MDSAA

Master Degree Program in

Data Science and Advanced Analytics

CRYPTOCURRENCY PORTFOLIO OPTIMIZATION USING GENETIC ALGORITHMS

Maikel Ricardo Pereira de Sousa

Project Work

presented as partial requirement for obtaining the Master Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

CRYPTOCURRENCY PORTFOLIO OPTIMIZATION USING GENETIC ALGORITHMS

by

Maikel Ricardo Pereira de Sousa

Project Work presented as partial requirement for obtaining the Master's degree in Advanced Analytics, with a Specialization in Data Science / Business Analytics

Supervisor: Leonardo Vanneschi

November 2022

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisbon, November 2022.

ACKNOWLEDGMENTS

Listing the people that have made this work possible is not only important but also necessary to not let their support go unnoticed. I should start by thanking my tutor, Dr Leonardo Vanneschi for providing me with the proper guidance during the development of this master thesis, I am honored to have worked with him.

Then I shall thank my parents for instilling in me the importance of hard work and perseverance. It would be impossible not to mention an important influence such as my high school year's teacher and friend Gilberto, for showing me the transcendental nature of books and learning, virtues that became the building blocks on my life. I want to also thank my girlfriend Marlene for being kind, supportive and patience during a challenging time in our lives, I couldn't done it without you and Frida. Luis and Sarah, my dearest friends, without you I would certainly have derailed a long time ago, thank you for your continuous support.

Lastly, I would have to thank my past self, for having the courage to take a step forward, risking being uncomfortable for some time, a step, that without knowing, allowed me to unveil a fulfilling path for my future as well as a re-awakening of my curiosity.

ABSTRACT

Blockchain most discussed application has been in cryptocurrency, being Bitcoin its first. Unbeknownst to many, Bitcoin not only introduced a new digital means of exchange creating fertile ground for others trying to emulate it. Cryptocurrencies took relevance beyond computer science spheres to reach a place of relevance as a security for several investors, making it relevant to study beyond computer science spheres. Economists, statisticians, and portfolio managers, have taken the subject of cryptocurrencies as case study for open digital finance, opening questions regarding price behaviors and objective investment strategies, creating opportunities of research on fields such as machine learning. Nevertheless, each cryptocurrency has its technicalities, and different value proposals, making the subject relatable, in some sense, to traditional financial instruments, from which some lessons can be of use, for example, how it is difficult to come up with a unique approach or toolset that forecasts prices and optimizes investments, at least in a general sense. Here the objective is to tackle a cryptocurrency portfolio optimization by means of genetic algorithms. Two approaches are suggested, the first *Limited Trading* approach, consists on using genetic algorithms to find how much of the coins to invest for profit considering to sell at the last day. Second approach is called *Open Trading*, much like the first, intends to find profit but it allows buying and selling at each timestep. In both cases respecting budgeting limitations and using forecast price values of each coin, but for this, instead of delving into the complexities of thousands of possible coins and algorithms, it was used a combination of machine learning methods to forecast prices of the coins in the portfolio. It was found that *Limited trading* outperforms its counterpart in *fitness* (expected portfolio value) and *%Return*. It was also found a genetic algorithm parametrization successful for both strategies, and highlighting the value of the theoretical proposal for fitness optimization based on matrix operations for *Open Trading*, which has room for improvement and further development in fields beyond portfolio optimization.

Contents

List of Figures

1 Introduction

Let's contextualize this work and specify the problem we set to tackle and how it is pretended to be solved.

1.1 Background and problem identification

Blockchain technology is the protocol and client through which transactions are stored, it is better understood as a ledger where exchanges of assets (tangible or intangible) between users are recorded, such ledger is also immutable and shared. But explaining this technology without referring to Bitcoin wouldn't be possible, because it was introduced on a paper by Satoshi Nakamoto [Nakamoto (2008)], probably a pseudonym, that no one has been able to connect to an individual or group of people. As consequence of Satoshi's paper, Blockchain has been one of the most ground-breaking technologies emerged since the internet mass adoption, creating potential possibilities and innovation into the future.

The mainstream didn't catch right away on the world changing capabilities of this technology, but eventually its potential started to show, firstly as a source to store value, privacy, and decentralization, given to the run-of-the-mill person the possibility for protection against uncontrollable factors, such as hyper-inflation or persecution from authoritarian regimes.

During its raise in popularity controversial discussions with people sharing their views and comments on this technology were common, something understandable when dealing with technologies with potential to disrupt, but one of the leading voices in these discussions, specially on the technological front, is Andreas M. Antonopoulos (Andreas M (2019)). He stresses on the importance to discern what is important and what is not on the crypto space, for this, he explains his criteria by the *Five Pillars of Open Blockchains*, to discern when something really needs a blockchain or it doesn't. These pillars are:

a) *Open:* it is free to access or participate without authorization, vetting, verification of nationality, ethnic origin, ect. Even more, it does not even require for the user to be a human, a piece of software can use blockchain.

b) *Borderless:* it does not matter where you live or where you travel.

c) *Neutral:* a system of money should not care who the sender or the recipient is or the purpose, it should only get the transaction through.

d) *Censorship-Resistant:* no person or entity can stop you from sending money.

e) *Public:* is the idea where everything you do is verifiable on the network by everybody else.

Eventually, cryptocurrencies won relevance within the financial space and became a new type of security, making it relevant to study and understand how to incorporate them in different investment strategies, but this type of assets are particularly volatile, creating an extra layer of difficulty when optimizing for return.

As an immediate consequence, a need for a way to know how these assets might behave can be of use. Here is where forecasting and optimization tasks become relevant. In both fronts this work aims to tackle these by machine learning approaches. First, to generate an accurate forecast a machine learning approach will be developed, and for a proper optimization this work would use evolutionary algorithms.

1.2 Specific Objectives

This work has the aim to reduce complexity in cryptocurrency use as a security or investment asset when dealing with a huge pool of possibilities and a limited budget. Price volatility is a factor that can be inconvenient to reach a profitability objective, nevertheless, it is expected that this solution finds the best investment strategy for time spawns no higher than 15 days out of a pool of coins.

Achieving this goal, while maximizing return, requires some intermediate steps to be set in place to build a solid path towards that:

a) Study forecasting approaches for a bulk of historical streams with different trends, seasonality, and historic cycles.

b) Study genetic algorithms maximization approaches that respect budgeting limitations and independent strategies.

c) Compare the returns between two different optimization approaches.

The reason for the choice of using genetic algorithms for the optimization has several fronts. One is due it's simplicity when searching the right parametrization from scratch, initial experimentation and computing speed play an important part in the decision process, as opposed to other algorithms where finding the right parameters combination can be tricky and computationally expensive, specially when done from scratch, to list one example, Neural Networks typically used on Deep Learning, are dependent on factors such as the network architecture (input and hidden layers size), batch size, leaning rate, activation function and so forth. Another reason is that genetic algorithms removed, at a certain extent, the need to go back and retool the forecast approach, allowing more focus on the genetic algorithm parametrization, experimentation and results analysis. Finally, personal curiosity on these algorithms played its part on wanting to tackle a portfolio optimization.

2 Methodology

Two different investment strategies will be adopted:

- *Limited trading:* Containing exactly two buy/sell opportunities, given a certain 15 day interval, at start (buy) and at the end (sell).
- *Open trading:* Containing flexibility in the positions at each time-step for a given 15 day interval, this means that the algorithm is open to buy and/or sell on each day, any of the coins in the portfolio.

Handling effectively the forecasting step becomes of vital importance for success at the subsequent optimization step, therefore historical data is an important asset to build good price predictions. For this purpose the *insvestpy* python library will be our primary tool. *Investpy* has an API that returns historical information about crypto assets, such as Volume traded, Open, Highest, Lowest, and Closing prices. The retrieved details for daily open prices for the coins will be used on building a forecast.

2.1 Forecasting

The historical prices for each coin chosen to be part of the investment portfolio would have have its particularities (seasonality, cycles and trend).Therefore the way these factors are going to be accounted for is through a combination of univariate Time-Series Machine Learning Forecasting approach from historical prices. Several methods will be wrapped to get the better of each in predicting future prices.

- *MARS:* also know as Multivariate Adaptive Regression Splines, is a flexible regression method that automatically searches for interactions and non-linear relationships to fit hinge functions:

  $$h(x-t) = [x-t]_+ = \begin{cases} x-t, & x > t \\ 0, & x \leq t \end{cases},$$

  (line segments) on sections of the time series. In this work the hinge functions used from the *pyearth* library are smoothed to better model the trend component of the time-series.

- *MLP Regressor:* or also called Multi layer perceptron is a fully connected class of feed-forward artificial neural network. This is a machine learning algorithm that consists of three types of layers, input, hidden, and output layers. The first receives an input signal that is passed onto the hidden layers for processing, in this work the shape of this hidden layer is (300,200,150), producing as output a prediction for the time-step. The model has as target the residuals between the historical prices and MARS values.

- *XGB Regressor:* or also called extreme gradient boosting, is a tree based machine learning algorithm also known as distributed gradient-boosted decision tree. Works as a ensemble learning algorithm, which means that combines multiple decision trees to create a prediction Tianqi Chen (2016). The residuals from the previous step (MLP Regressor) would be fitted with extra calendar features, the intention is to account for any seasonality left.

A sequencial use of these methods in the order listed above would create the price predictions of each coin individually to then be used on the portfolio optimization which would mold the genetic algorithm choices. More details to follow on the next section.

2.2 Problem Setup

Before diving in, it is important to remark that the problem was tackled with two different optimization approaches and each will be described below, but let's highlight their mutual structures. The forecast step was generated by wrapping different machine learning methods (those presented in the last subsection), and use it's output to train a genetic algorithm able to engineer a portfolio choice. The distinction between these two approaches is important to highlight since it implies differences between the populations as well as the fitness function implementation.

To better illustrate how GA portfolio optimization will be handled, let's define each case:

1. Let $C = \{C_i, 0 \leq i \leq n\}(n \in \mathbb{N})$ be a set of coins. From each coin the time series containing the open price values can be extracted and map each to a forecasted value on the expected price of the coin for a day k, $f(C_i, days = k)$. For this particular purpose it is required to keep $f(C_i, days = 15)$, for each i such that $0 \leq i \leq n$.

   Instead of finding the best subset of currencies that maximizes the expected value for the portfolio, let's take into consideration the entire set of possible cryptocurrencies (C) and find the vector of real values α_i that maximizes the next expression:

   $$\sum_{i=0}^{n} \alpha_i f(C_i, days = 15)$$

   If re-written to a vectorial form the previous equation can be expressed as,

$$\vec{\alpha} * \vec{f}(C_{15}) = \sum_{i=0}^{n} \alpha_i f(C_i, days = 15). \quad (1)$$

Where the set of α_i's are real values that can be used as a vector representing an individual on the population from which the GA will start evolving.

For this setup, the implementation will be able to search small and big scalars for the vector entrances, for example, when the problem setup faces high prices and a relatively small budget, the algorithm can be able to find optimal solutions pointing to 1% or 0.1% of a Bitcoin.

2. Let $C = \{C_i, 0 \leq i \leq n\}(n \in \mathbb{N})$ be a set of coins. From each coin the time series containing the open price values can be extracted and map each to a forecast vector with the expected price of the coin on each day $k = 1, 2, ..., 15$, $(\overrightarrow{f_k(C_i)})$. Let $f(C)$ be the matrix with the forecasted prices for each coin on all the time-steps, defined as:

$$f_{(-1,14)}(C) = \begin{pmatrix} P_{-1}(C_1) & P_0(C_1) & \dots & P_{14}(C_1) \\ P_{-1}(C_2) & P_0(C_2) & \dots & P_{14}(C_2) \\ \vdots & \vdots & \vdots & \vdots \\ P_{-1}(C_n) & P_0(C_n) & \dots & P_{14}(C_n) \end{pmatrix},$$

Where the first column in the matrix above are the last available prices in the historical and the remaining the forecasted prices for each coin at each time-step.

Let $A = (\alpha_{(i,j)}) \in \mathbb{R}^{nx15}$ be a matrix representing the trading positions for each day within the interval. The intention is to find a matrix A that maximizes the following:

$$\sum_{i=1}^{15} \sum_{j=1}^{n} \alpha_{(i,j)}.[f_{(-1,13)}(C) - f_{(0,14)}(C)] \quad (2)$$

Which represents the expected return of trading for a given forecast f. Certain conditions are required on A to ensure that the GA finds admissible solutions, for example, a solution can not be accepted if implies selling, at any time-step, more coins that it already has, as well as not exceeding the given budget.

2.3 Data Management

To allow a dynamic use and analysis of the forecasting output, a MySQL database was used to store the forecasting results for each coin, avoiding redundancy, and latency when the optimization step takes place.

- *Crypto Forecast:* to keep the forecasted prices of the model for each coin and time step as well as real prices when available.
- *Model Metrics:* to store model performance metrics such as MAE, MSE, MAPE and SMAPE on training and validation sets.

During the optimization, data is extracted from the DB creating price vectors for the *limited trading* scenario and a price tensor for the *open trading* case. Important to highlight that the price tensor is created by stacking two matrices, one containing real price values when available and the other containing the forecasted prices.

3 Literature Review

To better understand how a portfolio optimization method was developed a theoretical overview would be necessary. Before delving into the literature is important to highlight, just as a guiding principle, by means of which tools the solution is obtained: *Machine Learning Forecasting* and *Genetic Algorithms* are such tools. Now that we've stated the theoretical aim of the section let's get to each a bit more in detail.

3.1 Time-Series Forecasting

Definition: In simpler terms a Time-Series is a sequence of data points taken sequentially over time. Formally, it can be defined as an stochastic process $\{X_t\}_{t \in T}(T \subseteq \mathbb{R})$. Where $X(t)$ or X_t denotes the value of the random variable X at time point t.

When dealing with time-series forecasting problems three components are considered important to model better a solution. These are:

1. *Trend:* long-term movement of the values.
2. *Seasonality:* seasonal variations in the values, characterized by regular time windows (daily, weekly, monthly).
3. *Cycles:* variability that is not of a fixed frequency or that rises and falls on a pattern that is not seasonal. Some examples of these are promotional campaigns or bank holidays.

Since yesteryear the subject of Time-Series has been studied extensively, mainly driven by contributions from disciplines such as mathematics, astronomy, demographics and statistics. Many innovations came initially from mathematics, later statistics, and finally machine learning Auffarth (2021). But time-series forecasting for cryptocurrencies is a relatively new field, therefore establishing a common ground for creation of certain baselines would require to take past knowledge developed on other fields well established. One example of these is *stock market forecasting*, this field shares some resemblance with cryptocurrency investing, although not as regulated in current days.

While the intention here is to tackle a forecasting task of several coins, each with its particularities, some literature can shine a light on the way to move forward. Ritika Chopra (2021) by reviewing 148 articles can shine a brighter light on a feasible path, here is discussed how AI approaches perform on generating predictions for the stock market. Traditional theories from the 70's, such as *efficient market hypothesis* states that stock prices are an indicator of all known market knowledge and its participants optimally use it (Burton G. Malkiel (1970)) and *random walk theory* states that future prices are completely random. Eventually with innovations on prediction methods and computational innovation, more sophisticated approaches emerged, some based into different explanatory variables such as psychological, behavioral, and economic, as well as using novelty technologies, such as machine learning, being this last one the more promising of the approaches. Ritika Chopra (2021) makes the case, that AI-based models not only attain a competitive edge but also shine an important light on certain methods, "neural and hybrid-neuro approaches are ideal for market forecasting since they provide more accurate outcomes than the traditional models".

Alhomadi (2021) tackles forecasting of one among the most important indexes which tracks the stock performance of 500 large companies listed on stock exchanges in USA, the *S&P 500*. Alhomadi (2021) uses different supervised machine learning models, among those XGBoost, and it highlights that regardless of the model used, high accuracy prediction on return is not always achieved, but backs up Burton G. Malkiel (1970) in the *efficient market hypothesis*.

Although machine learning approaches have a competitive advantage for stock market time-series forecasting, cryptocurrencies prices might be influenced on different factors besides financial, some examples are: search engines and social media. Smuts (2019) bears the case that by using a long short term memory *(LSTM)* recurrent neural network with data obtained from Google Trends and Telegram produces accurate results for a one-week time spawn. Kevin Martin (2020) approaches bitcoin price forecasting by using blockchain data and a feature selection pipeline, their final model is an ensemble model that has a gradient boosting regressor (a simpler version of XGBoost, which is used here) as one of its components, the performance is assessed by means of *Mean Absolute Percent Error* (described below).

Kao et al. (2013) shows how an approach on forecasting problems by means of wrapping methods, and these outperform traditional statistical approaches, such as ARIMA, for finance forecasting application, one of the building blocks to build the model showed on this paper is MARS (also used here).

The literature discussed above shines a clearer path to follow on the forecasting approach of this research and makes the case for a forecasting model built from a machine learning ensemble, given the differentiated fact of each coin time-series. Also by considering only features related to the historical prices of each coin the model falls under the *efficient market hypothesis* (Burton G. Malkiel (1970)).

Even that when assessing forecasting approaches, regardless of the underlying Time-Series, is clear now how there's no "one fits all" forecast solution for such endeavor, but regardless of this fact it is important to understand how well a solution fits the task of predicting prices for the 15 days investing window. Therefore, using a proper metric is paramount, specially one that is both, visually informative and quantifies how well it fits the two periods, training (past historical) and testing (15 days investing window).

There's not just a single metric that is apt for the purpose of any arbitrary application or dataset. Depending on the dataset, you might have to search and try different error metrics and see which one best captures your objective (Auffarth, 2021, 107). Some popular metrics such as *MSE* (Mean Squared Error), *MAE* (Mean Absolute Error), *RMSE* (Root Mean Square Error) fail on being visually helpful when analyzing the fit of the model because of the different price ranges between coins.

The *mean percentage error (MAPE)*,

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|e_t|}{|y_t|}$$

Where $e_t = y_t - f(x_t)$, also called prediction error or residual and $f(x_t)$ is the prediction of the model for time step t. 0 represents a perfect model, and higher than 1 means the model's predictions are higher than the targets. But the MAPE does not have an upper bound which is a downside to being the metric of choice. Nevertheless, the *symmetric mean absolute percentage error (SMAPE)* defined as:

$$SMAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|e_t|}{\left(\frac{|y_t|+|f(x_t)|}{2}\right)}, \tag{3}$$

Not only has a lower bound but also an upper bound, which makes the percentage much easier to interpret (Auffarth, 2021, 111).

3.2 Genetic Algorithms

As it has been said genetic algorithms take a fundamental role. Therefore, some theoretical foundation has to be set in place to better showcase how this implementation was done. Genetic Algorithms are a subset of bio-inspired algorithms, specifically from the evolution theory of Charles Darwin.

Genetic Algorithms usually start from randomly generated group of potential solutions. In this group a *selection algorithm* takes place generating a new population of potential solutions where for each member a function called *fitness* determines the validity of each subject for the problem at hand. This random population are changed by means of *genetic operators* (mutation and crossover), each with a certain probability, to then obtain a new population of possible solutions. The process of obtaining an offspring from another population is denoted as *generations*, and is possible to preset how many times this process will be performed.

There is another important parameter within this evolution process, and is used to keep the best solutions. This parameter is known as *elitism*, consist on saving the best individuals into the offspring.

The combination of all these parameters allow for countless configurations, therefore, let's describe the components of each of the parameters that were used here.

3.2.1 Selection

Tournament Selection: This selection algorithm does not evaluate the fitness of all the individuals but only a part of them. When an individual has to be selected, a subset of individuals are randomly chosen from the current population. The selected individual is the one with better fitness among these. The method contains two layers, one consists in a selection independent from fitness and random, and a second completely deterministic and based on fitness.

This is the only method listed in the selection algorithm, and there is a reason for this, since the population is formed by random float numbers, it proved to be the most efficient algorithm within the basic algorithms for selection available in the literature. For example, by using *rank selection*, the algorithm didn't prove to outperform tournament selection in terms of fitness or execution time.

3.2.2 Crossover

A crossover operation consists on changing the syntactic characteristics of individuals to generate an offspring. The ones used here were:

- *Multiple Points Crossover:* this crossover method consists on choosing a number of random positions k and creating an offspring from the sliced sections of the two parents. The version most known of this crossover operator is *single point crossover* where $k = 1$. In this work, it is set by default $k = 3$ since it introduced enough variation in the population without disrupting so much that it didn't improved in terms of fitness.

- *Arithmetic Crossover:* consist of choosing a random linear interpolation between the parents. More formally, let be P_1 and P_2 two individuals in the population and $\alpha \in \mathbb{U}(0,1)$, i.e. a random number extracted from an uniform distribution in the interval $(0,1)$, then the offspring O_1 and O_2 are of the form:

$$O_1 = (\alpha * P_1) + ((1-\alpha) * P_2),\ O_2 = (\alpha * P_2) + ((1-\alpha) * P_1).$$

3.2.3 Mutation

A mutation operation creates a new individual by modifying portions of the syntactic structure of it, such change is usually a random one. The operators used here were:

- *Swap Mutation:* picks two random positions within an individual and swaps them.
- *Inversion Mutation:* picks two random positions within an individual to extract a slice, and inverts the order the slice.
- *Scramble Mutation:* picks two random positions within an individual to extract a slice, and shuffles the slice.

4 Results and Discussion

For each investment strategy (Limited and Open Trading), the portfolio performance was studied during 4 different set of 15 days time windows in 2021 for the top 100 crypto currencies by market cap (according to February 2022 data) while at the same time exploring 4 promising Genetic Algorithms parameter combinations.

The year 2021 was considered the perfect testing ground to assess this investment approach due the following factors:

- Enough historical information available for the top 100 cryptocurrencies. Main ingredient for a robust model to tackle forecasting tasks.
- Uncertainty regarding pandemic COVID-19, adding a layer of unprecedented variability on the markets.
- Different market shifts happened during this year in terms of crypto market capitalization.

4.1 Experimental Design

Here each of the components, forecasting and optimization, layout will be described. To assess the outcomes between them with enough significance each combination was executed 100 times along 4 different time windows during the year 2021.

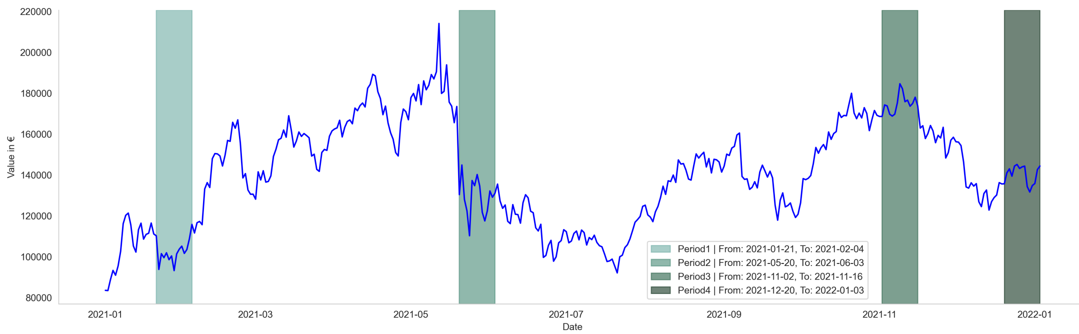

Figure 1: Value of the coins in the portfolio during 2021 and experimentation intervals

Above we can see the prices of the coins in the portfolio and how the prices behaved during each interval. More details on the particularities of each investment period will be addressed below.

4.1.1 Forecast Design

This step was the building block of this project, because based on its accuracy, when optimizing, algorithm portfolio decision was closer to be one that reached a profit.

Finding a good forecasting method able to produce accurate results required an initial trial and error approach, but in this sense the literature established a good way forward for exploration. Some of the factors into consideration were: varying historical availabilities, seasonality, trends and business cycles, therefore personalizing a method for each coin, although would be preferred, it wasn't scalable. Also, while considering that the forecasting window was smaller (15 days) when compared with the available historical for most of the coins (over 2 years), even with some overfitting, a good fitted method for historical would be able to follow shifts for the predetermined forecasting window.

Deciding on a baseline coin to build a forecast method is an important factor, since its architecture, although flexible enough to model each coin it will naturally influence how the model will predict other coins, because a coin can have its individual particularities that the baseline coin may not have. For example, if one were to consider correlation or relevance within the cryptomarket prices, the influence of any coin changes over time. Considering that historically Bitcoin has been the origin for crypto and drives most of the influence on the market in terms of news, prices and regulations, baseline was tested over Bitcoin prices.

As mentioned before the forecasting model was built by merging three methods, each with the idea to model different components on the time-series:

- *MARS:* models time-series trend adding a smoothing factor to avoid overfitting. It was found by experimentation that MARS smoother generalized better for the particular setup.

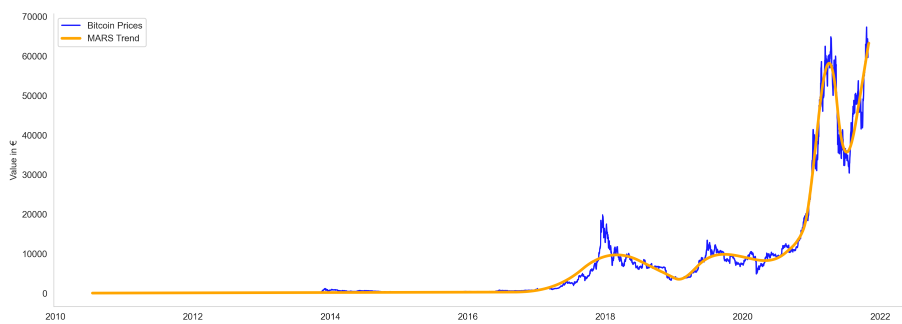

Figure 2: Bicoin and MARS

  Here we can observe how MARS is effective at designing a trend for the historical prices, even more, it creates a smoother version of this trend which would allow to pass remaining non-linear information into the following step.

- *MLP Regressor:* using as target the residuals between MARS and historical, and as input rolling averages of these residuals with windows of 365, 120, and 90 days as well as exponential weighted moving average and standard deviation, which is designed to reveal changes in the strength, direction, momentum, and duration of any residual trend (Auffarth, 2021, 312).

- *XGB Regressor:* using the residuals from the MLP Regressor predictions as target and calendar features as input (Month, Day, Week) to model remaining seasonality.

4.1.2 Genetic Algorithm Optimization Design

For each of the optimization strategies (Limited and Open) it was used as input the Crypto Forecast SQL Table described in the Methodology section, which contained the results from the forecast for the different coins in a 15 days time windows. Depending on the strategy applied, the script performed a preprocessing step.

From initial experimentation certain combination on optimization parameters were considered for further exploration, and these are the optimization representatives to which the optimization would be assessed comparatively:

Identifier	**Selection**	**Crossover**	**Mutation**	**Crossover Prob**	**Mutation Prob**
Comb 1	Tournament	multiple points	scramble	0.8	0.5
Comb 2	Tournament	arithmetic	scramble	0.75	0.5
Comb 3	Tournament	arithmetic	swap	0.88	0.5
Comb 4	Tournament	multiple points	inversion	0.87	0.37

Now that the mutual parameters for both optimization scenarios has been presented, their particularities will be shown in detail.

4.1.2.1 Limited Trading

As shown in (1), the Limited Trading strategy aims to find a vector of real values $\vec{\alpha}$ that maximizes $\vec{\alpha} * \vec{f}(C_{15})$, where $\vec{f}(C_{15})$ is the vector of the forecasted values.

The remaining parameters of the strategy are:

- *Budget:* 10M€ was defined as investment limit.
- *Fitness Function:* implemented as (1), with the limitation that when the portfolio value its above the budget then fitness was assigned to its opposite (additive inverse).
- *Population Size:* 400.
- *Generations:* 400.
- *Valid values:* random numbers on the interval $[0, 5]$ uniformly distributed.
- *Elitism:* TRUE. With the intention to keep each generation the best performer in terms of fitness.

4.1.2.2 Open Trading

As shown in (2), the Open Trading strategy aims to find a matrix $A = (\alpha_{(i,j)}) \in \mathbb{R}^{nx15}$ representing the trading positions for each day within the 15 days interval.

Remaining parameters on this strategy are as follow:

- *Budget:* 10M€ was defined as investment limit.
- *Fitness Function:* implemented as (2), by doing the sum across the 15 days of the daily value of the portfolio according to the forecast. Two limitations where set in place, the first being when the portfolio value its above budget. Second, when there's solutions that intend to sell at any timestep more than what is already available. In both cases fitness was assigned to the opposite.
- *Population Size:* 400.
- *Generations:* 400.
- *Valid values:* random numbers on the interval $[-1, 5]$ uniformly distributed.
- *Elitism:* TRUE. With the intention to keep each generation the best performer in terms of fitness.

4.2 Optimization Parametrization Findings

The comparison process for both approaches (Limited and Open) would be in terms of the fitness function and the percentage of return (%Return) which is the quotient between the difference of the real value of the portfolio at the end (15 days after) by its value at the beginning. Although %Return is highly dependent on the forecast accuracy to fit the real prices this assessment ensures to find the best condition regardless of the interval since the underlying data is the same for each combination.

4.2.1 Limited Trading Findings

To compare each combination across the 400 generations, 4 Periods and 4 combinations let's graphically analyze the average evolution of fitness and %Return for each strategy:

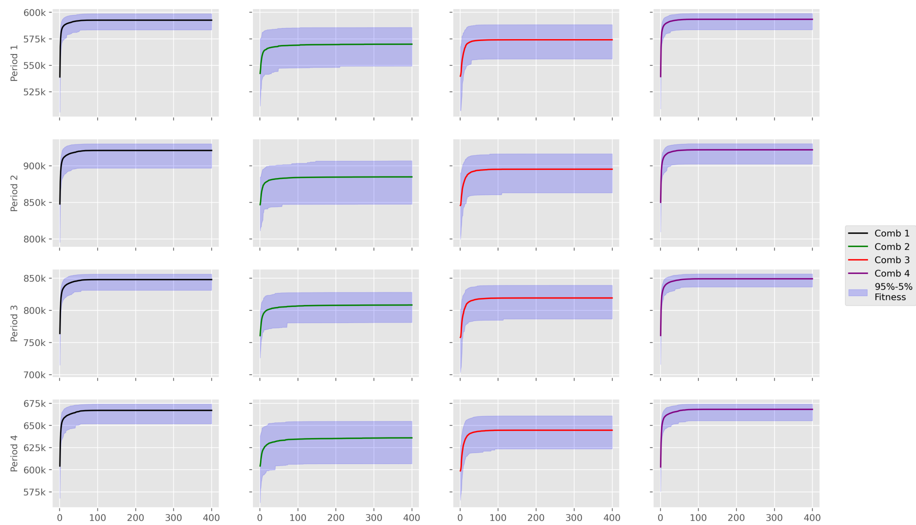

Figure 3: Generational Fitness for each combination by period (Limited Strategy)

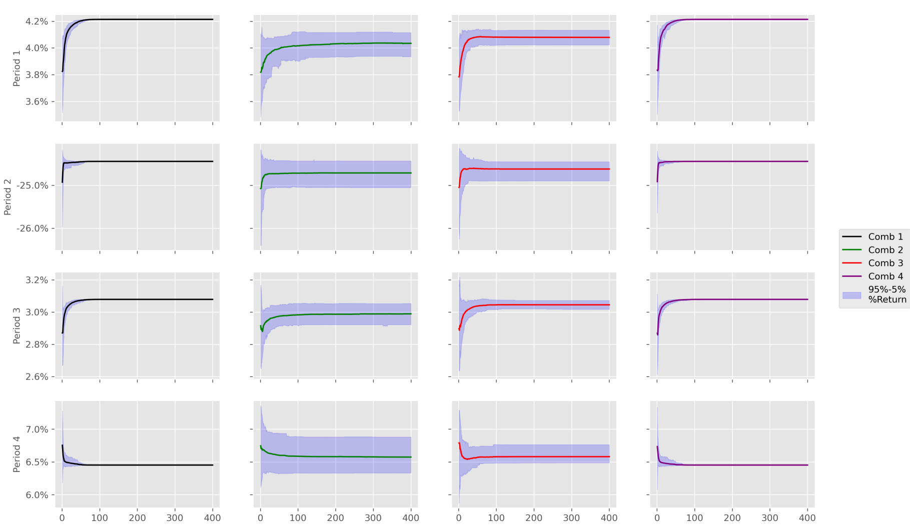

Figure 4: Generational Return for each combination by period (Limited Strategy)

From the last two figures we can see how regardless of the period *Comb 4* and *Comb 1* outperform in average both criteria, and shows less dispersion as it moves forward on the generations. Also, for *Period 2* none of the strategies generate a positive %Return, which is due the nature of the period and something adressed below. Since, *Comb 4* and *Comb 1* appear to be quite similar on average, is important to analyze with more detail what happens at generation 400:

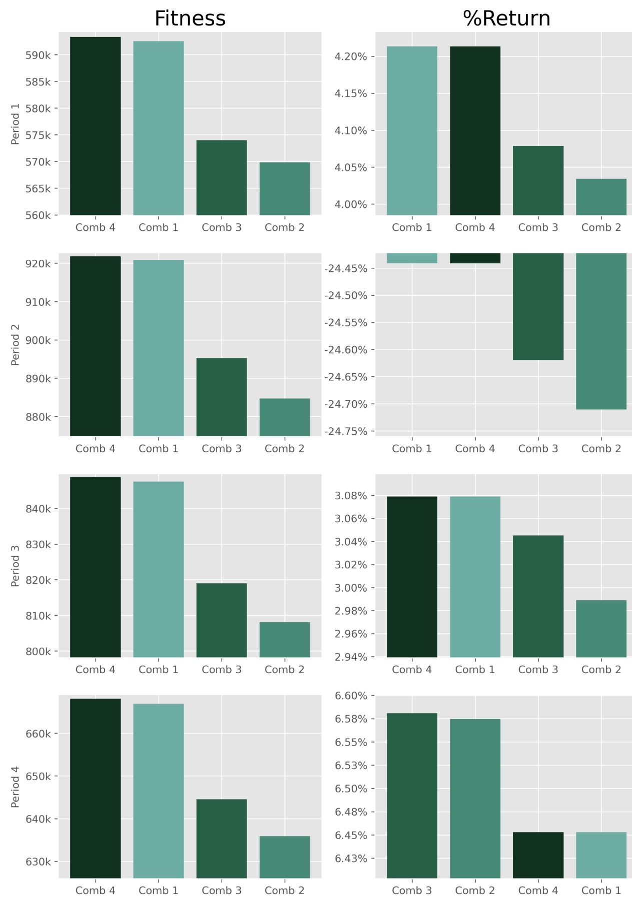

Figure 5: Average Fitness and %Return at $Gen = 400$ (best to worse)

Analyzing the figure above we can see how regardless of the period *Comb 4* fitness shows dominance over *Comb 1*, also *Comb 4* %Return outperforms *Comb 1* in two of the four periods. While being both quite similar, under this criteria, *Comb 4* comes as the best parametrization for the problem at hand.

4.2.2 Open Trading Findings

The assessment for this strategy would be analogous.

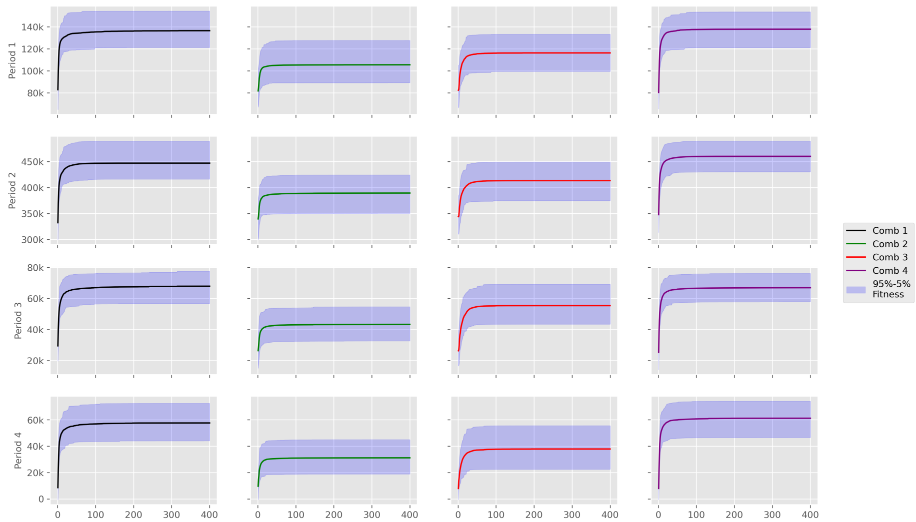

Figure 6: Generational Fitness for each combination by period (Open Strategy)

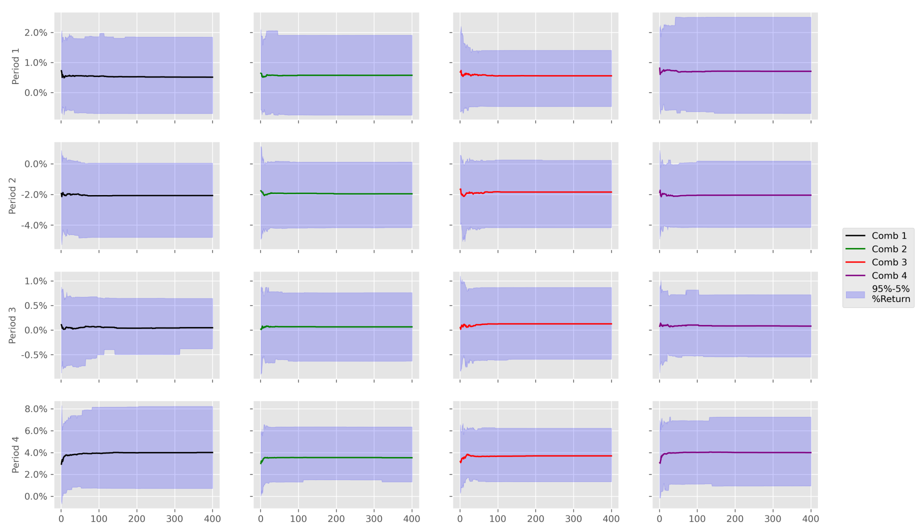

Figure 7: Generational Return for each combination by period (Open Strategy)

For this strategy from the last two figures we can see how it reaches higher Fitness values and wider 95%-5% bounds than the single strategy, but returns for some periods don't ensure a positive outcome, considering the amount invested.

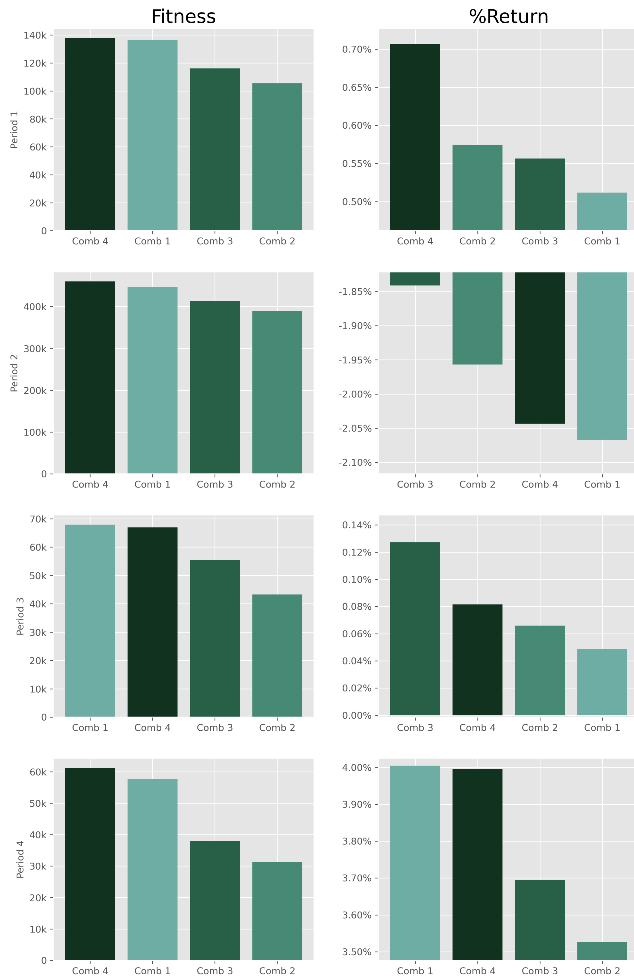

Figure 8: Average Fitness and %Return at $Gen = 400$ (best to worse)

Analyzing the figure above we can see that in three of the four periods *Comb 4* fitness shows dominance in average over *Comb 1*, also *Comb 4* %Return outperforms *Comb 1* in three out of the four periods and is the overall best performer (in average) in two periods. Therefore, considering this criteria *Comb 4* comes as the best parametrization for the open strategy.

Nevertheless, when comparing Open Strategy and Limited Strategy, the latter outperforms the former, being *Comb 4* the preferred parametrization for both.

4.3 Underlying conditions of each period

Each study period interacts not only with the genetic algorithm optimization but also with some underlying factors, such as the forecasting errors on the testing period and how is the influence of returns/losses for each coin in the portfolio.

4.3.1 Forecasting Errors

Understanding forecasting errors between training and validation sets allows us to visualize how the distribution of errors behave and influence the optimization, specially considering how the two sets have a significantly different samples between them.

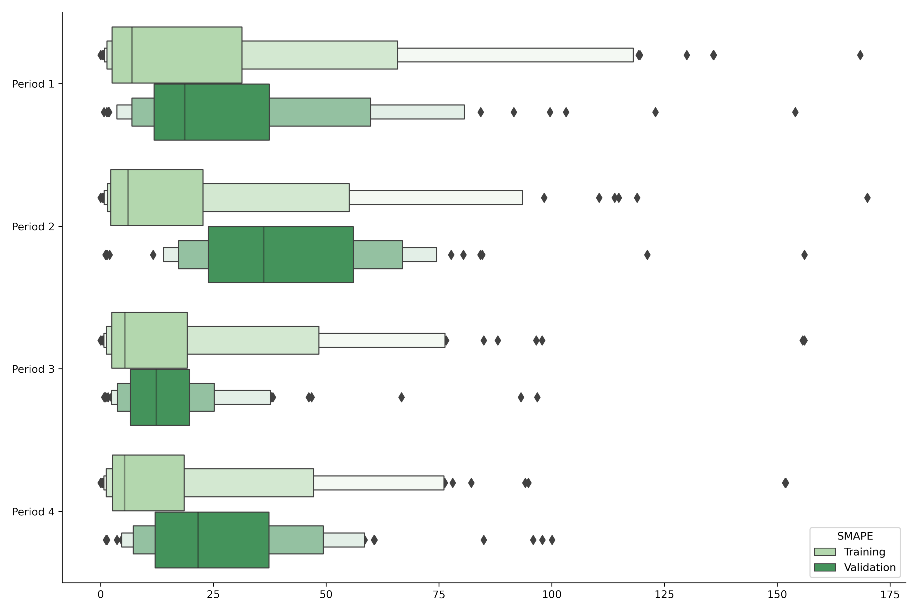

Figure 9: Distribution of errors for training and validation sets on each period.

One important highlight shown on 9 is that the SMAPE distribution errors for the training set has extreme values that are positioned relatively closer, meaning reduction in standard deviation, and how the forecasting algorithm shows learning capabilities. Although is limited on the generalization front, on average, is noticeable some overfitting between the two sets on every period, but in general terms SMAPE distribution errors for the validation set show less dispersion than its counterpart.

Nevertheless a low SMAPE is only relevant for the Open trading strategy where the daily trading errors can influence returns, and is in fact one of the handicaps of this strategy, that when combined with the next section explains its results.

4.3.2 Returns/Losses for each coin in the portfolio

When considering the Open Trading results, for the four testing periods, and introducing the possibility of daily trading becomes important to understand the algorithm results, for this how the market behaved daily becomes of paramount importance since it would influence the algorithm choices, because some coins and days may have a higher influence on positive returns, therefore this understanding on a first instance comes by analyzing the daily positive returns of each coin (regardless of the magnitude).

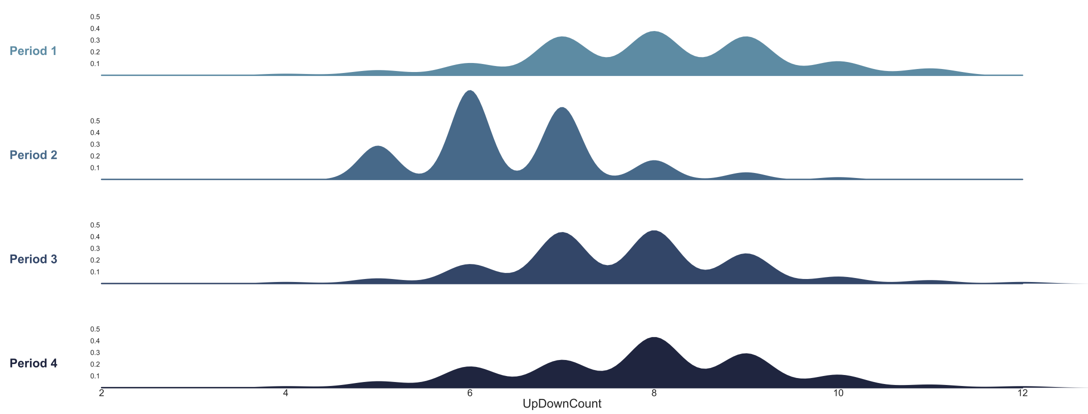

Figure 10: Amount of daily increases by coin in the portfolio

In 10 Period 1 and Period 4 show a positive daily balance per coin and the same positive outcome is obtained on the same periods from the Open Strategy. But daily increments only explains how the daily relevance of the coins is distributed and influence the genetic algorithm choices but when comparing Periods 1 and 4 distributions seem quite similar, therefore the magnitude of these returns becomes important on explaining their differences.

At this point understanding the difference in outcomes between Period 1 and Period 4 by excluding the magnitudes of the daily changes becomes insufficient, therefore taking a dive into the magnitude of the daily changes can provide a better understanding on this approach, therefore a comparative analysis on the mean Daily percent changes by coin between periods is considered. For this universe (daily percent changes by coin) is deemed relevant a comparison on the percentiles of the averages distribution, since the periods contain some coins with extreme averages.

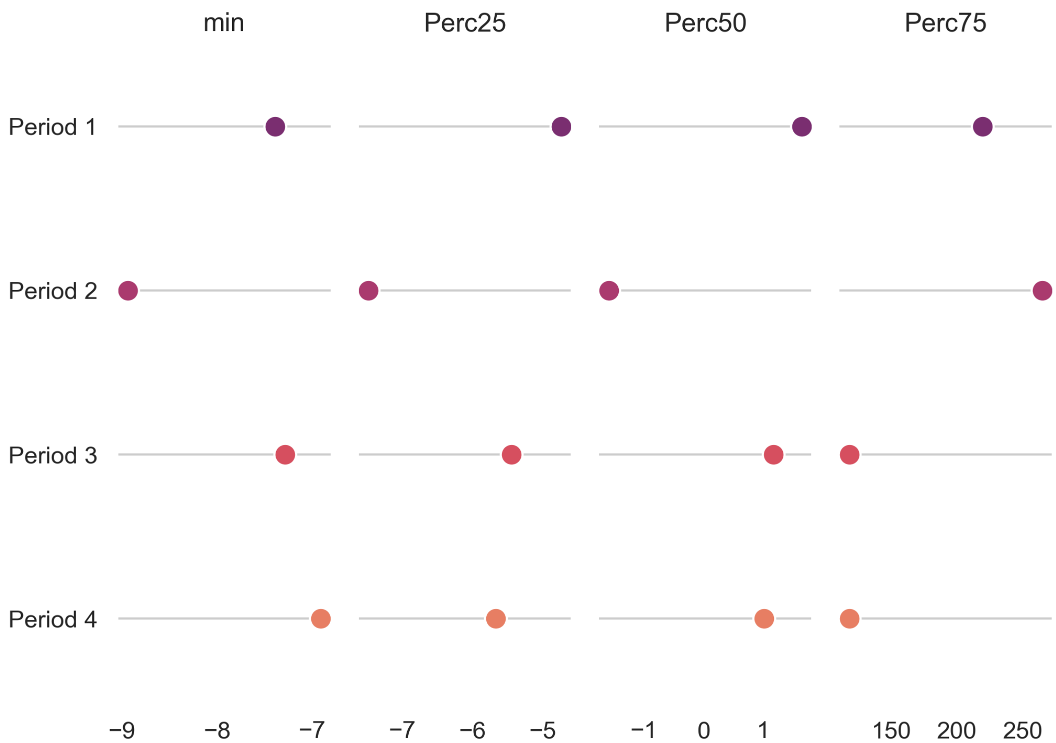

Figure 11: Mean Daily Percentage change by coin

11 percentile comparison between periods allows to note how period 4, although a subset of a ”bear market” 1, possesses less extreme coin percentage averages, and with the addition of daily balances as well as lower forecasting errors makes the interval an ideal investment period for the Open Strategy. Foremost, in the four periods it is the only one that possesses such proprieties at the same time.

5 Conclusions

This research was focused on approaching a portfolio optimization by using two different Genetic algorithm strategies (Single and Open). Before obtaining any results the expectation was that the approach of daily transactions outperformed the single transaction, instead, by introducing a multiple trading path, result is that it is dependent on the forecasting approach, requiring a high reliability for each time-step making it not compatible with the objective of this research to take an approach of an uniform and high level forecasting model, in which, for some periods this forecasting layout is likely not to produce errors small enough for a positive return, specially when considering the nuances of cryptocurrencies trading as well as the limitations on the variables provided to the model (only price historical).

Nevertheless, with the same layout and a single trading opportunity, results were mostly positive unless the period at hand has an overall downward trend. Considering all the possible configurations for each of the building blocks (forecasting, genetic algorithm configuration, testing periods) solving a cryptocurrency portfolio optimization with low complexity is by no means impossible, the limited trading approach proved to be a successful setup and with the parametrization of *Comb 4* (4.1.2).

Future research might want to test similar approaches with different forecasting layouts as well as complementary data sources, instead of limiting it to rely only on time-series historical, also by testing this approach on a broader securities mix (gold, bonds, stocks) and not with the limiting factor of a crypto only portfolio.

References

Alhomadi, A. (2021). Forecasting stock market prices: A machine learning approach. *Utah State University*.

Andreas M, A. (2019). (https://youtu.be/qlahxo-d-64) the five pillars of open blockchains.

Auffarth, B. (2021). *Machine Learning for Time-Series with Python*. Packt, 35 Livery Street Birmingham.

Burton G. Malkiel, E. F. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25:383–417.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., and Chang, C.-H. (2013). A hybrid approach by integrating wavelet-based feature extraction with mars and svr for stock index forecasting. *Decision Support Systems*, 54(3):1228–1244.

Kevin Martin, Izzat Alsmadi, M. R. M. A. (2020). Combining blockchain and machine learning to forecast cryptocurrency prices. *Second International Conference on Blockchain Computing and Applications (BCCA)*.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *www.bitcoin.org*.

Ritika Chopra, G. D. S. (2021). Application of artificial intelligence in stock market forecasting: A critique, review, and research agenda. *Journal of Risk and Financial Management*.

Smuts, N. (2019). What drives cryptocurrency prices? an investigation of google trends and telegram sentiment. *Association for Computing Machinery*, 46.

Tianqi Chen, C. G. (2016). Xgboost: A scalable tree boosting system. *Cornel University*.

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa