Feature Selection with the MS-EPSO Algorithm to Predict Cardiac Pathology in Children and Teenagers

Mário Tasso Ribeiro Serra Neto
Computer Science Master's degree
Computer Science Department
2020

Advisor
Inês de Castro Dutra, Assistant Professor, Science Faculty, University of Porto

Co-advisor
Vladimiro Henrique Barrosa Pinto de Miranda, Full Professor, Engineering Faculty, University of Porto

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/_________

Abstract

Appropriate data modelling and feature selection are important steps to produce reliable predictive models that can focus on relevant predictive variables. This work presents a study of the Maximum Search Limitations - Evolutionary Particle Swarm Optimization (MS-EPSO) when applied to the Feature Selection (FS) problem. MS-EPSO is used to select features of 5 distinct datasets composed by a real-world scenario to predict cardiac pathology in children and teenagers and 4 validated benchmarks found at the Machine Learning (ML) literature. Feature selection is combined with ML models and its performance is compared by applying 13 optimization algorithms and 4 distinct traditional FS strategies. A custom and simple ML pipeline is performed in order to, respectively, construct the final model to predict cardiac pathology and evaluate the collection of algorithms for the FS problem. Results and analysis of each experiment show that the FS approach implemented by MS-EPSO can improve prediction quality over other FS methods when using the same machine learning models.

Resumo

Modelagem de dados e seleção de variáveis são importantes etapas para produzir modelos confiáveis que conseguem focar em variáveis com relevância preditiva. Este trabalho apresenta um estudo do *Maximum Search Limitations - Evolutionary Particle Swarm Optimization* (MS-EPSO) quando aplicado ao problema de seleção de variáveis (FS). MS-EPSO é utilizado para selecionar variáveis de 5 distintos conjuntos de dados compostos por um cenário do mundo-real para prever patologia cardíaca em crianças e adolescentes e 4 conjuntos de dados validados encontrados na literatura de Aprendizado de Máquina (ML). A seleção de variáveis é combinada com modelos de ML e sua performance é comparada pela aplicação de 13 algoritmos de otimização e 4 distintas estratégias de FS. Um procedimento customizado e um simples de ML é aplicado para, respectivamente, construir o modelo final para prever patologia cardíaca e avaliar a coleção de algoritmos para o problema de FS. Os resultados e analises de cada experimento mostram que a abordagem de FS implementada no MS-EPSO podem melhorar a qualidade das predições em relação a outros métodos de FS quando utilizando os mesmos modelos de ML.

Acknowledgments

A special thank my grandmother for the encouragement to face a new challenge across the sea, where together with my parents, they were able to provide the necessary support for completing this dissertation and course.

A thank to my advisors, where even with their tight schedule, they were always there to challenge and guide me through the path of the course and dissertation.

A thank to professor Leonel Carvalho, which gave assistance during multiple brainstorms to design the algorithm applied on this dissertation

Finally, a thank to friends that I met in Porto and to the Zé's group (Brazilian friends), you all helped me in some way during these years and I am grateful for that. Also, to a special person who accepted the hardest challenge with me, even if the distance was not friendly with us, you helped with everything and will always have a special place in my heart.

Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

ABC	Artificial Bee Colony
ABC+ES	ABC + Evolution Strategies
ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BPSO	Binary PSO
CE+EPSO	Cross Entropy Method + EPSO
CP	Cardiac Pathology
CV	Cross-validation
CSO	Competitive Particle Optimizer
DT	Decision Tree
DEEPSO	Differential Evolution EPSO
EA	Evolutionary Algorithms
E-	Embeded Feature Selection
EPSO	Evolutionary Particle Swarm Optimization
FE's	Function Evaluations
FS	Feature Selection
GA	Genetic Algorithm
HEL	History Emergency Level
K-CV	k-Fold Cross Validation
KNN	K-Nearest Neighbors
LR	Logistic Regression
MWU	Mann-Whitney U
MI	Mutual Information
ML	Machine Learning
MS-EPSO	Maximum Search Limitations - EPSO
OHE	One-hot-encoding
OvALL	One vs All
PSO	Particle Swarm Optimization
RF	Random Forest
SI	Swarm Intelligence
SK-CV	Stratified K-CV
W-	Wrapper Feature Selection

Chapter 1

Introduction

Studies on the medical area regarding Cardiac Pathology (CP) are constantly present in the literature. In 1850 [1], heart disease researches were conducted through a medical-patient follow-up, analyzing during months the behavior of a specific disease. After more than a century, according to [2], the state of art of cardiac pathology was advanced due to a new technique named endomyocardial biopsy, allowing a better understanding of the disease behavior and later, with the advance of the technology, this technique was enhanced, where the doctor was able to visualize a real-time procedure using a monitor. In the last decades, systems were created to store records about patients and diseases, leading to better diagnostics [3]. In the present, even more sophisticated methods are employed to circumvent distinct problems, for instance: statistical analysis of age and genre to verify the influence of cardiac pathology in patients with marfan syndrome; and the applications of deep learning methods to classify heart pathology's based on the heart murmur [4].

Computational research at cardiac pathology domain is performed through data, where it can be acquired from distinct sources and shapes for instance: images; videos; time series and clinical data [5]. In this study, clinical data was acquired from a cardiovascular hospital, where the study performed by Ferreira et al. [6] show that feature selection can provide significant improvements to machine learning models in order to predict the pathology.

Feature Selection (FS) is a well known optimization problem used to discover the best subset of features in a dataset [7]. To solve this problem, statistical methods or ML algorithms are commonly used to gather information, regarding the dataset and feature's importance's, to further select the subsets. In a counterpart, some of these methods are not capable to find relevant data information or require thresholds limitations to the number of selected featured. Since feature selection is applied to capture the best subset of features, it is not straightforward to determine how many features will be allocated to it. Therefore, optimization algorithms are usually applied, where Swarm Intelligence (SI) algorithms emerges a viable strategies to overcome feature selection strategies issues [8].

Swarm Intelligence is a field of Artificial Intelligence (AI) that employs bio-inspired algorithms to solve diversified problems [9]. Between SI algorithms, the Particle Swarm Optimization (PSO) is a classic algorithm presented as a good approach to solve problems in distinct research fields, including data mining [10]. Despite the great overall performance of PSO, other algorithms were devised to deal with bottlenecks found in the canonical algorithm. One popular approach is the Evolutionary Particle Swarm Optimization (EPSO) [11], which is used in many optimization competitions [12] and was used as basis for the Maximum Search Limitations - EPSO (MS-EPSO) [13].

MS-EPSO is an algorithm devised to enhance the initial stage of EPSO optimization process, results presented in [13] show that the algorithm present a significant convergence boost at the initial stage and can be competitive against other optimization techniques. In this work, the intersection of MS-EPSO and ML models is applied to solve the FS problem and predict cardiac pathology in children and teenagers. The experiment is performed through a data mining pipeline, extending the idea of Ferreira et al. [6], applying a more sophisticated algorithm to the feature selection task, verifying the capabilities of MS-EPSO to be a rival for FS strategies.

In the proposed pipeline, data pre-processing and feature engineering tasks are performed, leading to feature selection performed by MS-EPSO, whose main goal is to predict the cardiac disease in children and teenagers. Also, a benchmark experiment is applied to evaluated the performance of MS-EPSO. In both experiments, a comparison with 17 state-of-the-art SI, Evolutionary Algorithms (EA) and FS strategies to select the best subset of features. After feature selection evaluation, the best feature subset obtained in the cardiac pathology dataset is propagated to four popular ML models, where each were assessed with tuned hyperparameters, completing the pipeline and verifying the following:

- Feature engineering may be more relevant than maintaining multiple features in the dataset;
- Performance of SI, EA and standard techniques for Cardiac Pathology FS;
- Performance of MS-EPSO exclusive strategies when applied to a binary problem;
- Impact of feature selection to predict cardiac pathology;

This dissertation is organized as follows:

- **Chapter 2**: describes fundamental concepts of cardiac pathology, machine learning, feature selection and swarm intelligence that are going to be used in subsequent chapters;
- **Chapter 3**: presents the state of the art of SI in to solve feature selection and cardiac pathology; the attempts to solve cardiac pathology; and indicating where this work modifies the standards.
- **Chapter 4**: describes the characteristics of MS-EPSO, regarding its differences when compared to EPSO and PSO and the goal of inserting rules and normal distributed search for strategy selection in the optimization process;

- **Chapter 5**: points out the applied pipeline to predict cardiac pathology and also describes the experiment with benchmarks to evaluate general feature selection of MS-EPSO;
- **Chapter 6**: show the statistical results obtained during the experiment and denotes whether MS-EPSO had a significant performance;
- **Chapter 7**: outlines the conclusion of the work, suggesting when the application of the algorithm is significant and pointing out future works for MS-EPSO and cardiac pathology.

Chapter 2

Background

This chapter describes fundamental concepts that support the remainder of this dissertation. Section 2.1 details the concepts of Swarm Intelligence, its standard mechanisms and analogy behind PSO and EPSO. Section 2.2 details the concepts related with Cardiac Pathology and the importance of features to detect the problem. Section 2.3 describes the classic machine learning pipeline used for problem solving. Finally, Section 2.4 describes the theory behind the feature selection problem.

2.1 Swarm Intelligence

Swarm Intelligence (SI) is a branch of Artificial Intelligence (AI) which encompasses bio-inspired algorithms to solve diversified optimization problems [14]. In this category, each algorithm simulates the collective intelligence of species that can be found in real world, for instance, the bees foraging behavior, the movement of flocking birds, the ant colony system, etc. Prominent algorithms that are commonly used in the literature are: Artificial Bee Colony (ABC) [15]; Ant Colony Optimization (ACO) [16]; and Particle Swarm Optimization (PSO).

Besides the theoretical environment of SI algorithms, the No Free Lunch theorem [17] corroborates the fact that there is not an algorithm capable of solving all problems, being one of the reasons to focus in algorithms for a specific area. For instance: ACO algorithm is a well known strategy to solve the traveling salesman problem and job scheduling; ABC is an algorithm to be applied in continuous domain [15]; PSO presented good results on distinct optimization domains [18].

Regarding SI's applications, surveys [15, 18] have presented a huge variety of them in economics, engineering, computer science, etc. In the data mining area, it is shown by [10] that ACO and PSO are the most used algorithms, where the approaches may be the parameter tuning or training of a Machine Learning (ML) model or the feature selection to identify important features among others for a specific problem.

Along the years, many modifications of these algorithms were presented, creating a novel strategy for a specific problem or general purpose. Among these modifications, the Evolutionary Particle Swarm Optimization (EPSO) devised by Miranda and Fonseca [11] was used as basis in many competitions with discrete, continuous and mixed variables domain. Even with the differences between EPSO and PSO, there is a standard mechanism applied before the customized optimization process, where the basic operators are described in the next subsections.

2.1.1 Base Swarm Intelligence Algorithm

The standard mechanism is applied to each algorithm based on the swarm intelligence. The basic structure is represented by a 2-Dimensional matrix of shape Number of Solutions (NS) x Problem Dimension (D) and a vector of shape NP. Each row of the matrix is the candidate solution of a specific problem, which is evaluated by a function f subject to constraints g.

The main structure is generated in a common phase named Initialization phase, where the initial values of the matrix are generated according to 2.1,

$$x_{ij}^{t=0} = l_j + \phi * (u_j - l_j), \tag{2.1}$$

where x_{ij}^0 is the j-th problem variable of the i-th particle in the swarm, ϕ is a random number sampled from a uniform distribution and set into a range between [0, 1], u_j and l_j are respectively the upper and lower bounds of the problem for the j-th variable. With the initialized matrix, each solution will have its respective fitness value calculated (x_{if}), finishing the initialization phase that leads to the individual mechanism performed by the SI algorithm until a stopping criteria is reached. The stopping criteria is usually the number of Function Evaluations (FE's) which stands by the number of times a solution was evaluated by the objective function.

2.1.2 PSO

Introduced by Eberhart and Kennedy [19], the Particle Swarm Optimization (PSO) is an SI algorithm based on the movement of flocking birds, the algorithm refers to the candidate solutions and the population as, respectively, particles and swarm, where each particle is composed by the following attributes: velocity (v); position (x); a fitness (x_f) value associated to the position; a local best position ($\hat{x}$); and a local best fitness ($\hat{x}_f$). At each generation t, the position is updated according to the movement rule (2.2),

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{2.2}$$

where the i-th particle of the swarm is moved according to the new velocity given by (2.3)

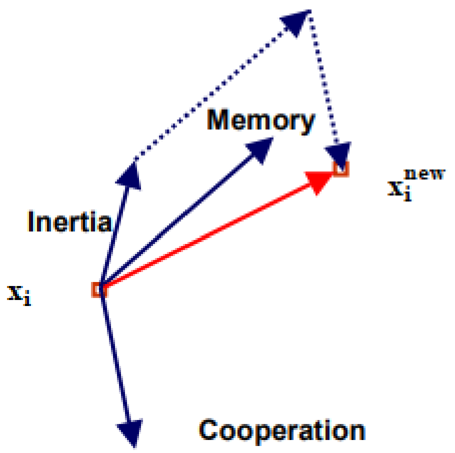


Figure 2.1: Depiction of PSO movement

$$v_i^{t+1} = w \times v_i^t + \phi_1 \times c_1 \times (\hat{x_i}^t - x_i^t) + \phi_2 \times c_2 \times (x_g^t - x_i^t), \tag{2.3}$$

where w, c_1 and c_2 are control parameters to adjust the particle movement of distinct components of the formula, while ϕ_1 and ϕ_2 are random numbers sampled from a uniform distribution between range $[0, 1]$. After moving a particle, the fitness value is calculated according to an objective function. If the new position x_i^{t+1} has a better fitness value when compared to x_i^t, the local best position is replaced and it is compared with the position of the global best particle in the swarm x_g.

The movement equation of PSO is divided into three terms: inertia; memory; and cooperation. An example of PSO movement is presented in Figure 2.1, where the inertia term is the capability of the particle to keep moving in the same direction, the memory is a movement towards the best trajectory ever visited by the particle, and lastly, the cooperation guides the particle to the best particle direction. The pseudocode of PSO is show in Algorithm 1.

2.1.3 EPSO

Along the years, PSO has been criticized regarding its optimization mechanism. The works developed by Miranda and Fonseca [11], and Zeng and Cui [20], show that, at each iteration, when the i-th particle is the current global best, the last term weighted by c_2 would be excluded, resulting in a movement performed only with the first and second terms. Also, when a particle replaces its local best at generation t, *t+1* generation would have its second term excluded, since the current position is the same as the local best position. Besides the criticisms related to the movement rule, there is a deep discussion about the correct values for each control parameter, requiring a parameter tuning for each distinct application. As an approach to circumvent issues found on classic PSO, Miranda and Fonseca [11] developed a novel algorithm named Evolutionary PSO (EPSO), which combines strategies found on the AI literature to enhance the algorithm

Algorithm 1: PSO pseudocode

```
Input: Objective function f(x), D, LB, UB, NP, NFE, $c_1$, $c_2$, $w$
Output: Best solution found $P_g = \{x_1, x_2, x_3, ..., x_D\}$
// After each function evaluation:
// 1) Increment FEs counter;
// 2) Check for a possible new global best;
// 3) Check the stopping criteria.
$FEs \leftarrow 0$
$MaxV \leftarrow abs(UB - LB)$
$MinV \leftarrow MaxV * -1$
// Initialization phase
for $i \leftarrow 0$ to to $NP$ do
    Initialize particle ($x_i$) between [LB, UB]
    Initialize velocity ($v_i$) between [MinV, MaxV]
    $x_{if} \leftarrow f(x_i)$
end
Save all local best information ($\hat{x}_i, \hat{x}_{if}, \hat{x}_{i\mu}, \hat{x}_{i\sigma}$)
Save global best information ($x_g, x_{gf}, x_{g\mu}, x_{g\sigma}$)
// Optimization process
repeat
    for $i \leftarrow 0$ to to $NP$ do
        $v_i \leftarrow$ MoveParticle($x_i$, $v_i$, $\hat{x}_i$, $x_g$, $c_1$, $c_2$, $w$)
        $x_i \leftarrow$ UpdatePosition($x_i$, $v_i$)
        $\hat{x}_i, \hat{x}_{if} \leftarrow$ Compare($x_i$, $\hat{x}_i$)
        $x_g, x_{gf} \leftarrow$ Compare($\hat{x}_i$, $x_g$)
    end
until $FEs == NFE$
```

optimization process.

The merge of evolution strategies with PSO introduces: 4 dynamic weights w_i for the particle movement; replicas; and reproduction components, into the basic algorithm. In EPSO, the dynamic weights replace the static weights, e.g. movement formula control parameters, found in PSO and also each particle has its own collection of weights, which differs from PSO that has weights for the whole population. Regarding the optimization process, EPSO starts by generating replicas, each replica will copy the particle information and their weights are mutated according to Equation (2.4),

$$w_{rk}^{t} = w_{ik}^{t} + \tau N(0,1), \tag{2.4}$$

where τ is the mutation rate and $N(0,1)$ is a sample drawn from a Gaussian distribution of mean 0 and standard deviation and w_{ik}^{t} is the k-th weight of the i-th particle. With this new collection of weights, each of the *NR* replicas and the original particle are moved with the new movement formula (2.5),

$$v_i^{t+1} = w_{i1}^{t} \times v_i^{t} + w_{i2}^{t} \times (\hat{x}_i^{t} - x_i^{t}) + w_{i3}^{t} \times P[(x_{g^*}^{t} - x_i^{t})], \tag{2.5}$$

where P stands for a communication factor, e.g. a binary mask filled with ones with probability cp. Furthermore, Equation (2.5) introduces a perturbation in the global best as x_{g^*}:

$$x_{g^*} = x_g \times (1 + w_{i4} \times N(0, 1)). \tag{2.6}$$

In the end, a tournament between all replicas and the original particle is performed. The winner is assigned as the new i-th particle and, in case of the winner is a replica, the weights are also replaced. Finally, the current particle is compared with the local and global best particles as it happens in PSO. EPSO pseudocode is presented in Algorithm 2.

Algorithm 2: EPSO pseudocode

```
Input:  Objective function f(x), D, LB, UB, NP, NFE, $\tau$, CP, NR, MLL
Output: Best solution found $P_g = \{x_1, x_2, x_3, ..., x_D\}$
// After each function evaluation:
// 1) Increment FEs counter;
// 2) Check for a possible new global best;
// 3) Check the stopping criteria.
$FEs \leftarrow 0$
$MaxV \leftarrow abs(UB - LB)$
$MinV \leftarrow MaxV * -1$
// Initialization phase
for $i \leftarrow 0$ to to NP do
    Initialize particle ($x_i$) between [LB, UB]
    Initialize velocity ($v_i$) between [MinV, MaxV]
    Initialize strategic weights ($w^*_{i1}$, $w^*_{i2}$, $w^*_{i3}$, $w^*_{i4}$) between [0, 1]
    $x_{if} \leftarrow f(x_i)$
end
Save all local best information $(\hat{x}_i, \hat{x}_{if}, \hat{x}_{i\mu}, \hat{x}_{i\sigma})$
Save global best information $(x_g, x_{gf}, x_{g\mu}, x_{g\sigma})$
// Optimization process
repeat
    for $i \leftarrow 0$ to to NP do
        $best_{replica} \leftarrow$ GenerateReplicas$(NR, w^*_i, \tau)$
        $best_{replicaf} \leftarrow f(best_{replica})$
        $x_{new} \leftarrow$ MoveParticle$(x_i, v_i, \hat{x}_i, x_g, CP)$
        $x_{newf} \leftarrow f(x_{new})$
        $x_i, x_{if}, v_i, w^*_i \leftarrow Compare(x_{new}, best_{replica})$
        $\hat{x}_i, \hat{x}_{if} \leftarrow$ Compare($x_i$, $\hat{x}_i$)
        $x_g, x_{gf} \leftarrow$ Compare($\hat{x}_i$, $x_g$)
    end
until $FEs == NFE$
```

2.2 Cardiac Pathology

Cardiac Pathology (CP) is a general term for heart or blood vessels diseases, which are a major cause of morbidity and mortality [21]. According to [22], CP is divided in 4 branches:

- **Congenital**: acquired in the birth, is a defect on the heart or great vessel structure;
- **Ischemic**: reduction of the blood flow in the coronary arteries;
- **Valvular**: damage or defect in one of the four heart valves;
- **Myocardium**: occurs when blood flow decreases or stops to a part of the heart, causing damage to the heart muscle.

The same author indicates that pericardial diseases and cardiac tumors can be included as heart diseases, but representing a small subset of conditions affecting the heart. As a preliminary warn to a heart disease, common found symptoms are: dyspnea; fatigue; breathlessness; palpation; syncope; and edema [22, 23].

In clinical evaluation approaches, there are two important variables that indicate a possible pathology [22, 24]:

- **Heart sound**: divided in two categories, the S_1 sound is caused by closing of the mitral and tricuspid valves, and the S_2 sound is caused by closing of the aortic and pulmonary valves;
- **Heart murmur**: commonly present in children but often the first sign of cardiac pathology, the challenge is to detect where the murmur is normal or abnormal. Murmur is present when the blood is whirling as it passes through the heart.

Besides them, there is other information that may be used to validate the diagnosis [6]:

- **Heart rate**: heartbeat measured by the number of contractions (beats) of the heart per minute (bpm);
- **Blood pressure**: is the pressure of circulating blood on the walls of blood vessels. It is measured by systolic and diastolic blood pressures, which are parts of the cardiac cycle. Respectively, it is a measure when some chambers of the heart muscle contract after refilling with blood and the pressure in the arteries when the heart rests between beats;
- **Pulse rate**: allows a professional to accurately measure the heart rate.

For a better analysis of each component described, concerning the hospital process, doctors are required to perform a diagnosis. In the primary care, a doctor or nurse may collect data about the patient and later indicate his/her health state that would indicate preliminary symptoms of a cardiac pathology, being the major reason to develop a system to assist the primary care.

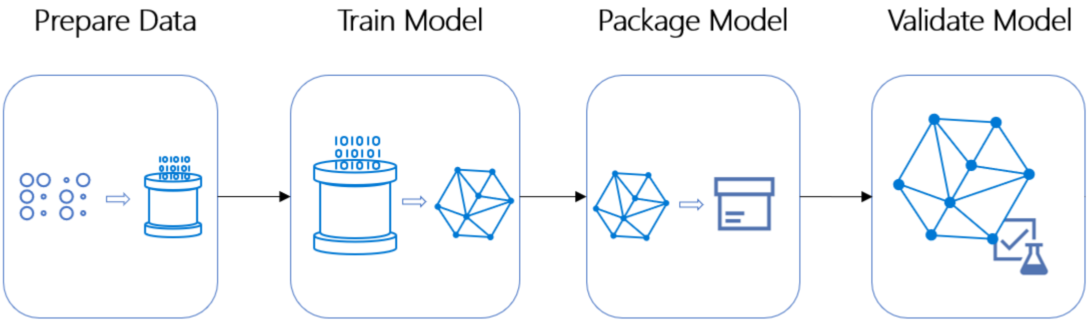


Figure 2.2: Machine Learning pipeline

Source: Adapted from [25]

2.3 Machine Learning

Machine Learning (ML) is a sub field of Artificial Intelligence (AI) which explores the capability of a computer to adapt to new circumstances, detect patterns and predict new data [26]. These capabilities are usually performed by a model, which is one or more algorithms combined, after its construction during a four-stage ML pipeline, presented in Figure 2.2. Before moving towards the pipeline, the problem should be established, where, according to [26], can be divided into:

- **Supervised learning problem**: both the inputs and outputs of a component can be perceived;
- **Unsupervised learning problem**: when there is no hint at all about the correct outputs;
- **Reinforcement learning problem**: an agent receives an evaluation of its actions (e.g. a robot cleaning the correct spot in a living room).

and also can be a:

- **Classification problem**: task of approximating a mapping function from input variables to **discrete** output variables;
- **Regression problem**: task of approximating a mapping function from input variables to **continuous** output variables.

Since the goal of this dissertation is to predict cardiac pathology and the output variable is binary, the subsequent theory of ML will be focused on a supervised learning and classification problem.

2.3.1 Input/Output

Initially, it is assumed that a dataset D is acquired and preprocessed, resulting in a 2D dataset with N instances (rows) and M features (columns). The same dataset is divided according to the objective of the problem, for example: Table 2.1 represents an artificial dataset where X_1 and X_2 represent the grades of each student and y means if the student was approved (A) or not (R). In this example, there are 2 features and 6 instances, where all features are continuous and the target variable (e.g. output) is categorical. It is worth mentioning that the *ID* column does not represent useful information for the model, being excluded from the inputs [27].

ID	X_1	X_2	y
1	7.0	7.2	A
2	9.0	2.5	R
3	8.0	8.2	A
4	4.5	4.5	R
5	6.5	7.5	A
6	9.5	7.5	A

Table 2.1: Artificial student grades dataset

At the current state of the dataset presented in 2.1, the X variables are ready to be used, which differ from the y that are represented as letters. ML models are usually prepared for learning numerical variables, requiring an encoding process to prepare the data for the subsequent phases. The encoding is performed according to the classification task that can be divided in [28, 29]:

- **Binary**: an output variable represented by two classes as shown in Table 2.1;
- **Multiclass**: when a variable has more than two classes, e.g. a monkey and an elephant are represented as an animal and not individually;
- **Multilabel**: assigns to each sample a set of target labels;
- **Multioutput-multiclass**: a single model has to handle several joint classification tasks.

In this work, multiclass and binary classes are used for, respectively, the cardiac pathology dataset and some of the selected benchmarks. For binary data, the standard encoding method was used, where a number is assigned to an specific class, e.g. for Table 2.1, A and R would be respectively 0 and 1. To handle the multiclass, depending of the classifier that was assigned to solve the problem, the One vs All (OvALL) strategy or the one-hot-encoding (OHE) should be used. OvALL strategy will be detailed in subsection 2.3.4.5, while OHE consists in creating a binary vector with size equal the number of distinct classes, and assigning the index of the standard label encode to this position as can be see in Table 2.2. With the processed dataset, it is possible to advance in the pipeline.

y	y_{ohe1}	y_{ohe2}	y_{ohe3}
A	1	0	0
B	0	1	0
C	0	0	1
B	0	1	0
A	1	0	0
C	0	0	1

Table 2.2: One-hot-encoding for a 3 classes (A, B, C) dataset

2.3.2 Validation

The training and testing phases on ML are, respectively, where the model acquires knowledge from the preprocessed data and validates it [26, 27]. These two phases represent the last three stages of the pipeline presented in Figure 2.2. As mentioned in the previous section, a dataset is required to fit the learning model. With the acquisition of this model and the dataset, the process can be summarized as presented in Figure 2.3.

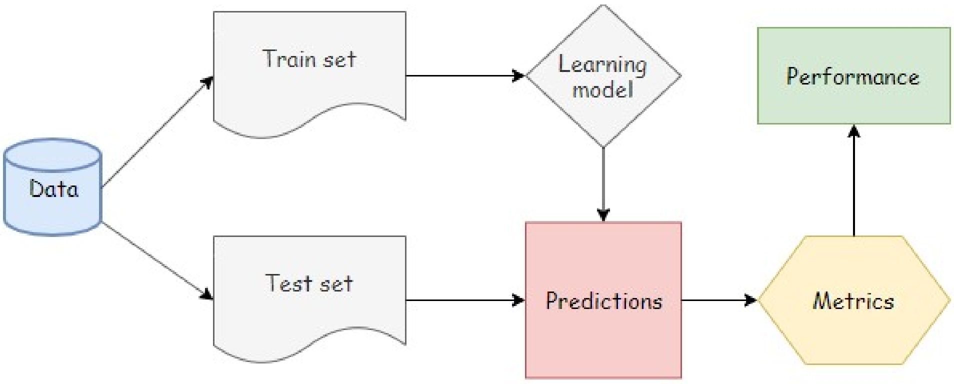


Figure 2.3: Summary of the train and test phases

2.3.2.1 Train

Initially, the dataset is divided in four sets: X_{train}; X_{test}; y_{train}; y_{test}, both train splits are used in the train phase, while the test remains untouched until the end of the training. The model will receive the X_{train} and apply its own learning mechanism, generating a $\hat{y}$ dataset that is compared with the y_{train}. Finally, the comparison is evaluated through a metric that will estimate its performance on this set of data.

Besides this simple evaluation, there are other strategies to estimate the risk of a learner or to perform model selection, for example: the cross-validation (CV) [30]. In CV, the most popular strategy is the k-Fold cross-validation (k-CV) that applies the following steps [31]:

1. Shuffle the dataset;
2. Split the data in k folds of the same length (same number of instances);
3. Create and evaluate a model for each fold;
4. Calculate the cross validation score.

When handling unbalanced datasets, the Stratified k-Fold Cross Validation (Sk-CV) is used to rearrange the data, ensuring that each fold is a good representative of the whole [31]. As mentioned in [32] for both SK-CV or K-CV, the parameter k is not straightforward, being usually 5 or 10. As suggested in [31], when k increases, the variance reduces but it increases the bias, the opposite happens when the k-CV has a low k value.

Finally, the CV score can be calculated using the mean of each fold or the summation of TP, TN, FN, TN. According to Forman and Scholz [33], when performing cross-validation, the best way of obtaining scores is by summing up correct and incorrect classifications per class and calculating the final score. This process is used to estimate the predictive capabilities of a ML model in unseen data and the general model's performance [26]. When the user is satisfied with the score achieved in the k-CV, the learning model is maintained to be used in the test phase.

2.3.2.2 Test

The testing phase represents the validation stage of the ML pipeline. As mentioned in the previous section, the original data is divided into 4 subsets and, in this stage, the test data that was not used on the training (unseen data) is used. When evaluating the testing score, it means that the model predictive capabilities are being analyzed. Similar to the the training phase, each model has its own procedure to give these predictions, returning a $\hat{y}$ dataset that is compared with y_{test} and evaluated by a metric.

In addition to the testing analysis, the fit should be analyzed. As stated in [34], fit refers to how well the function is being approximate. This concept is applied to machine learning in the following characteristics:

- **Overfitting**: the model acquires knowledge for a specific set of data and may not be able to generalize;
- **Underfitting**: the model is not able to generalize the training data and have a poor predictive performance;
- **Good fit**: a mid term between over and underfitting.

To circumvent these issues, the cross-validation strategy mentioned in this chapter can be used. Also, there are other techniques applied during the process of the algorithms that allow to perform a good fit, which will be detailed when discussing the algorithms in the subsequent sections.

2.3.3 Evaluation Metrics

This section discusses the Accuracy, F-Score and Receiver Operating Characteristic (ROC) metrics for supervised learning classification problems since they are commonly used in the literature. According to Sokolova et al. [35], binary classification is usually represented by a confusion matrix represented in Table 2.3, where: *TP* are True Positive; *FP* - False Positive; *TN* - True Negative; *FN* - False Negative. Accounting the results obtained by the model, each metric will calculate the score using the confusion matrix, but using a different mathematical model. It is worth mentioning that these metrics share the same goal, a maximization problem that has an optimal value in 1.0 and the worst value at 0.0.

Class x **Prediction**	**Positive**	**Negative**
Positive	TP	FN
Negative	FP	TN

Table 2.3: Confusion matrix for binary classification

2.3.3.1 Accuracy

Being the most popular metric used in the literature [35, 36], accuracy is defined by Equation 2.7,

$$accuracy = \frac{TP + TN}{TP + FP + FN + FP}. \tag{2.7}$$

Regarding the mathematical model, accuracy has been criticized since it may lead to a biased score. An example described by [37], it assumes an unbalanced dataset where, for each positive class, there are 100 negative classes. The same author indicates that a model with a good fit would produce an 99% accuracy, classifying correctly each negative class since its majority, while misclassifying all positive cases.

2.3.3.2 Receiver Operating Characteristic

ROC is described as a comprehensive function to evaluate the performance of a model [35]. The mathematical model can be described by Equation 2.8:

$$ROC = \frac{P(x|positive)}{P(x|negative)}, \tag{2.8}$$

where $P(x|C)$ denotes the conditional probability that a data entry has the class label C. As mentioned by [35], ROC results deals with the most positive to the most negative classification, being a metric that can be easily analyzed and one of the favorite ones to deal with unbalanced datasets. Regarding an unbalanced datasets, it is has been proven that metrics based on precision and recall may return a more realistic result [38].

2.3.3.3 F-score

F-score is a precision-recall based metric, therefore, it is required to understand the mathematical model of both metrics to understand the main goal. Initially, recall or sensitivity, is calculated by 2.9:

$$recall = \frac{TP}{TP + FN}, \tag{2.9}$$

while precision is given by 2.10:

$$precision = \frac{TP}{TP + FP}, \tag{2.10}$$

where both metrics are used to calculate the F-score, presented in 2.11:

$$F\text{-}score = 2 * \frac{recall * precision}{precision + recall}. \tag{2.11}$$

Recall may be defined as the model capability to predict positive results and precision is the proportion of positive results that are truly positive. The f-score would be the harmonic mean between these metrics and also can be used to achieve more realistic results for unbalanced datasets.

2.3.4 Algorithms

This subsection describes popular machine learning algorithms that are commonly used in distinct knowledge areas. The group of algorithms that will be mentioned here had great results along the years, classifying data that vary between raw, images, audio, videos, etc [26, 27].

2.3.4.1 Logistic Regression

The Logistic Regression (LR) is a linear statistical model that uses a logistic or sigmoid function to build and optimize a problem that has binary dependent data. This technique can also be extended to classify more than two values using distinct strategies [39]. Regarding its standard model, it can be seen as the probabilities of having the labeled data as 1 or 0.

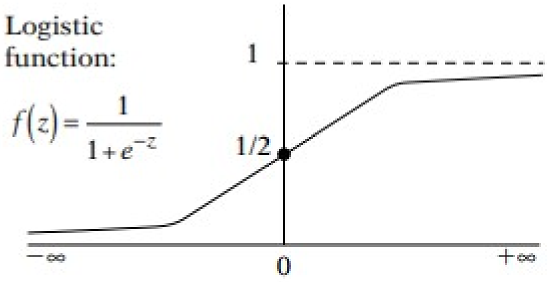


Figure 2.4: Logistic/Sigmoid function

Source: [40]

The model is based on the logistic function represented by Equation 2.12,

$$f(z) = \frac{1}{1+e^{-z}}, \tag{2.12}$$

which describes the input probability of being positive (1) or negative (0). Since that the probability between infinite numbers are verified, the function output has a range between 0 and 1 as it is possible to visualize in Figure 2.4. The logistic model z is represented in Equation 2.13,

$$z = \alpha + \sum B_i X_i, \tag{2.13}$$

where B_i are the learning parameters of the model, X_i are the dataset inputs and α a constant to weight the summation. The final cost function of the LR model is given by Equation 2.14 [41]:

$$P(X) = \frac{1}{1+e^{-(\alpha+\sum B_i X_i)}}. \tag{2.14}$$

In order to minimize the cost function, the LR model tries to optimize the learning parameters (B_i). Since the sigmoid derivative function can be easily calculated, these parameters are usually optimized using derivative methods, instead of general purpose optimization algorithms. With the best collection of learning parameters, the model is evaluated through the ML metric system presented in the previous section.

2.3.4.2 K-Nearest Neighbors

The Nearest Neighbors algorithm is a distance based algorithm used to cluster different instances in one same group [42]. What differs nearest neighbors from other classifiers is that the model does not requires the data output (y) to be labeled [43]. As an extension for that model, the K-Nearest Neighbors (KNN) can be used when the data is labeled, being controlled by the parameter k, which determines the number of neighbors to be considered during the model optimization stage.

The KNN algorithm starts with a data point with unknown classification. The following steps are applied for each instance found in the dataset: 1) the distance will be calculated between the new and known points and, if they are near, this known data point will be added as a nearest neighbor; 2) As next step, the algorithm will count the most frequent class among K neighbors examples based on the amount of nearest neighbors; 3) the frequency is used to label the new data point. This mechanism is applied to give the predictions that are further evaluated by the ML metric system [43].

Regarding the distance system, the Minkowski distance (Equation 2.15) can be used,

$$d(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{2.15}$$

where x and y are two instances of data. This metric is a generalized form for the following distances[44]:

- **Euclidean**: when p is equal to 2, it will calculate the shortest path between two points if they are continuously linear;
- **Manhattan**: when p is equal to 1, this metric assumes that the data has no linear dependence. When dealing with categorical variables, the distance can be analyzed as the amount of steps required to reach the other point.

Besides these metrics that are commonly used in the KNN algorithm, any metric can be used in order to calculate the distance when searching for nearest neighbors, where the major dependence is selecting the best metric for a specific type of data.

2.3.4.3 Neural Networks

Neural Networks or Artificial Neural Network (ANN) is a non-linear classifier that tries to mimic the behavior of neurons in a human brain. Regarding the math behind the model, the training phase of an ANN is based on the propagation of the data through its architecture [26]. The architecture shown in Figure 2.5, is the standard format, where it is represented by a group of layers where each may have distinct or equal number of neurons.

In this type of knowledge representation, the number of neurons in the input layer is the same of the number of inputs in the dataset, for example, in the artificial dataset 2.1, it would have 2 neurons. When propagating the data to the next layer, the neuron mathematical model can be described as:

$$y_i = f(\sum_{j=1}^{n} w_j x_j + b), \tag{2.16}$$

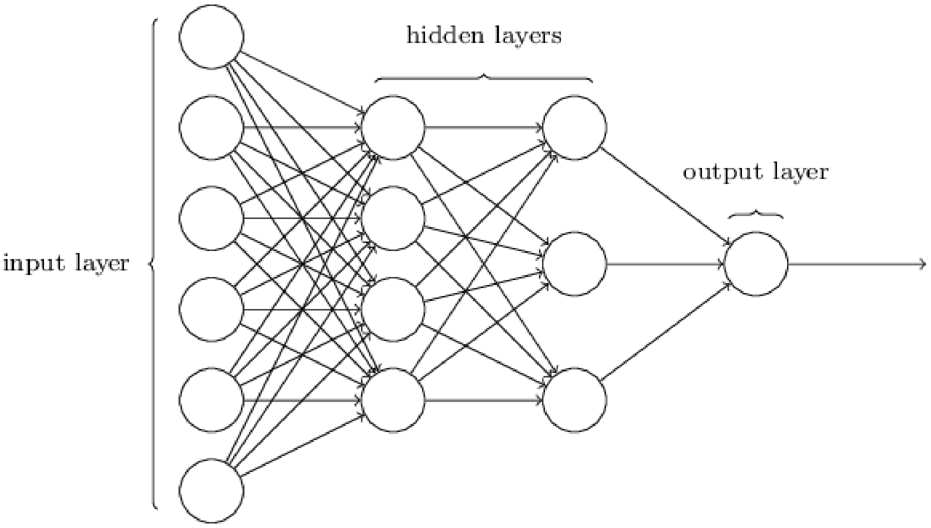


Figure 2.5: ANN architecture

Source: [45]

where w_j is the corresponding weight that connects the previous to the next layer, x_j is the propagated data from previous layer and f is a non-linear function, usually the sigmoid 2.12 for machine learning approaches, and b is a bias that can also be a learning parameter during the model optimization [45].

After propagating the data, the network output ($\hat{y}$) is compared with the original output (y) and the model training is evaluated through an error metric [45]. For classification problems with both multiclass or binary, this function is usually the Log loss/Categorical Cross-Entropy given by:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i * log_e(\hat{y}_i) + (1 - y_i) * log_e(1 - \hat{y}_i)], \tag{2.17}$$

where n stands for the number of predictions given for each data instance. The objective is to minimize the network error by updating its weights and, like the LR model, the derivative of these metrics can be easily computed, allowing this training phase to be performed by gradient based algorithms. Finally, when the weights are optimized and the training is finished, to compare the ANN model with other ML algorithms, the ML metric system is used in order to verify which algorithm had the best overall performance.

2.3.4.4 Random Forest

Random Forest (RF) is a term for ensemble methods composed by tree-type classifiers [46]. Each tree classifier is fitted on various sub-samples of the dataset to improve the predictive score and reduce the overfit [47].

According to [46, 47], The training phase of a RF model train multiple trees with distinct samples obtained from the original data, acquiring knowledge of a randomly selected subset of features and instances that will determine this data split. The number of features used on the training session is the control parameter defined by the user, where the model will randomly select the subset of features. When applying this model for a classification problem, the model output is given by the majority vote of each tree, where later this outputs are compared with the original outputs, and its results analyzed through ML metrics.

Regarding the model parameters, with a low number of variables is expected that the correlation between trees and the required processing time are minimized and also the following advantages: with a limited number of features, it would increase the predictive power; creates a model less sensitive when handling outliers; few hyperparameters; feature importance and accuracy are automatically generated [48]. The other main parameter is the tree-type classifier, where the commonly used is the Decision Tree.

Decision Tree (DT) is a supervised classification model represented by a upside down real-world tree [49]. In Figure 2.6, the DT basic structure is shown, where according to [48], it starts from the root (root node) and move downwards until a terminal leaf node is reached. The internal nodes are representations of a specific characteristics while the branches are a range of values for them [48].

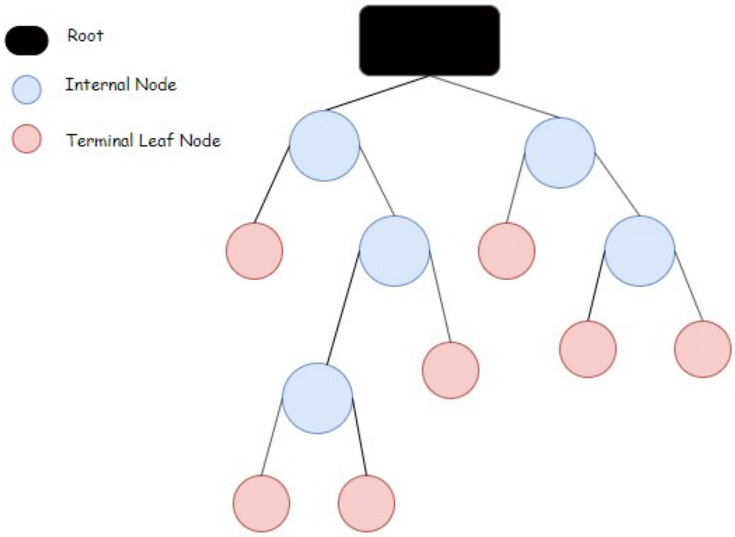


Figure 2.6: Decision Tree structure

Source: Adapted from [48]

The training phase propagates the data from the root until it reaches one terminal left node, passing through internal nodes which will try to optimize the best split for each feature. The best split would increase the model predictive capabilities of searching for the correct classification when going downwards on the tree architecture [48]. With the collection of DT models, it is possible to build the RF model and apply its process for problem solving. Also, like other ML algorithms, in the end of this training task, it will be evaluated through the ML metric system to be compared against other algorithms.

2.3.4.5 One vs All

In Table 2.4, each classifier used in this work is summarized regarding its capabilities of handling multiclass problems and how they are capable to predict/classify data. If one of the algorithms does not support multiclass problems, the One vs All (OvALL) strategy was used in order to extend the model for this task.

Model	Multiclass support	Learning mechanism
Logistic Regression	N	Optimizing the learning parameters
K-Nearest Neighbors	Y	Similarity between instances
Artificial Neural Network	Y	Optimizing the weights and/or bias
Random Forest	Y	Multiple instances of a tree-type classifier

Table 2.4: Summary of the ML methods used in this work

The OvALL can be described as the application of many classifiers of the same type, where each would try to predict the probability of being one specific class. Using the LR model as basis and a dataset with 10 different classes, Figure 2.7 describe the process to perform the training and perform the predictions $\hat{y}$. This process will perform the training steps of the model for each class and then the one with highest probability of being this specific class would be selected as output [50].

2.4 Feature Selection

Feature Selection (FS) is an optimization problem that emerges from a dataset, where the objective is to select the best subset of features that can be found on it. Between the benefits of FS, it is possible to cite:

- Faster training phase - less features would require less computational and also a machine learning algorithm with less complexity;
- Reduces overfitting - redundant features are excluded from the entire set;
- Improve the score - less misleading data would increase the obtained score by the algorithm.

while the cons are: FS does not guarantee that the score will be improved, therefore, if the algorithm requires a considerable time to perform FS and it fails to increase the ML algorithm performance, the FS step would require more time when compared to applying the algorithm to the entire dataset, leading to an unnecessary process.

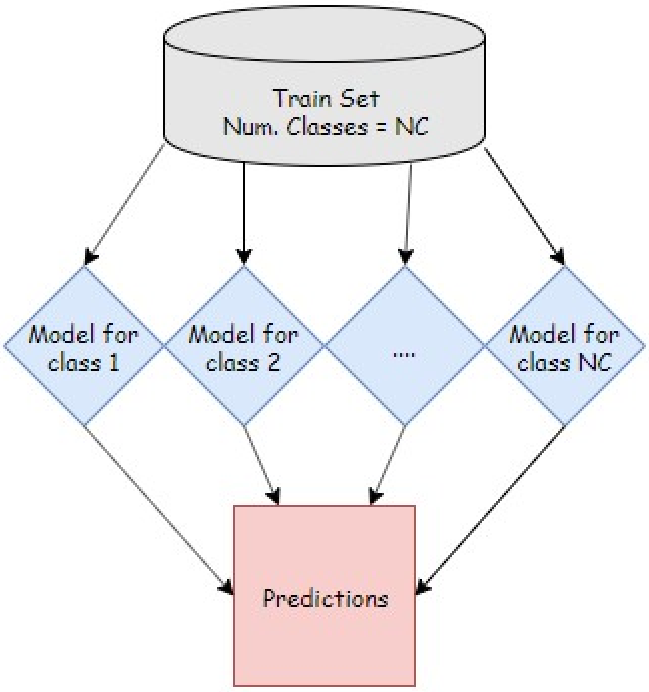


Figure 2.7: OvALL strategy for multiclass classification

Regarding the optimization task, FS is a bi-objective problem which requires a solution that would increase the model score at the same time it decreases the number of features. Prior the FS evaluation function, it is necessary to declare the solution, which is given by a binary array with size equal to the number of features in the dataset, where the i-th variable on it represents whether or not the i-th feature of the dataset should be used in the subset. FS evaluation function allows the comparison between distinct techniques to solve the problem. As suggested by Al-Tashi et al. [51], the objective function needs to penalize the number of features and the score, leading to the objective function in Equation 2.18

$$f(x) = \alpha Score(X_{train}, y_{train}) + \beta \frac{|S|}{|T|}, \tag{2.18}$$

where: the score stands for an ML metric acquired from the training stage of an ML model with the X input with selected features and y the target variable; S stands for the selected subset of features; T is the full set of features; α is a real number between [0, 1] range and β is given by (1 - α). The goal for Equation 2.18 is to penalize the learning algorithm score with the number of selected features, minimizing the score while minimizing features, where both score and features are weighted by two distinct parameters. However, in order to apply the same strategy for classification problems, a slightly modification should be performed, modifying the plus sign to minus, resulting in Equation 2.19:

$$f(x) = \alpha Score(X_{train}, y_{train}) - \beta \frac{|S|}{|T|}. \tag{2.19}$$

The modification would maximize the score while it still minimizes the number of features, allowing the algorithms to be compared regarding the obtained score applying the classification metrics as basis. With the definitions of a solution and the cost function to evaluate it, according to Guyon and Elisseeff [52], this problem is usually solved by three distinct approaches that will be explained on subsequent subsections: Filter; Wrapper and Embedded.

2.4.1 Filter

Filter strategies are commonly applied as part of the preprocessing phase where the main objective is to identify important features before creating any model that would require a higher computational time. According to Chandrashekar and Sahin [7], these methods are capable to create a ordered feature rank that would indicate the most prominent features in the dataset. Popular strategies of filter FS are the correlation criteria and mutual information.

2.4.1.1 Correlation Criteria

The correlation detects linear dependencies between the features and target variable. The simple correlation model is the Pearson correlation coefficient which is defined in Equation 2.20,

$$R(X_i) = \frac{cov(X_i, y)}{\sqrt{var(x_i) * var(y)}} \tag{2.20}$$

where cov is the covariance and var is the variance. Since correlation can achieve values between range [-1, 1], the absolute value of the result is used to select the most relevant features for the subset. Also, a percentage of features should be set in order to indicate how many features should be included.

2.4.1.2 Mutual Information

The Mutual Information (MI) measure the dependency between two variables. As defined in [7], MI uses the Shannon entropy as basis represented here by Equation 2.21,

$$H(y) = -\sum_{x} p(y)log(p(y)), \tag{2.21}$$

which stands for the information content in the target variable. When observing the X variable in relation to the target, the conditional entropy is defined as:

$$H(y|X) = -\sum_{x}\sum_{y} p(X, y)log(p(y|x)). \tag{2.22}$$

Equation 2.22 implies that by observing a variable X, the uncertainty in the output Y is reduced [7]. The decrease in uncertainty is given as:

$$MI(y, X) = H(y) - H(y|X). \tag{2.23}$$

Equation 2.23 gives the mutual information between X variable and target. If these two variables are independent, the MI score will be zero, otherwise the result will be greater than zero. Since the results range from zero to ∞, the result can be normalized in order to select the features with highest importance according to MI.

2.4.2 Wrapper and Embedded

These strategies were grouped due the similarity between them. Wrapper presents feature selection based on a learning algorithm and a recursive approach, while Embedded tries to minimize the complexity between them.

2.4.2.1 Wrapper

Wrapper is a strategy that uses the base predictor to create the best subset of features [7]. The strategy is given by reapplying the learning algorithm, where at each iteration, the less significant subset of features is removed. Applying a ML model with this kind of strategy requires the model to be fit at each iteration. With the fitted model, the feature importance is acquired from the learning mechanism (weights, bias, etc.) and a new subset of features is generated according to the feature exclude percentage parameter. The new subset is used to fit the model and this process is reapplied until a stopping criteria is reached, where this parameter is usually the number of iterations, therefore, the control parameters for this type of strategies are: 1) the learning algorithm parameters; 2) number of iterations; 3) percentage of features to be excluded.

The pro for this strategy is related to the number of distinct subsets that can be acquired during the process. The balance between the number of iterations and percentage of features can generate many distinct subsets that can search for the best subset for the specific learning algorithm. In a counterpart, when the problem have a high number of features, the model will have exponential performance, since the model is fitted at each iteration with the new generated subset. Also, the strategy parameters are problem dependent, consequently, having no rule to determine them. Since these parameters may lead to the best possible solution, a parameter tuning may circumvent that issue, but it increases even more the computational time.

2.4.2.2 Embedded

With the purpose to reduce the complexity time found in Wrapper strategy, the Embedded strategy try to insert the feature selection in the training process of the algorithm [7]. This is done by selecting an objective function that will evaluate the process, e.g. will rank the selected features in order to increase the quality of the predictions. This strategy is usually applied with the feature importance acquired by the learning algorithm, as an example: the information acquired in the decision tree or optimized weights of a neural network or logistic regression. As another approach to perform this feature selection, it is also possible to use the mutual information (Eq. 2.23) or correlation (Eq. 2.20) functions to discriminate the best feature subset.

2.4.3 Swarm Intelligence

Swarm intelligence algorithms are applied as a slightly modification of wrapper methods, which instead of removing recursively, will insert or remove features according to the algorithm rules. Applying an SI algorithm to FS can usually enhance the results when compared to other strategies, since its procedure will evaluate distinct solutions along the optimization process. Applying SI algorithms to FS problem requires two distinct components to be defined: the type of solution and the fitness function.

2.4.3.1 Solution

Since the FS problem requires an indication of whether or not the features should be included on the feature subset, the solutions are represented as a binary vector with size equal to the number of features in the dataset. If the j-th variable of the particle position has value 1, it indicates that the j-th feature on the dataset should be included in the feature subset S, while 0 values indicate the features associated with those positions should not be included.

2.4.3.2 Fitness Function

Along the years many approaches were described in order to apply any metaheuristic to the FS problem. Concerning the behavior of the FS problem, it is expected that the problem formulation would handle the capabilities of the algorithms to search for the best feature subset that would increase the classifier score while decreasing the number of features. The fitness function can be multi or single objective according to the capabilities of the selected algorithm. In case of MS-EPSO and other SI algorithms applied in this work, the single objective cost function, shown in Equation 2.24, is applied:

$$f(x) = \alpha \times Score(X_{train}, y_{train}) - \beta \frac{|S|}{|T|}, \tag{2.24}$$

where: the score stands for an ML metric acquired from the training stage of an ML model with the X input with selected features and y the target variable; S stands for the selected subset of features; T is the full set of features; α is a real number between [0, 1] and β is given by (1 - α). The goal for Equation 2.24 is to penalize the learning algorithm score with the number of selected features, maximizing the score while minimizing features, where both score and features are weighted by two distinct parameters.

2.4.4 Filter vs Wrapper vs Embedded vs SI

As show in this section, FS is a problem that can be solved with different strategies that can be performed by many distinct algorithms. The benefits of applying each strategy may vary in time complexity and predictive accuracy, where each strategy can be the best depending on the application goal. This subsection summarize the comparison between strategies ranking them based on these two benefits.

2.4.4.1 Time Complexity

Regarding time complexity, it is expected that Filters strategies would have the best performance. Filters does not compress model fitting on it, requiring only one fit - selected subset - during the training phase, which differs from the others. Embedded strategy would be the second faster strategy, since it requires two fits: 1) model fitting with all features; 2) model fitting with the selected subset. Following this rank, Wrapper and SI share the position of the slower strategy, where it is dependent on the number of generations for the SI and the balance between number of features to remove and iterations for wrapper.

2.4.4.2 Predictive Accuracy

Each strategy has a vast group of algorithms that can be applied for the feature selection purpose, therefore, the predictive accuracy of each is data dependent [7]. According to Guyon and Elisseeff [52], in a complex scenario with a huge amount of features, Filter methods tend to have the worst performance, since correlations or data information is not always the main discriminating factor . Wrapper strategy may find the best possible subset of features, depending on the balance of the parameters used and the excluded subset of features. The performance of wrapper is justified by the model, where the strategy will exclude information that is not significant for the specific model. Meanwhile for Embedded strategy, as an alternative for wrapper, it is expected that Embedded performance would perform worse than wrapper in a complex scenario, therefore regarding the predictive accuracy, the performance would be data dependent (data statistics and number of features). Finally, for SI algorithms, the feature selection is based on the algorithm rules, many works in the literature [8, 10] suggests SI techniques as a powerful strategy for FS, however, SI algorithms does not have any guarantee to find the optimum, sharing the same rank with wrapper in the predictive accuracy.

Summarizing the presented characteristics, the application of feature selection is dependent on data. Large scale datasets would require more computational time in order to select features and may have lower accuracy when compared to small or medium size datasets. Also, each strategy may be appropriate concerning the application goal. Conflict or danger situations in real-world scenarios may require a fast learning strategy since the results are instantly required, however, for the cardiac pathology, a accurate model is desired in order to reduce the number of false negatives, which are the worst case scenario concerning the disease.

Chapter 3

State of the Art

This chapter presents the state of the art of distinct computer science areas applied during the development of this work. The chapter is divided in two sections, where the first section outlines the feature selection state of the art, mentioning algorithms and strategies that can be applied, creating a relation with what was presented in the previous chapter, while the second section outlines the role of Artificial Intelligence at Cardiac Pathology, pointing out other approaches to assist the medical area on handling cardiac pathology.

3.1 Feature Selection

Along the years multiple Feature Selection (FS) strategies were developed, each based on a distinct objective function described in 2. This section will focus on single objective with penalty for features, which can be handled by MS-EPSO and many other optimization algorithms.

Some of those FS strategies do not return a solution based on an ML model, instead, these strategies are applied during the preprocessing stage, which is the case of Filter strategies. In this category, statistical tests are applied to obtain a subset based on the test criteria. Pearson Correlation and Mutual Information (mentioned in Chapter 2) are the most used strategies to quickly identify data patterns[7]. These methods are applied to capture data information and can be evaluated using many objective functions, however they are not necessarily optimizing the problem solution.

Since Wrapper and Embedded strategies share the same inspiration with distinct procedures, the state of the art for both techniques is represented by ML models. The feature importance of many ML models may be acquired from their learning parameters, for example, the weights and bias of a neural network, coefficients of a logistic regression, etc. The importance acts as the principal factor since it is the relevant information that wants to be captured by these strategies, however, it is not straightforward to define ML state of the art, since applications are data dependent. In this work, clinical data is used, therefore, the group of ML models presented in the previous chapter are the ones constantly referenced as algorithms that may have a reasonable performance when applied to any problem, including the medical area as presented in [53–55].

Like ML models, the state of the art of swarm intelligence is relative to the application area. Also, as mentioned in 2.4, the feature selection problem can be applied with distinct objective functions, which changes the algorithm strategy to solve the problem, therefore, the SI state of the art mentioned here regards algorithms that can be applied to solve single objective cost function.

3.1.1 Particle Swarm Optimization and Variants

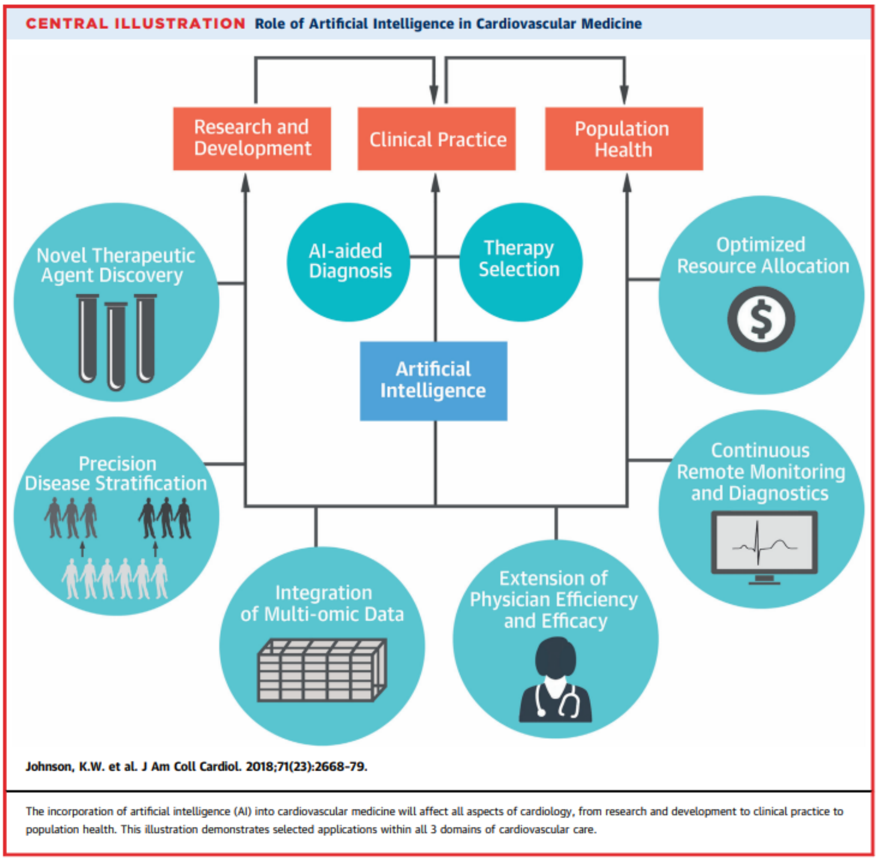
As presented in [7] and [8] an extensive list of SI algorithms were applied to the feature selection problem. Among these algorithms, it is possible to visualize that classic algorithms are presented as viable algorithms while their variations have the challenge to overcome issues. In Brezočnik et al. [8], more than 40 algorithms were analyzed, among them: Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), and Grey Wolf Optimization (GWO). However, more detailed analysis show that most used algorithms for feature selection are (in order): PSO, ACO, Bat Algorithm (BA), ABC and GWO. The research shows that PSO is used in almost 47% of all cases, while the four most frequently used algorithms cover more than three quarters of cases (79%).

Concerning PSO popularity for feature selection, the algorithm also had a huge number of enhancements or modifications, creating novel algorithms/modifications to handle general optimization problems. As PSO successful variants, its worth mentioning the Competitive Swarm Optimizer (CSO), which applies a mechanism based on tournament selection to select weights for the PSO movement equation. CSO also had attention for feature selection as shown in [56]. The algorithm, as a continuous optimizer, rounds the value to the nearest based on a parameter λ. Other PSO variation which received attention, not as recent as CSO, was Improved Binary Particle Swarm Optimization (IBPSO). The premise of IBPSO is to reset the global best particle, which will modify the last term of the PSO movement equation, therefore, this would remove solutions trapped in the local best. The algorithm was applied to a large scale dataset and enhanced PSO performance in 2.85% when selecting gene expression features.

Besides relevant PSO modifications for feature selection, other versions had outstanding performance on general optimization problems, especially, the EPSO algorithm mentioned in the previous chapter. The algorithm was capable to generate great solutions for optimization problems with both discrete and/or continuous variables, including being the base algorithm used by the top five algorithms on the World Congress of Computational Intelligence (WCCI) 2018 competitions. As an enhancement for EPSO, with modifications on the movement formula, the Differential Evolution EPSO (DEEPSO) [11] emerges with the insertion of two new memory components to store global and local best positions along iterations. DEEPSO is presented as a successful approach concerning EPSO applications, since the obtained results had statistical performance when compared to EPSO, however, Marcelino et al. [57] shows that EPSO has the potential to bypass local traps and generate better solutions when compared to DEEPSO, which is the algorithm created by the authors named CE+EPSO, an algorithm that applies the Cross Entropy method to initialize candidate solutions.

3.2 Cardiac Pathology and Artificial Intelligence

As shown in Johnson et al. [5], the computers are playing an important role in the medical area. Regarding cardiovascular medicine, Figure 3.1 points out AI role as part of medical workflow, where in most cases, AI strategies act as a decision making assistant for specialized employees of medical area. The applied pipeline on medical area is presented in Figure 3.2, where data can be acquired from different sources, leading to multiple feature engineering techniques that can be applied in order to prepare a clean dataset which is propagated to ML models. Among a group of models, the presented image corroborates information given in previous section, which points out favorite ML models and their pros and cons for cardiovascular applications, that were detailed in the previous chapter of this work.


CENTRAL ILLUSTRATION Role of Artificial Intelligence in Cardiovascular Medicine
Research and Development
Clinical Practice
Population Health
Novel Therapeutic Agent Discovery
AI-aided Diagnosis
Therapy Selection
Optimized Resource Allocation
Artificial Intelligence
Precision Disease Stratification
Continuous Remote Monitoring and Diagnostics
Integration of Multi-omic Data
Extension of Physician Efficiency and Efficacy

Johnson, K.W. et al. J Am Coll Cardiol. 2018;71(23):2668-79.

The incorporation of artificial intelligence (AI) into cardiovascular medicine will affect all aspects of cardiology, from research and development to clinical practice to population health. This illustration demonstrates selected applications within all 3 domains of cardiovascular care.

Figure 3.1: Role of artificial intelligence in cardiovascular medicine

Source: Johnson et al. [5]

With this overall pipeline review of machine learning for cardiovascular, to fit the dissertation purpose and create a comparison for both pipeline and algorithms perspective, this section will

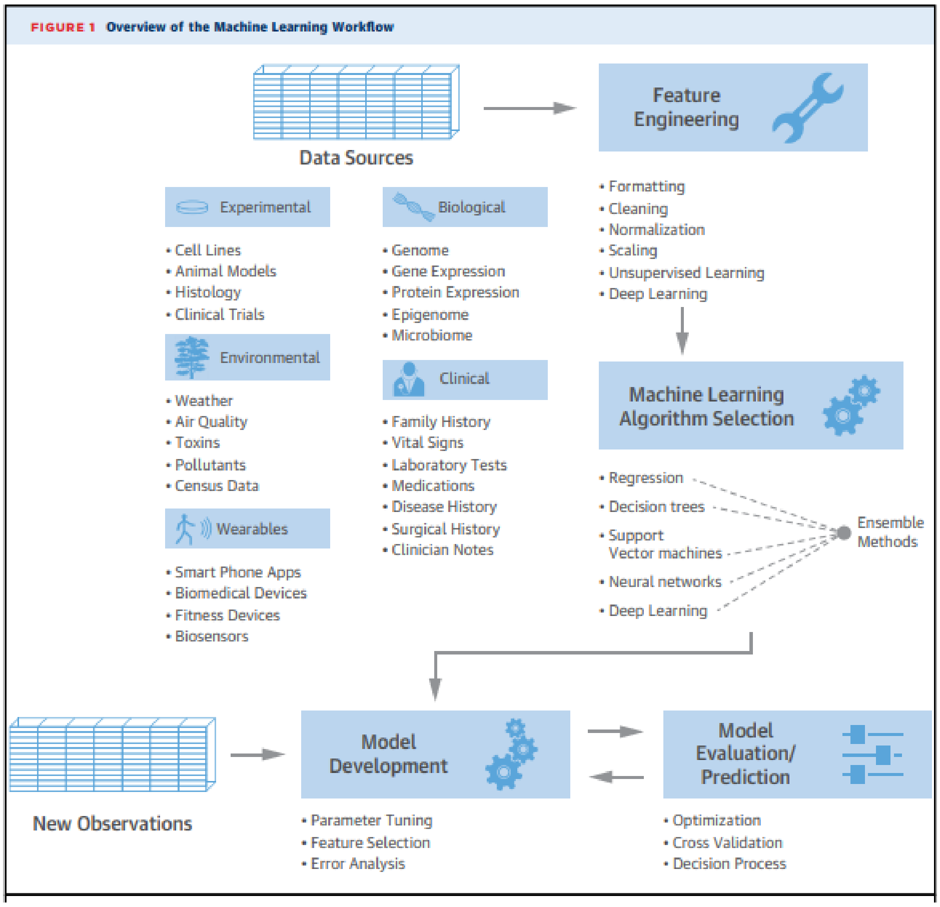


Figure 3.2: Machine learning pipeline for cardiovascular applications

Source: Johnson et al. [5]

cover most relevant computational attempts focusing in decision making assistants to handle cardiac pathology, highlighting swarm intelligence or feature selection applications in this medical domain.

3.2.1 Machine learning attempts to handle Cardiac Pathology

Cardiac pathology data sources are usually found as images, time series or clinical data, therefore, most of machine learning applications are developed based on the data domain available. As time series approaches, heart sound is commonly analyzed and classified by different techniques, being neural networks based approaches the most relevant ones as shown in [58, 59]. As well as time series, image recognition cardiovascular problems have been solved by neural networks approaches, especially by deep learning algorithms such as LSTM, CNN and auto encoders as pointed out in [59, 60]. At this stage, it is possible to visualize that neural networks have been widely applied to assist cardiac pathology, however, clinical approaches may have a higher diversity regarding strategies to circumvent the problem.

Moving forward to clinical approaches, which is related to the application developed in this work, steps from Figure 3.2 are mentioned in [61], where the general pipeline is once more stated as a viable choice for cardiovascular applications. According to Shameer et al. [61], Weng et al. [62], ML models are capable to improve predictive quality while increasing the counter of true positive and reducing false negative predictions. Especially in the work developed by [62], algorithms like random forest and logistic regression achieved a significant performance with accuracy over than 80%, being similar or better when compared to neural networks, increasing the candidate pool of ML models for this area. In fact, the ML core pipeline applied to cardiac pathology shown in Figure 3.2 can be used to model distinct applications on the area. Besides each research objective, what makes the pipelines different is the amount of processing performed at each stage and how they are applied, which is the case of the work developed by Ferreira et al. [6] whose focus was predicting pathology from clinical data, where statistical methods were applied as Filter strategies, and, even with limited scope, has shown that FS is important even for datasets with small or medium number of features.

Chapter 4

MS-EPSO

Despite its relative success in solving continuous optimization problems, PSO displays several documented performance issues [63]. To address these issues and at the same improve the robustness of the algorithm, a hybrid approach that combines Evolutionary Strategies (ES) with the classic PSO was proposed by Miranda and Fonseca [11]. The algorithm - Evolutionary Particle Swarm Optimization (EPSO) was able to overcome a crucial issue found in the PSO by introducing genetic operators of the ES: if the particle local best is the same as the global best, the cooperation term would be zero, resulting in a movement provided only by the inertia and memory. However, at the same time, it failed to address another problem: if a particle moves to a new local best position at generation t, its memory term in consecutive iterations will be zero. When a term is zero, the update rule (2.5) does not perform proper local search.

Obviously, a desirable outcome would be to intensify the local search in the region of said particle instead of halting it by having the memory term set to zero. Taking that into account, the Maximum Search Limitations - Evolutionary Particle Swarm Optimization (MS-EPSO) regulates the behavior of the EPSO movement rule by detecting convergence of solutions to accumulation points.

MS-EPSO[1] is an algorithm devised by the author of this thesis in [13]. The algorithm encompasses three distinct rules in one procedure, allowing the algorithm to spread the particles around the neighborhood of a specific particle, while removing solutions from possible local optima that can be found in the environment. Initially, the algorithm performs the following steps: calculate the position-wise statistical metrics mean ($x_{i\mu}$) and standard deviation ($x_{i\sigma}$); initialize the Exploration mode (EXP_i) of each particle with value 1; and initialize the Particle Local Limit (PLL_i) of each particle with value 0. Both EXP and PLL are binary control parameters automatically adjusted according to each rule.

During the optimization process of the algorithm, whenever a particle (including its replicas) fails to move into a better position when compared to its local best position, the PLL value is increased by 1. Combined with the Maximum Local Limit (MLL), the PLL plays an important

[1] The MS-EPSO code is available in https://github.com/MtrsN/MS-EPSO

role in MS-EPSO, since these two parameters are the ones that will establish the access of a specific particle into a specific rule. These rules are described on subsequent sections.

4.1 Rule 1

The first rule is the exploration stage of MS-EPSO. This strategy triggers when the following condition is satisfied:

$$PLL_i < MLL \land EXP_i \tag{4.1}$$

This rule will verify if the local limit has surpassed the maximum local limit and if the exploration mode is turned on. In this rule, one particle is sampled from a normal distribution with mean and standard deviation calculated from the global best position, e.g. $N(x_{g\mu}, x_{g\sigma})$. After it, this particle is evaluated and R replicas are generated. It is worth mentioning that each replica will still use Equation 2.4 to mutate the weights and Equation 2.3 to move in the environment. Also, this procedure changes the order of EPSO mechanism, where replicas are moved prior the particle.

4.2 Rule 2

Rule 2 is the exploitation stage of MS-EPSO. This rule is satisfied according to the following condition:

$$PLL_i < MLL \land \neg EXP_i \tag{4.2}$$

The particle that performs this rule will apply the standard mechanisms of EPSO, giving priority to the replicas and then moving the particle with standard Equation 2.5.

4.3 Rule 3

This corresponds to the restart mechanism of MS-EPSO and is satisfied with:

$$PLL_i \geq MLL \tag{4.3}$$

This would indicate that, compared to its local best position, the particle was not able to move to a better position. Initially, the particle PLL is set to 0 and, if it is the first time visiting this rule, its exploration mode is turned off. Next, one particle is sampled from a normal distribution using the mean and standard deviation of its local best position, e.g. $N(\hat{x}_{i\mu}, \hat{x}_{i\sigma})$. The new particle is evaluated and R replicas are created from it. Finally, the collection of weights of the current particle are reset, being generated using a uniform distribution between range [0, 1].

4.4 EPSO vs MS-EPSO

The neighborhood exploration performed by sampling a particle from the normal distribution will spread solutions around the global best particle environment, increasing the diversity in a local area not in the global. When applying this process in Rule 1, the replicas are capable to exploit the area with a distinct collection of weights, since each replica applies the weight mutation process. If the particle is not able to reach a prominent position, in the next generation, the Rule 3 would reallocate it on the same area of its local best solution. Assuming that EPSO has a great potential on exploitation, when combining the limitations with the movement equation for replicas and its canonical process in Rule 2, it is expected that MS-EPSO would achieve a faster convergence due to the exploration boost, while reaching the same or better position when compared to EPSO. The first comparison made in [13], MS-EPSO was applied to unconstrained benchmarks and constrained engineering design problems, where the presented results shows that MS-EPSO can challenge EPSO and other optimization algorithms.

Algorithm 3 presents the pseudo code of MS-EPSO following the same baselines of PSO and EPSO algorithms. It is possible to visualize that MS-EPSO insert rules that increases the number of functions to be implemented when compared to EPSO and PSO, however, only one rule is applied by each particle at each generation. When analyzing the operations performed inside each rule, Rule 2 is equal to EPSO procedure, while 1 and 3 are basically sampling from distributions, therefore, the operation in Rule 1 and 3 are faster than EPSO procedure, increasing the effort required but minimizing or at least maintaining the processing time.

4.5 Contribution

Besides relevant PSO variants applied to feature selection instances, other versions were reported to show outstanding performance in other families of continuous optimization problems. One such that we emphasize is EPSO. The algorithm reached good solution quality for continuous and integer optimization problems. Notwithstanding, EPSO is the base algorithm used by the top five algorithms in the World Congress of Computational Intelligence (WCCI) 2018 competitions. As an enhancement for EPSO, MS-EPSO emerges with the premise to initialize a better initial population. Since MS-EPSO had significant results in other instances and some state of the art PSO algorithms to the FS problem still have continuous variable encoding, this work allows to investigate: the performance of both EPSO and MS-EPSO to feature selection and their capabilities of assisting ML models to solve real-world problems; the performance of the unique normal distribution movement found in MS-EPSO and its behavior when dealing with a integer optimization problem; point out relevant features that may be used in clinical data approaches;

finally, compare the results 13 distinct metaheuristics, 4 feature selection strategies and literature results for cardiac pathology.

Algorithm 3: MS-EPSO pseudocode

```
Input: Objective function f(x), D, LB, UB, NP, NFE, $\tau$, CP, NR, MLL
Output: Best solution found $P_g = \{x_1, x_2, x_3, ..., x_D\}$
// After each function evaluation:
// 1) Increment FEs counter;
// 2) Check for a possible new global best;
// 3) Check the stopping criteria.
$FEs \leftarrow 0$
$MaxV \leftarrow abs(UB - LB)$
$MinV \leftarrow MaxV * -1$
// Initialization phase
for $i \leftarrow 0$ to to $NP$ do
    Initialize particle $(x_i)$ between [LB, UB]
    Initialize velocity $(v_i)$ between [MinV, MaxV]
    Initialize strategic weights $(w^*_{i1}, w^*_{i2}, w^*_{i3}, w^*_{i4})$ between [0, 1]
    Initialize local limit $(PLL_i) \leftarrow 0$
    Initialize exploration mode $(EXP_i) \leftarrow 1$
    $x_{i\mu} \leftarrow \mu(x_i)$
    $x_{i\sigma} \leftarrow \sigma(x_i)$
    $x_{if} \leftarrow f(x_i)$
end
Save all local best information $(\hat{x}_i, \hat{x}_{if}, \hat{x}_{i\mu}, \hat{x}_{i\sigma})$
Save global best information $(x_g, x_{gf}, x_{g\mu}, x_{g\sigma})$
// Optimization process
repeat
    for $i \leftarrow 0$ to to $NP$ do
        $Rule_1 \leftarrow PLL_i < MLL \wedge EXP_i \ ? \ 1 : 0$
        $Rule_2 \leftarrow PLL_i < MLL \wedge \neg EXP_i \ ? \ 1 : 0$
        $Rule_3 \leftarrow PLL_i \geq MLL \ ? \ 1 : 0$
        if $Rule_1$ then
            $x_{new} \leftarrow$ DrawFromGaussianDistribution$(x_{g\mu}, x_{g\sigma}, D)$ // $D$ samples
            $x_{newf} \leftarrow f(x_{new})$
            $best_{replica} \leftarrow$ GenerateReplicas$(NR, w^*_i, \tau)$ // EPSO strategy
            $best_{replicaf} \leftarrow f(best_{replica})$
        end
        if $Rule_2$ then
            $best_{replica} \leftarrow$ GenerateReplicas$(NR, w^*_i, \tau)$ // EPSO strategy
            $best_{replicaf} \leftarrow f(best_{replica})$
            $x_{new} \leftarrow$ MoveParticle$(x_i, v_i, \hat{x}_i, x_g, CP)$ // EPSO Movement
            $x_{newf} \leftarrow f(x_{new})$
        end
        if $Rule_3$ then
            $x_{new} \leftarrow$ DrawFromGaussianDistribution$(\hat{x}_{i\mu}, \hat{x}_{i\sigma}, D)$ // $D$ samples
            $x_{newf} \leftarrow f(x_{new})$
            $best_{replica} \leftarrow$ GenerateReplicas$(NR, w^*_i, \tau)$ // EPSO strategy
            $best_{replicaf} \leftarrow f(best_{replica})$
            $PLL_i \leftarrow 0$
            $EXP_i \leftarrow 0$
        end
        $x_i, x_{if}, v_i, w^*_i \leftarrow Compare(x_{new}, best_{replica})$
        $\hat{x}_i, \hat{x}_{if}, NewLocalBest? \leftarrow Compare(x_i, \hat{x}_i)$
        if $NewLocalBest$ then
            $\hat{x}_{i\mu} \leftarrow \mu(\hat{x}_i)$
            $\hat{x}_{i\sigma} \leftarrow \sigma(\hat{x}_i)$
        else
            $PLL_i \leftarrow PPL_i + 1$
        end
    end
until $FEs == NFE$
```

Chapter 5

Experiment

This chapter details the experiment performed to apply a collection of techniques/algorithms to feature selection and predict cardiac pathology in children and teenagers. All experiments were performed under a Python 3 environment where each FS algorithm were implemented and integrated to select features for the Scikit-learn ML algorithms.

The chapter is divided as follows: the first section describes the applied pipeline to achieve the predictions; the second section presents the applied algorithms and respective parameters; the third section outlines the cardiac pathology data acquired and exploratory analysis performed; finally, the last section presents other datasets (benchmarks) used to further evaluate the algorithms.

5.1 Experiment pipeline

The pipeline performed in this study is presented in Figure 5.1. This is divided into three stages: 1) Preprocessing - outside of rectangles; 2) Standard machine learning - yellow rectangle; 3) Feature selection and tuning - purple rectangle. Each stage will be detailed in the subsequent subsections.

5.1.1 Stage 1: Preprocessing

The pipeline is initialized by applying preprocessing techniques in order to prepare the dataset for the models. With the final dataset, the same is divided once into X_{train}, y_{train}, X_{test}, y_{test} sets where 33% of the data is allocated to the test set, therefore, 77% is used in the training phase. The same train and test sets are used by all algorithms and strategies in order to perform a correct comparison.

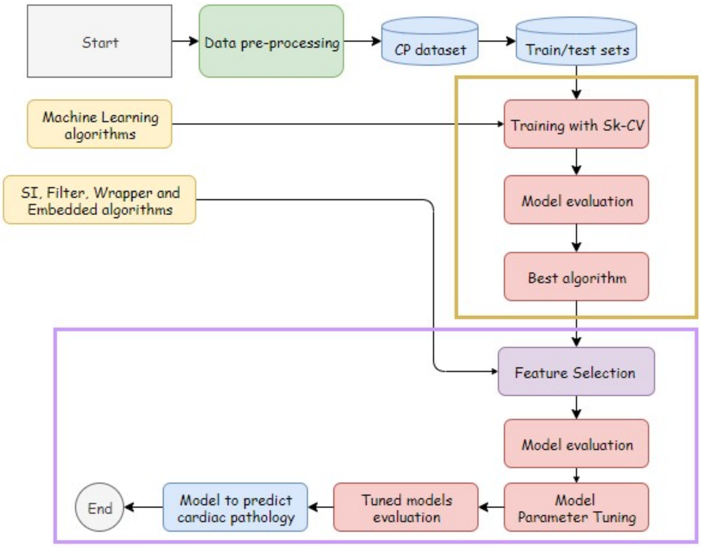


Figure 5.1: Adopted pipeline on this study

5.1.2 Stage 2: Machine Learning Training

In this stage, 5 distinct machine learning models that are commonly used in the literature: Logistic Regression (LR) [39]; K-Nearest Neighbors (KNN) [42]; Artificial Neural Networks (ANN) [45]; Random Forest (RF) [46]; and Support Vector Machine (SVM) [64], are applied.

The models are trained with train sets and their predictive capabilities are evaluated with test sets, where it is measured through the weighted accuracy. The results are compared by applying the Mann-Whitney U (MWU) test to verify the statistical significance between models. It is worth mentioning that all training sessions of ML models are performed under a Sk-cv with 10 folds, where this number was selected due to the properties of Sk-cv with 10 folds that would reduce bias and variance of the model [31]. The most relevant algorithm that yields the best model is used as basis for selecting the best subset of features.

5.1.3 Stage 3: Feature Selection and Tuning

The major contribution of the work is performed at this stage, where MS-EPSO is compared against state of the art techniques to select features for cardiac pathology. The best model acquired in Stage 2 is used with Wrapper (W), Embedded (E) and SI strategies, allowing the feature selection phase to be deeply explored. The collection of algorithms, described in the next section, is applied with the best model to capture the best subset of features. The feature selection is performed with the train sets, generating a model trained with the best subset. As last step, the model performs predictions for the feature subset of test set, where results are evaluated through the balanced accuracy. Finally, the other models are trained and evaluated with the best feature subset with their parameters tuned by a simple Grid Search approach and results are compared.

5.2 Algorithms and parameters

Table 5.1 presents all algorithms applied on this approach with their parameters and categories. The comparison includes SI algorithms, Evolutionary Algorithms (EA) and feature selection strategies, where population based algorithms also share a total of 20 solutions and 200 function evaluations. It is worth mentioning that values marked with a star in the table indicate that the algorithm has the base version parameters and its own, for instance: ABC + ES will have ABC parameters and its own parameters. The parameters were selected based on past approaches and suggestion of the authors that devised the algorithm.

Table 5.1: Algorithms and FS techniques used to predict cardiac pathology

Category	**Algorithm**	**Initials**	**Parameters**
SI	Artificial Bee Colony	ABC [65]	*MaxLimit*= 65
SI	ABC + Evolution Strategies*	ABC + ES [66]	*Replicas*= 1; τ = 0.2; τ' = 0.02
SI	ABC-X	ABC-X-M1 [67]	*MaxLimit*= 140
SI	Ant Colony Optimization	ACO [16]	α= 1; β= 5
SI	Binary PSO	PSO	*w*= 0.6; *c1, c2*= 1.8
SI	Competitive Swarm Optimizer	CSO [68]	ϕ = 0.2
SI	Cross Entropy + EPSO*	CE + EPSO [57]	σ= 0.8; β= 0.1
SI	EPSO	EPSO	τ= 0.8; *cp*= 0.9; *Replicas*= 1
SI	IBPSO	IBPSO [69]	Experiment parameters
SI	MS-EPSO*	MS-EPSO	*MaxLocalLimit*= 25
SI	Social Interaction ACO*	SIACO [70]	R= 3; S= 0; T= 5; P= 1
SI	Quantum PSO with Delta	QPSO [71]	*g*= 0.96
EA	Genetic Algorithm	GA	*cr*= 0.9; *mr*= 0.05;
W	Best model with Wrapper	W-Model	*Exclude worst 10% features*
E	Best model with Embedded	E-Model	*Features with importance* $\geq$ *than median*
F	Pearson Correlation	PC	*40% of features*
F	Mutual Information	MI	*40% of features*

5.3 Cardiac Pathology Data

The data for CP used in this work was collected in a cardiovascular hospital located at the northeastern part of Brazil. It contains 20 features of 17,874 anonymous patients. The dataset is pseudonymised, where patients personal information is modified by an artificial identifier which can be one way to comply both with the European Union's and Brazilian's new General Data Protection Regulation demands for secure data storage of personal information. In the subsequent pages, Table 5.3 describes statistical information found in continuous features of the dataset, while Table 5.2 outlines statistical information found in categorical features.

Table 5.2: Continuous features in the initial dataset

Attribute	**Range**	**Average ± Std.**	**Missing**
Age (cm)	2-19	8.6 ± 3.7	0
Height (cm)	51-198	130.2 ± 21.5	0
Weight (kg)	3.5-101.0	32.7 ± 15.0	0
Body Mass Index	12.0-33.6	18.4 ± 3.6	0
Heart Rate (bpm)	46-160	85.5 ± 11.0	310
Systolic Pressure	70-170	101.0 ± 10.7	20
Diastolic Pressure	35-120	62.1 ± 8.5	20

Table 5.3: Discrete features in the initial dataset

Attribute	**Values**	**Quantity (%)**	**Missing**	**Has pathology (%)**
Gender	Male	59	0	14.21
	Female	41		20.61
Age Range	Pre-School (2-6)	38.3	0	16.08
	School (6-10)	25.2		8.83
	Pre-teen (10-14)	28.3		7.95
	Teenager (14-19)	8.2		1.94
Body Mass Index Percentile	Low Weight	4.5	0	2.05
	Normal	48.7		16.98
	Overweight	17.1		6.18
	Obese	29.6		9.59
Systolic Blood Pressure (SBP)	Normal	91.6	20	31.21
	Limit	3.1		1.21
	Hypertense	5.3		2.39
Systolic Blood Pressure (SBP)	Normal	90.0	20	31.11
	Limit	6.6		2.07
	Hypertense	4.3		1.62
Blood Pressure Result (SBP/DPB)	Normal	86.2	20	29.43
	Limit	6.6		2.43
	Hypertense	7.2		2.95
Murmur	Low Weight	69.5	0	5.61
	Normal	30.4		29.07
	Overweight	0.1		0.06
	Obese	0.1		0.06
Second Heart Sound (*S*2)	Normal	69.5	54	5.61
	Fixed Split	30.4		29.07
	Unique	0.1		0.06
	Hyperphonetic	0.1		0.06
Pulses	Normal	99.8	17	34.61
	Limit	0.1		0.009
	Hypertense	0.1		0.002
Disease History 1	Asymptomatic	72.3	1789	27.72
	Cyanosis	1.0		0.40
	Precordial pain	9.7		3.25
	Dyspnea	6.1		2.49
	Palpitation	5.3		1.44
	Faint/Dizziness	3.2		0.68
	Weight gain	2.4		0.92
Disease History 2	Cyanosis	8.4	6889	4.87
	Precordial pain	18.1		5.51
	Dyspnea	22.9		8.11
	Palpitation	29.7		8.76
	Faint/Dizziness	12.9		4.22
	Weight gain	8.1		3.24
Visit Reason	Cardiopathy	5.7	1846	3.61
	Routine check-up	7.2		1.66
	Others	2.5		0.93
	Cardiology Screening	53.1		13.41
	Possible Cardiopathy	31.5		15.15

5.4 Benchmark Experiment

The benchmark experiment is an extension to the application of MS-EPSO to feature selection. The purpose of this experiment is to verify the performance of MS-EPSO when applied to other types of data, including only discrete, only real, and mixed (Discrete and Real) types of features, distinct number of features and classes. Following the idea of the cardiac pathology experiment, this experiment the steps of this experiment can be visualized in Figure 5.2.

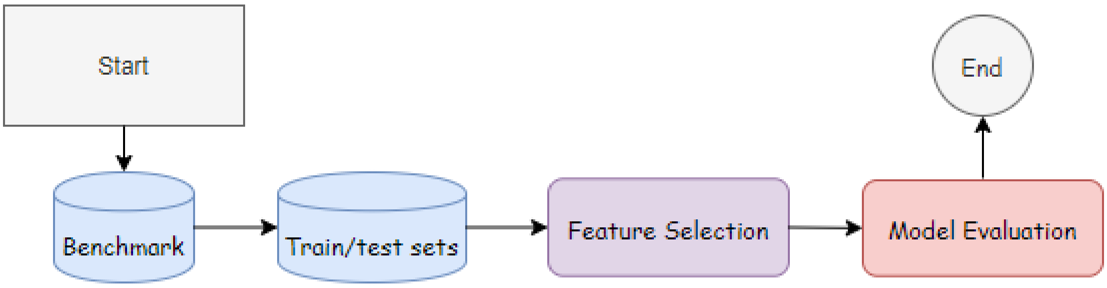


Figure 5.2: Benchmark experiment pipeline

The pipeline starts at the train/test split, where the dataset is once divided into X_{train}, y_{train}, X_{test}, y_{test} sets where 33% of the data is allocated to the test set, therefore, 77% is used in the training phase, same percentage applied at the cardiac pathology. The train set is used at the feature selection phase, where the best subset for the benchmark will be selected according to the algorithm criteria. In case of algorithms that are dependent on ML models to perform this task, the algorithm that yield the best model at the second phase of CP experiment is used, also, it will apply the same 10-fold Sk-CV strategy. At model evaluation phase, the best model make predictions for the test set, where these predictions are evaluated through the balanced accuracy metric.

The results obtained by the algorithms are compared regarding train and test phases, where MWU test is applied to measure statistical significance between them. The list of datasets used in this experiment are described in Table 5.4, where they vary in number of instances, features, classes and feature type. Also, it is worth mentioning that the same group of 17 algorithms with the same parameters as previous experiment are applied. The complete list of algorithms and parameters can be visualized in 5.1.

Table 5.4: Benchmark datasets used to evaluate the FS algorithms

Dataset	**Attributes**	**Features**	**Classes**	**Instances**
Breast Cancer Wisconsin	Mixed	32	2	569
UCI Digits	Discrete	64	10	5,620
Olivetti Faces	Real	4,096	40	400
MNIST	Discrete	784	10	70,000

Chapter 6

Results

This chapter presents the results obtained from the carried out experiment, and it is divided into three main sections: Section 6.1 shows results of the preprocessing step, highlighting the main findings of the exploratory analysis for the cardiac pathology data; Section 6.2 shows results of the conducted pipeline presented in Figure 5.1; Finally, Section 6.3 presents a comparison of the collection of algorithms and techniques to the FS problem when applied to benchmarks found in the literature.

6.1 Preprocessing

The processed dataset was analyzed through bivariate and multivariate analysis in order to explore patterns that could be found in the data. With the premise to create features with richer information and make ease the predictions, these patterns were used in feature engineering, allowing to generate novel features: the Body-Mass-Index (BMI) and the History based Emergency Level (HEL) that represents a relation between the S_2 sound state, patient history of disease

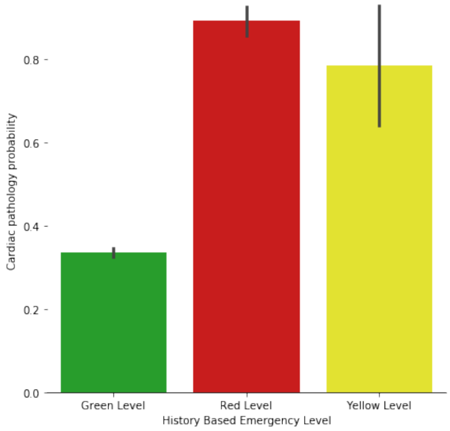


Figure 6.1: History based emergency level relation with cardiac pathology

and the target variable (if the patient had or not cardiac pathology). With the relation captured under the multivariate analysis, it was possible to generate a green, yellow or red state for the patient. Figure 6.1 shows the probabilities emergency levels related to cardiac pathology, while Figures 6.2 and 6.3 present, respectively, the pairwise Pearson correlation and normalized mutual information, which allow to confirm that the created features have significant information regarding other dataset features, fitting the premise of the feature engineering.

The final dataset used on this study has 14 features: Weight; Height; BMI; Age; Pulse rate; PPA - result of systolic blood pressure divided by diastolic blood pressure; Presence or absence of CP; type of the S_2; murmur type; cardiac frequency; history of disease; reason for being forwarded to the cardiology clinic; HEL; BMI. Regarding the description of the dataset, Table 6.1 presents the statistical summary of the continuous features while Table 6.2 presents the categorical variables. Preprocessing was performed in order to follow medical domain standards (e.g. age ranges, pressure rangers, etc) applying: data transformation; cleaning; normalization; removal of irrelevant features, like ID and features with more than 95% of missing values; and patients that had more than 10 features with missing values, ending in a dataset with a population of 9,484 (53% of the original dataset) and 12 features. It is worth mentioning that 6,144 individuals (64.96%) were healthy and 3,340 (35.31%) had CP.

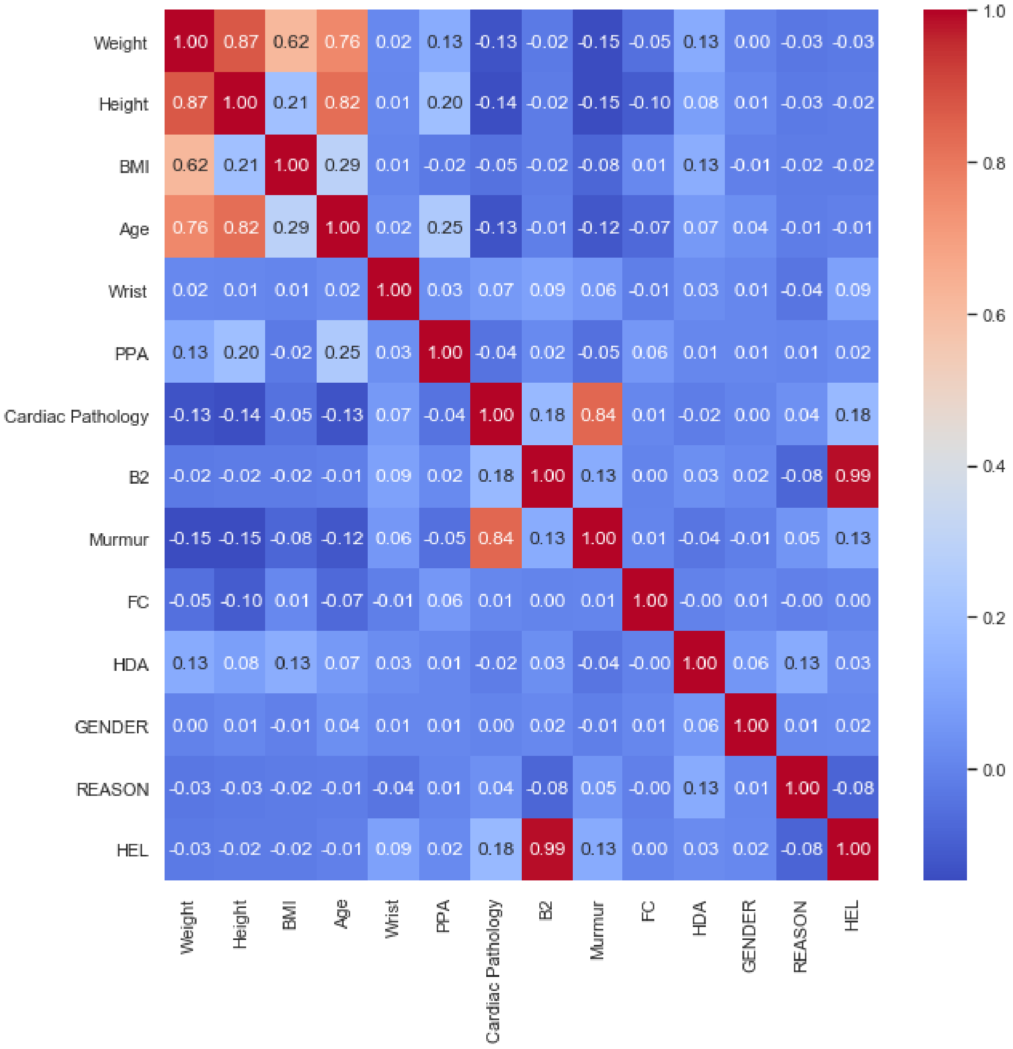
Table 6.1: Continuous features of the final dataset

Attribute	Range	Average ± Std.
Weight (kg)	4.2 - 118.0	34.52 ± 4.08
Height (m)	0.23 - 1.7	0.48 ± 1.98
Body Mass Index	4.08 - 19.04	15.0 ± 39.79

The final dataset can provide an insight about the feature selection with filter strategy, as an example: Pearson correlation as feature selection strategy would select features related to the patient and murmur to be inserted into the best subset, while other strategies would prioritize features that would increase the cost function score. The data now have 0 missing values and the correct shape to be propagated to ML models, therefore, the training phase was performed and its results are given in next section.

Table 6.2: Categorical features of the final dataset

Attribute	Values	Quantity (%)	Pathology probability (%)
Age Range	Pre-school (2-6)	40.85	41.32
	School (7-12)	35.39	31.45
	Teenager (12-19)	23.76	26.03
Disease History	Asymptomatic	55.70	37.96
	Faint/Dizziness	24.15	22.87
	Precordial pain	9.04	34.11
	Palpitation	3.42	23.93
	Dyspnea	4.67	40.43
	Weight gain	2.13	37.67
	Cyanosis	0.84	37.93
Gender	Male	60.52	33.60
	Female	39.48	33.74
Heart Rate	Normal	60.52	33.60
	Abnormal (Above/Under for age limit)	39.47	33.74
Heart Murmur	Normal	70.44	7.88
	Abnormal (Limit/Hypertense)	29.56	95.09
History based Emergency Level	Green level	97.49	32.30
	Yellow level	0.23	75.00
	Red level	2.27	87.74
PPA	Normal	59.71	35.28
	Abnormal (Limit/Hypertense)	40.29	31.24
Reason of visit	Had heart disease	5.56	62.10
	Routine Check-up	7.29	23.49
	Heart evaluation	52.43	24.51
	Suspicious heart disease	31.24	46.06
	Others	3.45	36.01
Second Heart Sound (S_2)	Normal	97.50	32.30
	Abnormal (Limit/Hypertense)	2.50	86.54
Wrist State	Normal	99.45	33.43
	Abnormal (Diminished femoral/Ample)	0.54	75.67



	Weight	Height	BMI	Age	Wrist	PPA	Cardiac Pathology	B2	Murmur	FC	HDA	GENDER	REASON	HEL
Weight	1.00	0.87	0.62	0.76	0.02	0.13	-0.13	-0.02	-0.15	-0.05	0.13	0.00	-0.03	-0.03
Height	0.87	1.00	0.21	0.82	0.01	0.20	-0.14	-0.02	-0.15	-0.10	0.08	0.01	-0.03	-0.02
BMI	0.62	0.21	1.00	0.29	0.01	-0.02	-0.05	-0.02	-0.08	0.01	0.13	-0.01	-0.02	-0.02
Age	0.76	0.82	0.29	1.00	0.02	0.25	-0.13	-0.01	-0.12	-0.07	0.07	0.04	-0.01	-0.01
Wrist	0.02	0.01	0.01	0.02	1.00	0.03	0.07	0.09	0.06	-0.01	0.03	0.01	-0.04	0.09
PPA	0.13	0.20	-0.02	0.25	0.03	1.00	-0.04	0.02	-0.05	0.06	0.01	0.01	0.01	0.02
Cardiac Pathology	-0.13	-0.14	-0.05	-0.13	0.07	-0.04	1.00	0.18	0.84	0.01	-0.02	0.00	0.04	0.18
B2	-0.02	-0.02	-0.02	-0.01	0.09	0.02	0.18	1.00	0.13	0.00	0.03	0.02	-0.08	0.99
Murmur	-0.15	-0.15	-0.08	-0.12	0.06	-0.05	0.84	0.13	1.00	0.01	-0.04	-0.01	0.05	0.13
FC	-0.05	-0.10	0.01	-0.07	-0.01	0.06	0.01	0.00	0.01	1.00	-0.00	0.01	-0.00	0.00
HDA	0.13	0.08	0.13	0.07	0.03	0.01	-0.02	0.03	-0.04	-0.00	1.00	0.06	0.13	0.03
GENDER	0.00	0.01	-0.01	0.04	0.01	0.01	0.00	0.02	-0.01	0.01	0.06	1.00	0.01	0.02
REASON	-0.03	-0.03	-0.02	-0.01	-0.04	0.01	0.04	-0.08	0.05	-0.00	0.13	0.01	1.00	-0.08
HEL	-0.03	-0.02	-0.02	-0.01	0.09	0.02	0.18	0.99	0.13	0.00	0.03	0.02	-0.08	1.00

Figure 6.2: Pairwise correlations of the final dataset features

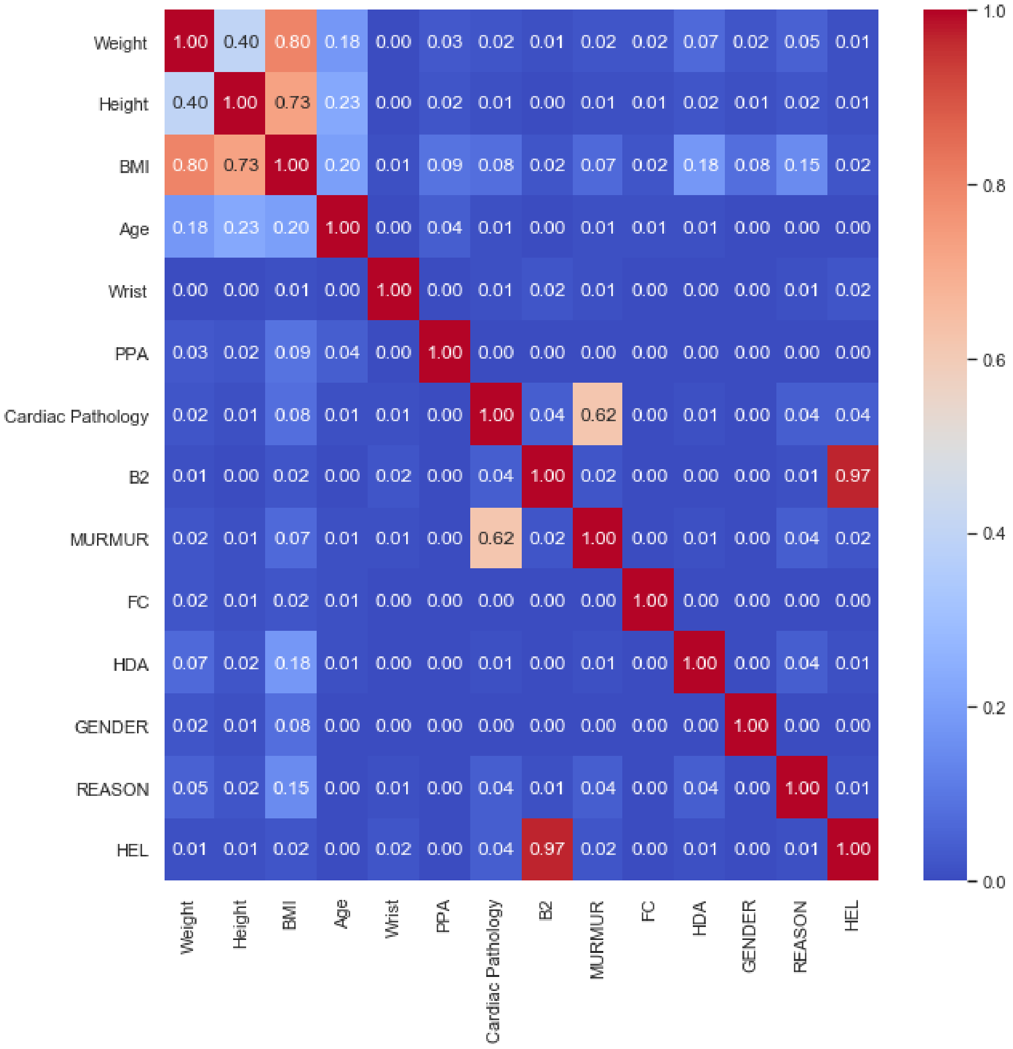


Figure 6.3: Pairwise normalized mutual information of the final dataset features

6.2 Cardiac Pathology

In the second stage of the adopted pipeline, five ML algorithms are applied to the CP dataset with the complete set of features. The results obtained by each algorithm are presented in Figure 6.4, which shows the mean of the Sk-cv and the obtained scores in the test set. This figure shows that the LR model achieved the best balanced accuracy with 86.03% on the mean of the stratified k-fold cross validation and 85.13% on the separated test set. When comparing logistic regression with the neural network, MWU indicates that both populations were acquired from different distributions, therefore, the LR model had statistical significance when compared to ANN ($p = 0.0369$; $p \leq 0.05$).

Since the LR model achieved the best score when applying all features in stage 1, we decided to use the LR algorithm as basis for the feature selection training. Table 6.3 presents: the mean and standard deviation obtained from the stratified k-fold cross validation; the test score obtained from the separated dataset; number of selected variables; algorithm rank when comparing the test score.

The results show that, regarding the objective function, ABC-X-M1 obtained the best score followed by ABC + ES and CE + EPSO. When comparing both algorithms using MWU test, the result indicates that these are not statistically significant ($p \geq 0.05$). MS-EPSO and PSO are the algorithms with subsequent best score, however, when comparing them against the ABC-X-M1, ABC + ES, IBPSO and CE + EPSO, these algorithms have statistical significance ($p \leq 0.05$). This significance may appear due to the low standard deviation acquired by all algorithms when performing their optimization process. Looking deeply at the obtained scores, it is clear that QPSO, Embedded-logistic regression (E-LR) and mutual information (MI) were not capable of finding relevant features.

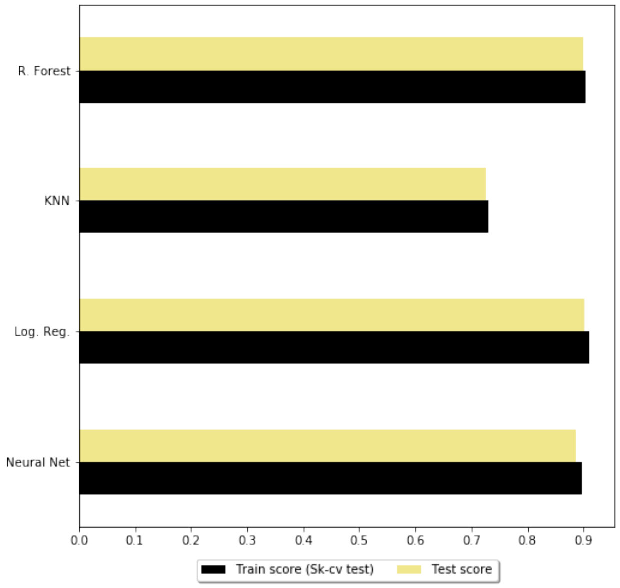


Figure 6.4: Mean of stratified k-cross validation and test set results of stage 2

After the feature selection, each subset was evaluated with the test set given by the train/test split performed in phase 1. The results show that MS-EPSO found a subset of features capable of generalizing the whole dataset, followed by CE+EPSO which also had a significant score, ABC + ES and ABC-X-M1. The difference on test score may be due to the number of selected features that would make ease discovering a pattern in the data. MS-EPSO selected the following features: Age; PPA; B2 state; heart murmur; and disease history, while ABC selected HEI instead of disease history. CE + EPSO selected the same as MS-EPSO but including the wrist state. ABC-X-M1 also selected the same as MS-EPSO but including HEI and wrist state. Finally, algorithms that had poor performance did not include any relevant feature concerning the medical concepts of cardiac pathology (the selected features were related to the patient age, height, BMI, gender and disease history).

The results of refitting the ML models were 0.9233% of balanced accuracy for the ANN, KNN, RF and SVM models, while the LR model achieved the 0.9267% shown in Table 6.3 based on MS-EPSO FS. The parameter tuning of the LR was performed to tune the optimization solver, regularization strength (C) and a decision variable to introduce bias in the cost function. The parameter tuning was able to enhance the training score, however, the test score was not changed. Final parameters of the LR model were: 100 iterations; C is 1.0; LBFGS solver with L2 penalty; and adding the bias to the LR cost function, where these parameters obtained: 0.9304% in the Sk-cv score; and 0.9267% in the test set.

Table 6.3: Feature selection of cardiac pathology dataset results

Algorithm	Sk-cv score	Test score	Variables	Rank
ABC	**0.9277 ± 0.0013**	**0.9064**	**5**	**2**
ABC + ES	**0.9309 ± 0.3640**	**0.9065**	**9**	**3**
ABC-X-M1	**0.9477 ± 0.1074**	**0.9065**	**8**	**3**
ACO	0.9277 ± 0.0013	0.9064	5	4
EPSO	0.9277 ± 0.0013	0.9064	5	4
CE + EPSO	**0.9300 ± 0.0009**	**0.9206**	**7**	**2**
CWO	0.9277 ± 0.0013	0.9064	5	4
IBPSO	**0.9477 ± 0.1074**	**0.9065**	**8**	**3**
MS-EPSO	**0.9286 ± 0.0101**	**0.9267**	**6**	**1**
PSO	0.9277 ± 0.0013	0.9064	5	4
SIACO	0.9277 ± 0.0013	0.9064	5	4
QPSO	0.5089 ± 0.0202	0.5000	5	6
GA	0.9277 ± 0.0013	0.9064	5	4
W-LR	0.9277 ± 0.0013	0.9064	5	4
E-LR	0.5089 ± 0.0202	0.5000	5	6
PC	0.8504 ± 0.1345	0.8509	5	5
MI	0.5089 ± 0.0202	0.5000	5	6

6.2.1 CP Results discussion

Applying feature selection is in fact relevant to predict CP with this dataset. When comparing the best score of 18 algorithms/strategies for feature selection plus ML models with all features plus ML models, a logistic regression model behaved approximately 7.5% better. Also, the KNN algorithm had a poor performance when using all features. With the FS, the model was able to have a competitive score when compared to others used in this work.

Regarding the feature selection stage, most of the SI/EA strategies outperformed the embedded and filters techniques, while the wrapper had competitive results. The SI approaches have a slightly better performance when compared to the wrapper strategy, however, the SI strategies require more computational time since they have to fit the model many times. Given that the best SI result is approximately 2% better and slower, the wrapper strategy has some advantage. But since we are dealing with a medical dataset where priority should be given to the reduction of false negatives, SI algorithms would be the best strategy to circumvent that issue.

The parameter tuning performed on the models was able to enhance in less than 1% the Sk-cv score, which is not significant, therefore, the parameter tuning could be avoided or combined with the feature selection to reduce the time required to complete the pipeline.

As the last comparison for the approach, another attempt was performed with feature selection for cardiac pathology from raw data. The complete experiment is described in [6]. The approach applied had focus on a preprocessing and elimination of features based on a few filter strategies. The best score obtained was close to 90% accuracy. Comparing our results with this, we achieved a 2 points improvement, also reducing the number of false negatives, yielding SI algorithms, but mainly MS-EPSO, as a viable alternative for feature selection in the medical domain.

To further implement this model at the hospital environment, the final model (Logistic Regression algorithm and its final weights optimized during the feature selection stage) needs to be encapsulated under a program interface. Through the interface, the doctor or nurse can insert the collected data, propagating it to the model which will receive the information and return a probability of being a positive case of cardiac pathology. The outcome will be analyzed by the doctor, assisting his decision making process regarding the presence or absence of the pathology.

6.3 Benchmark Results

The results obtained with the benchmark experiment are shown in Table 6.4. When evaluating the algorithm performance on Breast Cancer benchmark, all algorithms had the same performance on the test set, however, analyzing the objective function of the feature selection problem (training phase), ABC-X-M1 and IBPSO had the best performance among the methods, followed by canonical ABC, ABC + ES, ACO, GA and QPSO, while PSO was the third place. MS-EPSO had the same performance of CE + EPSO and CWO which was slightly better when compared to EPSO algorithm. Regarding the p-values, for the Breast Cancer dataset, the MWU test shows that each rank has significance when compared to subsequent rank, e.g. algorithms with rank 1 are statistically significant ($p \leq 0.05$) when compared to algorithms in rank 2, rank 2 is significant when compared to rank 3, until rank 7 which had the worst performance. However, looking forward to the number of selected features and score obtained in the test set, E-LR selected only 3 features, while other strategies have selected more than 12, which would set a priority for this method, followed by other non SI algorithms and EPSO.

MS-EPSO and CWO had the best performance on Digits dataset. Both algorithms were capable to produce the solution that achieved the best train and test score, which was statistically significant ($p < 0.05$) when compared to E-LR. E-LR had the second best performance followed by PSO and IBPSO, where these algorithms had a similar performance, which fail to reject the hypothesis of MWU test, therefore, there is not significance ($p \geq 0.05$) between them. When comparing E-LR and PSO to W-LR, the results were statistically significant ($p < 0.05$), presenting the top 3 algorithms for the Digits dataset. It is important to notice that W-LR and PC methods had a higher performance on test set instead of training set, which may indicate underfitting, however, since the values does not have a huge difference, it would require other kinds of analysis to further verify this issue. Other algorithms had a significant performance regarding the ML metric analysis, however, they were not significant when comparing to the top 3 ranked algorithms.

The Olivetti Faces dataset had the best subset selected by MS-EPSO, followed by EPSO and then W-LR, SIACO and CE + EPSO. Comparing these algorithms, MS-EPSO did not achieved statistical significance ($p \geq 0.05$) when compared to EPSO, however both algorithms had significance ($p < 0.05$) when compared to all other algorithms. The train score achieved by them, except ABC's and QPSO, was significant analyzing the ML metric. In a counterpart, the test score achieved by all algorithms was an average result, where only the top 4 achieved scores higher than 0.8.

The feature selection performed by the algorithms for the MNIST dataset achieved a great overall performance. All algorithms had more than 0.8913 balanced accuracy in the train phase and more than 0.8711 in the test phase. Regarding the optimization process, MS-EPSO achieved rank number 8, however, regarding the train phase, it is possible to order the algorithms by significance where the first group is: SIACO, ACO, EPSO, CE + EPSO, MS-EPSO, PSO, ABC, W-LR and E-LR; the second group is ABC + ES, ABC-X-M1 GA, CWO, QPSO and IBPSO; finally, the third group is PC and MI methods. These groups are statistically ranked as follows: Group 1 $\neq$ Group 2 $\neq$ Group 3, where the difference is given by MWU test with $p < 0.05$.

Table 6.4: Feature selection of benchmark datasets results

Dataset	Algorithm	Sk-cv score	Test score	Variables	Rank
Breast Cancer	**ABC**	**0.8253 ± 0.1245**	**0.8169**	**20**	**2**
	ABC + ES	**0.8253 ± 0.2945**	**0.8169**	**20**	**2**
	ABC-X-M1	**0.8377 ± 0.3274**	**0.8169**	**18**	**1**
	ACO	**0.8253 ± 0.1245**	**0.8169**	**20**	**2**
	BPSO	**0.8250 ± 0.2201**	**0.8169**	**19**	**3**
	CWO	0.8240 ± 0.1987	0.8169	16	4
	CE + EPSO	0.8240 ± 0.1987	0.8169	16	4
	EPSO	0.8227 ± 0.0013	0.8169	12	5
	IBPSO	**0.8377 ± 0.3274**	**0.8169**	**18**	**1**
	MS-EPSO	0.8240 ± 0.1987	0.8169	16	4
	SIACO	0.8240 ± 0.1987	0.8169	16	4
	QPSO	**0.8253 ± 0.1245**	**0.8169**	**20**	**2**
	GA	**0.8253 ± 0.1245**	**0.8169**	**20**	**2**
	W-LR	0.8083 ± 0.1551	0.8169	12	7
	E-LR	0.8053 ± 0.2002	0.8169	3	7
	PC	0.8083 ± 0.1551	0.8169	12	6
	MI	0.8083 ± 0.1551	0.8169	12	6
Digits	ABC	0.9492 ± 0.0042	0.9464	36	7
	ABC + ES	0.9646 ± 0.0024	0.9065	45	13
	ABC-X-M1	0.9648 ± 0.0018	0.9471	37	5
	ACO	0.9569 ± 0.0042	0.9468	41	6
	BPSO	**0.9681 ± 0.0016**	**0.9601**	**38**	**3**
	CWO	**0.9762 ± 0.0027**	**0.9733**	**56**	**1**
	CE + EPSO	0.9653 ± 0.0025	0.9206	42	11
	EPSO	0.9519 ± 0.0011	0.9064	37	14
	IBPSO	**0.9681 ± 0.0016**	**0.9601**	**38**	**3**
	MS-EPSO	**0.9762 ± 0.0027**	**0.9733**	**56**	**1**
	SIACO	0.9464 ± 0.0008	0.9554	25	4
	QPSO	0.9422 ± 0.0008	0.9374	25	9
	GA	0.9689 ± 0.0013	0.9352	42	10
	W-LR	0.9464 ± 0.0008	0.9554	25	4
	E-LR	**0.9683 ± 0.0008**	0.9704	**36**	**2**
	PC	0.9044 ± 0.0010	0.9082	25	12
	MI	0.9422 ± 0.0008	0.9374	25	8
Olivetti Faces	ABC	0.7801 ± 0.0273	0.7887	2051	10
	ABC + ES	0.7906 ± 0.0453	0.7929	2096	8
	ABC-X-M1	0.7801 ± 0.0150	0.7887	2051	10
	ACO	0.7884 ± 0.0249	0.7916	2019	9
	BPSO	0.9768 ± 0.0133	0.7766	2006	11
	CWO	0.7906 ± 0.0453	0.7929	2096	8
	CE + EPSO	0.9745 ± 0.0141	0.8059	1672	4
	EPSO	**0.9831 ± 0.0139**	**0.8103**	**812**	**2**
	IBPSO	0.9564 ± 0.0122	0.7994	1748	5
	MS-EPSO	**0.9801 ± 0.0141**	**0.8170**	**1552**	**1**
	SIACO	0.7801 ± 0.0150	0.7887	2051	10
	QPSO	0.7945 ± 0.0133	0.7987	1982	6
	GA	0.8882 ± 0.0322	0.7929	1959	7
	W-LR	**0.9589 ± 0.0149**	**0.8102**	**1638**	**3**
	E-LR	0.9564 ± 0.0122	0.7994	1748	5
	PC	0.8222 ± 0.0743	0.6213	1638	13
	MI	0.9523 ± 0.0071	0.7747	1638	12
MNIST	ABC	0.9289 ± 0.0003	0.9128	482	4
	ABC + ES	0.9254 ± 0.0001	0.9102	455	5
	ABC-X-M1	**0.9221** ± 0.0001	**0.9139**	**406**	**3**
	ACO	**0.9311 ± 0.0002**	**0.9166**	**713**	**2**
	BPSO	0.9353 ± 0.0004	0.9008	404	15
	CWO	0.9342 ± 0.0053	0.9042	494	10
	CE + EPSO	0.9353 ± 0.0005	0.9022	579	14
	EPSO	0.9348 ± 0.0005	0.9053	395	7
	IBPSO	0.9302 ± 0.0149	0.9034	852	13
	MS-EPSO	0.9348 ± 0.0004	0.9052	587	8
	SIACO	**0.9323 ± 0.0002**	**0.9196**	**713**	**1**
	QPSO	0.9121 ± 0.0002	0.9039	409	11
	GA	0.9194 ± 0.0007	0.9050	517	9
	W-LR	0.9308 ± 0.0008	0.9056	313	6
	E-LR	0.9339 ± 0.0004	0.9035	433	12
	PC	0.8913 ± 0.0003	0.8711	313	17
	MI	0.9242 ± 0.0005	0.8997	313	16

6.3.1 Benchmark result discussion

Since this experiment was to deeply analyze the MS-EPSO feature selection capabilities, four standard ml benchmarks were selected. The comparison was performed against many other FS strategies and swarm intelligence algorithms were the average rank of all algorithms is presented in table 6.5.

Algorithm	Avg. Rank	Final Rank
MS-EPSO	3.50	1
ABC-X-M1	4.75	2
ACO	4.75	2
SIACO	4.75	2
W-LR	5.00	3
IBPSO	5.50	4
ABC	5.75	5
CWO	5.75	5
E-LR	6.50	6
EPSO	7.00	7
QPSO	7.00	7
GA	7.00	7
ABC+ES	7.00	7
BPSO	8.00	8
CE+EPSO	8.25	9
MI	10.50	10
PC	12.00	11

Table 6.5: Feature selection rank for all algorithms/strategies

It is possible to visualize that MS-EPSO had the lower average position rank when compared to all other strategies, therefore, MS-EPSO can be a viable solution to assist a ML model on the predictive capability. Besides MS-EPSO performance, it is also possible to notice that other swarm intelligence algorithms can useful to enhance to predictive capability of the model. When comparing SI category against filters, filter algorithms had the worst feature selection scenario, which can be explained by the difficulty to find the best percentage of features to be selected by these methods. Concerning Wrapper and Embedded strategies, results show that both achieved overall good performance when compared to other SI methods and that the parameters can be selected easily selected when compared to filter strategies.

When applying the logistic regression model to these datasets with the Sk-cv with 10 folds, the standard deviation obtained was expected ue to the number of folds, which decreases bias that can be introduced by applying a low number of folds. Independent on the number of selected features, the score behavior was the same for all datasets, low standard deviation and good results due to the natural predictive capabilities when training a LR with one vs all strategy. The train/test scores may be improved if any of these tasks is performed: 1) the number of iterations, for all algorithms, is increased; a parameter tuning can be performed to tune the feature selection algorithms, although the computational complexity would increase, requiring a huge processing time to perform all the proposed pipeline; or tuning the parameters of the LR with the selected features, however, since the purpose of this experiment is to evaluate feature selection and not ML models, this strategy was not applied, which differs from the cardiac pathology experiment.

Chapter 7

Conclusion

This dissertation presented a recent algorithm named Maximum Search Limitations - Evolutionary Particle Swarm Optimization (MS-EPSO) devised by the author and its application to the feature selection problem. The algorithm was applied to select the best feature subset for cardiac pathology in children and teenagers of a real-world scenario and 4 distinct benchmarks found in the machine learning literature. Applied preprocessing assisted to prepare the dataset in order to be used by machine learning algorithms that were used as basis for feature selection of most strategies. The experiment was compared using 17 techniques including state of the art swarm intelligence algorithms and feature selection techniques. Results for that MS-EPSO can be competitive to other algorithms since it achieved the best results in the cardiac pathology experiment and the lower average rank in the benchmark experiment. For the CP dataset, in particular, the methods applied here allowed a reduction in the number of false negatives, which is very important in the medical domain.

Even with the competitive results, swarm intelligence approaches have the limitation of being computationally expensive due to the fitness evaluation process. Future works involves the insertion of MS-EPSO and fitness evaluation process under the General Purpose Graphics Processing Unit (GPGPU) architecture to mitigate this problem. For MS-EPSO specifically, the author is conducting experiments with large scale problems to analyze distinct application areas that can be fit by the algorithm. Concerning the optimization process, other mechanism found in other optimization methods that were applied in the dissertation can be inserted into the core algorithm to verify if these components can enhance the local or global search performed by the algorithm during the iterations. Concerning the cardiac pathology problem, in order to enhance the primary care for a faster and accurate process, other approaches are being analyzed to deeply investigate the pathology. For feature selection approaches with MS-ESPO, the author is studying the possibilities to extend MS-EPSO approach to multi-objective fitness evaluation and the possibilities of combining the parameter tuning and feature selection, solving two problems in one step performed by MS-EPSO. Finally, with the combination of all future works, it possible to insert the algorithm into a auto-machine learning framework that would perform a series of approaches including it to solve distinct machine learning or even deep learning problems.

Bibliography

[1] Robert Hunter Semple. Valvular disease of the heart. *London journal of medicine*, 2(23): 1019, 1850.

[2] Margaret E Billingham. Some recent advances in cardiac pathology. *Human pathology*, 10 (4):367–386, 1979.

[3] Stanley J Reiser and Stanley Joel Reiser. *Medicine and the Reign of Technology*. Cambridge University Press, 1981.

[4] Baris Bozkurt, Ioannis Germanakis, and Yannis Stylianou. A study of time-frequency features for cnn-based automatic heart sound classification for pathology detection. *Computers in biology and medicine*, 100:132–143, 2018.

[5] Kipp W Johnson, Jessica Torres Soto, Benjamin S Glicksberg, Khader Shameer, Riccardo Miotto, Mohsin Ali, Euan Ashley, and Joel T Dudley. Artificial intelligence in cardiology. *Journal of the American College of Cardiology*, 71(23):2668–2679, 2018.

[6] P. Ferreira, T. T. V. Vinhoza, A. Castro, F. Mourato, T. Tavares, S. Mattos, I. Dutra, and M. Coimbra. Knowledge on heart condition of children based on demographic and physiological features. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 314–319, June 2013. doi:10.1109/CBMS.2013.6627808.

[7] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[8] Lucija Brezočnik, Iztok Fister, and Vili Podgorelec. Swarm intelligence algorithms for feature selection: a review. *Applied Sciences*, 8(9):1521, 2018.

[9] Eric Bonabeau, Directeur de Recherches Du Fnrs Marco, Marco Dorigo, Guy Theraulaz, et al. *Swarm intelligence: from natural to artificial systems*, volume 1. Oxford university press, 1999.

[10] David Martens, Bart Baesens, and Tom Fawcett. Editorial survey: swarm intelligence for data mining. *Machine Learning*, 82(1):1–42, 2011.

[11] Vladimiro Miranda and Nuno Fonseca. Epso-evolutionary particle swarm optimization, a new algorithm with applications in power systems. In *IEEE/PES Transmission and Distribution Conference and Exhibition*, volume 2, pages 745–750. IEEE, 2002.

[12] José Rueda, Instván Erlich, and Kwang Lee. Modern heuristic optimization. http://sites.ieee.org/psace-mho/, 2019. Accessed: 2019-09-19.

[13] Mário Serra Neto, Marco Mollinetti, Vladimiro Miranda, and Leonel Carvalho. Maximum search limitations: Boosting evolutionary particle swarm optimization exploration. In *EPIA Conference on Artificial Intelligence*, pages 712–723. Springer, 2019.

[14] Marco Dorigo, Mauro Birattari, et al. Swarm intelligence. *Scholarpedia*, 2(9):1462, 2007.

[15] Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, and Nurhan Karaboga. A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review*, 42(1):21–57, 2014.

[16] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[17] David H Wolpert, William G Macready, et al. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[18] Yang Wei and Li Qiqiang. Survey on particle swarm optimization algorithm [j]. *Engineering Science*, 5(5):87–94, 2004.

[19] Russell Eberhart and James Kennedy. Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer, 1995.

[20] Jian-Chao Zeng and Zhi-Hua Cui. A guaranteed global convergence particle swarm optimizer. *Journal of computer research and development*, 8:1333–1338, 2004.

[21] T Modine and JM Elarid. Minimally invasive cardiac surgery, port-access and robotic surgery. In *Minimized Cardiopulmonary Bypass Techniques and Technologies*, pages 229–244. Elsevier, 2012.

[22] Walter L Kemp, Dennis K Burns, and Travis G Brown. *Pathology: the big picture*. Univerza v Ljubljani, Medicinska fakulteta, 2008.

[23] Joao Paulo Solano, Barbara Gomes, and Irene J Higginson. A comparison of symptom prevalence in far advanced cancer, aids, heart disease, chronic obstructive pulmonary disease and renal disease. *Journal of pain and symptom management*, 31(1):58–69, 2006.

[24] Senka Mesihović-Dinarević, Jasna Ibrahimović, Edo Hasanbegović, Emina Ićindić-Nakaš, and Aida Smajić. Heart murmur and anaemia in the pediatric population. *Bosnian journal of basic medical sciences*, 5(3):39–45, 2005.

[25] Microsoft Team. What are ml pipelines in azure machine learning? https://docs.microsoft.com/pt-pt/azure/machine-learning/service/concept-ml-pipelines, 2019. Accessed: 2019-09-19.

[26] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[27] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[28] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[29] Eric Jones, Travis Oliphant, Pearu Peterson, et al. Scipy: Open source scientific tools for python. *1*, 2001.

[30] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[31] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[32] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[33] George Forman and Martin Scholz. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter*, 12(1):49–57, 2010.

[34] Nenkat Venkatraman. The concept of fit in strategy research: Toward verbal and statistical correspondence. *Academy of management review*, 14(3):423–444, 1989.

[35] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006.

[36] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2): 1153–1176, 2015.

[37] Aida Ali, Siti Mariyam Shamsuddin, Anca L Ralescu, et al. Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl*, 7(3):176–204, 2015.

[38] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[39] Raymond E Wright. Logistic regression. *1*, 1995.

[40] David Wooff. Logistic regression: a self-learning text, 2004.

[41] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[42] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[43] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, 1(4):580–585, 1985.

[44] Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1):1304, 2016.

[45] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA:, 2015.

[46] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[47] Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.

[48] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[49] Yongheng Zhao and Yanxia Zhang. Comparison of decision tree methods for finding active objects. *Advances in Space Research*, 41(12):1955–1959, 2008.

[50] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of machine learning research*, 5(Jan):101–141, 2004.

[51] Qasem Al-Tashi, Said Jadid Abdul Kadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Alhussian. Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, 7:39496–39508, 2019.

[52] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[53] Stephenie C Lemon, Jason Roy, Melissa A Clark, Peter D Friedmann, and William Rakowski. Classification and regression tree analysis in public health: methodological review and comparison with logistic regression. *Annals of behavioral medicine*, 26(3):172–181, 2003.

[54] Rüdiger W Brause. Medical analysis and diagnosis by neural networks. In *International Symposium on Medical Data Analysis*, pages 1–13. Springer, 2001.

[55] Phong Thanh Nguyen, K Shankar, Wahidah Hashim, Andino Maseleno, et al. Brain tumor segmentation and classification using knn algorithm. *International Journal of Engineering and Advanced Technology*, 8(6 Special Issue):706–711, 2019.

[56] Shenkai Gu, Ran Cheng, and Yaochu Jin. Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 22(3):811–822, 2018.

[57] Carolina G Marcelino, Carlos Pedreira, Elizabeth F Wanner, Leonel M Carvalho, Vladimiro Miranda, and Armando L da Silva. Ce+ epso: a merged approach to solve scopf problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 69–70. ACM, 2019.

[58] Tamer Ölmez and Zümray Dokur. Classification of heart sounds using an artificial neural network. *Pattern recognition letters*, 24(1-3):617–629, 2003.

[59] Zaid Abduh, Ebrahim Ameen Nehary, Manal Abdel Wahed, and Yasser M Kadah. Classification of heart sounds using fractional fourier transform based mel-frequency spectral coefficients and stacked autoencoder deep neural network. *Journal of Medical Imaging and Health Informatics*, 9(1):1–8, 2019.

[60] Geert Litjens, Francesco Ciompi, Jelmer M Wolterink, Bob D de Vos, Tim Leiner, Jonas Teuwen, and Ivana Išgum. State-of-the-art deep learning in cardiovascular image analysis. *JACC: Cardiovascular Imaging*, 12(8):1549–1565, 2019.

[61] Khader Shameer, Kipp W Johnson, Benjamin S Glicksberg, Joel T Dudley, and Partho P Sengupta. Machine learning in cardiovascular medicine: are we there yet? *Heart*, 104(14): 1156–1164, 2018.

[62] Stephen F Weng, Jenna Reps, Joe Kai, Jonathan M Garibaldi, and Nadeem Qureshi. Can machine-learning improve cardiovascular risk prediction using routine clinical data? *PloS one*, 12(4), 2017.

[63] Andries P Engelbrecht. *Computational intelligence: an introduction.* John Wiley & Sons, 2007.

[64] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[65] Dervis Karaboga and Bahriye Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied soft computing*, 8(1):687–697, 2008.

[66] Marco Antônio Florenzano Mollinetti, Daniel Leal Souza, Rodrigo Lisbôa Pereira, Edson Koiti Kudo Yasojima, and Otávio Noura Teixeira. Abc+ es: Combining artificial bee colony algorithm and evolution strategies on engineering design problems and benchmark functions. In *International Conference on Hybrid Intelligent Systems*, pages 53–66. Springer, 2016.

[67] Doğan Aydın, Gürcan Yavuz, and Thomas Stützle. Abc-x: a generalized, automatically configurable artificial bee colony framework. *Swarm Intelligence*, 11(1):1–38, 2017.

[68] Ran Cheng and Yaochu Jin. A competitive swarm optimizer for large scale optimization. *IEEE transactions on cybernetics*, 45(2):191–204, 2014.

[69] Xiaohui Yuan, Hao Nie, Anjun Su, Liang Wang, and Yanbin Yuan. An improved binary particle swarm optimization for unit commitment problem. *Expert Systems with applications*, 36(4):8049–8055, 2009.

[70] Demison Rolins de S. Alves, Mario Tasso Ribeiro Serra Neto, Fabio dos Santos Ferreira, and Otavio Noura Teixeira. Siaco: a novel algorithm based on ant colony optimization and game theory for travelling salesman problem. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*, pages 62–66, 2018.

[71] Lin-cheng Zhou, Hui-zhong Yang, and Chun-bo Liu. Qpso-based hyper-parameters selection for ls-svm regression. In *2008 Fourth International Conference on Natural Computation*, volume 2, pages 130–133. IEEE, 2008.