# Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the Financial Markets

João Pedro Pinto Brito Nobre,
joaonoobre@gmail.com,
Instituto Superior Técnico

***Abstract*—This paper presents an approach combining Principal Component Analysis (PCA), Discrete Wavelet Transform (DWT), Extreme Gradient Boosting (XGBoost) and a Multi-Objective Optimization Genetic Algorithm (MOO-GA) to create a system that is capable of achieving high returns with a low level of risk associated to the trades. PCA is used to reduce the dimensionality of the financial input data set while maintaining the most valuable parts of each feature and the DWT is used to perform a noise reduction to every feature while keeping its structure. The resultant data set is then fed to an XGBoost binary classifier that has its hyperparameters optimized by a MOO-GA in order to achieve, for every analyzed financial market, the best performance. The proposed approach is tested with real financial data from four different financial markets, each with its own characteristics and behavior. The importance of the PCA is analyzed and the results obtained show that it greatly improves the performance of the system. In order to improve even more the results obtained in the system using PCA, the PCA and the DWT are then applied together in one system and the results obtained show that this system is capable of outperforming the Buy and Hold (B&H) strategy in three of the four analyzed financial markets, achieving an average rate of return of 43.15% in the portfolio, while the B&H achieves on average 29.02%.**

***Index Terms*—Financial Markets, Principal Component Analysis (PCA), Discrete Wavelet Transform (DWT), Extreme Gradient Boosting (XGBoost), Multi-Objective Optimization Genetic Algorithm (MOO-GA), Dimensionality Reduction, Noise Reduction.**

## I. Introduction

The continuous evolution in the Machine Learning and Artificial Intelligence areas and the fact that the financial markets information is becoming more accessible to a larger number of investors results in the appearance of sophisticated trading algorithms that consequently are starting to have a significant influence on the market behaviour.

According to the Efficient Market Hypothesis (EMH) [1], a stock market time series is nearly unforecastable and that is because it's impossible to beat the market since the share prices already have all the relevant available information into account including the past prices and trading volumes. As such, price fluctuations respond immediately to new information and don't follow any pattern, being unpredictable and stopping investors from earning above average returns without taking many risks. This hypothesis consequently implies that a binary classifier that tries to identify if the difference between the price of a stock in a day with the price of the same stock in the day after is positive or negative wouldn't perform better than random guessing since the market price will always be the fair one, and therefore unpredictable.

In this paper, an approach combining Principal Component Analysis (PCA) for dimensionality reduction, the Discrete Wavelet Transform (DWT) for noise reduction and an XGBoost (Extreme Gradient Boosting) binary classifier whose hyperparameters are optimized using a Multi-Objective Optimization Genetic Algorithm (MOO-GA), is presented. Using PCA the high dimensional financial input data set is reduced to a lower dimensional one, maximizing the variance in the lower dimensional space and while keeping the main essence of the original data set. This dimensionality reduction allows for a better identification of patterns in the training data that consecutively results in a better generalization ability and an higher accuracy by the XGBoost binary classifier. The DWT further performs noise reduction to this reduced data set in order to remove irrelevant data samples that may have a negative impact in the performance of the system, while still preserving the main structure of the data. The XGBoost binary classifier has its hyperparameters optimized using the MOO-GA in order to achieve the best performance for each analyzed financial market. Then, the classifier is trained using the set of hyperparameters obtained through the optimization process and, using the predictions made, it outputs a trading signal with the buy and sell orders, with the objective of maximizing the returns, while minimizing the levels of risk associated to the trades made.

This paper is organized as follows: Section II addresses the theory behind the developed work. Section III presents the architecture of the solution with an explanation of each of its components. Section IV presents the case studies and the analysis of the results. Section V summarizes the paper's content and supplies its conclusion.

## II. Related Work

To succeed in the modern stock market, one has to build an algorithm that, with low risk, is able to achieve high returns. An ideal intelligent algorithm would predict stock prices and help the investor buy stocks before its price rises and sell before its price falls. Since it's very difficult to forecast with precision if a stock's price will rise or decline due to the noisy, non-linear and non-stationary properties of a stock

market time series, appropriate data pre-processing techniques and optimization algorithms are required in order to increase the accuracy of the system. In order to do so, methods like Technical analysis and Machine Learning are being used in an attempt to achieve good results.

Technical analysis [2] refers to the study of past price movements in order to try to predict its future behaviour and it is the only type of analysis that will be applied in this paper.

The utilization of Machine Learning methods in financial markets is done in an attempt to develop algorithms capable of learning from historical financial data and other information that might affect the market and make predictions based on these inputs in order to try to maximize the returns. Machine Learning algorithms can process data at a much larger scale and with much larger complexity, discovering relationships between features that may be incomprehensible to humans, this way achieving good results. Therefore, by exploiting the relationships between the input data, consisting of historical raw financial data as well as technical indicators, and learning from it, these models make predictions about the behaviour of a stock price that can be used in order to create a trading strategy capable of obtaining high returns.

### *A. Principal Component Analysis (PCA)*

Having a data set of large dimensions can often be a problem due to the fact that it may lead to higher computational costs and to overfitting. Therefore one may want to reduce the data set dimension in order to make the data manipulation easier and lower the required computational resources, improving the performance of the system and while keeping as much information as possible from the original data.

Principal Component Analysis (PCA) is one of the simplest and most used dimensionality reduction methods and can be used to reduce a data set with a large number of dimensions to a small data set that still contains most of the information of the original data set. This is done by transforming the original variables to a new set of uncorrelated variables, known as principal components, ordered such that the retention of variance present in the original variables decreases as the order of the principal component decreases. In this way, this means that the first principal component retains the maximum variance that was present in the original data set. By performing this transformation, a low dimensional representation of the original data is achieved while keeping its maximal variance.

The purpose of the PCA is to make a projection from the main components of an high-dimensional data set onto a lower dimensional space, without changing the data structure, and obtaining a set of principal components that are a linear combination of the features present in the original data set that reflect its information as much as possible. The goal is to retain the dimensions with high variances and remove those with little changes in order to reduce the required computational resources [3], which results in a set of principal components that has the same dimension of the original data set or lower in the case of performing dimensionality reduction since only the principal components that retain most of the original data set variance will be retained.

### *B. Discrete Wavelet Transform (DWT)*

Fourier transform based spectral analysis is the most used tool for an analysis in the frequency domain. According to Fourier theory, a signal can be expressed as the sum of a series of sines and cosines. However, a serious limitation of the Fourier transform is that it cannot provide any information of the spectrum changes with respect to time. The wavelet transform is similar to the Fourier transform but with a different merit function. The main difference is that instead of decomposing the signal into sines and cosines, the wavelet transform uses functions that are localized in both time and frequency. The basic idea of the wavelet transform is to represent any function as a superposition of a set of wavelets that constitute the basis function for the wavelet transform. Wavelets are small waves located in different times and can be stretched and shifted to capture features that are local in time and local in frequency, therefore the wavelet transform can provide information about both the time and frequency domains. The wavelets are scaled and translated copies, known as the daughter wavelets, of a finite-length oscillating waveform known as the mother wavelet. The selection of the best wavelet basis depends on the characteristics of the original signal to be analyzed [4] and the desired analysis objective. In the end, the result will be a set of time-frequency representations of the original signal, all with different resolutions, this is why the wavelet transform can be referred to as a multi-resolution analysis. There are two types of wavelet transform, the Continuous Wavelet Transform (CWT) and the Discrete Wavelet Transform (DWT), in the system proposed in this paper only the DWT is used.

The DWT decomposes the signal into a set of wavelets that is mutually orthogonal. The discrete wavelet is related to the mother wavelet [5] as presented in Equation 1, where the parameter $m$ is an integer that controls the wavelet dilation, the parameter $k$ is an integer that controls the wavelet translation, $s_0$ is a fixed scaling parameter set at a value greater than 1, $\tau_0$ is the translation parameter which has to be greater than zero and $\psi$ is the mother wavelet.

$$\psi_{m,k}(t) = \frac{1}{\sqrt{s_0^m}} \psi \left( \frac{t - k\tau_0 s_0^m}{s_0^m} \right) \tag{1}$$

An algorithm to calculate the DWT was developed by Mallat [6], using a process that is equivalent to high-pass and low-pass filtering in order to obtain the detail and approximation coefficients, respectively, from the original signal. Each iteration of this process is called a level of decomposition and produces one approximation coefficient and $j$ detail coefficients, with $j$ being the chosen decomposition level. Figure 1 represents the tree structure of the DWT decomposition process of a signal to determine the approximation and detail coefficients for a decomposition level of 2, where $x[n]$ represents the original input signal and $h[n]$ and $g[n]$ represent an high-pass and a low-pass filter, respectively.

The result of the DWT is a multilevel decomposition with the signal being decomposed in approximation and detail coefficients at each level of decomposition. The DWT can provide a perfect reconstruction of the signal after inversion, i.e by

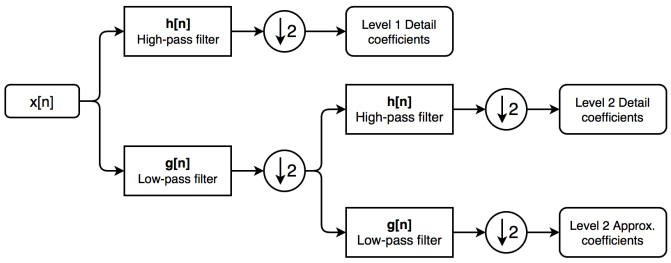


Fig. 1. DWT decomposition of a signal for a decomposition level of 2.

performing the DWT of a signal and then use the obtained coefficients in the reconstruction phase, in ideal conditions the original signal can be again obtained. The reconstruction phase is the reversed process of the decomposition, done by performing the inverse discrete wavelet transform using the same wavelet basis that was used in the decomposition phase. Figure 2 represents the DWT reconstruction process of a signal for a decomposition level of 2, where $x'[n]$ represents the reconstructed signal.

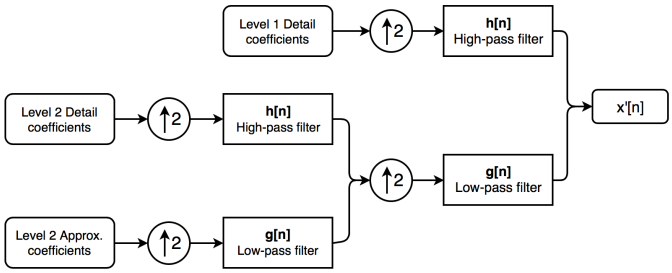


Fig. 2. DWT reconstruction of a signal for a decomposition level of 2.

However, one of the main utilities of the DWT is its capability to reduce the noise of a noisy signal. Supposing the given data is in the form $y(n) = x(n) + e(n)$, where $y(n)$ is the observed data, $x(n)$ is the original data and $e(n)$ is Gaussian white noise with zero mean and variance $\sigma^2$, the main objective of denoising the data is to reduce the noise as much as possible and recover the original data $x(n)$ with as little loss of important information as possible. The important features of many signals are captured by a subset of DWT coefficients that is typically much smaller than the original signal itself. By thresholding all the coefficients, one can choose the subset of relevant coefficients that will be kept after the DWT. Donoho *et. al* [7] propose the shrinkage process by thresholding the coefficients, in which a hard or a soft threshold value must be chosen. When performing hard thresholding, all the coefficients smaller than the threshold value are set to zero while if soft thresholding is used, the values for both positive and negative coefficients are shrinked towards zero. Although there is not much theory about which thresholding method is better, when using soft thresholding the results are smoother and the noise is almost fully suppressed [7] in comparison to the hard thresholding, in this paper soft threshold will be applied. To determine the threshold values, the BayesShrink method [8] is used, in which an unique threshold is estimated for each wavelet subband.

After that process the result will be a new set of DWT coefficients already thresholded, used in the inverse DWT to reconstruct the original signal. By discarding these irrelevant coefficients and only using the coefficients that capture the important features of the original signal in the reconstruction phase, the result will be a denoised signal with the main features of the original signal.

### C. Extreme Gradient Boosting (XGBoost)

In Machine Learning, boosting [9] is an ensemble technique that attempts to create a strong learner from a given number of weak learners, i.e models that only perform slightly better than random guessing. An ensemble is a set of predictors, all trying to predict the same target variable, which are combined together in order to give a final prediction. Using ensemble methods allows for a better predictive performance compared to a single predictor alone by helping in reducing the bias and variance in the predictions [10]. The main principle of boosting is to iteratively fit a sequence of weak learners to weighted versions of the training data. After each iteration, more weight is given to training samples that were misclassified by earlier rounds. In the end of the process, all of the successive models are weighted according to their performance and the outputs are combined using voting for classification problems or averaging for regression problems, creating the final model.

XGBoost [11], short for Extreme Gradient Boosting, is a machine learning system based on Friedman's [12] Gradient Boosting. Although XGBoost is based on the original Gradient Boosting model [12], there are some improvements to the original model that help increase the performance. XGBoost uses a tree ensemble model which is a set of classification and regression trees (CART) [13]. This type of boosting, using trees as base learners, is called Tree Boosting. Because one tree might not be enough to obtain good results, multiple CARTs can be used together and the final prediction is the sum of each CART's score. The model can be written as Equation 2:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}, \tag{2}$$

where $f$ is a function in the functional space $\mathcal{F}$, with $\mathcal{F} = \{f(x) = w_{q(x)}\}(q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ being the set of all possible CARTs, where $q$ represents the structure of each tree that maps an example to the corresponding leaf index, $T$ is the number of leaves in the tree, $w$ is the leaf weight and $K$ represents the number of trees. The objective function to optimize becomes the one represented in Equation 3, trained in an additive way by adding the $f_t$ that helps in the minimization of the objective, where $\hat{y}_i^{(t-1)}$ represents the prediction of the instance $i$ at iteration $t-1$, $l(y_i, \hat{y}_i^{(t-1)})$ is the training loss function and $\Omega$ is the regularization term.

$$\mathcal{L}^{(t)} = \sum_{i}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{3}$$

The regularization term is calculated using Equation 4 and is used to control the variance of the fit in order to control the flexibility of the learning task and to obtain models that

generalize better to unseen data. Controlling the complexity of the model is useful in order to avoid overfitting the training data.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{4}$$

While the original Gradient Boosting model adds the weak learners one after another, XGBoost does it in a parallel way similar to the Random Forest method that grows trees parallel to each other, i.e XGBoost builds the tree itself in a parallel way using all of the computer's CPU cores during the training, resulting in a greater computational speed.

### D. Genetic Algorithm

A genetic algorithm (GA) is a meta-heuristic algorithm for optimization that belongs to the area of Evolutionary Computation, inspired by natural processes like evolution, survival and adaptation to the environment. The GA [14] is used to effectively search complex spaces for an optimal solution to a given optimization problem. A GA is composed by a set of individuals (the different candidate solutions) called population. Each individual, called chromosome, is evaluated according to a fitness function that reflects the performance of the individual, i.e its fitness in the given optimization task. Each parameter to be optimized on the chromosome is called the gene.

The first step in the GA is to create an initial random population. After all the chromosomes have their genes defined, the initial population is created and go through the evaluation process where each chromosome of the population is evaluated and a fitness value is assigned according to its performance at the given optimization task. Then some of the individuals are selected for reproduction, with the individuals with an highest fitness being more likely to be chosen. Next, the crossover operator is applied where the selected individuals are combined creating an offspring with some features of each parent, i.e the offspring's genes are a combination of its parents genes. Finally, the mutation operator is applied and it randomly changes the genes of a chromosome in a probabilistic way in order to employ genetic diversity to the population. These processes are repeated in each generation. When the termination conditions are achieved it means that the GA converged towards a stable solution.

*1) Multi-Objective Optimization:* Although genetic algorithms are more used regarding single-objective optimization (SOO) problems, they also are of extremely use in multi-objective optimization problems when there is more than one objective and the objectives are of conflict to each other. Most of the real world optimization problems involve more than one objective to be optimized, therefore the purpose in using a MOO approach is to find the solution that reflects a compromise between all objectives [15]. What makes MOO problems more complex to solve than SOO problems is that there is no unique solution, but a set of different optimal solutions where each solution represents a trade-off between the different objectives, this set is called the Pareto front. In multi-objective optimization, the candidate solutions are compared using dominance relationship. A solution $x$ dominates solution $y$, if the solution $x$ is no worse than solution $y$ in all objectives and if the solution $x$ is better than $y$ at least on one objective [16]. The non-dominated solutions are considered the fittest among all the solutions and are called Pareto-optimal solutions or non-dominated set of solutions.

In this paper for solving the multi-objective optimization problem proposed, namely the XGBoost hyperparameters' optimization, the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) was chosen [17]. Unlike the SOO method, the NSGA-II algorithm optimizes each objective simultaneously without being dominated by any other solution. It is a fast multi-objective optimization algorithm that uses an elitism principle and is proven to find a much better set of solutions, as well as better convergence near the Pareto-optimal front compared to other evolutionary algorithms, as shown by Deb *et. al* [17].

## III. Proposed Approach

### A. System Architecture

This paper presents a system that is capable of detecting the best entry and exit points in the market in order to maximize the returns in financial markets while minimizing the risk. For that purpose, a system using PCA, DWT, an XGBoost binary classifier and a MOO-GA is proposed. Figure 3 presents the architecture of the system. The system was implemented in Python programming language.

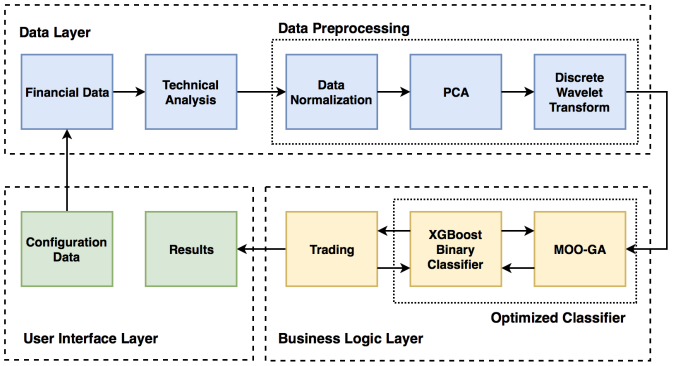


Fig. 3. Architecture of the proposed system.

### B. Target Formulation

In this paper, the objective is to predict whether the close price for day $t+1$, $Close_{t+1}$, will have a positive or a negative variation with respect to the close price in the current day $t$, $Close_t$. As such, a supervised learning solution is proposed, more specifically a binary classifier. The target variable to be predicted, $y$, is the signal of the variation in the close price for day $t+1$ with respect to the close price in day $t$ and follows a binomial probability distribution $y \in \{0,1\}$, where it takes the value 1 if the variation in close price was positive and the value 0 if the variation in close price was negative. This target can be mathematically defined as presented in Equation 5.

$$y_t = \begin{cases} 1 & \text{if } \frac{Close_{t+1} - Close_t}{Close_t} \geq 0 \\ 0 & \text{if } \frac{Close_{t+1} - Close_t}{Close_t} < 0 \end{cases} \tag{5}$$

### C. Technical Analysis Module

The technical analysis module receives as input the raw financial data from the financial data module and applies several technical indicators to it. The main purpose of using technical indicators is that each one provides basic information about past raw financial data in a way different from each other and thus combining different technical indicators together helps in the detection of patterns in the financial data, this way increasing the performance of the predictive system. This module will create the data set that will be used as input to the data preprocessing module. This data set consists of the combination between the set of 26 technical indicators used and 5 raw financial data features, resulting in a data set with 31 features presented in Table I.

TABLE I
LIST OF THE 31 FEATURES OUTPUT TO THE DATA PREPROCESSING MODULE.

| Technical Indicators | Raw Financial Data |
|---|---|
| RSI | Open |
| MACD and Signal Line | High |
| MACD Histogram | Low |
| PPO | Adj. Close |
| ADX | Volume |
| Momentum | |
| CCI | |
| ROC | |
| Stochastic %D and %K | |
| Williams %R | |
| SMA20, SMA50, SMA100 | |
| EMA20, EMA50, EMA100 | |
| Bollinger Bands | |
| PSAR | |
| OBV | |
| Chaikin Oscillator | |
| MFI | |
| ATR | |

### D. Data Preprocessing

When dealing with real-word data, there is always the chance that it is imperfect which can lead to misleading results by the machine learning system. Therefore, improving the overall quality of the data will consequently improve the results [18]. In order to do so, the raw data fed to the system must be preprocessed. In this paper, the main stages of the data preprocessing are data normalization, PCA and DWT.

*1) Data Normalization Module:* The data set is first divided into training set, validation set and test set. The training set will be used to train the model, the validation set will be used to tune the hyperparameters in order to achieve a good generalization during the training phase and to avoid overfitting the training data and the test set will be used as out-of-sample data in order to test the performance of the final model in unseen data. These data sets are then normalized using the Min-Max normalization technique presented in Equation 6, which will rescale every feature in the data sets to the range of [0,1].

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6}$$

*2) PCA Module:* The PCA module receives the normalized input data set, containing the 26 technical indicators and the 5 raw financial data features totaling 31 normalized features and, in order to reduce the risk of overfitting the data and to lower the computational costs of the system, the PCA will transform the data set with 31 features in a lower dimensional one, while still retaining most of the original data set variance. The PCA first fits its model with the normalized training set in order to determine the components that represent the directions of maximum variance. Then, the principal components are ordered by the amount of variance they explain and only the ones that add up to at least 95% of the variance of the original training set are retained. Finally, the data is projected onto the principal components, resulting in a data set that has a lower dimension than the original data set since it only retains the data samples that better explain the relations between the features. The reduced data set is then fed to the Wavelet module.

*3) Wavelet Module:* Although the data set was already simplified in the PCA module, having its dimension reduced and only retaining the data that better explain the relations between the features, obtaining a more compact representation of the original data set, some irrelevant data samples that may have a negative impact in the training and forecasting performance of the system may still exist. While the PCA technique removed irrelevant data points in the feature subset, the DWT technique will perform a noise reduction in the time domain in each of the features present in the data set reduced by the PCA. This process reduces the influence of the noise in the data set while retaining the important components of each feature as much as possible.

The wavelet basis tested in this system for each financial market are:

- Haar wavelet;
- Daubechies wavelet of order: 3, 6, 9, 15 and 20;
- Symlet wavelet of order: 3, 6, 9, 15 and 20.

Although with more decomposition levels more noise can be removed and therefore resulting in a better representation of the trend of each feature, this could also result in removing fluctuations that carry market characteristics. Therefore in this system, the levels of decomposition tested are 2, 5 and 7, in order to find the optimal denoising level of decomposition for each financial market.

The DWT starts by specifying the wavelet basis, order and the level of decomposition used. Then, for each of the training set's features, the DWT performs the multilevel decomposition which will result in one approximation coefficient and $j$ detail coefficients, with $j$ being the chosen decomposition level. In order to calculate the approximation and detail coefficients for both the validation and test sets, one data point at a time is added to the training set, the coefficients for the new signal are calculated and the coefficients corresponding to the data point added are saved, in order to avoid the coefficients of having future information into account. This procedure is performed until all the points in the validation and test sets have their respective coefficients calculated. Then thresholding is applied to the obtained detail coefficients and the signal is

reconstructed, resulting in a denoised version of each of the original data set's features. After the DWT is applied, the data set is fed to the XGBoost module.

### *E. XGBoost Module*

*1) XGBoost Binary Classifier:* The XGBoost binary classifier is the responsible for the classification process of the system. The output of the classifier, $\hat{y}_t$, is the predicted value given the current observation, $x_t$, corresponding to the actual day, $t$. The variable $\hat{y}_t$ is in the range [0,1] and the set of all $\hat{y}_t$ corresponds to the predicted trading signal which indicates if the system should take a Long or Short position in the next trading day.

Before the XGBoost binary classifier algorithm starts, a set of parameters must be chosen. The parameters which define a Machine Learning system architecture are referred to as hyperparameters. Because every time series has its own characteristics, for every time series analyzed there is a different set of optimal hyperparameters, i.e hyperparameters that allow for the model to have a good generalization capacity and therefore achieve good results in out-of-sample data. Therefore, in order to achieve the best results in each of the analyzed financial markets, the optimal set of hyperparameters has to be found. In order to do so, the MOO-GA approach is used, as presented in the next section.

The preprocessed data output by the data preprocessing module is fed to the XGBoost binary classifier, as well as the target variable array, $Y$, to be predicted and both are divided into training, validation and test sets. With both the data sets and the XGBoost hyperparameters defined, the training phase starts. The generalization ability of the system is measured by how well the system performs with unseen data. Thus, after the training phase is finished, the out-of-sample validation set is used in order to test the model achieved during the training phase to validate the generalization ability of the achieved model. This validation set is used during the MOO process and helps to choose the best performing solutions with unseen data. After the training and validation phases are over, the final model i.e, the model that was trained using the best set of hyperparameters found by the MOO-GA, is created and the outputs are produced are compared to the test set in order to validate the quality of the predictions. The outputs, as already explained, are in the form of probabilities of the data point belonging to either one of the classes, 0 or 1.

*2) Multi-Objective Optimization GA:* When creating a Machine Learning model, there are many design choices when it comes to the model architecture. Most of the times the user doesn't know beforehand how should the architecture of a given model be like, thus exploring a range of possibilities is desirable. This is the case with the XGBoost binary classifier's hyperparameters, which are the parameters that define the classifier's architecture. This process of searching for the ideal model architecture is called hyperparameter optimization.

A multi-objective optimization approach is taken instead of a single-objective one due to the fact that the implemented system, although being in the first place a Machine Learning system, is also a system with the ultimate goal of making profitable trades in the stock market and with a low risk. As such, naturally a statistical measure to evaluate the system's performance with respect to the predictions made has to be used, in this case the accuracy, but on the other hand a metric to evaluate the capacity of the system to achieve good returns while minimizing the risk must also be used, in this case the Sharpe ratio. Therefore, the candidate solutions are evaluated with respect to the two chosen objective functions: the accuracy of the obtained predictions and the Sharpe Ratio. The set of each solution represents the fitness function to be optimized by the MOO-GA, with the goal of maximizing each of the objective functions. Thus, given the XGBoost binary classifier, the financial data set $X$ and the target variable array $Y$, the MOO-GA is going to search and optimize the set of the XGBoost binary classifier hyperparameters with the goal of maximizing the accuracy of the obtained predictions and the Sharpe Ratio. In order to find the best set of hyperparameters that aim at maximizing the two objective functions mentioned before, the MOO-GA approach is based on the Non-dominated Sorting Genetic Algorithm-II (NSGA-II).

Since there are many hyperparameters present in the XGBoost binary classifier, only the ones that have a significant impact on the architecture of the binary classifier and thus have a greater influence on its overall performance will be optimized. The chosen hyperparameters to optimize are the ones that also have a greater influence in the bias-variance trade-off and they are: the Learning Rate, the Maximum Tree Depth, the Minimum Child Weight and the Subsample and each one these hyperparameters constitute a gene of the chromosome in the MOO-GA.

In the proposed MOO-GA, the Two-Point Crossover operator was used, in which two points are selected on the parent's strings and everything between the two selected points is swapped between the parents. The mutation rate chosen is 0.2 which means that each new candidate solution generated by the crossover operator has a probability of 20% of suffering a mutation. Hypermutation is also used which is a method for reintroducing diversity into the population in an evolutionary algorithm. In this system, the mutation rate is adjusted during the evolutionary process in order to help the algorithm jump out of a local optimum. Therefore, when the hypermutation trigger is fired, the overall mutation rate increases from its original value of 0.2, and this happens when:

- The obtained set of non-dominated solutions doesn't change in 4 generations, increasing the overall mutation rate by 0.1;
- The obtained set of non-dominated solutions doesn't change in 6 generations, increasing the overall mutation rate by 0.15;
- The obtained set of non-dominated solutions still hasn't changed in 8 generations, again increasing the overall mutation rate by 0.15.

When the set of non-dominated solutions changes, the overall mutation rate goes back to its original value of 0.2.

The fitness function is a function that estimates the success of the candidate solution in solving the desired problem, i.e it determines how fit the candidate solution is. Since in the first place, the developed system's predictor is a binary

classifier, the accuracy of the XGBoost binary classifier in making predictions must be taken into account in the MOO process and it is one of the fitness functions. The accuracy of the model is determined after the model parameters are learned and fixed and no learning is taking place. Then the out-of-sample set is fed to the model and the number of mistakes are recorded, after comparing them to the true targets and the accuracy of the predictions made is calculated, using the expression in Equation 7.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Number\ of\ Predictions} * 100 \quad (7)$$

However, the goal of the system is also to maximize the returns obtained in the market while minimizing the risk associated to the trades. Thus, a fitness function capable of measuring the performance of the system in making profitable trades while minimizing the risk is also used. This fitness function is the Sharpe ratio, presented in Equation 8. The Sharpe Ratio is a measure of risk-adjusted return that divides the mean of the returns by the standard deviation of the returns. The standard deviation of the returns is a way of quantifying the risk.

$$SharpeRatio = \frac{AverageReturns}{StandardDeviationOfReturns} \quad (8)$$

After all the candidate solutions have their fitness value measured, the set of non-dominated solutions (Pareto front) is created. Since the objective is to maximize both the accuracy and the obtained Sharpe ratio, the desired solution is the one that performs equally better for both objective functions as improving one objective while not improving the other is not helpful. Therefore, in order to choose from the set of non-dominated solutions the one that will be used in the final model, both the accuracy and Sharpe ratio arrays, each with a length equal to the length of the set of non-dominated solutions, are normalized using the Min-Max normalization method, rescaling the fitness values to the range [0,1]. Then, for each row these two fitness values are summed and represent the final fitness value for the corresponding non-dominated solution. The chosen solution is the one that has the maximum final fitness value, meaning that it isn't optimal for neither one of the objectives, but reflects the compromise between the two objectives.

The termination conditions of a GA are important in determining when a GA run should end. Therefore, the termination conditions shall prevent that the GA has a premature termination and, on the other hand, that it runs for too long while making small or no improvements at all. The two termination conditions chosen for the MOO-GA in this system are if the algorithm reaches 100 generations or if the set of non-dominated solutions doesn't change for 10 generations. When one of the termination conditions is achieved, the MOO-GA stops its execution and outputs the solution with the highest fitness score that will be used to train the final XGBoost binary classifier which will output the predictions that constitute the trading signal, used in the next module.

In Table II, some of the most important parameters used in the XGBoost module are presented. In this system the same set of hyperparameters was used both in the PCA, XGBoost binary classifier (apart from the ones being optimized) and in the MOO-GA, regardless of the financial data being analyzed, in order to demonstrate that the proposed system is robust and capable of working with different financial markets. The only varying parameters between financial markets are the ones used in the DWT since the objective is to find both the best wavelet basis and level of decomposition for each analyzed financial market and the set of XGBoost hyperparameters that are optimized by the MOO-GA.

TABLE II
PARAMETERS OF THE IMPLEMENTED SYSTEM.

| **XGBoost Binary Classifier Module** | |
|---|---|
| Type of booster | gbtree |
| Objective | binary:logistic |
| Loss function | AUC |
| Number of estimators | 100 |
| **MOO-GA Module** | |
| Number of individuals | 128 |
| Number of generations | 100 |
| Number of gen. without changing the set of non-dominated solutions | 10 |
| Probability of Crossover | 50% |
| Probability of Mutation | 20% |
| Hypermutation increase after 4 gen. | 10% |
| Hypermutation increase after 6 and 8 gen. | 15% |

### *F. Trading Module*

The trading module is the module responsible for simulating trades in real financial markets. It receives as input the trading signal output by the XGBoost module and financial data and simulates the market orders in a financial market. In order to execute the trades, this simulator was designed as a state machine with the states Long, Short and Hold. Given the trading signal, a state machine is executed according to the market orders present in the trading signal. Given that the predictions made by the XGBoost binary classifier, $p(\hat{y}_t)$, represent the predicted probability of $y_t$ belonging to a class, since the two classes chosen represent the variation in close price for day $t+1$ with respect to the close price in day $t$, the trading signal can be constructed using these predictions. The trading signal is constructed using the predictions made by the XGBoost binary classifier in the following way:

- If $p(\hat{y}_t) \geq 0.5$ the chosen class is 1, which means that the stock's close price for day $t+1$, $Close_{t+1}$, is expected to have a positive variation, thus representing an opportunity to buy in day $t$, i.e the Long position is taken. This action is represented by the position 1 in the training signal, in the corresponding day;
- Conversely, if $p(\hat{y}_t) < 0.5$ the chosen class is 0, which means that the stock's close price for day $t+1$, $Close_{t+1}$, is expected to have a negative variation, thus representing an opportunity to sell in day $t$, i.e the Short position is

taken. This action is represented by the position 0 in the training signal, in the corresponding day.

Therefore, the trading signal has its values in the range [0,1] and the trading module is the responsible for interpreting these values and transforming them into trading actions.

## IV. Results

In order to train, validate and test the proposed system, financial data from five different financial markets is used. This financial data consists of the daily prices (Open, High, Low, Close and Adjusted Close) and Volume over the period of 25/02/2003 to 10/01/2018. From this data, 60% is used to train the system in order to generate all the predictive models, 20% to validate the models obtained and 20% to test the final model obtained after all the training has been done. In order to test the robustness of the proposed system to different financial markets, each with their own behaviour and this way ensuring diversity in the experiments made, the experiments are performed in the following financial markets: Corn futures contract, Exxon Mobil Corporation stocks, Home Depot Inc. stocks and S&P 500 index.

In each transaction made in a financial market, a fee applies and it is called transaction cost. The transaction costs are included in the trading module and are: 0.1% of the transaction value in the Corn futures contract, 0.005 USD per stock transacted in Exxon Mobil and Home Depot stocks and 1 USD per stock transacted in S&P 500 index.

The metrics used in order to evaluate the system's performance in the financial markets are: the number of transactions made, the rate of return (ROR), the maximum drawdown (MDD), the accuracy of the predictions made and the Sharpe ratio.

### A. Case Study I - Influence of the PCA technique

In the first case study, the influence of the PCA technique in the performance of the implemented system is analyzed. In order to do so, two systems are compared, the first being the basic system, i.e the system whose input data set is the normalized data set containing the 31 financial features and the second being the system whose input data set is the one obtained after applying PCA to reduce the dimensionality of the normalized data set containing the 31 financial features. The results obtained for each system are presented in Table III, along with the Buy and Hold strategy results, and for each financial market the best results obtained are highlighted in bold.

By examining the obtained results it can be concluded that the use of the PCA technique plays an important role in improving the performance of the system since it allows not only to obtain higher returns, but also to achieve higher accuracy values meaning that it allows the XGBoost binary classifier to produce less complex models capable of a good generalization ability to unseen data and helping in avoiding overfitting the training data, this way allowing the system to achieve higher returns than the Buy and Hold strategy in the Corn futures contract and Exxon Mobil stocks. In the other two analyzed financial markets, the use of the dimensionality reduction is vital for the system to obtain positive returns since in the basic system the generalization capacity wasn't good enough to classify correctly the test set. Without the application of the PCA the system can only achieve higher returns than the Buy and Hold strategy in the Exxon Mobil Corporation stocks. Furthermore, the use of the PCA technique also allows the system to obtain an higher Sharpe ratio and a lower MDD in the Corn futures contract, in Home Depot Inc. and in the S&P 500 index, which means that the solutions obtained generate good returns with lower associated risk to the trades made, when compared to the basic system and the Buy and Hold strategy.

TABLE III
B&H results and average results of the basic system and the system with PCA.

| | **B&H** | **Basic System** | **System with PCA** |
|---|---|---|---|
| **Corn** | | | |
| Transactions | 2 | 62 | **47** |
| ROR (%) | -15.92 | -17.18 | **8.90** |
| MDD (%) | 31.1 | 30.2 | **24.9** |
| Accuracy (%) | - | 50.7 | **51.3** |
| Sharpe Ratio | -2.32 | -1.94 | **-0.41** |
| **Exxon Mobil** | | | |
| Transactions | 2 | 87 | **231** |
| ROR (%) | 2.51 | 9.56 | **15.63** |
| MDD (%) | 25.5 | 25.9 | **25.9** |
| Accuracy (%) | - | 50.9 | **52.5** |
| Sharpe Ratio | -0.96 | 1.70 | **0.65** |
| **Home Depot Inc.** | | | |
| Transactions | **2** | 101 | 13 |
| ROR (%) | **97.43** | -26.56 | 64.64 |
| MDD (%) | **16.8** | 52.2 | 25.8 |
| Accuracy (%) | **-** | 47.3 | 53.4 |
| Sharpe Ratio | **1.55** | 1.14 | 1.34 |
| **S&P 500** | | | |
| Transactions | **2** | 188 | 185 |
| ROR (%) | **32.05** | -17.04 | 21.97 |
| MDD (%) | **14.14** | 26.5 | 11.8 |
| Accuracy (%) | **-** | 48.9 | 52.6 |
| Sharpe Ratio | **0.62** | 0.15 | 1.27 |

A graphical representation of the comparison of the returns obtained using the basic system and the system with PCA is presented in Figure 4 for the Corn futures contract for the testing period and it can be observed that, as mentioned before, with the introduction of the PCA the system is capable of obtaining higher returns.

### B. Case Study II - Combining PCA and DWT

In the second case study, after having analyzed the importance of the PCA in increasing the overall performance of the system, the DWT technique is now combined with the PCA technique to achieve a system that not only performs dimensionality reduction to the financial input data set but also performs a noise reduction procedure to this data set, in order to analyze if this two techniques applied together allow the system to achieve even better results than the system

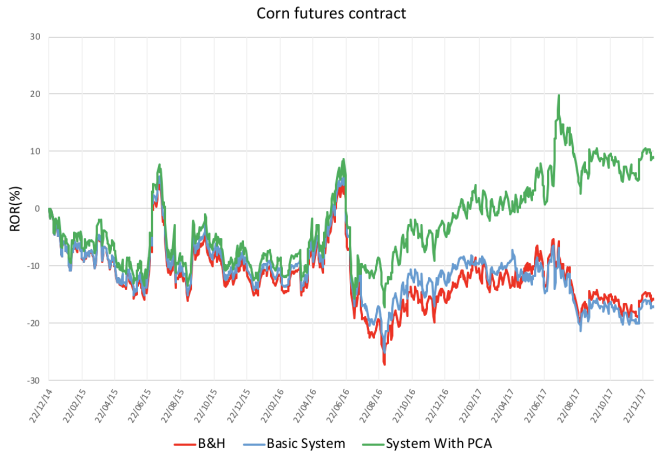


Fig. 4. B&H returns and average returns of the basic system and the system with PCA, for the Corn futures contract.

using just PCA. The results obtained for each system are presented in Table IV, along with the Buy and Hold strategy results, and for each financial market the best results obtained are highlighted in bold. Since presenting all the analyzed combinations of PCA and the different wavelet basis, orders and levels of decomposition would be too extensive, in Table IV only the two best performing combinations of PCA and the different wavelet basis, orders and levels of decomposition are presented.

TABLE IV
B&H RESULTS AND AVERAGE RESULTS OF THE SYSTEM WITH PCA AND WITH PCA AND DWT.

| | **B&H** | **System with PCA** | **PCA and Haar Lv.5** | **PCA and Db3 Lv.5** |
|---|---|---|---|---|
| **Corn** | | | | |
| Transactions | 2 | 47 | **77** | 87 |
| ROR (%) | -15.92 | 8.90 | **25.86** | -3.95 |
| MDD (%) | 31.1 | 24.9 | **22.9** | 24.9 |
| Accuracy (%) | - | 51.3 | **51.9** | 51.2 |
| Sharpe Ratio | -2.32 | -0.41 | **0.65** | -0.63 |
| **Exxon Mobil** | | | | |
| Transactions | 2 | 231 | **183** | 176 |
| ROR (%) | 2.51 | 15.63 | **22.68** | 16.28 |
| MDD (%) | 25.5 | 25.9 | **22.9** | 20.0 |
| Accuracy (%) | - | 52.5 | **51.5** | 50.1 |
| Sharpe Ratio | -0.96 | 0.65 | **1.34** | 1.03 |
| **Home Depot** | | | | |
| Transactions | 2 | 13 | 50 | **15** |
| ROR (%) | 97.43 | 64.64 | 73.69 | **98.23** |
| MDD (%) | 16.8 | 25.8 | 18.8 | **17.3** |
| Accuracy (%) | - | 53.4 | 52.8 | **54.1** |
| Sharpe Ratio | 1.55 | 1.34 | 1.19 | **1.55** |
| **S&P 500** | | | | |
| Transactions | **2** | 185 | 253 | 292 |
| ROR (%) | **32.05** | 21.97 | 25.78 | 10.93 |
| MDD (%) | **14.14** | 11.8 | 12.6 | 15.6 |
| Accuracy (%) | **-** | 52.6 | 51.3 | 50.9 |
| Sharpe Ratio | **0.62** | 1.27 | 2.23 | 0.29 |

By examining the obtained results it can be concluded that in all the financial markets analyzed, with the introduction of the DWT denoising to the system with PCA, the system has a better performance, not only in the returns obtained, but also in the other evaluation metrics. The system resultant from the combination of PCA and DWT that performed the better between all the analyzed systems is the system with PCA and Haar wavelet with a level of decomposition of 5, obtaining higher returns than the Buy and Hold strategy in two of the four analyzed financial markets (Corn futures contract and Exxon Mobil Corporation stocks), with the system using PCA and Daubechies of order 3 wavelet with a level of decomposition of 5 obtaining higher returns in the Home Depot Inc. stocks.

If only one combination of PCA and wavelet basis, order and level of decomposition was to be used, this would be PCA and Haar wavelet with level of decomposition of 5 since it's the combination that allows the system to obtain, in general, the best results, not only in terms of returns but also in the other evaluation metrics. Given that the Haar is the simplest wavelet, it is the more flexible one, meaning that it can be applied to signals with different characteristics and achieve an acceptable performance in each one. Although with the Daubechies and Symlet wavelets the resultant signal is smoother, the Haar wavelet allows for a better detection of sudden changes in the signal which is a very important characteristic when dealing with highly oscillating signals such as financial market signals.

Furthermore, the obtained accuracy values in the system combining PCA and DWT are all above 50%, with the worst accuracy obtained being 51.5% in the Exxon Mobil Corporation stocks and the best being 54.1% in the Home Depot Inc. stocks. Therefore, these obtained accuracy values prove that, in contrast to the EMH theory, the financial markets aren't purely chaotic and unforecastable, meaning that a Machine Learning system developed to trade in financial markets can learn from historical financial data and use some of the patterns learned in order to come up with trading strategies that can be applied to more recent time periods and that perform better than random guessing.

Overall, it can be concluded that the combination of the PCA and DWT denoising techniques allows the system to obtain better results than the system using just PCA. This is due to the fact that in this system the PCA reduces the dimension of the financial input data set and the DWT performs a denoising to this reduced data set, which not only helps in avoiding overfitting the training data, since there are less features in the data set, but also helps in removing some irrelevant samples that could harm the system performance, therefore aiding the system in the learning process and increasing its generalization ability. However, this increase in performance can only be verified when the appropriate wavelet basis, order and level of decomposition are used, since an inadequate use of the DWT can result in a worse system performance and even in lower returns than using just PCA.

A graphical representation of the best and average returns obtained using the system with PCA and the Haar wavelet and a level of decomposition of 5 is presented in Figure 5 for the Corn futures contract for the testing period.

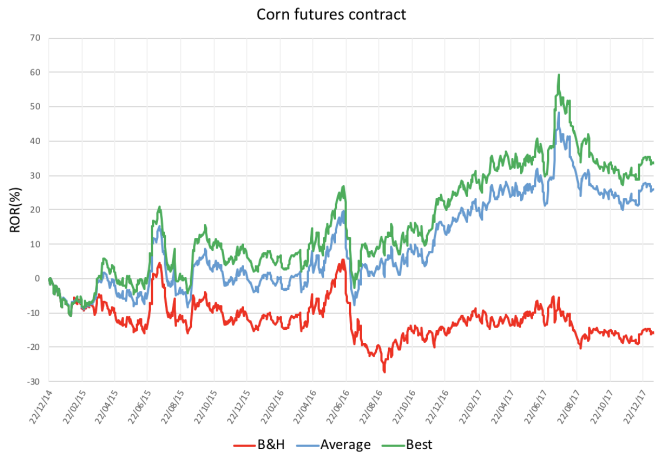


Fig. 5. B&H returns and the best and average returns of the system with PCA and Haar wavelet with a level of decomposition of 5, for the Corn futures contract.

It can be observed that the proposed system behaves well in highly oscillating periods, i.e periods with a sudden change in direction where the system can adopt a different position and therefore increase the returns obtained. As mentioned before, the use of the Haar wavelet and its ability in detecting sudden changes in a signal is crucial for the system to be able to change its position in these highly oscillating time periods, while at the same time not compromising its ability in identifying trends and therefore also profiting in less oscillating time periods, which is also why it was the wavelet basis that obtained overall the better results. Furthermore, the average rate of return of the portfolio is 43.15% when considering for each financial market the best combination of PCA and DWT, while the B&H achieves 29.02% on average.

## V. Conclusions

In this paper, a system combining PCA and DWT for dimensionality reduction and noise reduction, respectively, with an XGBoost binary classifier optimized using a MOO-GA is presented, with the purpose of achieving the maximum returns possible, while minimizing the level of risk in the trading strategy. The data preprocessing done using the PCA and the DWT, together with the optimization of the hyperparameters of the classifier using a MOO-GA help in creating a system that is robust enough to obtain good results in financial markets with different behaviours. It was concluded that, combining the PCA with the DWT, the input data set has its dimension reduced and some irrelevant samples are discarded through the noise reduction phase, resulting in an input data set that will not only make the system less prone to overfitting but also enhance its generalization capabilities, this way obtaining better results than just using PCA.

Some ideas for future work are: the use of the Wavelet Packet Transform in which both the approximation coefficients and the detail coefficients are decomposed; using Bayesian Optimization instead of using a GA, which is a promising new method for optimization problems; the use of different fitness functions for the GA like the Risk Return Ratio (RRR), that has into account the risk and the number of days spent in the market; a more in-depth analysis of the influence of the MOO-GA parameters variation in each case study in order to verify if better results can be obtained using a different set of parameters.

## References


[1] B. G. Malkiel. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82, 2003.

[2] S. B. Achelis. *Technical Analysis from A to Z.* McGraw-Hill Education, $2^{nd}$ edition, 2013. ISBN:978-0071826297.

[3] Y. He, K. Fataliyev, and L. Wang. Feature selection for stock market analysis. *Neural Information Processing. ICONIP 2013. Lecture Notes in Computer Science*, 8227:737–744, 2013.

[4] A. Galli, G. Heydt, and P. Ribeiro. Exploring the power of wavelet analysis. *IEEE Computer Applications in Power*, 9(4): 737–744, Oct. 1996.

[5] P. S. Addison. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*. CRC Press, $1^{st}$ edition, 2002. ISBN:978-0750306928.

[6] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[7] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? *Journal of the Royal Statistical Society, Ser. B*, pages 371–394, 1995.

[8] S. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, Sept. 2000.

[9] R. E. Schapire. A brief introduction to boosting. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 2:1401–1406, July 1999.

[10] P. Bühlmann. Bagging, boosting and ensemble methods. *Handbook of Computational Statistics*, pages 985–1022, 2012.

[11] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[12] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, Oct. 2001.

[13] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN:978-0412048418.

[14] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, 1992. ISBN:978-0262082136.

[15] P. Ngatchou, A. Zarei, and A. El-Sharkawi. Pareto multi objective optimization. *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, Nov. 2005.

[16] K. Deb. Multi-objective optimization using evolutionary algorithms: An introduction. *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 1–24, Feb. 2011.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr. 2002.

[18] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2):111–117, Jan. 2006.