Multi-Objective Optimization Based Machine Learning with Real-Life Applications

by

Nandini Rakala

Master of Science in Applied Mathematics and Computing
Department of Mathematics
Manipal University
2015

Bachelor of Science in Mathematics (Honours)
Department of Mathematics
Sri Sathya Sai Institute of Higher Learning
2012

A dissertation submitted to the College of Engineering and Science of Florida Institute of Technology in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

Melbourne, Florida
May, 2020
We the undersigned committee hereby approve the attached dissertation

Multi-Objective Optimization Based Machine Learning with Real-Life Applications

by Nandini Rakala

Munevver Mine Subasi, Ph.D.
Associate Professor
Department of Mathematical Sciences
Committee Chair

Luis Daniel Otero, Ph.D.
Associate Professor
Department of Computer Engineering and Sciences
Outside Committee Member

Ersoy Subasi, Ph.D.
Assistant Professor
Department of Computer Engineering and Sciences
Committee Member

Nezamoddin Nezamoddini-Kachouie, Ph.D.
Associate Professor
Department of Mathematical Sciences
Committee Member

Vladislav Buxshtynov, Ph.D.
Assistant Professor
Department of Mathematical Sciences
Committee Member

Munevver Mine Subasi, Ph.D.
Associate Professor and Department Head
Department of Mathematical Sciences
ABSTRACT

Title:
Multi-Objective Optimization Based Machine Learning with Real-Life Applications

Author:
Nandini Rakala

Major Advisor:
Munevver Mine Subasi, Ph.D.

We propose a multi-objective optimization-based Machine Learning method, called Multi-Objective Evolutionary Hybrid Classification Algorithm (MOEHCA), for binary classification problems. Our method incorporates the evolutionary feature selection using the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) for the underlying Machine Learning classification problem, thereby, combining the advantages of filter, wrapper, and evolutionary search methods, in a hybrid multi-objective manner, finding the trade-off solutions of multiple objectives in a Pareto-based setting. The MOEHCA method is structured in two phases: Evolutionary Classification Phase-I and Generalized Classification Phase-II. The proposed method is defined as a multi-objective optimization problem involving the simultaneous optimization of two given objectives, producing a set of optimal solutions, called the Pareto-set, where each solution of this set has a different trade-off between the two objectives. We apply the proposed approach to analyze real-life datasets including the Wisconsin Breast Cancer, Coronavirus Disease (COVID-19), and Pima Indians Diabetes datasets, providing a computational comparison of our proposed approach with frequently used traditional classification algorithms.
We also show how our proposed MOEHCA approach can be integrated with the Logical Analysis of Data (LAD) method to solve binary classification problems in a multi-objective manner, producing patterns with minimum degree and maximum coverage. The proposed method is applied to the African American Study of Chronic Kidney Disease with Hypertension to predict the combination of Pareto-optimal protein peaks with minimum degree and maximum coverage, that are either positive (predict rapid progression) or negative (predict slow progression) bio-markers. Computational experiments are performed to demonstrate the utility of our proposed MOEHCA method against widely-used classification algorithms. The results show that our proposed approach achieves comparable performance on par by optimizing multiple objectives simultaneously in a Pareto-based manner, and is both flexible and robust against high-dimensional data. In a broader sense, our proposed approach explains the interactions among different features which help obtain deeper insights into the underlying patterns in data, which might further help to develop the existing knowledge of therapeutic intervention and prognostic study of real-life medical data-sets with multiple objectives and without the need to define a scalar weight-based trade-off factor apriori.
# Table of Contents

Abstract iii

List of Figures ix

List of Tables xi

Acknowledgments xii

Dedication xiv

1 Introduction 1

2 Background and Methods 7

2.1 Machine Learning Methods 7

2.1.1 Supervised Machine Learning Classification Methods 11

2.1.1.1 Decision Trees 11

2.1.1.2 Random Forests 12

2.1.1.3 Gradient Boosted Trees 12

2.1.1.4 Support Vector Machines (SVM) 13

2.1.1.5 Neural Networks/Deep Learning 13

2.1.1.6 Naïve Bayes 14

2.1.1.7 Generalized Linear Models 14
3 Multi-Objective Evolutionary Hybrid Classification Approach

3.1 Multi-Objective Evolutionary Hybrid Classification Algorithm (MOE-HCA) ........................................... 41

3.1.1 Performance Measures ........................................ 44

3.2 Real-Life Applications of MOEHCA ............................ 46

3.2.1 Medical Application I: Classification of Tumors in Breast-Cancer ..................................................... 46

3.2.1.1 Methodology ............................................... 47

3.2.1.2 WBC: Results ............................................. 48

3.2.2 Medical Application II: Classification of COVID-19 Patient Outcomes ................................................. 52

3.2.2.1 Background and Introduction ............................. 52

3.2.2.2 Methodology ............................................. 53

3.2.2.3 COVID-19: Results ....................................... 61

3.2.3 Medical Application III: Pima Indians Diabetes .......... 70

3.2.3.1 Methodology ............................................. 70

3.2.3.2 Pima Indians Diabetes: Results ......................... 72

4 Multi-Objective Logical Analysis of Data (MOLAD) .... 74

4.1 MOLAD Methodology ........................................... 75

4.1.1 Pattern Generation: Constraints ............................ 76

4.1.2 MOLAD Model ............................................. 78

4.1.3 Framework of Proposed MOLAD Approach ............. 79

4.2 Medical Application IV: African-American Study of Chronic Kidney Disease with Hypertension (AASK) ......... 81

4.2.1 Methodology ............................................. 85

vii
4.2.2 AASK: MOEHCA Evolutionary Phase-I 85
4.2.3 AASK: MOEHCA Evolutionary Phase-I Results 87
4.2.4 AASK: MOEHCA-MOLAD Phase II 88
4.2.5 AASK: MOEHCA-MOLAD Phase-II Results 92
4.3 Computational Experiments: Comparison of MOEHCA with Traditional Classification Algorithms 94

5 Conclusion 97

References 101
List of Figures

1.1 Predictive Modeling Flowchart ........................................ 2
2.1 Machine Learning Framework ........................................... 16
2.2 Illustration of Evolutionary Algorithm Framework ................. 21
2.3 Pareto Plot of ZDT1 Test Function obtained using NSGA-II in Python 34
2.4 Example: Patient Diet Daily Intake Record .......................... 35
3.1 Evolutionary Classification Phase-I: Performance Evaluation Framew- 
   ork ..................................................................................... 45
3.2 WBC: Target Class Distribution .......................................... 47
3.3 WBC: MOEHCA Pareto-plot .............................................. 49
3.4 WBC: MOEHCA Decision Tree Model ................................. 50
3.5 WBC: Risk Factors .......................................................... 51
3.6 COVID-19 Patient Data Binary Classification ........................ 54
3.7 COVID-19: Single-Objective Classification Models ............... 55
3.8 COVID-19: Decision Tree Model ........................................ 57
3.9 COVID-19: MOEHCA Evolutionary Classification Phase-I Model . 58
3.10 COVID-19: Feature Importance Plot Using Shapley Values .... 63
3.11 COVID-19: Risk Plot ...................................................... 64
3.12 Chronic Diseases Leading to COVID-19 .............................. 65
3.13 COVID-19: Chronic Diseases by Percentage Contribution . . . . . 66
3.14 Days between Symptom Onset and COVID-19 Confirmation by
    Outcome Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
3.15 COVID-19: Patient Age Group Comparison between China and
    United States . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
3.16 Pima Indians Diabetes Dataset: Target Class Distribution . . . . . 71

4.1 Framework of Proposed MOEHCA-MOLAD Approach . . . . . . . 80
4.2 Prevalence of Renal Disease in USA . . . . . . . . . . . . . . . . . 82
4.3 AASK Proteomics Data Target Class Distribution [60] . . . . . . . 84
4.4 Phase I: Process Flow Diagram in Rapid Miner Studio. [46] . . . . 86
4.5 Support-Set: Pareto Optimal Proteins . . . . . . . . . . . . . . . . 89
4.6 Accuracy, Run-time, and ROC Comparison of AASK Hybrid Clas-
    sification Model Phase-I. . . . . . . . . . . . . . . . . . . . . . . . 91
## List of Tables

3.1 COVID-19: Single-Objective Decision Tree Performance Metrics . . 56

3.2 Pima Diabetes: MOEHCA Fuzzy Classifier Model . . . . . . . . . . 73

4.1 Support-Set: Pareto Optimal Proteins . . . . . . . . . . . . . . . . . . . . . . . . 90

4.2 Significant Serum Proteomic Features Associated with CKD . . . . 92

4.3 MOLAD AASK Maximum Coverage Model . . . . . . . . . . . . . . . . . . . . . . . . 93

4.4 Performance Comparison of Proposed MOEHCA Approach with
    Traditional Classification Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
Acknowledgments

I am extremely fortunate to have had Dr. Munevver Mine Subasi, Associate Professor and Department Head of Mathematical Sciences, both as my academic and research advisor, and owe immense thanks to her, without whose support and guidance this work would have been impossible.

I am highly thankful to my committee members, Dr. Vladislav Bukshtynov, Dr. Nezamoddin N. Kachouie, Dr. Luis Daniel Otero, and Dr. Ersoy Subasi, for providing me with helpful review and valuable suggestions.

I am much obliged to my school, Florida Institute of Technology, for the quality academic, and excellent research facilities, and am thankful to the Department of Mathematical Sciences for providing me the Teaching Assistant position between the years of 2016 to 2018.

I would like to thank Dr. Michael Lipkowitz for providing the African American Study of Chronic Kidney Disease with Hypertension (AASK) Proteomics dataset and acknowledge the AASK sponsors, the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), NIH Office of Research in Minority Health, and King Pharmaceuticals.

This work was supported in part by an appointment to the Research Participation Program at the U.S. Food and Drug Administration administered by the
Oak Ridge Institute for Science and Education through an inter-agency agreement between the U.S. Department of Energy and the U.S. Food and Drug Administration. I would like to thank my research mentor at FDA, Alex Viehmann, for his unparalleled support, and would like to acknowledge the Oak Ridge Institute for Science and Education (ORISE) for providing me with the Predictive Analytics Fellowship opportunity at FDA.

Above all, I am deeply grateful to my family and friends, on embarking this journey with me together, and always being there for my emotional and intellectual ups and downs.
Dedication

Dedicated to the Lord of the Universe, my loving mother- Varalaxmi Rakala, my inspiring father- Manohar Rakala, my wonderful aunt- Jagadeshwari Rakala, and my dear brother- Dr. Geet Rakala, for being the backbone of my life, without whom, I would not have reached up till this point...
Chapter 1

Introduction

Sophisticated methods and tools offered by Machine Learning are essential to address the growing data mining needs in real-life problems, where large-volumes of data are collected in order to analyze them in the context of hypotheses to be tested or questions to be answered. Most real-life problems in general, have several complementary or competing objectives to be optimized simultaneously, and hence, the process of mining real-life data inherently involves steps to optimizing multiple objectives in the underlying problem. In Machine Learning literature, most of the data analysis problems are commonly treated by transforming the multiple objectives into one weighted scalarized objective or by treating the multiple objectives as a single-objective optimization problem, and transforming all but one objective into constraints [6, 20, 44]. Despite the growing prominence of Machine Learning in data analysis and extensive research efforts channeled to the development of methodologies and tools, there is still room for proposing alternative approaches which integrate multi-objective optimization into the data analysis process.
Multi-objective optimization caters to achieving multiple goals, subject to a set of constraints, with a likelihood that the objectives will compete with each other. Multi-objective optimization can also be explained as a multi-criteria decision-making process, in which multiple objective functions have to be optimized simultaneously. In many cases, optimal decisions may require trade-offs between competing objectives [4]. The obtained set of solutions are called non-dominating Pareto-optimal solutions (dominating other solutions or not dominated by other solutions) as compared to the rest of the obtained solutions. The Pareto-optimal term is named after the Italian engineer and economist, Vilfredo Pareto (1848 - 1923), who had first coined and applied this concept. The solutions thereby obtained using multi-objective optimization are no longer optimal with respect to a single objective, but are partially optimal with respect to multiple competing criteria as objectives.

Predictive Modeling is a Machine Learning task of classification or regression consisting of the steps shown in Figure 1.1 in the most simplest form.

![Figure 1.1: Predictive Modeling Flowchart](image-url)
A typical data analysis pipeline consists of four steps [3], including the following:

(i) data pre-processing (data transformation, imputation, feature selection, feature reduction, or support-set selection),

(ii) class discovery (clustering) or class comparison and discrimination (regression or classification),

(iii) evaluation (statistical tests or cross-validation), and

(iv) interpretation of the knowledge extracted from the data.

With the advent of Machine Learning methods and tools, research in many fields has been shifted from hypothesis-driven to data-driven decision-making process. Machine Learning (ML) and Artificial Intelligence (AI) are the two biggest buzzwords of the 21st century, and undoubtedly what every industry seems to be getting their hands on. In the Machine Learning process, training a model involves feeding the machine with vast amounts of data in order to uncover hidden patterns and deep insights originating from the data itself. Machine Learning discovers patterns that are impossible to see with human eye and relies on algorithms that learn from past examples. Definitely, in this intriguing field, there are data-sets with millions, billions, or even trillions of data points, and with the exponential growth of technology, we not only need better tools to understand the data that we currently have, but also need to prepare ourselves for the data we will have in the future.
One typical class of Machine Learning problems is the task of classification which requires classifying the underlying data points into binary or multi-class data through an approximation of the hidden function. Well-known classification algorithms such as support vector machines [15, 55], neural networks [25, 7], decision trees [7, 24], $k$-Nearest Neighbor [48], and Naïve Bayes [24, 7], etc., have been designed to solve binary classification problems where a learning model is constructed to separate observations in two predefined classes.

Over the last decade, efforts on solving Machine Learning problems using Pareto-based multi-objective optimization methodology have gained increasing importance as it has been shown that Pareto-based multi-objective learning approaches are more realistic and powerful compared to learning algorithms with a scalar cost function in addressing various topics of machine learning, such as clustering, feature selection, improvement of generalization ability, knowledge extraction, and ensemble generation. One common benefit of the different multi-objective learning approaches is that a deeper insight into the learning problem can be gained by analyzing the Pareto front composed of multiple Pareto-optimal solutions. Evolutionary algorithms, inspired by natural evolution, aim to optimize difficult objective functions without computing derivatives [36].
In this dissertation, we propose a Multi-Objective Evolutionary Hybrid Classification Algorithm (MOEHCA) using the Non-dominated Sorting Genetic Algorithm II (NSGA-II), an evolutionary method that takes advantage of the non-dominated pareto-based search optimization criteria for the underlying classification task with multiple objectives [22]. The proposed approach can be integrated with well-known and commonly used machine learning methods and can be easily adopted for diverse real-life problems involving data analysis.

The organization of this dissertation is as follows. In Chapter 2, we give a brief overview of the machine learning methods, involving feature selection and classification tasks, and multi-objective optimization techniques that serve as a basis for our proposed data analysis methodology, discussing some advantages and disadvantages through a thorough review of literature on the same. In Chapter 3, we develop our proposed multi-objective optimization based Machine Learning algorithm. We use our proposed method to analyze three publicly available real-life data-sets, including Wisconsin Breast Cancer, Pima Indians Diabetes, and Coronavirus Disease (COVID-19) data-sets, to demonstrate the utility of the multi-objective optimization based data analysis approach. Chapter 4 integrates the proposed methodology with Logical Analysis of Data (LAD) that solves two-class classification problems in a multi-objective manner producing patterns with minimum degree and maximum coverage. Multi-objective optimization based LAD approach is then applied to the African-American Study of Chronic Kidney Disease with Hypertension.
Finally, in Chapter 5, we conclude with a discussion of the advantages of our proposed multi-objective optimization based Machine Learning approach based on the empirical results obtained from our analyses of the real-life data-sets. We also give a brief interpretation of the results in the context of the underlying real-life problems analyzed.
Chapter 2

Background and Methods

In this Chapter, we give a brief overview of the Machine Learning methods, involving feature selection and classification tasks, and multi-objective optimization techniques that serve as a basis for our proposed multi-objective optimization based data analysis methodology that we shall develop in Chapter 3. We also discuss some advantages and disadvantages through a thorough review of literature on the same.

2.1 Machine Learning Methods

Machine Learning is about finding models for the underlying data-sets, describing the given data and leading into deeper insights, such that the models can be generalized and further applied on future unseen data points. Below, we define common terminology used in the Machine Learning literature.
Definition 2.1.1 (Instance Space)  An instance space is composed of a set of n random variables $X = X_1 \times \ldots \times X_n$ [44].

Definition 2.1.2 (Features/Attributes)  The random variables $X_i$, $i = 1, \ldots, n$, which define an instance space are called Features or Attributes or Variables [44].

Definition 2.1.3 (Instance/Observation)  An element $x \in X$ is called an Instance or Observation [44].

Definition 2.1.4 (Supervised Learning)  A Machine Learning technique which aims at finding a function $f : X \to Y$ deriving the label value $y \in Y$ from a given observation $x \in X$, where $Y$ is the target variable [44].

Definition 2.1.5 (Loss Function)  A convex function $Q$ with arity 2, positive range, and $Q(x, x) = 0$ is called a Loss Function, where the arity of a function is defined as the number of arguments [44].

Definition 2.1.6 (Weighted Sum Method)  A method which scalarizes a group of objectives into one objective by adding each objective pre-multiplied by a scalar weight, where the weight of an objective is chosen in proportion to the relative importance of the target [21].

Disadvantage: Weighted Sum Methods cannot find certain Pareto-optimal solutions within the case of a non-convex objective space, and only work for convex objective space [21].
**Definition 2.1.7 (ε-constraint Method)** An ε-constraint method converts a multi-objective optimization problem into a single-objective optimization problem by retaining one of the objectives, converting the rest of the objectives into constraints with specific user-defined values [21].

Advantage: ε-constraint methods are applicable to either convex or non-convex problems.

Disadvantage: The ε vector must be chosen carefully so it is within the minimum or maximum values of the individual objective function [21].

**Definition 2.1.8 (Weighted Metric Method)** Combines multiple objectives using the weighted distance metric of any solution from the foremost effective solution $Z^*$ [21].

Advantage: Guarantees finding all Pareto-optimal solutions with ideal solution $Z^*$. Disadvantages: Requires knowledge of minimum and maximum objective values, requires solution $Z^*$ which may be found independently optimizing each objective function. For small $p$-value, not all Pareto-optimal solutions are obtained and as $p$ increases, the method becomes non-differentiable [21].

**Definition 2.1.9 (Classification Learning)** If the label $Y$ is constrained to $(-1, +1)$, the classification task is called binary classification or binomial classification learning. A classification problem with number of classes greater than 2, can be transformed into a set of binomial classification problems by techniques like one-vs-one or one-vs-all [44].
Definition 2.1.10 (Regression Learning) If the label space $Y$ is not discrete but continuous, for example $Y = \mathbb{R}$, no predefined classes exist and the task becomes assigning a numeric value to new/unseen observations instead of classifying the unseen data points into the predefined classes. This learning task is called regression learning and it is a generalized case of the classification problem [44].

In both classification and regression predictive modeling machine learning tasks, the goal is to find a function $f$ which predicts the correct value for a given observation. In order to achieve this, we define a loss function which must be minimized, which is defined as the penalty of error during the learning process [44, 28].

Definition 2.1.11 (Unsupervised Learning) In some cases, there is no specific label attribute $Y$ and hence, it is not possible to learn a function $f$ which can be used to predict the value of $Y$ for unseen data points $x \in X$ [44].
2.1.1 Supervised Machine Learning Classification Methods

In this section, we briefly describe some of the main classification and regression algorithms used in the machine learning literature [28, 12]. We concentrate our discussion on those algorithms that we use in the constructions of our proposed method, and those used for performing computational experiments in subsequent Chapters 3 and 4.

2.1.1.1 Decision Trees

Decision tree methods construct a model of decisions made based on actual values of features in the data. Decision trees are trained on data involving both classification and regression problems. Decision trees are often fast and accurate, but also known for their over-fitting feature. Each node represents a splitting rule for one specific feature. For classification task, this rule separates values belonging to different classes, whereas for regression task, it separates them in order to reduce the error in an optimal manner for the selected criteria [47, 46]. The building of new nodes is repeated until the stopping criteria are met. A prediction for the class label is determined depending on the majority of observations which reach this leaf during a generation, while an estimation for a numerical value is obtained by averaging the values in a leaf. The label attribute is nominal for classification and numeric for regression. Some variations of decision trees algorithm found in literature are C4.5, [60], ID3, Random Forest, and Classification and Regression Trees (CART) [46]. The criterion used for measuring the accuracy of a test is based on measures, such as entropy or information gain.
2.1.1.2 Random Forests

A random forest is an ensemble of random trees, specified by the number of trees. Each node of a tree represents a splitting rule for one specific feature. Only a subset of features, specified by the subset ratio criteria, is considered for the splitting rule selection. This rule separates values in an optimal manner for the selected parameter criterion. For classification the rule separates values belonging to different classes. For regression, it separates them in order to reduce the error made by the estimation. The process of building of new nodes is repeated until the stopping criteria is met. A concept called pruning can be leveraged to reduce complexity of the model by replacing sub-trees with less predictive power [47].

2.1.1.3 Gradient Boosted Trees

A gradient boosted model is an ensemble of either regression or classification tree models. Both are forward-learning ensemble methods that obtain predictive results through gradually improving estimations. Boosting is a flexible nonlinear regression procedure that helps improving the accuracy of trees. By sequentially applying weak classification algorithms to the incrementally changed data, a series of decision trees are created that produce an ensemble of weak prediction models. While boosting trees increases their accuracy, it also decreases speed and human interpretability. The gradient boosting method generalizes tree boosting to minimize these issues [47].
2.1.1.4 Support Vector Machines (SVM)

SVM takes a set of input data and predicts for each given input, the two possible classes comprising the input, thus making the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of the two classes, an SVM training algorithm builds a model that assigns new observations by constructing a hyperplane or set of hyperplanes in a high- or infinite- dimensional space, which can be used either for classification or regression. A functional margin of separation is achieved by the hyperplane that has the largest distance to nearest training data points of any class. In general, the larger the margin the lower the generalization error of the classifier [46].

2.1.1.5 Neural Networks/Deep Learning

A neural network consists a set of nodes or connections, having an associated weight. A network is typically represented as a graph organized in layers. The first layer consists of as many nodes as the number of input variables in the data, with each node in that layer receiving a single input corresponding to the associated input variable. The last layer consists of as many nodes as the number of classes in the data, each node having no connection to other nodes. For two-class classification problems, the final layer typically contains a single node, determining a binary output. The intermediate layers, if any, are present to allow an extended flexibility of the network, so that it can learn complex functions. Typically a stochastic gradient descent algorithm is utilized to update the weights to globally minimize the number of errors made on the set of training observations. Variations in the cost function allow the adjustment of the algorithm to a classification or regression setting [9, 60].
Deep Learning is based on a multi-layer feed-forward artificial neural network that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons. Advanced features such as adaptive learning rate, rate annealing, momentum training, dropout and L1 or L2 regularization enable high predictive accuracy [47].

2.1.1.6 Naïve Bayes

Naïve Bayes is a high-bias, low-variance classifier that builds good models even for small scale data-sets. The fundamental assumption of this computationally inexpensive method is that the features are independent of each other, given the value of the target. The independence assumption vastly simplifies the calculations needed to build the Naïve Bayes probability model. Assumptions about conditional probability distributions such as the Gaussian probability densities for individual features need to be made to compute the probability model [47].

2.1.1.7 Generalized Linear Models

Generalized Linear Models provide an extension to the traditional linear models by fitting generalized linear models to the data through maximizing the log-likelihood. The elastic net penalty can be used for parameter regularization. The model fitting computation is parallel, extremely fast, and scales well for models with a limited number of predictors with non-zero coefficients [47].
2.1.1.8 Logistic Regression

Logistic regression fits a linear model of the independent variables and is used when the dependent variable is categorical. Logistic regression is a linear method, but the predictions are transformed using the logistic function, predicting a nonlinear function of the dependent variable. Logistic regression uses Maximum Likelihood Estimation (MLE) to maximize the likelihood that the dependent variable values are obtained from the values of the independent variables.

2.2 Feature Selection Methods

Feature Selection, also referred to as variable selection or attribute subset selection or support-set selection is a Machine Learning data prepossessing task for classification or regression for data reduction by selecting signal and discarding the noise present in the underlying data. If left untouched, using all the original features from the data-set can result in the machine being trained on and modeled to learning the noisy patterns rather than the actual signal, thereby resulting in a biased model which only performs well on the training data-set and fails significantly when exposed to future unseen data [53]. Feature selection is mathematically formulated as a combinatorial optimization problem, where the function to optimize is the generalization performance of the predictive model. The design variables are the inclusion, 1, or the exclusion, 0, of the individual features. An exhaustive selection of all original features would evaluate a subset of different combinations, mathematically given by $2^n$, where $n$ is the number of features. The general machine learning framework incorporating the different feature selection methods is displayed in Figure 2.1.
2.2.1 Wrapper Methods

Wrapper methods work by evaluating a subset of features employing a Machine Learning algorithm using an optimization search strategy through the space of possible feature subsets, evaluating each subset supported through the evaluation of the performance of the underlying wrapper-based Machine Learning algorithm. These methods are greedy-type algorithms and are computationally expensive [18, 50].
Wrapper methods present two main advantages:

- They detect the interaction between variables.
- They find the optimal feature subset for the desired Machine Learning algorithm, resulting in better predictive accuracy than filter methods.

**Wrapper Algorithm:**

- Looks for a subset of features selecting a subset from the available ones using a top-down or bottom-up approach.
- Builds a Machine Learning model by training a specific algorithm on the previously-selected subset of features.
- Evaluates model performance of the underlying Machine Learning algorithm, evaluating the newly-trained model with a specific performance metric.
- Repeats the whole process until no improvement is observed in solution.

Some common wrapper methods are Forward Feature Selection, Backward Feature Elimination, and Bidirectional Search [38, 18, 17]. We present the Forward Selection and Backward Feature Elimination methods in the next subsection, as we will be using them to perform computational experiments comparing against our proposed evolutionary feature selection approach in Chapter 4.
2.2.1.1 Forward Selection

Forward Selection is an iterative method in which we first start with having no features within the model. In each iteration, we keep adding the feature which best improves our model until the addition of a new feature does not improve the performance of the model [45].

The steps for Forward Selection technique are as follows:

- Step 1: Choose a Significance Level
- Step 2: Fit models by considering one feature at a time, selecting the features with a small \( p \)-value. A total of \( N \) models are possible.
- Step 3: Fit all possible models with one extra feature added to the previously selected feature(s).
- Step 4: Select the features with minimum \( p \)-value. If \( p \)-value of significance level is less than the set confidence interval, then visit Step 3, otherwise terminate the strategy.

2.2.1.2 Backward Elimination

In Backward Elimination, we start with including all the independent features, removing insignificant features with highest \( p \)-value (greater than the significance level). The steps involved in Backward Elimination are as follows [45]:

- Step 1: Choose a significance level with a set confidence interval.
- Step 2: Fit the full model including all the original features.
• Step 3: Consider the feature with highest $p$-value. If the $p$-value $> \text{significance level}$ then visit Step 4, otherwise terminate the strategy.

• Step 4: Remove the feature which is taken into consideration.

• Step 5: Fit the model without the feature from Step 4.

• Step 6: Repeat the whole process from Step 3 until no further improvement.

### 2.2.2 Filter Methods

Filter Methods are commonly used as a data preprocessing step. The choice of features are independent of any Machine Learning algorithms used. Filter type methods select variables irrespective of the model. These methods are particularly effective in computation time and robust to over-fitting [38, 13]. Some commonly used Filter Methods include the following:

• Pearson’s Correlation Coefficient: Used as a measure for quantifying linear dependence between two continuous variables. Its value ranges from -1 to +1.

• Linear Discriminant Analysis (LDA): Used for finding a linear combination of features that separates two or more classes of a categorical variable.

• Analysis of Variance (ANOVA): Similar to LDA aside from the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.
• Chi-Square: Statistical test applied to groups of categorical features to gauge the likelihood of correlation or association between them using their distribution.

• Principal Component Analysis (PCA): Converts original features into the linear combinations of existing ones, called Principal Components. PCA finds the direction of maximum variance in high-dimensional data, projecting it onto smaller dimensional subspace.

2.2.3 Hybrid Methods

Hybrid Methods combine the various approaches to urge the foremost effective possible feature subset. The method starts by performing filter methods by eliminating constant, quasi-constant and duplicated features. Then, within the second step, using wrapper methods to pick out the foremost effective feature subset from the previous step, and vice-versa.

The advantage hybrid methods offer is that they incorporate the effective advantages from other feature selection methods, and can reduce their disadvantages, resulting in the following [17, 52]:

• High performance and accuracy,

• Better computational complexity than wrapper methods,

• Models that are more flexible and robust against high dimensional data.
2.2.4 Evolutionary Feature Selection

Figure 2.2 shows an illustration of Evolutionary Feature Selection approach adapted from RapidMiner [45].

![Evolutionary Algorithm Framework Diagram]

Figure 2.2: Illustration of Evolutionary Algorithm Framework
 Evolutionary Feature Selection Algorithm:

1. **Crossover**: Parents create offspring.
   - Single split point at the same position for both parent bit vectors.
   - Create off-springs using front part of the first parent combined with the back part of the second parent and vice-versa.
   - Can lead to large jumps in the fitness landscape. These jumps allow evolutionary algorithms to cope much better with multi-modal fitness landscapes.
   - Less likelihood to be stuck in a local extremum [45].

2. **Mutation**: Individuals undergo small changes.
   - Process of flipping a single bit vector from 0 to 1 and vice-versa.
   - Likelihood for flipping the selection is $1/n$, where $n$ is the number of features.
   - Leads to a small movement in the fitness landscape [45].

3. **Evaluation**: Evaluate all individuals in the current population.

4. **Selection**: Selection based on survival of the fittest, i.e., likelihood for survival is higher for fitter individuals.
2.2.5 Comparison: Feature Selection Techniques

Below is an overall comparison of the feature selection methods we have discussed in this Chapter.

**Brute Force**

- Delivers optimal results, but is computationally not feasible on large data-sets due to the evaluation of $2^n$ feature subsets, where $n$ is the number if features.

**Forward Selection/Backward Elimination**

- Delivers results quicker.
- Runs into the first possible local optimum.
- In most cases, fails to deliver the optimum result.

**Evolutionary Algorithms**

- Feasible even for large data-sets.
- Better performance than forward selection/backward elimination.
- Reliable optimization algorithm for multi-modal fitness landscapes.
The filter-based feature selection approaches involve manually setting a threshold for the correlation coefficient, and hyper-parameter tuning. This type of a manual choosing of the given trade-off factor may not be feasible in real-life. The main disadvantage of this approach is that many separate optimizations with different weighting factors need to be performed to examine the trade-offs among the objectives as given by the equation (2.1):

\[ E_{\text{reg}}(\omega) = E_{\text{emp}}(\omega) + C \cdot \Omega(\omega) \]  

where \( E_{\text{reg}}(\omega) \) is the regularized risk, \( E_{\text{emp}}(\omega) \) is the empirical risk, \( C \) is the trade-off factor, \( \Omega(\omega) \) is the structural risk, and \( \omega \) is the function parameter [44].

### 2.3 Multi-Objective Optimization

A general Multi-Objective Optimization Problem (MOOP) [22, 6] is of the form as shown in equation (2.2).

\[
\begin{align*}
\min \text{(max)} & \quad Z_m(X), \quad m = 1, 2, \ldots, M \\
\text{subject to} & \\
q_j(X) & \geq 0, \quad j = 1, 2, \ldots, J \\
r_k(X) & = 0, \quad k = 1, 2, \ldots, K \\
x_i(L) & \leq x_i \leq x_i(U), \quad i = 1, 2, \ldots, n
\end{align*}
\]  

(2.2)
where \( X = (x_1, x_2, ..., x_n)^T \) is a vector of \( n \) decision variables, \( Z \) is the objective function, \( q_j, j = 1, 2, ..., J \) are linear/nonlinear inequality constraint functions, \( r_k, k = 1, 2, ..., K \) are linear/nonlinear equality constraint functions, \( x_i(L) \) and \( x_i(U) \) are the lower and upper bounds of \( x_i, i = 1, ..., n \), respectively.

Multi-objective optimization is naturally inherent in data mining, consisting of different criteria which need be optimized. In classical statistical learning methods, multiple objectives are usually combined as a single-objective scalar function using a weight-based trade-off factor, which involves determining this trade-off apriori, requiring the need to find a set of solutions (Pareto-set) with respect to two criteria, namely error and model complexity. This approach can also be used to solve non-convex optimization problems with multiple objectives, and if only a unique optimum exists, the solutions of the resulting Pareto set would collapse into one single solution [44]. Therefore, Multi-Objective Optimization problems deal with competing objectives, i.e., while one objective increases the other decreases, and there is no unique global solution but a set of Pareto-optimal solutions.

**Definition 2.3.1 (Feasible Solution)** A solution that satisfies all constraints and variable bounds. The set of all feasible solutions is called the feasible region, or \( S \). The objective space is constituted by the possible values of the \( Z \) objective functions for all solutions in \( S \) [16].
Definition 2.3.2 (Domination) A solution $x_1$ is said to dominate the other solution $x_2$ if both conditions 1 and 2 below are true:

- Condition 1: $x_1$ is no worse than $x_2$ for all objectives.
- Condition 2: $x_1$ is strictly better than $x_2$ in at least one objective.

The mathematical notation for $x_1$ dominates $x_2$ is: $x_1 \preceq x_2$ [16]

Definition 2.3.3 (Non-Dominated Set) Among a set of solutions $X$, the non-dominated set of solutions $X'$ are those that are not dominated by any member of the set $X$ [16].

Definition 2.3.4 (Globally Pareto-Optimal set) The non-dominated set of the entire feasible search space $S$ is defined as the Globally Pareto-Optimal set [16].

2.3.1 Multi-Objective Evolutionary Algorithms (MOEA)

While the traditional optimization methods treat a single candidate solution, Evolutionary/Genetic Algorithms (EA/GA) fundamentally treat multiple candidate solutions [21]. Below are a few commonly used multi-objective evolutionary algorithms found in literature, which are separated into two categories based on the elitism property that preserves an already found Pareto-optimal solution from being deleted by carrying forward the elites from one generation to the other [52, 56].
Non-Elitist MOEAs:

- Vector Evaluated Genetic Algorithm
- Vector Optimized Evolutionary Strategy
- Weight-based Genetic Algorithm
- Multi Objective Genetic Algorithm (MOGA)
- Non-dominated Sorting Genetic Algorithm (NSGA)

Elitist MOEAs:

- Distance-based Pareto Genetic Algorithm
- Pareto-based Genetic Algorithm
- Non-dominated Sorting Genetic Algorithm (NSGA-II)

Elitist Non-dominated Sorting GA: Advantages

- The diversity among non-dominated solutions is maintained using the crowding procedure. No extra diversity control is required.
- Elitism protects an already found Pareto-optimal solution from being deleted.

Elitist Non-dominated Sorting GA: Disadvantage

- When there are over $K$ members within the primary non-dominated set, some Pareto-optimal solutions may give their places to other non-Pareto-optimal solutions.
2.3.1.1 NSGA-II: Non-Dominated Sorting Genetic Algorithm

NSGA-II is a multi-objective evolutionary algorithm. Evolutionary algorithms were developed because the classical direct and gradient-based techniques have the following problems when leading with non-linearities and complex interactions:

1. The convergence to an optimal solution depends on the chosen initial solution.
2. Most algorithms tend to get stuck to a sub-optimal solution.

NSGA-II algorithm [22, 61] has been demonstrated as one of the most efficient algorithms for multi-objective optimization. It incorporates a fast non-dominated sorting approach with computational complexity of $O(pm^2)$ (where $p$ is the number of objectives and $m$ is the population size). It uses the fast non-dominated sorting to rank the population fronts using a parameter called crowding distance, where the value of a particular solution is the average distance of its two neighboring solutions, calculated in the same front. Then, tournament selection is made between two individuals randomly selected from parent population. The individual with lower front number is selected if the two individuals come from different fronts. The individual with higher crowding distance is selected if the two individuals are from the same front. Then, both the crossover and mutation operators are used to generate a new offspring population. Finally, the parent and offspring populations are combined together using a fast non-dominated sorting and crowding distance assignment, in order to rank the combined population. Only the best individuals are selected as the new parent population.
Thereby, NSGA-II has well distributed non-dominated solutions in the Pareto front. It has global search ability of evolutionary algorithms combining the search ability of local search strategy simultaneously, and exhibits the following three properties [21]:

(i) It uses an elitist principle, i.e., the elites of a population are given the opportunity to be carried over to the next generation.

(ii) It uses an explicit diversity preserving mechanism (crowding distance).

(iii) It emphasizes the non-dominated solutions.

**NSGA-II Algorithm** [22]

Step 1. Population Initialization.

Step 2. Non-dominated sorting process of the population that has been initialized.

Step 3. Once the sorting is complete, the crowding distance value is assigned front wise. The individuals in population are selected with the best rank and crowding distance.

Step 4. Selection of individuals is allotted employing a binary tournament selector with crowded-comparison operator.

Step 5. Genetic Operators are coded using simulated binary crossover and polynomial mutation.

Step 6. Offspring and current generation populations are recombined and the individuals of subsequent generations are set selectively.

Step 7. The new generation is filled by each front until the population size exceeds the current population size.
2.4 WEKA Machine Learning Methods

In this section, we list three Machine Learning methods from WEKA data mining software [60] which we will use to integrate with our proposed methodology in the subsequent chapters.

2.4.1 Multi-Objective Evolutionary Search

Multi-Objective Evolutionary Search [35] explores the feature space using the NSGA-II Multi-objective Evolutionary Algorithm. Two objectives are optimized, where the first one is to be maximized and is chosen by the evaluator, and the second one is to minimize the subset cardinality. The non-dominated solution in the last population with the best fitness for the first objective is displayed as the output, with the following characteristics:

- Fixed-length binary representation.
- Uniform random initialization.
- Binary tournament selection.
- Ranking based on non-domination level with crowding distance.
- Self-adaptive variation operators (uniform crossover, one-bit flip mutation).
2.4.2 Logical Analysis of Data: Max Rule Generator

Implements the Maximized Prime Patterns heuristic described in the *Maximum Patterns* in data-sets [10]. It generates one pattern (rule) per observation, while attempting to do the following:

(i) Maximizes the coverage of observations belonging to the same class.

(ii) Prevents the coverage of too many observations from outside the same class. The amount of spurious coverage allowed is controlled by the minimum purity parameter.

**Definition 2.4.1** A maximum positive pattern is a positive pattern of maximum coverage in the positive class [10].

**Definition 2.4.2** A maximum negative pattern is a negative pattern of maximum coverage in the negative class [10].

2.4.3 Multi-Objective Evolutionary Fuzzy Classifier

The Multi-Objective Evolutionary Fuzzy Classifier [35] constructs a fuzzy rule-based classifier by using the ENORA or NSGA-II Multi-objective Evolutionary Algorithm. Two objectives are optimized, where the first one can be configured to maximize the accuracy, or to maximize area under the Receiver Operating Characteristic (ROC) curve, or to minimize root mean squared error, and the second objective is to minimize the number of fuzzy rules of the classifier. The non-dominated solution in the last population with the best fitness for the first objective is shown as output. Multi-objective Evolutionary Fuzzy Classifier is a multivariate feature selection method in which both the search strategy and classifier are based on multi-objective evolutionary computation with the following characteristics:
• Variable-length representations with float points and categorical input variables characterized by a population of variable length rule-sets where each rule-set is a potential solution.

• Random initialization where individuals are uniformly distributed with respect to the number of rules indicated in the constraints (maxRules).

• All individuals contain at least one rule for each possible output.

• Binary tournament selection. Constraint handling using a repair algorithm.

• Ranking based on non-domination level with crowding distance.

• Self-adaptive variation operators which work on different levels of the fuzzy classifier.

2.4.4 Comparative Study of Multi-Objective Optimization Methods in Literature

Evolutionary Multi-Objective Algorithms are applicable to a wide-range of problems and their population based approach assures an inherent set of best qualified solutions. Therefore, they are well-suited methods, whenever analytical methods fail to induce the precise Pareto-set. Multiple results on standard benchmark problems indicate that the Non-Dominated Sorting Genetic Algorithm (NSGA-II) is well-suited for Pareto optimization with two and three objectives. It clearly outperforms established techniques like Strength Pareto Evolutionary Algorithm 2 (SPEA2), Evolutionary Multi-Objective Optimization Algorithm (EMOA), and Multi-Objective Genetic Algorithm (MOGA), both in terms of convergence and
performance. The test problems used in literature reveal that results are well-distributed on the Pareto-front. Besides the comparison on state-of-the-art benchmarks, the NSGA-II has been tested on challenging real-life application problems [62, 40, 1].

Simulation results on difficult test problems show that NSGA-II is able, for most problems, to find a much better spread of solutions and better convergence near the true Pareto-optimal front. We test the application of NSGA-II using the Python code [8, 2], on a Multi-Objective Optimization Constraint Test Function, Zitzler–Deb–Thiele’s function N. 1 (ZDT1) [23], given by the two objective functions, \( f_1(x) \), \( f_2(x) \) in equations (2.3) and (2.4) and one constraint function \( g(x) \) given in equation (2.5) [26, 8].

\[
\begin{align*}
\text{Minimize} & \quad f_1(x) = x_1 & (2.3) \\
\text{Minimize} & \quad f_2(x) = g(x) \left[ 1 - \sqrt{x_1/g(x)} \right] & (2.4) \\
\text{Subject to} & \quad g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i & (2.5) \\
& \quad 0 \leq x_i \leq 1, \quad i = 1, ..., n
\end{align*}
\]
As the number of simulation runs increases, the Pareto-plot (red points) in Figure 2.3 converges to the true Pareto-optimal front (solid black curve) representing the optimal values of $f_1(x)$ and $f_2(x)$. The true Pareto-front (solid black curve) is obtained by running 100 generations of the NSGA-II algorithm using the Python code [8, 2], on the given Test Function, $ZDT1$ defined by equations (2.3), (2.4) and (2.5).

Figure 2.3: Pareto Plot of ZDT1 Test Function obtained using NSGA-II in Python
2.5 Logical Analysis of Data (LAD)

Consider a doctor who would like to find out the combination of food items which causes headache to one of his patients, and asks the patient to maintain a daily record of food intake for a week as shown in Figure 2.4. The following questions capture the essence of LAD methodology:

- How to come up with a short list of variables (items) sufficient to explain the presence or absence of a disease?
- How to detect patterns (i.e., combinations of variables) causing the disease?
- How to build theories (i.e., collection of patterns) explaining every observed disease?

<table>
<thead>
<tr>
<th>Day</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
<th>Item 6</th>
<th>Item 7</th>
<th>Item 8</th>
<th>Headache</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 2.4: Example: Patient Diet Daily Intake Record

2.5.1 Methodology

Introduced and developed by Peter L. Hammer in 1986, Logical Analysis of Data (LAD) is a pattern-based two-class classification technique which integrates principles of combinatorics, optimization, and the theory of Boolean functions [11].
LAD is a multi-step procedure consisting of discretization, support set selection, pattern generation, classification, and cross-validation. The key ingredient of two-class LAD method is the identification of patterns, distinguishing between pairwise disjoint subgroups of observations in a data-set [32, 3].

2.5.2 Main Components of LAD

The three main components of LAD are: Discretization/Binarization, Pattern Generation, and Formulation of Theory/Model. We describe in detail each of these components in the following subsections.

2.5.2.1 LAD: Discretization/Binarization

Discretization or Binarization is the process of transforming numerical features to categorical or discrete ones. Discretization helps reduce noise in the data, improves accuracy, and produces robust results. In any given data-set, a large number of feasible cutpoints are identified by standard methods such as equal width, equal intervals, methods based on entropy, chi-square tests, etc. in order to discretize the features. Many of these cutpoints may be redundant. LAD can reduce the set of cutpoints to include only those which are necessary by solving a set-covering problem. Another set-covering problem is solved to identify the smallest subset of variables, called the support set, which can distinguish positive observations from negative ones [58, 5].
2.5.2.2 LAD: Pattern Generation

Patterns are logical rules which identify clusters of positive and negative observations that are very similar in description. A pure positive pattern is defined as a rule which covers only positive points and none of the negative points. Similarly, a pure negative pattern covers only negative observations and none of the positive ones [58, 5]. There are several parameters associated with fuzzy LAD patterns. 

Coverage of a pattern $P$ is defined by the number of samples covered by $P$. 

Degree of a pattern $P$ is defined by the number of conditions in $P$. 

Homogeneity of a pattern $P$ is defined by the proportion of positive samples covered by $P$ to the total number of samples covered by $P$. 

Prevalence of a pattern $P$ is defined as the proportion of positive samples covered by $P$ to total number of positive samples in the data-set.

2.5.2.3 LAD: Theory and Model

LAD model is a collection of positive and negative patterns which provide the same separation of the positive and negative points as the entire set of patterns generated in the pattern generation step. The entire collection of patterns is referred to as a Pandect. An LAD model is obtained from the Pandect by solving a set-covering problem. When the size of the Pandect is very large, efficient greedy heuristics are used to solve the set-covering problem.
Given a binary data-set, \( \Omega = \Omega^+ \cup \Omega^- \), where \( \Omega^+ \cap \Omega^- = \emptyset \), a pattern \( P \) is a sub-cube of \( \{0, 1\}^n \), where \( n \) is the number of features in the data-set:

\[
P = \bigwedge_{j \in N_P} x_j,
\]

where \( N_P \subseteq \{1, \ldots, n\} \) and \( x_j \in \{0, 1\} \). We define a pattern \( P^* \) to be strong (strict) Pareto optimum for the multi-objective problem if and only if there is no other pattern \( P \) such that \( \text{Coverage}(P) \subset \text{Coverage}(P^*) \), where

\[
\text{Coverage}(P) = \{ x \in \Omega : P(x) = 1 \}.
\]

\( P^* \) is a weak (non-strict) Pareto optimum if and only if \( \text{Coverage}(P) \subseteq \text{Coverage}(P^*) \) for some pattern \( P \) [58, 5]. A maximum positive pattern \( P \) is defined as a positive pattern of maximum coverage in the positive class. Similarly, a maximum negative pattern \( N \) is a negative pattern of maximum coverage in the negative class [10].

In Chapter 3, we propose an alternative classification method, called Multi-Objective Evolutionary Hybrid Classification Algorithm (MOEHCA), for two-class binary classification tasks. As the name suggests, our method incorporates the evolutionary feature selection using NSGA-II for the underlying Machine Learning classification problem, thereby combining the advantages of filter, wrapper, and evolutionary search methods, in a hybrid multi-objective approach.
Chapter 3

Multi-Objective Evolutionary Hybrid Classification Approach

In this Chapter, we develop a two-class classification method, called as Multi-Objective Evolutionary Hybrid Classification Algorithm (MOEHCA). As the name suggests, our method incorporates the evolutionary feature selection using the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) for the underlying Machine Learning classification problem, thereby, combining the advantages of filter, wrapper, and evolutionary search methods, in a hybrid multi-objective approach. We use NSGA-II for finding the trade-off solutions of multiple objectives in a Pareto-front with two objectives of minimizing the error and minimizing the model complexity using a wrapper-based Machine Learning classifier, evaluating the fitness of the objective functions in each generation run of the NSGA-II evolutionary algorithm.
Thereby, we define the Machine Learning binary classification problem as a Multi-Objective Optimization (MOO) problem, solving for minimum degree and maximum accuracy models by simultaneously optimizing the objective function evaluations as individual vector functions, and without the need to determine a trade-off factor apriori.

Two objective functions are simultaneously optimized using a specific performance metric of the underlying Machine Learning classifier. The first objective function is set as minimizing the degree of the feature-subset obtained in each generation run of our experiment. The second objective function can be set as either maximizing the accuracy or maximizing the Area Under Curve (AUC) or minimizing the classification error, using the obtained feature-subset from the first objective. The fitness function is evaluated for each generation using the Pareto-optimal feature subset, and best performing individuals are carried forward to the next generation in an evolutionary non-dominated sorting manner preserving the elites of a generation, as well as maintaining the diversity of the population using the crowding distance obtained within each generation run.

We test the application of our proposed MOEHCA approach on three publicly available real-life medical datasets, including Wisconsin Breast Cancer, Coronavirus Disease (COVID-19), and Pima Indians Diabetes.
3.1 Multi-Objective Evolutionary Hybrid Classification Algorithm (MOEHCA)

The algorithm we propose is structured in two phases:

**Phase-I:** Evolutionary Classification: In the first phase, we integrate the NSGA-II [22] multi-objective optimization algorithm as discussed in Chapter 2, in an evolutionary wrapper-based feature selection approach using a Machine Learning classifier, while optimizing two objective functions: Minimizing the degree of features and maximizing the prediction accuracy of the Machine Learning classifier with information gain criteria, which reduces bias towards multi-valued features/attributes by taking the number and size of branches into account when choosing a feature. We evaluate the fitness of our objective functions by running \( k \)-folding cross-validation experiments within each generation, for a total of \( n \) generation runs, with early stopping criteria if no improvement occurs within \( k \)-rounds of each generation, where \( k < g \).

MOEHCA: Evolutionary Classification – Phase-I

- **Step 1.** Input: Training Data
- **Step 2.** Set Target
- **Step 3.** Optimize Evolutionary Selection using NSGA-II.
  
  * Set minimum number of attributes = 2; range: 1 to \( \infty \).
  
  * Set population size = \( \begin{cases} 
  m/k & \text{if } m \geq n; \ k < m \\
  n/k & \text{if } n > m; \ k < n 
  \end{cases} \)
where $m$ is the number of observations, $n$ is the number of features in the dataset, and $k$ is a positive integer of range: 1 to $\infty$.

* Set maximum number of generations = $g$; range: 1 to $\infty$.

* Set local random seed.

* Initialize: $P_{\text{initialize}} = 1.0$; range: 0.0 to 1.0.

* Set $P_{\text{mutation}} = 1.0$; range: $1/n$ to 1.0.

* Set $P_{\text{crossover}} = 1.0$; range: 0.0 to 1.0.

* Set crossover-type = uniform.

• **Step 4.** Keep best individual from each generation, carrying forward to next generation using $g$ generation runs, with early stopping in $k$ rounds if no improvement occurs.

• **Step 5.** Perform $k$-fold cross validation by setting the same local random seed for reproducibility.

  * Training Phase: Machine Learning classifier.

  * Testing Phase: Apply model and Evaluate Performance – Objective 1: Maximize accuracy or minimize classification error of the classification model. Perform $k$-fold cross-validation experiments in each generation using $g$ generation runs with early stopping criteria if no improvement in solution occurs within $k$ generation rounds.
• **Step 6.** Set Performance Evaluation – Objective 2: Minimizing the size of feature subset/support-set.

• **Step 7.** Repeat Steps 4 through 6 with early stopping if no improvement occurs.

• **Step 8.** Output: Pareto-optimal feature subset plot and confusion matrix.

**Phase-II:** In the second phase, we input the preprocessed data containing the Pareto-optimal feature subset obtained from the evolutionary Phase-I of our approach, and show how this can be used as a training data to any general two-class classification algorithm, running $k$-fold cross-validation experiments for a given $k$, thereby testing the generalization ability of our proposed approach in obtaining minimum degree maximum accuracy models in a Pareto-based multi-objective hybrid manner.

**MOEHCA: Generalized Classification – Phase-II**

• **Step 1.** Input: Preprocessed data containing Pareto-optimal feature subset from Evolutionary Classification Phase-I.

• **Step 2.** Train General Classification Algorithm.

• **Step 3.** Test Data on Classifier Model.

• **Step 4.** Output: Minimum Degree, Maximum Accuracy Model.
3.1.1 Performance Measures

We evaluate the performance of the classification model using the traditional performance measures, including accuracy, which is the proportion of correctly classified observations, precision, which is the proportion of correctly classified positive observations, recall, which is the proportion of actual positives identified correctly, and F1-score, which can be interpreted as a weighted average of the precision and recall, where an F1-score reaches its best value at 1 and worst score at 0.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}
\]

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

\[
F1-Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where \(TP\) is True Positive, \(TN\) is True Negative, \(FP\) is False Positive, and \(FN\) is False Negative. Figure 3.1 illustrates the performance evaluation framework for Evolutionary Classification Phase-I.
Figure 3.1: Evolutionary Classification Phase-I: Performance Evaluation Framework
3.2 Real-Life Applications of MOEHCA

In this section, we present an application of our proposed MOEHCA approach on three real-life medical datasets: Wisconsin Breast Cancer, Coronavirus Disease (COVID-19), and Pima Indians Diabetes using an evolutionary wrapper-based feature selection approach with a Decision Trees classifier specifying the Gain Ratio criteria, defined as the ratio of information gain to intrinsic information, while minimizing the number of features and maximizing the prediction accuracy of the decision tree classifier. The information gain criteria reduces bias towards multi-valued features by taking the number and size of branches into account when choosing a feature. We evaluate the fitness of our objective functions by running 10-folding cross-validation experiments within each generation, for 30 generation runs, with early stopping criteria if no improvement occurs within 10 rounds of each generation.

3.2.1 Medical Application I: Classification of Tumors in Breast-Cancer

We provide an application of the Evolutionary Classification Phase-I of our proposed MOEHCA approach using RapidMiner Studio environment for running our experiments [46], on the Wisconsin-Breast Cancer (WBC) dataset obtained from UCI Machine Learning data repository [59] for the classification of two risk groups of tumors: Benign (B) and Malignant (M), consisting of 9 attributes. The target class distribution is as shown in Figure 3.2.
3.2.1.1 Methodology

The WBC data as obtained from the UCI data repository was already clean with no further need for data preprocessing. We applied our proposed MOEHCA Evolutionary Classification Phase-I with 10-folding cross-validation experiments in 30 generation runs. We specified early stopping criteria if no improvement within 10 rounds of each generation run, evaluating our objective functions’ fitness value in each generation run by minimizing the number of features used in the decision tree model and maximizing the prediction accuracy of the Decision Tree classifier using RapidMiner Studio [46].
3.2.1.2 WBC: Results

The Pareto-optimal solutions were found in Generation 25 of the 10-folding cross-validation experiments with early stopping criteria. Figure 3.3 provides the Pareto-optimal plot obtained in generation 25 of the experiment run with an accuracy of $\approx 96\%$ with four features explaining the WBC MOEHCA Decision Tree model in Figure 3.4.

The risk factors associated with WBC data are provided in Figure 3.5, showing that the important factors which contradict a Benign tumor, are Uniformity of cell shape, Clump Thickness, Normal Nucleoli, and Marginal Adhesion.
Figure 3.3: WBC: MOEHCA Pareto-plot
Figure 3.4: WBC: MOEHCA Decision Tree Model
Figure 3.5: WBC: Risk Factors
3.2.2 Medical Application II: Classification of COVID-19 Patient Outcomes

3.2.2.1 Background and Introduction

2019 Novel Coronavirus (2019-nCoV or COVID-19 or Coronavirus Disease) is a virus first detected in Wuhan, China, which was identified as the cause of an outbreak of respiratory illness. Early on, many of the patients in the outbreak in Wuhan, China reportedly had some link to a large seafood and animal market, suggesting animal-to-person spread. However, a growing number of patients reportedly did not have exposure to animal markets, indicating person-to-person spread. At the time of writing this dissertation, the Coronavirus Disease is a global public health emergency and it is still unclear how easily or sustainably this virus is spreading between people, as defined by the Centers for Disease Control (CDC) [39].

The dataset used in this study had daily level information on the number of affected cases, deaths, and recovery from COVID-19. The data was collected from January 22, 2020 through March 31, 2020, gathered from the Data Repository by John Hopkins Center for Systems Science and Engineering [37, 29, 39].

We applied supervised Machine Learning classification algorithms for classifying the available COVID-19 patient data into two binary groups of recovered and deaths, thereby assessing the performance, interpreting the predictive models built using feature importance plots, and analyzing the patterns/key factors associated with the patients infected by the Coronavirus Disease.
3.2.2.2 Methodology

First, the structured data was preprocessed and staged as a Machine Learning Binary Classification Problem with two outcomes: Recovered (1) and Death (0). The data was further cleaned for missingness in target and a new derived variable named *Days between symptom onset and COVID-19 confirmation* was coded using the available data. Total Patient Population with confirmed COVID-19 binary outcomes (recovered or death) was 667, as shown in Figure 3.6.

Single-objective Machine Learning classification algorithms such as Naïve Bayes, Generalized Linear Model, Logistic Regression, Fast Large Margin, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees, Extreme Gradient Boosting, Support Vector Machines, were built and evaluated in an automated matter using 10-folding cross-validation experiments in RapidMiner’s Auto Model environment [47] as shown in Figure 3.7. The best performing single-objective decision tree performance metrics are listed in Table 3.1 and the corresponding decision tree model is given in Figure 3.8.

Then, our proposed MOEHCA Evolutionary Classification Phase-I experiments were performed with two objectives: maximizing accuracy and minimizing the number of features, using RapidMiner Studio [46]. The performance of our model is validated using 10-folding cross-validation experiments within each generation of the 30 generation runs, with early stopping criteria if no improvement occurs within 10 generation rounds using our proposed MOEHCA Evolutionary Phase-I as shown in Figure 3.9.
Figure 3.6: COVID-19 Patient Data Binary Classification
Figure 3.7: COVID-19: Single-Objective Classification Models
<table>
<thead>
<tr>
<th>Criterion</th>
<th>Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>88.8%</td>
<td>± 5.9%</td>
</tr>
<tr>
<td>Classification Error</td>
<td>13.2%</td>
<td>± 5.5%</td>
</tr>
<tr>
<td>AUC</td>
<td>80.8%</td>
<td>± 7.0%</td>
</tr>
<tr>
<td>Precision</td>
<td>83.3%</td>
<td>± 9.0%</td>
</tr>
<tr>
<td>Recall</td>
<td>56.9%</td>
<td>± 12.3%</td>
</tr>
<tr>
<td>F Measure</td>
<td>56.9%</td>
<td>± 12.3%</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>56.9%</td>
<td>± 12.3%</td>
</tr>
<tr>
<td>Specificity</td>
<td>88.8%</td>
<td>± 1.9%</td>
</tr>
</tbody>
</table>
Figure 3.9: COVID-19: MOEHCA Evolutionary Classification Phase-I Model
An XGBoost classification model [19] for COVID-19 was developed for analyzing model interpretability using Shapley values [43, 42] using Python’s Scikit-Learn Machine Learning library [14, 51]. XGBoost algorithm is a regularized version of Gradient Boosting Machine (GBM), available in the Scikit-Learn compatible API [14, 51] in Python. The objective function is the sum of loss function evaluated over all predictions and a sum of regularization term for all predictors. The Boosting technique can reduce the variance by averaging many different trees (where each one is solving the same problem).

Unlike GBM, where tree pruning stops once a negative loss is encountered, XGBoost grows the tree to a maximum depth and then prunes backward until the improvement in loss function is below a given threshold using the following two criteria:

- In the first iteration, all instance weights are equal.

- Ensemble parameters are optimized in a stage-wise manner calculating optimal parameters for the next classifier, by holding fixed what was already calculated.

The results of our 10-folding cross-validation experiments showed that XGBoost model had an area under the Receiver Operating Characteristic (ROC) curve as 0.93. ROC is a probability curve and the area under the curve (AUC) represents the degree or measure of separability and is used as a performance measurement for the classification problem. AUC tells how much model is capable of distinguishing between classes. Higher the AUC (between 0-1), better the model is at distinguishing between patients with binary outcomes.
Feature Importance types can be defined as follows:

- **Weight**: number of times a feature is used to split the data across all trees.
- **Gain**: average gain across all splits the feature is used in.
- **Cover**: average coverage across all splits the feature is used in.
- **Total Gain**: total gain across all splits the feature is used in.
- **Total Cover**: total coverage across all splits the feature is used in.

The SHAP Summary Plot is a proof from game theory on the fair allocation of profits which leads to a uniqueness result for feature attribution methods in Machine Learning. These unique values are called Shapley values, after Lloyd Shapley who derived them in the 1950’s [43].

Using SHAP Values, the features are sorted by mean\(|TreeShap|\) combining the above different feature importance techniques. Given that we would like to implement a method which is both consistent and accurate at interpreting our classifier model, it turns out there is only one way to allocate feature importance, and therefore, we use the SHAP summary plot instead of the typical bar chart of feature importance. The SHAP summary plot also displays the most important features along with their range of effects over the individual observations in the underlying dataset.
Algorithm for XGBoost Machine Learning Model in Python:

(1) Divide the data into 70% Training and 30% Test sets.

(2) Further divide the Training dataset into Training and Evaluation datasets using 10-fold cross validation.

(3) Perform Hyper Parameter Tuning on the training dataset of the $i^{th}$ folding experiment ($i = 1, \ldots, 10$) to get the optimal parameters.

(4) Perform Early Stopping to check the best early stopping rounds using 10-folding cross validation experiments.

(5) Train using all the data from the training dataset and predict using Evaluation dataset to check the performance, repeating steps 3 and 4 until no further improvement.

(6) Finally, predict using the Test dataset and save the best performing model as the scorecard for predicting on unseen future data.

3.2.2.3 COVID-19: Results

The overall best performing model was the MOEHCA Evolutionary Classification Phase-I model, simultaneously optimizing two objectives: AUC and Accuracy, in a non-dominated pareto-based search criteria, showing an improved accuracy of approximately 89% compared to the single-objective best performing decision tree model at 86% accuracy.
The feature importance and risk plots for COVID-19 are displayed in Figure 3.10 and Figure 3.11, respectively. The COVID-19 risk plot was built for the best performing Decision Tree classification model in RapidMiner Studio [46].

Top five features contributing to COVID-19 Outcomes as shown in the feature importance plot in Figure 3.10 are:

1. Chronic Disease
2. Wuhan (within China) or Not in Wuhan (other countries)
3. Days between symptom onset and COVID-19 confirmation
4. Latitude and Longitude
5. Country: United States

Top chronic diseases contributing approximately 50% of COVID-19 patient population with outcomes (Figures 3.12 and 3.13) are:

1. Hypertension
2. Diabetes
3. Asthma
4. Chronic Kidney Disease
5. Any combination of above chronic diseases
Figure 3.10: COVID-19: Feature Importance Plot Using Shapley Values
Figure 3.11: COVID-19: Risk Plot
Figure 3.12: Chronic Diseases Leading to COVID-19
Figure 3.13: COVID-19: Chronic Diseases by Percentage Contribution
The results in Figure 3.14 show that for those people who ended up dying from COVID-19, the average number of days from severe symptom onset such as acute respiratory failure, pneumonia, septic shock and/or fatigue, to confirmed diagnosis was approximately 1.5 days. It was more than double that, an average of approximately 3.5 days, for those with mild symptoms such as cough and fever who ended up recovering (Figure 3.14).

Latitude and longitude, which measure a location’s north-south and east-west angles respectively, played a key role. Large number of confirmed COVID-19 cases in the studied outcomes were found in locations with low latitude and high longitude, which during the time the data was collected included China, Philippines, Singapore, United States and Vietnam.

Figure 3.15 shows a comparative study of age-groups which was an important factor shown by the COVID-19 classification model. Cases of COVID-19 in the United States, whether resulting in recovery or death, tended to afflict those above 50 years old, whereas in China the range was far broader, between 19- and 80-years-old.
Figure 3.14: Days between Symptom Onset and COVID-19 Confirmation by Outcome Type
Figure 3.15: COVID-19: Patient Age Group Comparison between China and United States
3.2.3 Medical Application III: Pima Indians Diabetes

3.2.3.1 Methodology

We integrate the Evolutionary Classification Phase-I of our proposed MOEHCA approach, with a Multi-Objective Evolutionary NSGA-II Fuzzy Classifier instead of Decision Trees, for the classification of Pima Diabetes obtained from UCI Machine Learning data repository [59] for the classification of those tested positive/negative using the WEKA Machine Learning environment [60], constructing a fuzzy rule based classification model.

The dataset consisted of 768 observations with 8 attributes, involving no missing values as shown below in Figure 3.16. 10-folding cross validation experiments were performed in an automated manner using Decision Trees, Random Forest, Logistic Regression, SVM, Naïve Bayes, and Deep Learning, using the RapidMiner’s Auto Model environment [46], and the NSGA-II based Fuzzy Classification method was run using the WEKA data mining software [35, 60].

The Multi-objective Evolutionary Fuzzy classifier constructs a fuzzy rule-based classifier by using the ENORA or NSGA-II Multi-objective Evolutionary Algorithm as discussed in Section 2.3.1. The NSGA-II based fuzzy classifier was selected based on better performance compared to Decision Trees and other classification algorithms. The first objective was configured to maximize accuracy, and the second one was configured to minimize the number of fuzzy rules of the classifier, optimizing the two objectives simultaneously using 10-fold cross-validation experiments within each of 20 generation runs with initial population size of 100. The final model performance was evaluated using a 25% test-split.
Figure 3.16: Pima Indians Diabetes Dataset: Target Class Distribution
Pima Diabetes MOEHCA Fuzzy Classifier Model Input Parameters:

- Maximum similarity: 0.4
- Minimum variance parameter: 30.0
- Maximum variance parameter: 2.0
- Minimum number of rules: 2
- Maximum number of rules: 12
- Maximum number of labels: 5
- Evaluation criteria: AUC

- Objective 1: AUC, \( f_{min} = 0.0, f_{max} = 1.0 \), maximize
- Objective 2: number of rules, \( f_{min} = 2.0, f_{max} = 12.0 \), minimize

3.2.3.2 Pima Indians Diabetes: Results

The MOEHCA Fuzzy Classifier Model for Pima Indians Diabetes is displayed in Table 3.2 with four Pareto-optimal rules classifying the two groups: Tested Positive and Tested Negative. The overall accuracy of the model with four Pareto-optimal classifier rules was 81.25% with an AUC of 0.7734. The model performance metrics are as listed below:

- Number of rules = 4.0
- Constraint: similarity = 0.4
- Correctly classified instances: 81.25%
- Incorrectly classified instances: 18.75%
Table 3.2: Pima Diabetes: MOEHCA Fuzzy Classifier Model

<table>
<thead>
<tr>
<th>Features</th>
<th>Rules</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
<th>Rule 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preg</td>
<td>Mean</td>
<td>5.7027</td>
<td>1.8976</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plas</td>
<td>Mean</td>
<td>157.3088</td>
<td>18.8541</td>
<td>Moderately High</td>
<td>Moderately High</td>
<td>--</td>
<td>Moderately High</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plas</td>
<td>Mean</td>
<td>197.1088</td>
<td>6.6333</td>
<td>--</td>
<td>--</td>
<td>High</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pres</td>
<td>Mean</td>
<td>30.369</td>
<td>34.7248</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skin</td>
<td>Mean</td>
<td>88.8948</td>
<td>32.3744</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insu</td>
<td>Mean</td>
<td>543.7591</td>
<td>275.8132</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td>Mean</td>
<td>16.4529</td>
<td>2.4196</td>
<td>Moderately Low</td>
<td>Moderately Low</td>
<td>--</td>
<td>Moderately Low</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td>Mean</td>
<td>29.8729</td>
<td>6.3832</td>
<td>--</td>
<td>--</td>
<td>Medium</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pedi</td>
<td>Mean</td>
<td>0.789</td>
<td>0.25</td>
<td>Null</td>
<td>Moderately High</td>
<td>Moderately Low</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pedi</td>
<td>Mean</td>
<td>1.2574</td>
<td>0.1191</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>Mean</td>
<td>67.8325</td>
<td>18.4411</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
<td>Null</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td>Mean</td>
<td>--</td>
<td>--</td>
<td>Tested Negative</td>
<td>Tested Negative</td>
<td>Tested Positive</td>
<td>Tested Positive</td>
</tr>
</tbody>
</table>
Chapter 4

Multi-Objective Logical Analysis of Data (MOLAD)

In this Chapter, we provide a Multi-Objective extension to a two-class classification algorithm, known as Logical Analysis of Data (LAD) as previously discussed under Section 2.5 in Chapter 2, developing a Multi-Objective Logical Analysis of Data (MOLAD) binary classification model.

There are certain instances of Pareto-based Machine Learning approach used in Logical Analysis of Data literature, but much of the research at the intersection of data mining and multi-objective optimization has been focused around the representation of multiple objectives as one weighted scalarized objective with a user-defined trade-off factor. Although inherently multi-objective in nature, these problems throughout most of the literature surrounding LAD have been treated by transforming the multiple objectives as multiple single-objective optimization problems, or transforming all but one objective into constraints [20, 11, 33, 31, 54].
Such scalarization techniques require some form of prior knowledge about the expected solution in order to define the weight-based scalar trade-off factor. Also, it might be possible to miss certain Pareto-optimal solutions in the case of a non-convex objective space [44]. From the viewpoint of multi-objective optimization, there is no single learning model that can satisfy multiple objectives simultaneously. In this sense, Pareto-based multi-objective optimization is one way to deal with the multiple objectives simultaneously [21, 44].

### 4.1 MOLAD Methodology

**Problem:** Suppose an optimal pattern of a given two-class data should be a function of multiple objectives such as minimizing the degree of patterns (number of features used in the description of patterns), maximizing the coverage of positive and negative patterns, minimizing the classification error, and maximizing the accuracy of the applied Machine Learning algorithm.

Possible trade-offs exist between the optimal solutions for the separate objectives, which need to be determined by solving the problem as a multi-objective optimization problem, optimizing the objectives simultaneously without the need to determine a trade-off factor apriori.

In this section, we provide a multi-objective extension to the Logical Analysis of Data problem formulated by Ryoo and Jang [54], by splitting the weight-based single-objective scalar function into two separate vector-based objective functions, thereby solving the problem as a Multi-Objective LAD optimization problem.
Input:

- $D = D_1 \cup D_2$: Two-class binary dataset, $D_1$ and $D_2$, with $m$ observations and $n$ features.
- $C = \{C_1, C_2\}$: Family of two classes $C_1$ and $C_2$ in $D$.

Output: Pareto-optimal strong patterns covering maximum observations from positive class $C_p$, and no observations from negative class $C_k$, where $k \neq p$, and vice-versa.

4.1.1 Pattern Generation: Constraints

Associate a vector $y = (y_1, ..., y_{2n}) \in \{0, 1\}^{2n}$ to pattern $P_{C_p}$, where the components $y_1, ..., y_{2n}$ of vector $y$ are relative to the features:

- $y_j = 1$, $j = 1, ..., n$ ⇒ $j$th binary feature is in $P_{C_p}$.
- $y_{n+j} = 1$, $j = 1, ..., n$ ⇒ the negation of the $j$th binary feature is in $P_{C_p}$.

Since a pattern cannot include both $j$th binary feature and its negation, we impose the following condition [31, 54]:

$$y_j + y_{n+j} \leq 1, \ j = 1, \cdots, n$$ (4.1)

Define a penalty vector $w = (w_1, w_2, ..., w_m)$ as follows:

$$w_i = \begin{cases} 
1 & \text{if } P(o_i) = 0, \ i \in I^+ \\
0 & \text{if } P(o_i) = 1, \ i \in I^+
\end{cases}$$
Then an LAD pattern of degree \( d \) can be obtained as an optimal solution of the following optimization problem [54]:

\[
\text{Minimize} \quad d + \sum_{i \in I^+} w_i \\
\text{subject to} \\
\sum_{j=1}^{2n} o_{ij}y_j + nw_i \geq d, \quad i \in I^+ \\
\sum_{j=1}^{2n} o_{ij}y_j \leq d - 1, \quad i \in I^- \\
y_j + y_{n+j} \leq 1, \quad j = 1, \cdots, n \\
\sum_{j=1}^{2n} y_j = d \\
1 \leq d \leq n \\
w_i, y_j \in \{0, 1\}, \quad i = 1, \cdots, m, \quad j = 1, \cdots, 2n
\]

where
\( d \): Degree of the pattern
\( w_i \): Penalty associated with the non-coverage of a positive pattern \( P(o_i) \)
\( o_{ij} \): Observation \( i \) in the \( j^{th} \) column
\( y \): unknown binary vector
\( I^+ \): index set of positive observations
\( I^- \): index set of negative observations.

Ryoo and Jang [54] proved that an optimal solution of problem (4.2) is a pure pattern with minimum degree and maximum coverage.
4.1.2 MOLAD Model

Note that while problem (4.2) involves two objectives of minimizing the number of binary features used in a pattern and maximizing its coverage, it is formulated as a single objective, scalar weight-based optimization problem. Below, we provide an extension of problem (4.2), proposing a Multi-Objective Logical Analysis of Data (MOLAD) model with two objective functions, \( f_1 \) and \( f_2 \), where the first objective is to minimize the degree, and the second objective is to maximize the coverage of a positive pattern. Problem 4.3 can be defined for negative patterns in a similar manner.

Minimize \( f_1(x) = d \)

Maximize \( f_2(x) = -\sum_{i \in I^+} w_i \)

subject to

\[
\begin{align*}
\sum_{j=1}^{2n} o_{ij}y_j + nw_i & \geq d, \quad i \in I^+ \\
\sum_{j=1}^{2n} o_{ij}y_j & \leq d - 1, \quad i \in I^- \\
y_j + y_{n+j} & \leq 1, \quad j = 1, \cdots, n \\
\sum_{j=1}^{2n} y_j & = d; \quad 1 \leq d \leq n \\
w_i, y_j & \in \{0, 1\}, \quad i = 1, \cdots, m; j = 1, \cdots, 2n
\end{align*}
\]

where \( d \): Degree of the pattern, \( w_i \): Penalty associated with the non-coverage of a positive pattern \( P(o_i) \), \( o_{ij} \): Observation \( i \) in the \( j^{th} \) column, \( y \): unknown binary vector, \( I^+ \): index set of positive observations, and \( I^- \): index set of negative observations.
4.1.3 **Framework of Proposed MOLAD Approach**

The MOEHCA method we developed in Chapter 3 is structured here into two phases incorporating the Multi-Objective LAD extension as follows: Evolutionary Phase-I and MOLAD Phase-II. In the Evolutionary Phase-I, we use the NSGA-II evolutionary algorithm with two objectives of minimizing the degree and maximizing the accuracy of the wrapper-based Decision Tree classifier as the fitness function evaluation with information gain criteria. We run $k$-folding cross-validation experiments, where $k = 10$, in each of 30 generation runs, with early stopping criteria in 10 generation rounds if no improvement occurs in the objective functions’ fitness evaluation. The fitness function evaluation is specified as the accuracy of the underlying decision tree classification model. The preprocessed data containing the Pareto-optimal feature subset/support-set obtained using our Evolutionary Phase-I is then given as an input training dataset to the LAD max rule classifier [30] in MOLAD Phase-II, producing maximum coverage patterns with minimum degree. The performance accuracy of our MOLAD Phase-II model is validated using a 25\% test-split.

The general framework of our proposed MOEHCA Evolutionary Phase-I and MOLAD Phase-II approach is given in Figure 4.1.
Figure 4.1: Framework of Proposed MOEHCA-MOLAD Approach
4.2 Medical Application IV: African-American Study of Chronic Kidney Disease with Hypertension (AASK)

In this section, we provide a specific application of our MOEHCA-MOLAD approach on the African American Study of Kidney Disease and Hypertension (AASK) Proteomics dataset to classify slow and rapid progressors of Chronic Kidney Disease, producing minimum degree and maximum coverage patterns using our proposed approach.

The main function of kidney is to remove excess water and waste products from the blood. It also regulates the levels of minerals such as sodium, calcium, and potassium in blood. One suffers from Chronic Kidney Disease (CKD), also known as Renal Disease, when kidney loses its function gradually, and usually permanently. The rate of decline of kidney function is measured by the Glomerular Filtration Rate (GFR).

CKD is divided into five stages with increasing severity. When the disease reaches its end stage a kidney transplant or dialysis is required to stay alive. According to the 3rd National Health and Nutrition Examination Survey (NHANES III), about 7-12 million people suffer from moderate CKD in the U.S. as shown in Figure 4.2.
Figure 4.2: Prevalence of Renal Disease in USA (Age > 20, NHANES III)
The African American Study of Kidney Disease with Hypertension (AASK) [49, 57], a randomized double-blinded treatment trial, was motivated by the high rate of hypertension-related renal disease in the African-American population and the scarcity of effective therapies. The study involved 1,094 Participants and Investigators and Staff from 21 AASK Clinical Centers and Coordinating Centers, and was sponsored by National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), National Institutes of Health (NIH) Office on Research in Minority Health, and King Pharmaceuticals.

In our experiments of the proposed MOEHCA Evolutionary Phase-I and MOLAD Phase-II, we apply it to the classification of AASK Proteomics dataset with 5751 Protein Peaks, 116 observations, 57 Rapid (Positive Samples), and 59 Slow (Negative Samples) progressors. The target class distribution of the dataset is as displayed in Figure 4.3.
Figure 4.3: AASK Proteomics Data Target Class Distribution [60]
4.2.1 Methodology

In the Evolutionary Phase I of our experiment, we apply the NSGA-II algorithm for evolutionary feature selection minimizing the size of the feature-subset in a wrapper-based Decision Tree classifier. We run 10-folding cross-validation experiments maximizing the accuracy of the Decision Tree classifier with information gain ratio as the criteria for fitness function evaluation. We set the experiment to run for 30 generations with early stopping criteria if no improvement within each of 10 generation rounds using RapidMiner Studio [46], as displayed in Figure 4.4. We obtain the set of Pareto-optimal proteins (Figure 4.5), and input the preprocessed data containing the Pareto-optimal support-set (Table 4.1) as the training data into the MOLAD Phase-II of our experiment using LAD max rule classifier [30] in the WEKA Machine Learning environment [60].

4.2.2 AASK: MOEHCA Evolutionary Phase-I

The process flow for Evolutionary Phase-I using RapidMiner Studio is shown in the Figure 4.4. We start the process by first importing the AASK Proteomics dataset into the process flow. We then optimize the feature selection using NSGA-II evolutionary approach running 10-folding cross-validation experiments within each generation of 30 generation runs with early stopping criteria if no improvement in 10 generation rounds. We evaluate the fitness function using the accuracy of the wrapper-based Decision Tree classifier with information gain criteria, thereby producing Pareto-optimal support-sets with minimum degree and maximum accuracy simultaneously in a multi-objective manner.
Figure 4.4: Phase I: Process Flow Diagram in Rapid Miner Studio. [46]
4.2.3 AASK: MOEHCA Evolutionary Phase-I Results

The degree of the Pareto-optimal patterns obtained are displayed on the $y$-axis in the Pareto-optimal generation plot (Figure 4.5), and are negative in nature since in reality we are minimizing the degree of patterns, which in a multi-objective optimization setting is equivalent to the negative of maximizing the degree of patterns. The $x$-axis in the Pareto-plot shows the accuracy of the wrapper-based classification algorithm, thereby producing Pareto-optimal results in the 25$^{th}$ generation run of the 10-folding cross-validation experiments.

Table 4.1 provides an overview of the model interpretability showing the combination of features selected from the pareto-optimal plot (Figure 4.5). The Pareto-plot displayed in Figure 4.5 shows that as the number of generations increases, the accuracy also increases reaching a maximum accuracy with minimum degree combination of patterns at generation 25 of the Pareto-front. The Pareto-plot shows that 8 protein peak combinations are significant to predict rapid/slow progressors of CKD with an overall classification accuracy of about 86%.

We run $k$-folding cross-validation experiments using the obtained preprocessed Pareto-optimal feature subset data to train widely used Machine Learning classification algorithms using the Phase-II of our proposed MOEHCA’s Generalized Classification approach shown in Chapter 3. We run the Naïve Bayes, Generalized Linear Regression, Logistic Regression, Deep Learning, Decision Trees, Random Forests, and Gradient Boosted Trees, comparing the accuracy and run-times of the different classification models using the automated Machine Learning in RapidMiner Studio Auto Model environment [47, 46]. The ROC comparison chart
as obtained in Figure 4.6 shows that the Decision Trees classifier model has the overall best accuracy of \( \approx 83\% \) with the least amount of run-time. The other machine learning algorithms also have a comparable performance with higher AUC, thereby validating the obtained Pareto-optimal support-set, showing that our proposed MOEHCA’s Evolutionary Phase-I can be generalized using other traditional machine learning methods with comparable performance and accuracy.

### 4.2.4 AASK: MOEHCA-MOLAD Phase II

The pareto-optimal support-set obtained from MOEHCA’s Evolutionary Phase 1 is used as an input training data for MOLAD Phase 2, using the LAD max rule classifier as discussed in Section 2.4.2 [30, 10], producing minimum degree maximum coverage patterns as output in the WEKA machine learning environment [60].

The eight most significant pareto-optimal protein peaks obtained using the Evolutionary Phase-I of our proposed MOEHCA approach are listed in Table 4.2.
Figure 4.5: Support-Set: Pareto Optimal Proteins
<table>
<thead>
<tr>
<th>Index</th>
<th>Features</th>
<th>Names</th>
<th>accuracy</th>
<th>number of attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>M1550, M2156</td>
<td>0.767</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>M1550, M3156</td>
<td>0.767</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>M1550, M1554, M2022, M1556, M2022</td>
<td>0.855</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>M1550, M1554, M3156</td>
<td>0.817</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>M1550, M1554, M1556, M2022, M1556, M2022</td>
<td>0.838</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>M1550, M2022, M1556</td>
<td>0.808</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>M1550, M2022, M3156, M2022</td>
<td>0.817</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>M1550, M1554, M3156, M2022, M1556, M2022</td>
<td>0.838</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>M1550, M2022, M1556</td>
<td>0.808</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>M1550, M1554, M2022, M1556, M2022</td>
<td>0.846</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>M1550, M1554, M3156</td>
<td>0.846</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>M1550, M2156</td>
<td>0.767</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>M1550, M2156</td>
<td>0.767</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>M1550, M2156</td>
<td>0.767</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 4.6: Accuracy, Run-time, and ROC Comparison of AASK Hybrid Classification Model Phase-I.
Table 4.2: Significant Serum Proteomic Features Associated with CKD

<table>
<thead>
<tr>
<th>M1550</th>
<th>M1554</th>
<th>M2022</th>
<th>M3152</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4668</td>
<td>M6912</td>
<td>M8374</td>
<td>M10666</td>
</tr>
</tbody>
</table>

4.2.5 AASK: MOEHCA-MOLAD Phase-II Results

The Phase-II AASK MOLAD model is provided in Table 4.3. The model provides minimum degree and maximum coverage patterns explaining the rapid and slow progressions of Chronic Kidney Disease, consisting of 8 significant Pareto-optimal protein peaks, with 25 positive patterns and 19 negative patterns. Performance evaluation of the obtained MOLAD model on 25% test split shows an overall accuracy of 86% with Correctly Classified Instances: 86.21%; and Incorrectly Classified Instances: 13.79%, using our proposed MOEHCA approach.
Table 4.3: MOLAD AASK Maximum Coverage Model

<table>
<thead>
<tr>
<th>Pattern</th>
<th>M155G (0.14, 1.25)</th>
<th>M1554 (0.6125)</th>
<th>G2023 (4.705, 5.99)</th>
<th>M4812 (6.42)</th>
<th>M4868 (6.895, 1.905)</th>
<th>M6612 (1.498)</th>
<th>M8174 (1.22)</th>
<th>M9066 (0.375)</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>&lt;= 0.625</td>
<td>&gt; 4.193 and &lt;= 1.25</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>&gt; 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 4.705</td>
<td>&gt; 0.625</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>&lt;= 0.625</td>
<td>&gt; 4.705</td>
<td>&lt;= 0.625</td>
<td>&lt;= 4.705</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5</td>
<td>&lt;= 0.625</td>
<td>&gt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P6</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 4.705</td>
<td>&gt; 0.625</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P7</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 4.705</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P8</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P9</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P10</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P11</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P12</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P13</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P14</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P15</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P16</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P17</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P18</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P19</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P20</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P21</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P22</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P23</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P24</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P25</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P27</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P28</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P29</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P30</td>
<td>&lt;= 0.625</td>
<td>&lt;= 0.625</td>
<td>&lt;= 1.25</td>
<td>Positive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Computational Experiments: Comparison of MOEHCA with Traditional Classification Algorithms

In Table 4.4, we present the performance comparison of our proposed MOEHCA approach with five commonly used Machine Learning algorithms, implemented using RapidMiner studio [46] and Machine Learning software package WEKA [60].

The classification methods used in this study for comparison were Support Vector Machines (SVM), Decision Trees, Nearest Neighbors, Neural Networks and the Logical Analysis of Data (LAD). For the first two datasets: Wisconsin Breast Cancer (WBC) and Pima Indians Diabetes, the experiment run information for the five methods: SVM, Decision Trees, Nearest Neighbors, Neural Networks and LAD was studied by Hammer and Bonates [32, 59], and for the AASK dataset, the LAD classification method was studied by Subasi et al. [57].

We compare the accuracy of our proposed MOEHCA method against the results of the traditionally used classification methods running 10-folding cross validation experiments, validating the performance accuracies on test data. In the case of our proposed MOEHCA approach, we run 10-folding cross-validation experiments for 30 generations, specifying an early stopping criteria if no improvement is found within each of the 10 generation runs. For the COVID-19 and AASK Proteomics datasets, we run the traditional classification algorithms as discussed above for comparison with our proposed MOEHCA approach using automated Ma-
It can be seen that the accuracy of our proposed approach is comparable with that of the frequently used classification methods. It is commonly known from the literature that the Wisconsin Breast Cancer dataset is already clean, on which many data analysis methods provide highly accurate diagnostic models. On the other hand, it is known that for Pima Indians Diabetes dataset, it is hard to find accurate computational models [32]. Moreover, the COVID-19 dataset was not applicable to the Nearest Neighbors and LAD methods, due to the algorithm limitations in handling several categorical and text variables with missing values, as the COVID-19 data was extremely noisy and required preprocessing by one-hot encoding and treating missing values as missing. The AASK Proteomics dataset consisting of 5,751 variables required the task of data preprocessing by selecting optimal feature support-sets, in which case, the SVM method failed due to the lack of robustness on large-scale data as the AASK Proteomics dataset, resulting in a high computational time, and eventually running into a run-time error.

Our real-life experiments show that our proposed MOEHCA approach results in a comparable performance on par with the accuracy of other frequently used two-class classification algorithms, as well as simultaneously optimizing multiple objectives in a hybrid evolutionary Pareto-based manner. This study concludes the computational evaluation of our proposed MOEHCA approach for analysis of real-life datasets, showing that our approach is both accurate and generalizable.
Table 4.4: Performance Comparison of Proposed MOEHCA Approach with Traditional Classification Algorithms

<table>
<thead>
<tr>
<th>Classification Algorithms</th>
<th>Support Vector Machines</th>
<th>Decision Trees</th>
<th>Nearest Neighbors</th>
<th>Neural Networks</th>
<th>LAD</th>
<th>MOEHCA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wisconsin Breast Cancer</strong></td>
<td>95.30%</td>
<td>93.50%</td>
<td>94.10%</td>
<td>93.70%</td>
<td>95.00%</td>
<td>95.99%</td>
</tr>
<tr>
<td><strong>Pima Indians Diabetes</strong></td>
<td>77.90%</td>
<td>77.70%</td>
<td>72.60%</td>
<td>74.50%</td>
<td>77.20%</td>
<td>81.25%</td>
</tr>
<tr>
<td><strong>COVID-19</strong></td>
<td>85.00%</td>
<td>86.8%</td>
<td>NA</td>
<td>77.9%</td>
<td>NA</td>
<td>89.37%</td>
</tr>
<tr>
<td><strong>AASK</strong></td>
<td>NA</td>
<td>51.00%</td>
<td>57.76%</td>
<td>50.0%</td>
<td>80.6%</td>
<td>86.21%</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion

The major contribution of this dissertation can be summarized as the development of an alternative Machine Learning approach, called as Multi-Objective Evolutionary Hybrid Classification Approach (MOEHCA), to overcome some of the difficulties of traditional two-class classification algorithms which generally use a weight-based combination of trade-off factors to solve multi-objective optimization problems inherent in data mining. In order to achieve this goal, we define two separate objective functions in a vector form optimizing simultaneously using our proposed MOEHCA methodology, where the first objective is to minimize the degree of the feature-subset and second objective is to maximize the accuracy/AUC of the wrapper-based classifier in the form of Pareto-optimal solutions, without the need to define a trade-off factor apriori.
Our proposed Multi-Objective Evolutionary Hybrid Classification Algorithm uses the NSGA-II method with Machine Learning classifier in Evolutionary Classification Phase I. In the Generalized Classification Phase-II of our proposed approach, we show how the Pareto-optimal feature subset obtained as the output of Phase-I can be used as input training data to train a classification model generalizing to any two-class classification method. To show the generalization applicability of our proposed approach, the method is applied to three real-life medical data-sets: Wisconsin Breast Cancer [59], COVID-19 [37, 29], and Pima Indians Diabetes [59], optimizing multiple objectives simultaneously.

We further analyze the feature importance and risk plots for the Wisconsin Breast Cancer and the COVID-19 classification models, gaining insights into the complexities of the patterns in the underlying data, and the interaction of features resulting in the prediction of the classified risk groups.

The COVID-19 classification model confirms several important insights analyzing the patterns in the modeled COVID-19 data, including the following [41]:

- 50 percent of those who have COVID-19, whether they recovered or died, had a combination of chronic diseases: hypertension, diabetes, asthma, and chronic kidney disease. This highlights how a weakened immune system is a major gateway to COVID-19 and reinforces the importance of paying special attention to people in this category.
For those people who ended up dying from COVID-19, the average number of days from severe symptom onset such as acute respiratory failure, pneumonia, septic shock and/or fatigue, to confirmed diagnosis was 1.5 days. It was more than double that, an average of 3.5 days for those with mild symptoms such as cough and fever who ended up recovering.

Cases of COVID-19 in the United States, whether resulting in recovery or death, tended to afflict those above 50 years old, whereas in China the range was far broader, between 19- and 80-years-old.

Latitude and longitude, which measure a location’s north-south and east-west angles respectively, played a key role. Large number of confirmed COVID-19 cases in the studied outcomes were found in locations with low latitude and high longitude, which during the time the data was collected included China, Philippines, Singapore, United States, and Vietnam.

We further show how the Pareto-optimal feature subset constructed in Evolutionary Phase-I can be integrated with the Logical Analysis of Data (LAD) max rule classifier [30, 10] to produce minimum degree and maximum coverage patterns in a multi-objective manner. The resulting Multi-Objective LAD model is applied to the African-American Study of Chronic Kidney Disease with Hypertension to predict the combination of Pareto-optimal protein peaks with minimum degree and maximum coverage, that are either positive (predict rapid progression) or negative (predict slow progression) bio-markers.
Computational experiments are performed to demonstrate the utility of our proposed MOEHCA approach against widely-used classification algorithms. The results show how our proposed approach achieves comparable performance on par by optimizing multiple objectives simultaneously in a Pareto-based manner, and is both flexible and robust against high-dimensional data.

In a broader sense, our proposed MOEHCA approach explains the interactions among different features which help obtain deeper insights into the underlying patterns in data, which might further help developing the existing knowledge of therapeutic intervention and prognostic study of real-life medical data-sets with multiple objectives and without the need to define a scalar weight-based trade-off factor apriori.
References


[38] S. Kaushik. Introduction to feature selection methods with an example (or how to select the right variables?). Analytics Vidhya, December 2016.


