UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

JONAS DA SILVEIRA BOHRER

Enhancing Classification with Hybrid Feature Selection: A Multi-Objective Genetic Algorithm for High-Dimensional Data

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Márcio Dorn

Porto Alegre
June 2024

CIP — CATALOGING-IN-PUBLICATION

Bohrer, Jonas da Silveira

Enhancing Classification with Hybrid Feature Selection: A Multi-Objective Genetic Algorithm for High-Dimensional Data / Jonas da Silveira Bohrer. – Porto Alegre: PPGC da UFRGS, 2024.

91 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2024. Advisor: Márcio Dorn.

1. Feature selection. 2. Dimensionality reduction. 3. Genetic algorithm. 4. High-dimensional. 5. Multi-objective. 6. Optimization. 7. Classification. 8. Machine learning. I. Dorn, Márcio. II. Título.

AGRADECIMENTOS

Agradeço profundamente aos meus amigos e família que me acompanharam e incentivaram nesse processo de mestrado. Agradeço por acreditarem em mim e por me darem alguns empurrõezinhos quando necessário. Agradeço ao meu orientador Márcio por toda a ajuda e direcionamento e às demais pessoas do laboratório de Bioinformática Estrutural e Biologia Computacional cujos trabalhos antecederam esse e me ajudaram de diferentes formas.

ABSTRACT

Feature selection is a fundamental step in machine learning, serving to reduce dataset redundancy, accelerate training speed, and improve model quality. This is particularly crucial in high-dimensional datasets, where the excess of features presents challenges for pattern recognition and data analysis. Recent methods proposed for high-dimensional data are often tailored for specific domains, leaving a lack of consensus on a universally recommended solution for general use cases. This paper proposes a hybrid feature selection approach using a multi-objective genetic algorithm to enhance classification performance and reduce dimensionality across diverse classification tasks. The proposed approach narrows the search space of possible relevant features by exploring the combined outputs of classical feature selection methods with novel genetic algorithm operators. This enables the evolution of combined solutions potentially not explored by the original methods, generating optimized feature sets in a process that adapts to different data conditions. Experimental results demonstrate the effectiveness of the proposed method in high-dimensional use cases, offering improved classification performance with reduced feature sets. In summary, our hybrid method offers a promising solution for addressing the challenges of high-dimensional datasets by enhancing classification performance in varying domains and data conditions.

Keywords: Feature selection. dimensionality reduction. genetic algorithm. high-dimensional. multi-objective. optimization. classification. machine learning.

Aprimorando a Classificação com Seleção Híbrida de Variáveis: um Algoritmo Genético Multi-Objetivo para Dados de Alta Dimensionalidade

RESUMO

A seleção de variáveis é um passo fundamental no aprendizado de máquina, servindo para reduzir a redundância do conjunto de dados, acelerar a velocidade de treinamento e melhorar a qualidade de modelos. Isto é particularmente crucial em conjuntos de dados de alta dimensionalidade, onde o excesso de variáveis representa desafios para tarefas de reconhecimento de padrões e análise de dados. Os métodos recentes propostos para dados de alta dimensionalidade são frequentemente desenvolvidos para domínios específicos, gerando uma falta de consenso sobre uma solução universalmente recomendada para casos de uso gerais. Este artigo propõe uma abordagem híbrida de seleção de variáveis usando um algoritmo genético multiobjetivo para melhorar o desempenho da classificação e reduzir a dimensionalidade em diversas tarefas de classificação. A abordagem proposta restringe o espaço de busca de possíveis variáveis relevantes através da exploração dos resultados combinados de métodos clássicos de seleção de variáveis através de novos operadores de algoritmo genético. Isto permite a evolução de soluções combinadas potencialmente não exploradas pelos métodos originais, gerando conjuntos de variáveis otimizados em um processo que se adapta a diferentes condições de dados. Os resultados experimentais demonstram a eficácia do método proposto em casos de uso de alta dimensionalidade, oferecendo melhor desempenho de classificação com conjuntos de variáveis reduzidos. Em resumo, o método híbrido proposto oferece uma solução promissora para lidar com os desafios de conjuntos de dados de alta dimensionalidade, melhorando o desempenho da classificação em diversos domínios e condições de dados.

Palavras-chave: redução de dimensionalidade, seleção de atributos, seleção de variáveis, algoritmo genético, multi-objetivo, alta dimensionalidade, otimização, classificação, aprendizado de máquina.

LIST OF ABBREVIATIONS AND ACRONYMS

CV	*Cross-validation*
GA	*Genetic Algorithm*
MOGA	*Multi-objective Genetic Algorithm*
NSGA	*Non-dominated Sorting Genetic Algorithm*
RF	*Random Forest*
SNP	*Single Nucleotide Polymorphism*
SVM	*Support Vector Machine*

LIST OF FIGURES

Figure 2.1 A simple decision tree based on binary target variable Y. 26
Figure 2.2 A confusion matrix used for summarizing classifications. 32
Figure 2.3 A representation of the crowding-distance calculation. 40
Figure 2.4 A representation of the NSGA-II algorithm. 42

Figure 4.1 A simplified visual depiction of the multi-objective algorithm. 51
Figure 4.2 A population of solutions in the proposed genetic algorithm. 52
Figure 4.3 A visual depiction of the multi-objective algorithm. 53

Figure 5.1 A depiction of the experiment plan for the proposed method, divided in two phases. 64

Figure 6.1 Average macro F1-Score performance with varying numbers of features for CuMiDa Leukemia dataset. 71
Figure 6.2 Average macro F1-Score performance with varying numbers of features for CuMiDa Breast Cancer dataset. 72
Figure 6.3 Average macro F1-Score performance with varying numbers of features for Eye Color SNPs dataset 73
Figure 6.4 Average macro F1-Score performance with varying numbers of features for p53 Mutants dataset 74
Figure 6.5 Average macro F1-Score performance with varying numbers of features for Arrhythmia dataset. 76

LIST OF TABLES

Table 3.1 Summary of notable related works using genetic algorithms for feature selection. 45

Table 5.1 Dataset details and label distributions 61
Table 5.2 List of feature selection methods used in the baseline experiments....65
Table 5.3 Baseline experiments parameters 66
Table 5.4 Multi-objective genetic algorithm optimization experiments parameters.....67

Table 6.1 CuMiDa Leukemia dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.....71
Table 6.2 CuMiDa Breast Cancer dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.....72
Table 6.3 Eye Color SNPs dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold. 73
Table 6.4 p53 Mutants dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold. 74
Table 6.5 Arrhythmia dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold. 75
Table 6.6 Average execution times across 10 executions for each dataset....77

LIST OF ALGORITHMS

1 Relief main loop algorithm. Source: (KIRA; RENDELL, 1992).....24
2 Basic genetic algorithm structure.....37
3 Fast non-dominated sorting approach. Source: (DEB et al., 2002).....39
4 NSGA-II main loop algorithm. Source: (DEB et al., 2002).....42
5 The sampling process. A total of n_s samples or $individuals$ are generated, each representing a set of n_f features between min_f and max_f features. Features are selected from the pool of feature sets according to probability $selection\_chance_f$, described in equation 4.1. 56
6 The mutation process. A group of $mutation\_candidates$ is mutated by activating previously inactive features and disabling previously enabled features. The features chosen for activation are sourced from the pool of feature sets. 57

CONTENTS

1 INTRODUCTION 12
1.1 Motivation 14
1.2 Objectives 15
1.3 Dissertation Structure 16
2 THEORETICAL BACKGROUND 17
2.1 Feature Selection 17
2.1.1 Categorization of Supervised Feature Selection methods 18
2.1.2 Feature Selection methods used in this work 20
2.1.2.1 ANOVA F-Test 20
2.1.2.2 Kruskal-Wallis Test 21
2.1.2.3 Mutual Information 22
2.1.2.4 mRMR 22
2.1.2.5 Relief-F 24
2.1.2.6 Decision Tree 25
2.1.2.7 Lasso 27
2.1.2.8 Linear SVM 28
2.1.2.9 Random Forest 29
2.2 Feature set evaluation 30
2.2.1 Machine learning and feature selection 30
2.2.2 Classification metrics 32
2.3 Genetic Algorithms 34
2.3.1 The basic genetic algorithm 35
2.3.2 Multi-objective genetic algorithms 36
2.3.3 NSGA-II: the non-dominated sorting genetic algorithm 38
2.3.3.1 Non-dominated sorting 38
2.3.3.2 Crowding-distance 40
2.3.3.3 Main loop 41
2.4 Chapter summary 43
3 RELATED WORK 44
3.1 Chapter summary 50
4 PROPOSED METHOD AND IMPLEMENTATION 51
4.1 Genetic algorithm structure 51
4.2 Feature importances 54
4.3 Pool of Feature Subsets 54
4.4 Sampling operator 55
4.5 Mutation operator 56
4.6 Fitness function 58
4.7 Chapter summary 59
5 EXPERIMENTS 60
5.1 Datasets 60
5.1.1 CuMiDa - Leukemia and Breast Cancer datasets 60
5.1.2 Eye Color SNPs dataset 62
5.1.3 UCI Database datasets 62
5.2 Experiment design 63
5.2.1 Baseline experiments 64
5.2.2 Multi-objective genetic algorithm optimization experiments 66
5.3 Chapter summary 68

6 RESULTS AND DISCUSSION....69
6.1 Classification performances....69
6.2 Execution times....75
6.3 Chapter summary....78
7 CONCLUSION....79
8 PUBLICATIONS....81
8.1 Publications in Journals....81
REFERENCES....82
APPENDIX A — RESUMO ESTENDIDO EM PORTUGUÊS....89

1 INTRODUCTION

In the age of big data, datasets frequently suffer from the "curse of dimensionality", wherein data comprises so many dimensions that it becomes extremely noisy and difficult to analyze with traditional statistical methods. Advances in technology enable better collection and storage of data from more and more sources every day, with increasingly detailed descriptions. This tends to make high dimensionality a permanent issue in data analytics (ZHAI; ONG; TSANG, 2014). In such scenarios, where human abstractions from noisy or voluminous data become limited or impractical to achieve, machine learning techniques represent a solid alternative to enable the extraction of information. Still, even with their typical capability of exploring non-obvious patterns in large datasets, machine learning, and pattern recognition techniques are often negatively affected by high dimensionality (GAO et al., 2017).

Dimensionality reduction is discussed in high-dimensional and voluminous data cases, where feature selection or feature extraction are typically proposed to simplify data representations while avoiding loss of information. Both techniques "have the advantages of improving learning performance, increasing computational efficiency, decreasing memory usage, and building better generalization models" (LI et al., 2017). While feature extraction achieves this by transforming the original feature set into a new, abstracted, and reduced set of features, feature selection preserves the original state of features. Instead, it selects the important ones and removes the unimportant or detractor ones. When dimensionality reduction is performed with interpretability and explainability in mind, feature selection naturally becomes the recommended option (LI et al., 2017). Examples of applications with particular emphasis on identifying essential features include studies of cancer and other diseases (HAMBALI; OLADELE; ADEWOLE, 2020; ALHENAWI et al., 2022; SINGH; SIVABALAKRISHNAN, 2015; GRISCI; FELTES; DORN, 2019), genome-wide association studies (TADIST et al., 2019; HEINRICH et al., 2023; PUDJIHARTONO et al., 2022), and multiple other bioinformatics and medical fields (SAEYS; INZA; LARRANAGA, 2007; REMESEIRO; BOLON-CANEDO, 2019).

In feature selection, many methods have been proposed to provide ideal selections of features (or dimensions) for data originating from different domains and having different characteristics. Nevertheless, no single solution performs better than the others in all use cases. Various methods are available under different specializations, with well-established categories in the literature from both a methodology perspective and a data

perspective (LI et al., 2017).

This lack of consensus is especially evident in challenging combinatorial problems, such as cancer and disease research and genome-wide studies. In these fields, measuring the importance of features — such as biomarkers, DNA sequences, characteristics of images, and other diverse measurements — is crucial for understanding biological or medical problems. However, the data often consists of small numbers of samples containing up to thousands or millions of features. For this type of application, research has been conducted to propose new feature selection methods or to evaluate widespread strategies. Results naturally tend to show different recommendations for different scenarios over the years, with no firm consensus on methods, even in recent years (AKAY, 2009; ABEEL et al., 2010; NGUYEN; WANG; NGUYEN, 2013; SINGH; SIVABALAKRISHNAN, 2015; HAMBALI; OLADELE; ADEWOLE, 2020; ALHENAWI et al., 2022). This issue is not limited to the life sciences domain, as similar feature selection applications aim to understand the importance of features in varied problems in engineering domains and others (PIRI et al., 2023; FADAEE; RADZI, 2012).

Feature selection in high-dimensional data poses a significant combinatorial search challenge, as evaluating the impact of every possible combination of features is exceedingly complex (FERRI et al., 1994). Genetic algorithms (GAs), a popular branch of methods for solving combinatorial problems, offer a solution. GA is a metaheuristic capable of approximating the optimal solution within a reasonable execution time. They belong to a broader group of biology-inspired algorithms known as evolutionary algorithms, which share three main characteristics (YU; GEN, 2010): they are population-based, meaning they improve a population of solutions over time, referred to as individuals; they are fitness-oriented, evaluating solutions based on fitness criteria and favoring the fittest individuals; and they are variation-driven, exploring the solution search space through operations that modify individuals, akin to genetic alterations.

For feature selection problems, GAs offer strategies that can be used to explore and optimize subsets of features in ways that do not require evaluating every single scenario but instead generate random or semi-random solutions and iteratively adapt and re-use well-performing solutions to improve fitness. Handling high-dimensional data is a common theme of research for evolutionary solutions (PIRI et al., 2023).

Many variations of GAs have been proposed over the last decades with successful applications in multiple fields (KATOCH; CHAUHAN; KUMAR, 2021). This work will focus on using a multi-objective genetic algorithm (ZHOU et al., 2011), a predomi-

nant class among GAs, for the feature selection problem. This class of genetic algorithms works by optimizing the solutions for a variable number of conflicting objectives, whereas the traditional GA would have a single-objective fitness function. Given the multiple fitness objectives, multi-objective GAs tend to present not one individual as the best solution for a problem but instead provide a Pareto set of dominant individuals that best optimize the multiple fitness objectives in different degrees, usually following the concept of Pareto domination established by the notorious NSGA-II (DEB et al., 2002).

NSGA-II is easily one of the most popular multi-objective genetic algorithms in the literature, having inspired multiple other approaches and remaining a solid option in multi-objective GAs. The structure of NSGA-II and some of its core characteristics — such as the Pareto domination and crowding distance mechanisms — and other classic GA operators are adapted within the proposed method of this work to solve the conflicting problems of performing dimensionality reduction while maintaining features that enable effective data separation.

1.1 Motivation

Given the challenge of selecting important features in high-dimensional data scenarios, either for improving machine learning classification performance or for analysis, research or interpretability purposes, feature selection methods are capable of providing reliable results even with little domain knowledge. Still, different methods tend to have very different performances according to the characteristics of the data, and no consensus exists on a recommended generalist approach for data of different domains, especially in real-life datasets. This work proposes a versatile method to yield robust feature selections from high-dimensional datasets. It employs a multi-objective genetic algorithm that evaluates and combines the individual strengths of various existing feature selection methods, adapting the feature selection process to diverse data types. The heuristic search strategy leverages the knowledge gained from applying these methods to the data to narrow the search space of solutions and to consolidate subsets of features that consistently outperform the original methods based on classification metrics while ensuring dimensionality reduction. Beyond merely enhancing metrics in specific scenarios, the proposed heuristic aims to harness the unique advantages of different methods to deliver reliable ensemble solutions to feature selection challenges across various domains and data patterns. Moreover, it ensures the exploration of intermediate solutions that combine those proposed by

the initial methods. The following chapters will elucidate how the method optimizes and stabilizes feature selections by leveraging feature importance information and employing modified operators within a multi-objective genetic algorithm based on NSGA-II.

1.2 Objectives

The objective of this work is to propose, describe, and implement a feature selection method based on a multi-objective genetic algorithm, adapted to combine the outputs of different feature selection methods to provide robust selections of features in different settings of high-dimensional data, based on the objectives of classification performance improvement and dimensionality reduction. To do so, this work aims to:

1. Describe the literature background related to feature selection in high-dimensional data, as well as the necessary background of machine learning literature to describe the usage of classification models and their metrics;
2. Describe the main characteristics of genetic algorithms and especially multi-objective genetic algorithms, explaining their usage for feature selection and exploring recent related work in the literature;
3. Describe the proposed method, the usage of information acquired from the execution of other feature selection methods, and the custom operators developed to process this information in the proposed multi-objective GA, as well as the fitness function used to evaluate feature sets;
4. Describe the experiments designed to evaluate the method, generating comparable experiments with other methods and applying them to multiple classification datasets, exploring applications of the methods in different data settings;
5. Describe and discuss the results obtained in comparison to other existing methods, raising the benefits and disadvantages of the proposed method.

With this, the expectation is that the method proves itself capable of outperforming the original feature selection methods in the proposed settings, reinforcing that genetic algorithms are a viable and flexible option to optimize the identification and selection process of important features in high-dimensional data. Given the fact that most feature selection methods proposed in the literature are developed and evaluated for specific scenarios, and not as generalist solutions for the feature selection problem, the method should be evaluated in use cases with different data characteristics for the classification problem.

1.3 Dissertation Structure

The next chapters of this dissertation are divided into the following structure:

- **Chapter 2: Theoretical background**. A review of the theoretical background for feature selection, feature set evaluation, and genetic algorithms is presented. Existing methods used either as a base for the proposed solution or as part of the experimentation process are described in this chapter.
- **Chapter 3: Related work**. A review of the literature surrounding the combination of feature selection and genetic algorithms is presented, with an exploration of similar works produced in recent years, comparing the types of feature selection approaches, fields of application, and target datasets. Additionally, the usage of single or multiple objectives in the proposed genetic algorithms is also analyzed, and an overview of possible improvements is highlighted.
- **Chapter 4: Proposed method and implementation**. The proposition and details of the implementation are provided in this chapter, highlighting the different operators created as part of a multi-objective genetic algorithm solution for the optimization of feature sets.
- **Chapter 5: Experiments**. A description of the experiments designed to evaluate the proposed method are described in this chapter, mentioning the feature selection methods used as a base for optimization, the datasets chosen to undergo feature selection, and the parameters used during the multi-objective genetic algorithm optimization process, as well as the methods and metrics used to evaluate the quality of the selected feature sets.
- **Chapter 6: Results and discussion**. The results from experimentation are summarized in this chapter, providing metrics to illustrate the gains in classification performance in comparison to the base methods applied before in the optimization, along with a discussion of such results and a summary of execution times observed in the process.
- **Chapter 7: Conclusion**. In this chapter, the final considerations for this work are presented, reflecting on the work objectives, proposed solution and the results achieved, as well as opportunities for future work.

2 THEORETICAL BACKGROUND

In this chapter, the theoretical background for the proposed work is presented in three sections, providing the basis of knowledge required in the topics of feature selection, the usage of machine learning metrics in the evaluation of feature sets resulting from feature selection, and genetic algorithms, which are the basis of the proposed method. The sections also explore the relationship between the three topics and explain some of the methods and metrics used in the implementation and evaluation of results in the next chapters.

2.1 Feature Selection

When data has a considerably large number of dimensions, handling it becomes more and more complex due to increases in the required computational processing time, memory consumption, and general interpretability impairments generated by the excess of noisy information. This set of difficulties is known as the "curse of dimensionality" (VERLEYSEN; FRANÇOIS, 2005). Additionally, when high-dimensional datasets possess a small number of samples in comparison to their large number of features, they are affected by the "*large p, small n*" problem, leading to more trouble in efficiently extracting useful information from this data, either via human analysis or learning algorithms. In this sense, dimensionality reduction surges as a necessary pre-processing task when dealing with such data conditions.

Dimensionality reduction is the process of lowering the number of features of the data. This is not only important from the computational and the classification perspective, but also from the point of view of extracting useful information, necessary in fields such as biological or medical research. The main group of algorithms for dimensionality reduction is feature extraction, a set of methods that transforms the original feature space into a different space with a new set of axes by combining its features and finding the ones that most preserve the original information (VARSHAVSKY et al., 2006). This new feature space often has better discriminatory power, but the extracted features lack real-world meaning for better interpretation (ALELYANI; TANG; LIU, 2013; KRIZEK, 2008; ANG et al., 2016).

While feature extraction can be useful from the computational view, its lack of interpretability leaves it with little use for the discovery of informative features. However,

a subgroup of dimensionality reduction techniques, called feature selection (FS), solves this problem by choosing small subsets of features instead of combining them, usually through the removal of irrelevant, redundant, or noisy features. This is better suited for biological data as it leads to better performance and model interpretability (MIAO; NIU, 2016).

The most basic categorization of feature selection approaches refers to the presence or absence of values that can be used as a reference for the supervision of data separation tasks. In other words, the presence of classes, numerical values or other markers in the target data that specify a definitive distinction of data points into subgroups or quantitative values. In this sense, methods are typically categorized into supervised, unsupervised, and semi-supervised, as in the classical division of machine learning tasks (LI et al., 2017).

Supervised problems refer to data that contain one or more labels or targets capable of indicating distinct classes or measurements for each sample. Classification and regression are examples of supervised tasks, and in these cases, feature selection algorithms leverage previously known information to identify subsets of relevant features by employing different strategies such as removing noise or redundancy from the data or optimizing model metrics. Unsupervised problems refer to data that lacks such objective labels or targets, relying entirely on the features to arbitrarily separate the data. Clustering is the most common application for unsupervised problems, utilizing various metrics and approaches to evaluate the similarity or distance of samples and grouping them into informative clusters. Finally, semi-supervised problems mix the two scenarios, with portions of labeled and unlabeled data.

The focus of this work is feature selection for supervised tasks, especially classification. Under this category, the methods can be further divided according to the type of strategy used for feature selection. Filter methods, wrapper methods, and embedded methods are the most common classifications in the literature (LI et al., 2017), but recent specialized categories such as hybrid methods and ensemble methods (ANG et al., 2015) bring important distinctions to represent methods that combine different approaches.

2.1.1 Categorization of Supervised Feature Selection methods

According to the classification proposed by ANG et al. (ANG et al., 2015), feature selection methods for supervised learning tasks can be separated into 5 distinct types:

- *Filter* methods are most remarkable for being independent of a learning algorithm, and thus usually represent the fastest and most scalable methods among all feature selection categories. They benefit from less overfitting problems than methods that rely on learning algorithms, but may fail to identify important interactions between features that learning models are exceptional at uncovering.
- *Wrapper* methods use the performance of a learning algorithm to estimate the quality of the selected features during the iterations of its selection process. Because of this dependency, they typically require more resources than other methods, and may overfit the feature selection to the specific model used to evaluate the feature sets. Still, they represent a multivariate approach to feature selection, remarkably capable of identifying interactions between features.
- *Embedded* methods embed feature selection into the learning phase of a learning algorithm, turning it into a part of the classification model itself. This usually favors the computational resources required in the joined process of feature selection and learning phase of a model, because the interaction is optimized to reduce redundant evaluations of features that may occur when using wrapper methods. The downside is that feature selection, in this case, is also tied to the performance of a model, thus overfitting can happen. Still, similarly to wrapper methods, they possess the capability of easily uncovering hidden relationships between features.
- *Ensemble* methods aim to create feature sets from the aggregation of different subsets of features, usually generated from independent executions of a feature selector on different subsets of data. The aggregation generally brings higher stability to the final feature set, which increases its reliability for analysis and learning performance.
- *Hybrid* methods originate from the combination of two or more methods of the same, or different categories, usually aiming to combine the strengths of one category to avoid the problems in another.

Notoriously, most of the related work covered in chapter 3 is classified either as Wrapper, with one or more models being used to guide feature selection or the evolution of genetic algorithm populations, or as Hybrid approaches, using combinations of different categories of feature selection methods to achieve the end result of selection, as is the case for this work.

2.1.2 Feature Selection methods used in this work

Many classic feature selection methods could be mentioned as the reference for modern feature selection approaches, since different methods were historically developed for different applications and considering different computational limitations. Still, methods usually share fundamental similarities, such as mechanisms to rank or assign individual importance weights for features of a dataset (XU et al., 2019).

In this work, a list of feature selection methods is used to generate importances for features of a target dataset (as described in detail in the methodology in chapter 4), which is used as input for the optimization process of a genetic algorithm. The list of feature selection methods used by the proposed MOGA is a flexible choice, but the default selection of 8 methods used in the experiments for this work is elaborated in this section and referenced in section 5.2.

The list of methods is composed of 5 filter methods and 4 embedded methods, all selected for their fast performance and predominance in the fields of statistics and feature selection.

2.1.2.1 ANOVA F-Test

Analysis of variance, also known as ANOVA, is one of the most well-known statistical methods for hypothesis testing in statistics, that has multiple variations and a large history of applications in multiple fields of science (ST; WOLD et al., 1989) since its first proposition by Ronald Fisher (FISHER, 1928).

In feature selection, the original parametric method (also called one-way ANOVA) is classified as an extremely lightweight univariate filter method and considered most adequate for numerical variables with normal distributions. It provides, for each feature, a numerical F-value based on the evaluation of the relationship between each individual dimension and, in supervised problems, the classes existent in the target output variable. In summary, a feature with a high F-value is estimated to have a high impact on the prediction of the target output. The F-value can be calculated for a feature following equation 2.1, sourced from Kim et al. (KIM, 2017).

$$F = \frac{Intergroup\ variance}{Intragroup\ variance} = \frac{\sum_{i=1}^{K} n_i(\bar{Y}_i - \bar{Y})^2/(K-1)}{\sum_{i,j=1}^{n} (Y_{ij} - \bar{Y}_i)^2/(N-K)} \tag{2.1}$$

The F value for a variable is calculated as the ratio between intergroup variance

and intragroup variance, or, in classification terms, the ratio between the variance between classes and variance within a specific class. In the equation, K is the total number of groups, n_i is the number of observations in group i, $\bar{Y}_i$ is the mean of group i, $\bar{Y}$ is the overall mean among all groups, Y_{ij} is the jth observation of group i and N is the total number of all observations available in all groups. When calculating this equation for each variable in a dataset, it is possible to differentiate between the estimated importance of each of them in the class prediction task.

2.1.2.2 Kruskal-Wallis Test

The Kruskal-Wallis Test, similar to ANOVA, is also a well-known statistical method for hypothesis testing. It differs from ANOVA essentially in its application: whereas ANOVA is applied mostly to normally distributed data, the Kruskal-Wallis test is mostly recommended for non-normal distributions and data with diverging variances among groups. Kruskal-Wallis is also known as the non-parametric version of one-way ANOVA (MCKIGHT; NAJAB, 2010).

In feature selection, the Kruskal-Wallis method is classified as a univariate filter method with very lightweight execution, most adequate to non-normal distributions, and therefore typically more effective in handling outlier values than other methods such as ANOVA. The assumption being that the variable may follow a non-normal distribution, the method bases itself on comparing group rankings instead of group means, as ANOVA would (MCKIGHT; NAJAB, 2010). The test statistic H can be calculated for each variable following equation 2.2, sourced from Hecke et al. (HECKE, 2012).

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1),\ N = \sum_{i=1}^{k} n_i. \tag{2.2}$$

The H value represents the variance of ranks among groups for the analyzed variable in relation to the groups represented by the target output variable. Ranks are assigned to each data point according to the values in the dataset, with the smallest value receiving the smallest rank (rank 1), the second smallest receiving the second smallest rank (rank 2), and so on. In the case of ties, the rank of all tied elements is defined as the average of all ranks that would be assigned if elements were assigned in order (for example, tied elements in positions 5, 6, 7, and 8, would receive an average rank of 6.5.) (HECKE, 2012). R_i, then, represents the sum of the ranks in each group i, which contains n_i elements, out of the k groups being represented. Finally, N is used as a regularization parameter

calculated according to the number of elements in all groups.

2.1.2.3 Mutual Information

Mutual information is a statistical method that emerged from the field of information theory that employs the concepts of entropy and divergence to measure the statistical dependency between two variables. It has the advantage of measuring both linear and non-linear relationships between variables and is the basis of many methods derived from its initial definition (VERGARA; ESTÉVEZ, 2014).

For feature selection, the approach proposed by Ross et al. (ROSS, 2014) and used in this work is considered a univariate filter method, which uses a lightweight nearest-neighbors approach to estimate the mutual information (MI) value for a pair of variables. Higher MI values indicate a higher dependency between the two variables, which indicates, for supervised learning problems, a higher influence in the prediction of a target variable output. Equations 2.3 and 2.4 represent the calculation of the mutual information I between variables X and Y defined by Ross et al. (ROSS, 2014).

$$I_i = \psi(N) - \psi(N_{x_i}) + \psi(k) + \psi(m_i) \tag{2.3}$$

In equation 2.3, I_i is calculated for each data point i based on the nearest k neighbors (a user parameter choice, usually a small integer) among the N_{x_i} data points of the same y group (or class) as data point i. In this case, d is defined as the distance to this kth neighbor, and is used to calculate the number of m_i neighbors contained within this distance, from all groups. With these parameters, the digamma function ψ (ROSS, 2014) is used to compute I_i.

$$I(X, Y) = \langle I_i \rangle = \psi(N) - \langle \psi(N_{x_i}) \rangle + \psi(k) + \langle \psi(m_i) \rangle \tag{2.4}$$

Considering the individual mutual information I_i for each data point in X and Y, the estimated resulting mutual information I for X and Y is calculated in equation 2.4 by averaging I_i over all data points.

2.1.2.4 mRMR

Minimal-redundancy-maximal-relevance (mRMR) is a remarkable method developed by Peng et al. (PENG; LONG; DING, 2005) specifically for the task of supervised

feature selection and is proposed as a much simpler equivalent to the maximal statistical dependency criterion based on mutual information.

In the original work (PENG; LONG; DING, 2005), the method used for feature selection (or feature ranking) in mRMR is classified as a multivariate filter approach, and consists of a greedy search for features that results in the incremental inclusion of new features instead of providing an importance (or equivalent) value to all features as other previously mentioned filter methods like ANOVA, Kruskal-Wallis or mutual information do. This results in short execution times for small selections of features, while performing a multivariate analysis of variables, but can result in long execution times if large selections of features are expected. Equations 2.5, 2.6 and 2.7 represent the relations defined by Peng et al. (PENG; LONG; DING, 2005).

$$\max D(S,c),\ D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c). \tag{2.5}$$

The process starts with equation 2.5, which represents the maximization of $D(S,c)$, the relevance estimation for a set of features S, and a target label c. D represents the individual relevance estimation for each feature x_i from S in relation to a target class c, measuring how important the feature is in predicting the target label via $I(x_i;c)$, the value of mutual information (ROSS, 2014).

$$\min R(S),\ R = \frac{1}{|S|^2}\sum_{x_i,x_j \in S} I(x_i;x_j). \tag{2.6}$$

After one first candidate feature is identified via maximizing equation 2.5, simply adding new features via the same criteria could likely add redundancy to the feature set in the form of similar features. Thus, equation 2.6 takes place to avoid redundancy against the set of already selected features. In this equation, R represents the redundancy estimation for a pair of features x_i and x_j, where x_i is a candidate for selection, and x_j is an already selected feature. The condition effectively measures which features have their information already represented by other similar selected features, thus indicating a lesser relevance for the supervised learning task than other features that add unseen information. Therefore, $minR(S)$ represents the minimization of redundancy for features within S.

$$\max \Phi(D,R),\ \Phi = D - R. \tag{2.7}$$

To conciliate the two criteria in a final relation, equation 2.7 defines a Φ operator

that should be maximized to achieve the "minimal-redundancy-maximal-relevance" criterion for selecting a feature. The operator can be evaluated as many times as necessary until the number of desired features is achieved. This results in greedily adding a new feature to a selection of features, and reiterating the operator for the remaining set of features to identify the next candidate until the final set is selected.

2.1.2.5 Relief-F

Relief-F is an efficient and popular multivariate filter algorithm for feature selection proposed by Kononenko (KONONENKO, 1994), extending the original definition of the Relief algorithm proposed by Kira and Rendell (KIRA; RENDELL, 1992). The original Relief was designed for supervised feature selection of binary classification problems, with robust performance in both discrete and continuous variables, and focused on identifying relations between variables. The contribution provided by Relief-F extends the method to properly deal with multi-class classification problems, and also to better adapt the method to datasets containing noisy or incomplete data.

The original process defined in Relief (KIRA; RENDELL, 1992) is described in Algorithm 1. In the algorithm, the first step is to randomly select a reference instance (or sample) R from the dataset (line 4), and identifying the nearest instance from the same class (a hit) named H, and the nearest instance from another class (a miss) named M (line 5). Then, for each attribute in the dataset, calculate the score or weight associated with the feature in relation to the difference between R and H, and between R and M (line 7), also described in equation 2.8. Finally, the values are normalized with m, which represents the number of samples picked as reference points (and the number of iterations of the algorithm) for calculating the weights $W[A]$ of each feature A.

Algorithm 1: Relief main loop algorithm. Source: (KIRA; RENDELL, 1992)

```
1  begin
2      set all weights W[A] := 0;
3      for i := 1 to m do
4          randomly select an instance R;
5          find nearest hit H and nearest miss M;
6          for A in all_attributes do
7              W[A] := W[A] − diff(A, R, H)/m + diff(A, R, M)/m;
8          end
9      end
10 end
```

$$W[A] := W[A] - \frac{diff(A, R, H)}{m} + \frac{diff(A, R, M)}{m} \tag{2.8}$$

In general, the idea behind equation 2.8 is that if feature values for same-class samples (R and H) are very similar, the feature is important to distinguish the class. Otherwise, if they are too different, the feature is irrelevant. For feature values in samples from different classes (R and M), divergent values indicate an important feature. Alternatively, if features in this scenario are too similar, they are not important. This calculation $W[A]$ for a feature A is accumulated for all m sampled reference values R and represents the final feature score, which can be used to rank the most important features.

$$W[A] := W[A] - \frac{diff(A, R, H)}{m} + \sum_{C\ !=\ class(R)} \frac{P(C) \times diff(A, R, M(C))}{m} \tag{2.9}$$

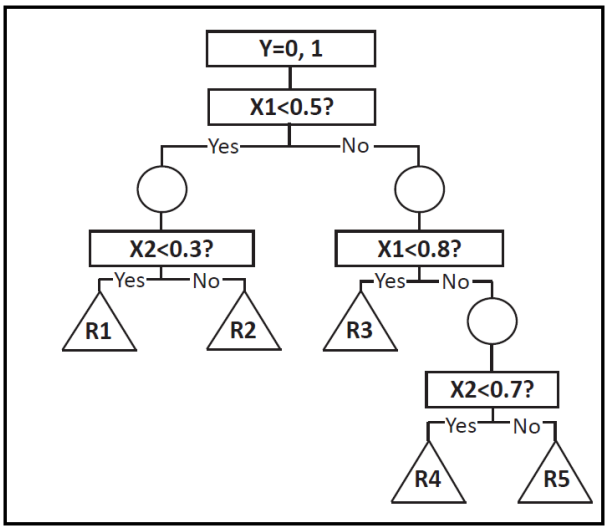
Relief-F extends the original algorithm by proposing a replacement for equation 2.8 as defined in equation 2.9. In the new equation, the calculation now considers multiple classes by selecting one nearest miss sample $M(C)$ for every class C that is not the class of R. The contribution of each individual class is averaged with weight of the probability of each class $P(C)$. The objective is to estimate the ability of features to separate each pair of classes regardless of which two classes are closest to each other (KONONENKO, 1994).

2.1.2.6 Decision Tree

Decision trees are yet another classic approach used in data mining and feature selection for their simple functionality and effective outputs in multiple fields, presenting a good performance in the analysis of discrete and continuous variables, and also in datasets with missing data (SONG; YING, 2015). Decision trees can be interpreted as an embedded feature selection method since feature selection is part of the learning process of a decision tree method (LAL et al., 2006). In this sense, decision trees typically evaluate the importance of a feature simply by removing it from the training subset and evaluating the variation in the performance of the final model.

A decision tree model is usually composed of nodes and branches, and built via a series of operations including splitting, stopping, and pruning. Figure 2.1 illustrates an example of a decision tree built for a binary target variable Y and receiving two continuous

Figure 2.1: A simple decision tree based on binary target variable Y.



Source: (SONG; YING, 2015)

variables as input, $X1$ and $X2$.

The basic concepts required to build a decision tree model are summarized by Song and Ying (SONG; YING, 2015) as:

- Nodes: nodes are usually divided into root nodes, internal nodes, or leaf nodes. Root nodes are also known as decision nodes, and represent a condition or choice that divides the outputs into two or more subsets. Internal nodes are connected to parent nodes and child nodes and also represent a choice or condition that further divides the options into smaller subsets. Leaf nodes are the result of the set of choices defined by their ancestor nodes.
- Branches: branches represent the outcomes of a choice from root or internal nodes, usually representing the decision of a classification rule.
- Splitting: splitting is the process of refining the classification decision by creating new child nodes and branches, starting from the route node, based on an input variable that best splits records according to the target variable and specific data characteristics. These characteristics vary according to implementation and can include metrics such as entropy, Gini index, classification error, information gain, etc. The splitting process usually continues until certain stopping criteria are achieved.
- Stopping: stopping criteria is necessary to avoid overly complex models. Typical stopping criteria include limiting the minimum number of records in a leaf, limiting the number of records in a node before splitting, or limiting the depth of nodes the

tree can accept from root to leaf nodes.

- Pruning: sometimes trees may end up too large and complex, and it may be necessary to prune some nodes to achieve more robust results. This is a common approach to simplify models after the initial tree is generated, and the process is performed by selecting sub-trees and excluding nodes, and evaluating the resulting performance of the model according to the desired criteria, such as error rates.

After the iterative process of splitting, multiple logical rules based on features are derived from the tree. These rules fundamentally indicate the important features for the classification process, and therefore provide a ranking of important features after a final model is selected.

2.1.2.7 Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) is a linear model first designed with the goal of producing interpretable models for regression use cases and signal processing by providing relevant subsets of features in a final model (TIBSHIRANI, 1996). Lasso combines the concepts of subset selection and ridge regression, removing features altogether or simply shrinking their weight in the model decision process. One notable benefit of the Lasso method is its inherent good performance in subsets of data with high dimensionality and low sample sizes. Since feature selection is part of the learning process of the model, Lasso can be considered an embedded feature selection method (LAL et al., 2006).

The main principle behind the Lasso is its regularization variable selection performed via minimization of a sum of squared errors, as defined in equation 2.10 (FONTI; BELITSER, 2017). In the equation, $\mathbf{Y}$ is the vector representation of the output or target variable, while $\mathbf{X}$ is the vector representation of the input variables and β represents the vector of coefficients of the model. Parameter t represents the upper bound of the sum of coefficients of the model, used in the regularization process, and n and k represent the number of samples in the dataset.

$$\text{minimize } \left(\frac{\|\mathbf{Y} - \mathbf{X}\beta\|_2^2}{n} \right), \text{ subject to } \sum_{j=1}^{k} \|\beta\|_1 < t \tag{2.10}$$

The optimization process for equation 2.10 is equivalent to the parameter estimation represented in equation 2.11.

$$\hat{\beta}(\lambda) = \underset{\beta}{\text{argmin}} \left(\frac{\|\mathbf{Y} - \mathbf{X}\beta\|_2^2}{n} + \lambda \|\beta\|_1 \right), \text{ where } \lambda \geq 0 \tag{2.11}$$

In equations 2.10 and 2.11, t and λ have a reverse relationship in that as t increases to infinity, λ decreases 0, turning the problem into an ordinary least squares. As t becomes 0, regularization coefficients shrink to 0 and λ increases to infinity (FONTI; BELITSER, 2017). During the regularization process of such equations, it is expected that some of the coefficients are reduced to zero, indicating that the respective features are discarded from the model, while relevant features have their coefficient increased. The coefficients of the final model can, therefore, be used to infer feature importance from the feature subset.

2.1.2.8 Linear SVM

Linear support vector machines (Linear SVMs) are a variation of traditional SVMs that use a linear kernel function. SVMs are classification models designed to estimate boundaries between data points to separate samples of different classes (BOSER; GUYON; VAPNIK, 1992). The concept starts with the definition of support vectors, a selection of samples closest to the decision boundary, which are then used as part of a decision function that estimates the boundary between classes. Given this process of using a small selection of samples for the boundary estimation, SVMs are typically efficient models with good comparable performance, especially in high-dimensional datasets (GUYON et al., 2002).

For feature selection, Linear SVMs can be used as an embedded method, as individual weights are assigned to different features in the training process and can be used for ranking and elimination. Equation 2.12, as defined Guyon et al. (GUYON et al., 2002) describes the optimization process performed in typical Linear SVMs.

$$\begin{cases} \text{Minimize over } \alpha_k : \\ J = (1/2) \sum_{hk} y_h y_k \alpha_h \alpha_k (\mathbf{x}_h . \mathbf{x}_k + \lambda \delta_h k) - \sum_k \alpha_k \\ \text{subject to :} \\ 0 \leq \alpha_k \leq C \text{ and } \sum_k \alpha_k y_k = 0 \end{cases} \tag{2.12}$$

In equation 2.12, training samples $\mathbf{x}_k$ are n dimensional feature vectors, and $\mathbf{y}_k$ represents the encoded class labels. Parameters C and λ are positive constants called soft margin parameters, and $\delta_h k$ is the Kronecker symbol with $\delta_h k = 1$ if $h = k$ and 0

otherwise (GUYON et al., 2002). Finally, parameters α_k are the parameters or weights to be adjusted in order to minimize the equation.

$$D(\mathbf{x}) = \mathbf{w}.\mathbf{x} + b \tag{2.13}$$

$$\text{with } \mathbf{w} = \sum_k \alpha_k y_k \mathbf{x}_k \text{ and } b = \langle y_k - \mathbf{w}.\mathbf{x}_k \rangle$$

After the weight optimization, the decision function for an input vector $\mathbf{x}$ is defined in equation 2.13. In the equation, the weight vector $\mathbf{w}$ is a linear combination of α_k, y_k, and $\mathbf{x}_k$, where most weights α_k are zero. All training values with non-zero weights are defined as support vectors. The bias value b is calculated as an average over marginal support vectors, defined after the inequality condition $0 < \alpha_k < C$.

Once the weight adjustment process is finished and the model is generated, weights $\mathbf{w}$ can be used to infer the ranking of features, where $\mathbf{w}_i$ can be used as the ranking criteria for each feature i.

2.1.2.9 Random Forest

Random forests is a popular machine learning method designed for classification and regression tasks, based on the ensemble of multiple tree models combined with independent random vectors sample from the input data for each predictor to generate a combined solution for an output variable (BREIMAN, 2001). Because it uses an ensemble of tree models (such as decision trees) as its core strategy, random forests are inherent variable selectors and can be considered embedded feature selection approaches when used for this purpose, since the variable selection process is part of the learning process when generating tree models (SPEISER et al., 2019).

From the original definition by Breiman (BREIMAN, 2001), random forests consist of a collection of tree-based models $h(x, \Theta_k, k = 1, ...)$, which generates a class prediction for an input x using k independent random vectors Θ_k, all based on the same distribution. For each vector Θ_k and input x, a number of k different tree estimators $h_1(x), h_2(x), ..., h_k(x)$ are generated.

$$h(x, \Theta_k) = \{h_1(x), h_2(x), ..., h_k(x)\} \tag{2.14}$$

The tree estimator depth, the number of variables selected by each tree, and the set of features selected depend on the parametrization of the underlying tree algorithm

used. The original random forests method recommends the usage of linear combinations of the input variables of constant size M, and trees with no depth limit, without pruning (BREIMAN, 2001). Typically, the selection of variables inside the tree from each subset Θ_k is random as traditional decision tree algorithms are based on random variable sampling (SONG; YING, 2015). The final vote of the ensemble model for input (x) is defined as the majority vote between all estimators $h_k(x)$.

After the generation of the ensemble model, feature selection can be performed in a number of ways, either as an aggregation of the individual feature weights extracted from the base tree models, or based on the exclusion of trees (and respective variable subset exclusion) and the resulting impact on predictive performance, usually following a feature ranking strategy (SPEISER et al., 2019).

2.2 Feature set evaluation

After performing feature selection in a dataset, it is crucial to evaluate the quality of the selected subset of features, and measure the potential losses of information that may happen due to removing unknowingly relevant features. In this section, we address the evaluation of feature selection tasks for supervised problems. First, we introduce the reasoning behind the usage of traditional machine learning metrics, such as classification metrics, for the evaluation of subsets of selected features in terms of preserving or improving the separation of data. After that, we describe the classification metrics that will be used across this work.

2.2.1 Machine learning and feature selection

When speaking of feature selection, it is perhaps essential to elaborate on the basis of machine learning theory and its high correlation with feature selection in literature, even if both fields exist and can be used separately and for different purposes. Both fields historically branch from data mining into individual research fields, but feature selection was naturally adopted as an important tool for the optimization of machine learning tasks in several aspects, notably model classification performance, resource utilization, data interpretability, and readability of results (LI et al., 2017). Nonetheless, machine learning techniques are also used as part of the mechanisms that compose several feature selection

methods. During this work, several concepts and metrics of machine learning techniques will be used to evaluate the effectiveness of feature selection methods.

Computer science fundamentally originates from the constant need to automate human tasks via machines, yet we frequently fail to be able to fully articulate through programs or instructions the criteria humans use to perform such tasks. From this, machine learning arises as a concept based on "learning through exposition", idealized as an approach closer to that which humans and other animals use. On the other hand, it also comes from the need to perform increasingly complex tasks required by technological advancements, where humans fail to provide reliable abstractions of large and complex data — such as extracting information from medical data, analyzing genomic and astrological data, predicting the weather, etc (SHALEV-SHWARTZ; BEN-DAVID, 2014).

In many ways, the objectives behind the applications of feature selection for human analysis are the same as using it for machine learning: the intention is to narrow down data sources to relevant and ideally non-redundant features, to effectively differentiate between data points while requiring fewer resources. Thus, machine learning models and metrics are great indicators of the quality of a selection of features provided by a feature selection algorithm — a concept that originated whole branches of feature selection algorithms, classified as wrapper methods. Even so, machine learning metrics and models can be used to evaluate other classes of feature selection algorithms, further explored in section 2.2.2.

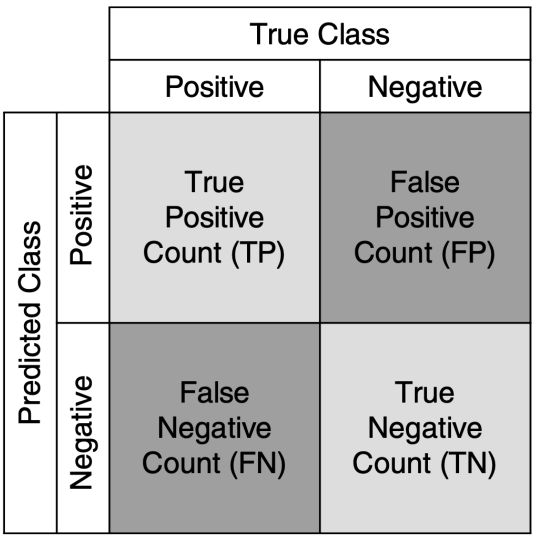
Fundamentally, machine learning follows the same major division previously mentioned for feature selection methods in section 2.1, being mostly divided into supervised and unsupervised learning, and sometimes semi-supervised. The supervised and unsupervised machine learning fields have evolved over time into multiple branches of algorithms designed for different types of inference goals. In this work, the focus will be centered around supervised learning tasks, specifically classification tasks, where labels are defined as two or more distinct known categorical classes.

The next subsections will explain the fundamental concepts of classification metrics and models used in this work to evaluate the results of the proposed feature selection solution.

2.2.2 Classification metrics

When evaluating a classification task, a set of predictions performed by a classifier is typically summarized in hits and errors using a confusion matrix, also known as a coincidence matrix or classification matrix (OLSON; DELEN, 2008).

Figure 2.2: A confusion matrix used for summarizing classifications.



Source: (OLSON; DELEN, 2008)

Figure 2.2 exemplifies the disposition of predicted labels for a two-classes problem. In the diagonal, the correctly predicted labels are divided into true positives and true negatives, and the incorrectly predicted labels are divided into the sides, as false positives and false negatives. These categorizations are defined as follows:

- True Positives (TP): labels correctly predicted as belonging to the target class.
- True Negatives (TN): labels correctly predicted as not belonging to the target class.
- False Positives (FP): labels incorrectly predicted as belonging to the target class.
- False Negatives (FN): labels incorrectly predicted as not belonging to the target class.

From these counts, several metrics can be calculated to evaluate different aspects of a classification performance, and the most common are described in the following equations:

$$A\ (Accuracy) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.15}$$

$$P\ \textit{(Precision)} = \frac{TP}{TP + FP} \tag{2.16}$$

$$R\ \textit{(Recall)} = \frac{TP}{TP + FN} \tag{2.17}$$

$$F1\ \textit{(F1-Score)} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{2.18}$$

Accuracy (2.15) describes the sum of correctly classified positives and negatives divided by the total number of classifications performed, providing a proportion of correctly labeled samples, but ignoring the individual class performance in multi-class problems. Precision (2.16) describes the sum of correctly classified positives divided by the sum of correctly and incorrectly classified positives, essentially rating the effectiveness of the classifier when assigning positive labels and decreasing in value as incorrect positive labels are assigned. Recall (2.17), on the other hand, evaluates the effectiveness of the classifier in correctly recognizing positive labels from the evaluated samples, penalizing the metric as incorrect negative labels are assigned. Finally, F1-Score (2.18), also known as F1-metric, utilizes precision and recall in a harmonic mean as an aggregated metric, largely penalizing its value if at least one of the original metrics underperforms significantly.

Out of the mentioned metrics, Accuracy (2.15) is the only one that measures the performance across all classes directly, while the others serve the purpose of measuring the performance of individual classes in a classification problem. To simplify the summarization of such metrics in multi-class problems, which may be misleading as the number of classes increases or if the number of samples of each class becomes unbalanced, macro-averaging and micro-averaging are the predominant approaches applied in machine learning problems (TAKAHASHI et al., 2022).

Macro-averaging (2.19, 2.20, 2.21) consists of calculating the arithmetic mean of a chosen metric over the classes of a multi-class problem, with each class having the same weight despite the number of samples it contains. Micro-averaging (2.22, 2.23, 2.24), on the other hand, consists of calculating the average value of a chosen metric by considering all the samples of all classes at once, resulting in a metric that is more influenced by larger classes and therefore more representative of them. The two approaches can be described by the following equations, considering the metrics presented previously, where i represents each class in an r-classes classification problem (TAKAHASHI et al., 2022):

$$P_{ma}\ \textit{(Macro-Averaged Precision)} = \frac{1}{r}\sum_{i=1}^{r} P_i \tag{2.19}$$

$$R_{ma}\ \textit{(Macro-Averaged Recall)} = \frac{1}{r}\sum_{i=1}^{r} R_i \tag{2.20}$$

$$F1_{ma}\ \textit{(Macro-Averaged F1-Score)} = \frac{1}{r}\sum_{i=1}^{r} F1_i \tag{2.21}$$

$$P_{mi}\ \textit{(Micro-Averaged Precision)} = \frac{\sum_{i=1}^{r} TP_i}{\sum_{i=1}^{r} (TP_i + FP_i)} \tag{2.22}$$

$$R_{mi}\ \textit{(Micro-Averaged Recall)} = \frac{\sum_{i=1}^{r} TP_i}{\sum_{i=1}^{r} (TP_i + FN_i)} \tag{2.23}$$

$$F1_{mi}\ \textit{(Micro-Averaged F1-Score)} = 2\frac{P_{mi} \times R_{mi}}{P_{mi} + R_{mi}} \tag{2.24}$$

In this work, macro-averaging will be preferred over micro-averaging, since many real-life classification problems tend to contain considerable class imbalance problems (ALI; SHAMSUDDIN; RALESCU, 2013).

2.3 Genetic Algorithms

Genetic algorithms (GAs) are a type of metaheuristic often used for solving large combinatorial problems with adaptable execution times, which makes them exceptional candidates for improving solutions in feature selection. They are part of a larger group of biology-inspired algorithms named evolutionary algorithms, which share three main characteristics (YU; GEN, 2010): they are *population-based*, improving a population of solutions over time, called individuals; they are *fitness-oriented*, meaning solutions are evaluated according to fitness criteria, giving preference to the fittest individuals; they are *variation-driven*, inducing exploration of the solution search space through operations that perform changes in individuals, mimicking genetic alterations.

The first definition of Genetic algorithms (GAs) to gain notoriety was the work by J.H. Holland (HOLLAND, 1992), which described GAs as computational methods based on the principle of evolving candidate solutions over iterations based on evolutionary theories from biology. The inspiration comes from the behavior observed in populations of

biological organisms, which compete amongst themselves for resources over generations of individuals. Ever since its initial proposal, GAs have been adapted to multiple applications in numerous fields, tackling computationally demanding problems with heuristics that simplify the exploration of large spaces of possible solutions.

Beasley et al. (BEASLEY; BULL; MARTIN, 1993) describe GAs as a direct analogy of natural behavior, where individuals in a population compete with their pairs for food, water, shelter, and other resources, as well as candidate mates for reproduction. Individuals who are most successful in surviving will likely have more chances to mate and therefore produce more offspring. Individuals who perform badly in surviving will likely have fewer chances and therefore produce little to no offspring. In this sense, genes from the most "fit" individuals, who are best adapted to the environment, will likely spread to other individuals in future generations, as their offspring is also likely to have better chances of succeeding in survival, eventually surpassing the potential of their parents and becoming more and more specialized in the tasks required for surviving in the environment.

2.3.1 The basic genetic algorithm

Translating this description into a generalist problem-solving method, standard GAs manipulate populations of "individuals" that represent solutions to a given problem (BEASLEY; BULL; MARTIN, 1993). Over a number of generations, these individuals are evaluated by a fitness function and receive numeric scores of their performance. The scores naturally indicate which individuals are more fit to solve the problem and therefore should have more chances at reproduction (producing new individuals that inherit some of their characteristics), or simply survival (being passed on to the next generation of individuals to be evaluated and compared again). In reproduction, the characteristics of each individual are passed on to the offspring through their "genotype", which is composed of a representation of parameters often called genes, or sometimes referred to as chromosomes. Individuals with low fitness scores are considered bad candidates for reproduction and are often discarded in favor of new individuals generated by strategies such as recombinations through operations called crossovers, and randomizations of initially successful genotypes through mutations.

Both the concept of elitism, designed to preserve the best solutions, and the other mechanisms related to passing on or sharing genes present in good-performing individ-

uals, namely mutation and crossover, result in the spread of characteristics of successful solutions in the population of individuals as generations pass (PURSHOUSE; FLEMING, 2002). This makes GAs converge into progressively better solutions for the problem being solved, but some attention is required to concerns such as maintaining a decent diversity of solutions in the population, avoiding convergence limited to local optima, and controlling the amount of randomization in new candidates, which can end up producing repetitive and ineffective solutions. Beyond elitism strategies, such challenges are usually tackled directly by the mutation and crossover operators, whose main function is to explore new gene combinations while preserving likely successful genes from the current solutions (LIM et al., 2017).

A pseudo-code representing a standard procedure for GAs based on the original description by Holland (HOLLAND, 1992) can be seen in Algorithm 2, employing fitness evaluation, elitism, crossovers, and mutations to individuals over generations. In the algorithm, the initial population is created and assigned a initial fitness score (lines 2 to 5), and for a defined number of generations, the population is updated using the 3 mechanisms mentioned previously: the elitism strategy decides which $E*S$ solutions to preserve to the next generation and the remaining are discarded (line 7); the crossover operator creates a number $C*S$ of offspring individuals by combining genes of the most fit individuals (line 8); and the mutation operator creates $M*S$ by altering the genes of random individuals, usually from the offspring set (line 9). After this, fitness scores are calculated once again (lines 10 to 12) and the procedure repeats for the remaining generations (lines 6 to 13). In this scenario, the number of preserved individuals ($E*S$), the number of offspring generated via crossover ($C*S$), and the number of mutated individuals ($M*S$) should equal the number of S individuals.

2.3.2 Multi-objective genetic algorithms

Genetic algorithms branched over time into multiple approaches applied to different types of problems (KATOCH; CHAUHAN; KUMAR, 2021). Many aspects have been improved by different techniques, mostly related to the maintenance of diversity and elitism, improved crossover and mutation operators, more complex representations of genotypes, or most notably the reinterpretation of the evaluation of the fitness of candidate solutions for complex problems.

The limiting factor of the originally proposed fitness function is that its output

Algorithm 2: Basic genetic algorithm structure

```
Data: N: number of generations, S: population size, E: elitism rate, C:
      crossover rate, M: mutation rate
Result: evolved candidate solutions
begin
    initialize population with S individuals;
    for individual in population do
        evaluate fitness;
    end
    for generation in N do
        apply elitism to E * S most fit individuals;
        apply crossover to C * S most fit individuals;
        apply mutations to M * S random individuals;
        for individual in population do
            evaluate fitness;
        end
    end
end
```

numeric value is considered the only real objective in survival, no matter how complex the function is or how many parameters are used to describe it. This is an issue for problems where the impact of distinct, usually contradictory objectives cannot be precisely measured into a single numeric objective. An example of this is feature selection, where the number of features should ideally be the smallest possible, but higher numbers of features usually mean more representative feature sets when it comes to data separation, thus acting as opposite objectives that need to be optimized with no clear balance between them.

To overcome the issue of having a single objective value to evaluate often contradicting evolutionary objectives, multi-objective genetic algorithms (MOGAs) were eventually proposed by Fonseca et al. (FONSECA; FLEMING, 1993), where the concept of Pareto dominance was introduced to handle the optimization of multiple objectives by preserving non-dominated solutions in separate Pareto fronts for each objective, thus evaluating each objective independently. This concept originated a series of MOGAs denominated Pareto-based MOGAs (KATOCH; CHAUHAN; KUMAR, 2021). A similar proposal of MOGA was made in Horn et al. (HORN; NAFPLIOTIS; GOLDBERG, 1994), where the concept of Pareto dominance was used in a solution named niched Pareto genetic algorithm (NPGA). After that, Deb et al. (DEB et al., 2000) developed the non-dominated sorting genetic algorithm (NSGA), which was further improved in its highly successful successor, the NSGA-II (DEB et al., 2002), representing a landmark in the

field of MOGAs.

Another relevant class of MOGAs worth of mention is the decomposition-based genetic algorithms (KATOCH; CHAUHAN; KUMAR, 2021), but these will not be explored in this section given the focus on (and higher notoriety of) Pareto-dominated GAs. In the next subsection, we analyze the inner workings of the NSGA-II, which is used as a basis for the proposed method in this work.

2.3.3 NSGA-II: the non-dominated sorting genetic algorithm

NSGA-II, proposed initially by Deb et al. (DEB et al., 2002), is one of the most relevant GAs in literature. Multiple small improvements have been proposed to the original algorithm since its initial publication, but the core of what differentiates NSGA-II from other multi-objective approaches still ties back to its initial proposal and the concepts of non-dominated sorting and crowding-distance (VERMA; PANT; SNASEL, 2021).

2.3.3.1 Non-dominated sorting

The NSGA-II algorithm centers itself on the concept of Pareto domination, where solutions that are not surpassed under the objective metric by any other solution form the optimal "front" of solutions to the objective at hand. In this case, NSGA-II defines the "fast non-dominated sorting approach" as its strategy to sort the population into different domination levels (DEB et al., 2002).

Algorithm 3 describes the sorting approach. For each individual p in population P, the first step is to calculate the number of solutions that dominate the current solution, the domination count n_p, and the set of solutions that are dominated by it, S_p (lines 6 to 12). The solutions with a domination count of zero will be denominated as the first front, F_1 (lines 13 to 16). The dominated sets S_p for each of these solutions in the first front are then iterated over (lines 21 to 27), and each dominated solution has its domination count decreased by one (line 22). At the end of the iteration, all solutions that reach a domination count of zero are denominated as the second domination front, F_2 (line 30). The process then repeats (from lines 18 to 31) for the dominated sets of this second front, and a third domination front F_3 is formed, and so on until all fronts are identified, resulting in a complexity of $O(MN^2)$, where M is the number of objectives and N is the size of the population of solutions.

Algorithm 3: Fast non-dominated sorting approach. Source: (DEB et al., 2002)

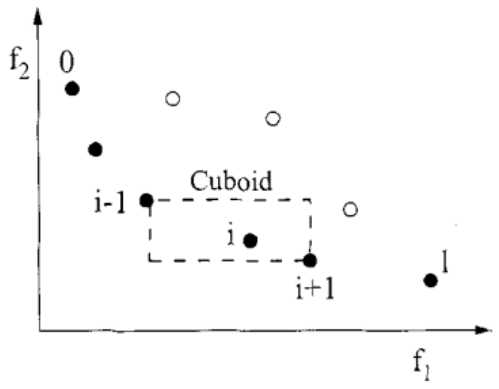
```
Data: P: population of solutions
Result: F: domination fronts
begin
    for p in P do
        S_p = ∅;
        n_p = 0;
        for q in P do
            if p dominates q then
                S_p = S_p ∪ {q};
            else
                if q dominates p then
                    n_p = n_p + 1;
                end
            end
            if n_p == 0 then
                p_rank = 1;
                F_1 = F_1 ∪ {p};
            end
            i = 1;
            while F_i ≠ ∅ do
                for p in F_i do
                    Q = ∅;
                    for q in S_p do
                        n_q = n_q − 1;
                        if n_q == 0 then
                            q_rank = i + 1;
                            Q = Q ∪ {q};
                        end
                    end
                end
                i = i + 1;
                F_i = Q;
            end
        end
    end
end
```

2.3.3.2 Crowding-distance

The crowding-distance approach is a strategy used to preserve the diversity of solutions in a population, prioritizing individuals who occupy different spots in the dimensional space of possible solutions. It differentiates from previous diversity preservation techniques such as the one proposed in the first iteration of NSGA by not requiring user-defined parameters to achieve a good performance — thus reducing the need for problem-specific fine-tuning — and having a comparably smaller computational complexity of $O(MN \log N)$, where M is the number of objectives and N is the size of the population of solutions. The crowding-distance requires the calculation of a density estimation metric and a crowded-comparison operator.

Figure 2.3 displays a visual representation of the density estimation used in the crowding-distance calculation process. To achieve the distance calculation, solutions are sorted according to the objective metrics in each dimension. In the image, dimensions are represented by f_1 and f_2, but the calculation can be made for more than two objectives. The definition of density states that for each individual solution i, the distance $i_{distance}$ is the average side length of the cuboid (or normalized distance) between the nearest solutions on either side of the initial solution i, namely i_{-1} and i_{+1}. Solutions on the boundaries of the search space, e.g. solutions 0 and 1, receive an infinite $i_{distance}$ value.

Figure 2.3: A representation of the crowding-distance calculation. Points marked in filled circles are solutions of the same non-dominated front.



Source: (DEB et al., 2002)

Once the density calculation is finished, the crowding-distance operator takes place to ensure uniformity in the spread of solutions in the Pareto front. The operator defines a sorting $\prec_n$ of the solutions based on two factors, the non-domination rank i_{rank},

and the crowding-distance $i_{distance}$, as described in expression 2.25.

$$i \prec_n j \text{ if } (i_{rank} < j_{rank}) \text{ or } (i_{rank} = j_{rank}) \text{ and } (i_{distance} > j_{distance}) \tag{2.25}$$

The sorting considers first the lowest non-domination rank i_{rank}, and for solutions with equal non-domination ranks, the higher crowding-distances $i_{distance}$ are prioritized, resulting in the prioritization of solutions in less dense regions of the Pareto front.

2.3.3.3 Main loop

With the definitions of the fast non-dominated sorting and the crowding-distance calculation, we can now describe the main loop of NSGA-II. In the first generation, a clean population of solutions P_0 is generated, sorted according to domination as a fitness measure, and goes through the usual process of binary tournament selection, recombination, and mutation operators to create N offspring solutions named Q_0.

After the first generation, the process in Algorithm 4 takes place, for every generation t, to apply both the non-dominated sorting and the crowding-distance strategies. First, a population R_t is formed with the combination of population P_t and the offspring population Q_t, resulting in a combined size of $2N$ (line 2). This population is then sorted according to domination (line 3), the crowding distance assignment takes place (line 7), and a new population P_{t+1} is created by including the newly formed domination fronts F, from the first front ($i = 1$) to the subsequent fronts ($i = i + 1$), until the number of individuals N is reached (lines 6 to 10), since the solutions in the first fronts are the best in the population. The last front to be included in the population may be smaller than the remaining amount of individuals needed to reach N solutions, thus a last sorting is performed to pick the solutions in regions with lesser density using the operator $\prec_n$ (line 11). Figure 2.4 shows a depiction of the same process with all the mentioned steps.

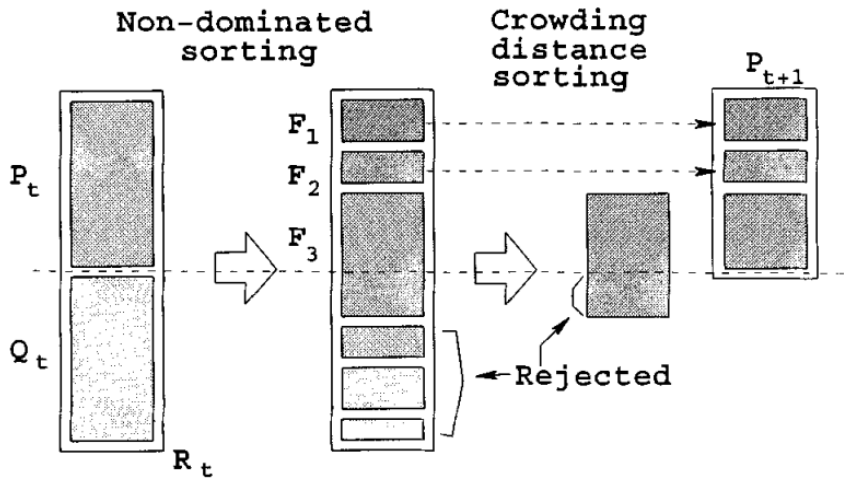
After that, the new population is formed (lines 12 to 13) and ready to be used in the next generation $t + 1$, where the procedure in Algorithm 4 is repeated after the usual operators of the genetic algorithm (the selection, crossover, and mutation). The overall complexity of the process is considered $O(MN^2)$, where M is the number of objectives and N is the size of the population of solutions. This is mostly due to the non-dominated sorting, which has the same complexity.

Algorithm 4: NSGA-II main loop algorithm. Source: (DEB et al., 2002)

Data: t: generation identifier, P_t: population of solutions for t, Q: population of offspring for t, F_i: domination front i, $\prec_n$: crowding-distance operator, N: number of individuals in populations
Result: P_{t+1}: new population for $t+1$

```
1  begin
2      R_t = P_t ∪ Q_t;
3      F = fast-non-dominated-sort(R_t);
4      P_{t+1} = ∅;
5      i = 1;
6      while |P_{t+1}| + |F_i| ≤ N do
7          crowding-distance-assignment(F_i);
8          P_{t+1} = P_{t+1} ∪ F_i;
9          i = i + 1;
10     end
11     sort(F_i, ≺_n);
12     P_{t+1} = P_{t+1} ∪ F_i[1 : (N − |P_{t+1}|)];
13     P_{t+1} = make-new-pop(P_{t+1});
14     t = t + 1;
15 end
```

Figure 2.4: A representation of the NSGA-II algorithm.



Source: (DEB et al., 2002)

2.4 Chapter summary

In this chapter, section 2.1 presented an introduction to the theoretical background of feature selection, exploring the reasoning behind its applications and its usage focused on classification tasks for supervised learning problems. In the same section, a review of the feature selection methods used in this work is presented.

A description of the general sense of the relationship between machine learning and feature selection is provided in section 2.2, describing the usage of machine learning metrics in the evaluation of the quality of feature sets provided by feature selection methods.

Lastly, a bridge was also made to genetic algorithms in section 2.3, and the multi-objective non-dominated sorting genetic algorithm (NSGA-II) is explained in detail to serve as a foundation for the implementation proposed in chapter 4.

3 RELATED WORK

In feature selection for high-dimensional data, often also characterized by a scarcity of samples, recent literature has seen a surge in efforts to develop generalist methods capable of addressing diverse domain-specific challenges. Real-world domains are frequent sources of high-dimensional data, with areas like cancer and disease research and genome-wide studies as common focal points for benchmarking and evaluating newly introduced methodologies (ALHENAWI et al., 2022).

The study by Hambali et al. (HAMBALI; OLADELE; ADEWOLE, 2020) highlights that recent contributions using feature selection in the field of cancer research tend to employ filter methods more frequently than other categories, given their more deterministic and efficient nature when compared to wrapper and embedded approaches, which may scale badly as the number of dimensions grows. The authors also mention the increasing amount of research directed towards hybrid methods combining the different characteristics of the classic feature selection categories. Still, even with recent advances, Alhenawi et al. (ALHENAWI et al., 2022) reinforce the need for further research in order to achieve more efficient feature selection methods for high-dimensional data such as in the case of microarray data processing (especially used in cancer research) in terms of both computation time and accuracy of the classification tasks.

The work by Piri et al. (PIRI et al., 2023) addresses a similar concern and offers a comprehensive review of hybrid methodologies recently introduced for the feature selection problem, spanning various domains, including biomedical research. These methodologies explore the fusion of evolutionary algorithms with other techniques to reduce high-dimensional datasets into smaller, more manageable ones. Such combinations typically enhance the efficacy of the initial solutions, yielding results that are notably more dependable than those obtained from non-hybrid methods, especially pure filter and wrapper methods. Furthermore, the review identifies crucial aspects often overlooked by most methods in their original formulations, such as:

1. Failing to assess the quality of the solution using real-world applications (such as the biomedical domain), often relying on a small number of datasets and occasionally neglecting the evaluation of high-dimensional datasets with substantial feature counts.
2. Not considering multiple objectives, typically concentrating fitness evaluations solely on error rates while overlooking the dimensionality reduction criteria.

3. Increasing execution times compared to more straightforward methods and lacking comparative analyses with other established approaches.

Considering evolutionary solutions similar to the approach proposed in this work, several notable examples from recent years validate their performance with real-world data, particularly in cancer-related and other biomedical applications. These examples combine genetic algorithms (GAs) with other methods for feature selection and the identification of relevant biomarkers in datasets, which typically consist of a limited number of samples but can include thousands of features. In all these studies, combining techniques leads to enhanced classification performance compared to baseline methods that do not utilize GAs. These works are listed in Table 3.1 and discussed in this section.

Table 3.1: Summary of notable related works using genetic algorithms for feature selection.

Publication	**GA category**	**FS category**	**Dataset domains**	**N. Data sets**	**Max. Features**[1]	**Samples**[1]	**Evaluation metrics**[2]
Aličković and Subasi (2017)	Single-objective	Ensemble	Breast Cancer	2	32	569	Accuracy, Area under curve (ROC), F-Score
Aalaei et al. (2016)	Single-objective	Wrapper	Breast Cancer	3	34	198	Accuracy, Specificity, Recall
Ahmad et al. (2015)	Single-objective	Wrapper	Breast Cancer	1	9	699	Accuracy, Specificity, Recall
Liu et al. (2018)	Single-objective	Hybrid	Multiple domains	5	240	7399	Accuracy, Specificity, Recall
Sayed et al. (2019)	Single-objective	Ensemble	Cancer domains	3	27578	276	Accuracy
Maleki, Zeinali and Niaki (2021)	Single-objective	Hybrid	Lung Cancer	1	23	1000	Accuracy, Specificity, Recall

Continue on the next page

[1]Dimensions of the dataset under evaluation with the highest number of features.
[2]Metrics used to evaluate the quality of the final feature subset after the feature selection process.

Table 3.1: Summary of notable related works using genetic algorithms for feature selection (cont.).

Publication	**GA category**	**FS category**	**Dataset domains**	**N. Data sets**	**Max. Features**[1]	**Samples**[1]	**Evaluation metrics**[2]
Deng et al. (2022)	Multi-objective	Hybrid	Cancer domains	14	60483	123	Accuracy, F-Score, Precision, Recall
Hasnat and Molla (2016)	Multi-objective	Hybrid	Multiple domains	3	7129	72	Accuracy
Tan, Lim and Cheah (2014)	Multi-objective	Hybrid	Multiple domains	3	32	569	Accuracy, Precision, Recall
Xue et al. (2021)	Multi-objective	Hybrid	Multiple domains	10	649	1000	Inverted generational distance (IGD), hypervolume (HV)
Wang, Li and Li (2015)	Multi-objective	Hybrid	Multiple domains	10	180	3186	Area under curve (ROC)
Bouraoui, Jamoussi and BenAyed (2018)	Multi-objective	Hybrid	Multiple domains	8	60	208	Accuracy
Kundu and Mallipeddi (2022)	Multi-objective	Hybrid	Multiple domains	18	512	5856	Accuracy, McNemar's test

Each work in Table 3.1 is categorized according to the number of objectives optimized by the genetic algorithm (single-objective vs. multi-objective) and the categorization of the feature selection (FS) method, either as proposed by the authors or according to the definition by Ang et al. (ANG et al., 2015). Additionally, the table summarizes the characteristics of the datasets used to evaluate each method, highlighting the diversity of dataset domains, the number of datasets evaluated, and the dimensions and sample sizes of the dataset with the highest number of features. Finally, the metrics used to evaluate the final results of each method are also presented.

Aličković et al. (ALIČKOVIĆ; SUBASI, 2017) employ an ensemble approach

that combines a genetic algorithm for the first stage of feature selection with a rotation forest classifier for the subsequent stage in the task of breast cancer classification using two Wisconsin Breast Cancer datasets. This method improves the performance of simple classifiers such as Random Forests (RFs) and Support Vector Machines (SVMs), showing that simple models when paired with effective feature selection, can outperform more complex models and offer valuable insights for medical research. Similarly, Aalaei et al. (AALAEI et al., 2016) and Ahmad et al. (AHMAD et al., 2015) use wrapper approaches that integrate genetic algorithm-based feature selection with various classifiers to analyze datasets from the same Wisconsin Breast Cancer database, further demonstrating the enhancements provided by feature selection in this context.

Liu et al. (LIU et al., 2018) utilize a hybrid genetic algorithm for feature selection and classification optimization for gene selection and cancer diagnosis. This method was evaluated using five microarray datasets. The results were validated based on classification performance and the biological significance of the findings. The study demonstrated that the genetic algorithm could identify genes not detected by other evaluated approaches while surpassing them in terms of classification performance and dimensionality reduction.

Sayed et al. (SAYED et al., 2019) employ a nested Genetic Algorithm approach, where two GAs are used to evaluate different types of microarray data, thereby exploring their inherent interconnections. This method achieves exceptionally high performance in colon cancer classification and the report highlights the selected biomarkers' biological relevance. The validation extends to lung cancer datasets, where the approach again outperforms more straightforward methods in terms of classification performance. Maleki et al. (MALEKI; ZEINALI; NIAKI, 2021) use a single-objective hybrid approach that combines a nearest neighbors model to evaluate feature sets generated by a genetic algorithm performing feature selection for lung cancer classification. The study briefly discusses the implied correlation between the solution's performance in the classification task and its ability to uncover interesting data patterns relevant to early-stage lung cancer diagnosis.

While the mentioned works utilize GAs in their process and provide significant insights into the general usage of GAs in complex fields, none of them directly explore a multi-objective genetic algorithm (MOGA) structure such as the one proposed by Deb et al. (DEB et al., 2002), or other similar Pareto domination approaches. They deal with the dimensionality of the feature sets as either a parameter to be specified (such as the user deciding the number of features to be kept or removed), a secondary factor that is

improved as a result of the optimization of performance, or a metric that is combined along with classification performance into a single mixed objective, instead of evaluating it separately from the classification metrics as a unique goal that requires specialized optimization.

In this sense, Deng et al. (DENG et al., 2022) propose a hybrid solution that utilizes a multi-objective genetic algorithm to optimize feature sets initially selected by an XGBoost model (CHEN; GUESTRIN, 2016). This approach is employed for gene selection across 14 cancer datasets, evaluating the results using simple models such as Support Vector Machines (SVM) and Naïve Bayes. The study demonstrates improved performance, reduced dimensionality, and decreased execution time, even for datasets with up to 60,483 features. A similar study was previously conducted by Hasnat et al. (HASNAT; MOLLA, 2016) on a smaller scale, applying a multi-objective GA to optimize an initial feature selection performed by a correlation filter layer. This approach was evaluated against three cancer datasets containing up to 7,129 features, yielding promising results. An earlier example is the work of Tan et al. (TAN; LIM; CHEAH, 2014), where a multi-objective GA was applied to datasets with up to 32 features. This study included two disease classification problems and a human motion detection and classification problem, showing improved results compared to the initial model performance and providing a detailed analysis of the Pareto-domination behavior of the generated solutions. In all three cases, the primary objectives are dimensionality reduction and enhancing one or more classification metrics.

Xue et al. (XUE et al., 2021) propose multi-objective genetic algorithms for optimizing classification, specifically focusing on crossover operators to enhance the exploration of the solution search space. Their solution is tested on 10 datasets, including those in cancer research and other fields, yielding promising results. However, these datasets contain fewer features (up to 649) and samples than most other experiments mentioned. A similar situation was observed in other related works evaluated on datasets with limited dimensions. For instance, Wang et al. (WANG; LI; LI, 2015) tested a multi-objective GA emphasizing redundancy reduction on datasets with up to 180 features. Similarly, Bouraqui et al. (BOURAOUI; JAMOUSSI; BENAYED, 2018) employed a multi-objective GA for feature selection and model optimization on datasets with up to 60 features.

Recently, Kundu et al. (KUNDU; MALLIPEDDI, 2022) proposed a multi-objective GA that leverages prior knowledge obtained from various classic feature selection meth-

ods within a MOGA framework to optimize feature sets across multiple domains, similar to the approach proposed in this work. Their method utilizes this information during the initial population generation process by using the top features to form a small portion of the initial population of feature subsets. In contrast, the remaining feature subsets are created using random features. The underlying hypothesis is that the standard genetic algorithm process will be enough to replicate and preserve the genes representing important features as generations progress. Although the results presented are promising, the evaluations were conducted on datasets with relatively low dimensionality (up to 512 features). Moreover, the study does not elaborate on how previous knowledge could be utilized in other crucial genetic algorithm operations beyond samplings, such as mutations and crossovers.

From the works mentioned in Table 3.1, it is notable that not all methods are evaluated on datasets with a substantial number of features (more than a thousand), even though all of them are evaluated using real-life data. These datasets typically have small sample sizes, and most evaluations are performed using traditional classification metrics such as accuracy, precision, recall, and F-score. Among these, accuracy is the most frequently used metric despite being potentially problematic. This is because many real-life datasets have imbalanced class distributions, and accuracy can be a misleading performance indicator for minority classes in these scenarios (ALI; SHAMSUDDIN; RALESCU, 2013). Additionally, while all the mentioned methods utilize GAs for various purposes, not all employ multi-objective approaches. Those that do, however, tend to present better results and provide more in-depth analysis across a wider variety of datasets compared to methods developed with a single objective in mind. From the works discussed, three key areas stand out as opportunities for exploration in a new combined solution:

1. Multi-objective approaches can separately evaluate two or more conflicting objectives, such as dimensionality reduction and classification performance improvement. This allows for greater flexibility in identifying the optimal compromise solution that yields the best results based on the relative importance assigned to each objective;
2. Applying an initial round of feature selection before optimizing feature sets with a GA can enhance classification performance and execution time by eliminating noisy features. Various methods can be employed in this initial round to enrich GA populations and offer different insights about the evaluated datasets. This information can be combined into mixed solutions using customized genetic algorithm

operators (sampling, mutation, crossover).

3. Ideally, the datasets utilized for evaluations should consist of real-world data, and the algorithm should demonstrate the ability to handle genuinely high-dimensional data comparable to other datasets used in similar studies. Evaluation should be conducted using classification metrics.

The approach outlined in this work capitalizes on these opportunities to shape the solution's foundation. Based on item 1, NSGA-II is used as a multi-objective heuristic to address two concurrent objectives: generating reduced feature sets for dimensionality reduction and enhancing classification performance by simplifying the data separation task. Based on item 2, the approach leverages prior knowledge provided by an initial round of feature selection, conducted using traditional methods, to reduce the exploration space of the proposed multi-objective GA through customized sampling, crossover, and mutation operators. This strategy ensures the utilization of relevant features while minimizing the computational burden of assessing potentially irrelevant ones. Lastly, based on item 3, the method is evaluated across a diverse array of real-world classification datasets, particularly within the life sciences domain. These datasets encompass challenging scenarios for classification and analysis, each tailored to specific research objectives, thereby demonstrating the versatility and effectiveness of the approach.

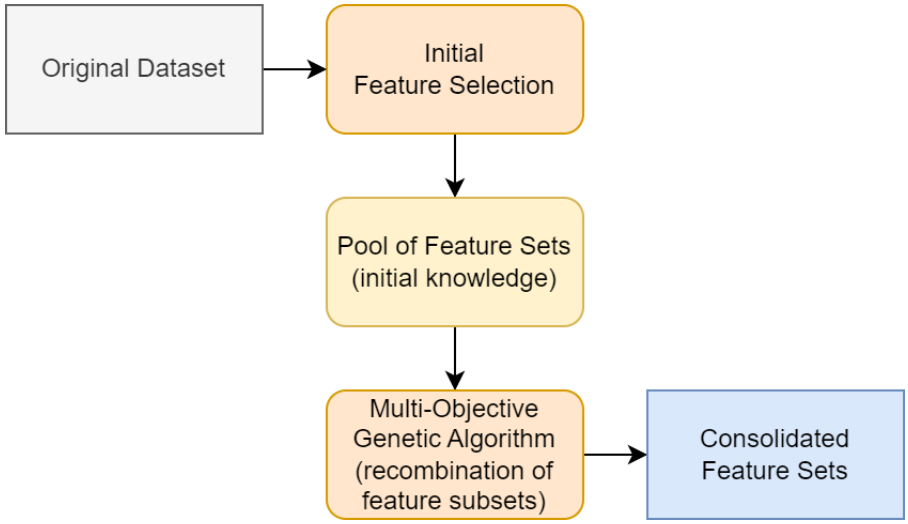
3.1 Chapter summary

In this chapter, a review of recent contributions in the literature surrounding single-objective and multi-objective genetic algorithms designed for feature selection was presented. Major faults in the current proposed methods were raised based on reviews of such literature and three items were proposed as opportunities to be explored in a new solution: employing multi-objective approaches to conciliate dimensionality reduction and classification performance, using classical feature selection methods as a source of knowledge for these approaches, and using real-world data to evaluate the performance of the proposed solution.

4 PROPOSED METHOD AND IMPLEMENTATION

In this chapter, a description of the proposed method is provided and each of the mechanisms and operators is explained in detail. The proposed method employs a multi-objective genetic algorithm based on NSGA-II, optimizing two measurable objectives: i) maximizing classification metrics for the given classification problem and ii) minimizing the number of features in the feature sets. Custom internal operators are designed to create and modify feature subsets across generations, leveraging initial knowledge obtained from other feature selection methods as a starting point for exploration within the multi-objective search space of possible solutions. The final result is a set of solutions that form a Pareto front, representing the best-performing feature subsets across a range of minimum to maximum feature counts. Figure 4.1 illustrates a simplified version of this process.

Figure 4.1: A simplified visual depiction of the multi-objective algorithm.



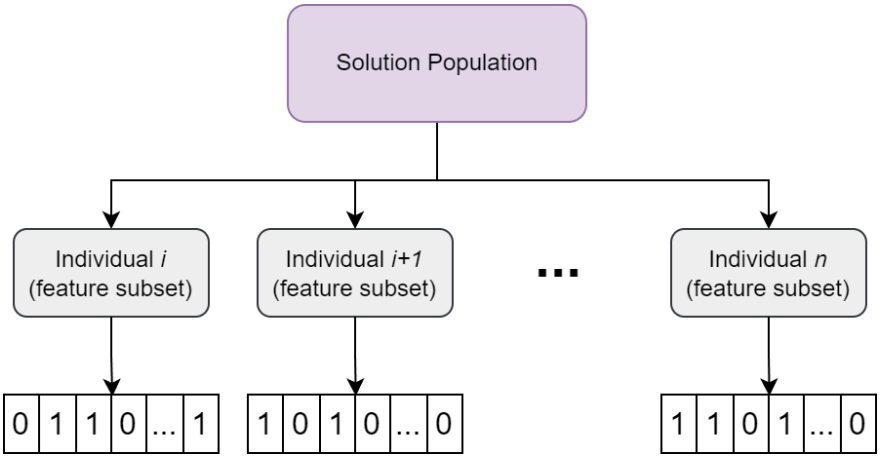
The following sections explain how the multi-objective genetic algorithm (NSGA-II) is employed in this context. They describe the generation and utilization of prior feature selection knowledge by the algorithm's customized internal operators and, finally, the evaluation of the feature sets.

4.1 Genetic algorithm structure

In the classic genetic algorithm structure, populations consist of individuals (candidate solutions to a problem) represented by a list of chromosomes (or genotypes). These

individuals are evolved and optimized through generations to maximize a fitness function. In the context of feature selection, individuals represent lists of features, with the genotype composed of a series of 0s and 1s indicating whether a chromosome (feature) is included (1) or excluded (0) in the feature subset represented by the individual. Figure 4.2 illustrates examples of individuals configured in this manner. During the recombination and mutation of individuals across generations, the genetic algorithm mixes, alters, and enhances candidate sets of features over time, guided by the fitness function.

Figure 4.2: A population of solutions in the proposed genetic algorithm.

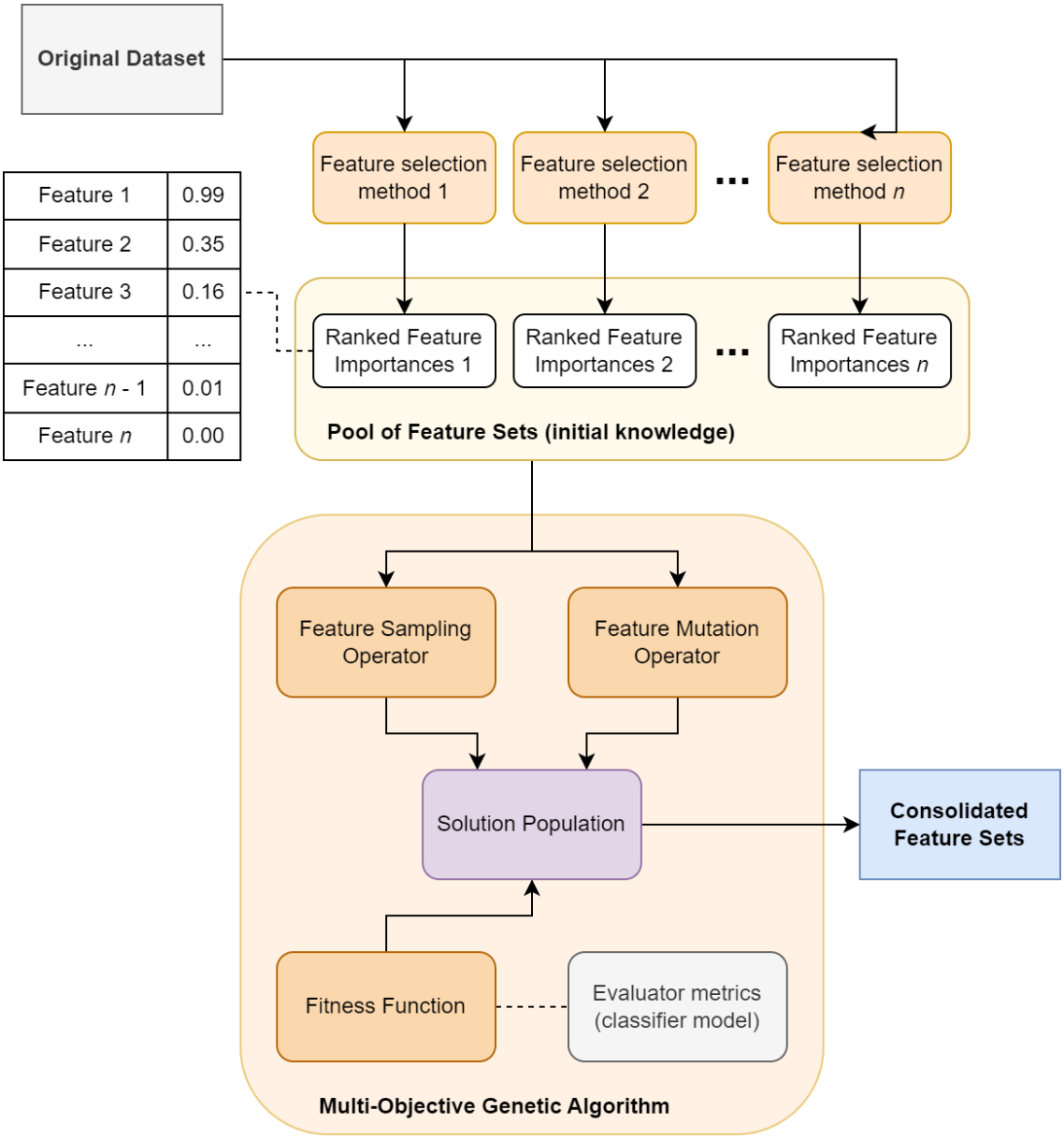


NSGA-II serves as the foundational framework for this purpose, incorporating classic components from the original method while introducing specialized operators for the following processes: 1) Solution Sampling: Custom operators generate the initial solutions in the population, ensuring a diverse starting point; 2) Mutation: Directed mutations in genotypes enforce diversity within the population, promoting exploration of the solution space; and 3) Fitness Function: Adapted specifically for classification problems, the fitness function evaluates the effectiveness of feature subsets in enhancing classification performance. These tailored operators enhance the algorithm's capability to address feature selection and classification optimization challenges.

The described multi-objective process optimizes the two previously defined objectives: maximizing classification metrics and minimizing the size of feature subsets. The operators manipulate portions of already reduced feature sets obtained from other feature selection methods, combining them in compositions of feature subsets potentially not fully explored by those methods. A more detailed depiction of the custom components, along with a representation of the initial knowledge input into the genetic algorithm, is shown in Figure 4.3.

Given NSGA-II's effectiveness in exploring and expanding Pareto fronts in large

Figure 4.3: A visual depiction of the multi-objective algorithm. In the image, a pool of feature sets containing different ranked lists of features sorted by feature importance values provides the candidate chromosomes for new individuals in the GA populations, through the sampling operator, and for mutation operations in existing individuals, through the mutation operator.



multidimensional search spaces (DEB et al., 2002), the algorithm has excellent potential for generating new and optimized feature subsets within the bi-dimensional search space consisting of the dual objectives of minimizing the number of features and maximizing classification potential.

The following sections first explain the role of feature importances provided by feature selection methods in the optimization process and how these importances are leveraged by the customized operators using a pool of feature sets. Subsequently, the implementation of these operators and their expected influence over the search space are described in detail.

4.2 Feature importances

For each evaluated dataset, the method requires a list of feature importances that can represent a priority order of all features in the dataset, as illustrated in the example table in Figure 4.3. These importances serve as the primary reference for deciding which features should be retained or discarded in feature sets when constraining their size. The feature importances can be obtained from traditional feature selection methods, extracted from classifiers, or sourced from equivalent methods.

Feature selection methods in the literature adopt diverse approaches with varying priorities for deciding which features to retain during dimensionality reduction. In our approach, we utilize a variety of feature selection methods to provide different perspectives on which features should be prioritized and retained during the feature-discarding process. A list of the default feature selection methods used to generate the feature importances for the multi-objective genetic algorithm is described in section 5.2.1.

4.3 Pool of Feature Subsets

When multiple feature selection methods are used to generate ordered feature sets based on feature importances, these individual ordered feature sets need to be stored somewhere to be used in the further steps of the process. This conceptual storage is referred to as a pool of feature sets, as depicted in Figure 4.3.

The concept of a pool of feature sets is utilized by the sampling and mutation operators implemented for the solution. The goal of this pool is to provide the algorithm

with multiple options of prioritized feature subsets, each selected by different methods. Within each feature set, features are sorted by the importance values assigned to them by the respective methods. This pool serves as a source of prior knowledge for both operators, guiding the genetic algorithm to explore solutions within a reduced and already optimized search space.

As a default, the implementation uses a pre-selected list of feature selection methods that explore different approaches to feature selection, each with different priorities that, as a result, generate diverse feature sets. The default methods used are listed in section 5.2.1 and described in further detail in section 2.1.2. This choice of feature selection methods used to generate the pool of feature sets is ultimately flexible and works with any feature selection method capable of providing a ranking or list of feature importances for a set of features.

4.4 Sampling operator

The sampling operator is used in the genetic algorithm to generate the entire initial population and new samples as needed. The sampling method consists of generating n_s distinct samples from the pool of feature sets, resulting in, by default, an equal proportion of individuals for each of the sets available in the pool. Each sample is generated by selecting a limited number n_f of features according to their importance value in the feature set. To enable variable solutions and to avoid excessive repetition of the selected genes, a randomness factor is applied to control the probability of selecting a feature based on its importance. This ensures that even though high-ranked features are prioritized in the sampling process, they are less likely to always be chosen for new samples if the probability factor is high enough.

The chance of a feature f being picked directly relates to its feature importance i_f, generated according to the feature selection method used, to the probability factor p, defined as a parameter, and to the maximum feature importance in the feature set $max(i)_{fs}$, as defined in equation 4.1,

$$selection\_chance_f = \frac{i_f}{(1+p) * max(i)_{fs}} \tag{4.1}$$

where p values greater than zero ensure that even top-ranked features don't get selected every time.

To properly explore the size constraint of the problem at hand — reducing the sizes of feature sets —, feature sets are created with limited n_f features, a random value between a minimum min_f value and a maximum max_f value. The complete sampling process is described in Algorithm 5.

In the algorithm, for each feature set F in the pool of feature sets F_pool, a proportion $prop_F$ of individuals is sampled (lines 1 to 7), resulting in n_s*$prop_F$ individuals for each feature set F. Thus, n_s*$prop_F$ iterations j are performed (lines 2 to 6), where a random integer value n_f is picked between min_f and max_f (line 3), and used to generate a new individual $indv_j$ representing a subset of n_f features from the original feature set F (line 4), where each feature f can be selected with probability $selection\_chance_f$. Finally, the individual is added to the new set of $individuals$ (line 5) and the process repeats for the remaining individuals.

Algorithm 5: The sampling process. A total of n_s samples or $individuals$ are generated, each representing a set of n_f features between min_f and max_f features. Features are selected from the pool of feature sets according to probability $selection\_chance_f$, described in equation 4.1.

Data: F_pool, $proportions$, n_s, max_f, min_f, p
Result: $individuals$
1 **for** F, $prop_F$ ***in*** *(F_pool, $proportions$)* **do**
2 **for** j ***in*** n_s*$prop_F$ **do**
3 pick random integer n_f between min_f and max_f;
4 generate $indv_j$ by selecting n_f features f from F with chance $selection\_chance_f$;
5 add $indv_j$ to $individuals$ list;
6 **end**
7 **end**
8 return $individuals$;

The resulting initial population is expected to have a high diversity of feature sets, since it is built by combining a proportional number of samples derived from each of the different initial feature importance sets in the feature pool, further diversified by random set sizes.

4.5 Mutation operator

The mutation operator is used in the genetic algorithm to modify existing solutions by inserting variations in their composition in an attempt to increase fitness.

The proposed mutation operator acts by including and removing features from the target individuals representing feature subsets. The mutation process begins by picking, during each iteration of mutations, a random ordered feature set from the pool of feature sets to serve as a source of chromosomes (in this case, features) for the mutation in target individuals. Once chosen, random features are selected from the best-ranked max_f features of the selected ordered set to be activated in each target individual selected for mutation. The mutation process ensures that the activated chromosome is not already enabled in the target individual and, in turn, turns off another random chromosome to preserve the size constraint of the solution. The complete mutation process is translated to Algorithm 6.

In Algorithm 6, first a random feature set F is selected from F_pool (line 1), and a $chromosome\_pool$ is created (line 2) from the top max_f features according to previously ranked feature importances i_f. Then, for each individual $indv$ from the list of individuals $mutation\_candidates$, a random inactive feature f_1 and a random active feature f_2 are selected to be activated, and deactivated, respectively (lines 4 to 7). After that, mutated individual $indv$ is added to the list of $mutated\_individuals$ and the process is repeated for each of the remaining target individuals.

Algorithm 6: The mutation process. A group of $mutation\_candidates$ is mutated by activating previously inactive features and disabling previously enabled features. The features chosen for activation are sourced from the pool of feature sets.

Data: F_pool, $mutation\_candidates$, max_f
Result: $mutated\_individuals$
1 pick random F from F_pool;
2 generate $chromosome\_pool$ from top max_f features of F;
3 **for** $indv$ ***in*** $mutation\_candidates$ **do**
4 pick random feature f_1 from $chromosome\_pool$ not active in $indv$;
5 pick random feature f_2 active in $indv$;
6 mutate $indv$ by activating f_1;
7 mutate $indv$ by deactivating f_2;
8 add $indv$ to $mutated\_individuals$;
9 **end**
10 return $mutated\_individuals$;

The result of using this limited set of chromosomes for exploration is that mutations are not entirely random but instead directed to exploring features with measured importance in the classification task while maintaining a degree of freedom for possibly unexplored feature combinations to be explored.

4.6 Fitness function

The fitness function in a genetic algorithm is used to objectively evaluate an individual or solution within the context of the problem being solved, returning a fitness score that helps compare solutions and prioritize the most fit for survival. For a classification problem, the fitness function must measure the effectiveness of a subset of features against the classification dataset. Given that numerous individual solutions are created and evaluated across the generations of the genetic algorithm, the fitness function needs to be efficient. Traditional classification metrics such as precision, recall, accuracy, F1-Score, or use-case-specific metrics can be employed for this purpose.

Any classifier can generate these metrics, but when aiming for efficiency and evaluation speed, generic, simple classifiers such as support vector machines, k-nearest neighbors, and others are the typical choices for evaluation in feature selection for classification (PIRI et al., 2023). They can be trained against a dataset containing the subset of features specified by each individual and evaluated with traditional classification metrics. One or more of these metrics can then be used as outputs of the fitness function, providing the necessary information to decide whether solutions are fit for the problem or not in terms of classification effectiveness.

The other important goal to consider in the multi-objective approach is dimensionality reduction, so a parallel objective in the fitness evaluation process should be to minimize individual feature subset size. Thus, the fitness function employed for the proposed method evaluates the two objectives: i) maximizing classification metrics (for example, F1-Score) and ii) minimizing the number of features in the feature sets. The classification metric and the classifier used in the experimentation process are defined in chapter 5 and could be changed to better suit any target classification problem.

Considering that NSGA-II expands the solution fitness of dominating individuals through its crowding distance mechanism on both dimensions of optimization, the result is a group of solutions evenly distributed across the different possible feature set sizes on one dimension, while maximizing the classification metric on another dimension by keeping only the best solutions for each feature set size.

4.7 Chapter summary

This chapter provided a detailed explanation of the proposed method, which leverages a multi-objective genetic algorithm structure to optimize feature sets of independent feature selection methods for classification problems.

The multiple sections explain the individual parts that compose the algorithm, highlighting differences from the original NSGA-II structure used as a basis, and detailing the logic and reasoning behind the proposed operators for the sampling process (section 4.4), the mutation process (section 4.5), and the fitness function (section 4.6).

5 EXPERIMENTS

In this chapter, the effectiveness of the proposed method is validated using different high-dimensional datasets. The datasets are described in section 5.1, and the experiments are described in section 5.2. These experiments aim to demonstrate the improvement in classification performance achieved by the optimized feature sets created by the multi-objective GA, compared to traditional feature selection methods. The results for each dataset are discussed individually in chapter 6.

5.1 Datasets

The datasets considered for evaluating the proposed method contain a large number of features, each with varying degrees of relevance to the classification task. Each dataset comes from applications in the field of life sciences and benefits from feature selection from a classification and an analytical perspective, revealing interesting patterns that are not always obvious to researchers. Thus, in these experiments, we consider the classification performance of the selected features in each dataset as a direct indicator of their analytical relevance in the real-world problem at hand. In this sense, feature selection highlights relevant features in the dataset, which can be candidates for more in-depth analysis.

Table 5.1 and the following subsections summarize the datasets used for experimentation, explaining the context they are used in and how they benefit from the proposed method compared to other feature selection methods.

5.1.1 CuMiDa - Leukemia and Breast Cancer datasets

Gene expression data was utilized to assess the effectiveness of the proposed method. We employed two datasets sourced from the CuMiDa database (FELTES et al., 2019): Leukemia (GSE28497) and Breast Cancer (GSE70947) (refer to Table 5.1). These datasets represent high-quality curated microarray data sets tailored specifically for evaluating machine learning solutions in cancer research. Notably, they offer the advantage of employing more contemporary techniques than other commonly used cancer datasets (GRISCI et al., 2024). Prior to analysis, the dataset underwent thorough preprocessing

Table 5.1: Dataset details and label distributions

Dataset	N. Features	N. Samples	N. Classes	Label	Count	Proportion
CuMiDa Leukemia (GSE28497)	22284	281	7	B-CELL_ALL	74	26.3 %
				B-CELL_ALL_ETV6-RUNX1	53	18.9 %
				B-CELL_ALL_HYPERDIP	51	18.1 %
				B-CELL_ALL_T-ALL	46	16.4 %
				B-CELL_ALL_TCF3-PBX1	22	7.8 %
				B-CELL_ALL_HYPO	18	6.4 %
				B-CELL_ALL_MLL	17	6.0 %
CuMiDa Breast Cancer (GSE70947)	35982	289	2	normal	146	50.5 %
				breast_adenocarcinoma	143	49.5 %
Eye Color SNPs	126018	500	3	2V-M-CC	201	40.2 %
				3CE-PR	190	38.0 %
				1AZC-AZE	109	21.8 %
Arrhythmia (UCI repository)	279	438	9	Normal	245	55.9 %
				Right bundle branch block	50	11.4 %
				Ischemic changes	44	10.0 %
				Sinus bradycardy	25	5.7 %
				Others	22	5.0 %
				Old Anterior Myocardial Infarction	15	3.4 %
				Old Inferior Myocardial Infarction	15	3.4 %
				Sinus tachycardy	13	3.0 %
				Left bundle branch block	9	2.1 %
p53 Mutants (UCI repository)	5409	16772	2	inactive	16449	99.1 %
				active	144	0.9 %

steps, including background correction, normalization, and quality assessment of samples. Furthermore, manual editing was conducted to eliminate erroneous probes. Baseline performances of fundamental classifiers are provided as reference points (FELTES et al., 2019).

Feature selection can be applied to these use cases to reduce the considerable amount of features available in the datasets (22284, and 35982 features, respectively) and increase classification performance, as well as identify highly relevant features for cancer research. It is noteworthy that despite the abundance of features, both datasets exhibit small sample sizes, a common issue across diverse domains of life sciences research. Furthermore, while the Breast Cancer dataset contains only two classes evenly distributed in the data, the Leukemia dataset contains seven different classes with uneven distribution, with the majority class representing 26.3% of the samples, and the minority class only 6.0%. This imbalance in label distributions, coupled with small sample sizes, poses significant challenges for learning algorithms (HE; GARCIA, 2009).

5.1.2 Eye Color SNPs dataset

The proposed method was also evaluated using a SNP (single nucleotide polymorphisms) dataset. This dataset, which is not publicly available, was utilized to investigate SNPs associated with phenotypes such as eye color in humans (KAYSER, 2015), with a focus on Brazilian populations. Identifying informative SNPs is crucial for understanding the human genome and uncovering correlations between SNPs and various phenotypes, conditions, and diseases. Feature selection plays an essential role in this identification process. The dataset comprises 126,018 features, each representing one of the two alleles of a selection of SNPs from specific genes across 500 samples, as detailed in Table 5.1.

The Eye Color SNPs dataset presents a formidable challenge for learning algorithms, primarily due to its exceptionally high feature count coupled with a limited sample size. The abundance of features often leads learning models to prioritize noisy data during the training process, resulting in prolonged training times for common classification models and degraded performance (HE; GARCIA, 2009). Additionally, the dataset exhibits some class imbalance, with the majority class comprising 40.2% of the samples and the minority class only 21.8%.

5.1.3 UCI Database datasets

The proposed method was additionally evaluated using two publicly available life sciences datasets sourced from the UCI machine learning repository (KELLY; LONGJOHN; NOTTINGHAM, 2021): the Arrhythmia dataset (GUVENIR et al., 1998) and the P53 mutants dataset (LATHROP, 2010), as outlined in Table 5.1.

The first dataset, the Arrhythmia dataset (GUVENIR et al., 1998), is employed to discern the presence and subtypes of cardiac arrhythmia across 16 different labels. This relatively small dataset comprises 452 samples and 279 features (GUVENIR et al., 1998). However, due to the scarcity of samples (less than 5) for certain classes, the dataset was condensed to focus on the 9 most significant classes for this experiment, resulting in 438 samples and 279 features. In the second dataset, features derived from biophysical models of mutant p53 proteins are leveraged to predict p53 transcriptional activity, with all class labels determined through *in vivo* assays (LATHROP, 2010). All samples and features are included in the experimentation process for this particular use case, with 16772 samples and 5409 features.

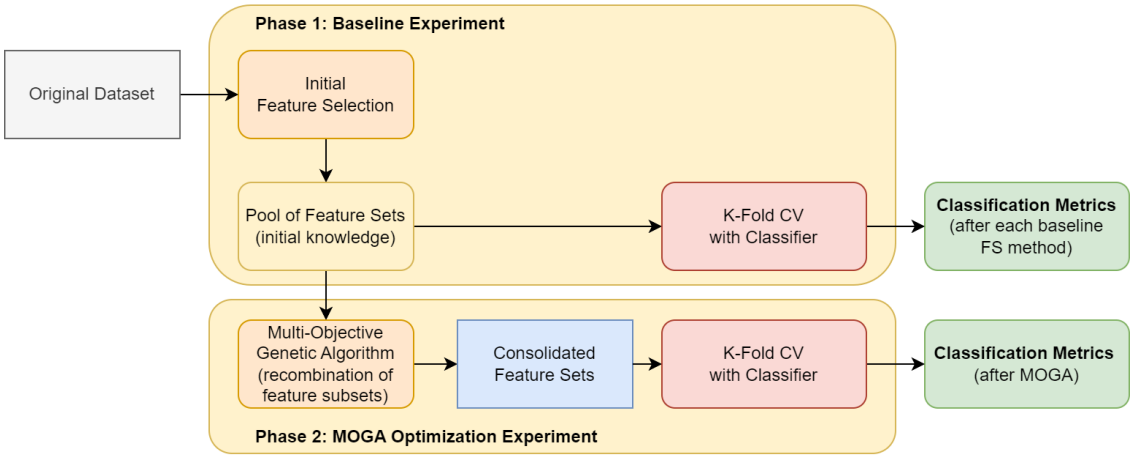
The Arrhythmia dataset stands out among the datasets examined in these experiments due to its comparably small feature count and the most pronounced class imbalance issue in terms of sample counts. The majority class constitutes 55.9% of the utilized samples, while the minority class comprises only 2.1%, totaling merely 9 samples. In this scenario, it is likely that the performance on minority classes will be significantly affected by the small sample size.

In contrast, the p53 Mutants dataset boasts the largest sample size among all datasets, accompanied by a substantial number of features. Despite this, the dataset also has severe class imbalance issues, although it contains a greater number of samples for the minority class compared to the other highly imbalanced datasets analyzed in the experiments. In this case, higher execution times can be expected compared to other datasets due to its larger sample sizes, a factor that often increases the training time required by classifiers used in the fitness evaluation process of the proposed method.

5.2 Experiment design

The experiments are divided into two phases. The first phase represents the baseline experiments, and it generates ranked feature importances for the pool of feature sets using a set of feature selection methods. Additionally, classification metrics are generated for these ranked features using a classifier in a cross-validation process, a data resampling method used to evaluate models with less over-fitting impact than simple training and test data splits (BERRAR, 2019). Moving to the optimization phase, the ranked feature importances obtained from the baseline phase serve as input for the multi-objective genetic algorithm. Subsequently, the resulting optimized feature sets undergo evaluation using the same cross-validation procedure employed in the baseline phase. The metric chosen for the evaluation is the macro-averaged F1-Score, a metric adequate for datasets with imbalanced or balanced distributions of classes and derived from recall and precision (ALI; SHAMSUDDIN; RALESCU, 2013). Figure 5.1 provides a simplified representation of the process employed in each experiment phase, with further elaboration provided in the subsequent subsections.

Figure 5.1: A depiction of the experiment plan for the proposed method, divided in two phases.



5.2.1 Baseline experiments

The first phase of the experiment process (illustrated in Figure 5.1) involves creating the pool of feature sets. This is accomplished by applying a list of feature selection methods to a given dataset, generating importance values for all of its features. Once generated, the feature importances are used to decide which features to retain or to remove when reducing the dimensionality of the feature sets for the classification evaluations.

The list of feature selection methods to be used by the implementation is flexible as long as they can provide feature importance. The more methods we use, the more variety of initial knowledge we provide to the GA. Eight feature selection methods are used for the experiments, all of which are summarized in Table 5.2. ANOVA F-Test (KIM, 2017) and Kruskal Wallis Test (MCKIGHT; NAJAB, 2010) are filter methods used for efficient univariate analysis of features. Decision Tree (SONG; YING, 2015), Lasso (TIBSHIRANI, 1996), Linear SVM (GUYON et al., 2002), and Random Forest (SPEISER et al., 2019) are classic learning models that can provide importance weights obtained during the learning process, thus serving as an embedded approach capable of uncovering multivariate relations and requiring negligible computational time in most use cases. mRMR (PENG; LONG; DING, 2005), Mutual information (ROSS, 2014), and Relief-F (KONONENKO, 1994) are also filter methods but perform much more complex, sometimes multivariate analyses, with a particular focus on feature interactions and reduction of redundancy in the feature sets. Method implementations used in the experiments were sourced from the works of Pedregosa et al. (PEDREGOSA et al., 2011), Urbanowicz et

al. (URBANOWICZ et al., 2017) and Mazzanti (MAZZANTI, 2023).

Table 5.2: List of feature selection methods used in the baseline experiments

Method	Method Type	Variable Analysis Type
ANOVA F-Test	Filter	Univariate
Kruskal Wallis Test	Filter	Univariate
Decision Tree	Embedded	Multivariate
Lasso	Embedded	Multivariate
Linear SVM	Embedded	Multivariate
Random Forest	Embedded	Multivariate
mRMR	Filter	Multivariate
Mutual information	Filter	Univariate
Relief-F	Filter	Multivariate

Each feature selection method is then evaluated for each dataset and all feature subsets within a minimum and maximum number of features based on the ranking of features provided by the feature selection method. The values that specify the numbers of features to be evaluated are defined, in these experiments, as a range of integers starting from a minimum number of features to a maximum number of features, as described in the parameters in Table 5.3. In other words, with the parameter values in the Table 5.3, ranked subsets of features are evaluated using a minimum of 2 to 50 features from the list of ordered features. These respective parameters could be changed to increase or reduce computation time since evaluating more sets of different numbers of features will require additional processing. The parameters were empirically set to a standard pair of the minimum and maximum number of features across all datasets for an easier comparison. However, they could be adapted to each use case for more efficient results.

A stratified k-fold cross-validation approach (STONE, 1974) was used during the experiments to stabilize classification metrics by aggregating evaluations of different folds of data. For additional stability of the metrics, each execution, starting at the importance generation step and ending with the cross-validation step, is repeated 10 times. The metrics provided as a result are the averaged values across all runs. Table 5.3 specifies the number of folds and runs utilized for the datasets and also describes the target classification metric and evaluator used in the experiment to evaluate metrics for the resulting feature sets — in this case, an SVM, which provides the F1-Score values to the fitness evaluation. For the case of the Arrhythmia dataset, a smaller number of folds was chosen due to the limited amount of samples available (5) for its minority class.

Table 5.3: Baseline experiments parameters

Dataset	Parameter	Value
All datasets (common)	Min. Features	2
	Max. Features	50
	Target Metric	F1-Score (Macro)
	Evaluator	Linear SVM
	Runs	10
CuMiDa Leukemia (GSE28497)	CV folds (K-Fold)	15
CuMiDa Breast Cancer (GSE70947)	CV folds (K-Fold)	15
Eye Color SNPs	CV folds (K-Fold)	15
Arrhythmia (UCI repository)	CV folds (K-Fold)	5
p53 Mutants (UCI repository)	CV folds (K-Fold)	15

5.2.2 Multi-objective genetic algorithm optimization experiments

In the second phase of the experiment process (depicted in Figure 5.1), the focus is on optimizing feature sets using the MOGA and evaluating its outcomes. This phase begins by utilizing the feature importance values acquired during the baseline experiments in Phase 1. These values are stored in the corresponding pool of feature sets created for each dataset and serve as a resource in the genetic algorithm. The GA is executed for each dataset 10 times, repeated to ensure result stability and the resulting metrics represent average values across all runs.

This step replicates most of the parameters from the baseline experiments (Table 5.3) in Table 5.4: the ranges of numbers of features, the evaluators used for classification, the number of folds in cross-validation, and the target metric. Additionally, other parameters specific to the execution of the genetic algorithm are included in the new table. The final results are presented in the next chapter.

Table 5.4: Multi-objective genetic algorithm optimization experiments parameters.

Dataset	Parameter	Value
All datasets (common)	Min. Features	2
	Max. Features	50
	Target Metric	F1-Score (Macro)
	Evaluator	Linear SVM
	Runs	10
CuMiDa Leukemia (GSE28497)	CV folds (K-Fold)	15
	Generations	200
	Population size	200
	Prob. Factor	1.25
CuMiDa Breast Cancer (GSE70947)	CV folds (K-Fold)	15
	Generations	200
	Population size	200
	Prob. Factor	1.25
Eye Color SNPs	CV folds (K-Fold)	15
	Generations	400
	Population size	200
	Prob. Factor	1.25
Arrhythmia (UCI repository)	CV folds (K-Fold)	5
	Generations	200
	Population size	200
	Prob. Factor	1.25
p53 Mutants (UCI repository)	CV folds (K-Fold)	15
	Generations	400
	Population size	210
	Prob. Factor	1.25

5.3 Chapter summary

In this chapter, section 5.1 describes the datasets selected for the process of evaluation of the method previously proposed in chapter 4, in the context of classification problems. The selected datasets represent high-dimensional data from real-life applications of medical and biology research fields and are mostly available for online access.

Section 5.2 describes the experiments designed to evaluate the application of the method in each of the datasets, separating the comparison into two phases: evaluating the performance before the optimization provided by the MOGA, and after the optimization.

6 RESULTS AND DISCUSSION

In this chapter, results are presented for the experiments performed for each dataset. After completing all experiments, the outcomes obtained from the initial baseline feature selection methods employed are compared with the results derived from the optimization conducted through the multi-objective genetic algorithm in the subsequent phase for each specific use case. Additionally, the final section delves into the discussion of runtime performances.

6.1 Classification performances

Tables 6.1, 6.2, 6.3, 6.4, and 6.5 display the aggregated statistics across multiple runs conducted for each dataset. All tables present, for each method, the average performance of macro-averaged F1-Score over the test set during cross-validations for the range of numbers of features explored within the minimum and maximum boundaries defined in parameter tables 5.3 and 5.4, for the number of runs specified in the same tables. As a complement, Figures 6.1, 6.2, 6.3, 6.4, and 6.5 present aggregated statistics for the same executions across each dataset. These figures showcase the Pareto fronts of dominant solutions generated during the multi-objective process, depicted in blue. Additionally, they illustrate the individual performances of the initial feature selection methods, along with a box plot summary of their averaged scores. For reference, the performance of the classifier considering all available features is represented as a horizontal dashed line.

The results for the CuMiDa Leukemia and CuMiDa Breast cancer datasets are summarized in Tables 6.1 and 6.2, respectively. Across the evaluations, the Genetic Algorithm (GA) executions consistently enhanced the performance of the target metric during the cross-validation process. Notably, for the Leukemia dataset, the average performance surpassed that of the best baseline method by approximately 10.01 points (or 13.22%), with similar improvements observed in both the maximum (by 5.37 points) and minimum results (by 8.79 points) across all feature numbers, while maintaining a relatively low standard deviation. Regarding the Breast cancer dataset, the GA performance exceeded that of the best baseline method by approximately 1.93 points (or 2.05%), with improvements seen in both the maximum (by 2.02 points) and minimum results (by 1.73 points) across all feature numbers, accompanied by the lowest standard deviation.

An alternative visual representation of the results presented in Tables 6.1 and 6.2

is provided in Figures 6.1 and 6.2. In these figures, the performance of the dominant feature sets generated by GA is highlighted in blue across the range of evaluated feature numbers. The dominant GA solutions consistently outperform the initial feature selection results utilized in the GA optimization process across nearly all evaluated feature numbers. Moreover, they surpass the performance of the baseline classifier, which incorporates all features, with fewer features than the initial feature selection methods.

For the Eye Color SNPs dataset, Table 6.3 exhibits a similar trend of enhancements across all metrics of the target metric. There is an improved average performance by approximately 6.15 points (or 8.55%), alongside significantly improved minimum (by 5.4 points) and maximum (by 8.22 points) performances across varying numbers of features, all while maintaining a comparably low standard deviation.

Figure 6.3 displays the same results presented in table 6.3 but highlights an interesting behavior. In this particular dataset, the performances of the dominant GA-generated solutions are closer to the performance of the initial feature selection methods when evaluating smaller feature set sizes. This is expected since the evaluated dataset contains a small subset of highly representative features and a wide majority of less representative, complementary features. Still, the GA manages to fine-tune the feature sets better as the number of features increases. In contrast, most of the initial methods fail to identify the less representative but still important secondary features.

The performance of the methods on the last two datasets, the UCI p53 Mutants dataset, and the UCI Arrhythmia dataset, are presented in tables 6.4 and 6.5, respectively. In both use cases, the dominant GA performances highly surpass the average, minimum, and maximum performances. For the first dataset, p53 Mutants, the average performance is improved by approximately 7.33 points (or 11.45%), while the minimum performance is improved by 2.5 points and maximum performance is improved by 8.13 points. For the second dataset, Arrhythmia, the average performance is improved by approximately 9.76 points (or 21.12%), while the minimum performance is improved by 8.78 points and maximum performance is improved by 4.12 points.

Figures 6.4 and 6.5 further detail the performances on these two datasets, highlighting that not only was the GA performance better on average, minimum, and maximum performances, but also across all feature numbers evaluated by the experiments, for both use cases. While for p53 Mutants no method achieves the same performance as obtained by the original set of 5409 features, the GA approach nearly reaches the same result with 50 features, and the remaining methods fall largely behind. For Arrhythmia,

Table 6.1: CuMiDa Leukemia dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.

Method	Avg. (± Std. Dev.)	Max.	Min.
ANOVA F-Test	0.6414 (± 0.0067)	0.7759	0.2037
Decision Tree	0.7574 (± 0.0073)	0.7863	0.3364
Kruskal Wallis	0.7177 (± 0.0070)	0.8230	0.3421
Lasso	0.7295 (± 0.0080)	0.8796	0.2712
Linear SVM	0.7439 (± 0.0122)	0.8720	0.2253
Mutual Information	0.7401 (± 0.0075)	0.8027	0.3604
NSGA-II Optimization	**0.8575 (± 0.0099)**	**0.9333**	**0.4685**
Random Forest	0.6758 (± 0.0411)	0.7807	0.2992
Relief-F	0.7309 (± 0.0063)	0.7766	0.3806
mRMR	0.7109 (± 0.0047)	0.7825	0.3211

Figure 6.1: Average macro F1-Score performance with varying numbers of features for CuMiDa Leukemia dataset.

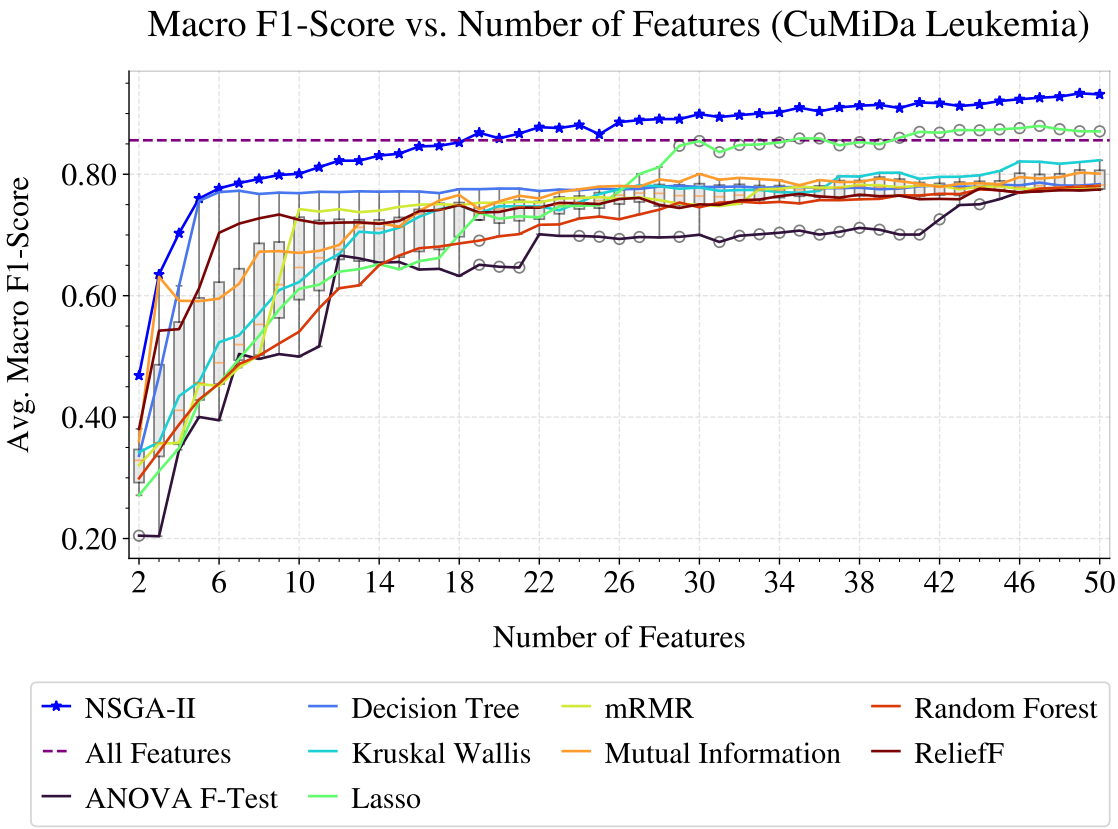


Table 6.2: CuMiDa Breast Cancer dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.

Method	Avg. (± Std. Dev.)	Max.	Min.
ANOVA F-Test	0.8873 (± 0.0046)	0.9009	0.8657
Decision Tree	0.8763 (± 0.0080)	0.8853	0.8698
Kruskal Wallis	0.8892 (± 0.0041)	0.9096	0.8595
Lasso	0.9423 (± 0.0030)	0.9591	0.8379
Linear SVM	0.8909 (± 0.0066)	0.9373	0.7941
Mutual Information	0.8765 (± 0.0044)	0.8953	0.8338
NSGA-II Optimization	**0.9616 (± 0.0029)**	**0.9793**	**0.8871**
Random Forest	0.8626 (± 0.0149)	0.8805	0.8358
Relief-F	0.8364 (± 0.0055)	0.8458	0.8177
mRMR	0.8933 (± 0.0041)	0.9081	0.8228

Figure 6.2: Average macro F1-Score performance with varying numbers of features for CuMiDa Breast Cancer dataset.

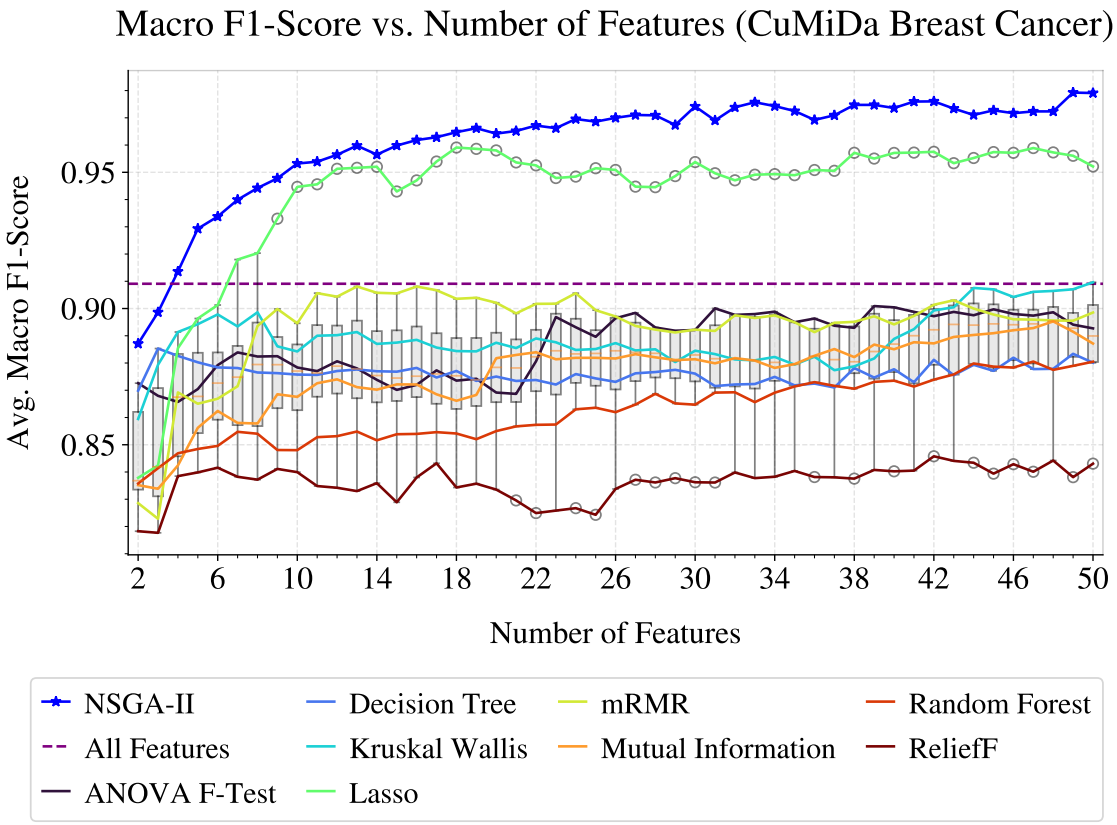


Table 6.3: Eye Color SNPs dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.

Method	Avg. (± Std. Dev.)	Max.	Min.
ANOVA F-Test	0.6375 (± 0.0064)	0.6602	0.4672
Decision Tree	0.7195 (± 0.0109)	0.7354	0.5784
Kruskal Wallis	0.6360 (± 0.0067)	0.6599	0.4681
Lasso	0.7140 (± 0.0074)	0.7479	0.6310
Linear SVM	0.6330 (± 0.0072)	0.6607	0.4669
Mutual Information	0.6500 (± 0.0177)	0.6714	0.5188
NSGA-II Optimization	**0.7810 (± 0.0072)**	**0.8301**	**0.6324**
Random Forest	0.6468 (± 0.0210)	0.6747	0.5385
Relief-F	0.6087 (± 0.0067)	0.6444	0.4671
mRMR	0.6722 (± 0.0068)	0.7020	0.5119

Figure 6.3: Average macro F1-Score performance with varying numbers of features for Eye Color SNPs dataset.

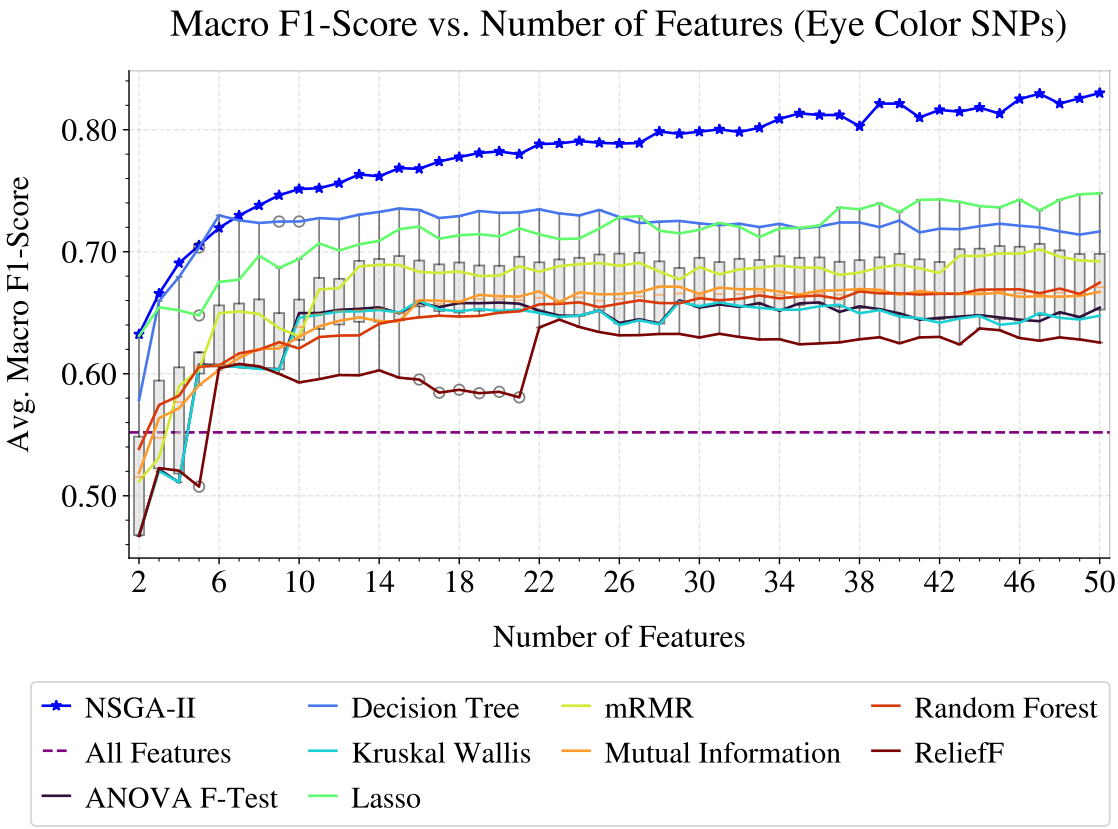


Table 6.4: p53 Mutants dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.

Method	Avg. (± Std. Dev.)	Max.	Min.
ANOVA F-Test	0.5798 (± 0.0036)	0.6318	0.4977
Decision Tree	0.5483 (± 0.0121)	0.5995	0.4978
Kruskal Wallis	0.5117 (± 0.0041)	0.5626	0.4977
Lasso	0.6400 (± 0.0062)	0.6952	0.5526
Linear SVM	0.5076 (± 0.0026)	0.5503	0.4977
Mutual Information	0.5299 (± 0.0080)	0.6026	0.4977
NSGA-II Optimization	**0.7133 (± 0.0126)**	**0.7765**	**0.5776**
Random Forest	0.5848 (± 0.0203)	0.6643	0.4977
Relief-F	0.5108 (± 0.0020)	0.5687	0.4977
mRMR	0.6080 (± 0.0051)	0.6684	0.5009

Figure 6.4: Average macro F1-Score performance with varying numbers of features for p53 Mutants dataset.

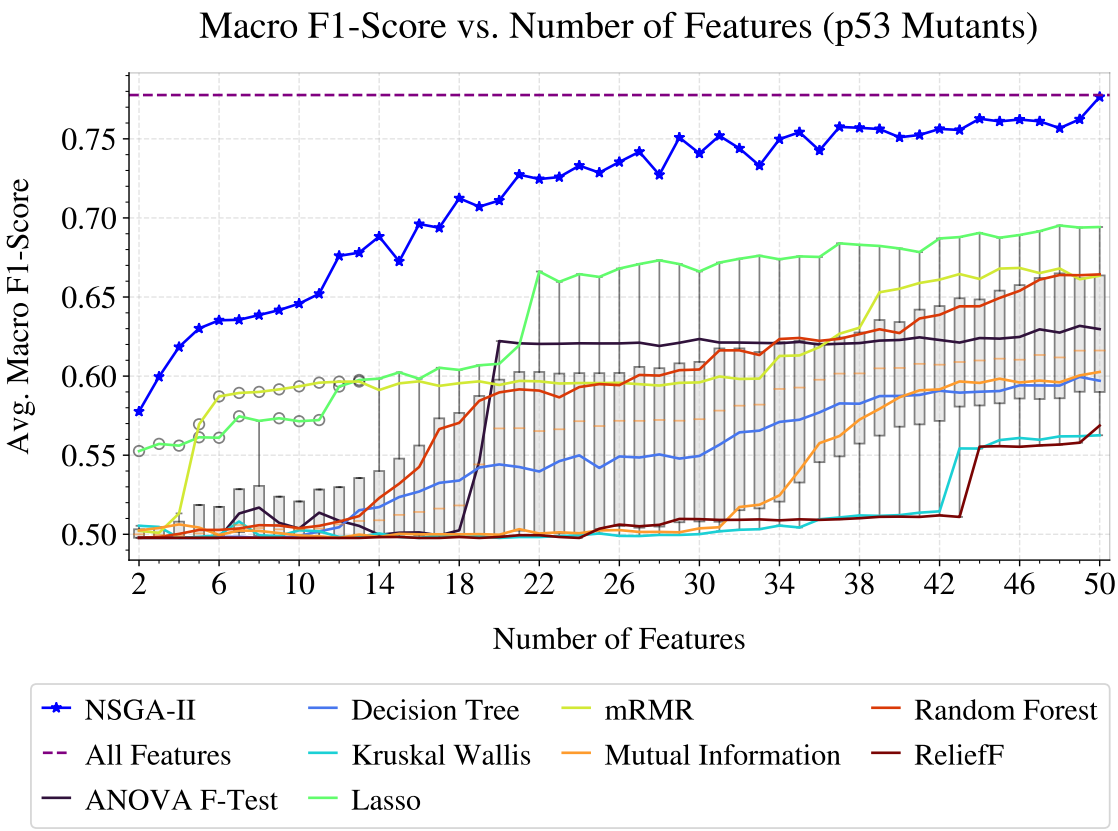


Table 6.5: Arrhythmia dataset results. Aggregations over Test F1-Score (macro-averaged) across 10 executions. The best results are in bold.

Method	Avg. (± Std. Dev.)	Max.	Min.
ANOVA F-Test	0.4109 (± 0.0086)	0.5217	0.1795
Decision Tree	0.4184 (± 0.0232)	0.4831	0.1527
Kruskal Wallis	0.4136 (± 0.0116)	0.4716	0.1799
Lasso	0.4093 (± 0.0132)	0.4970	0.1618
Linear SVM	0.4597 (± 0.0128)	0.5690	0.1519
Mutual Information	0.4082 (± 0.0235)	0.4792	0.1727
NSGA-II Optimization	**0.5598 (± 0.0078)**	**0.6102**	**0.2846**
Random Forest	0.4082 (± 0.0194)	0.5057	0.1481
Relief-F	0.3746 (± 0.0138)	0.4780	0.1968
mRMR	0.4622 (± 0.0114)	0.5275	0.1944

the only method to surpass the performance of the original feature set is the GA approach.

The results presented for the 5 distinct use cases reveal the efficacy of the multi-objective genetic algorithm in optimizing the results from the initial feature selection performed by classic methods by combining and refining them. The datasets utilized represent challenging use cases for feature selection and classification, containing high numbers of features and significantly low sample sizes — which requires feature selection to be precise and stable in order to avoid negative effects on the classification performance and to improve the potential for analysis of the datasets.

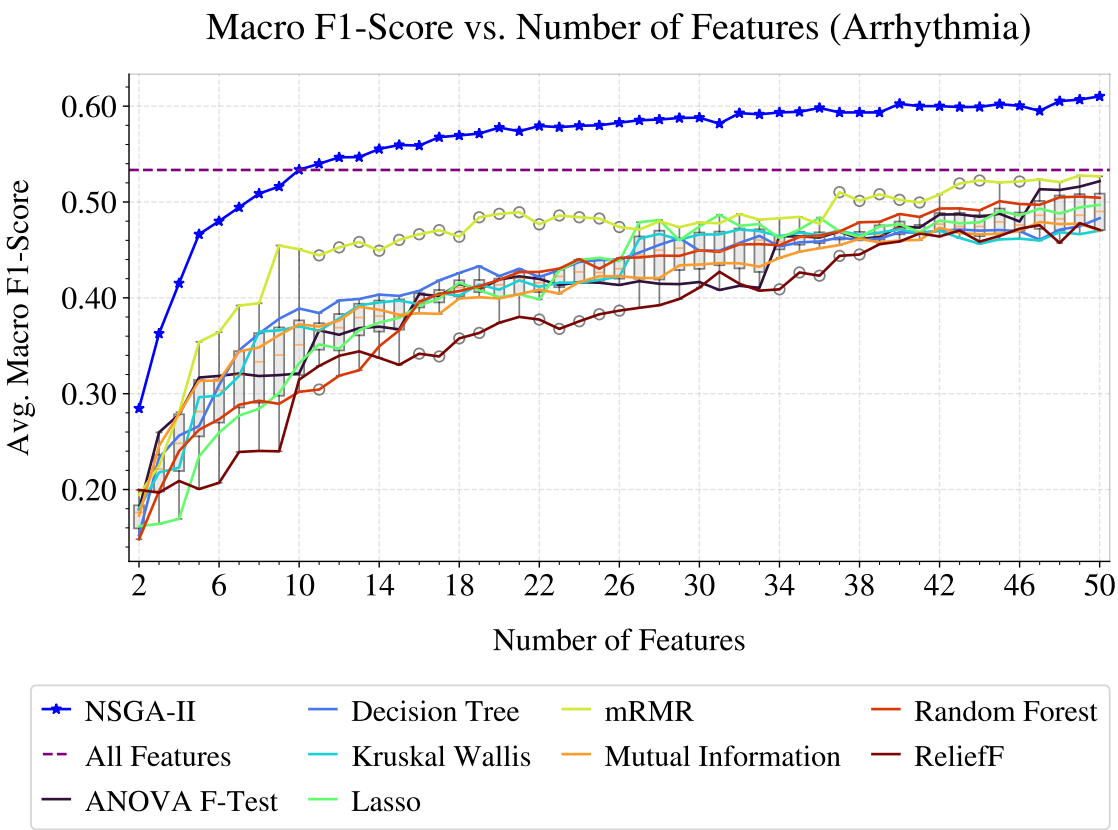
6.2 Execution times

The average run times for the MOGA for each of the datasets across the different executions are reported in table 6.6, along with the average combined time required to run all baseline methods.

Sample sizes can largely affect execution time, as seen in the p53 Mutants use case. In this scenario, the execution time is larger even in comparison to other use cases with considerably wider feature sets, such as the Eye Color SNPs dataset, as evaluating more samples requires longer training times of classifiers for the evaluation of each feature subset generated by the GA.

Wider datasets also naturally require longer run times since there is a larger quantity of possible combinations of features to be evaluated. This becomes evident in the case of the Arrhythmia dataset, where its small set of features is quickly optimized by the

Figure 6.5: Average macro F1-Score performance with varying numbers of features for Arrhythmia dataset.



GA in comparison to other wider datasets with similar sample sizes. Nevertheless, while feature counts do influence total execution time, the sample sizes influence more heavily the duration in the evaluated datasets, noticeably in the p53 Mutants use case. This is expected, as the algorithm mitigates the impact of feature counts on execution time by limiting the size of the feature sets evaluated, as outlined in the parameterization details provided in Table 5.4.

Table 6.6: Average execution times across 10 executions for each dataset.

Dataset	N. Features	N. Samples	Baseline FS Avg. Exec. Time (hours)	MOGA Avg. Exec. Time (hours)
CuMiDa Leukemia	22284	281	1:25:38.171	1:30:55.614
CuMiDa Breast Cancer	35982	289	0:44:09.730	0:49:31.859
Eye Color SNPs	126018	500	3:12:59.683	3:25:14.767
Arrhythmia	279	452	0:00:38.144	0:40:40.854
p53 Mutants	5409	16772	6:40:31.863	6:41:14.786

The reported execution times for the proposed MOGA are, in most use cases, comparable to the total time required by all evaluated baseline methods combined, except the Arrhythmia use case. Given the repeated evaluations performed over generations in the GA, this increased run time is expected. It is likely acceptable in such use cases, where the main goal is achieving higher classification performance with smaller sets of features and not necessarily shorter execution times. Still, experiment parameters could be altered to shorten execution times, such as reducing the feature numbers evaluated, either by reducing the distance between minimum and maximum feature numbers or by reducing the number of steps evaluated in between these limits (e.g., incrementing feature numbers by 2 or more features at a time). Another option would be to reduce GA-specific parameters, such as the number of generations or population sizes, or to use models with faster training times for evaluation.

The execution time of each feature selection method used in the baseline experiments is not included in this section since comparing run times is not necessarily the goal of this work, and methods were not individually optimized for the use cases for a fair comparison.

6.3 Chapter summary

This chapter presented the results of the experiments described in chapter 5, performed to evaluate the optimization of feature sets generated via the proposed multi-objective genetic algorithm for 5 different high-dimensional classification use cases.

The results explored both visually (in images 6.1, 6.2, 6.3, 6.4, 6.5) and statistically (in tables 6.1, 6.2, 6.3, 6.4, 6.5) show consistent and considerable performance improvements obtained through optimization for all use cases when compared to the feature sets initially generated by traditional feature selection methods, providing dominant solutions in a search space that defines a range of possible numbers of features.

The execution time is also provided for each use case in section 6.2, displaying an acceptable processing time for even the most complicated dataset in the context of exploratory analysis, which could potentially be further optimized with proper parameter tuning.

7 CONCLUSION

In this work, we propose a hybrid feature selection method that utilizes a multi-objective genetic algorithm to generate feature sets by combining and optimizing previously selected feature sets generated by other conventional feature selection methods. Based on NSGA-II, the genetic algorithm provides stable, high-performing, and reduced feature subsets for optimizing classification problems and simplifying data analysis tasks. The method works by maximizing a target metric, such as classification metrics, and exploring diverse combinations of feature sets in a constrained search space of a determined range of possible numbers of features. This results in a frontier of Pareto-dominant candidate solutions considering the two objectives: improving the classification performance and reducing the dimensionality of the feature sets.

The method consistently outperforms the baseline feature selection methods in the experiments conducted, surpassing their results by combining and altering the feature sets they initially proposed. The evaluations encompass expressively high-dimensional datasets with diverse sample sizes drawn from real-world life sciences domains. The classification performance of the selected feature sets is presumed to reflect their analytical relevance to the respective problem each dataset represents. The method consistently yields significantly improved classification performance with smaller feature sets across all of the use case scenarios and within the constraints of feature set sizes.

A critical aspect of feature selection is that different methods typically perform better or worse depending on the characteristics of the evaluated datasets, and no consensus exists on which method is the best. Thus, adopting a heuristic that integrates multiple methods in search of a combined solution enables the discovery of optimal and diverse feature sets in the experimented situations and ensures adaptability to varying dataset characteristics. Additionally, the resulting combined feature sets seem to provide a unique view of the data that the other evaluated methods fail to achieve individually, possibly enabling the identification of relations between features not directly explored by the initial methods and potentially providing valuable information for research in domains that deal with high-dimensional data.

The biggest limitation of the proposed method is the execution time since it requires the execution of all initial methods and the evaluation of new combined feature sets as the GA creates them. This can lead to escalating execution times, especially with datasets containing a high volume of samples. To address cases with execution time con-

straints, the method can be parameterized to reduce execution time at the expense of also reducing the number of feature sets evaluated. Still, high-dimensional data analysis will likely opt for longer execution times if that means achieving more representative feature sets capable of effectively differentiating the data samples. Another limitation stems from the dependency of the GA on the initial knowledge provided by the chosen feature selection methods. In cases where the knowledge provided by these methods is ineffective, the GA may struggle to produce optimal feature sets.

In future iterations of this work, several improvements could be explored. Firstly, expanding the evaluation to encompass high-dimensional datasets across other domains would offer more insights into its adaptability to other data characteristics and challenges, including larger feature sets, sample sizes, or differing label distributions. Additionally, the proposed customized genetic algorithm operators could be easily adapted to other multi-objective approaches, including newer variations of MOGAs like NSGA-III or alternative non-pareto-dominated methodologies. Moreover, optimizing our implementation for high-performance frameworks or platforms could reduce the execution time without compromising result quality. Lastly, conducting an in-depth comparison with existing multi-objective feature selection frameworks would provide valuable insights into the efficacy of our solution and its potential limitations.

Furthermore, the method is provided as an open-source Python library available at GitHub: <https://github.com/sbcblab/MOO-HFS>.

8 PUBLICATIONS

This chapter lists the publications that originated from this work.

8.1 Publications in Journals

- **BOHRER, J.**; DORN, M.. Enhancing classification with hybrid feature selection: A multi-objective genetic algorithm for high-dimensional data. Expert Systems with Applications, p.124518. Qualis: A1.

REFERENCES

AALAEI, S. et al. Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets. **Iranian journal of basic medical sciences**, Mashhad University of Medical Sciences, v. 19, n. 5, p. 476, 2016.

ABEEL, T. et al. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. **Bioinformatics**, Oxford University Press, v. 26, n. 3, p. 392–398, 2010.

AHMAD, F. et al. A ga-based feature selection and parameter optimization of an ann in diagnosing breast cancer. **Pattern Analysis and Applications**, Springer, v. 18, p. 861–870, 2015.

AKAY, M. F. Support vector machines combined with feature selection for breast cancer diagnosis. **Expert systems with applications**, Elsevier, v. 36, n. 2, p. 3240–3247, 2009.

ALELYANI, S.; TANG, J.; LIU, H. Feature selection for clustering: A review. **Data Clustering: Algorithms and Applications**, v. 29, p. 110–121, 2013.

ALHENAWI, E. et al. Feature selection methods on gene expression microarray data for cancer classification: A systematic review. **Computers in biology and medicine**, Elsevier, v. 140, p. 105051, 2022.

ALI, A.; SHAMSUDDIN, S. M.; RALESCU, A. L. Classification with class imbalance problem. **Int. J. Advance Soft Compu. Appl**, v. 5, n. 3, p. 176–204, 2013.

ALIČKOVIĆ, E.; SUBASI, A. Breast cancer diagnosis using ga feature selection and rotation forest. **Neural Computing and applications**, Springer, v. 28, p. 753–763, 2017.

ANG, J. C. et al. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 13, n. 5, p. 971–989, 2015.

ANG, J. C. et al. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 13, n. 5, p. 971–989, 2016.

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of genetic algorithms: Part 1, fundamentals. **University computing**, v. 15, n. 2, p. 56–69, 1993.

BERRAR, D. Cross-validation. In: RANGANATHAN, S. et al. (Ed.). **Encyclopedia of Bioinformatics and Computational Biology**. Oxford: Academic Press, 2019. p. 542–545. ISBN 978-0-12-811432-2. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B978012809633820349X>.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**. New York, NY, USA: Association for Computing Machinery, 1992. (COLT '92), p. 144–152. ISBN 089791497X.

BOURAOUI, A.; JAMOUSSI, S.; BENAYED, Y. A multi-objective genetic algorithm for simultaneous model and feature selection for support vector machines. **Artificial Intelligence Review**, Springer, v. 50, p. 261–281, 2018.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Available from Internet: <http://doi.acm.org/10.1145/2939672.2939785>.

DEB, K. et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: SCHOENAUER, M. et al. (Ed.). **Parallel Problem Solving from Nature PPSN VI**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 849–858. ISBN 978-3-540-45356-7.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.

DENG, X. et al. Hybrid gene selection approach using xgboost and multi-objective genetic algorithm for cancer classification. **Medical & Biological Engineering & Computing**, Springer, v. 60, n. 3, p. 663–681, 2022.

FADAEE, M.; RADZI, M. A. M. Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review. **Renewable and sustainable energy reviews**, Elsevier, v. 16, n. 5, p. 3364–3369, 2012.

FELTES, B. C. et al. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. **Journal of Computational Biology**, v. 26, n. 4, p. 376–386, 2019. PMID: 30789283. Available from Internet: <https://doi.org/10.1089/cmb.2018.0238>.

FERRI, F. et al. Comparative study of techniques for large-scale feature selection. In: GELSEMA, E. S.; KANAL, L. S. (Ed.). **Pattern Recognition in Practice IV**. North-Holland, 1994, (Machine Intelligence and Pattern Recognition, v. 16). p. 403–413. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9780444818928500407>.

FISHER, R. A. The general sampling distribution of the multiple correlation coefficient. **Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character**, The Royal Society London, v. 121, n. 788, p. 654–673, 1928.

FONSECA, C. M.; FLEMING, P. J. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: **Proceedings of the 5th International Conference on Genetic Algorithms**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. p. 416–423. ISBN 1558602992.

FONTI, V.; BELITSER, E. Feature selection using lasso. **VU Amsterdam research paper in business analytics**, Business Analytics Master Amsterdam, The Netherland, v. 30, p. 1–25, 2017.

GAO, L. et al. Learning in high-dimensional multimedia data: the state of the art. **Multimedia Systems**, v. 23, n. 3, p. 303–313, Jun 2017. ISSN 1432-1882.

GRISCI, B. I.; FELTES, B. C.; DORN, M. Neuroevolution as a tool for microarray gene expression pattern identification in cancer research. **Journal of Biomedical Informatics**, v. 89, p. 122–133, 2019.

GRISCI, B. I. et al. The use of gene expression datasets in feature selection research: 20 years of inherent bias? **WIREs Data Mining and Knowledge Discovery**, v. 14, n. 2, p. e1523, 2024.

GUVENIR, H. et al. **Arrhythmia**. 1998. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5BS32.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. **Machine learning**, Springer, v. 46, p. 389–422, 2002.

HAMBALI, M. A.; OLADELE, T. O.; ADEWOLE, K. S. Microarray cancer feature selection: Review, challenges and research directions. **International Journal of Cognitive Computing in Engineering**, Elsevier, v. 1, p. 78–97, 2020.

HASNAT, A.; MOLLA, A. U. Feature selection in cancer microarray data using multi-objective genetic algorithm combined with correlation coefficient. In: IEEE. **2016 International Conference on Emerging Technological Trends (ICETT)**. [S.l.], 2016. p. 1–6.

HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on knowledge and data engineering**, Ieee, v. 21, n. 9, p. 1263–1284, 2009.

HECKE, T. V. Power study of anova versus kruskal-wallis test. **Journal of Statistics and Management Systems**, Taylor & Francis, v. 15, n. 2-3, p. 241–247, 2012.

HEINRICH, F. et al. Exploring the potential of incremental feature selection to improve genomic prediction accuracy. **Genetics Selection Evolution**, v. 55, n. 1, p. 78, Nov 2023.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1992. ISBN 0262082136.

HORN, J. D.; NAFPLIOTIS, N.; GOLDBERG, D. E. A niched pareto genetic algorithm for multiobjective optimization. **Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence**, p. 82–87 vol.1, 1994. Available from Internet: <https://api.semanticscholar.org/CorpusID:18549555>.

KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. **Multimedia Tools and Applications**, Springer, v. 80, p. 8091–8126, 2021.

KAYSER, M. Forensic DNA Phenotyping: Predicting human appearance from crime scene material for investigative purposes. **Forensic Science International: Genetics**, v. 18, p. 33–48, 2015.

KELLY, M.; LONGJOHN, R.; NOTTINGHAM, K. **The UCI Machine Learning Repository**. 2021. Available from Internet: <https://archive.ics.uci.edu>.

KIM, T. K. Understanding one-way anova using conceptual figures. **Korean journal of anesthesiology**, Korean Society of Anesthesiologists, v. 70, n. 1, p. 22, 2017.

KIRA, K.; RENDELL, L. A. A practical approach to feature selection. In: SLEEMAN, D.; EDWARDS, P. (Ed.). **Machine Learning Proceedings 1992**. San Francisco (CA): Morgan Kaufmann, 1992. p. 249–256. ISBN 978-1-55860-247-2. Available from Internet: <https://www.sciencedirect.com/science/article/pii/B9781558602472500371>.

KONONENKO, I. Estimating attributes: Analysis and extensions of relief. In: BERGADANO, F.; RAEDT, L. D. (Ed.). **Machine Learning: ECML-94**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 171–182. ISBN 978-3-540-48365-6.

KRIZEK, P. **Feature selection: stability, algorithms, and evaluation**. Thesis (PhD) — Czech Technical University in Prague, 2008.

KUNDU, R.; MALLIPEDDI, R. Hfmoea: A hybrid framework for multi-objective feature selection. **Journal of Computational Design and Engineering**, Oxford University Press, v. 9, n. 3, p. 949–965, 2022.

LAL, T. N. et al. Embedded methods. In: **Feature extraction: Foundations and applications**. Springer, 2006. p. 137–165. Available from Internet: <https://doi.org/10.1007/978-3-540-35488-8_6>.

LATHROP, R. **p53 Mutants**. 2010. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5T89H.

LI, J. et al. Feature selection: A data perspective. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 50, n. 6, dec 2017. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/3136625>.

LIM, S. M. et al. Crossover and mutation operators of genetic algorithms. **International journal of machine learning and computing**, v. 7, n. 1, p. 9–12, 2017.

LIU, X.-Y. et al. A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. **IEEE Access**, IEEE, v. 6, p. 22863–22874, 2018.

MALEKI, N.; ZEINALI, Y.; NIAKI, S. T. A. A k-nn method for lung cancer prognosis with the use of a genetic algorithm for feature selection. **Expert Systems with Applications**, Elsevier, v. 164, p. 113981, 2021.

MAZZANTI, S. **mRMR: An implementation of the minimum Redundancy Maximum Relevance (mRMR)**. 2023. <https://github.com/smazzanti/mrmr>. 2024.

MCKIGHT, P. E.; NAJAB, J. Kruskal-wallis test. **The corsini encyclopedia of psychology**, Wiley Online Library, p. 1–1, 2010.

MIAO, J.; NIU, L. A survey on feature selection. **Procedia Computer Science**, Elsevier, v. 91, p. 919–926, 2016.

NGUYEN, C.; WANG, Y.; NGUYEN, H. N. Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. Scientific Research Publishing, 2013.

OLSON, D. L.; DELEN, D. **Advanced Data Mining Techniques**. 1st. ed. Springer, 2008. (Springer Books, 978-3-540-76917-0). ISBN 3540769161. Available from Internet: <https://ideas.repec.org/b/spr/sprbok/978-3-540-76917-0.html>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PENG, H.; LONG, F.; DING, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 27, n. 8, p. 1226–1238, 2005.

PIRI, J. et al. Literature review on hybrid evolutionary approaches for feature selection. **Algorithms**, v. 16, n. 3, 2023. ISSN 1999-4893. Available from Internet: <https://www.mdpi.com/1999-4893/16/3/167>.

PUDJIHARTONO, N. et al. A review of feature selection methods for machine learning-based disease risk prediction. **Frontiers in Bioinformatics**, v. 2, 2022.

PURSHOUSE, R. C.; FLEMING, P. J. Why use elitism and sharing in a multi-objective genetic algorithm? In: **Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002. (GECCO'02), p. 520–527. ISBN 1558608788.

REMESEIRO, B.; BOLON-CANEDO, V. A review of feature selection methods in medical applications. **Computers in Biology and Medicine**, v. 112, p. 103375, 2019.

ROSS, B. C. Mutual information between discrete and continuous data sets. **PloS one**, Public Library of Science San Francisco, USA, v. 9, n. 2, p. e87357, 2014.

SAEYS, Y.; INZA, I.; LARRANAGA, P. A review of feature selection techniques in bioinformatics. **bioinformatics**, Oxford University Press, v. 23, n. 19, p. 2507–2517, 2007.

SAYED, S. et al. A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. **Expert Systems with Applications**, Elsevier, v. 121, p. 233–243, 2019.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. USA: Cambridge University Press, 2014. ISBN 1107057132.

SINGH, R. K.; SIVABALAKRISHNAN, M. Feature selection of gene expression data for cancer classification: a review. **Procedia Computer Science**, Elsevier, v. 50, p. 52–57, 2015.

SONG, Y.-Y.; YING, L. Decision tree methods: applications for classification and prediction. **Shanghai archives of psychiatry**, Shanghai Mental Health Center, v. 27, n. 2, p. 130, 2015.

SPEISER, J. L. et al. A comparison of random forest variable selection methods for classification prediction modeling. **Expert systems with applications**, Elsevier, v. 134, p. 93–101, 2019.

ST, L.; WOLD, S. et al. Analysis of variance (anova). **Chemometrics and intelligent laboratory systems**, Elsevier, v. 6, n. 4, p. 259–272, 1989.

STONE, M. Cross-validatory choice and assessment of statistical predictions. **Journal of the royal statistical society: Series B (Methodological)**, Wiley Online Library, v. 36, n. 2, p. 111–133, 1974.

TADIST, K. et al. Feature selection methods and genomic big data: a systematic review. **Journal of Big Data**, Springer, v. 6, n. 1, p. 1–24, 2019.

TAKAHASHI, K. et al. Confidence interval for micro-averaged f 1 and macro-averaged f 1 scores. **Applied Intelligence**, Springer, v. 52, n. 5, p. 4961–4972, 2022.

TAN, C. J.; LIM, C. P.; CHEAH, Y.-N. A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. **Neurocomputing**, Elsevier, v. 125, p. 217–228, 2014.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 58, n. 1, p. 267–288, 1996.

URBANOWICZ, R. J. et al. **Benchmarking Relief-Based Feature Selection Methods**. 2017. ArXiv e-print. https://arxiv.org/abs/1711.08477.

VARSHAVSKY, R. et al. Novel unsupervised feature filtering of biological data. **Bioinformatics**, Oxford University Press, v. 22, n. 14, p. e507–e513, 2006.

VERGARA, J. R.; ESTÉVEZ, P. A. A review of feature selection methods based on mutual information. **Neural computing and applications**, Springer, v. 24, p. 175–186, 2014.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. In: CABESTANY, J.; PRIETO, A.; SANDOVAL, F. (Ed.). **Computational Intelligence and Bioinspired Systems**. Heidelberg: Springer, 2005. p. 758–770.

VERMA, S.; PANT, M.; SNASEL, V. A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. **Ieee Access**, IEEE, v. 9, p. 57757–57791, 2021.

WANG, Z.; LI, M.; LI, J. A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure. **Information Sciences**, Elsevier, v. 307, p. 73–88, 2015.

XU, X. et al. Review of classical dimensionality reduction and sample selection methods for large-scale data processing. **Neurocomputing**, Elsevier, v. 328, p. 5–15, 2019.

XUE, Y. et al. Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. **Knowledge-Based Systems**, Elsevier, v. 227, p. 107218, 2021.

YU, X.; GEN, M. Introduction to evolutionary algorithms. **Industrial Engineering & Management Systems**, v. 9, n. 4, p. 348–349, 2010.

ZHAI, Y.; ONG, Y.-S.; TSANG, I. W. The emerging" big dimensionality". **IEEE Computational Intelligence Magazine**, IEEE, v. 9, n. 3, p. 14–26, 2014.

ZHOU, A. et al. Multiobjective evolutionary algorithms: A survey of the state of the art. **Swarm and evolutionary computation**, Elsevier, v. 1, n. 1, p. 32–49, 2011.

APPENDIX A — RESUMO ESTENDIDO EM PORTUGUÊS

Dados de alta dimensionalidade são um problema conhecido em diversas áreas do conhecimento — dados com números grandes de variáveis são difíceis de interpretar, tanto por seres humanos quanto por algoritmos de aprendizagem. Este tipo de dado é comum em áreas do conhecimento como biologia, medicina, engenharias e outras, onde grandes números de marcadores ou variáveis são utilizados para o entendimento e predição de determinados eventos ou medições. Exemplos de utilização deste tipo de dados são estudos de câncer e outras doenças, análises de DNA e características genéticas, estudos de fenômenos físicos e naturais, projeções de engenharia, e outros. Nestes casos, o excesso de informações dificulta a associação de determinadas variáveis ou suas combinações aos resultados observados, e atrasa o processo de estudo, pesquisa e identificação de causas envolvido. Para lidar com tais cenários de dados de alta dimensionalidade, surge dentro da área de pesquisa de redução de dimensionalidade o conceito de seleção de variáveis.

No processo de seleção de variáveis são preservadas as variáveis originais, um aspecto importante para a posterior interpretação do problema associado aos dados. A literatura atual oferece muitas opções de métodos de seleção de variáveis, com diferentes categorias baseadas na metodologia adotada e nas características dos dados ou do problema sendo resolvido pelos mesmos. Ainda assim, não existe uma recomendação de método genérico o suficiente para todos os casos de uso, e frequentemente métodos bem-sucedidos em algumas áreas tem más performances em outras. Assim, surge a necessidade de estudos e propostas de métodos focados em generalizar o processo de seleção de variáveis para domínios e aplicações variados.

Neste trabalho, propomos um método híbrido de seleção de variáveis que utiliza um algoritmo genético multi-objetivo para selecionar conjuntos de variáveis em dados de alta dimensionalidade. O método combina e otimiza conjuntos de variáveis previamente selecionados por outros métodos clássicos de seleção de variáveis através de uma estrutura de algoritmo genético. Baseado no NSGA-II, o algoritmo genético fornece subconjuntos de variáveis estáveis, de alto desempenho e reduzidos para otimizar problemas de classificação e simplificar tarefas de análise de dados. O método funciona maximizando uma métrica alvo, como métricas de classificação, e explorando diversas combinações de variáveis em um espaço de busca de soluções restrito a um determinado intervalo de quantidades possíveis de variáveis. Isso resulta em uma fronteira de Pareto de soluções

candidatas dominantes em termos de dois objetivos mensuráveis específicos: melhorar o desempenho de métricas de classificação e reduzir a dimensionalidade dos conjuntos de variáveis.

O método proposto baseia-se na proposição de novos operadores de algoritmo genético compatíveis com a estrutura do NSGA-II. São propostos três operadores: um operador de amostragem, um operador de mutação e um operador de avaliação de adequação. Os novos operadores são responsáveis, respectivamente, por: i) a geração de novas soluções (ou subconjuntos de variáveis); ii) a modificação de soluções existentes (alterando partes de subconjuntos em busca de melhorias); e iii) a avaliação de soluções ao longo do processo de acordo com os objetivos estabelecidos, através de métricas de classificação e do tamanho dos conjuntos avaliados. Estes novos operadores utilizam a informação prévia fornecida pela rodada inicial de métodos clássicos para gerar e alterar seleções de variáveis que combinam as contribuições de cada método inicial utilizado, limitando o espaço de busca de soluções às variáveis selecionadas por eles antecipadamente. Com os novos operadores, o processo de algoritmo genético acontece ao longo de várias gerações, onde os mesmos são utilizados para gerar soluções progressivamente mais adaptadas a resolver o problema de classificação representado pelos dados.

O método supera consistentemente os métodos de seleção de variáveis utilizados como base de comparação nos experimentos conduzidos, melhorando seus resultados ao combinar e alterar os conjuntos de variáveis propostos inicialmente. As avaliações abrangem conjuntos de dados de alta dimensionalidade, com diversos tamanhos de amostra e extraídos de aplicações de mundo real de domínios das ciências biológicas e médicas. Presume-se que o desempenho da classificação dos conjuntos de recursos selecionados reflita sua relevância analítica para o respectivo problema que cada conjunto de dados representa. O método produz consistentemente um desempenho de classificação significativamente melhorado com conjuntos de variáveis menores em todos os cenários de experimentação e dentro das restrições de tamanho estabelecidas para os conjuntos de variáveis.

Um aspecto crítico da seleção de variáveis é que diferentes métodos normalmente apresentam desempenho melhor ou pior, dependendo das características dos conjuntos de dados avaliados, e não existe consenso sobre qual método é o melhor em todos os cenários. Assim, a adoção de uma heurística que integra múltiplos métodos em busca de uma solução combinada permite a descoberta de conjuntos de variáveis diversificados e adaptados às diversas características dos conjuntos de dados vistos. Além disso, os

conjuntos de variáveis resultantes de combinações de outros métodos parecem fornecer uma visão única dos dados que os outros métodos avaliados não conseguem alcançar individualmente, possivelmente permitindo a identificação de relações entre variáveis não exploradas diretamente pelos métodos iniciais. Essas novas conexões têm o potencial de fornecer informações valiosas para domínios de pesquisa que utilizam tais tipos de dados de alta dimensionalidade.

A maior limitação observada no método proposto é o tempo de execução, pois requer a execução de todos os métodos base de seleção de variáveis e a avaliação de novos conjuntos de variáveis à medida que o algoritmo genético os cria. Isto pode levar a tempos de execução cada vez maiores, especialmente com conjuntos de dados que contêm um grande volume de amostras. Para atender aplicações com restrições de tempo de execução, o método pode ser parametrizado para reduzir o tempo de execução às custas de reduzir também o número de conjuntos de variáveis avaliados durante o processo. Ainda assim, para estudos e análises envolvendo dados de alta dimensionalidade, tempos de execução mais longos podem representar um problema aceitável caso isso signifique alcançar conjuntos de variáveis mais representativos, capazes de eficazmente representar as diferenças das amostras de dados. Outra limitação observada decorre do fato de o algoritmo genético depender do conhecimento inicial fornecido pelos métodos de seleção de variáveis escolhidos. Nos casos em que o conhecimento fornecido por esses métodos é ineficaz, o GA pode ter dificuldades para produzir conjuntos de variáveis ideais.

Em futuras iterações deste trabalho, diversas melhorias poderão ser exploradas. Em primeiro lugar, expandir a avaliação para abranger conjuntos de dados de alta dimensão em outros domínios de dados ofereceria mais esclarecimentos sobre a sua adaptabilidade a dados com outras características e desafios, incluindo dados com conjuntos maiores de variáveis, mais amostras ou comportamentos diferentes. Além disso, os operadores de algoritmo genético propostos neste trabalho poderiam ser facilmente adaptados a outras abordagens multiobjetivo, incluindo variações mais recentes de MOGAs como NSGA-III ou metodologias alternativas que não envolvam o conceito de dominação de Pareto. Além disso, otimizar a implementação para estruturas ou plataformas de alto desempenho poderia reduzir o tempo de execução sem comprometer a qualidade dos resultados. Por último, a realização de uma comparação aprofundada com outras técnicas de seleção de variáveis multiobjetivo existentes forneceria informações valiosas sobre a eficácia da nossa solução e suas potenciais limitações.