PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY KASDI MERBAH OF OUARGLA

Faculty Of New Technologies Of Information And Communication

Department Of Computer Science And Information Technology

MASTER thesis

Domain: Computer science and Information technology

Field : Computer science

Speciality: industrial computing

by : Koutti Djihane and Sehili Safa

Theme

Automating Machine Learning Pipelines: Meta-heuristic Optimizers for Enhanced Performance

Evaluation Date 22/06/2024

Jury:

President	khelifa Meriem	MCB	University of Ouargla
Examiner	Kaoudja Zineb	MCB	University of Ouargla
Supervisor	Mechalikh Charafeddine	MCB	University of Ouargla

Academic year: 2023/2024

Acknowledgements

This research was undertaken at the Department of Computer Science and Information Technology, Kasdi Merbah Ouargla University. We would like to take this opportunity to express our appreciation. Firstly, we would like to thank God for giving us the strength and patience to carry out this humble work. We are immensely grateful to Dr. Mechalikh Charafeddine for his continuous support, help, guidance, motivation, and patience throughout this thesis. A special thanks to Pr. Kherfi Mohammed Lamine for his invaluable support and encouragement, which were instrumental in the successful completion of this research. We want to express our gratitude to Koutti's and Sehili's family members. My father, may God have mercy on him, my mother Djamila, my brothers Saber and Mohammed Lamine, and my sister Hana. Also, my colleague Safa who supported me throughout the year. I would also like to extend my heartfelt gratitude to my father, my mother Rahima, and my unique brother Mohammed, as well as all my sisters: Maroua, Bouthaina, and Bouchra. Your unwavering support and belief in me have been a constant source of strength. Additionally, my colleague Djihane deserves special mention for her invaluable support over the past year. Your encouragement and assistance have been deeply appreciated. we would like to thank everyone who helped us improve our work and who gave us any remarks that helped us perfect this manuscript.

Abstract

Machine learning has become an indispensable tool across diverse domains, facilitating automated decision-making and predictive modeling based on data-driven insights. Classification, a widely used technique in this field, faces challenges with datasets containing numerous features, including redundant and irrelevant ones, complicating the learning process and increasing computation time. This study aims to enhance classification accuracy through optimal feature selection techniques. The methodology involves a comparative analysis of classifiers (decision trees, support vector machines, neural networks) and the evaluation of automatic and manual feature selection methods. Additionally, the integration of meta-heuristic optimizers into the feature and classifier selection process is explored. Results demonstrate significant improvements in classification accuracy and efficiency, underscoring the effectiveness of optimized feature selection. The study's findings contribute to more efficient machine learning processes and better model performance.

Key words: *Machine Learning, Classification, Feature Selection, Hyperparameters Tuning, Meta-heuristic Optimizers.*

Résumé

L'apprentissage automatique est devenu un outil indispensable dans divers domaines, facilitant la prise de décision automatisée et la modélisation prédictive basée sur des analyses de données. La classification, une technique largement utilisée dans ce domaine, est confrontée à des défis avec des ensembles de données contenant de nombreuses caractéristiques, y compris des caractéristiques redondantes et non pertinentes, compliquant le processus d'apprentissage et augmentant le temps de calcul. Cette étude vise à améliorer la précision de la classification grâce à des techniques de sélection de caractéristiques optimales. La méthodologie implique une analyse comparative des classificateurs (arbres de décision, machines à vecteurs de support, réseaux neuronaux) et l'évaluation des méthodes de sélection de caractéristiques automatiques et manuelles. De plus, l'intégration d'optimisateurs méta-heuristiques dans le processus de sélection des caractéristiques et des classificateurs est explorée. Les résultats démontrent des améliorations significatives de la précision et de l'efficacité de la classification, soulignant l'efficacité de la sélection optimisée des caractéristiques. Les conclusions de l'étude contribuent à des processus d'apprentissage automatique plus efficaces et à une meilleure performance des modèles.

Mots clés : *Apprentissage Automatique, Classification, Sélection de Caractéristiques, Optimisation des Hyperparamètres, Optimiseurs Méta-heuristiques.*

Table des matières

Table des figures

List of Abbreviations

AI :	Artificial Intelligence
ANN :	Artificial Neural Network
CSV :	Comma_separated Values
DT :	Decision Tree
FSA :	Feature Selection Algorithm
FP :	False Positive
FN :	False Negative
GA :	Genetic Algorithm
GWO :	Gray Wolf Optimizer
KNN :	K-Nearest Neighbor
LR :	Logistic Regression
ML :	Machine Learning
RF :	Random Forest
SL :	Supervised Learning
SVM :	Support Vector Machines
TP :	True Positive
TN :	True Negative

Chapitre 1

General Introduction

Machine learning has become an indispensable tool across diverse domains, facilitating automated decision-making and predictive modeling based on data-driven insights. Classification is one of the extensively used techniques in this field that requires set of features for learning process [11].

However, improving the learning ability of classification algorithms is more complex especially for dataset containing huge amount of features. Moreover, it makes the classification process tedious and thus a relatively longer time is required for learning every characteristic of the training data. This is due to the existence of redundant and irrelevant features in data which complicates the performance of learning algorithms thereby increasing the computation time [12]. It is necessary to eliminate these irrelevant features from the dataset to achieve effective learning process. Hence, optimal feature selection techniques are required for improving the classification accuracy [13]. These processes reduce the dimensionality of data and make the learning process more efficient.

In this study, we embark on a comprehensive evaluation to address key objectives.

- Firstly, we aim to compare the performance of different classifiers, including decision trees, support vector machines, and neural networks, on a given dataset, assessing their accuracy, precision, recall, and other metrics.

- Secondly, we seek to evaluate the impact of both automatic and manual feature selection methods on model performance, discerning their relative advantages in enhancing accuracy and interpretability. A particular focus of our investigation lies in the integration of Meta-heuristic optimizers into the feature and classifiers selection process, exploring their potential to augment model performance through efficient navigation of the feature space.

The remainder of this thesis is as follows : Chapter 2 presents the literature review. We introduce the generalities of AI and some definitions of classifiers and Feature Engineering. Chapter 3 describes our approach for Optimizing Machine Learning Pipelines with Meta-Heuristic Optimizers, involving of Feature Selection, Classifier Selection, and Hyperparameter Tuning. Chapter 4 in this chapter we describe the dataset and analyze the results. Finally, Chapter 5 concludes the thesis and provides future directions.

Chapitre 2

Literature Review

2.1 Artificial intelligence(AI) :

AI is the development of computer systems that are capable of doing tasks that usually necessitate human cognition. This multidisciplinary field integrates computer science, mathematics, cognitive science, and other fields to create intelligent robots that can perceive, reason, learn, and interact with their surroundings. This means that its systems strive to duplicate or simulate human intelligence, allowing machines to understand and process natural language, recognize images, make decisions, solve problems, and even exhibit creativity. These computers can quickly and accurately do complex tasks thanks to their ability to analyze enormous volumes of data, detect patterns, and extract useful data. The use of AI is widespread and is constantly growing in various areas. AI is revolutionizing every aspect of our lives, from entertainment and transportation to healthcare and banking. It has the ability to better decision-making, automate repetitive operations, increase productivity, and open up fresh prospects for creativity. The two main categories of AI are general AI and narrow AI Narrow AI is created to carry out particular tasks inside a constrained domain. like autonomous systems, recommendation algorithms, and speech recognition systems. On the other hand, general AI refers to highly autonomous machines that have human-level intelligence and are capable of performing a variety of activities. Overall, artificial intelligence is a potent instrument that can enhance human abilities, resolve challenging issues, and influence the future of many industries, fostering progress and opening up new opportunities. This is due to the multiplicity of the strength of his different techniques and approaches, including natural language processing, computer vision, robotics, expert systems, and machine learning which has played a crucial role in advancing AI.

2.2 Maching Learning

Machine learning (ML) is a discipline of artificial intelligence that allows machines to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.

Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data and can independently learn, grow, develop, and adapt.

ML derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process. ML algorithms use computation methods to learn directly from data instead of relying on any predetermined equation that may serve as a model.

The performance of ML algorithms adaptively improves with an increase in the number of available samples during the 'learning' processes. For example, deep learning is a sub-domain of machine learning that trains computers to imitate natural human traits like learning from examples. It offers better performance parameters than conventional ML algorithms.
While machine learning is not a new concept – dating back to World War II when the Enigma Machine was used – the ability to apply complex mathematical calculations automatically to growing volumes and varieties of available data is a relatively recent development.

Today, with the rise of big data, IoT, and ubiquitous computing, machine learning has become essential for solving problems across numerous areas, such as :

- Computational finance (credit scoring, algorithmic trading).
- Computer vision (facial recognition, motion tracking, object detection).
- Computational biology (DNA sequencing, brain tumor detection, drug discovery).
- Automotive, aerospace, and manufacturing (predictive maintenance).
- Natural language processing (voice recognition)[1].

2.2.1 How does machine learning works

ML algorithms are molded on a training dataset to create a model. As new input data is introduced to the trained ML algorithm, it uses the developed model to make a prediction.

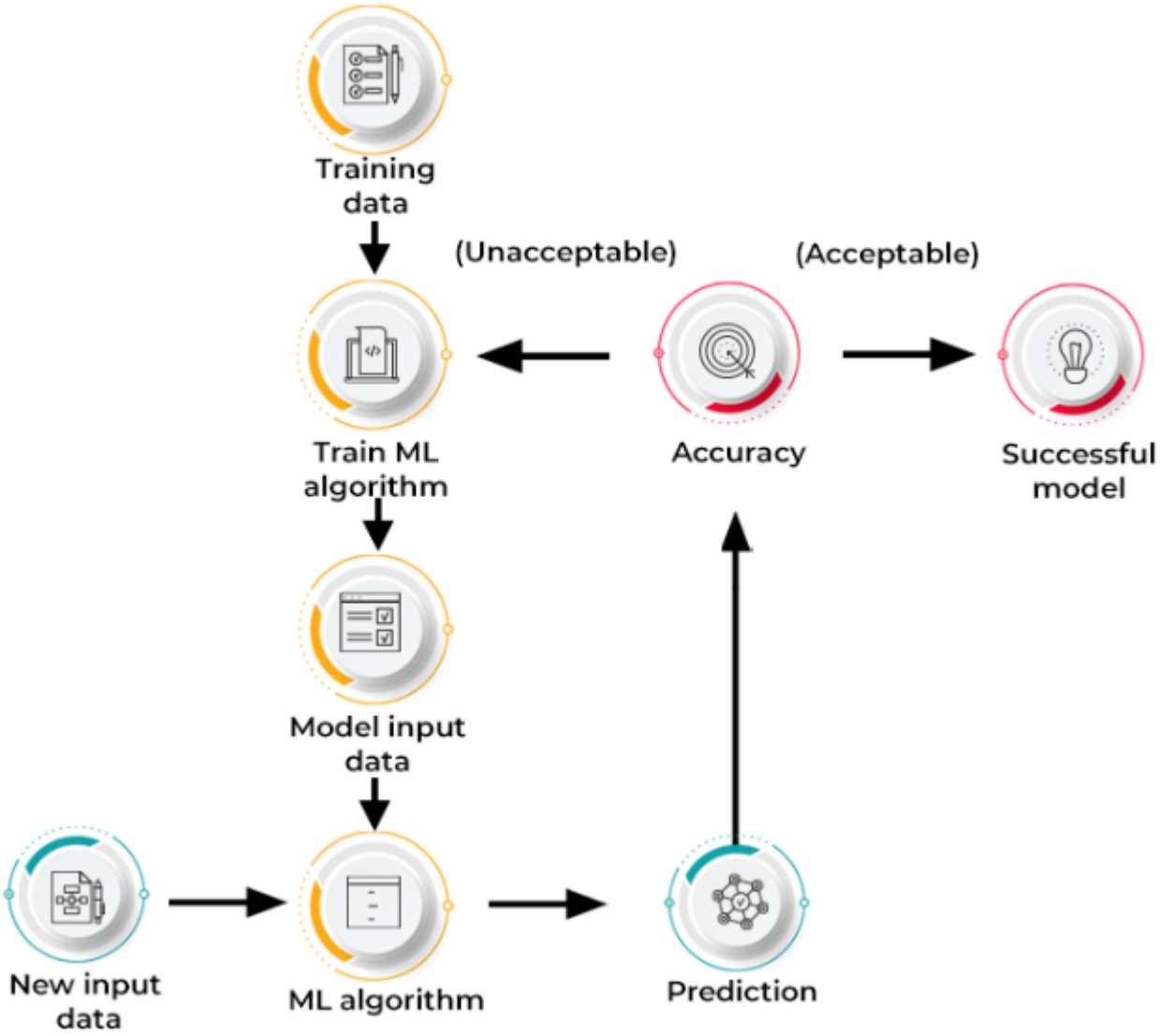

FIGURE 2.1 : how does machine learning works(reprinted from [1].

The above illustration discloses a high-level use case scenario. However, typical machine learning examples may involve many other factors, variables, and steps. Further, the prediction is checked for accuracy. Based on its accuracy, the ML algorithm is either deployed or trained repeatedly with an augmented training dataset until the desired accuracy is achieved[1].

2.3 Types of machine learning :

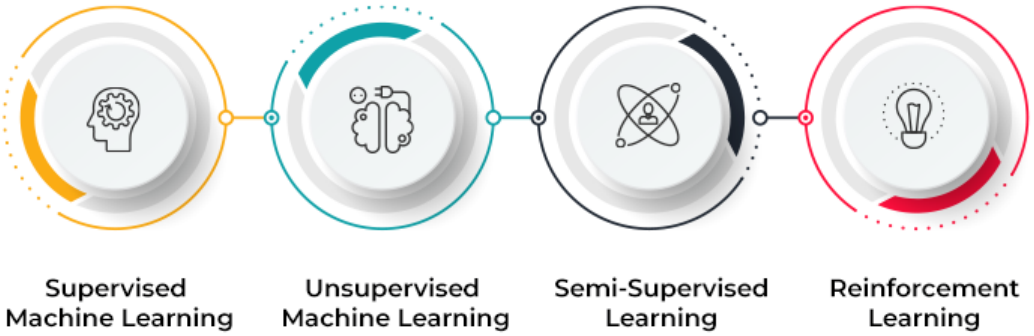

FIGURE 2.2 : Types of machine learning(reprinted from [1].

2.3.1 Supervised learning(SL) :

Supervised machine learning involves training a model on a labeled dataset, where each example consists of input data and corresponding output labels. The goal is for the model to learn the mapping between inputs and outputs, enabling it to make predictions on unseen data accurately. Supervised learning is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns.

It is just like a diligent student learning from a teacher, supervised learning algorithms learn from labeled data.

For example, consider an input dataset containing images of parrots and crows. First, the machines were trained to understand images, including the color, eyes, shape, and size of parrots and crows. After training, given an input image of a parrot, the machine is expected to recognize the object and predict the output. The trained machine examines various features of the objects in the input image, such as color, eyes, shape, etc., to make the final prediction. This is the process of object recognition in supervised machine learning [2].

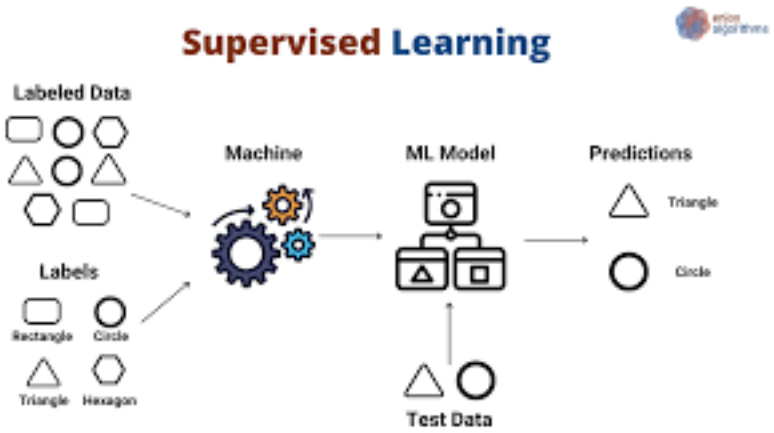

FIGURE 2.3 : Supervised learning(reprinted from [2].

The main goal of SL techniques is to map input variables (a) to output variables (b). Supervised machine learning is further divided into two broad categories :

Classification

In classification problems, algorithms categorize test data into specific groups or classes. For example, they can distinguish between apples and oranges or identify spam emails to filter them out from your inbox. Common classification algorithms include linear classifiers, SVM, DT, and RF.

Regression

Regression is another supervised learning approach that focuses on understanding the relationship between independent and dependent variables. Regression models are useful for predicting numerical values based on various data points. For instance, they can forecast sales revenue for a business based on different factors. Popular regression algorithms include linear regression, logistic regression, and polynomial regression.

2.3.2 Unsupervised learning

Unsupervised learning is a learning technique that does not require supervision. Here, machines are trained on unlabeled datasets and can predict outputs without supervision. Unsupervised learning algorithms are designed to group unsorted data sets based on the inputs' similarities, differences, and patterns[14].

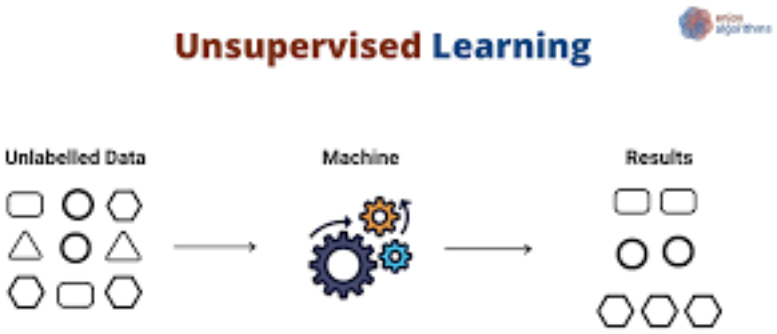

FIGURE 2.4 : unsupervised learning(reprinted from [3].

For example, consider an input dataset containing images of containers filled with fruit. The image here is unknown to the machine learning model. When we feed a dataset into a machine learning model, the model's job is to recognize patterns in objects, such as color, shape, or differences in the input image, and classify them. After classification,the machine predicts the output when tested on a test dataset. Unsupervised machine learning is encouraged and classified into two sorts :

Clustering : The clustering method alludes to gathering objects into clusters based on parameters such as likenesses or contrasts between objects. For case, gathering clients by the items they buy. A few known clustering calculations incorporate the K-Means Clustering Calculation, Mean-Shift Calculation, DBSCAN Calculation, Central Component Examination, and Free Component Investigation.

Affiliation : Affiliation learning alludes to distinguishing ordinary relations between the factors of an expansive dataset. It decides the reliance on different information things and maps-related factors. Normal applications incorporate web utilization mining and advertising information examination. Prevalent calculations complying with affiliation rules incorporate the Apriori Calculation, Eclat Calculation, and FP-Growth Calculation.

2.3.3 Semi-supervised learning :

Semi-supervised learning comprises characteristics of both administered and unsupervised machine learning. It employments the combination of labeled and unlabeled datasets to prepare its calculations. Utilizing both sorts of datasets, semi-supervised learning overcomes the downsides of the alternatives said over.

Consider an illustration of a college understudy. An understudy learning a concept beneath a teacher's supervision in college is named administered learning. In unsupervised learning, an understudy self-learns the same concept at domestic without a teacher's direction. In the interim, an understudy changing the concept after learning beneath the heading of a college instructor could be a semi-supervised frame of learning[1].

2.3.4 Reinforcement learning :

Support learning may be a feedback-based preparation. Here, the AI component naturally takes stock of its environment by the hit trial strategy, takes activity, learns from encounters, and progresses execution. The component is compensated for each great activity and penalized for each off-base move. Hence, the fortification learning component points to maximizing the rewards by performing great activities[1].

2.4 Overview of Machine Learning Classifiers

2.4.1 Decision Tree (DT) :

Decision trees are machine learning algorithms used for classification and regression tasks. It works by partitioning recursively a data of smaller subset based on the values of the input features until a decision can be made about the target variable. In a classification job, a decision tree is constructed by continually dividing the input data into branches based on the values of the input features. The method selects the feature at each split that gives the target variable the highest information gain, i.e., reduces impurity or entropy in the data the most [15] .

$$IG(D, s) = \text{Impurity}(D) - \frac{N_{\text{left}}}{N} \text{Impurity}\left(D_{\text{left}}\right) - \frac{N_{\text{right}}}{N} \text{Impurity}\left(D_{\text{right}}\right) \tag{2.1}$$

- IG(D, s) : The information gain of a split.
- Impurity(D) : The impurity of the dataset before the split.
- Nleft : The number of instances in the left node child.
- Nright : The number of instances in the right child nodes.

- N : The total number of instances in the dataset.
- Impurity (Dleft) : The impurities of the left child nodes.
- Impurity(Dright) : The impurities of right child nodes.

A simple example of a Decision tree is shown below in Figure 1.1.

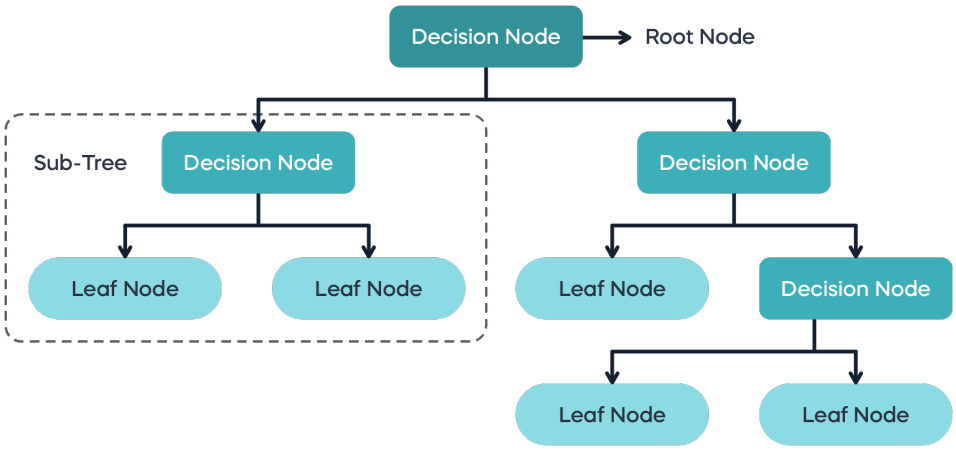

FIGURE 2.5 : The decision tree algorithm(reprinted from [4].

2.4.2 Random Forest(RF) :

Random forest regression is a machine learning approach which clarifies the classification and regression problem and contains plenty of decision trees. The 'Forest' drawn from this technique is qualified by bagging or bootstrap aggregating. To elevate the efficiency, it uses bagging as an ensemble meta-algorithm. The conclusion of this algorithm is based upon the forecast of the decision tree by taking the mean output from different trees. It has been concluded that random forest regression is more reliable than the decision tree model and gives an adequate way of approaching missing data without hyper-parameter tuning. Random forest regression clarifies the problem of over-fitting in decision trees[16, 17, 18]

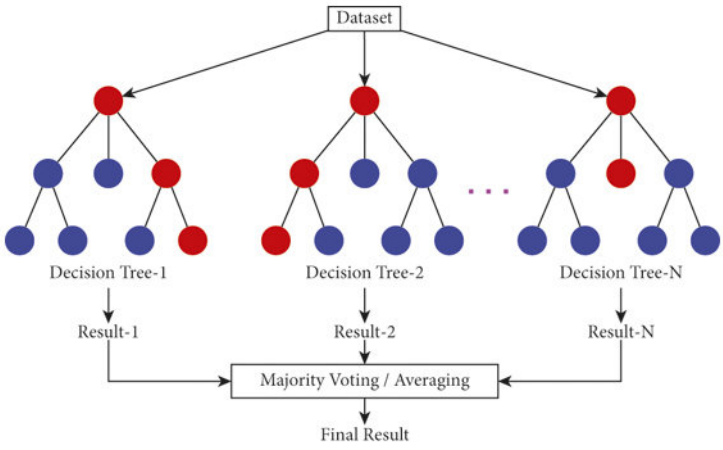

FIGURE 2.6 : Random forest(reprinted from[5])

2.4.3 Logistic Regression (LR) :

In supervised machine learning algorithms, Logistic regression is defined as a statistical model used for binary classification problems, where the goal is to predict the probability of an event occurring or not occurring. by fitting a line to separate two classes.
The LR model can be represented mathematically as[19] :

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \tag{2.2}$$

$P(Y = 1|X)$:The probability of the dependent variable Y being 1 given the values of the independent variables X.

$\beta_n, \beta_1, \beta_2, \ldots, \beta_n$: The coefficients or weights associated with the independent variables.

$X1, X2, ., Xn$: The values of the independent variables.

The likelihood function in LR is based on the assumption that the observations are independent and identically distributed. It can be represented as :

$$L(\beta) = \prod \left[P(Y = 1 \mid X)^Y \cdot (1 - P(Y = 1 \mid X))^{(1-Y} \right\| \tag{2.3}$$

- L(B) : The likelihood function that depends on the coefficients .
- $\prod$:The product operator, which multiplies together all the terms.
- P(Y = 1|X) : The predicted probability of Y being 1 given the value of x.

LR examines the link between the available data (referred to as independent variables) and the likelihood of the event (referred to as the dependent variable) in order to create predictions. Using coefficients, it calculates the influence of each independent variable on the likelihood. In practice, it's easier to work with the log-likelihood function, which is the natural logarithm of the likelihood function :

$$LL(\beta) = \log L(\beta) = \sum [Y \cdot \log(P(Y = 1 \mid X)) + (1 - Y) \cdot \log(1 - P(Y = 1 \mid X))] \quad (2.4)$$

- LL(β)= :The log-likelihood function.
- Σ : The summation operator adds up all the terms.
- Y : The dependent variable.
- P(Y = 1|X) : The predicted probability of Y being 1 given the value of X.

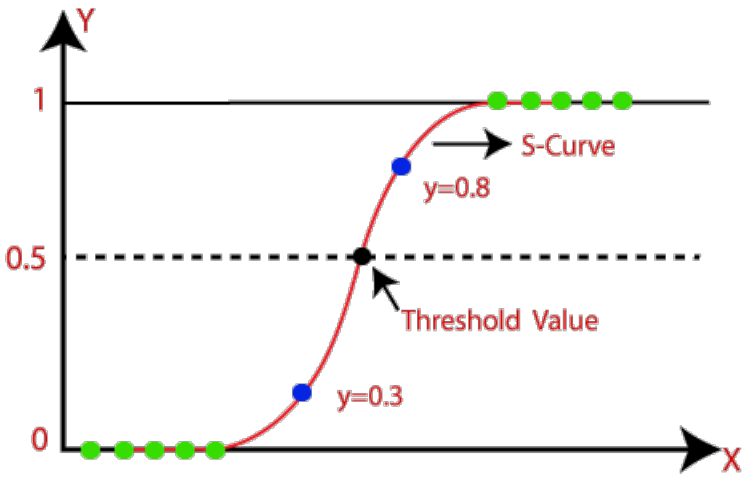

FIGURE 2.7 : The output of Logistic Regression algorithm(reprinted from[6])

2.4.4 K-Nearest Neighbor(KNN) :

The K-Nearest Neighbor is one of the simplest classification methods. In this method, the training samples are referred to as Nearest Neighbors. Moreover, the class labels of the test sample of the K-Neighbors decide the classification of the test sample. The value of k is important and must be sensibly chosen if the k value is too small, then the classifier may suffer the over-fitting issue due to noise in training data. Moreover, when the k value is too large, the issue of misclassification may occur as a classifier [20, 21].

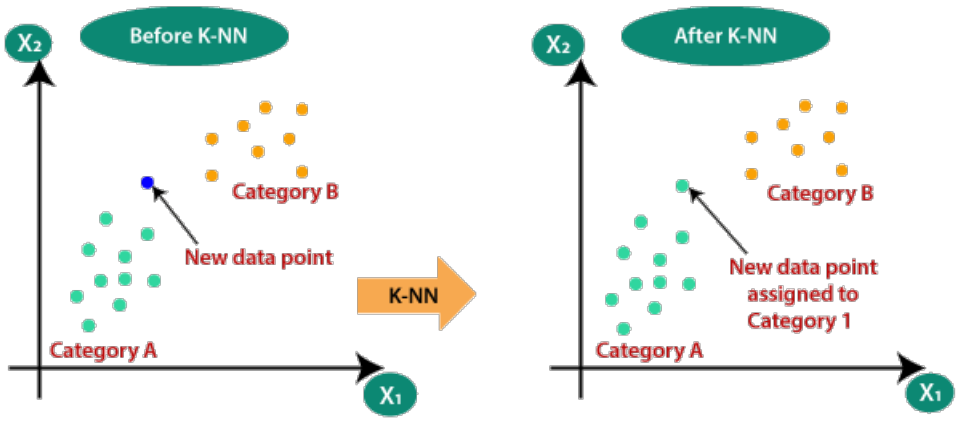

FIGURE 2.8 : Image showing before and after algorithm KNN [7]

2.4.5 Support vector machine(SVM) :

Support vector machine (SVM) [22] is a supervised machine learning algorithm that is looking to determine the best hyperplane that separates the different classes or groups in the dataset. SVM tries to find the hyperplane that best separates classes while maximizing the margin at the same time minimizing the distance between support vectors.

Support vector coordination at simply the closest two points to the margin from each class. The best hyperplane fulfills the following equation :

$$W^t x - b = 0$$

- w : the weights.
- x : data points.
- b : the bias.

The hyperplane and support vectors are pointed out in Figure 2.9 below :

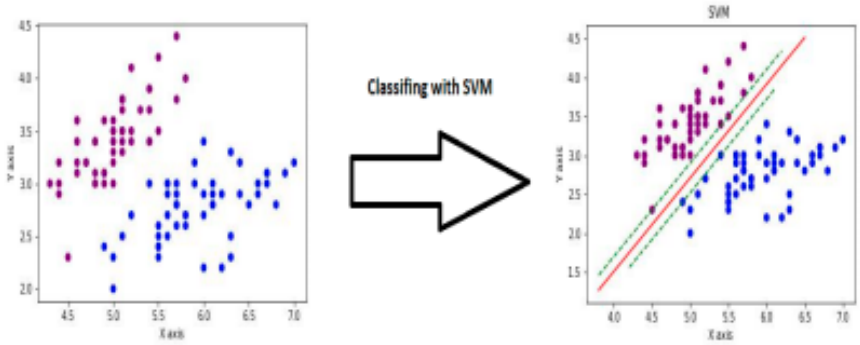

FIGURE 2.9 : The output of SVM algorithm.

2.4.6 Artificial Neural network(ANN) :

A neural network mimics the human brain's processing of data, while deep learning, a subset of machine learning, utilizes interconnected nodes arranged in layers, akin to the brain's structure. This approach fosters adaptive learning in computers, enabling them to learn from errors and enhance performance over time. Consequently, artificial neural networks excel in tackling intricate tasks like document summarization and facial recognition with heightened precision.

2.5 Feature Engineering Techniques :

2.5.1 Manual feature Engineering :

Manually-engineered features based on vibration signals can be difficult to interpret, especially in a real-time manner, other than by an experienced vibration analyst. It involves :

Dimensionality reduction :

The dimensionality reduction is wide spread preprocessing in high dimensional data analysis, visualization and modeling. One of the simplest ways to reduce dimensionality is by Feature Selection ; one selects only those input dimensions that contain the relevant information for

solving the particular problem. Feature Extraction is a more general method in which one tries to develop a transformation of the input space onto the low-dimensional subspace that preserves most of the relevant information [23].

Feature Selection :

High dimensional data consists on features that can be irrelevant, misleading, or redundant which increase search space size resulting in difficulty to process data further thus not contributing to the learning process. Feature subset selection is the process of selecting best features among all the features that are useful to discriminate classes. Feature selection algorithm (FSA) is a computational model that is provoked by a certain definition of relevance. Feature selection methods can be distinguished into three categories : filters, wrappers, and embedded/hybrid method. Wrapper methods perform better than filter methods because the feature selection process is optimized for the classifier to be used. However, wrapper methods have expensive to be used for large feature space because of high computational cost and each feature set must be evaluated with the trained classifier that ultimately make the feature selection process slow[24]. Filter methods have low computational cost and are faster but with inefficient reliability in classification as compared to wrapper methods and better suitable for high dimensional data sets. Hybrid/embedded methods are recently developed which utilize the advantages of both filters and wrapper approaches. A hybrid approach uses both an independent test and performance evaluation function of the feature subset [25]. Filters methods can be further categorized into two groups, namely feature weighting algorithms and subset search algorithms. Feature weighting algorithms assign weights to features individually and rank them based on their relevance to the target concept [26]. A well-known algorithm that relies on relevance evaluation is Relief.

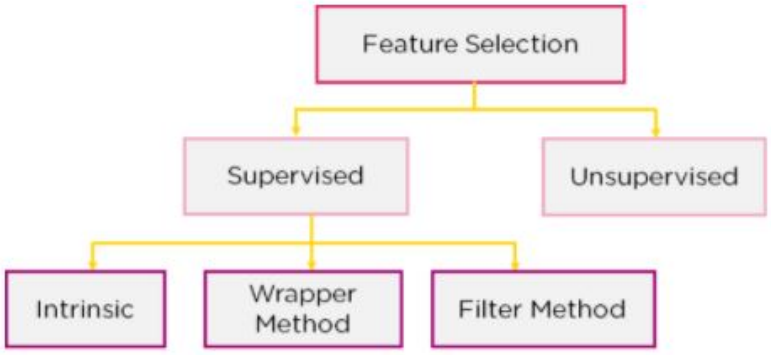

FIGURE 2.10 : Feature Selection Methods : Filters Wrappers(reprinted from[8])

Feature scaling :

Feature scaling is one of the most pervasive and difficult problems in machine learning, yet it is one of the most important things to get right. In order to train a predictive model, we need data with a known set of features that needs to be scaled up or down as appropriate. After a scaling operation, the continuous features become similar in terms of range. Although this step is not required for many algorithms,It is still a good idea to do so. Distance-based algorithms like k-nearest neighbor and k-means, on the other hand, require scaled continuous features as model input. There are two common ways for scaling[27].

2.5.2 Automatic Feature Engineering :

Automatic Feature Engineering is the process whereby computer algorithms are employed to automatically identify, generate, and select the most relevant features from raw data for use in machine learning models. This process aims to optimize the predictive power of models by crafting features that capture essential patterns or relationships within the data, without requiring direct human intervention or deep domain expertise. The process of Automatic Feature Engineering is pivotal in data science and machine learning, as it can significantly enhance model accuracy and efficiency by ensuring that only the most informative and relevant features are used. This not only saves considerable time and resources in the model development phase but also can lead to more robust and generalizable models [28].

2.5.3 Meta-heuristic Optimization Algorithms for Feature Selection and Tuning

Artificial Neural Networks(ANNs) are adopted for classification in data mining fielddue to its higher performance [29]. It is a universal function approximation algorithm for modelling linear and non-linear data with a desired accuracy[30]. The conventional statistical methods have certain drawbacks like complications in the fusion of secondary data and so on. Thus, artificial neural networks are considered as the suitable alternative for these conventional statistical methods. ANNs possess many advantages such as easily adapting to different kinds of data, arbitrary decision boundary capabilities and their nonparametric nature. The learning of training data in neural networks typically takes place in an iterative way which considers all the patterns in the dataset for learning. Therefore, ANNs are called as the data dependent models [30]. During training, the weights of the ANN are adjusted until the actual output of the network and desired output of the network are as close as possible. Hence, ANNs

can be effectively utilised for mapping an input to a desired output, for classifying data and for learning the patterns in the dataset provided.

Feature selection (FS) and feature weighting (FW) are the vital and broadly used data pre-processing methods in machine learning during classification [31]. FS is a combinatorial search problem that eliminates the redundant and irrelevant features and preserves the relevant features related to the dataset [**?**]. Hence, the feature selection process minimises the number of features in the dataset and thereby speeds up the learning process by reducing computational complexity. This reduction in number of features makes the dataset easier to understand and manageable for further classification process [32]. FW is a continuous search problem in which the weights are allotted to features based on their relevance [33]. It approximates the optimal degree of influence of distinct features. Depending upon the individual feature values of the query and instance, the weights are assigned to the features dynamically [34]. FW approaches are suitable when the relevancy of features varies in data [35]. During classification process, each feature in the dataset will have different contribution. That is, some features will be more important than others while solving the classification problem. Hence, higher weights are allotted to the relevant features and lower weights are allotted to the less relevant and redundant features [36].

The techniques in FS and FW are classified under two methods namely, filter method and wrapper method [37]. The filter method is used for filtering the insignificant features which contains lesser option during data analysis. It does not employ any learning algorithm for evaluating the features. The filter methods select a subset with large number of features or even select all the features in the dataset. Thus, a suitable threshold is necessary for choosing the subset. The selected features from the filter method are analysed based on data characteristics like information measures, correlation, consistency and distance in the feature space. The wrapper method uses predictive accuracy of a predetermined learning algorithm for determining the quality of the selected features. Generally, meta-heuristic algorithms are employed as learning algorithms in wrapper methods. The wrapper methods are mostly applied for feature weighting and feature selection process because filter methods have low classification accuracy than wrapper methods [38, 39].

Meta-heuristic algorithms are nature inspired algorithms that mimic the natural activities for solving optimization problem in several computations [40]. The development of nature inspired algorithm lies in the point that it takes its sole inspiration from nature. These inspirations from nature have the ability to solve complex optimization problems. Nature serves as an abundant and massive resource of inspiration for solving complex as well as

hard computing problem in the field of data science as it possesses extremely diverse, robust, complex, dynamic and fascinating substances. It always helps to determine optimal solutions for stochastic problems thereby maintaining a good balance between its key elements. This is the principle behind meta-heuristic optimization algorithms. An optimization algorithm always focuses on exploring and exploiting a search space to maintain a good balance between them [41]. In an algorithm, the exploration phase explores several best locations in the search space while exploitation phase searches the optimal solutions over the best locations [42]. There are many optimization algorithms based on exploration and exploitation process. Each optimization algorithm has its own nature and complexity making it efficient in specific optimization problem and they may be ineffective in other optimization problems. Thus, there is always a need for new optimization algorithms [43].

The optimization process for high dimensional spaces has become more complex under noisy environment because the data is not enough to create a complete mathematical model [44]. Hence, a powerful algorithm is needed for high-dimensional optimization issues. Moreover, no-free-lunch theorem had described that the existing optimization algorithms are not applicable for all optimization problems [45]. This motivates the analysts to develop new and more effective optimization algorithms for solving specific problems in various fields. Furthermore, each meta-heuristic algorithm has similar performance on every optimization problems. Thus, the unresolved problems in existing algorithms can be solved by introducing new meta-heuristic algorithms. In metaheuristic algorithms, the search space exploration is well accomplished and global optimum exploitation is very consistent than other optimization algorithms. Also, it does not get stuck in the local optimum which makes it appropriate for solving problems in engineering sector. Different combinations of metaheuristic algorithms are also introduced in integration with different concepts. For instance, different versions of PSO invented for FS in high-dimensional data include fast hybrid PSO [46], bare-bones PSO [47], variable-size cooperative co-evolutionary PSO [48], and multi-objective PSO with fuzzy cost [49].

Numerous optimization algorithms are developed from the behaviour of some animals or insects in nature namely ant colonies, bees swarm and so on [50]. This is because the biological activities of birds and animals are responsible for specific roles both individually and as a group, to achieve a specific task in their daily routine or lifetime. As a result, they have attracted the attention of data analysts to resolve numerous difficulties in science and engineering sector [51]. For example, Particle Swarm Optimization (PSO) is inspired from the biological behaviour of bird flocking and fish schooling [52], Lion Optimization Al-

gorithm (LOA) simulates the activities of lions and their co-operation characteristics [53], Social Spider Optimization (SSO) algorithm is inspired from the nature of spiders [54], Whale Optimization Algorithm (WOA) imitates the actions of hump-back whales [55], Grey Wolf Optimizer (GWO) imitates the hunting skill and social leadership of grey wolves [56], Artificial Bee Colony (ABC) algorithm mimics the cooperative behaviour of bee colonies [57], Ant Colony Optimization (ACO) simulates the food searching behaviour of ant colonies [58] and so on. These algorithms are applied in different fields like data mining, machine learning and engineering design.

Chapitre 3

Proposed Approach

In this chapter, we describe our comprehensive approach for automating three critical processes in the machine learning pipeline : feature selection, classifier selection, and hyperparameters tuning. Our methodology employs advanced meta-heuristic optimizers, specifically the Gray Wolf Optimizer (GWO) and Genetic Algorithms (GA), to efficiently navigate the search space and identify optimal solutions. It is worth mentioning that the input of our models is a CSV (Comma-separated Values) dataset. CSV is a simple file format used to store tabular data, where each line of the file represents a row of data, and each value within a row is separated by a comma. It is typically used for representing structured data. Figure 3.1 illustrates the overall workflow of our approach.

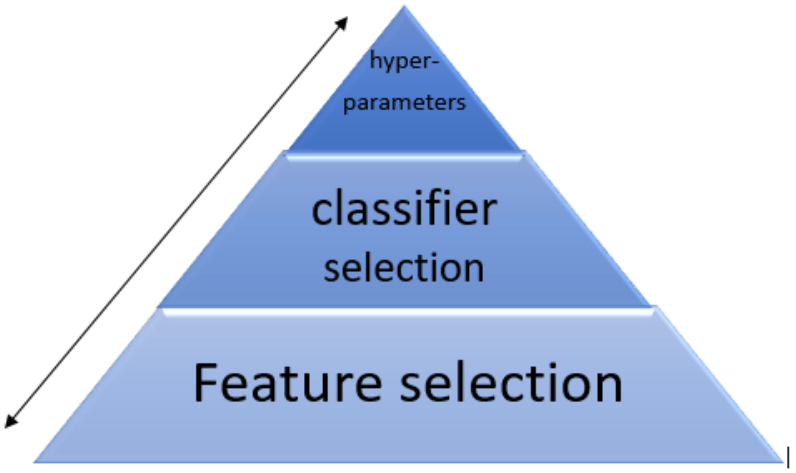

FIGURE 3.1 : The processes we aim to automate

3.1 Manual method

In this section, we will explain the steps of our work where we used SVM, DT, and LR as our classifier methods,and feature engineering techniques , at the end will be comparing the results of the methods.

3.1.1 Dataset selection :

In this research, we initially considered three datasets : breast cancer, Parkinson's disease, and lung cancer. These datasets were selected based on their relevance and the availability of comprehensive data that could be used for detailed analysis and model training. Each dataset contains various features crucial for disease diagnosis and prognosis. However, for the purpose of this study, we decided to focus on the lung cancer dataset for a detailed implementation of manual feature selection techniques.

Description of chosen datasets :

This section focuses on the classification of lung cancer using a diverse set of patient data attributes. The section aims to develop an accurate and reliable model that can assist in the early detection and diagnosis of lung cancer based on patient characteristics and risk factors with kaggle Notebook . The chosen dataset contains details about the individuals with cancer, such as their gender, age, smoking, yellow-fingers, anxiety, peer-pressure, chronic lung disease, fatigue, allergy, wheezing, alcohol Consuming, coughing, shortness of Breath, swallowing difficulty, chest Pain, lung cancer.

3.1.2 Classification

There are many machine learning classification methods, among which we use :

A. Decision Tree :

The DT is usually used to describe the data and can be used to handle a wide range of data types which made it a powerful tool.

B. Random forest :

RF is a multi-decision tree so the training time will be large, and also the testing time since it takes the most voted the prediction of as an outcome .

C.Logistic Regression :

LR is a machine learning method that is used for simplicity, low computation time for training, and prob-ability modeling.

D.K-Nearest neighbor (KNN) :

KNN is a method that takes no training time but takes a lot of time to test that is why called a lazy method which is we did not test it.

E.Support Vector Machine (SVM) :

SVM is the most commonly used supervised machine learning algorithm for classification for its utility and accurate predictions making it an effective method.

3.1.3 Metric

In the context of machine learning for medical diagnosis, the performance metric might change based on the specific task and issue being addressed.

.Accuracy : By comparing the proportion of accurate forecasts to all predictions, accuracy assesses how accurate the model's predictions are overall. Accuracy might not be enough, though, when working with datasets that are unbalanced. Calculated as :

$$\frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{3.1}$$

where TP refers to true positive (i.e., where the models correctly predicts a positive class), TN refers to true negative (i.e., where it correctly predicts a negative class), whereas FP (false positive) and FN (false negative) when, it gets them wrong.

.Recall : It measures the model's ability to correctly identify positive cases. It is calculated as :

$$\frac{TP}{(TP + FN)} \tag{3.2}$$

.Precision : It assesses how well the model can recognize positive cases among the anticipated positive cases. It is determined as :

$$\frac{TP}{(TP + FP)} \tag{3.3}$$

.F1 Score : F1 Score is the harmonic mean of precision and recall,providing a fair assessment of both measurements. It is advantageous when classes are unbalanced since it combines precision and recalls into a single value. Calculated as :

$$\frac{2 * (\text{ Precision } * \text{ Recall })}{(\text{ Precision } + \text{ Recall })} \tag{3.4}$$

3.2 Automatic method :

3.2.1 Gray Wolf Optimizer

The Gray Wolf Optimizer (GWO) is a metaheuristic optimization algorithm inspired by the social behavior of gray wolves in nature. It simulates the hunting process of gray wolves, consisting of three main types of wolves : *alpha*, *beta*, and *delta*. These wolves represent the three best solutions found so far. Additionally, there are a number of other wolves representing the remaining solutions.

The position update equation for each wolf in GWO is defined as follows :

$$\mathbf{X}_{\text{new}} = \mathbf{A} - \mathbf{C} \cdot \mathbf{D}, \tag{3.5}$$

where $\mathbf{X}_{\text{new}}$ is the new position of the wolf, $\mathbf{A}$ is the position of the alpha wolf, $\mathbf{C}$ is a random coefficient matrix, and $\mathbf{D}$ is the distance vector between the current wolf and the alpha wolf.

The algorithm proceeds through iterations, updating the positions of the wolves until convergence criteria are met.

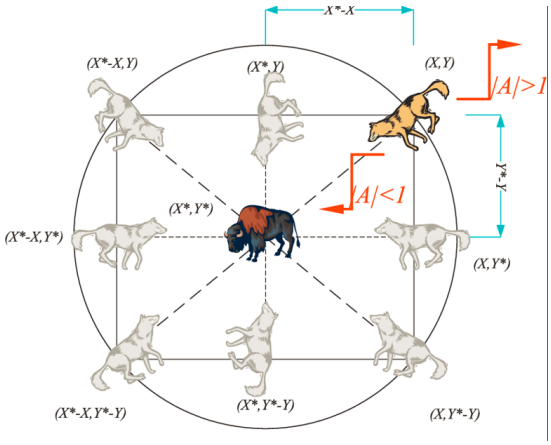

FIGURE 3.2 : Behavior of GWO Algorithm using Meta-heuristics method(reprinted from[9])

Feature Selection and Model Tuning Using GWO

Given a dataset with N instances and M features, the goal is to find a subset of features that optimally represent the data.

In our case, accuracy is used as the fitness function for feature selection (refer to Eq. (3.1))

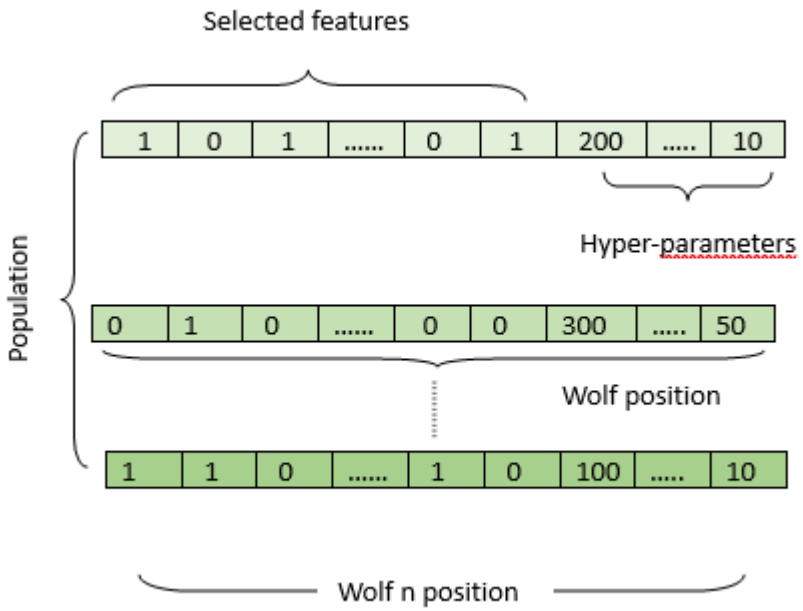

FIGURE 3.3 : Feature selection using GWO

The resulting pseudo-code is as follows :

Algorithm 1: Feature Selection using GWO

Initialize the wolf population randomly;
Evaluate the fitness of each wolf;
for *each iteration* **do**
    Update the positions of alpha, beta, and delta wolves using Equation (3.5);
    Update the positions of the other wolves using Equation (3.5);
    Evaluate the fitness of each wolf;
    Select the new alpha, beta, and delta wolves;
end

This pseudo-code outlines the iterative process of GWO for feature selection. The wolves represent different subsets of features, and their positions are updated to search for the optimal subset that maximizes the classification performance.

3.2.2 Genetic Algorithm

Genetic Algorithms (GAs) are search algorithms inspired by natural selection and genetics. They simulate natural evolutionary processes, operating on chromosomes, which encode the structure of living beings. Unlike other search methods, GAs search across a population of points and work with coded parameter sets rather than the parameter values themselves.

Due to these characteristics, GAs serve as general-purpose optimization algorithms. They can search irregular spaces, making them suitable for various applications, including function optimization, parameter estimation, and machine learning.

Terminology for Genetic Algorithm

1.Population

The population contains a set of possible solutions for the stochastic search process to begin. GA will iterate over multiple generations till it finds an acceptable and optimized solution. The first generation is randomly generated.

2.Chromosome

It represents one candidate solution present in the generation or population. A chromosome is also referred to as a Genotype. A chromosome is composed of Genes that contain the value for the optimal variables.

3.Phenotype

It is the decoded parameter list for the genotype that is processed by the Genetic Algorithm. Mapping is applied to the genotype to convert it to a phenotype.

4.The Fitness function

Also know as the objective function. It evaluates the individual solution or phenotypes for every generation to identify the fittest members.

The pseudo-code for feature selection using GAs is as follows :

Algorithm 2: Feature Selection using Genetic Algorithm

Initialize the population of chromosomes randomly;
Evaluate the fitness of each chromosome;
for *each iteration* **do**
    Select parent chromosomes from the population based on their fitness;
    Apply crossover to generate offspring chromosomes;
    Apply mutation to the offspring chromosomes;
    Evaluate the fitness of each offspring chromosome;
    Select the best chromosomes from the current population and offspring to form the new population;
end

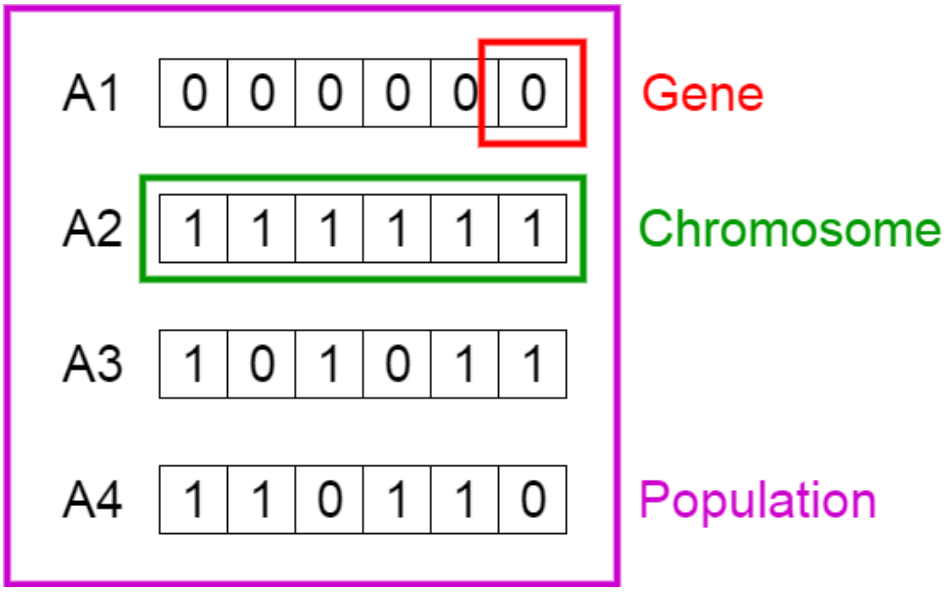

FIGURE 3.4 : Terminology for Genetic Algorithm(reprinted from[10])

Chapitre 4

Experiment and results

In this chapter, we will describe the materials that have been used, the used parameters, and analyze the results.

4.1 Used technologies

1.Kaggle : Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides. As of today (2021), there are over 8 million registered users on Kaggle[59].

2.Python (Programming language) : is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and has a large standard library, making it suitable for a wide range of applications, including web development, data analysis, artificial intelligence, and scientific computing. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's popularity has grown rapidly, and it has a large and active community of developers who contribute to its open-source ecosystem[60].

3.Scikit-learn : is a Python library for machine learning and data analysis. It offers a wide range of tools and algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. Scikit-learn integrates well with other scientific Python libraries

and provides a consistent API. With scikit-learn, users can easily implement machine learning models, preprocess data, and evaluate model performance using various metrics[61].

4.Matplotlib : is a Python library for creating visualizations and plots. It offers a wide range of tools for generating different types of graphs and charts. Matplotlib is highly customizable, allowing users to personalize their visualizations with colors, labels, and other elements. It is widely used in data analysis and scientific research[62].

5.Pandas : is a popular Python library for data manipulation and analysis. It provides efficient data structures and functions for tasks like cleaning, transforming, and analyzing structured data. Pandas simplifies data handling and integrates well with other scientific Python libraries[63] .

4.2 Experiment set-up

4.2.1 Materials

This experiment has been done in both an online editor (kaggle) and a local machine. The machine that has been used for this work has the following specifications :

1. Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz 2.59 GHz.

2. 16 Gb RAM.

4.3 Implementation

To ensure a smooth development of the model and achieve the research objective, a series of steps were followed. These steps involved analyzing the raw data using different techniques to obtain a thorough understanding of the dataset. Additionally, the dataset underwent preparation, which included feature grouping and data cleaning processes. Subsequently, individual models were created and trained using the provided training data. The outcomes of each model were then combined to obtain a comprehensive result.

4.4 Dataset

As said before, this dataset contains information on patients with lung cancer named lung cancer prediction.

4.4.1 Features description :

Gender : The gender of the patient.

Age : The age of the patient.

Smoking : The level of smoking of the patient.

Yellow_Fingers : The level of Yellow-Fingers exposure of the patient.

Anxiety : The level of anxiety of the patient.

Peer_Pressure : The level of Peer-Pressure of the patient.

chronic Lung Disease : The level of chronic lung disease of the patient.

Fatigue : The level of fatigue of the patient.

Allergy : The level of allergy of the patient.

Wheezing : The level of wheezing of the patient.

Alcohol Consuming : The level of alcohol use of the patient.

Coughing : The level of Coughing of the patient.

Shortness of Breath : The level of shortness of breath of the patient.

Swallowing Difficulty : The level of swallowing difficulty of the patient.

Chest Pain : The level of chest pain of the patient.

lung cancer : The level of lung cancer of the patient.

4.4.2 Data Understanding :

This data set was taken from Kaggle and consists of 309 rows and 16 columns.

4.4.3 Data Preparation :

Firstly, we checked for duplicates in the dataset using df.duplicated().sum(), summed up the occurrences of duplicates, and then removed duplicate rows using the drop_duplicates() function (refer to Figure 4.1).

```
In [4]:
#Checking for Duplicates
df.duplicated().sum()

Out[4]:
33
```

FIGURE 4.1 : The number of duplicates in this data set

```
In [5]:
#Removing Duplicates
df=df.drop_duplicates()
```

FIGURE 4.2 : The step of duplication removal

Then, we checked for the count of null values in each column of the dataset using the is null().sum() function. Luckily, this dataset does not have any null value.

```
In [6]:
#Checking for null values
df.isnull().sum()

Out[6]:
GENDER                  0
AGE                     0
SMOKING                 0
YELLOW_FINGERS          0
ANXIETY                 0
PEER_PRESSURE           0
CHRONIC DISEASE         0
FATIGUE                 0
ALLERGY                 0
WHEEZING                0
ALCOHOL CONSUMING       0
COUGHING                0
SHORTNESS OF BREATH     0
SWALLOWING DIFFICULTY   0
CHEST PAIN              0
LUNG_CANCER             0
dtype: int64
```

FIGURE 4.3 : Image showing null values.

Finally, we eliminated the 'GENDER' column for data cleaning by removing irrelevant or unnecessary columns.

```
[29]:
df = df.drop(columns='GENDER')
```

FIGURE 4.4 : Example of column removal

After that, we transformed the categorical column 'LUNG CANCER' into a numerical column 'lung cancer'. The function checked each value of 'Lung cancer' and assigned a numerical value (1 for 'YES', 0 for 'NO'). This was done using a LabelEncoder from Sklearn .

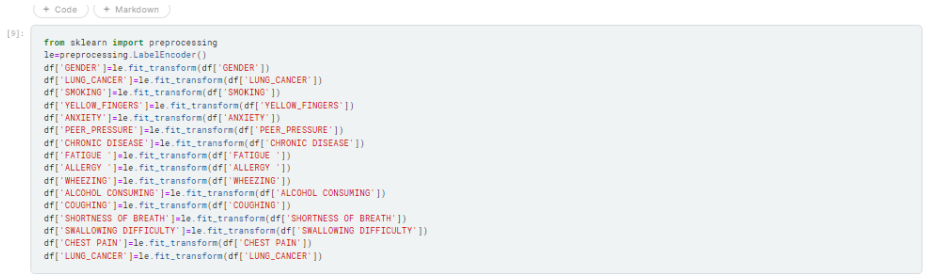

FIGURE 4.5 : Conversion of Categorical LUNG CANCER to Numerical Values Using LabelEncoder.

Overall, these steps ensured that the dataset was free of null values and duplicate entries, preparing it for further analysis and modeling.

4.4.4 Correlation

This analysis utilized the Pearson correlation coefficient to explore the linear relationships between the variables in our dataset. The resulting correlation matrix provides a comprehensive overview of these pairwise correlations. Each value within the matrix represents the strength and direction of the linear association between two specific variables. Values range from -1, indicating a perfect negative correlation, to +1, signifying a perfect positive correlation. A value of 0 suggests no correlation between the variables.

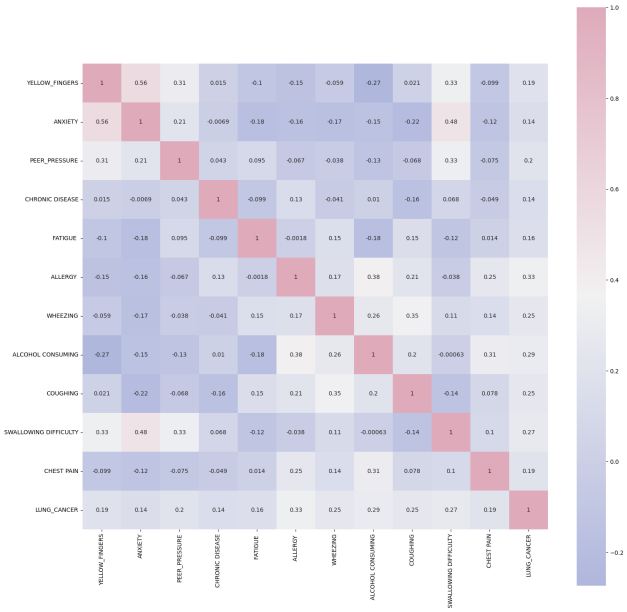

FIGURE 4.6 : A correlation matrix reveals the strength of relationships between variables.

Correlation analysis

This part refines the correlation analysis. It creates a new heatmap by filtering the original matrix to only show correlations with a strength (absolute value) of 0.40 or higher. The heatmap uses a blue colormap, where darker shades represent strong negative correlations and lighter shades represent strong positive correlations. This helps you focus on the most impactful relationships between variables in your data, making it easier to identify potential areas for further investigation.

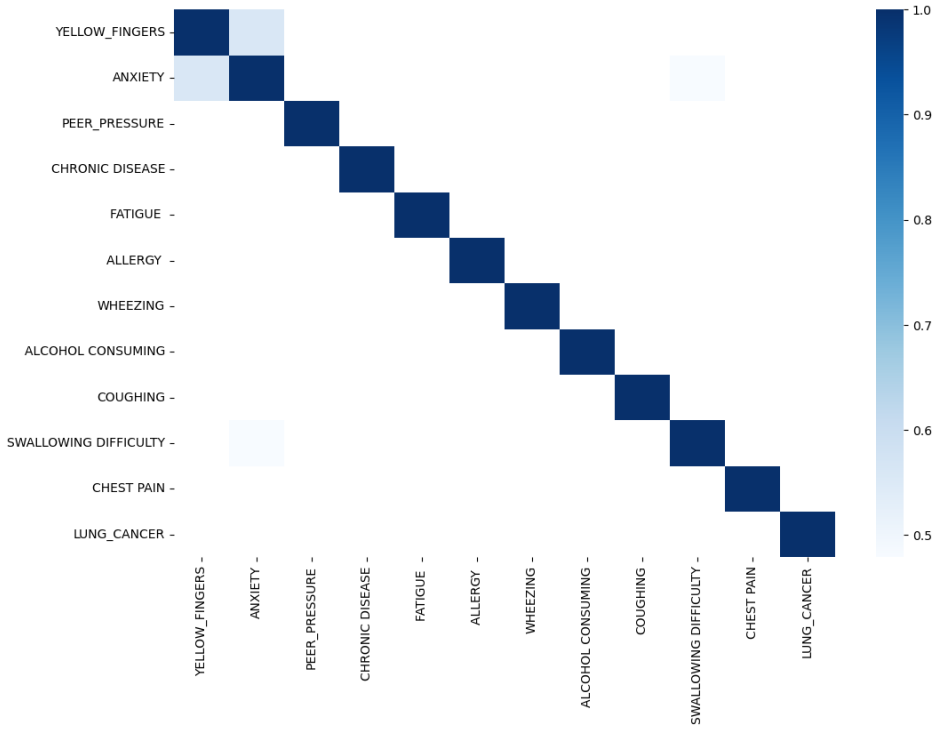

FIGURE 4.7 : The Heatmap of Strong Correlations.

4.5 Preprocessing

4.5.1 Importing Dependencies

The necessary libraries are imported, including sklearn, creating the feature : matrix X and the target variable y from the data dataframe.

4.5.2 Splitting the data :

The data is split into training and testing sets using the train_test_split function from scikit_learn. The feature matrix X and the target variable y are divided into X_train, X_test, y_train, and y_test, respectively. The test size is set to 0.25, which means 25% of the data will be used for testing, while the remaining 75% will be used for training the model.

4.5.3 Model training :

We trained multiple machine learning models, evaluated their performance individually. Let us start with logistic regression :

After splitting the data, we import Logistic Regression from sklearn.linear_model. This

logistic regression model is a commonly used algorithm for binary classification tasks. An instance of the logistic regression model, lr_model, is created with random_state=0 to maintain consistent results. The fit method is then applied to the training data (X_train and y_train), allowing the model to learn the underlying patterns and relationships in the data.

```
[38]: #Splitting data for training and testing
      from sklearn.model_selection import train_test_split
      X_train, X_test, y_train, y_test= train_test_split(X, y, test_size= 0.25, random_state=0)

[39]: #Fitting training data to the model
      from sklearn.linear_model import LogisticRegression
      lr_model=LogisticRegression(random_state=0)
      lr_model.fit(X_train, y_train)

[39]: LogisticRegression(random_state=0)
```

FIGURE 4.8 : Model training.

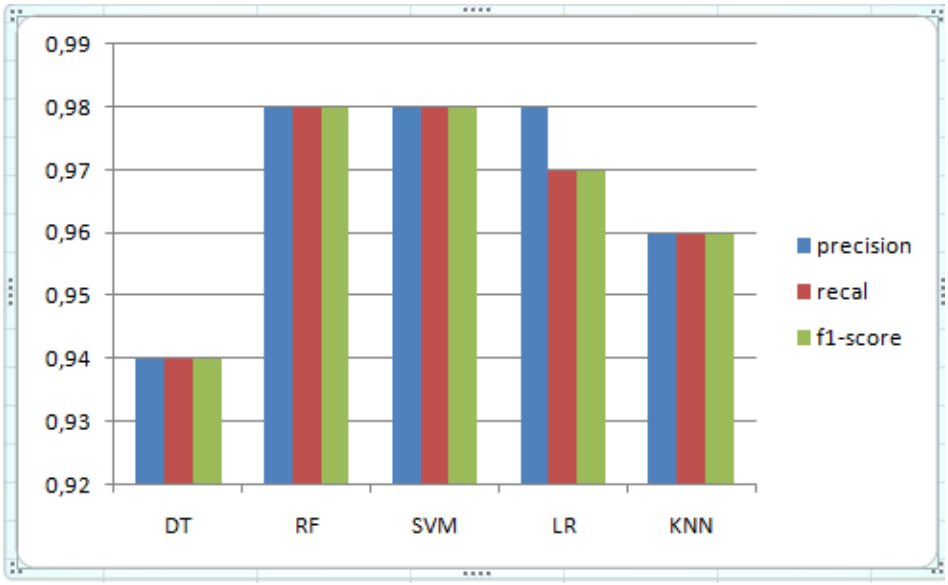

FIGURE 4.9 : Comparison of classification algorithms Performances

The figure 4.9 shows a bar chart comparing the performance metrics (precision, recall, and F1-score) of different machine learning algorithms : Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest Neighbors (KNN).

The RF and the SVM algorithm shows the highest overall performance across all three metrics, closely followed by Logistic Regression,Decision Tree has the lowest performance among the compared algorithms.

Figure 4.10 provides a comparative analysis of the accuracy of five different machine learning models.

To conclude, RF,DT,ANN and SVM have the highest accuracies, indicating strong performance on this dataset, while LR has the lowest accuracy, suggesting it may not be the best choice for this particular dataset.

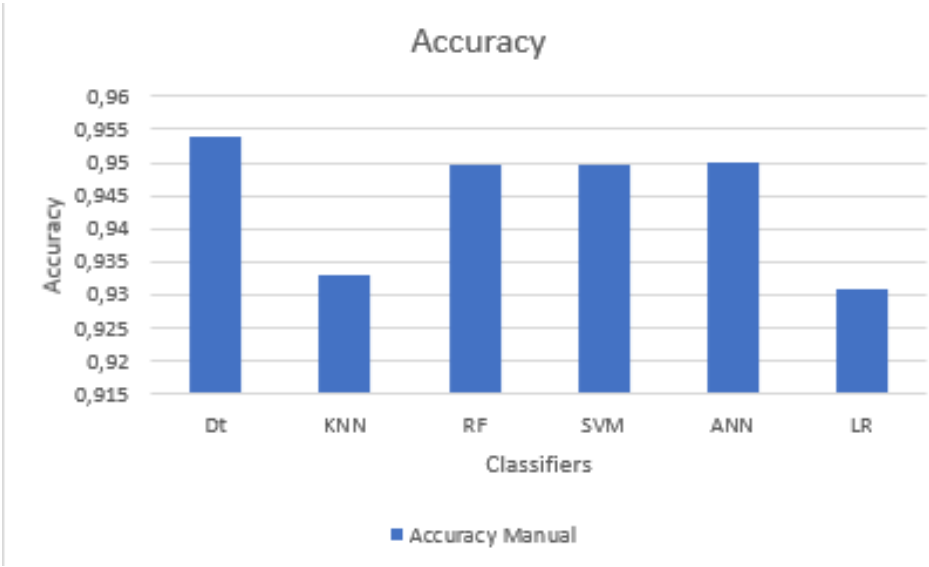

FIGURE 4.10 : Accuracy across different classification algorithms.

4.5.4 Automatic vs Manual

In analyzing the accuracy of different classifiers for Parkinson's Disease, Breast Cancer, and Lung Cancer datasets, we observe distinct trends in the performance of manual versus automatic classification methods.

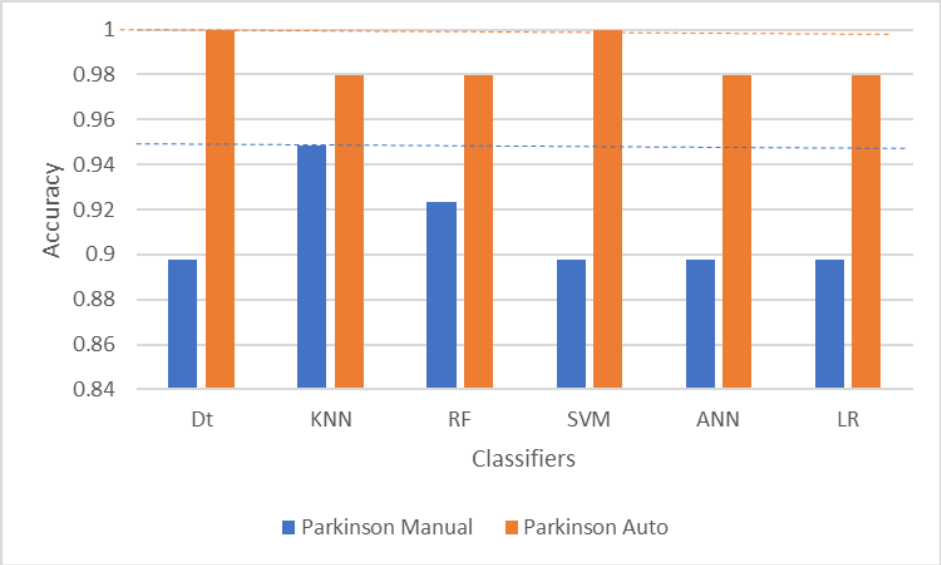

FIGURE 4.11 : The accuracy of parkinson Data.

For Parkinson's Disease The maximum accuracy achieved manually is 0.96 using the Random Forest (RF) classifier, whereas the automatic method reaches a peak accuracy of 0.99 across several classifiers, including Decision Tree (Dt), K-Nearest Neighbors (KNN), RF, Artificial Neural Networks (ANN), and Logistic Regression (LR). This reflects a percentage improvement of approximately 3.13

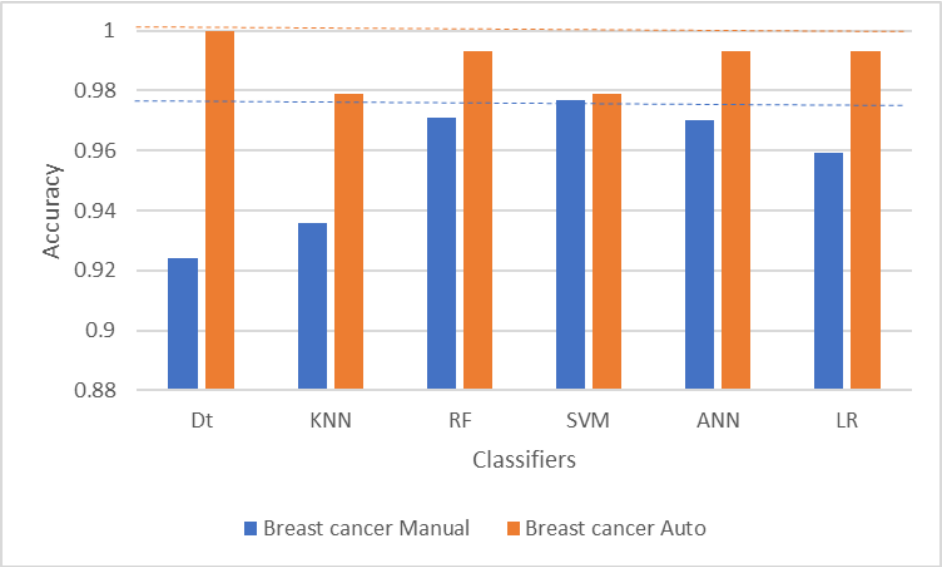

FIGURE 4.12 : The accuracy of breast cancer Data.

In the Breast Cancer dataset, the RF classifier also shows the highest manual accuracy at 0.97, while the Logistic Regression (LR) classifier stands out in the automatic method with an accuracy of 0.99. This corresponds to a percentage increase of around 2.06 For the

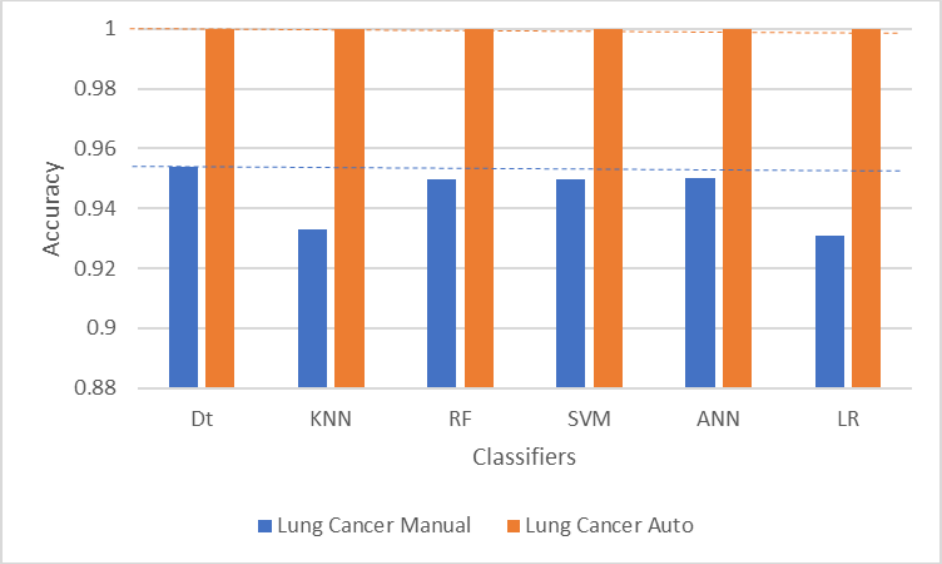

FIGURE 4.13 : The accuracy of lung cancer Data.

Lung Cancer dataset, the RF classifier again achieves the highest manual accuracy at 0.95. In contrast, the automatic method attains an accuracy of 0.99 using multiple classifiers, indicating a significant enhancement of approximately 4.21%.

Conclusion : In all cases (lung cancer, breast cancer, and Parkinson's disease), automatic classification methods consistently outperform manual methods across all classifiers, often achieving perfect accuracy. This demonstrates the effectiveness and potential superiority of automatic classification in medical diagnosis tasks.

4.5.5 GWO vs GEN

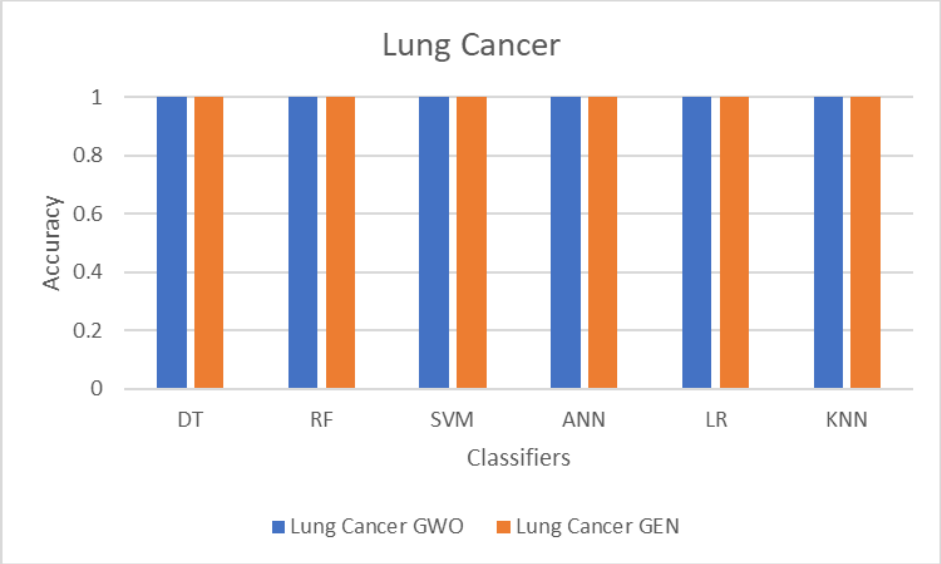

FIGURE 4.14 : The accuracy of lung cancer Data(GWO,GEN).

Lung cancer anaylsis : Both GEN and GWO perform almost identically across all classifiers, achieving near-perfect accuracy. This uniform performance suggests that both optimization algorithms are highly effective for the Lung Cancer dataset, and there is no significant advantage of one over the other.

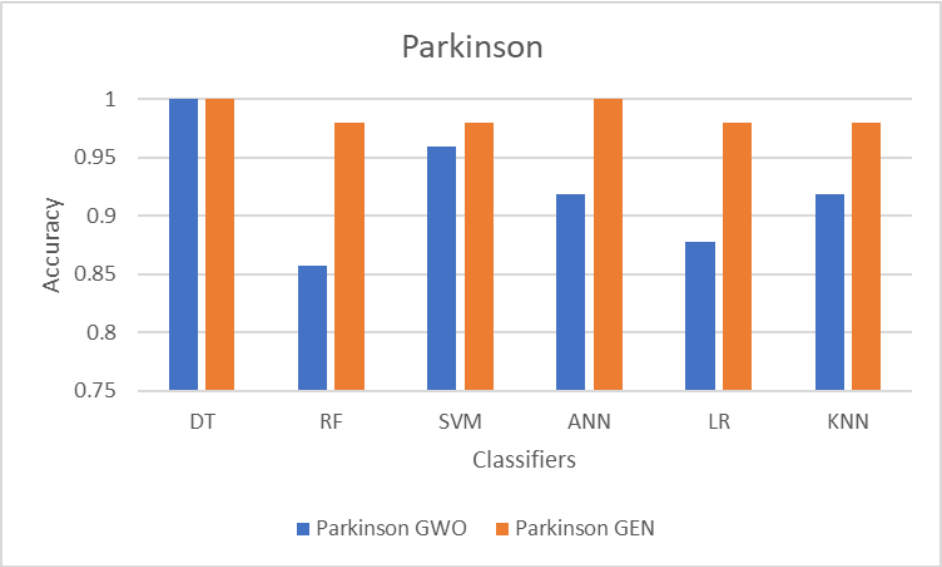

FIGURE 4.15 : The accuracy of parkinson Data(GWO,GEN).

Parkinson analysis : In the analysis of the Parkinson dataset, both GEN and GWO achieve perfect accuracy in Decision Tree (DT) classification. However, in other classifiers, GEN consistently outperforms GWO : Random Forest (RF) with an accuracy of 0.97 vs. 0.82, Support Vector Machine (SVM) with both above 0.95 but GEN slightly higher, Artificial Neural Network (ANN) with GEN close to 1 vs. GWO at 0.87, Logistic Regression (LR) at 0.97 vs. 0.85, and K-Nearest Neighbors (KNN) at 0.96 vs. 0.88. These results highlight GEN's superior performance across various classifiers.

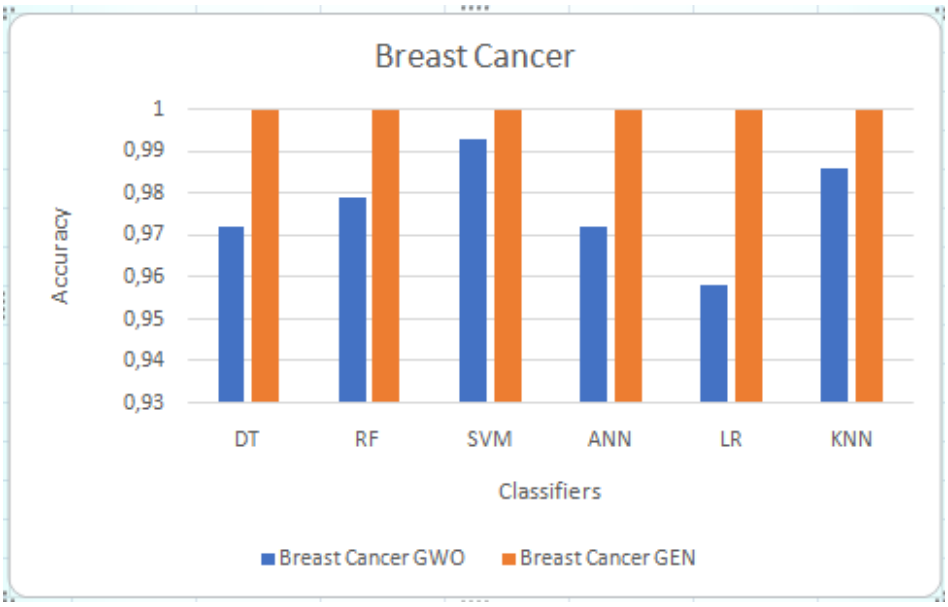

FIGURE 4.16 : The accuracy of breast cancer Data(GWO,GEN).

Although the Genetic Algorithm (GEN) provided the best overall accuracy with decision trees (DT), the Grey Wolf Optimizer (GWO) showed better performance in some cases.

In conclusion, the observed performance differences of GA and GWO across various machine learning algorithms highlight the nuanced strengths and weaknesses of each optimization approach. GA's capability to balance exploration and exploitation proved advantageous in optimizing hyperparameters for Decision Trees, Artificial Neural Networks, and k-Nearest Neighbors, likely due to its ability to efficiently explore complex solution spaces and converge towards diverse, high-quality solutions. Conversely, GWO, with its emphasis on exploration through hierarchical repositioning of search agents, may not always effectively exploit promising solutions once discovered, particularly in scenarios where precise parameter tuning is crucial, such as with Support Vector Machines and Random Forests. Therefore, the choice between GA and GWO for optimizing machine learning algorithms should consider the specific the complexity of the parameter space, and the desired trade-off between exploration and exploitation.

4.5.6 Convergence

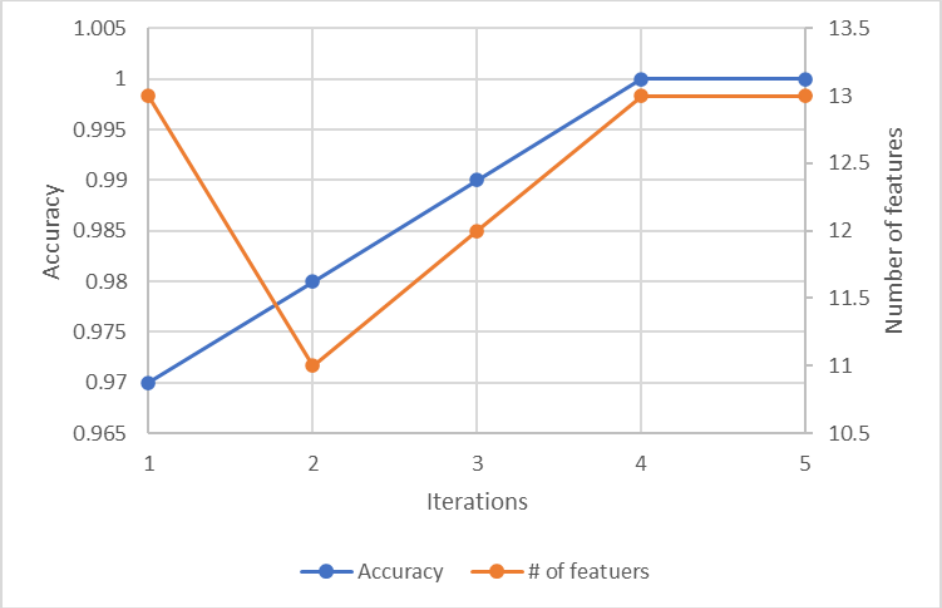

FIGURE 4.17 : The convergence.

The optimization process demonstrated remarkable efficiency by achieving near-optimal solutions within a minimal number of iterations, often three or fewer. The observed progression in accuracy from 0.97 to perfect scores of 1.0 highlights the optimizer's capability to swiftly converge towards higher performance. This efficiency is particularly noteworthy in scenarios where maximizing accuracy is critical, such as in medical datasets where errors can have significant implications. Therefore, the effectiveness of the optimizer in rapidly improving accuracy underscores its potential utility in real-world applications where quick and reliable optimization is essential.

Regarding the number of features, their count appeared random because the primary goal was to maximize accuracy. Therefore, any number of features that enhanced accuracy was deemed beneficial. The fitness function did not attempt to decrease the number of features either ; doing so could potentially lead to decreased accuracy. However, this approach may conflict with the nature of the current field, especially in medical datasets where errors can have serious consequences. Thus, compromising accuracy to reduce features could contradict the operational context.

4.6 The Result :

our study compared manual and automated optimization processes using GA and GWO. Automation with GA and GWO proved advantageous over manual tuning, offering efficiency and reproducibility in exploring hyperparameter spaces. GA effectively balanced exploration and exploitation, swiftly improving model accuracy from 0.97 to perfect scores of 1.0 in just a few iterations. GWO, proficient in exploring diverse solutions through hierarchical repositioning, occasionally lagged in exploiting promising solutions compared to GA.

Our automated tool demonstrated the ability to optimize, fine-tune, and select the best classifier for various datasets, significantly reducing hours of manual work and correlation checks. The primary focus on maximizing accuracy underscored the tool's capability to enhance model performance rapidly. However, further research is needed to optimize both accuracy and feature reduction simultaneously, crucial for improving model interpretability and efficiency in practical applications.

Chapitre 5

General Conclusion

Machine learning has proven to be an indispensable tool in a variety of domains, enabling automated decision-making and predictive modeling through data-driven insights. Among the array of techniques, classification stands out as a fundamental method that requires a robust set of features for the learning process. However, the complexity of improving the learning ability of classification algorithms escalates when dealing with datasets containing an extensive number of features. This complexity arises from the presence of redundant and irrelevant features, which not only complicate the performance of learning algorithms but also increase computation time.

In this thesis, we addressed the necessity of eliminating irrelevant features from datasets to achieve an effective learning process. We emphasized the importance of optimal feature selection techniques in enhancing classification accuracy and efficiency by reducing data dimensionality.

A notable aspect of our investigation was the integration of Meta-heuristic optimizers into the feature and classifier selection process. We explored the potential of these optimizers to enhance model performance through efficient navigation of the feature space. Our findings demonstrated that Meta-heuristic optimizers significantly contribute to the optimization of machine learning pipelines by effectively balancing feature selection, classifier selection, and hyperparameter tuning.

The key contributions of this thesis can be summarized as follows : a comparative analysis of different classifiers revealed insights into their performance across various metrics, guiding the selection of appropriate models for specific tasks ; the evaluation of feature selection methods underscored the importance of eliminating irrelevant features to improve classification accuracy and reduce computational load ; and the integration of Meta-heuristic optimizers showcased their efficacy in optimizing the entire machine learning pipeline, leading to better

model performance.

The future work will involve applying the proposed methods to a wider variety of datasets from different domains, such as healthcare, finance, and image recognition, will help validate their generalizability and effectiveness. Exploring domain-specific adaptations of feature selection and classifier optimization techniques will also be crucial in ensuring these methods' effectiveness across diverse applications.

Bibliographie

[1] Author's Name. What is machine learning ?, 2023. Accessed : 2024-06-14.

[2] N. Neha. A guide to supervised learning. https://medium.com/@ngneha090/a-guide-to-supervised-learning-f2ddf1018ee0, 2020. Accessed : June 14, 2024.

[3] Metehan Kozan. Supervised and unsupervised learning : An intuitive approach. https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644, 2020. Accessed : June 14, 2024.

[4] 365 Data Science. Decision trees : Machine learning tutorial. https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/, 2020. Accessed : June 14, 2024.

[5] Muhammad Yaseen Khan, Abdul Qayoom, Muhammad Nizami, Muhammad Shoaib Siddiqui, Shaukat Wasi, and Khaliq-Ur-Rahman Raazi Syed. Automated prediction of good dictionary examples (gdex) : A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques, 09 2021.

[6] JavaTpoint. Logistic regression in machine learning, 2024. Accessed : 30 May, 2024.

[7] JavaTpoint. K-nearest neighbor algorithm for machine learning, 2024. Accessed : 30 May, 2024.

[8] Simplilearn. Feature selection in machine learning, 2024. Accessed : 30 May, 2024.

[9] Seyed Mohammad Mirjalili, Shahrzad Saremi, Seyed Mohammad Mirjalili, and Leandro dos Santos Coelho. Multi-objective grey wolf optimizer : A novel algorithm for multi-criterion optimization. *Expert Syst. Appl.*, 47 :106–119, 2016.

[10] Scott Matthew. Introduction to genetic algorithms — including example code, 2019. Accessed : 2024-06-14.

[11] B Xue, L Cervante, L Shang, WN Browne, and M Zhang. Multi-objective evolutionary algorithms for filter based feature selection in classification. *Int J Artif Intell Tools*, 22(04) :1350024, 2013.

[12] G Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Trans Inf Theory*, 14(1) :55–63, 1968.

[13] MA Tahir, A Bouridane, and F Kurugollu. Simultaneous feature selection and feature weighting using hybrid tabu search/knearest neighbor classifier. *Pattern Recogn Lett*, 28(4) :438–446, 2007.

[14] dida.do. Supervised vs. unsupervised learning. https://dida.do/blog/supervised-vs-unsupervised-learning, 2020. Accessed : June 14, 2024.

[15] Lior Rokach and Oded Maimon. Decision trees. *Data mining and knowledge discovery handbook*, pages 165–192, 2005.

[16] Mia Huljanah, Zuherman Rustam, Suarsih Utama, and Titin Siswantining. Feature selection using random forest classifier for predicting prostate cancer. 546(5) :052031, 2019.

[17] D Jayaraj and S Sathiamoorthy. *Random forest based classification model for lung cancer prediction on computer tomography images*. 2019.

[18] Mohammad M Ghiasi and Sohrab Zendehboudi. Application of decision tree-based ensemble learning in the classification of breast cancer. *Computers in biology and medicine*, 128 :104089, 2021.

[19] Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M Ingersoll. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1) :3–14, 2002.

[20] Abdulsalam Alarabeyyat, Mohannad Alhanahnah, et al. *Breast cancer detection using k-nearest neighbor machine learning algorithm*. 2016.

[21] Ram MurtiRawat, Shivam Panchal, Vivek Kumar Singh, and Yash Panchal. *Breast Cancer detection using K-nearest neighbors, logistic regression and ensemble learning*. 2020.

[22] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines : Theory and applications. In *Advanced course on artificial intelligence*, pages 249–257. Springer, 1999.

[23] Nikolay Chumerin and Marc M Van Hulle. *Comparison of two feature extraction methods based on maximization of mutual information*. 2006.

[24] KDnuggets. Feature engineering explained. decembre 2018.

[25] Lalitha Rangarajan et al. Bi-level dimensionality reduction methods using feature selection and feature extraction. *International Journal of Computer Applications*, 4(2) :33–38, 2010.

[26] L. Yu and H. Liu. *Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution*. 2010.

[27] Harshil Patel. *Feature Engineering Explained : Crafting Data Inputs for Machine Learning*. 2024.

[28] Ambika Kaul, Saket Maheshwary, and Vikram Pudi. Autolearn—automated feature generation and selection. In *2017 IEEE International Conference on data mining (ICDM)*, pages 217–226. IEEE, 2017.

[29] J Schmidhuber. Deep learning in neural networks : an overview. *Neural Netw*, 61 :85–117, 2015.

[30] T Kavzoglu and PM Mather. The use of feature selection techniques in the context of artificial neural networks. In *Proceedings of the 26th annual conference of the remote sensing society*, 2000.

[31] S Galeshchuk. Neural networks performance in exchange rate prediction. *Neurocomputing*, 172 :446–452, 2016.

[32] Y Zhou, G Cheng, S Jiang, and M Dai. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput Networks*, 174 :107247, 2020.

[33] J Weston, S Mukherjee, O Chapelle, M Pontil, T Poggio, and V Vapnik. Feature selection for svms. In *Adv Neural Inf Process Syst*, volume 13, pages 668–674, 2000.

[34] HC Peng, C Ding, and FH Long. Minimum redundancy-maximum relevance feature selection. pages 70–71, 2005.

[35] JD Jr Kelly and L Davis. A hybrid genetic algorithm for classification. In *IJCAI*, pages 645–650, 1991.

[36] M Dialameh and MZ Jahromi. A general feature-weighting function for classification problems. *Expert Syst Appl*, 72 :177–188, 2017.

[37] D Wettschereck, DW Aha, and T Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artif Intell Rev*, 11(1–5) :273–314, 1997.

[38] ML Raymer, WF Punch, ED Goodman, LA Kuhn, and AK Jain. Dimensionality reduction using genetic algorithms. *IEEE Trans Evol Comput*, 4(2) :164–171, 2000.

[39] D Singh and B Singh. Hybridization of feature selection and feature weighting for high dimensional data. *Appl Intell*, 49(4) :1580–1596, 2019.

[40] E Alba and B Dorronsoro. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans Evol Comput*, 9(2) :126–142, 2005.

[41] M Lozano and C García-Martínez. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification : overview and progress report. *Comput Oper Res*, 37(3) :481–497, 2010.

[42] H Shayeghi, A Ghasemi, M Moradzadeh, and M Nooshyar. Day-ahead electricity price forecasting using wpt, gmi and modified lssvm-based s-olabc algorithm. *Soft Comput*, 21(2) :525–541, 2017.

[43] DH Wolpert and WG Macready. No free lunch theorems for optimization. *IEEE Trans Evol Comput*, 1(1) :67–82, 1997.

[44] XF Song, Y Zhang, DW Gong, and XZ Gao. A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Trans Cybern*, 99 :1–14, 2021.

[45] XF Song, Y Zhang, DW Gong, and XY Sun. Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recogn*, 112 :107804, 2021.

[46] Xuefeng Song, Yang Zhang, Yunan Guo, Xiaoyan Sun, and Yulei Wang. Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Transactions on Evolutionary Computation*, 24(5) :882–895, 2020.

[47] Yun Hu, Yang Zhang, and Dunwei Gong. Multiobjective particle swarm optimization for feature selection with fuzzy cost. *IEEE Transactions on Cybernetics*, 51(2) :874–888, 2020.

[48] Ahmed Darwish. Bio-inspired computing : algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal*, 3(2) :231–246, 2018.

[49] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[50] Mohammad Yazdani and Fariborz Jolai. Lion optimization algorithm (loa) : a nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*, 3(1) :24–36, 2016.

[51] Erik Cuevas, Manuel Cienfuegos, Raúl Rojas, and Arturo Padilla. A computational intelligence optimization algorithm based on the behavior of the social-spider. In *Computational Intelligence Applications in Modeling and Control*, pages 123–146. Springer, 2015.

[52] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95 :51–67, 2016.

[53] Xuefeng Song, Lichun Tang, Shuai Zhao, Xinyi Zhang, Liang Li, Jia Huang, and Wen Cai. Grey wolf optimizer for parameter estimation in surface waves. *Soil Dynamics and Earthquake Engineering*, 75 :147–157, 2015.

[54] Emary Emary, Hossam M Zawbaa, and Aboul Ella Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172 :371–381, 2016.

[55] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[56] Ricardo Santos Parpinelli, Heitor Silvério Lopes, and Alex Alves Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4) :321–332, 2002.

[57] Ioannis Michelakos, Nikos Mallios, Elpiniki Papageorgiou, and Michael Vassilakopoulos. Ant colony optimization and data mining. In *Next Generation Data Technologies for Collective Computational Intelligence*, pages 31–60. Springer, 2011.

[58] Deepak Singh and Balwinder Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97 :105524, 2020.

[59] DataCamp. What is kaggle ?, 2023. Accessed : 2024-06-10.

[60] Python Software Foundation. Python, 2023. Accessed : 2024-06-10.

[61] Scikit-learn. About scikit-learn. https://scikit-learn.org/stable/about.html, 2021. Accessed : September 2021.

[62] Matplotlib. About matplotlib. https://matplotlib.org/stable/users/index.html, 2021. Accessed : September 2021.

[63] Pandas. About pandas. https://pandas.pydata.org/about/index.html, 2021. Accessed : September 2021.

[64] A Dey. Machine learning algorithms : a review. *Int J Com put Sci Inf Technol*, 7(3) :1174–1179, 2016.

[65] J Perez-Rodriguez, AG Arroyo-Pena, and N Garcia-Pedrajas. Simultaneous instance and feature selection and weighting using evolutionary computation : Proposal and study. *Appl Soft Comput*, 37 :416–443, 2015.

[66] Alaa Tharwat and Aboul Ella Hassanien. Chaotic antlion algorithm for parameter optimization of support vector machine. *Applied Intelligence*, 48(3) :670–686, 2018.

[67] Mahdi Khishe and Mohammad Reza Mosavi. Chimp optimization algorithm. *Expert Systems with Applications*, 149 :113338, 2020.

[68] Iain D Couzin and Mark E Laidre. Fission-fusion populations. *Current Biology*, 19(15) :R633–R635, 2009.

[69] Christophe Boesch. Cooperative hunting roles among tai chimpanzees. *Human Nature*, 13(1) :27–46, 2002.

[70] Craig B Stanford, Janette Wallis, Ekwoge Mpongo, and Jane Goodall. Hunting decisions in wild chimpanzees. *Behaviour*, 131(1-2) :1–18, 1994.

[71] Mohammad Reza Mosavi, Mahdi Khishe, and Mostafa Akbarisani. Neural network trained by biogeography-based optimizer with chaos for sonar data set classification. *Wireless Personal Communications*, 95(4) :4623–4642, 2017.

[72] J.J. Berrill. *The Tuniccafa.* The Royal Society, London, 1950.

[73] John Davenport and George H Balazs. Fiery bodies—are pyrosomas an important component of the diet of leatherback turtles ? *Biology*, 37 :33–38, 1991.

[74] Amir Ghasemi-Marzbali. A novel nature-inspired meta-heuristic algorithm for optimization : bear smell search algorithm. *Soft Computing*, 24(2) :1–33, 2020.