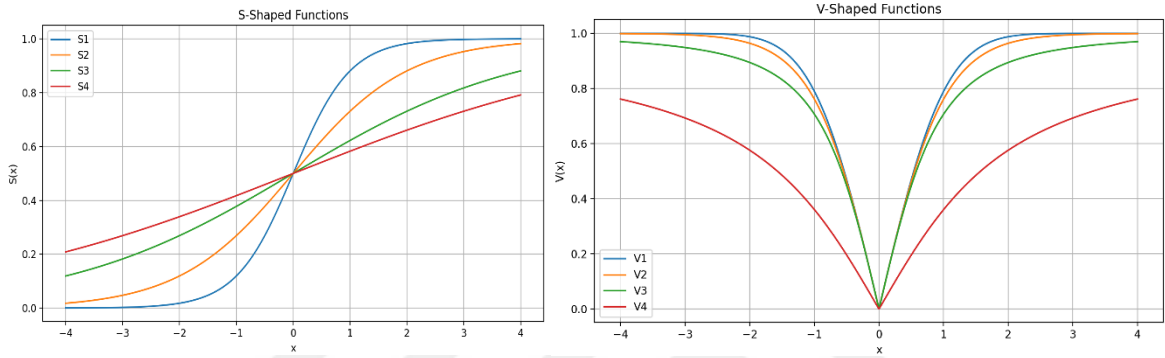


Figure 4.2: S-Shaped and V-Shaped Transformation Functions [117].

Table 4.1: S-Shaped and V-Shaped Transformation Functions.

Type	Name	Function
S-Shaped	S1	$S(x) = \frac{1}{1 + e^{-2x}}$
	S2	$S(x) = \frac{1}{1 + e^{-x}}$
	S3	$S(x) = \frac{1}{1 + e^{\left(\frac{-x}{2}\right)}}$
	S4	$S(x) = \frac{1}{1 + e^{\left(\frac{-x}{3}\right)}}$
V-Shaped	V1	$V(x) = \left \operatorname{erf} \left(\frac{\sqrt{x}}{2} x \right) \right $
	V2	$V(x) = \left \tanh (x) \right $
	V3	$V(x) = \left \frac{x}{\sqrt{1+x^2}} \right $
	V4	$V(x) = \left \frac{2}{\pi} \arctan \left(\frac{2}{\pi} x \right) \right $

The choice between either function is dependent on the specific requirements of the optimization problem at hand and the characteristics of the solution. S-shaped transformation gives gradual and smooth transition from status 1 to 0 and vice versa critical in the context where sensitivity to changes in input is necessary. On the other hand, V-Shaped gives rather sharp transformation decision necessary in scenarios of decisive response to input changes. In this study, S-Shaped transformation functions are considered to encode the solution $\vec{\pi}$ values to binary form. The sigmoid smooth S-Shaped functions reduces selection pressures thereby improve exploration of the search space.

In this study, the steps of solution encoding are summarized as follow:

For every solution vector $\vec{\pi}$ in the population, the continuous values are encoded using one of S-Shaped functions before evaluation.

The binary-encoded vector $\vec{\pi}$ is then used to determine the active set of genes using the index of 1s in binary vector $\vec{\pi}$ itself. New dataset $\bar{D}$ with dimension of $N \times \bar{M}$ is created where N denotes the number of samples and $\bar{M}$ is the number of selected features, satisfying $\bar{M} \leq M$ and M is the original number of features, see Figure 4.3. In S-Shaped transformation functions, binary encoding of continuous values to binary is performed according to the following:

$$Encode(x_i) = \begin{cases} 1 & \text{if } rand \leq S(x_i) \\ 0 & \text{else} \end{cases} \quad (4.2)$$

Where *rand* is a random number generated between [0,1]. For every value v_i in solution vector $\vec{\pi}$, transformed value is calculated according to selected transformation function. In the example above, the S-Shaped transformation is applied then if the value of a random generated number is less or equal to, an encoded value of 1 is given. This means that the gene at index i is selected. Figure 4.3 illustrates a complete overview of solution encoding used in MRFO.

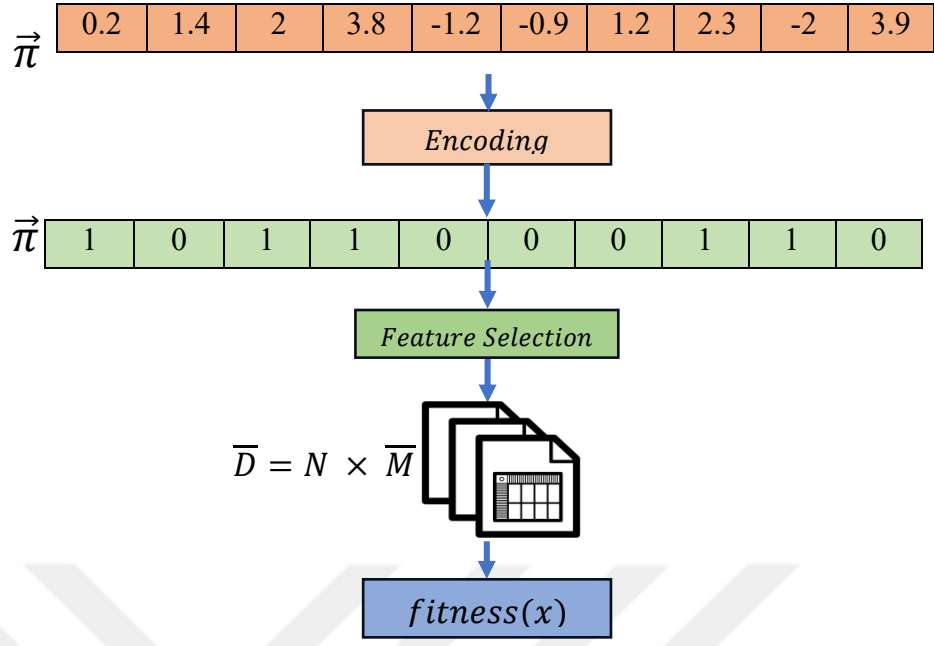


Figure 4.3: Solution Encoding Steps in the Proposed Algorithm.

The newly formed $\bar{D}$ is then divided into 5 folds of training and testing by 5-fold cross validation.

Performance metrics are measured.

Fitness value is returned after applying optimization fitness function to performance metrics. This value is then compared and used to determine the effectiveness of solution vector $\vec{\pi}$.

4.5 ALZHEIMER MICROARRAY GSE DATASETS

Alzheimer genes microarray datasets are obtained from the National Library of Medicine (NCBI) in csv file format [112]. The details and the dimension of each GSE dataset is depicted as follow:

Table 4.2: Alzheimer GSE Dataset Dimension.

DATASET NAME	DIMENSION	
	Samples	Features
GSE1297	31	16379
GSE5281	161	54674
GSE33000	624	39279
GSE44768	329	39279
GSE44770	388	39279
GSE132903	195	42178

- a. GSE1297: The dataset in question is downloaded from the Gene Expression Omnibus and is specifically the GSE1297 datasets. Made publicly available by Eric M Blalock in 2004, it contains AD and CTL from 22 American patients with AD and 9 CTLs; given that most patients develop AD over the age of 60, the number of old patients is relatively high. In total, there are 31 samples and 16,379 genes; each sample includes such features as Age, Gender, Level, Cell type, and Developmental stage. The dataset is structured in such a way that the first row of the table represents identification of each gene column, with the next rows representing the identification card (AD or CTL) up to gene expression levels [120].
- b. GSE5281: The dataset GSE5281 was obtained from the Gene Expression Omnibus (GEO) and published in 2006. This dataset consists of data obtained from 87 Alzheimer's disease patients and 74 controls, totaling 161 samples and 54,674 genes. Each sample has specific details, such as age, gender, cell type, ethnicity, developmental stage, genetic differences, and more. The dataset is organized in such a manner that the first row serves as column identifiers, the subsequent columns denote patient status either Alzheimer's or CTL and the values in these columns represent gene expressions [121].
- c. GSE33000: The GSE33000 dataset from the Gene Expression Omnibus (GEO), made public in 2011. The dataset includes data from 310 Alzheimer's disease (AD) patients, 157 with mild cognitive impairment (MCIs), and 157 controls (CTLs), comprising 624 samples and 39,279 genes. Each sample contains detailed demographic information like age and gender. The dataset's structure features a first row of gene identifiers, while subsequent rows label the patient's status (AD, MCIs, or CTL) and display gene expression levels [122].
- d. GSE44768: The GSE44768 dataset examines molecular interactions in the brain tissues of individuals with late-onset Alzheimer's disease (LOAD) in comparison to those without dementia. The process involves constructing a gene regulation network to discover crucial molecular alterations in the pathogenesis of Alzheimer's disease. Specifically, it emphasizes the significance of an immune/microglia module. Information was gathered from the dorsolateral prefrontal cortex, visual cortex, and cerebellum, using a specialized 44K array. The data was then modified to account for several characteristics such as age, sex, and postmortem interval and was made

public in 2013. The dataset includes a total of 329 samples and covering 39,279 genes [123].

- e. GSE44770: The dataset GSE44770 examines the gene expression patterns in human brain tissues from individuals with late-onset Alzheimer's disease (LOAD) and those without dementia. This research used a specially designed Agilent 44K array to examine alterations in molecular interactions and cellular pathways linked to late-onset Alzheimer's disease (LOAD). The specimens were obtained from the dorsolateral prefrontal cortex, visual cortex, and cerebellum. The research revealed distinct molecular interaction patterns altered by late-onset Alzheimer's disease (LOAD) and used a network-based methodology to emphasize the importance of the immune/microglia module in LOAD pathogenesis. [124]. Dataset is consisted of 388 samples and 39,279 genes.
- f. GSE132903: The GEO dataset GSE132903 specifically examines Alzheimer's disease and contains RNA-seq data obtained from the entorhinal cortex and superior frontal gyrus regions of human brains. These areas play a crucial role in memory and cognitive functioning, which are often affected in individuals with Alzheimer's disease. The purpose of this dataset is to provide a deeper understanding of the alterations in gene expression that are linked to the illness, with the potential to discover new biomarkers or targets for therapy. The GSE132903 dataset was made public in 2019 with 42,178 genes across 195 samples. These samples include data from 97 Alzheimer's disease (AD) patients and 98 controls (CTLs), with individual characteristics such as age and gender documented [125].

4.6 MRFO OPTIMIZATION ALGORITHM

Manta Ray optimization algorithm is bio-inspired large-scale and parameters-free algorithm. MRFO, compared to other optimization algorithms, offers several key advantages. First, its adaptive foraging techniques that is inspired by Manta Ray foraging behavior. MRFO foraging techniques, *Chain*, *Cyclone*, and *Somersault*, enhance the exploitation and effective exploration of search space to locate the global optimal solution. Second, due to several reasons such as convergence speed and efficient computational complexity, MRFO algorithm is suitable for large scale optimization problems. With

solution encoding and fitness function detailed in previous sections, the flowchart below, Figure 4.4, describes the proposed MRFO gene selection model.

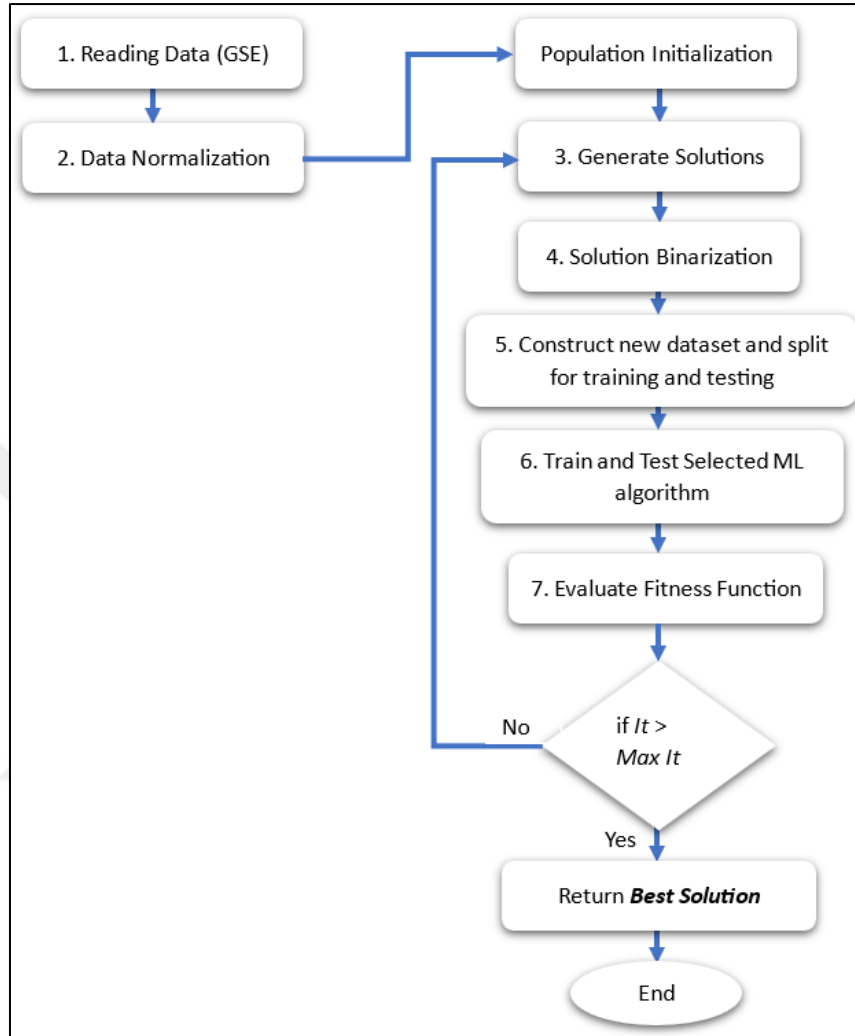


Figure 4.4: Flowchart of MRFO Gene Selection Algorithm.

To from preliminary assessment of MRFO gene selection optimization performance, MRFO was extensively tested using the selected Gene Expression Omnibus datasets. This analysis is critical to measure efficiency of MRFO according to small and large features space. The algorithm parameters such as population size and the number of optimization iterations were systematically varied and the results were investigated. The results highlighted several challenges, first, MRFO performance was acceptable in small datasets. However, it demonstrated weak exploration of genes space in large datasets. During rigorous tests, MRFO retained two-thirds of the total number of features in different large datasets. In this context, large or high-dimensional dataset means high features number.

Furthermore, neither the increase of iterations nor population size improved the performance. This indicates weak balance between exploration and exploitation and not correlated to the number of solutions.

It was also hypothesized that suboptimal performance of MRFO could be attributed to the sensitivity of machine learning algorithm utilized in evaluating the solution vector $\vec{\pi}$. However, testing different learning algorithms proved MRFO weak balanced exploration and exploitation strategies in genes selection of GSEs dataset. Given the observed limitations, three main enhancements are proposed in the following section to improve MRFO with emphasis on increasing the convergence speed, improving exploration of features space, and ensuring optimal prediction accuracy of Alzheimer disease. The following pseudocode and subsequent sections detail the proposed enhancements of MRFO related to this study.

```

WHILE stop criterion is not satisfied do
FOR i=1 TO N DO
IF rand < 0.5 THEN // Chain foraging ← Best Rank(1)
    IF  $t/T_{max} < rand$  THEN
         $x_{rand} = x_l + rand \cdot (x_u - x_l)$ 
        
$$x_i(t+1) = \begin{cases} x_{rand} + r \cdot (x_{rand} - x_i(t)) + \beta \cdot (x_{rand} - x_i(t)) \\ x_{rand} + r \cdot (x_{i-1}(t) - x_i(t)) + \beta \cdot (x_{rand} - x_i(t)) \end{cases}$$

    ELSE
        
$$x_i(t+1) = \begin{cases} x_{rand} + r \cdot (x_{best} - x_i(t)) + \beta \cdot (x_{best} - x_i(t)) \\ x_{rand} + r \cdot (x_{i-1}(t) - x_i(t)) + \beta \cdot (x_{best} - x_i(t)) \end{cases}$$

    END IF.
ELSE // Cyclone foraging
    
$$x_i(t+1) = \begin{cases} x_{rand} + r \cdot (x_{best} - x_i(t)) + \alpha \cdot (x_{best} - x_i(t)) \\ x_{rand} + r \cdot (x_{i-1}(t) - x_i(t)) + \alpha \cdot (x_{best} - x_i(t)) \end{cases}$$

← Sign Random Mutation(2)
END IF.

Compute the fitness of each individual  $f(x_i(t+1))$ . IF  $f(x_i(t+1)) < f(x_{best})$ 
THEN  $x_{best} = x_i(t+1)$ 
← Hybrid Boundary Function(3)

FOR i=1 TO N DO // Somersault foraging
    
$$x_i(t+1) = x_i(t) + S \cdot (r_2 \cdot x_{best} - r_3 \cdot x_i(t))$$

    Compute the fitness of each individual  $f(x_i(t+1))$ . IF  $f(x_i(t+1)) < f(x_{best})$ 
    THEN  $x_{best} = x_i(t+1)$ 
END FOR.
END WHILE.
Return the best solution found so far  $x_{best}$ 

```

Figure 4.5: MRFO Pseudocode and Proposed Enhancements.

4.6.1 Best Rank

In chain foraging, Manta Rays exhibit a behavior where they observe the position of plankton and swim towards it. The attractiveness of a position is determined by the concentration of plankton present. While the optimal solution remains unknown, MRFO considers the position with high plankton concentration as the best solution discovered thus far. MRFO retained approximately between 1/3 to 2/3 of the total number of features during optimization iterations. MRFO kept the same ratio even with increase in optimization iterations. This indicates that *Chian* foraging failure to explore new space areas and remained in local minima.

Due to solution hierarchy in population-based optimization algorithms, the top solution vector $\vec{\pi}_0$ in the population has the strongest effect on the remainder of the population. Such influence progressively diminishes for the subsequent solutions, such as the second, third, and so forth. During *Chain* foraging, see Figure 4.6, Manta rays align themselves head-to-tail, forming a foraging chain. During each iteration, positions of each solution $\vec{\pi}$ is updated based on the current best solution and the solution ahead of it. Such that, solution of $\vec{\pi}_i(t)$ is affect by solution position at $\vec{\pi}_{i-1}(t)$.

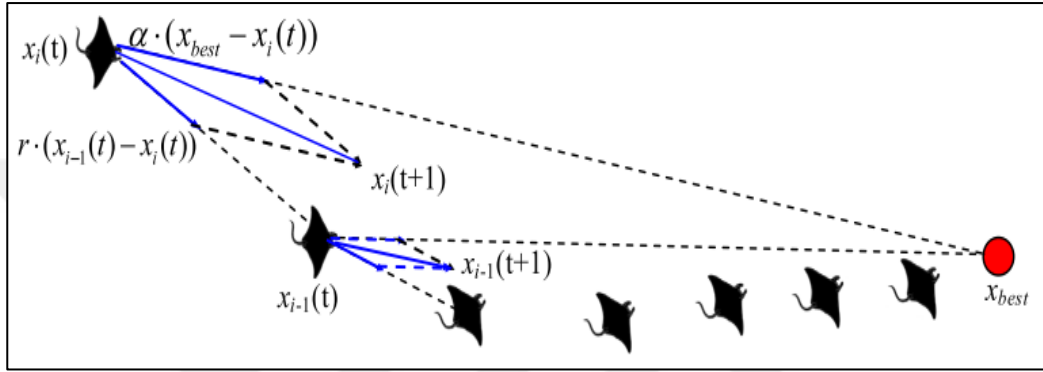


Figure 4.6: MRFO Chain Foraging [102].

Solutions Rank approach is the process of ranking solutions from best to worst based on their fitness value. Our proposed improve exploration of MRFO was inspired by Andi Tang et al. in [126], where he utilized a pool of elite top-three ranked solutions to improve MRFO *Chain* and *Cyclone* foraging exploitation. However, such approach increases the MRFO time complexity by $\sim O(3N - 6)$, where N is the population size. Increasing time complexity of features selection optimization algorithm is critical especially within the context of Alzheimer high dimensional microarrays datasets. The combinatorial space of genes subsets expands exponentially which significantly increase the computational cost. Thereby limiting the feasibility of solution implementation.

In our method, we propose swapping the first solution of the population, i.e. population *Head*, with the highest ranked solution (local best) in the population at each optimization iteration such that $Local_{best} \neq Global_{best}$. With time complexity increase of just $O(N)$, swapping $Local_{best}$ ensures guiding the rest of the population to better search area and improves the discovery rate of better solutions.

4.6.2 Sign Random Mutation

The second proposed enhancement is related to MRFO *Cyclone* phase exploitation, see Figure 4.7. When a group of Manta Rays discovers a cluster of plankton, a long spiral foraging chain is formed to navigate towards the food. With each individual progressing in spiral trajectory towards the food source, each individual in the swarm conducts exploration relative to its position and food position. Thereby, enhancing the exploitation of the region containing the best solution identified in the cyclone foraging strategy. This approach not only increases exploitation but also significantly enhances exploration.

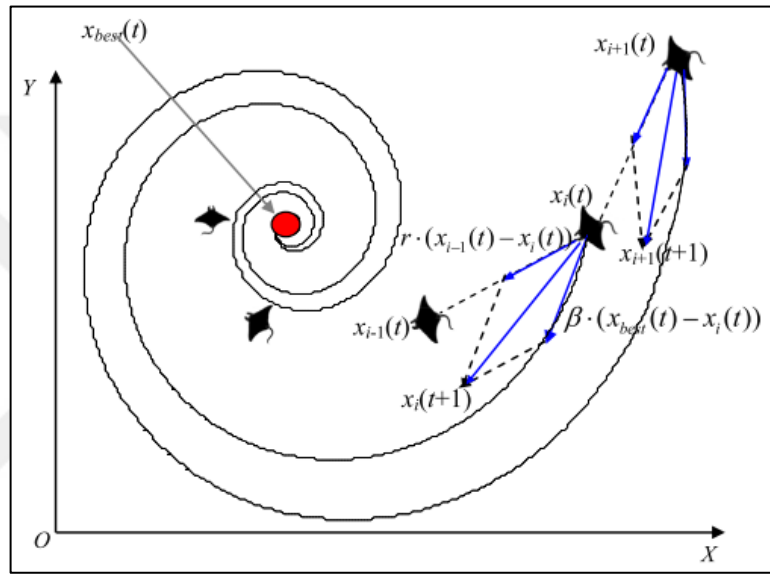


Figure 4.7: MRFO Cyclone Foraging [102].

Slow *Cyclone* exploit was evident in tests, MRFO retained the same number of genes throughout optimization cycle and with only marginal improvement in accuracy. Exploration and exploitation are correlated in the sense that weak exploration of high-quality search space areas consequently effect the exploitation of refining and improving the current solution. Testing V-Shaped transformation functions such as V1 (see Table 4.2), reduced the number of selected genes significantly, however, reduced low prediction accuracy was measured below 60% in some datasets. This decline in accuracy indicates that the selection was influenced by the sharp decisive cutoff transition of V-Shaped functions rather than the quality of selected genes. Hence, aggressive reduction in genes set diminishes the effect of other performance metrics in optimization fitness function thereby doesn't allow for comprehensive evaluation. Weak genes patterns and relationships will contribute to weak classification accuracy.

Improved exploration using *Best Rank* approach can contribute to exploitation outcome. However, slow exploits and strong exploration can cause optimization imbalance and reduces convergence speed and overall optimization performance. To improve *Cyclone* exploit speed, novel Sign Random Mutation is proposed. During *Cyclone* phases, *SignRandomMutation* is immediately applied to newly generated solution. The proposed method is comprised of three main steps:

- All active features/Genes from $Global_{best}$ solution is grouped in an array called *features*.
- Then, for a length of K , a set of uniquely sampled elements *subfeatures* is chosen. K is calculated as follow:
- $$K = \text{int}\left(\frac{\text{length}(\text{features}_{Global_{best}})}{1+r}\right)$$

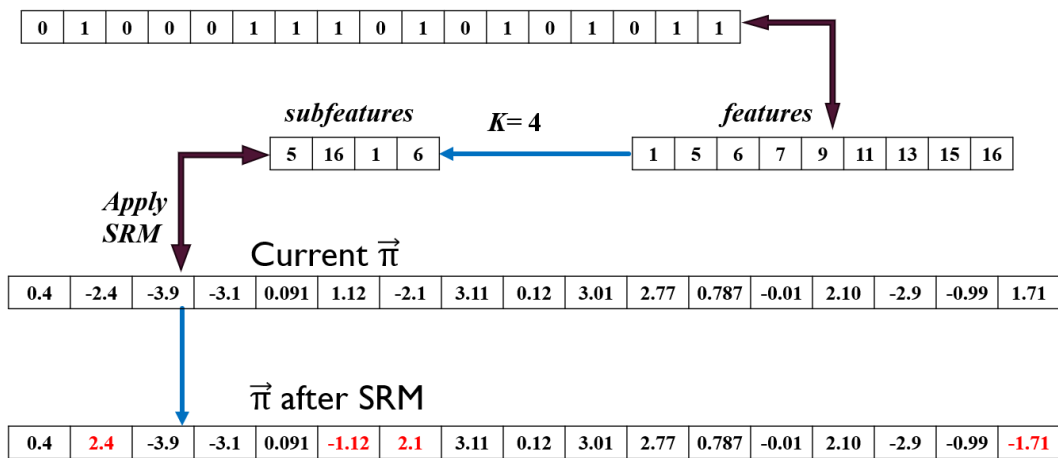
Where r is a random number generated between $[0, 1]$.

- Then the following is applied to solution vector $\vec{\pi}$:

for each element e in *subfeatures* list:

$$\vec{\pi}[e] = \vec{\pi}[e] * -1$$

To further understand SRM approach, let's consider the following example:



when values in solution $\vec{\pi}$ indexed at [5, 16, 1, 6] changed from [1.12, 1.71, -2.4, -2.1] to [-1.12, -1.71, 2.4, 2.1] respectively, this is translated to an increase or a decrease of genes activation probability. For example, the probability of generating a random value r between [0,1] such that $r \leq S(-2.4)$ is significantly lower than that when $r \leq S(2.4)$ after applying RSM. This means that the second gene is more likely to be activated in the new derived solution. Conversely, the probability was much higher for the 6th gene before applying SRM which means that this gene is more likely to be deactivated. This effectively increased the efficiency of *Cyclone* exploitation evident in test results of MRFO. Different solutions with similar/nearly-similar number of active genes were examined during different optimization iterations indicating that MRFO is attempting to exploit the current best solution.

4.6.3 Hybrid Boundary Function

The values of solution vector $\vec{\pi}$ generated by optimization algorithm must always be within predefined boundaries, as such, these boundaries set the search space limits to ensure the validity of the developed solution. All values v in $\vec{\pi}$ must satisfy $LowerBound \leq v \leq UpperBound$. However, when value exceed those limits, the value is often *clamped* to the nearest bound or *replaced* with new random value [127][128].

Both approaches slow the convergence of algorithm especially in features selection. New *clamped* value means that the feature is either *permanently* active or not active according to S-shaped and V-shaped transformations methods. For example, all the values greater than 4 will always be clamped to 4, and $S(4) \cong 1$. Thus, the feature/gene index at v_i will always be active.

Second, reinitialization to completely new value changes the exploration direction and cause search interruption renders it difficult for the algorithm to fully explore new search areas. Based on this analysis, we proposed a hybrid boundary function for MRFO that is based on optimization *coefficient* to randomly switch between either of the following two methods:

- a. *Clamping* to the nearest boundary value (often used in optimization algorithms).
- b. *Reinitialization* using genetic crossover between v_i in $\vec{\pi}$ and v_i in $Global_{best}$.

Optimization Coefficient can be calculated as follow:

$$Coefficient = \frac{Current\ Iteration}{Max\ Iteration} \quad (4.3)$$

It's worth mentioning that several variations of HBF were tested. Changing *clamping* method to *sign clamping*, for example, didn't improve the performance efficiency. EMRFOGS performance kept falling where the changes in number of active genes was chaotic and solution accuracy didn't improve. While changing *reinitialization* method to *reinitialization* with *sign inverse* instead of genetic crossover with $Global_{best}$ achieved marginal improvements during the early stages of optimization then dramatically increased the number of active genes with no improvements in classification accuracy. This behavior was the result of the contradictory behavior of HBF and *SignRandomMutation* methods causing chaotic search interruptions and reducing solutions efficiency.

The following is the pseudocode of proposed HBF.

Algorithm: Hybrid Boundary Function

Input: solution vector $\vec{\pi}$

Procedure:

```

    If coefficient < random  $r$ : # Clamping
        For each value  $v_i$  in  $\vec{\pi}$ :
            if  $v_i$  is out of boundary:  $v_i =$  nearest bound.
                such that if  $v_i \leq LowerBound$ , then  $v_i = LowerBound$ .
                and if  $v_i \geq UpperBound$ ,  $v_i = UpperBound$ .
        End For
    Else: # Crossover
        For each value  $v_i$  in  $\vec{\pi}$ :
            if  $v_i$  is out of boundary:  $v_i = v_i^{Global_{best}}$ 
        End For
    Return  $\vec{\pi}_{new}$ 

```

5. TESTS AND RESULTS

5.1 INTRODUCTION

To evaluate the optimization performance of proposed EMRFOGS (Enhanced Manta Ray Foraging Optimizer for Genes Selection), six of the most cited Alzheimer datasets in the literatures are tested thoroughly. Datasets demographics range from 16379 to 54674 thousand in features dimension space and from 31 to 624 in samples size. The selected datasets can effectively demonstrate EMRFOGS optimization performance across varied scenarios. Performance evaluation is based on several metrics and are illustrated for each dataset based on four machine learning algorithms. Empirical results are examined thoroughly assisted by convergence graphs. The following sections details the experimental settings and results discussion of the proposed EMRFOGS.

5.2 EXPERIMENTAL SETTINGS

The experimental setup for evaluating Enhanced Manta Ray Foraging Optimizer for Genes Selection (EMRFOGS) of Alzheimer disease genes selection for microarray dataset was conducted using Python 3.9 and PyCharm Community Edition. The experiments were carried on a machine equipped with 2.4 GHz Inter Core i7 processor and 16GB of system memory. All the dataset are organized in csv file format after preprocessing, obtained from NCBI. EMRFOGS genes selection algorithm parameters configuration includes setting population size to 50 with maximum optimization iteration of 100. To thoroughly assess the sensitivity of EMRFOGS for genes selection, four different machine learning algorithms were tested and evaluated individually across all GSE dataset. The experiment included four of the most cited classifiers in Alzheimer genes selection literatures: Support Vector Machine (SVM), Random Forest (RF), XGBoost Tree Classifier (XGBoost), and Naive Bayesian (NB). This assessment is to validate the consistency of EMRFOGS and generalization of findings across different learning models. Hence, to show that the genes selection performance is not tailored to specific dataset nor to learning model. Cross-validation of 5 folds is also implemented in the EMRFOGS tests. Due to small number of data samples, cross-validation ensures more accurate test results. Dividing data into 5×5 subsets, 4×1 for training and testing reduces model's bias and overfitting, and provide more reliable evaluation of performance metrics. In other words, model's evaluation is

less depended on the particular way of how data is divided. Last, evaluation of each learning model is based on accuracy, F1, precision, recall, and AUC curve.

5.3 RESULTS AND DISCUSSION

Empirical tests of selected Alzheimer datasets are detailed as follow:

5.3.1 GSE1297

The Tables 5.1 and 5.2 showcase the performance metrics of four machine learning models Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and XGBoost both before and after applying EMRFOGS genes selection. NB shows uniform performance across all performance metrics. RF and SVM classifiers exhibit identical performance with lowest precision scores in comparison to NB and XGBoost classifiers. This indicates weak classification with high number of False positives. XGBoost classifier stands out with high scores in Accuracy, Recall, and Precision, and with the best F1 score of 0.679. This suggests that XGBoost retained strong classification with effective balance between recall and precision compared to RF and SVM, depicted in the table below.

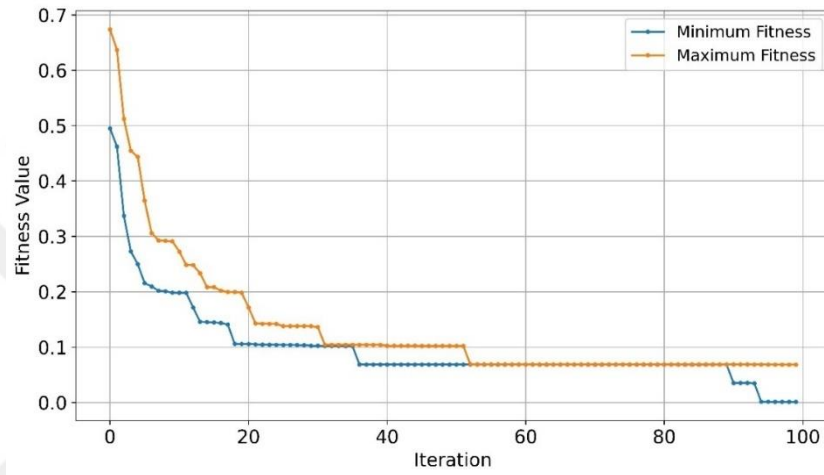
Table 5.1: GSE1297 Performance Measurements before Genes Selection.

Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.600	0.600	0.600	0.600
RF	0.700	0.576	0.700	0.489
SVM	0.700	0.576	0.700	0.489
XGBoost	<u>0.700</u>	<u>0.679</u>	<u>0.700</u>	<u>0.675</u>

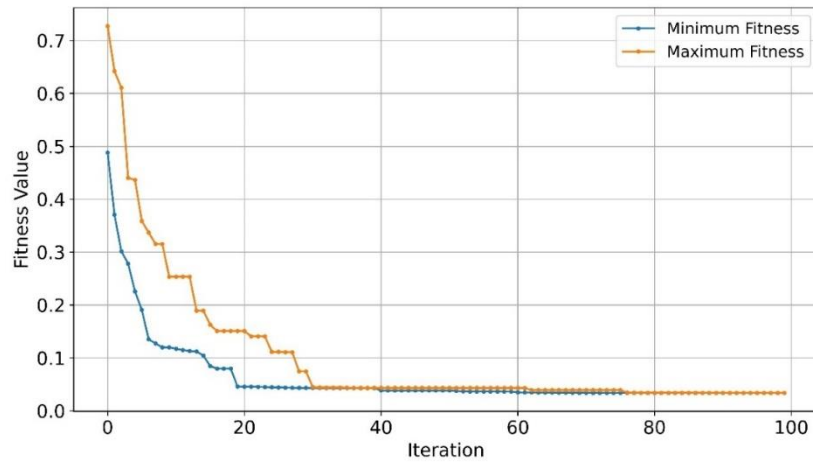
Table 5.2: GSE1297 Performance Measurements after Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.000</u>
RF	0.800	0.762	0.800	0.844	0.904
SVM	0.800	0.762	0.800	0.844	0.523
XGBoost	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>	<u>1.00</u>

Both NB and XGBoost shows perfect scores of 1.00 across all performance metrics including AUC, see Table 5.2. This is interpreted to either potential classifier overfitting to data or evidence of high effective selection of genes. In this context, effective genes selection implies that the selected subset of genes provided a comprehensive representation if the underlying representation and features relationships of data critical to prediction. Figure 5.1 shows convergence of minimum and maximum fitness values obtained at each optimization iteration with EMRFOGS-XGBoost convergence speed outperforming NB.



(a)



(b)

Figure 5.1: Convergence of Minimum and Maximum Fitness Values of (a) EMRFOGS-NB, (b) EMRFOGS-XGBoost, on GSE1297.

RF and SVM shows great improvements across all metrics after apply EMRFOGS genes selection evident in Table 5.3. With 14%, 24%, 14%, and 72% percent increase of RF classifier Accuracy, F1, Recall, and Precision respectively. RF classifier achieved also the highest AUC scores, and with strong EMRFOGS exploration and exploitation, RF classifier reduced false positives negatives and increased the selection of relevant features during optimization iterations. This is evident in Figure 5.2 as the minimum number of selected features starts to converge from early iterations, near iteration 20.

Table 5.3: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE1297 with 16379 Features.

	Number of Features	<i>Global_{best}</i> Convergence Iteration
NB	24	95
RF	<u>4</u>	95
SVM	5	<u>83</u>
XGBoost	6	91

In highlight of previous discussion, all models achieved a substantial reduction of the number of features evident in Table 5.3. Models such as NB and XGBoost maintain the highest number of features in comparison to RF and SVM, while achieving acceptable convergence of 95 and 91 iteration. RF retained the lowest number of features while SVM classifier was the fastest to converge among the three classifiers.

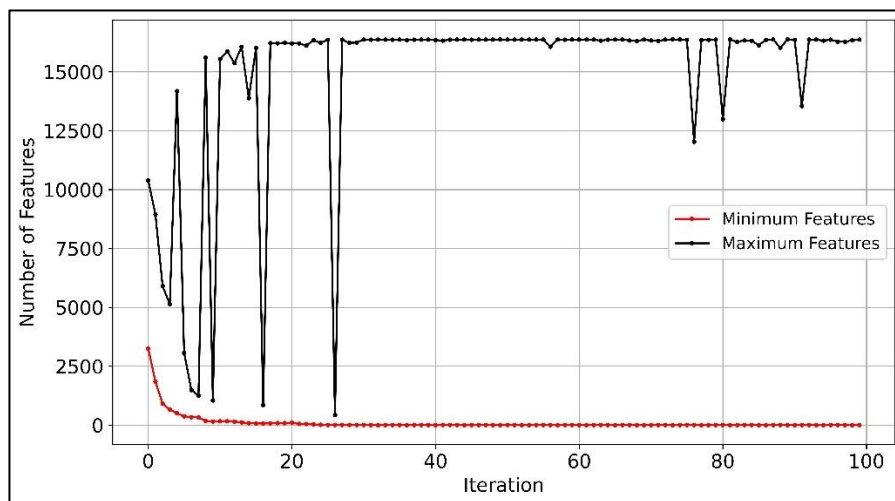


Figure 5.2: Minimum and Maximum Number of Features Selected by EMRFOGS-RF on GSE1297.

5.3.1.1 GSE1297 analysis

The impact of proposed EMRFOGS genes selection on GSE1297 is evident in the achieved increase of performance metrics across four different machine learning models. This demonstrates the effectiveness of proposed genes selection approach in prediction of Alzheimer disease. Second, it also demonstrates EMRFOGS efficient exploration and exploitation of the large features space of GSE1297. Models' specific responses are varied from one to another. Although all models exhibited enhancement in performance metrics, generalized performance can't be determined due to presence of possible overfitting in NB and XGBoost classifiers. All classifiers converged with nearly marginal difference with RF being the best performing classifier in GSE1297 dataset.

5.3.2 GSE5281

Similar to GSE1297 dataset, XGBoost and Naive Bayes remained competitive in classification metrics achieving similar Accuracy and F1 scores and outperformed NB in precision scores, see Table 5.4. NB performance was the poorest in comparison to other classifiers for GSE5281 dataset, however, NB achieved scores can be considered moderate with good balance between model predictions and the true outcome. Support Vector Machine (SVM) demonstrates an exceptional performance in accurate classification of data, achieving the highest scores in accuracy, F1, and recall at 0.938. Additionally, SVM measured precision of 0.945 which further highlights its efficient performance. Conversely to GSE1297, Random Forest was the second-best performing classifier for GSE5281. RF model exhibits superior performance with accuracy, F1, and recall scores of 0.8979, and a slightly higher precision of 0.903, see Table 5.4.

After applying EMRFOGS genes selection, SVM remains the best performing classifier with the highest performance measurement across all classifiers. SVM achieved the highest scores of 0.959 across all performance metrics with 0.962 in precision. RF and XGBoost classifiers achieved the same classification results with RF achieving an impressive AUC of 0.991. However, XGBoost classifier benefited significantly from the reduction of irrelevant genes after applying EMRFOGS. This shows an enhanced sensitivity performance after genes reduction when compared to RF. NB performance was moderate and an increase is evident across all metrics and AUC of 0.92.

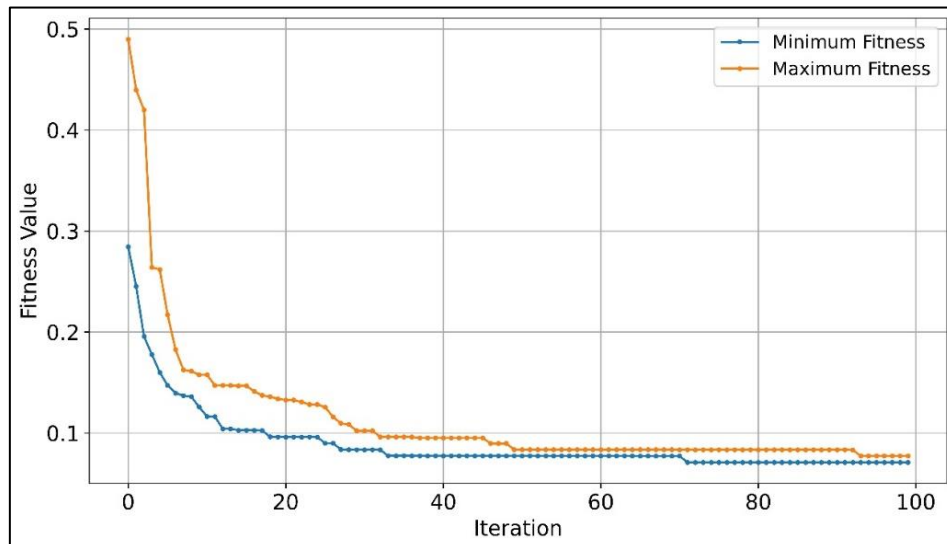
Table 5.4: GSE5281 Performance Measurements before Genes Selection.

Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.857	0.852	0.857	0.887
RF	0.8979	0.8971	0.8979	0.903
SVM	<u>0.938</u>	<u>0.938</u>	<u>0.938</u>	<u>0.945</u>
XGBoost	0.857	0.854	0.857	0.871

Table 5.5: GSE5281 Performance Measurements after Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	0.877	0.876	0.877	0.886	0.92
RF	0.918	0.918	0.918	0.918	<u>0.991</u>
SVM	<u>0.959</u>	<u>0.959</u>	<u>0.959</u>	<u>0.962</u>	0.984
XGBoost	0.918	0.918	0.918	0.918	0.983

The figures below show EMRFOGS fitness convergence for NB, SVM, and XGBoost classifiers on GSE5281 dataset.

**Figure 5.3:** Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE5281.

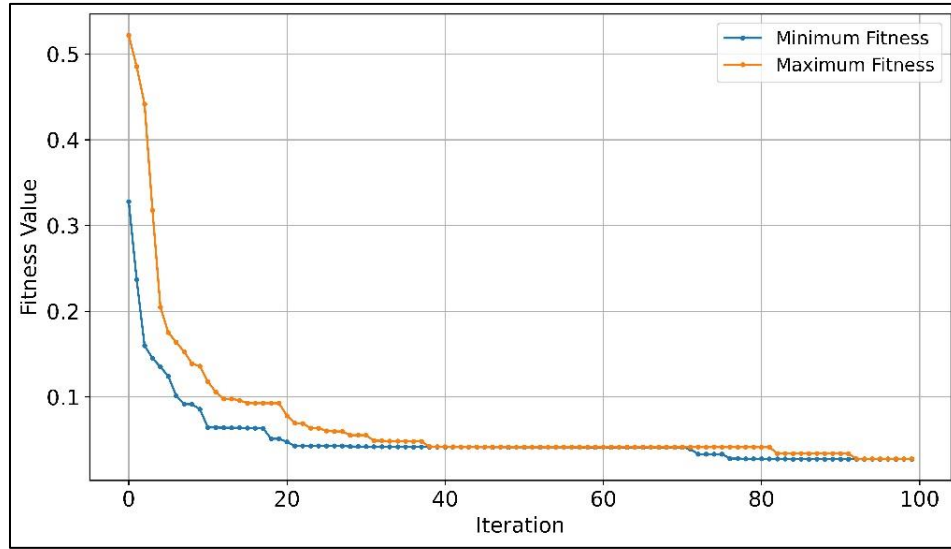


Figure 5.4: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-SVM on GSE5281.

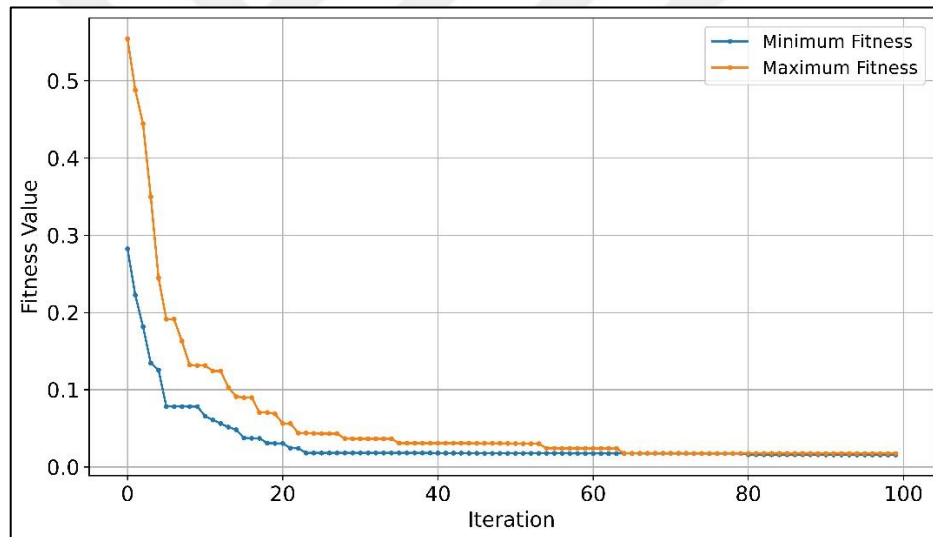


Figure 5.5: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-XGBoost on GSE5281.

While NB was the worst performing classifier, Table 5.6 shows that NB achieved the highest reduction in the number of genes, with only 162 genes out of 54674. This indicates that the NB is not sensitive enough to identify the right subset of genes contributing to increase of Alzheimer disease prediction, thereby the solution converged at early iteration 70 with no further improvements of prediction accuracy, see Figure 5.6. SVM and XGBoost retained the same convergence at iteration 89 with SVM being more efficient in identifying the relative genes of only 165 gene relatives to prediction process.

Table 5.6: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE5281 with 54674 Features.

	Number of Features	<i>Global_{best}</i> Convergence Iteration
NB	<u>162</u>	<u>70</u>
RF	238	72
SVM	165	89
XGBoost	220	89

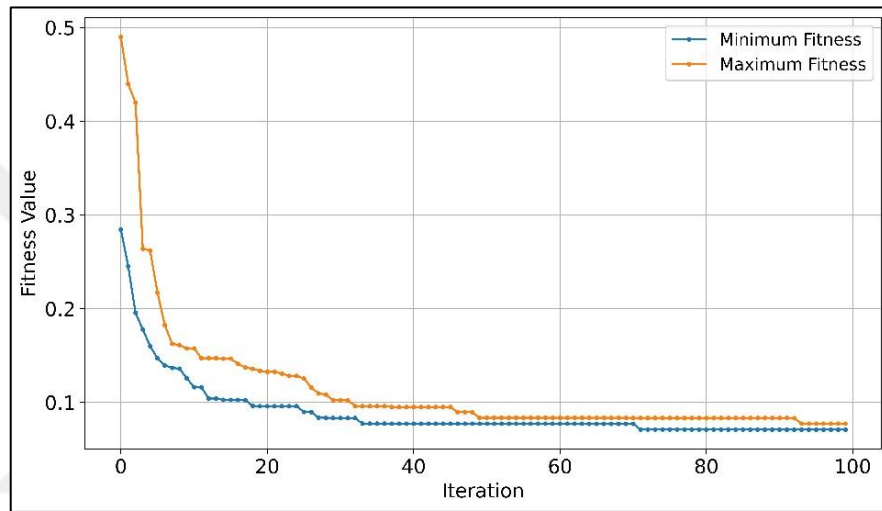


Figure 5.6: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE5281.

5.3.2.1 GSE5281 analysis

EMRFOGS achieved more generalized performance on GSE5281 in comparison to previous dataset. The performance increase is evident in Table 5.5 across all classifiers measured metrics. Additionally, high AUC scores of all models suggests good classification ability and discrimination between classes labels. Increase gain across all performance metrics is evident with percentage between 2% to 6%. This indicates that EMRFOGS strong balance of exploration and exploitation was consistent during each individual test. Considering the different theoretical and mathematical structure of each classification algorithm, EMRFOGS successfully navigated the features complex search space independent of utilized classification method efficiently reducing computational complexity and increasing prediction scores. Figure 5.7 illustrates ROC of four classifiers.

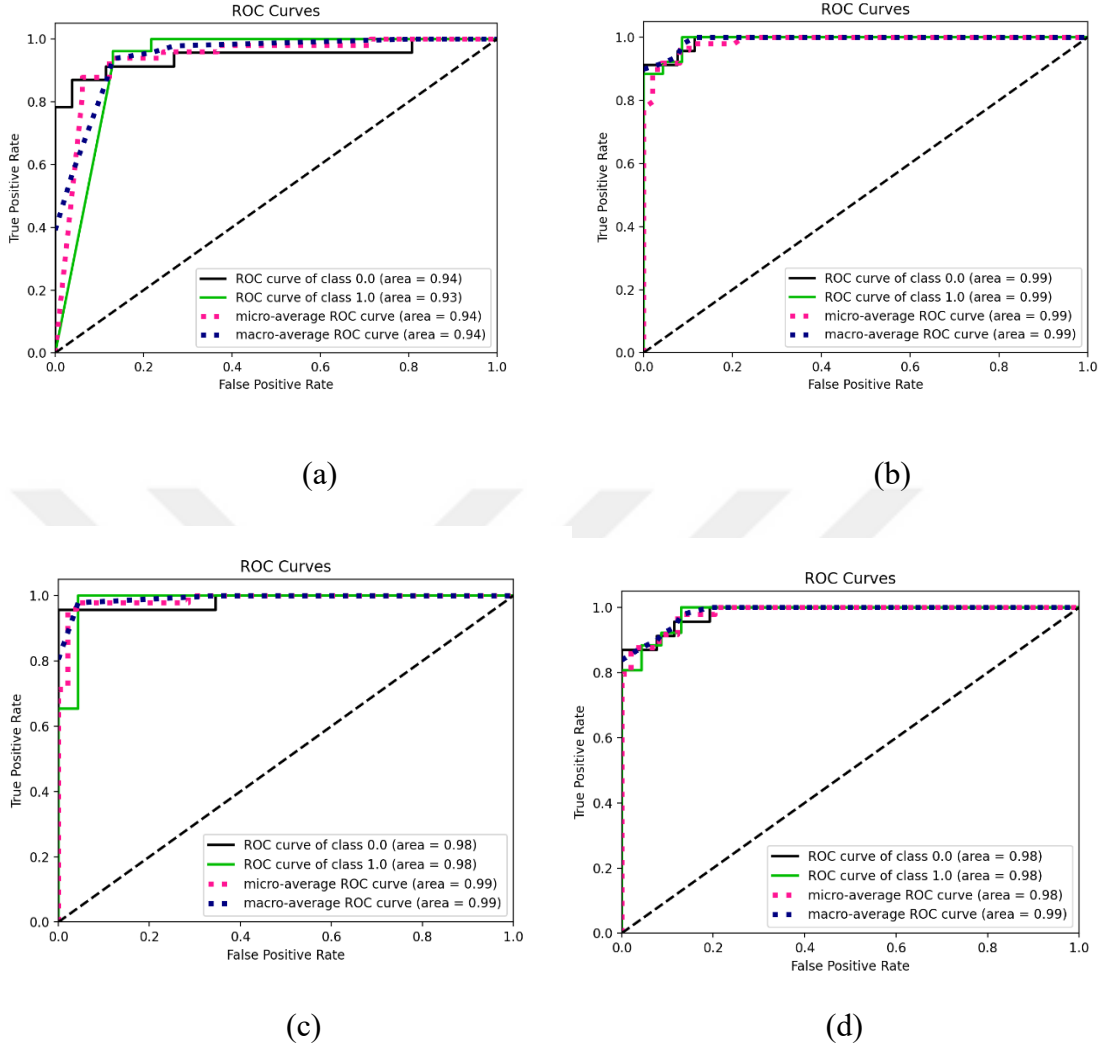


Figure 5.7: ROC for: (a) EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.

5.3.3 GSE33000

Similar to GSE1297, XGBoost classifier retained the highest performance scores in all measurements. In initial assessment and before genes selection, XGBoost outperformed classifiers with 0.920 score across Accuracy, F1, and Recall, and 0.921 in Precision. XGBoost high performance is evident in more than one dataset before genes selection. RF demonstrated strong performance as the second-best classifier with nearly $\sim 1.73\%$ weaker scores across all measures. Being the highest performing classifier after genes selection in GSE1297 dataset, RF remains competitive to other classifiers with good balance in both

classification and error minimalization. With SVM performance is nearly $\sim 7.19\%$ weaker than RF, NB remained the worst classifier in terms of performance metrics, see Table 5.7. While NB retained competitive scores of only $\sim 9.45\%$ less compared to best performing classifier in GSE5281 dataset, its performance is $\sim 28.13\%$ worse than best performing classifier in GSE33000. This decrease in performance can be attributed to NB developing classification bias towards the majority of class due to increase of unbalanced data samples from 161 to 624 for GSE5281 and GSE33000 respectively. In addition, it can be hypothesized that samples increase introduced noise and more variability to the data such that NB failed to capture underlying representation of data.

Table 5.7: GSE33000 Performance Measurements before Genes Selection.

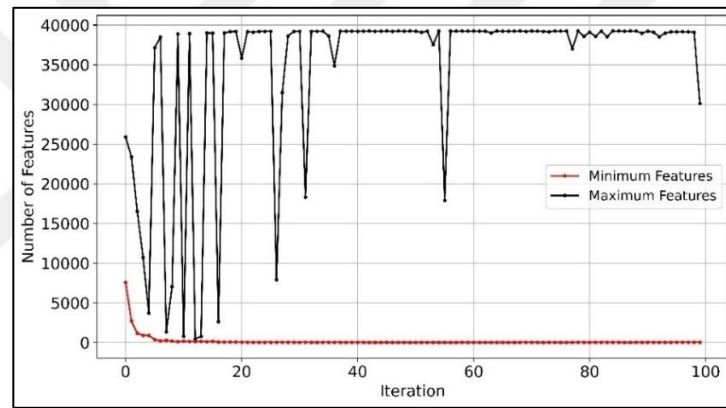
Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.718	0.720	0.718	0.748
RF	0.904	0.903	0.904	0.906
SVM	0.835	0.833	0.835	0.840
XGBoost	<u>0.920</u>	<u>0.920</u>	<u>0.920</u>	<u>0.921</u>

As for the third dataset GSE33000, applying EMRFOGS genes selection significantly increased performance metrics for NB and SVM. Starting with NB, the results depicted in Table 5.8 indicates a significant increase of nearly $\sim 17.68\%$ across all metrics with AUC of 0.930. the observed enhancements illustrate the impact of EMRFOGS algorithm genes selection in enhancing NB classification performance after removing of irrelevant genes, thus reducing data complex variability and noise. SVM classifier scores the highest performance across all metrics with $\sim 8.14\%$ improvement and AUC of 0.97 after EMRFOGS genes selection. Conversely for RF and XGBoost, both classifiers showed decrease in performance after genes selection with more than $\sim 2.1\%$.

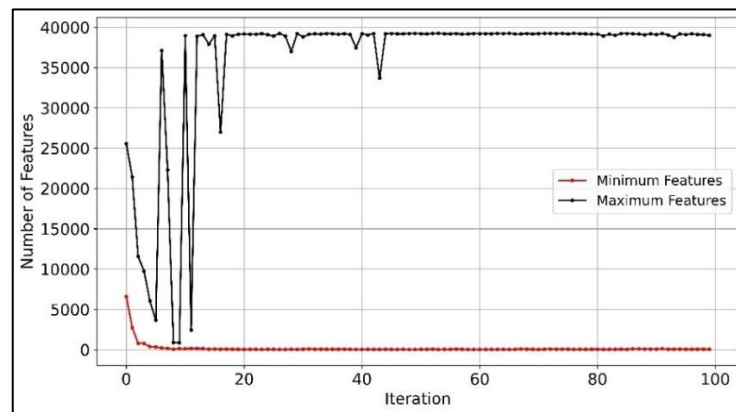
Table 5.8: GSE33000 Performance Measurements after Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	0.8457	0.8458	0.8457	0.8490	0.930
RF	0.872	0.871	0.872	0.886	0.966
SVM	<u>0.909</u>	<u>0.909</u>	<u>0.911</u>	<u>0.972</u>	0.97
XGBoost	0.8989	0.8983	0.8989	0.9013	<u>0.982</u>

Figure 5.8 shows superior convergence performance in minimum and maximum number of features for NB and SVM classifiers, with EMRFOGS performing better in terms of exploration and exploitation utilizing NB classifier.



(a)



(b)

Figure 5.8: Minimum and Maximum Number of Features Selected by: (A) EMRFOGS-NB, (B) EMRFOGS-SVM on GSE33000.

RF relies heavily on features randomness to construct multiple decision trees, resulted in elimination of certain features that were contributing to its higher performance before genes selection. Additionally, the removal of certain genes and sharp low number of genes retained by RF converged the EMRFOGS optimization to local minima and prevented further solution refinement, see Table 5.9 and Figure 5.9. Furthermore, XGBoost gradient boosting model sequentially builds trees and correct errors from previous trees. While XGBoost retained the highest number of genes as depicted in Table 5.9 and Figure 5.10, the removal of certain genes by EMRFOGS seems to negatively impact XGBoost learning process. To mitigate the negative impact of features selection on such models, one should consider fine-tuning of model's hyper-parameters or utilization of hybridized approach to assist the selection process such as features importance reevaluation.

Table 5.9: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE33000 with 39279 Features.

	Number of Features	<i>Global_{best}</i> Convergence Iteration
NB	70	<u>41</u>
RF	<u>64</u>	64
SVM	225	79
XGBoost	431	74

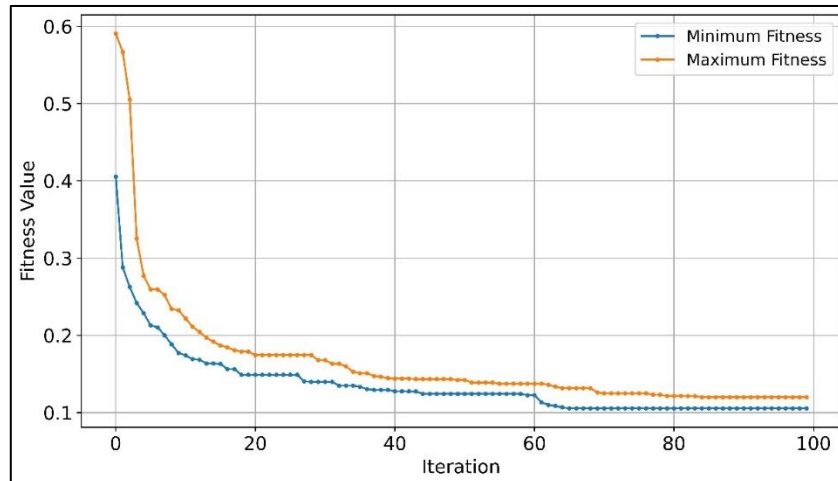


Figure 5.9: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-RF on GSE33000.

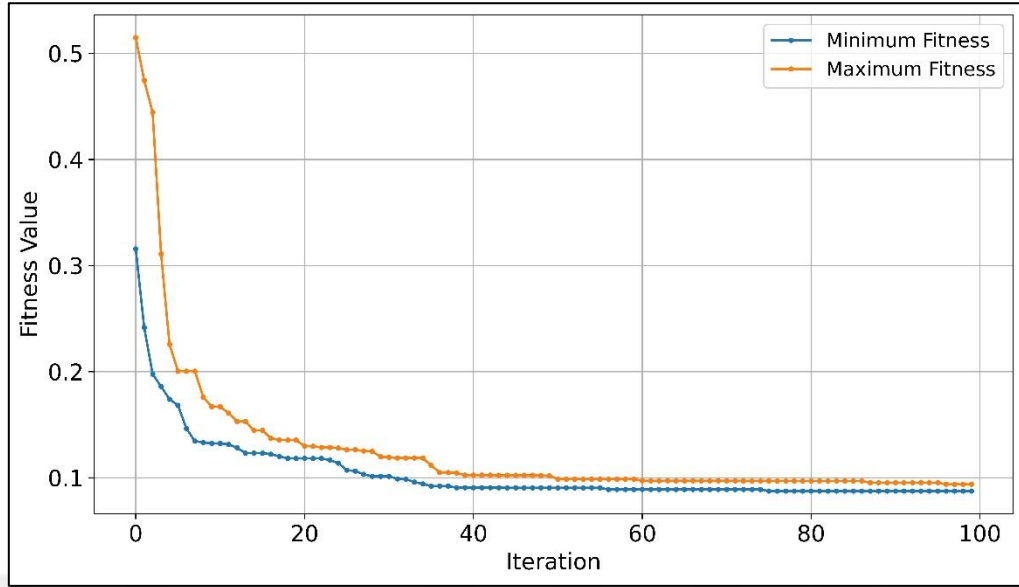


Figure 5.10: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-XGBoost on GSE33000.

5.3.3.1 GSE33000 analysis

The results obtained from testing GSE33000 showed different performance behavior in comparison to first two datasets. NB and SVM classifiers gained $\sim 17.68\%$ and $\sim 8.14\%$ performance increase respectively after applying EMRFOGS genes selection. Both models showed enhanced in precision and recall measures, translated to low false positive and false negative respectively which is particularly important in Alzheimer disease patients' classifications. However, attributed to noise, imbalance data classes, and learning model underlying theory, RF and XGBoost exhibited performance decrease of more than $\sim 2.1\%$. Fine tuning of hyper-parameters for both classifiers and utilization of hybridization methods is required to further investigate performance impact on GSE33000. All models ROC measures are illustrated in Figure 5.11.

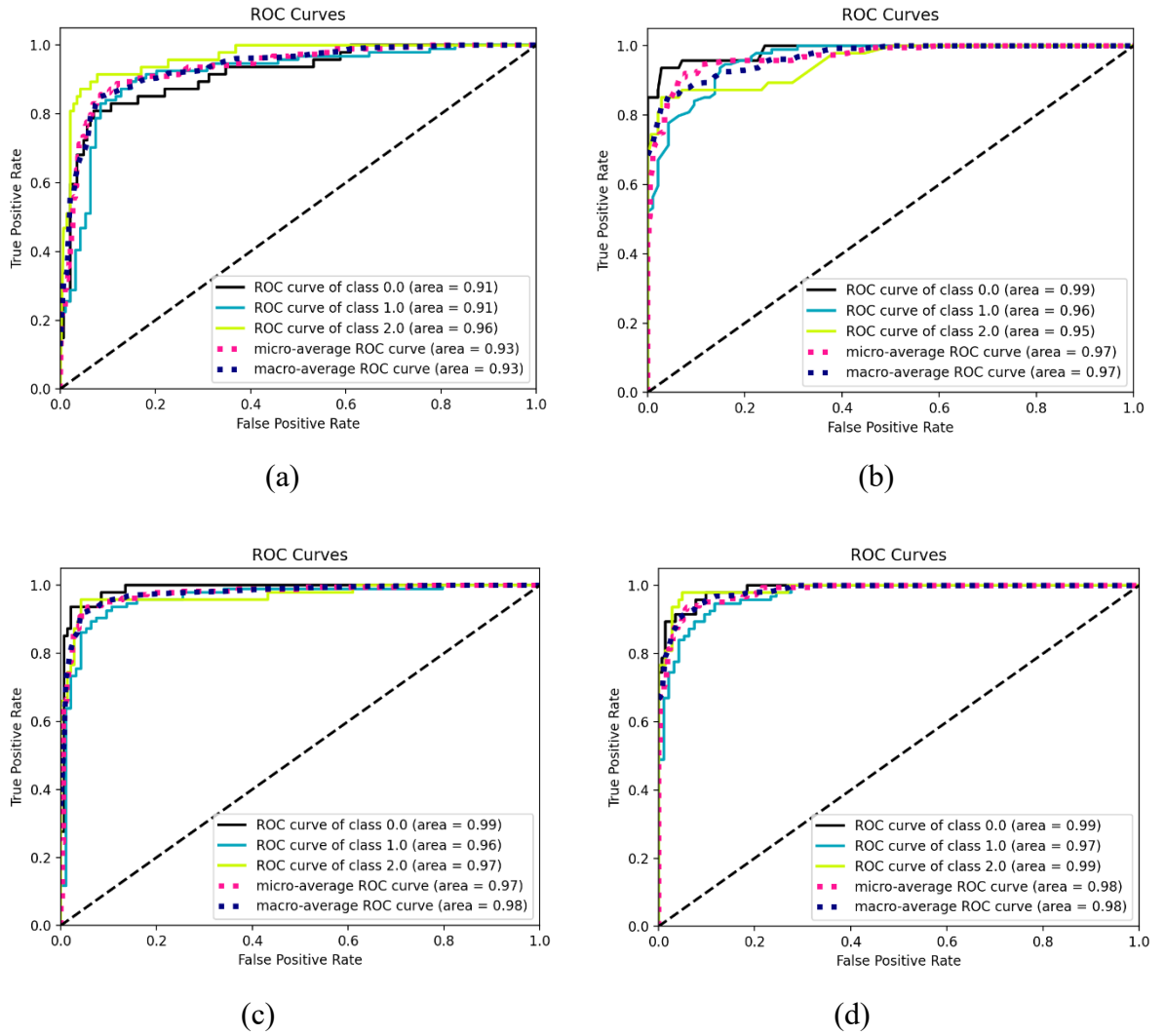


Figure 5.11: ROC for: (a) EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.

5.3.4 GSE44768

The results illustrated in Table 5.10 of GSE447768 dataset, SVM classifier received the highest classification performance across all metrics. It is clear from the evaluated genes expression datasets that SVM attained the best generalized performance across all GSE datasets. SVM performance is consistent and efficient in GSE higher-dimensional features space. XGBoost retained the second-best performing classifier showing strong results in all metrics of 0.884. The efficient high scores suggests that XGBoost effectiveness in handling the complexity of high dimensional space of GSE datasets is directly related to the number of samples. Given, GSE44768 is nearly half as much in the number of samples when

compared to GSE33000 with 624 samples. Although, further investigation of XGBoost classifiers relevant to GSE datasets is required. NB remained the worst performing classifier with nearly ~20.13% different in performance when compared to SVM. In light of current performance metrics and others obtained from previously tested GSE datasets, it is clear that NB classifier struggle to efficiently process high-dimensional datasets. While the NB maintained the lowest number of features and converged faster than other classifiers, accuracy and other performance metrics remained significantly low even after genes selection in comparison to other classifiers. This is not to reflect on the EMRFOGS performance, however, to emphasis that the low scores of NB is due to the fundamental assumption of NB theoretical model that treats each feature statistically independent from other features. While the independence assumption of NB simplifies the model and reduces computational burden which renders NB classifier to be especially suitable in classification scenarios of high-dimensional features spaces. It is evident that NB struggled to efficiently capture complex patterns of GSE datasets.

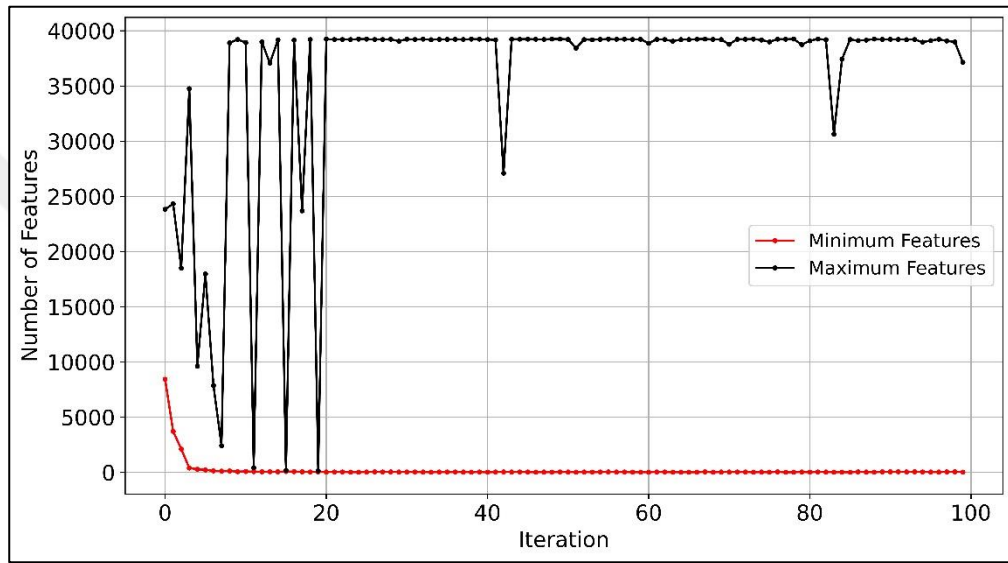
Table 5.10: GSE44768 Performance Measurements Before Genes Selection.

Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.7681	0.7688	0.7681	0.7707
RF	0.8405	0.8408	0.8405	0.8414
SVM	<u>0.9130</u>	<u>0.9134</u>	<u>0.9130</u>	<u>0.9199</u>
XGBoost	0.8840	0.8840	0.8840	0.8840

After applying EMRFOGS, see Table 5.11, NB classifier gained a significant increase outperforming all other classifiers in all metrics with scores of 0.971% and a 0.987% of AUC. Such improvement suggests that subset of selected genes by EMRFOGS aligned with NB's statistical model given the substantial decrease in number of genes from 39279 to only 119 gene, see Table 5.12. Figure 5.12 and 5.13 show features and fitness convergence of EMRFOGS of NB classifier. EMRFOGS shows strong exploration and features convergence from iteration ~5. SVM showed marginal performance improvement across all metrics similar to RF. However, XGBoost achieved perfect scores across all metrics suggesting similar overfitting behavior when tested on GSE1297.

Table 5.11: GSE44768 Performance Measurements Before Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	<u>0.9710</u>	<u>0.9711</u>	<u>0.9710</u>	<u>0.9728</u>	<u>0.9871</u>
RF	0.8405	0.8411	0.8405	0.8449	0.900
SVM	0.9275	0.9276	0.9275	0.9281	0.981
XGBoost	1.00	1.00	1.00	1.00	1.00

**Figure 5.12:** Minimum and Maximum Number of Features Selected by EMRFOGS-NB on GSE44768.

For this dataset, NB achieved the highest performance scores and was the fastest to converge at iteration 23 with 119 genes, see Table 5.12. XGBoost retained the lowest number of genes of only 59 with the highest number of genes retained by SVM. EMRFOGS with XGBoost shows similar exploration strength compared to NB classifier but with more solution discovery rate evident in Figure 5.14. Overall, all classifiers performance were superior and the genes reduction was achieved with high AUC scores.

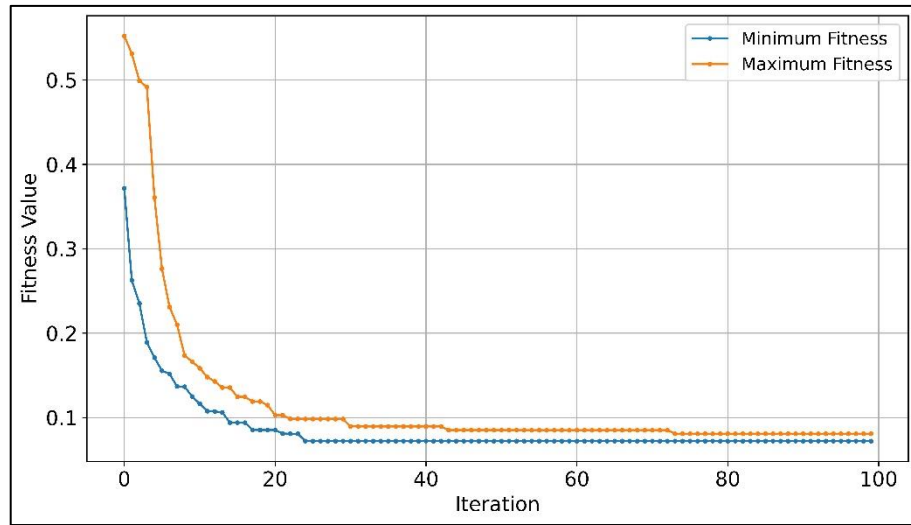


Figure 5.13: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-NB on GSE44768.

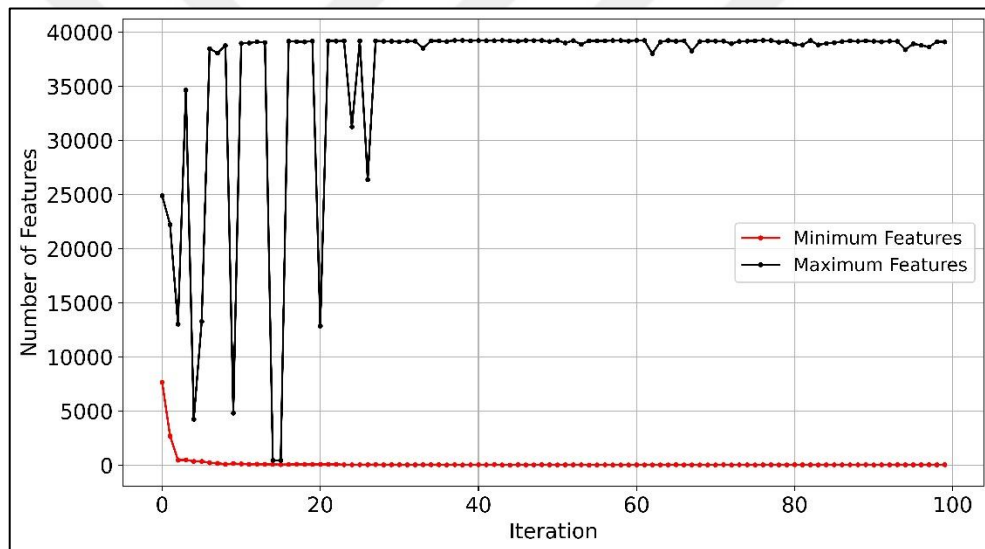


Figure 5.14: Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE44768.

Table 5.12: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE44768 with 39279 Features.

	Number of Features	<i>Global_{best}</i> Convergence Iteration
NB	119	<u>23</u>
RF	134	38
SVM	168	65
XGBoost	<u>59</u>	83

5.3.4.1 GSE44768 analysis

The comprehensive analysis of GSE44768 obtained results aligned with previously tested GSE datasets showed that NB classifier scored the highest improvement across all classifiers with 27.63% in performance metrics. RF and XGBoost classifiers performance aligned with decrease performing results obtained from GSE33000. RF showed no improvements after EMRFOGS genes selection while XGBoost shows potential of overfitting especially when considering AUC scores. It is an overall challenge to hypothesis the exact issues concerning varying degrees of performance enhancements across all classifiers, for the reason that genes selection optimization can be more compatible with some models over others in specific datasets. Therefore, it is imperative to further investigate RF and XGBoost classifiers on more GSE datasets with hyper-parameters tuning to generalize their performances.

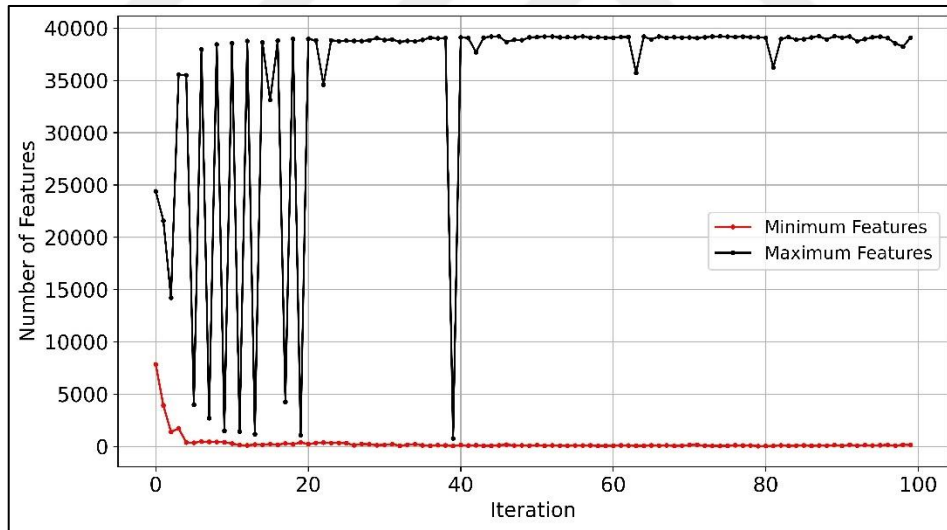
5.3.5 GSE44770

With 388 samples and 39,279 genes, SVM classifier took the lead again as the highest performing classifier across all metrics with AUC scores of 0.90, see Table 5.13. This further show SMV consistency and superior performance in class separation and error minimization of high-dimensional GSE datasets. RF and XGBoost scored the same across all metrics retaining the second-best classifiers after SVM with RF overscoring XGBoost in the number of the selected genes of 73 compared to 383, see Table 5.15. Overall, classification performance and generalization of four learning models on GSE44770 can be considered equally superior when compared to previously four GSE datasets with nearly ~5.95% efficiency difference between the best and the worst performing classifiers, SVM and NB.

Table 5.13: GSE44770 Performance Measurements before Genes Selection.

Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.8405	0.8411	0.8405	0.8449
RF	0.8695	0.8692	0.8695	0.8693
SVM	<u>0.8985</u>	<u>0.8989</u>	<u>0.8985</u>	<u>0.9088</u>
XGBoost	0.8695	0.8692	0.8695	0.8693

After applying EMRFOGS genes selection, SVM shows a significant improvement with nearly ~5.61% across all performance metrics with gained AUC of 0.98, see Table 5.14. RF shows decrease in performance similar to GSE33000. Conversely, XGBoost shows considerable improvement of ~5.81% with no potential of overfitting when compared to GSE33000 and GSE1297. However, EMRFOGS with XGBoost showed higher discovery rate of solutions compared to EMRFOGS with SVM, see Figures 5.15 and 5.16 respectively.

**Figure 5.15:** Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE44770.

NB also benefited from EMRFOGS genes selection optimization with marginal increase of nearly ~1.5%, however, NB remained the worst performing classifier across all GSE datasets even after EMRFOGS genes selection, see Table 5.14. Efficiency differences between the best and worst performing classifiers increased to ~11.90%, SVM and RF respectively. This suggests that genes selection optimization can be more compatible with

some models over others dependent on number of factors including the specifics and characteristics of utilized dataset.

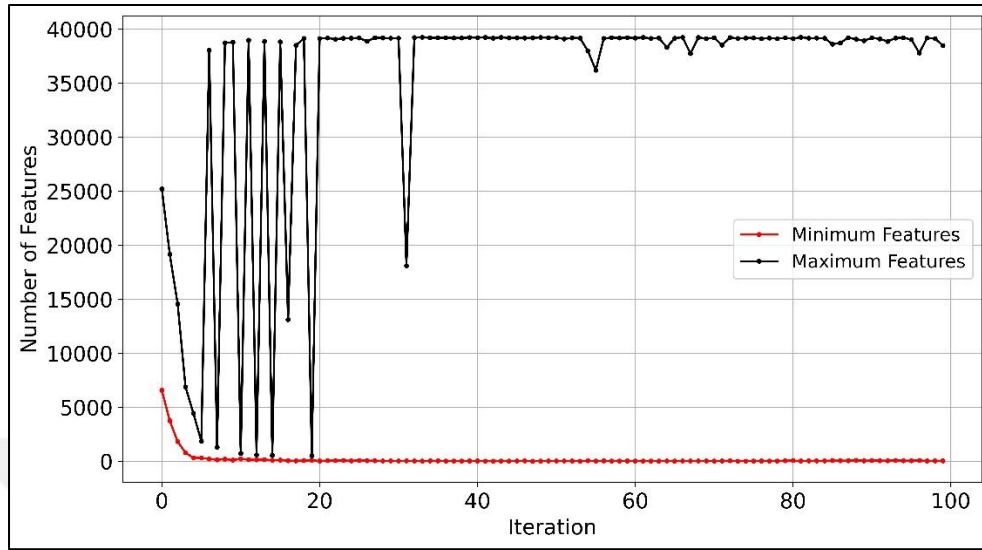


Figure 5.16: Minimum and Maximum Number of Features Selected by EMRFOGS-SVM on GSE44770.

Table 5.14: GSE44770 Performance Measurements after Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	0.8550	0.8553	0.8550	0.8787	0.931
RF	0.8405	0.8408	0.8405	0.8414	0.932
SVM	<u>0.9420</u>	<u>0.9413</u>	<u>0.9420</u>	<u>0.9474</u>	<u>0.9846</u>
XGBoost	0.9130	0.9130	0.9130	0.9130	0.970

RF performance in GSE44770 was similar to GSE33000, in both datasets RF performance decreased after EMRFOGS genes selection while retaining the lowest number of genes of 73 and 64 respectively, see Table 5.14. It can be hypothesized that this behavior is attributed to insensitivity of RF to different combination changes in subset of selected genes. Thereby, the sharp and low number of selected genes diminishes the effect of other objectives included in optimization fitness function. As such, forcing convergence to local minima and prevent optimization algorithm from the discovery of new global optimal solution with higher accuracy. This is evident when comparing minimum and maximum fitness convergence curves of EMRFOGS-RF with EMRFOGS-SVM, shown in Figures

5.17 and 5.18 respectively. Both curves never converged to optimal solution $Global_{best}$ suggesting that due to RF insensitivity failed to lead EMRFOGS exploration-exploitation to optimal search areas.

Table 5.15: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE44770 with 39279 Features.

	Number of Features	$Global_{best}$ Convergence Iteration
NB	314	97
RF	<u>73</u>	95
SVM	212	82
XGBoost	383	<u>60</u>

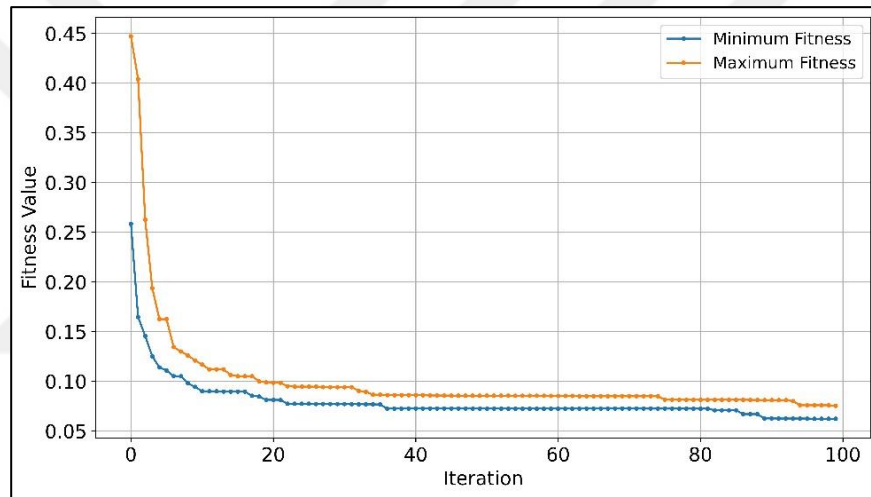


Figure 5.17: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-RF on GSE44770.

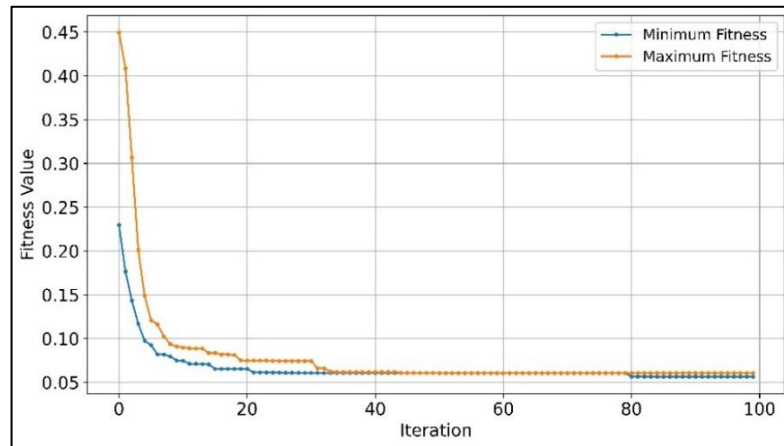


Figure 5.18: Convergence of Minimum and Maximum Fitness Values of EMRFOGS-SVM on GSE44770.

5.3.5.1 GSE44770 analysis

Overall, all the models performed substantially better with superior reduction in genes. It further shows that strong genes selection can enhance prediction of Alzheimer disease. Low efficiency difference is achieved in comparison to previously tested GSE datasets, which suggest more performance generalization of utilized classifiers. SVM was the highest performing classifier while RF showed similar behavior when tested on GSE333000. XGBoost shows no potential overfitting with $\sim 5.81\%$ performance improvements and notable AUC scores of 0.97. The extent of EMRFOGS enhancement is varied across all classifier, highlighting the importance of further investigation such as using hybridization approach, utilization of different optimization function, and learning models' parameters tuning. Strategic improvements can lead to substantial enhancement and more generalized performance across different datasets. All models ROC measures are illustrated in Figure 5.19.

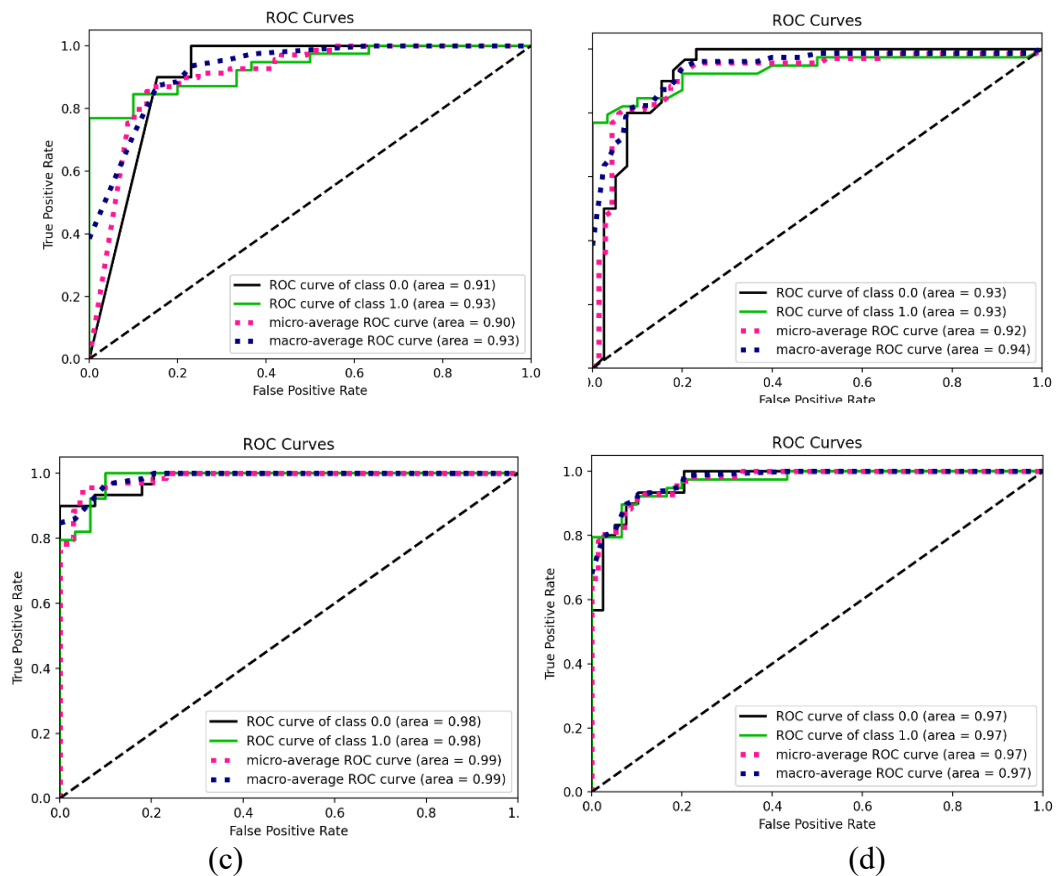


Figure 5.19: ROC for: (a) EMRFOGS-NB, (b) EMRFOGS-RF, (c) EMRFOGS-SVM, (d) EMRFOGS-XGBoost.

5.3.6 GSE132903

From the comprehensive analysis of GSE132903 dataset results depicted in Table 5.16, RF attained the highest performing classifier. The increased number of genes in GSE132903 of 42,178 from 39,279 in previously tested three GSE datasets suggests that RF performance is more dependent to the number of samples than number of features in dataset. RF outperformed SVM by nearly ~12.5% in performance scores, which the highest in comparison to RF and SVM scores across all tested GSE datasets. The reduced number of samples alleviate class imbalance and reduced data noise, thereby enhancing RF efficiency in capturing underlying representation of the data. Conversely, SVM performed the worst with accuracy of 0.77% among other classifiers with NB being the second-best scoring 0.86% across all performance metrics. XGBoost remained competitive with 0.84%.

Table 5.16: GSE132903 Performance Measurements before Genes Selection.

Model	Measurements			
	Accuracy	F1	Recall	Precision
NB	0.8644	0.8641	0.8644	0.8659
RF	<u>0.8813</u>	<u>0.8809</u>	<u>0.8813</u>	<u>0.8851</u>
SVM	0.7796	0.7795	0.7796	0.7798
XGBoost	0.8474	0.8469	0.8474	0.8508

With performance increase of nearly ~18.18%, SVM was the highest performing classifier across all metrics with 0.9152% and AUC of 0.9436, as such highlighting SVM's enhanced efficiency and accuracy after EMRFOGS genes selection, see Table 5.17. XGBoost maintained steady performance with AUC of 0.93 with no improvement in performance. Furthermore, XGBoost retained the highest number of genes and convergence iteration in more than one of the tested GSE datasets, see Tables 5.18, 5.9, and 5.6 respectively. It can be hypothesized that due to nature of gradient boosting and characteristics of GSE datasets, XGBoost requires a higher number of optimization iterations to accurately capture the complex genes relationships in GSE data. The performance of XGBoost showed significant variation in relation to the number of GSE data samples. Notably, there was a tendency towards overfitting in smaller datasets,

whereas higher number of optimization iterations is hypothesized for larger GSE data samples.

Table 5.17: GSE132903 Performance Measurements after Genes Selection.

Model	Measurements				
	Accuracy	F1	Recall	Precision	AUC
NB	0.8983	0.8981	0.8983	0.900	0.939
RF	0.8474	0.8474	0.8474	0.8508	0.9206
SVM	<u>0.9152</u>	<u>0.9152</u>	<u>0.9152</u>	<u>0.9157</u>	<u>0.9436</u>
XGBoost	0.8474	0.8474	0.8474	0.8479	0.9321

While NB classifier gained nearly ~3.48% performance increase, RF interestingly shows a decrease in performance across accuracy, F1, and recall, all dropping to 0.8474, with slight decrease in precision to 0.8508. NB was also the fastest to converge to subset of 155 selected genes at iteration 60 compared to RF 151 selected gene at iteration 78, see Table 5.18. SVM shows a substantial reduction of features space from 42,178 to only 46, attained at iteration 99. Overall, EMRFOGS performance remained consistent across all classifiers due to strong balance of exploration and exploitation. NB, RF, and XGBoost show a balance between the number of features and convergence times. However, it is evident from Figures 5.20, 5.21, 5.22 that EMRFOGS-XGBoost achieved better exploration performance in comparison to other classifiers

Table 5.18: Number of Selected Genes and Convergence Iteration of EMRFOGS Tested on GSE132903 with 42178 Features.

	Number of Features	<i>Global_{best}</i> Convergence Iteration
NB	155	<u>60</u>
RF	151	78
SVM	<u>46</u>	99
XGBoost	154	88

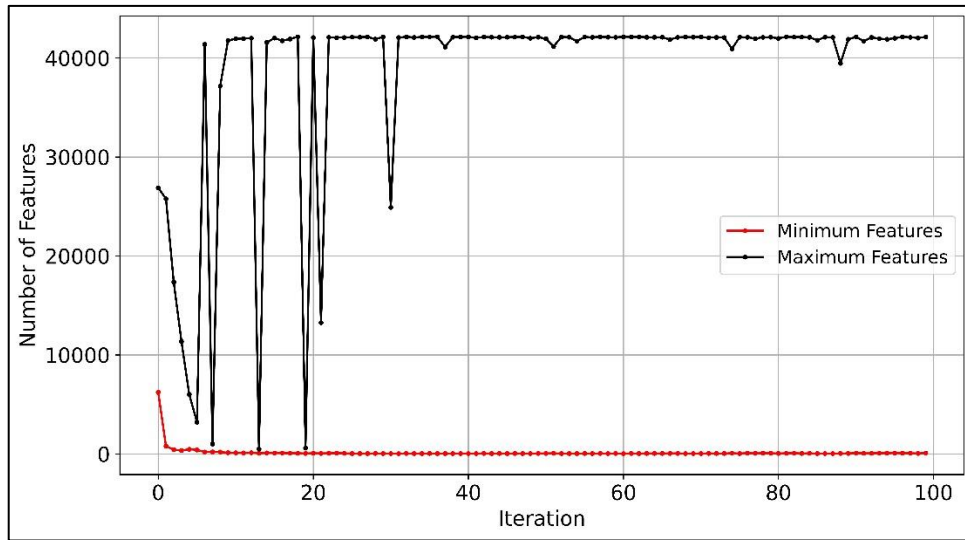


Figure 5.20: Minimum and Maximum Number of Features Selected by EMRFOGS-NB on GSE132903.

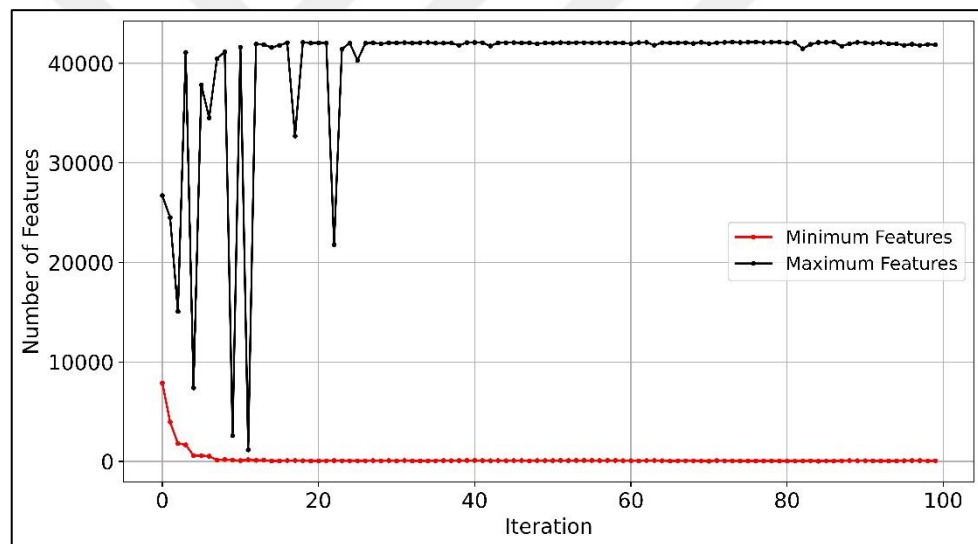


Figure 5.21: Minimum and Maximum Number of Features Selected by EMRFOGS-RF on GSE132903.

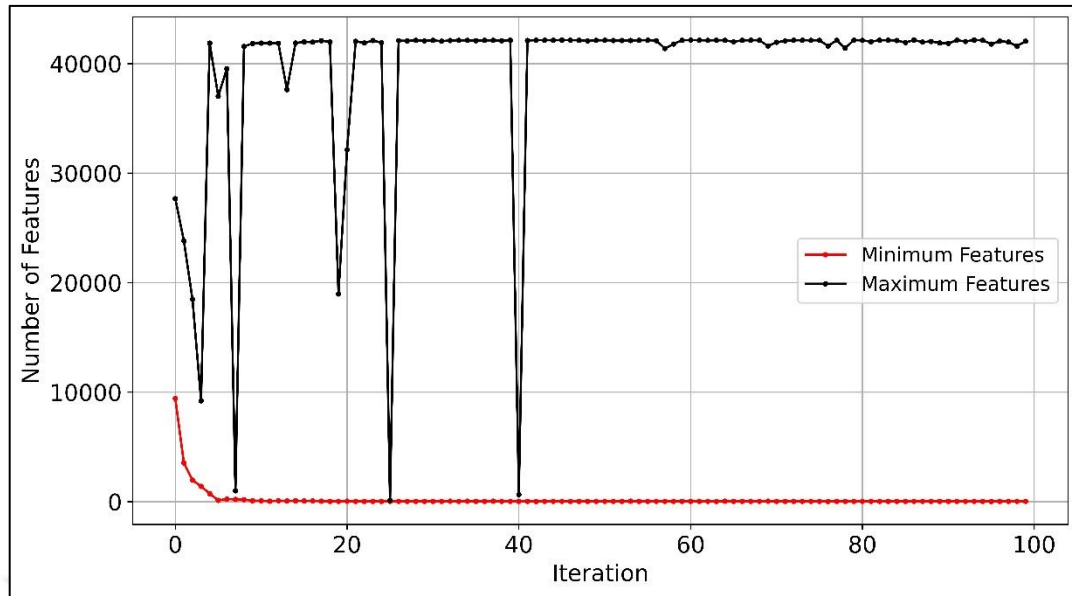


Figure 5.22: Minimum and Maximum Number of Features Selected by EMRFOGS-XGBoost on GSE132903.

5.3.6.1 GSE132903 analysis

Findings of GSE132903 aligns with previous GSE datasets. It became evident that SVM classifier attained the highest performance after EMRFOGS genes selection with high sensitivity and superior consistency to complex GSE datasets. NB, RF, and XGBoost maintained a good overall performance of GSE132903 with NB being the second-best classifier with AUC scores of 0.939 and the fastest to converge at iteration 60. NB in more than one tests retained the lowest number of selected genes but with lower accuracy. XGBoost performance varied from one dataset to another but with competitive results. Assessing different classifiers offers a thorough examination of suggested enhancements and helps in choosing the most appropriate learning models for further testing and in-depth GSE datasets related research. It not only facilitates the efficiency and behavior comparison of each classifier but also improves decision-making for future experimental designs and validation investigations.

6. CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

Alzheimer disease is a progressive neurological illness that manifests itself more severely as the time passes. This disease is characterized by the generation of extracellular amyloid plaques and aberrant amyloid beta build-up that causes the loss of neuronal tissues in the brain. Both of the pathogenesis and the etiology of Alzheimer's disease remain unknown to this time.

Through the analysis of gene expression microarrays, supervised machine learning algorithms have revolutionized the detection and early diagnosis of the disease. However, challenges surrounding this approach is summarized in the curse of dimensionality caused by high dimensional disease's microarray datasets. As such, reduces the effectiveness of disease diagnosis due to overfitting and bias. To address this challenge, genes selection methods are applied to alleviate the effect of Alzheimer high dimensional genes expression datasets.

This thesis presented a novel bio-inspired wrapper using Mant Ray Foraging Optimization Algorithm designed to efficiently explore the vast and large search space of genes expression datasets. The enhanced MRFO with Sign Random Mutation, Best Rank approach, and Hybrid Boundary Function proposed in this study not only reduced the number of genes significantly, improved the prediction accuracy, but also reduced the computation burden of tested classifiers. Evident from the results, the proposed EMRFOGS reduced the dimensional space of microarray genes expression datasets and significantly increased the prediction of Alzheimer disease and performance metrics through effective identification of relevant genes. While performance improvements varied from one classifier to another given the dimensional differences and statistical relationship distribution among features in each of the six tested datasets, proposed EMRFOGS can be used efficiently in clinical settings to provide precise prediction necessary for Alzheimer disease patients. Overall, performance analysis showed high and strong and balance of exploration-exploitation in navigating complex features space and adaptability to different optimization scenarios. In addition, convergence rate across evaluated classifiers utilizing six genes expression datasets was significant.

6.2 FUTURE WORKS

To advance the treatment of Alzheimer disease through use of machine learning and genes expression datasets, several areas of future research can be identified:

- a. **Algorithm Enhancement:** while EMRFOGS proved to be efficient and consistent across multiple evaluation scenarios, further enhancement of EMRFOGS may include enhancing genes selection based on hybrid approaches, such as inclusion of biomedical information of each gene in optimization fitness evaluation process. Second, the use of adaptive penalty function to eliminate weak solutions and increase convergence rate. Hybridization approach should not be considered as an improvement considering the increase of time complexity as such limiting the implementation of EMRFOGS in real-time diagnosis applications.
- b. **Algorithm Scalability:** to further improve scalability of the EMRFOGS through parallel processing and cloud computing of population solutions space. This not only contribute to optimization of extra-large optimization problems related to bigger microarray datasets, but also in exploration of sparse solution space with limited viable solutions.
- c. **Data Integration:** In addition to microarray data, the integration of multi-omics data such as genomics, proteomics, and metabolomics may provide a more thorough comprehension of Alzheimer's disease and discovery of relevant genes and novel biomarkers using EMRFOGS. This approach can provide valuable insights that cannot be obtained only from the examination of a single GSE dataset.

REFERENCES

- [1] M. Hüttenrauch et al., "Glycoprotein NMB: a novel Alzheimer's disease associated marker expressed in a subset of activated microglia," *Acta Neuropathol. Commun.*, vol. 6, no. 1, p. 108, Dec. 2018, doi: 10.1186/s40478-018-0612-3.
- [2] Preman, Pranav, Maria Alfonso-Triguero, Elena Alberdi, Alexei Verkhratsky, and Amaia M. Arranz. "Astrocytes in Alzheimer's disease: pathological significance and molecular pathways." *Cells* 10, no. 3 (2021): 540.
- [3] Siracusa, Rosalba, Roberta Fusco, and Salvatore Cuzzocrea. "Astrocytes: role and functions in brain pathologies." *Frontiers in Pharmacology* 10 (2019): 479091.
- [4] Vasile, Flora, Elena Dossi, and Nathalie Rouach. "Human astrocytes: structure and functions in the healthy brain." *Brain Structure and Function* 222, no. 5 (2017): 2017-2029.
- [5] Holtzman, David M., John C. Morris, and Alison M. Goate. "Alzheimer's disease: the challenge of the second century." *Science translational medicine* 3, no. 77 (2011): 77sr1-77sr1.
- [6] Rocca, Walter A., and Luigi Amaducci. "Epidemiology of Alzheimer's disease." *Neuroepidemiology* (2019): 55-96.
- [7] Lin, Chia-Ching John, and Benjamin Deneen. "Astrocytes: the missing link in neurologic disease?." In *Seminars in pediatric neurology*, vol. 20, no. 4, pp. 236-241. WB Saunders, 2013.
- [8] Science Photo Library: Alzheimer PET Scan: <https://www.sciencephoto.com/media/1144937/view/alzheimer-s-disease-pet-scans>, accessed in 27/05/2024.
- [9] Bumgarner, Roger. "Overview of DNA microarrays: types, applications, and their future." *Current protocols in molecular biology* 101, no. 1 (2013): 22-1.
- [10] Ehrenreich, Armin. "DNA microarray technology for the microbiologist: an overview." *Applied microbiology and biotechnology* 73, no. 2 (2006): 255-273.

- [11] Trevino, Victor, Francesco Falciani, and Hugo A. Barrera-Saldaña. "DNA microarrays: a powerful genomic tool for biomedical and clinical research." *Molecular medicine* 13 (2007): 527-541.
- [12] Bertucci, Francois, Sebastien Salas, Severine Eysteries, Valery Nasser, Pascal Finetti, Christophe Ginestier, Emmanuelle Charafe-Jauffret et al. "Gene expression profiling of colon cancer by DNA microarrays and correlation with histoclinical parameters." *Oncogene* 23, no. 7 (2004): 1377-1391.
- [13] Van Dam, Sipko, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes. "Gene co-expression analysis for functional classification and gene–disease predictions." *Briefings in bioinformatics* 19, no. 4 (2018): 575-592.
- [14] Chatterjee, Prantik, and Nikhil Ranjan Pal. "Construction of synergy networks from gene expression data related to disease." *Gene* 590, no. 2 (2016): 250-262.
- [15] Swanson, Kyle, Eric Wu, Angela Zhang, Ash A. Alizadeh, and James Zou. "From patterns to patients: Advances in clinical machine learning for cancer diagnosis, prognosis, and treatment." *Cell* (2023).
- [16] Vadapalli, Sreya, Habiba Abdelhalim, Saman Zeeshan, and Zeeshan Ahmed. "Artificial intelligence and machine learning approaches using gene expression and variant data for personalized medicine." *Briefings in bioinformatics* 23, no. 5 (2022): bbac191.
- [17] Pirooznia, Mehdi, Jack Y. Yang, Mary Qu Yang, and Youping Deng. "A comparative study of different machine learning methods on microarray gene expression data." *BMC genomics* 9 (2008): 1-13.
- [18] Crespo Márquez, Adolfo. "The curse of dimensionality." In *Digital Maintenance Management: Guiding Digital Transformation in Maintenance*, pp. 67-86. Cham: Springer International Publishing, 2022.
- [19] Verleysen, Michel, and Damien François. "The curse of dimensionality in data mining and time series prediction." In *International work-conference on artificial neural networks*, pp. 758-770. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

- [20] Bach, Francis. "Breaking the curse of dimensionality with convex neural networks." *Journal of Machine Learning Research* 18, no. 19 (2017): 1-53.
- [21] Altman, Naomi, and Martin Krzywinski. "The curse (s) of dimensionality." *Nat Methods* 15, no. 6 (2018): 399-400.
- [22] Yu, Lei, Yue Han, and Michael E. Berens. "Stable gene selection from microarray data via sample weighting." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9, no. 1 (2011): 262-272.
- [23] Mladenić, Dunja. "Feature selection for dimensionality reduction." In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*, pp. 84-102. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
- [24] Zebari, Rizgar, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction." *Journal of Applied Science and Technology Trends* 1, no. 1 (2020): 56-70.
- [25] Velliangiri, S., and S. J. P. C. S. Alagumuthukrishnan. "A review of dimensionality reduction techniques for efficient computation." *Procedia Computer Science* 165 (2019): 104-111.
- [26] Van Cauwenberghe, Caroline, Christine Van Broeckhoven, and Kristel Sleegers. "The genetic landscape of Alzheimer disease: clinical implications and perspectives." *Genetics in Medicine* 18, no. 5 (2016): 421-430.
- [27] Mahendran, Nivedhitha, PM Durai Raj Vincent, Kathiravan Srinivasan, and Chuan-Yu Chang. "Improving the classification of alzheimer's disease using hybrid gene selection pipeline and deep learning." *Frontiers in genetics* 12 (2021): 784814.
- [28] Ahmed, Shaymaa Taha, and Suhad Malallah Kadhem. "Predicting Alzheimer's Disease Using Filter Feature Selection Method." *benefits* 15 (2022): 16.
- [29] Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." *Computers & electrical engineering* 40, no. 1 (2014): 16-28.

- [30] Khalid, Samina, Tehmina Khalil, and Shamila Nasreen. "A survey of feature selection and feature extraction techniques in machine learning." In 2014 science and information conference, pp. 372-378. IEEE, 2014.
- [31] Dhal, Pradip, and Chandrashekhar Azad. "A comprehensive survey on feature selection in the various fields of machine learning." *Applied Intelligence* 52, no. 4 (2022): 4543-4581.
- [32] Kotsiantis, Sotiris. "Feature selection for machine learning classification problems: a recent overview." *Artificial Intelligence Review* 42, no. 1 (2011): 157-176.
- [33] Zou, Cuihua, Li Su, Mika Pan, Liechun Chen, Hepeng Li, Chun Zou, Jieqiong Xie, Xiaohua Huang, Mengru Lu, and Donghua Zou. "Exploration of novel biomarkers in Alzheimer's disease based on four diagnostic models." *Frontiers in Aging Neuroscience* 15 (2023): 1079433.
- [34] Finney, Caitlin A., Fabien Delerue, Wendy A. Gold, David A. Brown, and Artur Shvetsov. "Artificial intelligence-driven meta-analysis of brain gene expression identifies novel gene candidates and a role for mitochondria in Alzheimer's disease." *Computational and Structural Biotechnology Journal* 21 (2023): 388-400.
- [35] Ahmed, Shaymaa Taha, and Suhad Malallah Kadhem. "Optimizing Alzheimer's disease prediction using the nomadic people algorithm." *International Journal of Electrical and Computer Engineering* 13, no. 2 (2023): 2052.
- [36] Lai, Yongxing, Chunjin Lin, Xing Lin, Lijuan Wu, Yinan Zhao, and Fan Lin. "Identification and immunological characterization of cuproptosis-related molecular clusters in Alzheimer's disease." *Frontiers in Aging Neuroscience* 14 (2022): 932676.
- [37] Mahendran, Nivedhitha, and Durai Raj Vincent PM. "Ensemble-based Gene Selection and an Enhanced Deep Multi-Layer Perceptron-based Classification Model for Classifying Alzheimer's disease." (2022).
- [38] El-Gawady, Aliaa, Mohamed A. Makhlof, BenBella S. Tawfik, and Hamed Nassar. "Machine Learning Framework for the Prediction of Alzheimer's Disease Using Gene Expression Data Based on Efficient Gene Selection." *Symmetry* 14, no. 3 (2022): 491.

- [39] Huseby, Carol J., Elaine Delvaux, Danielle L. Brokaw, and Paul D. Coleman. "Blood Transcript Biomarkers Selected by Machine Learning Algorithm Classify Neurodegenerative Diseases including Alzheimer's Disease." *Biomolecules* 12, no. 11 (2022): 1592.
- [40] Mahendran, Nivedhitha, PM Durai Raj Vincent, Kathiravan Srinivasan, and Chuan-Yu Chang. "Improving the classification of alzheimer's disease using hybrid gene selection pipeline and deep learning." *Frontiers in genetics* 12 (2021): 784814.
- [41] Yang, Longxiu, Yuan Qin, and Chongdong Jian. "Screening for core genes related to pathogenesis of Alzheimer's disease." *Frontiers in Cell and Developmental Biology* 9 (2021): 668738.
- [42] Cheng, Jack, Hsin-Ping Liu, Wei-Yong Lin, and Fuu-Jen Tsai. "Machine learning compensates fold-change method and highlights oxidative phosphorylation in the brain transcriptome of Alzheimer's disease." *Scientific reports* 11, no. 1 (2021): 13704.
- [43] Karaglani, Makrina, Krystallia Gourelia, Ioannis Tsamardinis, and Ekaterini Chatzaki. "Accurate blood-based diagnostic biosignatures for Alzheimer's disease via automated machine learning." *Journal of clinical medicine* 9, no. 9 (2020): 3016.
- [44] Perera, Shehan, Kaveesha Hewage, Chamara Gunarathne, Rajitha Navarathna, Damayanthi Herath, and Roshan G. Ragel. "Detection of novel biomarker genes of Alzheimer's disease using gene expression data." In *2020 Moratuwa engineering research conference (MERCon)*, pp. 1-6. IEEE, 2020.
- [45] Patel, Hamel, Raquel Iniesta, Daniel Stahl, Richard JB Dobson, and Stephen J. Newhouse. "Working towards a blood-derived gene expression biomarker specific for Alzheimer's disease." *Journal of Alzheimer's Disease* 74, no. 2 (2020): 545-561.
- [46] Micev, Mihailo, Martin Čalasan, Ziad M. Ali, Hany M. Hasanien, and Shady HE Abdel Aleem. "Optimal design of automatic voltage regulation controller using hybrid simulated annealing–Manta ray foraging optimization algorithm." *Ain Shams Engineering Journal* 12, no. 1 (2021): 641-657.

- [47] Elattar, Ehab E., Abdullah M. Shaheen, Abdallah M. Elsayed, and Ragab A. El-Sehiemy. "Optimal power flow with emerged technologies of voltage source converter stations in meshed power systems." *IEEE Access* 8 (2020): 166963-166979.
- [48] Feng, Jiaying, Xiaoguang Luo, Mingzhe Gao, Adnan Abbas, Yi-Peng Xu, and Somayeh Pouramini. "Minimization of energy consumption by building shape optimization using an improved Manta-Ray Foraging Optimization algorithm." *Energy Reports* 7 (2021): 1068-1078.
- [49] Abd Elaziz, Mohamed, Dalia Yousri, Mohammed AA Al-qaness, Amr M. AbdelAty, Ahmed G. Radwan, and Ahmed A. Ewees. "A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation." *Engineering Applications of Artificial Intelligence* 98 (2021): 104105.
- [50] Hassan, Mohamed H., Essam H. Houssein, Mohamed A. Mahdy, and Salah Kamel. "An improved manta ray foraging optimizer for cost-effective emission dispatch problems." *Engineering Applications of Artificial Intelligence* 100 (2021): 104155.
- [51] Tang, Andi, Huan Zhou, Tong Han, and Lei Xie. "A modified manta ray foraging optimization for global optimization problems." *IEEE Access* 9 (2021): 128702-128721.
- [52] Breijyeh, Zeinab, and Rafik Karaman. "Comprehensive review on Alzheimer's disease: Causes and treatment." *Molecules* 25, no. 24 (2020): 5789.
- [53] Long, Justin M., and David M. Holtzman. "Alzheimer disease: an update on pathobiology and treatment strategies." *Cell* 179, no. 2 (2019): 312-339.
- [54] Piaceri, Irene, Benedetta Nacmias, and Sandro Sorbi. "Genetics of familial and sporadic Alzheimer's disease." *Frontiers in Bioscience-Elite* 5, no. 1 (2013): 167-177.
- [55] Clancy, Suzanne, and William Brown. "Translation: DNA to mRNA to protein." *Nature Education* 1, no. 1 (2008): 101.
- [56] Bar-Joseph, Ziv, Anthony Gitter, and Itamar Simon. "Studying and modelling dynamic biological processes using time-series gene expression data." *Nature Reviews Genetics* 13, no. 8 (2012): 552-564.

- [57] Uddin, Md S., Sharifa Hasana, Md Farhad Hossain, Md Siddiquil Islam, Tapan Behl, Asma Perveen, Abdul Hafeez, and Ghulam Md Ashraf. "Molecular genetics of early- and late-onset Alzheimer's disease." *Current Gene Therapy* 21, no. 1 (2021): 43-52.
- [58] Simek, Krzysztof, Krzysztof Fajarewicz, Andrzej Świerniak, Marek Kimmel, Barbara Jarząb, Małgorzata Wiench, and Joanna Rzeszowska. "Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data." *Engineering Applications of Artificial Intelligence* 17, no. 4 (2004): 417-427.
- [59] Strittmatter, Warren J., and Allen D. Roses. "Apolipoprotein E and Alzheimer disease." *Proceedings of the National Academy of Sciences* 92, no. 11 (1995): 4725-4727.
- [60] LaFontaine, Mike, and Medhane Cumbay. "Focus: the aging brain: amyloid-beta Alzheimer targets—protein processing, lipid rafts, and amyloid-beta pores." *The Yale journal of biology and medicine* 89, no. 1 (2016): 5.
- [61] Cheng, Yubo, Luke Saville, Babita Gollen, Ana Alvarez Veronesi, Majid Mohajerani, Jeffrey T. Joseph, and Athanasios Zovoilis. "Increased Alu RNA processing in Alzheimer brains is linked to gene expression changes." *EMBO reports* 22, no. 5 (2021): 52255.
- [62] Díaz-Uriarte, Ramón, and Sara Alvarez de Andrés. "Gene selection and classification of microarray data using random forest." *BMC bioinformatics* 7 (2006): 1-13.
- [63] Yang, Kun, Zhipeng Cai, Jianzhong Li, and Guohui Lin. "A stable gene selection in microarray data analysis." *BMC bioinformatics* 7, no. 1 (2006): 1-16.
- [64] Wang, Yu, Igor V. Tetko, Mark A. Hall, Eibe Frank, Axel Facius, Klaus FX Mayer, and Hans W. Mewes. "Gene selection from microarray data for cancer classification—a machine learning approach." *Computational biology and chemistry* 29, no. 1 (2005): 37-46.
- [65] Shukla, Alok Kumar, Pradeep Singh, and Manu Vardhan. "A two-stage gene selection method for biomarker discovery from microarray data for cancer classification." *Chemometrics and Intelligent Laboratory Systems* 183 (2018): 47-58.

- [66] Mutch, David M., Alvin Berger, Robert Mansourian, Andreas Rytz, and Matthew-Alan Roberts. "The limit fold change model: a practical approach for selecting differentially expressed genes from microarray data." *BMC bioinformatics* 3 (2002): 1-11.
- [67] Hernandez Hernandez, Jose Crispin, Béatrice Duval, and Jin-Kao Hao. "A genetic embedded approach for gene selection and classification of microarray data." In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 5th European Conference, EvoBIO 2007, Valencia, Spain, April 11-13, 2007. Proceedings 5*, pp. 90-101. Springer Berlin Heidelberg, 2007.
- [68] Kumar, Vipin, and Sonajharia Minz. "Feature selection: a literature review." *SmartCR* 4, no. 3 (2014): 211-229.
- [18] Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), pp.16-28.
- [69] Wah, Yap Bee, Nurain Ibrahim, Hamzah Abdul Hamid, Shuzlina Abdul-Rahman, and Simon Fong. "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy." *Pertanika Journal of Science & Technology* 26, no. 1 (2018).
- [70] Gnana, D. Asir Antony, S. Appavu Alias Balamurugan, and E. Jebamalar Leavline. "Literature review on feature selection methods for high-dimensional data." *International Journal of Computer Applications* 136, no. 1 (2016): 9-17.
- [71] Kohavi, Ron, and Dan Sommerfield. "Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology." In *KDD*, vol. 95, pp. 192-197. 1995.
- [72] Chen, Gang, and Jin Chen. "A novel wrapper method for feature selection and its applications." *Neurocomputing* 159 (2015): 219-226.
- [73] Maxwell, Aaron E., Timothy A. Warner, and Fang Fang. "Implementation of machine-learning classification in remote sensing: An applied review." *International journal of remote sensing* 39, no. 9 (2018): 2784-2817.

- [74] Soofi, Aized Amin, and Arshad Awan. "Classification techniques in machine learning: applications and issues." *J. Basic Appl. Sci* 13, no. 1 (2017): 459-465.
- [75] Nasteski, Vladimir. "An overview of the supervised machine learning methods." *Horizons. b* 4, no. 51-62 (2017): 56.
- [76] Bhavsar, Himani, and Mahesh H. Panchal. "A review on support vector machine for data classification." *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 1, no. 10 (2012): 185-189.
- [77] Cervantes, Jair, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. "A comprehensive survey on support vector machine classification: Applications, challenges and trends." *Neurocomputing* 408 (2020): 189-215.
- [78] Özöğür-Akyüz, Süreyya, Zakria Hussain, and John Shawe-Taylor. "Prediction with the SVM using test point margins." In *Data Mining: Special Issue in Annals of Information Systems*, pp. 147-158. Boston, MA: Springer US, 2009.
- [79] Liu, Quanzhong, Chihau Chen, Yang Zhang, and Zhengguo Hu. "Feature selection for support vector machines with RBF kernel." *Artificial Intelligence Review* 36, no. 2 (2011): 99-115.
- [80] Reeves, D. M., and G. M. Jacyna. "Support vector machine regularization." *Wiley Interdisciplinary Reviews: Computational Statistics* 3, no. 3 (2011): 204-215.
- [81] Chen, Di-Rong, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. "Support vector machine soft margin classifiers: error analysis." *The Journal of Machine Learning Research* 5 (2004): 1143-1175.
- [82] Mathur, Ajay, and Giles M. Foody. "Multiclass and binary SVM classification: Implications for training and classification users." *IEEE Geoscience and remote sensing letters* 5, no. 2 (2008): 241-245.
- [83] Cherkassky, Vladimir, and Yunqian Ma. "Practical selection of SVM parameters and noise estimation for SVM regression." *Neural networks* 17, no. 1 (2004): 113-126.
- [84] Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction." *Shanghai archives of psychiatry* 27, no. 2 (2015): 130.

- [85] Rigatti, Steven J. "Random forest." *Journal of Insurance Medicine* 47, no. 1 (2017): 31-39.
- [86] Speiser, Jaime Lynn, Michael E. Miller, Janet Tooze, and Edward Ip. "A comparison of random forest variable selection methods for classification prediction modeling." *Expert systems with applications* 134 (2019): 93-101.
- [87] Segal, Mark R. "Machine learning benchmarks and random forest regression." (2004).
- [88] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794. 2016.
- [89] Nalluri, Mounika, Mounika Pentela, and Nageswara Rao Eluri. "A Scalable Tree Boosting System: XG Boost." *Int. J. Res. Stud. Sci. Eng. Technol* 7, no. 12 (2020): 36-51.
- [40] Sagi, Omer, and Lior Rokach. "Approximating XGBoost with an interpretable decision tree." *Information sciences* 572 (2021): 522-542.
- [91] Ramraj, Santhanam, Nishant Uzir, R. Sunil, and Shatadeep Banerjee. "Experimenting XGBoost algorithm for prediction and classification of different datasets." *International Journal of Control Theory and Applications* 9, no. 40 (2016): 651-662.
- [92] Mitchell, Rory, and Eibe Frank. "Accelerating the XGBoost algorithm using GPU computing." *PeerJ Computer Science* 3 (2017): 127.
- [93] Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, and Tianyi Zhou. "Xgboost: extreme gradient boosting." *R package version 0.4-2* 1, no. 4 (2015): 1-4.
- [94] Rish, Irina. "An empirical study of the naive Bayes classifier." In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41-46. 2001.
- [95] Webb, Geoffrey I., Eamonn Keogh, and Risto Miikkulainen. "Naïve Bayes." *Encyclopedia of machine learning* 15, no. 1 (2010): 713-714.

- [96] Flach, Peter. "Performance evaluation in machine learning: the good, the bad, the ugly, and the way forward." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 9808-9814. 2019.
- [97] Dalianis, Hercules, and Hercules Dalianis. "Evaluation metrics and evaluation." *Clinical Text Mining: secondary use of electronic patient records* (2018): 45-53.
- [98] Calders, Toon, and Szymon Jaroszewicz. "Efficient AUC optimization for classification." In *European conference on principles of data mining and knowledge discovery*, pp. 42-53. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [99] Nahm, Francis Sahngun. "Receiver operating characteristic curve: overview and practical use for clinicians." *Korean journal of anesthesiology* 75, no. 1 (2022): 25.
- [100] Chong, Edwin KP, and Stanislaw H. Żak. *An introduction to optimization*. Vol. 75. John Wiley & Sons, 2013.
- [101] Yang, Xin-She. *Nature-inspired optimization algorithms*. Academic Press, 2020.
- [102] Zhao, Weiguo, Zhenxing Zhang, and Liying Wang. "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications." *Engineering Applications of Artificial Intelligence* 87 (2020): 103300.
- [103] Tang, Andi, Huan Zhou, Tong Han, and Lei Xie. "A modified manta ray foraging optimization for global optimization problems." *IEEE Access* 9 (2021): 128702-128721.
- [104] Tang, Andi, Huan Zhou, Tong Han, and Lei Xie. "A modified manta ray foraging optimization for global optimization problems." *IEEE Access* 9 (2021): 128702-128721.
- [105] Hassan, Mohamed H., Essam H. Houssein, Mohamed A. Mahdy, and Salah Kamel. "An improved manta ray foraging optimizer for cost-effective emission dispatch problems." *Engineering Applications of Artificial Intelligence* 100 (2021): 104155.
- [106] Asghari, Mohammad, Amir M. Fathollahi-Fard, S. M. J. Mirzapour Al-E-Hashem, and Maxim A. Dulebenets. "Transformation and linearization techniques in optimization: A state-of-the-art survey." *Mathematics* 10, no. 2 (2022): 283.

- [107] Beheshti, Zahra. "A time-varying mirrored S-shaped transfer function for binary particle swarm optimization." *Information Sciences* 512 (2020): 1503-1542.
- [108] Zhang, Dong-Yang, Jian-Hua Liu, Lei Jiang, Guan-Nan Bu, Ren-Yuan Hu, and Yi-Xuan Luo. "The improvement of v-shaped transfer function of binary particle swarm optimization." In *Advances in Swarm Intelligence: 11th International Conference, ICSI 2020, Belgrade, Serbia, July 14–20, 2020, Proceedings 11*, pp. 202-211. Springer International Publishing, 2020.
- [109] Suto, Jozsef, Stefan Oniga, and Petrica Pop Sitar. "Comparison of wrapper and filter feature selection algorithms on human activity recognition." In *2016 6th international conference on computers communications and control (ICCCC)*, pp. 124-129. IEEE, 2016.
- [110] Venkatesh, B., and J. Anuradha. "A review of feature selection and its methods." *Cybernetics and information technologies* 19, no. 1 (2019): 3-26.
- [111] Chen, Chih-Wen, Yi-Hong Tsai, Fang-Rong Chang, and Wei-Chao Lin. "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection"
- [112] National Library of Medicine, <<https://www.ncbi.nlm.nih.gov/>>, accessed in 02/05/2024.
- [113] Panadas – Python Data Analysis Library, < <https://pandas.pydata.org/>>, accessed in 02/05/2024.
- [114] Brezočnik, Lucija, Iztok Fister Jr, and Vili Podgorelec. "Swarm intelligence algorithms for feature selection: a review." *Applied Sciences* 8, no. 9 (2018): 1521.
- [115] Naik, Akshata K., Venkatanareshbabu Kuppili, and Damodar Reddy Edla. "Efficient feature selection using one-pass generalized classifier neural network and binary bat algorithm with a novel fitness function." *Soft Computing* 24, no. 6 (2020): 4575-4587.
- [116] Cinar, Ahmet Cevahir. "A comprehensive comparison of accuracy-based fitness functions of metaheuristics for feature selection." *Soft Computing* 27, no. 13 (2023): 8931-8958.

- [117] Tilahun, Surafel Lulseged, and Jean Medard T. Ngnotchouye. "Firefly algorithm for optimization problems with non-continuous variables: a review and analysis." *arXiv preprint arXiv:1602.07884* (2016).
- [118] Ghosh, Kushal Kanti, Ritam Guha, Suman Kumar Bera, Neeraj Kumar, and Ram Sarkar. "S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem." *Neural Computing and Applications* 33, no. 17 (2021): 11027-11041.
- [119] Mirjalili, Seyedali, and Andrew Lewis. "S-shaped versus V-shaped transfer functions for binary particle swarm optimization." *Swarm and Evolutionary Computation* 9 (2013): 1-14.
- [120] GSE1297, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE1297> >, accessed in 02/05/2024.
- [121] GSE5281, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5281> >, accessed in 02/05/2024.
- [122] GSE33000, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE33000> >, accessed in 02/05/2024.
- [123] GSE44768, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE44768> >, accessed in 03/05/2024.
- [124] GSE44770, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE44770> >, accessed in 03/05/2024.
- [125] GSE132903, < <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132903> >, accessed in 02/05/2024.
- [126] Tang, Andi, Huan Zhou, Tong Han, and Lei Xie. "A modified manta ray foraging optimization for global optimization problems." *IEEE Access* 9 (2021): 128702-128721.
- [127] Xu, Shenheng, and Yahya Rahmat-Samii. "Boundary conditions in particle swarm optimization revisited." *IEEE Transactions on Antennas and Propagation* 55, no. 3 (2007): 760-765.

- [128] Huang, Tony, and Ananda Sanagavarapu Mohan. "A hybrid boundary condition for robust particle swarm optimization." *IEEE antennas and wireless propagation letters* 4 (2005): 112-117.

