TÉCNICO LISBOA

Option Trading Based on Implied Volatility Forecasts using Genetic Algorithm

Francisco Sá Reis Machado e Costa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves
Member of the Committee: Prof. Joao Filipe Pererira Fernandes

September 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

This work proposes a financial strategy capable of trading options based on a implied volatility forecast. It firstly presents a new algorithm that forecasts implied volatility signals using two genetic algorithms. The first one uses technical indicators to forecast the direction of the implied volatility signals' movement, whereas the second optimizes the structure of the first one by finding the best configuration of its hyper parameters. The solutions were subsequently tested using a trading simulator, developed specifically for this work, that traded short and long positions of put and call options. Data from fifty different companies of the Standard & Poor's 500 (S&P 500) was used in the train and test phases, both from the time period between January 1st, 2011 and December 31st, 2015. Results show that implied volatility forecasts can be used to successfully trade options with profitable yields. Both long calls and short puts demonstrated to be good investment strategies.

Keywords

Option Trading, Volatility Forecast, Implied Volatility, Machine Learning, Genetic Algorithm, Technical Indicators, Technical Analysis

Resumo

Este trabalho propõe uma estratégia financeira capaz de transacionar opções baseada numa previsão de volatilidade implícita. Primeiro é apresentado um novo algoritmo que usa dois algoritmos genéticos para prever sinais de volatilidade implícida. O primeiro usa indicadores técnicos para prever a direção do movimento dos sinais de volatilidade implícita, equanto o segundo otimiza a estrutura do primeiro ao encontrar a melhor configuração dos seus hiper-parâmetros. A solução foi de seguida testada usando um simulador de transações, desenvolvido especificamente para este trabalho, que transacionou posições *short* e *long* de opções *put* e *call*. Na fase de treino e de teste usou-se dados de cinquenta empresas do S&P 500 do periodo entre 1 de Janeiro de 2011 e 31 de Dezembro de 2015. Os resultados demonstram que a previsão de volatilidade implicita pode ser usada para trasacionar opções com resultados rentáveis. Tanto *long calls* como *short puts* demonstraram ser boas estratégias de investimento

Palavras Chave

Comércio de Opções, Previsão de Volatilidade, Volatilidade Implicita, Aprendizagem Automática, Algoritmo Genético, Indicadores Técnicos, Análise Técnica

Acknowledgments

Aos meus pais, que me criaram e educaram, que garantiram que nada me faltou todos estes anos, mas principalmente que me ajudaram a ser a pessoa que sou;

À minha namorada, que sempre melhorou todos os bons momentos, sempre me consolou nos maus, e em nenhum deixou de me apoiar;

Ao meu orientador, que sempre transmitiu calma e pragmatismo, e sem o qual não teria feito esta tese;

A todos os amigos que fiz ao longo do meu curso, que, de uma forma ou outra, me ajudaram nos inúmeros desafios que foram surgindo. Principalmente à Ada, ao Mak e ao Corral, cuja amizade é das melhores coisas que este curso me deu;

À minha família, que embora não tenha escolhido, nunca trocaria;

E por fim aos Rapazes, que são para mim como família, e que hão de arranjar forma de se sentirem traídos por não serem os primeiros desta lista.

Obrigado por me terem ajudado e apoiado durante o meu curso e a minha tese. E acima de tudo, obrigado por estarem presentes.

Contents

List of Figures

Acronyms

ANN	Artificial Neural Network
NN	Neural Network
S&P 500	Standard & Poor's 500
S&P 100	Standard & Poor's 100
SPX	S&P 500 Index®
VIX	Volatility Index
VXF	Volatility Index Futures
CBOE	Chicago Board Options Exchange
OTCR	Open to Close Returns
HAR	Heterogeneous Autoregressive
ARCH	Autoregressive Conditional heteroskedasticity
GARCH	Generalized Autoregressive Conditional heteroskedasticity
EGARCH	Exponential Generalized Autoregressive Conditional heteroskedasticity
HAR_X	Augmented Heterogeneous Autoregressive
GA	Genetic Algorithm
RSI	Relative Strength Index
ROC	Rate of Change
StO	Stochastic Oscillator
MACD	Moving Average Convergence Divergence

XEMA Crossing Exponential Moving Averages

SMA Simple Moving Average

EMA Exponential Moving Average

ROI Return on Investment

SVR Support Vector Regression

1

Introduction

Contents

1.1 Overview

Financial markets have always attracted many investors in search of profits. Even though for many years stocks were the most traded financial instrument, along the years many derivatives ascended in popularity. One of those derivatives are options: contracts that give the buyer the right to buy/sell the underlying asset from/to the seller. Although options have the potential for higher percentage returns than stocks, they are also more complex financial instruments. An example of this increased complexity is the pricing of options. While other financial instruments follow the rule of supply and demand, option value has always been hard to determine, mostly relying on complex models to establish options prices.

The most used method of option pricing in financial markets is the Black-Scholes model. This model takes into factor seven parameters, and as mentioned in [1], the only one not directly observable from the market is the asset's implied volatility. As implied volatility has a direct correlation with an option's price, knowing the movement of one's value allows, even if incomplete, for a estimate of the movement of the other. Based on this idea an assumption was made: If one could make a forecast of a company's implied volatility, one could use this information to successfully trade options in the financial market

1.2 Work's purpose

This work aims to formulate a strategy capable investing in the financial marketing using options. In order to accomplish this, it firstly purposes to implement a machine learning algorithm that can forecast the movement of implied volatility signals. This machine learning algorithm will be divided in two genetic algorithms. The first will use technical indicators to compute a prediction of the implied volatility's behaviour, the second will find the first one's best hyper parameter configuration.

Secondly, a trading simulator will be developed in order to trade options of fifty companies of the Standard & Poor's 500 (S&P 500) during the period between 2011 and the end of 2015. The solution found by the machine learning algorithm will choose the best periods to open and close positions and a number of financial techniques will manage this trades to decrease investment risk.

1.3 Contributions

Bellow are the main contributions of this dissertation:

- The implementation of a genetic algorithm that, using technical indicators, forecasts the movements of implied volatility signals.
- Based on the assumption that a company's implied volatility has a direct correlation with its options prices, the use of implied volatility forecast to invest in options in the financial market.

- The implementation of a second genetic algorithm that improves the architecture of the first genetic algorithm by finding the best combination of its hyper parameters.

1.4 Document structure

This work is divided in five chapters as listed below:

- Chapter 1 (Introduction) starts by introducing the reader to the context in which this work exists. Then, the work's purpose is presented with a brief description of how the author proposes to achieve it. Finally a list of contribution is given and the overall structure of the document is presented.
- Chapter 2 (State-of-the-Art) firstly explains all the background concepts needed to fully understand this work implementation. Secondly, it elaborates on the related work, the studies and papers that contribute to this dissertation realization.
- Chapter 3 (Methodology) presents the program's architecture developed for this work.
- Chapter 4 (Results and Discussion) displays the results of the program. The case studies elaborated in chapter 3 are analysed and compared.
- Chapter 5 (Conclusion) presents the conclusions extracted from the work's result and suggests several changes and improvements for future works.

2

State-of-the-Art

Contents

2.1 Overview

This section explores key concepts for the overall comprehension of this master thesis. In a first instance, some background concepts necessary for the full understanding of this thesis are provided, alongside with the explanation of both the stock and option market and the chosen machine learning algorithm. Finally, it is presented a review regarding theoretical and practical studies under the subject of this work.

2.2 Background Concepts

2.2.1 Stocks and Derivatives

Stocks

When a company becomes publicly-held it can sell a portion of its ownership. To this security it is given the name stocks, also known as shares or equities, and it can then be sold privately or publicly in the stock market. [2]

When buying or selling stocks one can ”go long” or ”go short”. In the first scenario, a stock is purchased with the expectation that its value will increase. When this happens the stock buyer can sell the stock to profit from the stock value increase. On the other hand, going short or shorting occurs when one believes that the stock value will decrease, selling a stock (that the individual doesn't necessarily need to have, and can thus be seen as a lending) and afterwards buying a stock from the market.

When someone believes that the value of a stock will increase, one can say that this individual has a bullish sentiment, while if the belief is that the value will decrease is said to have a bearish sentiment.

Stock Market

The Stock Market, also known as Equity Market, is the collection of all the stock exchanges in the world which in turn are single markets where stocks can be bought and sold under certain regulations. Furthermore these exchanges also can trade commodities, currencies, bonds and derivatives based on stocks. [2]

Stock Splits

Stocks can only be bought in unitary amounts. for this reason whenever a stock's price increases, the respective company has the ability to do a stock split. Stock split is the name given to when a company divides its existing stocks, and doing so, increases the number of shares and decreases their price. For example, a company with 10.000 shares with a unitary value of 800$ decides to do a stock split with the ratio 2:1. This means that it divides its 10.000 shares into 20.000. The price of each share, as the total

value of the company didn't change, will decrease to 400$. Even though some ratios of stock splits tend to be more used than others, each company can choose the stock split's ratio that achieves the desired share value. For example, a company with 37.000 shares, each going for 258$ in the stock market, can apply a 258:100 ratio to achieve a share price of 100$. This would increase the company's number of shares to 95.460.

Derivatives

Derivatives are financial instruments whose returns are derived from an asset. This is, the return of the derivative is a function of the underlying asset's value such as: $R_D = f(V_A)$, where R_D is the return of the derivative D, V_A is the value of the underlying asset A and f is the function that relate the return of the derivative with the value of the underlying asset.

These assets can be of many types such as stocks, market indexes and currencies. Futures and options are example of derivatives. One can see derivatives as contracts between two or more parties in respect to an asset. [2] [3]

2.2.2 Options

One of the most common derivatives in financial markets are options. These are contracts, with an expiration date T also known as maturity, that give the buyer (contrary to futures), the opportunity (but not the obligation) to buy (call options) or sell (put options) the underlying asset for a certain price K, called exercise price or strike price.

Exercise is the name given to the action of actually using the option to buy/sell an asset, and the price for which an option is bought is called the premium of the option. [2] [3] [4]

Regarding the time when an option can be exercised ,there are two different possible situations to take into account:

- European options can only be exercised in the expiration date, i.e. the buyer of the option can only decide to buy/sell the underlying asset in the last day of the contract.
- American options can be exercised any time before the expiration date

Call Options

Call options are contracts that give its buyers the option of buying an asset until/at the expiration date. When exercising this option one would buy the stock for the agreed strike price and sell it in the market at the current market value. The profit would then be the stock market price minus the strike price and the option premium. As the option buyer makes profit if the stock market value increases one can say that who buys a call option has a bullish sentiment, while the call option seller has a bearish sentiment (since there is a belief that the stock market value will increase).

The big difference between buying a call option and buying directly the underlying asset is that, as we can see in figure 2.1, even though the maximum return of both these instruments is theoretically infinite (when the value of the underlying asset tends toward infinity), the maximum loss of buying an asset is much bigger than the maximum loss of buying the correspondent call option.

The maximum loss when buying a stock is the value of the stock at the time of purchase while the maximum loss of buying a call option is the premium of the option which is considerably lower. These outcomes happen when the value of the stock becomes zero (when a company goes bankrupt for example) and when the option is not exercised (when, until/at maturity, the value of the underlying asset is lower than the strike price). [2] [3] [4] [5]

For example, given the following situation:

- $V_\tau = 100\$$ is the Value of a stock S at time τ;
- $K = 120\$$ is the strike price of a call option relative to the stock S;
- $P_c = 5\$$ is the price of the call option at τ;
- T is the expiration date of the call option;
- For sake of simplicity the option in question is an european option (can only be exercised at the expiration date);
- The return of the call option R_c if exercised is $R_c = V_T - K - P_c$. ($R_c = -P_c$ otherwise);
- The return of the stock is $R_s = V_T - V_\tau$,

The individual *A* buys a stock at τ for $100\$$, at the same time the individual *B* buys a call option for $5\$(120\$$ strike price):

First situation: At T the value of the stock is $200\$$. In this case $R_s = 100\$(200\$ - 100\$)$ and $R_c = 75\$(200\$ - 120\$ - 5\$)$.

Second situation: At T the value of the stock is $150\$$. In this case $R_s = 50\$(150\$ - 100\$)$ and $R_c = 25\$(150\$ - 120\$ - 5\$)$.

Third situation: At T the value of the stock is $125\$$. In this case $R_s = 25\$(125\$ - 100\$)$ and $R_c = 0\$(125\$ - 120\$ - 5\$)$.

Forth situation: At T the value of the stock is $80\$$. In this case $R_s = -20\$(80\$ - 100\$)$ and $R_c = -5\$)$ (option not exercised).

Fifth situation: At T the value of the stock is $0\$$. In this case $R_s = -100\$(0\$ - 100\$)$ and $R_c = -5\$)$ (option not exercised).

In the first two situations the option is ”in the money”, this means that one can exercise it and gain profit since the stock market value is bigger than the option strike price plus the option premium. As figure 2.1 illustrates, when a call option is in the money the profit difference between buying the option or the underlying stock is only the premium value plus the difference between the option strike price and the value of the stock at τ ($25\$$ on this example).

In the third situation, exercising the option would not bring neither profit nor loss. To this situation is called being ”at the money”. The difference in profit from call option buying and stock buying is still the option premium value plus the difference between the option strike price and the value of the stock at τ

In the last two situations the stock value is lower than the option strike price plus the option premium. As such, exercising it would only bring a bigger loss. In this situation the option buyer would not exercise it and his loss would just be the option premium no matter how big the stock value decrease is. On the other hand we can see, using the same example and as figure 2.1 illustrated, that the loss of the stock buyer would be as big as the stock value drop with a maximum loss if the stock value became zero.

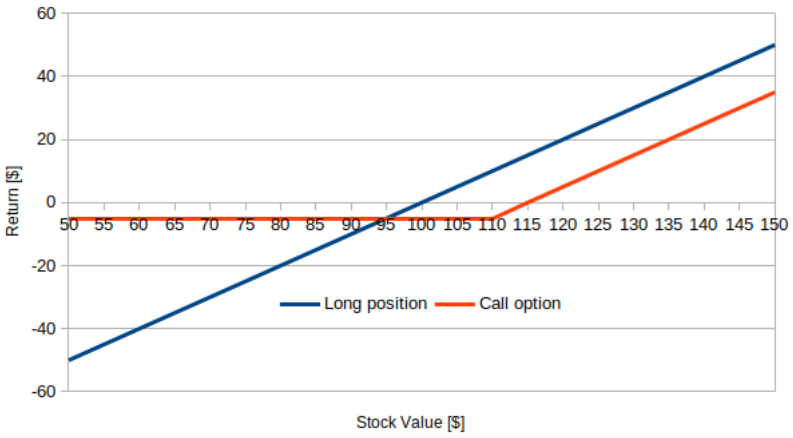

Figure 2.1: Call option returns vs. stock long position returns

Put Options

Put options are contracts that give its buyers the option of selling an asset until/at the expiration date. When exercising this option, one would buy the stock in the market at the current market price and sell it to the option seller at strike price, and the profit would then be the option strike price minus the stock market value and the option premium. As the option buyer makes profit if the stock market value decreases, one can say that the one who buys a put option has a bearish sentiment, while the put option seller has a bullish sentiment (that is, with the belief that the stock market value will decrease)

The big difference between buying a put option and going short on the underlying stock is that, as we can see in figure 2.2 , even though the maximum return of both these instruments is finite (when the value of the underlying stock becomes zero), the maximum loss of going short is much bigger than the maximum loss of buying the correspondent put option.

The maximum loss when going short on a stock is theoretically infinite, as in theory the value of the stock can increase to infinity while the maximum loss of buying a put option is the premium of the option-which is considerably lower. The second outcome happens when the option is not exercised (that is, when, until/at maturity, the value of the underlying asset is bigger than the strike price). [2] [3] [4] [5]

For example, given the following situation:

- $V_\tau = 100\$$ is the Value of a stock S at time τ;
- $K = 80\$$ is the strike price of a put option relative to the stock S;
- $P_p = 5\$$ is the price of the put option at τ;
- T is the expiration date of the put option;
- V_T is the value of a stock S at time T;
- For sake of simplicity the option in question is an european option (can only be exercised at the expiration date);
- The return of the put option R_p, if exercised, is $R_p = K - V_T - P_p$. $-P_p$ otherwise;
- The return of the stock is $R_s = V_\tau - V_T$,

The individual *A* sells a stock at τ for $100\$$ (goes short), at the same time the individual *B* buys the put option for $5\$$ ($80\$$ strike price):

First situation: At T the value of the stock is $200\$$. In this case $R_s = -100\$(100\$-200\$)$ and $R_p = -5\$$ (option not exercised).

Second situation: At T the value of the stock is $150\$$. In this case $R_s = -50\$(100\$ - 150\$)$ and $R_p = -5\$$ (option not exercised).

Third situation: At T the value of the stock is $75\$$. In this case $R_s = 25\$(100\$ - 75\$)$ and $R_p = 0\$(80\$ - 75\$ - 5\$)$.

Forth situation: At T the value of the stock is $35\$$. In this case $R_s = 65\$(100\$ - 35\$)$ and $R_p = 40\$(80\$ - 35\$ - 5\$)$.

Fifth situation: At T the value of the stock is $0\$$. In this case $R_s = 100\$(100\$ - 0\$)$ and $R_p = 75\$(80\$ - 0\$ - 5\$)$.

In the first two situations the stock value is bigger than the option strike price minus the option premium and so exercising it would only bring a bigger loss. In this situation the option buyer would not exercise it and his loss would just be the option premium no matter how big the stock value increase is. On the other hand, and analysing figure 2.1, one can say that the loss of going short on the stock would be as big as the stock value increases with no maximum ceiling.

In the third situation exercising the option would not bring neither profit nor loss. To this situation is called being ”at the money”. The difference in profit from put option buying and stock shorting is still the option premium value plus the difference between the value of the stock at τ and the strike price

In the last two situations the option is ”in the money”, meaning that one can exercise it and gain profit since the stock value is lower than the option strike price minus the option premium. As we can see in this situation and in figure2.1, when a put option is in the money the profit difference between buying the put option or going short on the underlying stock is only the premium value plus the difference between the value of the stock at τ and the option strike price ($25\$$ in this example).

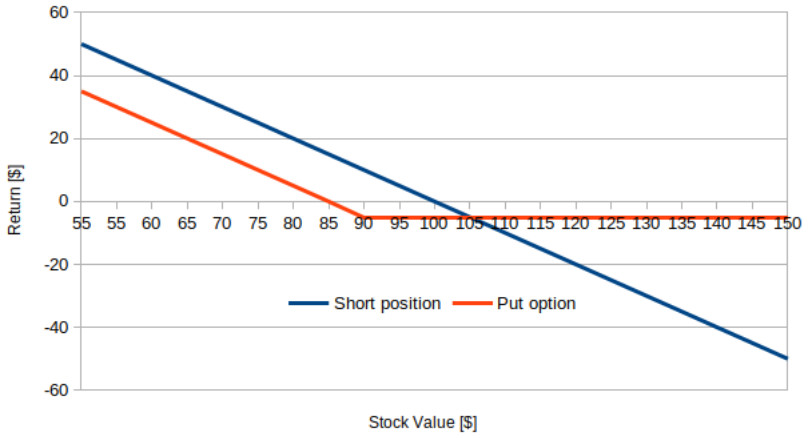

Figure 2.2: Put option returns vs. short position returns

2.2.3 Option Pricing

As options are a very complex financial instruments, rightfully evaluating its value is also a non elementary task. Compared for example to stock prices, which are only binded by supply and demand, option prices are influenced by seven different factors, enumerated below [6]:

- Type of option (Call or Put),
- value of the underlying asset,
- strike price K,
- time to maturity $T - \tau$ (time between the present day and the expiration date),
- risk free interest rate,
- stock volatility,
- dividends.

Having such a variety of constraints, option pricing is a financial area in continuously development. From empirical estimation to binomial and trinomial models, Black and Scholes's was the first model recognized as being reliable for option pricing [2] [6]

2.2.4 Greeks

Greeks, which have this name because they are usually represented by greek letters, are quantities that measure the sensitivity of a derivative price to a parameter of its underlying asset, with the assumption that all other parameters remain the same. Even though they have this incorrect assumption, greeks are a powerful tool to evaluate the change of risk of derivatives such as options. [2] [4] [7]

Given a derivative instrument Π with an underlying asset S:
The Delta of a derivative reflects the sensitivity of its value to changes in the value of underlying asset:

$$\Delta = \frac{\delta \Pi}{\delta S} \tag{2.1}$$

As the values of Π and S are directly proportional, Δ can be seen as the constant of proportionality. Call option value ranges between 0 and 1 whereas put options ranges between -1 and 0. As the expiration date approaches, the delta of an in the money call option tend to 1 and of an out of the money call option tend to 0. While the delta of an in the money put option tend to -1 and of an out of the money put option tend to 0.

The Gamma of a derivative reflects the sensitivity of its Delta with respect to changes on the value of the underlying asset. It is given as:

$$\Gamma = \frac{\delta^2 \Pi}{\delta S^2} = \frac{\delta \Delta}{\delta S} \tag{2.2}$$

For example, at the money options have higher Gammas than in- or out-of-the-money ones and the closer to maturity an option gets, the higher its Gamma tends to be. The high Γ in these situations means that the derivative's Δ can change drastically with the change of the underlying asset price.

The Theta of a derivative reflects the sensitivity of its value in relation to time:

$$\Delta = \frac{\delta \Pi}{\delta T} \tag{2.3}$$

Theta can be seen as a time decay. Assets like stocks have a Θ of zero while options tend to have a negative Θ meaning that its value is always decreasing (while all else being equal). The closer the option is to its maturity the bigger its Theta tends to be. An option with Θ = -0.40 would have its value decreasing 40 cents per day.

The Vega of a derivative reflects the sensitivity of its value with respect to changes in the volatility of the underlying asset (σ_S):

$$\Delta = \frac{\delta \Pi}{\delta \sigma_S} \tag{2.4}$$

Options have always a positive Vega, with american options having a higher Vega value than european ones as they can be exercised at any moment. A higher volatility in a stock makes the corresponding american options more probable to become in the money somewhere before maturity.

The Rho of a derivative reflects the sensitivity of its value with respect to changes in the risk-free interest rate (σ_r):

$$\rho = \frac{\delta \Pi}{\delta r} \tag{2.5}$$

The Rho can be seen as a percentage between the interest rate and the derivative value, as a ρ=1 would mean that a 1$\%$ increase in the interest rate of the underlying asset would increase the value of the derivative in 1$\%$

2.2.5 Black & Scholes Model

Black and Scholes model is the most well known model for option pricing. Published in 1973 by Fischer Black and Myron Scholes [5] [1], this model, as can be seen in 2.6 to 2.9, relates the value of an option premium with its strike price, time to maturity, interest rate and its underlying asset's value and standard deviation (volatility) [4]:

$$C = SN(d_1) - Ke^{-r(T-t)}Nd_2 \tag{2.6}$$

$$P = Ke^{-r(T-t)}N(-d_2) - SN(-d_1) \tag{2.7}$$

Where d_1 and d_2 can be written as:

$$d_1 = \frac{ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}} \tag{2.8}$$

$$d_2 = d_1 - \sigma\sqrt{(T-t)} = \frac{ln(S/K) + (r - \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}} \tag{2.9}$$

with:

- K is strike/exercise price of the call option,
- t is the time remaining until the expiration date (as a fraction of a year),
- S is the value of the underlying stock,
- r is the short-term risk-free interest rate,
- σ is the standard deviation of the underlying stock price,
- $N()$ is a cumulative normal probability,
- T is the exercise date,
- t is the current date.

.

In this model there is one not directly observable parameter - the asset volatility [2]. In the standard form of this model, the historical volatility is used. But, as a perfect volatility forecast is not possible, this method of option pricing is as accurate as the volatility estimator is precise.

Another application of this model is as volatility estimator. It can be altered to have as a parameter an option market price in order to extract the volatility implied by that option (implied volatility) [7].

2.2.6 Stock Volatility

Volatility is an asset characteristic that represents the fluctuation in its price. An high volatility means that the asset's price has a big probability of having a big fluctuation in either direction whereas a low volatility, means that the asset price has a lower probability of having a big fluctuation in either direction [7].

Implied Volatility

Implied volatility is a stock volatility estimate derived from the market prices of that stock's options. As mentioned in 2.2.5, one needs to back solve the Black and Scholes model with the option premium extracted from the market. The obtained volatility will be the one "implied" by those prices [7].

Historical/Realized volatility

Historical volatility, also known as realized volatility, is a volatility estimate derived from past events. The most common method of obtaining it is to compute the standard deviation of the last 21 to 23 days stock closing prices [7].

2.2.7 VIX

Chicago Board Options Exchange (CBOE) created the Volatility Index (VIX) in order to better represent the market volatility. Even though there are several VIXs, like 9DVIX (9 day expected volatility), 3MVIX (3 moths expected volatility), 6MVIX (6 month expected volatility) and 1YVIX (1 year expected volatility), whenever someone mentions VIX, is most likely referring to 3MVIX (3 month expected volatility).

In 30DVIX S&P 500 Index® (SPX) monthly and weekly options expiring in 23 to 37 days are used to compute value that translates very roughly into the expected percentage of change in S&P 500 index value over the next 30-day period (annualized).

For example, if the VIX is 15, this represents an expected annualized change of 15% which equates to a 1.25% change up or down for the S&P 500 over the next 30-day period. [8]

In order to calculate the 30DVIX one should first get the weekly and monthly SPX option with more than 23 days and less than 37 days. This narrows the options into two clusters as there is only two dates with weekly and/or monthly options in this interval. As we can see in figure 2.3, if the present day is February 10th, only the options of March 8th and 15th would be considered, being the near-term and next-term options, respectively. Once per week there is a shift and the next-term options becomes the near-term options, and there is a new next-term date. In this example March 15th and 22th will become the new near-term and next-term date, respectively.

February 2019

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

March 2019

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Monthly options Weekly options

Figure 2.3: Monthly and weekly options in February and March 2019

From these two groups are only used the out-of-the-money calls and puts centered around an at-the-money strike price K_0 (one for near-term options and one for next-term options).

Choosing the strike price with the lowest difference between the call put premium, the forward index prices, F_{near} and F_{next} are calculated using: $F = strike\_price + e^{RT} * diff_{call-put}$.

$K_{0_{near}}$ and $K_{0_{next}}$ are the strike prices equal or immediately bellow the corresponding F values. The only considered options are the puts with strike prices below K_0 and calls with strike prices above K_0. Starting from K_0 outwards all options with bid=0 should be excluded, and when two consecutive options have bid=0 no more options in that directions are considered.

For each strike price the midpoint price $Q(K_i)$ is calculated, which is the average between the bid and ask prices of the put($K < K_0$) or call($K > K_0$) option. For K_0 the midpoint price is the average between its put and call options midpoints.

ΔK_i is, for all strike prices except the limits, the average between the adjacent strike prices: $\frac{Ki-1+Ki+1}{2}$. For the limit options, this is the difference between K_i and the adjacent strike price.

The $\sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i)$ part of the equation 2.10 is the contribution of each option to the overall VIX value. The further away from K_0 an option strike price is, the smaller its contribution will be.

The final two steps are applying the already computed data into:

$$\sigma^2 = \frac{2}{T} \sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i) - \frac{1}{T} \left[\frac{F}{K_0} - 1 \right]^2 \tag{2.10}$$

Where:

- T: Time to expiration: $\text{minutes}_{\text{to\_expiration}}$ / $\text{minutes}_{\text{in\_a\_year}}$,
- F: Forward index level derived from option prices,
- K_0: The first strike price below the forward index level, F,
- K_i: The strike price of the i-th out-of-the-money option,
- ΔK_i: Interval between strike prices,
- R: Risk-free interest rate,

- $Q(K_i)$: Midpoint price of the bid-ask spread for each option with strike K_i.

And finally:

$$VIX = 100 * \sqrt{\left\{ T_1 \sigma_1^2 \left[\frac{N_{T_2} - N_{30}}{N_{T_2} - N_{T_1}} \right] + T_2 \sigma_2^2 \left[\frac{N_{30} - N_{T_1}}{N_{T_2} - N_{T_1}} \right] \right\} * \frac{N_{365}}{N_{30}}} \tag{2.11}$$

Where:

- T: Time to expiration: $\text{minutes}_{\text{to\_expiration}}$ / $\text{minutes}_{\text{in\_a\_year}}$,
- N_{T_1}: number of minutes of near-term-date to expiration,
- N_{T_2}: number of minutes of next-term-date to expiration,
- N_{30}: number of minutes in 30 days (43,200 min),
- N_{365}: number of minutes in a 365-day year (525,600 min),
- $_1$ refers to near-term calculations and $_2$ to next-term calculations.

2.2.8 Financial Analysis

In order to better predict financial signals, traders usually study and evaluate feature-like signals called indicators whose behavior can be studied in order to find trends in these financial signals. To the use of these indicators is given the name of indicator analysis [7].

There are two distinct types of indicators: technical and fundamental. With technical analysis the indicators are computed from the historical data of the subject with forecasting relevance. Different formulas can be applied to the original signal in order to compute different technical indicators.

Fundamental analysis, on the order hand, uses factors that can correlate to the movement of the signal one is trying to forecast [7]. For example, the number of cars sold yearly can be used to evaluate a car manufacturing company growth. Even though this is considerate to be a very powerful type of financial analysis, the fact that each fundamental signal must be acquire from each company, makes fundamental analysis harder to use than technical analysis.

2.2.9 Volatility Forecasting Models

ARCH model

Autoregressive Conditional heteroskedasticity (ARCH) Is considerate to be the first model for forecasting mean returns of an asset. Based on rolling standard deviation, it calculates the standard deviation of the next day using a weighted average of squared standard deviation values from a fixed number of past observations. Each observation has its own respective weight that can be determined independently from the others in order do achieve a more correct variance forecast. [9]

GARCH model

"Generalized Autoregressive Conditional heteroskedasticity (GARCH) is a generalization of the ARCH model in which there isn't a fixed number of past observation. The most used GARCH specificaion, GARCH(1,1), takes in consideration three factor: The weighted average of the long-run average varience, the forecasted variance for the present observation and the present observation's return.

The return of an asset on a present time t can be computed as:

$$r_t = m_t + \sqrt{\sigma} * \epsilon_t, \tag{2.12}$$

where r_t is the return of the asset, m_t is its mean value, ${\sigma^2}_t$ is its variance and ϵ is the error for the present observation.

The GARCH(1,1) model for forecasting a future variance of time $t+1$ can be then written as:

$${\sigma^2}_{t+1} = \omega + \alpha(r_t - m_t)^2 + \beta{\sigma^2}_t = \omega + \alpha{\sigma^2}_t{\epsilon_t}^2 + \beta{\sigma^2}_t, \tag{2.13}$$

where ${\sigma^2}_{t+1}$ is the forecasted variance and ω is the long run variance. The more generalized GARCH model is GARCH(p,q),

$${\sigma^2}_t = \omega + \sum_{i=1}^{q} \alpha_i (r_{t-i} - m_{t-i})^2 + \sum_{j=1}^{p} \beta_j {\sigma^2}_{t-j} = \omega + \sum_{i=1}^{q} \alpha_i {\sigma^2}_{t-i}{\epsilon^2}_{t-i} + \sum_{j=1}^{p} \beta_j {\sigma^2}_{t-j}, \tag{2.14}$$

2.2.10 Genetic Algorithm

As stated by David A. Coley in [10], p.1, *"Genetic algorithms (GAs) are numerical optimisation algorithms inspired by both natural selection and natural genetics."*. Genetic Algorithm (GA) are inspired, as David A. Coley says, in the Darwinism theory of biological evolution. This theory states that living beings evolve by passing its genes to their offspring (reproduction). It also states that some genes can make a specimen more adapted, or fit, to a specific environment than others. This advantage would increase its change of surviving and reproduce, leading to the passage of its offspring. With time better combinations of genes will be found. This is also the foundation of GA's.

Every GA have the same four fundamental characteristics:

1. a number of possible solutions for the problem, called population,
2. a method of evaluate how fitting each of these possible solution is,
3. a way of mixing elements from the population best solutions in order to get a new population,
4. a mutation factor to increase diversity and avoid local maximums.

If we look to nature is easy to find the inspiration from where these characteristics were taken from.

1. in any species there is a population of elements,
2. each of element of the population is "evaluated" by natural selection. The elements that survive in nature are "the best solutions" for survival which is the ultimate problem that needs finding a solution for,
3. through sexual reproduction two elements of the population mix their genome in order to create a new element with a combination of their characteristics. Asexual reproduction happens when a specimen creates an offspring with its exact genome. This also happen in GA when an element of a generation x continues to the generation $x + 1$,
4. whenever a new element is created, mutations in its genome occurs. That's how the first blond person got its blond hair but also how the cases of down syndrome occur.

In figure 2.4 all steps of a standard GA are displayed. These will be explained in the following sections.

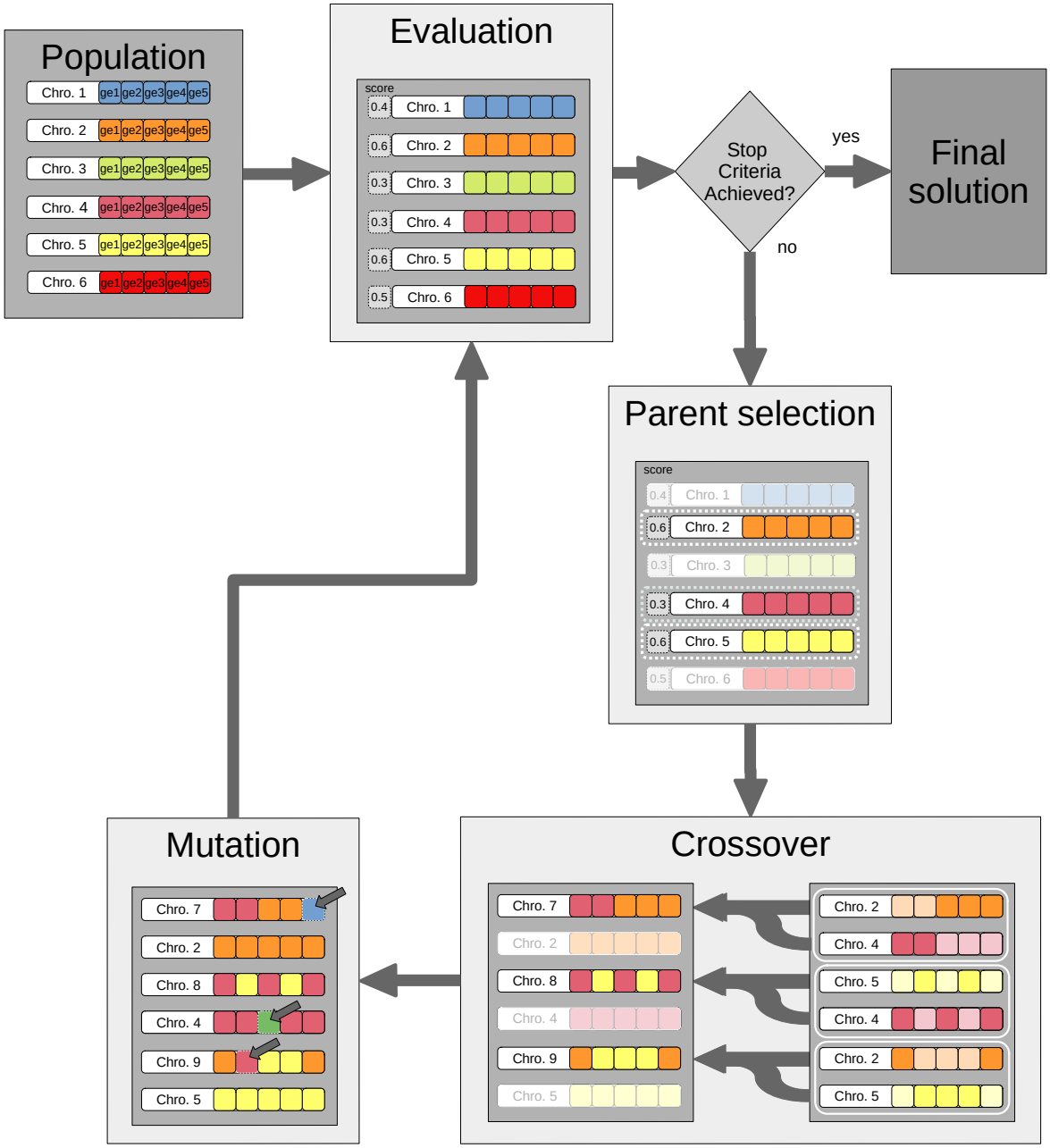

Figure 2.4: Diagram of a Genetic Algorithm

Population

First there is an initial population, where each element is called chromosome and is a possible solution to the problem. Each of these chromosomes have a number of genes, equal to the solution space, that correspond to the variables of the problem. For example, if one should use a GA to find a good linear regression of a set of data, each chromosome would have two genes, one for the values of a and one for the values of b in $y = a + bx$.

In the example of figure 2.4 the population is constituted by 6 different chromosomes, each one of them with 5 genes.

There are different methods of creating the first population [11]:

- Pseudo-random: the population is created with a random method (called pseudo as it is impossible to create a truly random number);
- Quasi-random: creates a population with consideration not only to the independence between elements (pseudo-random), but also to their coverage of the search space;
- Sequential diversity: the population is uniformly distributed throughout the search space;
- Parallel diversification: The search space is divided in a number of blocks equal to the number of elements of the population. Within each block a pseudo-random method is used to create an element;
- Heuristic initialization: uses an heuristic function to create the first population with the drawback of a premature convergence.

Evaluation

After creating a population, this one goes through the process of evaluation. Here each chromosome is given a score corresponding to how fitting its solution is to the problem. The evaluation is different for every problem and so is the score measure. In this example the score is given in a scale of 0 to 1.

Stopping criteria

In order to designate the final solution, there must exist a stopping criteria. Which can assume one of two possible types: Static or Adaptive [11]. In a static stopping criteria, the end of the search is known before hand (for example, by using a fixed number of iterations, time or computational resources). Using adaptive stopping criteria makes the algorithm search duration unknown (for example, by using a comparison the score of each element to a predetermined value and having the algorithm stop after a number of iterations with no progress).

Parent selection

The next step is to select the parents to proceed to the next population and to create new chromosomes. The chromosomes are assign a fitness value using either a *proportional fitness assignment* which is an absolute fitness value like the evaluation score or a *rank-based fitness assignment* which consists in a relative fitness like the position of the chromosomes after sorting them by their evaluation score.

After giving a fitness value to each chromosome a selection method must be applied. Even though there are countless methods two of the most used ones are the *Roulette wheel selection* and the *Tournament selection* [11]:

The *Tournament selection*, for example, consists in randomly choosing *n* elements of the population into a "tournament" group. From this group the most fittest chromosomes are chosen to reproduce. Finally one can just select the individuals with the highest evaluation scores. Even tough this seems the most logical approach, its emphasis in the fittest elements can lead to a local maximum due to a lack of diversity.

Crossover

After having selected the chromosomes that will create new offspring, one can apply one or more crossovers. These are methods of creating the offspring genes based on the parents' genes [11]. The following are some examples of crossover methods:

In the *Intermediate Crossover* each offspring gene is a weighted average of the two parents corresponding genes. this can be written as follows: $G_{off} = (x) * G_{p_1} + (1 - x) * Gp_2$, where G stands for gene, off for offspring, p_1 for parent 1 and p_2 for parent 2. The weight x can be independent between genes.

The *Geometric Crossover* consists in giving to each offspring gene the value of the square root of the two parents gene multiplication. $G_{off} = \sqrt{G_{p_1} * G_{p_2}}$

Using *Two-Point Crossover* one should pick two points at random. The genes between this two points are taken from the first parent and the rest of the offspring genes are taken from the other parent outside bounds of this two points.

One final example is to, for each offspring gene, picking at random the corresponding gene from either of the two parents.

Even though only four methods were exemplified here, one should take into account that many other methods exist. [11]

Replacement Strategies

There is a step, between the Crossover and Mutation blocks, that is not represented in figure 2.4: the replacement phase. It's in this step that is decided which chromosomes get to make part of the next population. There are two strategies [11]:

The *Generational Replacement*, where the new population will be comprised of the offspring created in the crossover phase. This means that none of the selected parents in the parent selection phase continue to the next generation.

The second strategy is the *Steady-State Replacement*. Contrary to the previous strategy, some of the chromosomes of the current population will make part of the new population. This number can vary from 1 to $N-1$, where N is the population size. Additionally the chromosomes that proceed from one generation to the next can be pick using one of the may parent selection methods from either the current parent group or the whole current population.

Mutation

The final step in a GA iteration is the mutation phase. At this stage, each chromosome of the new population has a probability p_c of suffering a mutation. Within the mutated chromosome each gene has a probability p_g of changing its value with a minimum of one gene being picked. These two probabilities must be carefully chosen as too small probabilities would not diversify enough the population and may lead to a local maximum, but too high probabilities might turn impossible for the algorithm to converge [11].

The amount of change must also be chosen. Even though a mutated gene must be able to reach every value of that variable range, its mutation should be local, meaning that the change in the value should be small compared to its range. For example a gene which value ranges from 1 to 50 and has the current value of 10 should be more likely to mutate to the value 15 than to 50. Hence, the most usual strategy is to apply a Gaussian probability with mean in the current value of the gene and a standard deviation σ that must be small and carefully chosen.

2.3 Related Work

After looking at the background concepts, this section focuses in the research made for this work. Firstly, papers gathered for the option pricing problem are presented. Then, and because of the connection between volatility and option value, some works on volatility are introduced. These works also reflect the introduction of machine learning models on the volatility forecast theme.

2.3.1 Option pricing research

Many models have been created over the years to better evaluate options value. The most widely used is the Black-Scholes model, first published in 1973 [5]. This formula takes into consideration several factors that influence an option value [6]. As explained in [1] the first factor the authors took into consideration was the underlying stock volatility. This is, of the seven factors, the only one not measurable from the market which makes forecasting volatility extremely important to forecast option value

2.3.2 Volatility research

Some authors have theorized a correlation between implied volatility and other volatility related signals. In [12], the authors theorize that historical volatility can be used to forecast implied volatility. A set of Grager non-causality models, was estimated between three volatility measures (twenty-day rolling standard deviation, intraday standard deviation and intraday high-low range) and VIX data for twenty-three securities. this models are statistical hypothesis tests created to determine whether the forecast capability a signal has on another. The authors used VIX to represent implied volatility of the american stock market and doing it so, concluded that both the rolling standard deviation and intraday high-low range show a great potential for volatility forecasting.

VIX is a signal developed by CBOE in 1993 to measure the expected market 30-day implied volatility using Standard & Poor's 100 (S&P 100) option prices [8]. In 2003 CBOE changed this signal to start using Standard & Poor's 500 (S&P 500) option prices, and to this day is the most used signal to represent the overall market volatility.

Many researchers tried to introduce VIX into volatility models. The authors of [13], for example, used a modified Heterogeneous Autoregressive (HAR) model to prove that VIX plays an important role in volatility forecasting. This modified HAR model consists in adding VIX into the original model. They also used the same method with "Large VIX" which is a signal that takes either the value of VIX if its value is greater than the average value of VIX from the previous 30 days or zero otherwise. Both these modified models have been applied to 13 markets of the G20 and led to better results than the original HAR model, confirming a potential role of VIX in volatility forecasting.

Other researchers have compared the forecast capability of machine learning algorithms to the of volatility models. In [14] the authors model the Volatility Index Futures (VXF) dynamics using a multi-layer augmented feed-forward Neural Network (NN). The also compare the NN's VXF Open to Close Returns (OTCR) predictions with those yielded by a logistic specification, a Naive model that always forecasts negative VXF OTCRs, a HAR model, and two Augmented Heterogeneous Autoregressive (HAR_X) models. Using Their work shows that the NN outperforms all other models.

The authors of [15] tried a different approach. Instead of comparing machine learning models to volatility models like in [14], their approach focused in using NN to improve the forecast capability of GARCH models. When applied to the three Latin-American stock markets (BOVESPA from Brazil, IPSA from Chile and IPyC from Mexico.), the results showed that the NN could increase the forecasting capabilities of the GARCH model.

Besides being used to improve other volatility models, machine learning algorithms have been shown to be capable of forecasting implied volatility signals and in some cases outperform these hybrid models. This is the case of [16] where the authors introduced a machine learning model comprised of a Gradient Descent Boosting, a Random Forest and a Support Vector Machine stacked with a NN. The results suggested that this Stacked-NN has a better forecasting capability when compared with other hybrid models like ANN-GARCH and ANN-EGARCH.

Another machine learning algorithm that shows great potential in volatility forecasting is GA's. In [17] the authors apply a GA to the Black-Scholes model to find implied volatility values. The results show that GA's outperforms the Newton-Raphson method.

In [18] on the other hand, the authors used a GA to optimize the parameters of Support Vector Regression (SVR). This hybrid approach was compared to a SVR and a GARCH model. The results show that the first outperformed the last two in implied volatility forecasting, demonstrating the usefulness of GA's in hybrid models.

3

Methodology

Contents

3.1 Overview

In this chapter the overall methodology will be presented. Firstly, a description of the used data is given and the corresponding methods of acquisition is presented. Then the processes to which this data were subjected are explained. An overview of the training phase is the given, which includes the structure of the two genetic algorithms. The chapter then terminates with the test phase presentation that consists on the functionality of the trading simulator developed for this work.

3.2 Structure

The structure of this program, as can be seen in figure 3.1, is divided into four segments: Data acquisition; data processing; training phase; and test phase.

In the first phase the raw data must be obtained, in this case from different sources. It is important to acquire data within the same time interval. Technical indicators are then extracted from each of these raw signals, to be used as input signals in the machine learning algorithm. The third phase is to train the system in order to obtain the fittest solution, this is, the combination of weights of each of the technical indicators that better forecasts the movement of companies' implied volatility . Using the solution from the training phase, the test phase consists of evaluating the performance of the proposed solution in a market simulator.

The first and second phases are sequential but the third and fourth are not. This last two phases are in fact cyclical as there are in total three training phases that are always followed by a corresponding test phase. The figure 3.4 Shows how these three train/test phases combination are structured. Each train phase has a duration of two years and each test phase has a duration of a single year. After each complete cycle, a one year shift is applied in the new cycle's train and test periods.

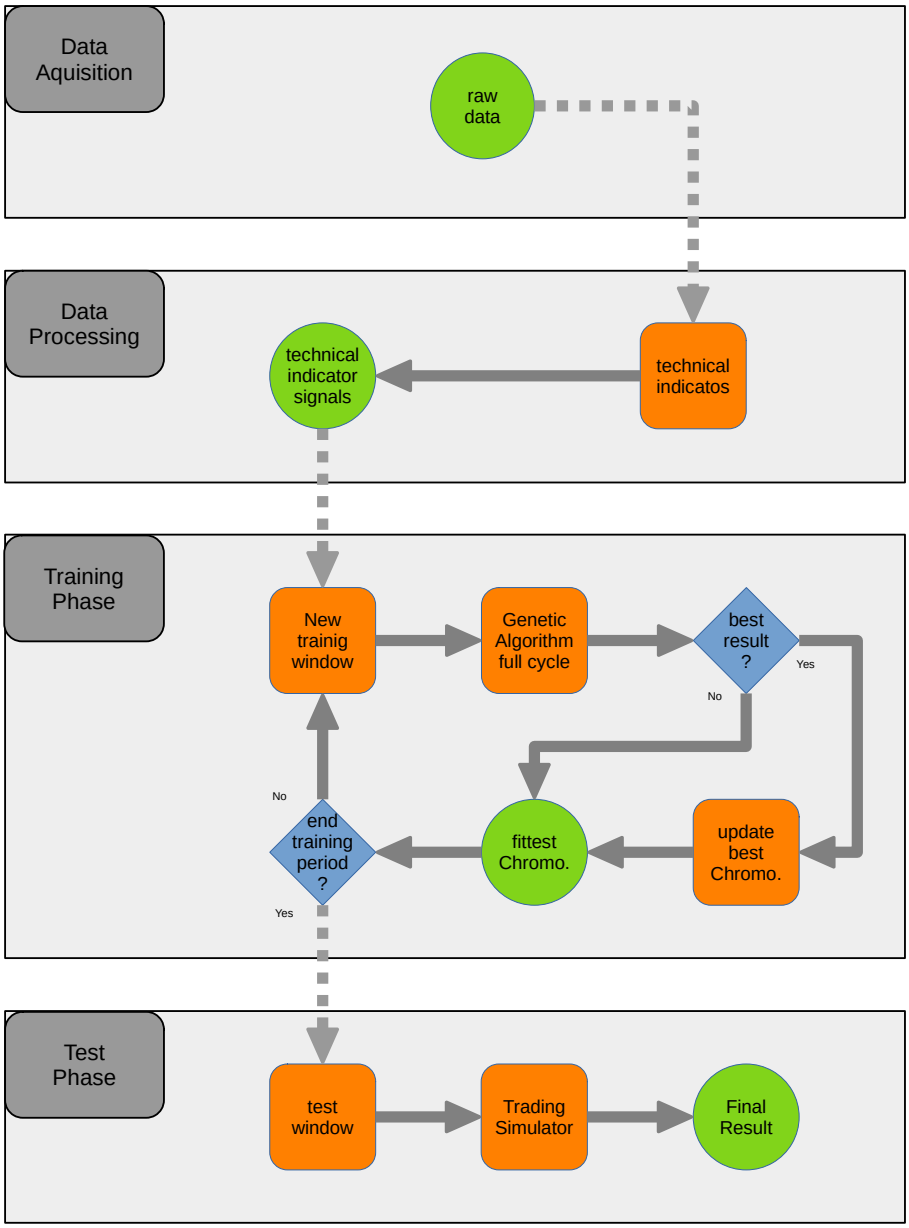

Figure 3.1: Structure of the program architecture

3.3 Algorithms

Genetic algorithms

A significant way that the proposed structure differs from others seen in chapter 2 is that two distinct genetic algorithms are used. The purpose of the first algorithm is to optimize the weights of the different technical indicators and the n-factors that all chosen technical indicators require as an input. But as the meta parameters of the algorithm had to be selected, a second genetic algorithm was implemented to handle their optimization.

The general structure of the training phase, comprised of the two genetic algorithms, can be seen in the figure 3.2. First a population of chromosomes is created in GA2, these chromosomes contain eight genes as shown in the figure 3.2, each one corresponding to the value of a variable used in GA1. After being created, the population is then evaluated, it is here that the GA1 runs. After the whole population has been evaluated, the algorithm checks if the stopping criteria is met. If yes than the algorithm has finished and solution has been found. If, on the other hand, the stopping criteria has not been met, the parents of the new generation are selected from this current generation. Using a crossover method, the new generation is created. The final step before the new generation is evaluated is to mutate some of the populations genes with a mutation method. The population is at this point again ready to be evaluated and the cycle continues until the stopping criteria has been met.

Coming back to the GA2's evaluation, each chromosome's evaluation is a full run of the GA1 with the configuration set by the values of the corresponding evaluated chromosome's genes. This algorithm replicates the behaviour of the GA2 as it has the same steps. The differences can be seen in the population generation and evaluation methods. The population generation now creates a population of chromosomes with ten genes, five corresponding to the weighs of the technical indicators signals and five that specify a variable used in the calculation of these indicators. The evaluation method of the GA1, showed in the figure 3.2, applies the five technical indicators to the implied volatility signal of each company and with a weighted average sets a forecast value F. Depending on this value, the predicted movement of the implied volatility, or *I.vol*, can be of decreasing($\downarrow$), stationary($\rightarrow$) or increasing($\uparrow$). After comparing with the real implied volatility movement, the forecast for that day and that company is labeled as correct or incorrect. The score of the evaluated chromosome is the percentage of correct forecasts, set between 0 and 1. The maximum score achieved by the GA1 full run is the score of the corresponding GA2 chromosome.

It is worth noting that by using a GA to train another GA the complexity of the overall algorithm increases exponentially, resulting in a exponentially increased running time. In order to compensate for this increased running time, the evaluation of the GA2's chromosomes and therefore the calling of the GA1 algorithm was implemented with multi threading. By running the GA2's evaluation in six threads

the algorithm was able to run six distinct GA1s at the same time reducing the overall running time of the train phase to one sixth of the time. As a reference each run of the GA2 takes around one week (168h) to complete, without this implementation this value is expected to escalate to around six weeks (1008h) demonstrating its importance.

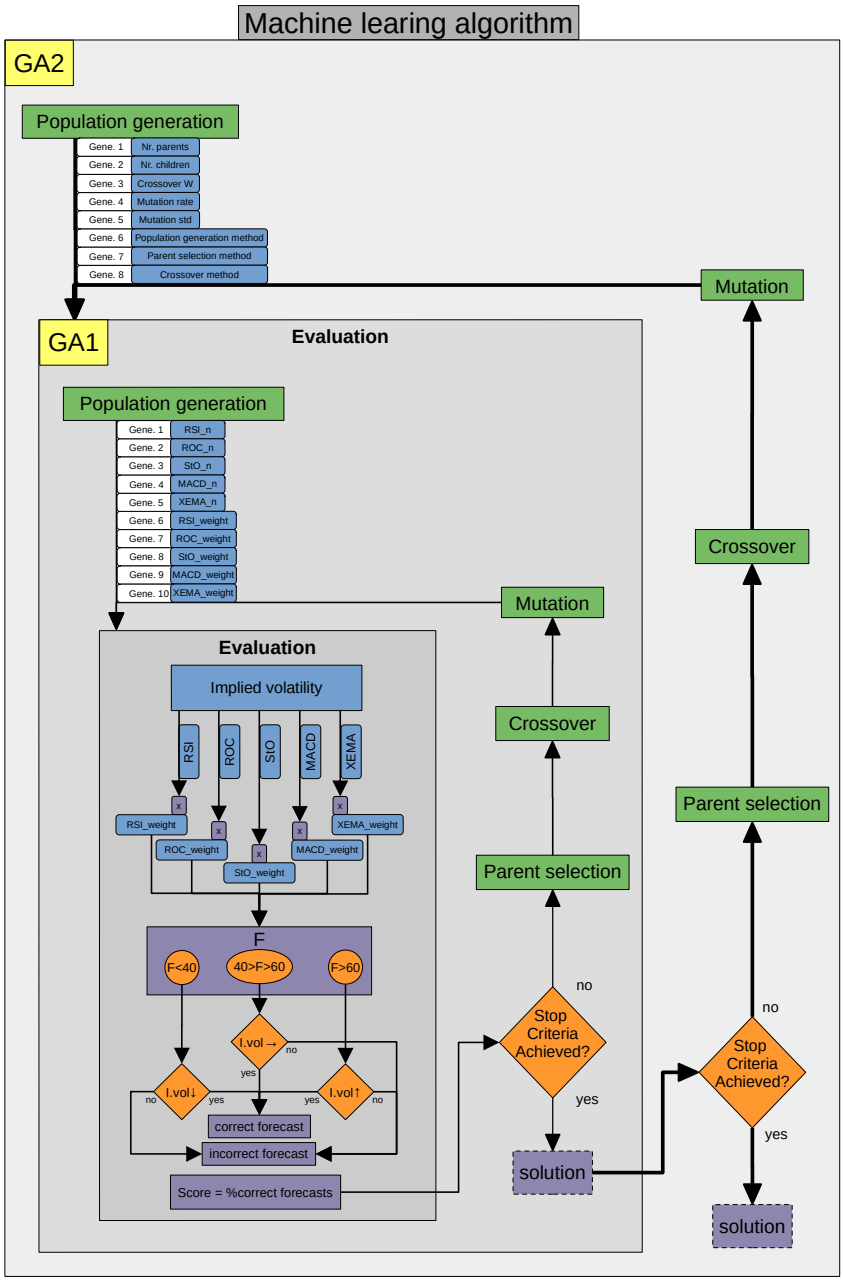

Figure 3.2: Structure of the program architecture

Trading simulator

In figure 3.2 one can see the structure of the trading simulator algorithm. This algorithm is responsible for testing the chromosome set as the best solution from the two genetic algorithms in the training phase.

For every day in the test period the algorithm firstly makes three checks: if there is any stock split, if the test period has ended and if there is any option in the portfolio with forty days to maturity. After these checks, the algorithm follows one of the three actions (open, stay, close) for every of the selected fifty companies. This companies can be seen in the table 3.1 and were selected for entering the S&P 500 top50 companies in terms of market capitalization during the time period this work focus on. Market capitalization is a company's total share value.

Ticker	Name	Ticker	Name
AAPL	Apple Inc.	XOM	Exxon Mobil Corporation
GOOGL	Alphabet Inc. Class A	WMT	Walmart Inc.
GE	General Electric Company	MSFT	Microsoft Corporation
IBM	International Business Machines Corporation	CVX	Chevron Corporation
JNJ	Johnson & Johnson	PG	The Procter & Gamble Company
PFE	Pfizer Inc.	T	AT&T Inc.
WFC	Wells Fargo & Company	JPM	JPMorgan Chase & Co.
KO	The Coca-Cola Company	PM	Philip Morris International Inc.
ORCL	Oracle Corporation	VZ	Verizon Communications Inc.
V	Visa Inc.	C	Citigroup Inc.
MRK	Merck & Co., Inc.	BAC	Bank of America Corporation
PEP	PepsiCo, Inc.	AMZN	Amazon.com, Inc.
QCOM	QUALCOMM Incorporated	CSCO	Cisco Systems, Inc.
CMCSA	Comcast Corporation	INTC	Intel Corporation
HD	The Home Depot, Inc.	DIS	The Walt Disney Company
MCD	McDonald's Corporation	UTX	Raytheon Technologies Corporation
UPS	United Parcel Service, Inc.	AMGN	Amgen Inc.
AXP	American Express Company	GILD	Gilead Sciences, Inc.
COP	ConocoPhillips	MMM	3M Company
NWSA	News Corporation	MO	Altria Group, Inc.
GS	The Goldman Sachs Group, Inc.	CVS	CVS Health Corporation
BMY	Bristol-Myers Squibb	UNP	Union Pacific Corporation
MA	Mastercard Incorporated	BA	The Boeing Company
LLY	Eli Lilly and Company	USB	U.S. Bancorp
OXY	Occidental Petroleum Corporation	FB	Facebook, Inc.

Table 3.1: Traded companies

For all these companies the action taken depends on the order from the solution chromosome and the type of position traded. If the close action is taken the algorithms closes all open positions of that company and if on the other hand the open action is taken, the algorithm makes three checks before opening a position. This behaviour is repeated for every company for every day in the test period.

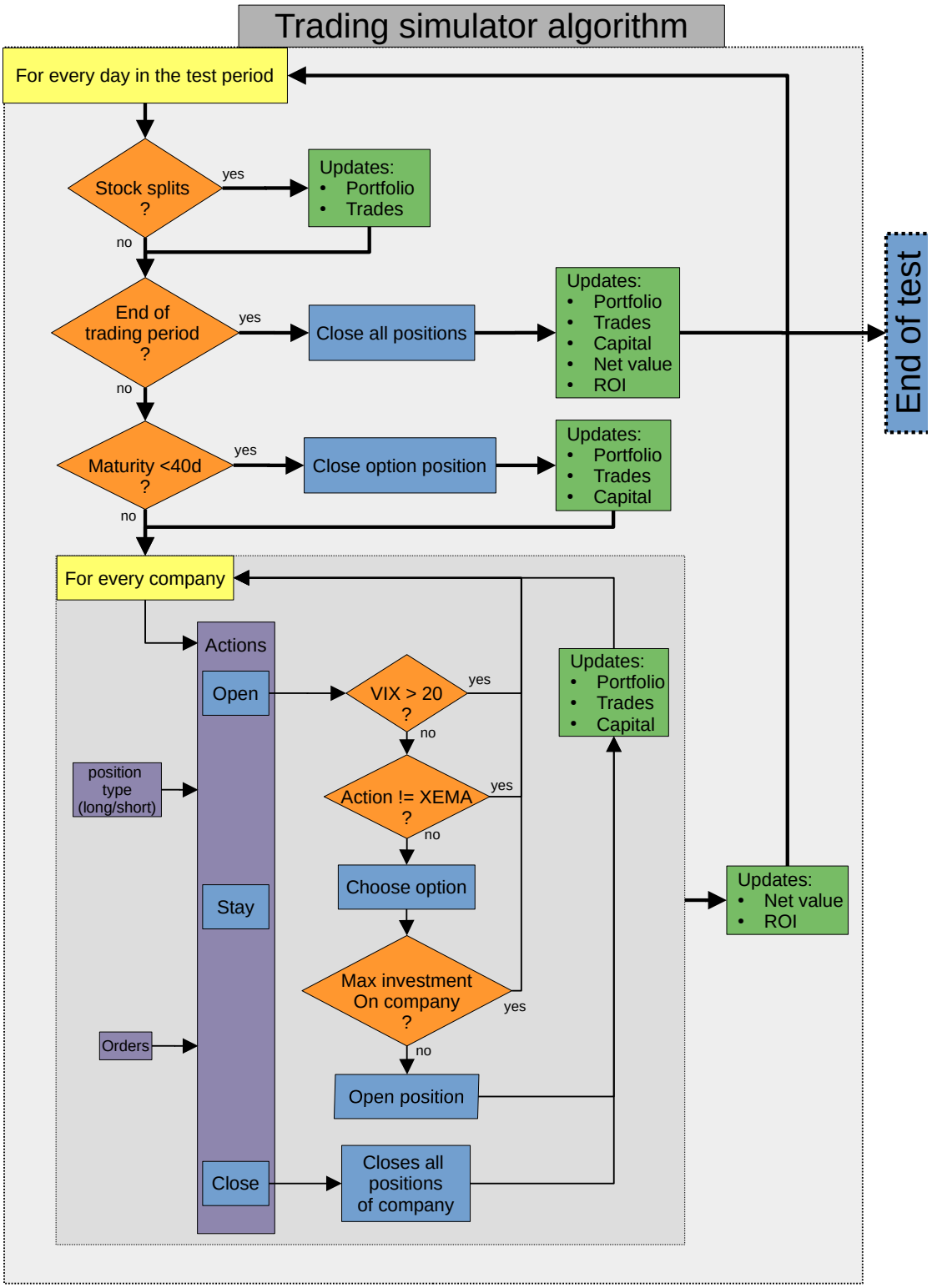

Figure 3.3: Structure of the program architecture

3.4 Data Acquisition

In any machine learning algorithm the first and most fundamental step is data acquisition.

In this work four different types of data had to be obtained: implied volatility, VIX, options information and stock splits. All data obtained correspond to the time period between January 1st of 2011 and December 31st of 2015. The top50 companies of the S&P 500, in terms of market share and for this time period, where chosen for the signals of all data types except VIX.

Implied volatility

The implied volatility signal has, as mentioned in 2.2.6, a direct correlation to option prices and so it was the subject of the machine learning algorithm forecast. This data was obtained from [19].

VIX

In order to be able to close positions and prevent new ones from being opened in periods where the market volatility was to high, and therefore option market values where too unpredictable, a threshold was implemented in the the 30 day VIX. During the market simulator, whenever the VIX value was above 20 points all positions where to be closed and new ones prevented from being opened despite the output of the machine learning model. This data was acquired from [20]

Options information

The financial objects traded in the market simulator (test phase) were options. And so, for each company of the selected 50, the close values, option symbol, and other information of all options traded during the selected time period had to be obtained from [21]

Stock splits

Finally, as option information data was not normalized for stock splits, *i.e.*, on the date of a company stock splits, the options close values changed drastically and the option symbol changed to accommodate the new strike price. This was a problem as on the date of stock splits, options in the portfolio of the simulation became nonexistent in the data for the following days. By knowing the stock split date and ratio for each company one can correct the portfolio whenever a stock split occurs. This data was obtained from [22].

3.5 Data Processing

From the raw data, feature-like signals called indicators can be computed and used as input in the machine-learning algorithm. These indicators are widely used in financial analysis and fall into two distinct groups: technical and fundamental. As fundamental indicators are usually much harder to come by, only technical indicators are used in this work. These pattern based signals can be computed from any signal with historical data.

In this work five different technical indicators were applied to the implied volatility signals of the selected companies. Each of the five selected technical indicators has a *n* variable that represents the number of days to which the formula is applied. As the choice of the *n* value affects the quality of the technical indicator signal, the specific *n* of each of the technical indicators is one of the optimized variables by GA1. From the implied volatility signal of each company fifty five different technical indicator signals were computed. These correspond to the technical indicator's formula applied with the *n* variable ranging from 5 to 60 and were used as input for the GA depending on the value of the corresponding gene.

RSI

Relative Strength Index (RSI) is a indicator that measures the magnitude of the change of a value between two dates. It measures the average increase and decrease of a signal of the previous n days and normalizes it between 0 and 100. As can be seen in 3.1, if the average loss is zero, meaning that the value of the signal increased every day in the computed time period, the RSI will be 100. Otherwise if the signal's value decreased everyday then the RSI value will be 0.

$$RSI_n = 100 - \left[\frac{100}{1 + \frac{AverageGain_n}{AverageLoss_n}} \right] \tag{3.1}$$

ROC

Rate of Change (ROC) measures the percentage in the change between the current value of a signal and its value n days before, as demonstrated in 3.2. This indicator does not have a maximum bound, as it ranges from -100 to $+\infty$. As such, in order to better accommodate its output to the GA's input, a normalization was applied between the values -50 and $+100$ which correspond to the value of the current day (t) being half and double the value of the $t - n$ day, respectively. This way, when a signal doubles its value the corresponding normalized ROC will be 100 (0 if the signal halves its value)

$$ROC_n(t) = \left[\frac{Price_t}{Price_{t-n}} - 1 \right] * 100 \tag{3.2}$$

StO

The Stochastic Oscillator (StO) compares the current day's (t) value of a signal to its highest and lowest value in the last n days. This computation follows equation 3.3 with a lower and upper limits of 0 and 1. As these limit are not ideal for the GA, a normalization was also applied but now between 0 and 1, converting this values to 0 and 100. This way if the current day has the highest value of the last n days this new normalized StO value will be 100 (0 if the current day is the lowest). It's also relevant to point that the bigger the time period n is, the smother the StO signal will be.

$$StO_n(t) = \left(\frac{Price_t - LowestPrice_n}{HighestPrice_n - LowestPrice_n} \right) \tag{3.3}$$

MACD

Moving Average Convergence Divergence (MACD) is a compound technical indicator as its value is the difference between the value of two other indicators. As shown in 3.4, this indicators are two Exponential Moving Average (EMA) signals with different time periods n and m with $n < m$.

$$MACD_{n,m}(t) = EMA_n(t) - EMA_m(t) \tag{3.4}$$

Each EMA signal is computed from the formula 3.5 where t is the current date and n is the time period for which the EMA is calculated. This type of moving average gives more relevance to newer data by applying a weight k, seen in 3.6, to the value of the current day. As the EMA formula requires a previous EMA value, for the first day of calculation a Simple Moving Average (SMA) is used instead.

$$EMA_n(t) = Price_t * k + EMA_{n-1}(1 - k) \tag{3.5}$$

$$k = \frac{2}{n - 1} \tag{3.6}$$

A SMA is perhaps the most simple technical indicator. Its value is nothing more than the mean value of the last n days of data. It's formula is observable in 3.7.

$$SMA_n(t) = \frac{\sum_{i=t-n}^{t} Price_i}{n} \tag{3.7}$$

MACD has a problem when used in a GA algorithm; its output is given in absolute values. For example, two signals, X and Y, that doubles their EMA will have different MACD values depending of the absolute value of the signal. Ex:

$$X\_EMA_{12} = 60, X\_EMA_{26} = 30, Y\_EMA_{12} = 600, Y\_EMA_{26} = 300$$

$$X\_MACD = X\_EMA_{12} - X\_EMA_{26} = 30$$

$$Y\_MACD = Y\_EMA_{12} - Y\_EMA_{26} = 300$$

As demonstrated, just by looking at a MACD value, nothing can be concluded about the momentum of the signal, Therefore an extra step is needed. A 9 day EMA of the MACD is computed and given the name of signal line (SL) and the new sl_MACD value will be proportional to the difference between the SL and MACD:

$$sl\_MACD_{n,m} = \begin{cases} 50 + \frac{MACD_{n,m} - SL}{SL} * 25, & \text{for } MACD_{n,m} > SL \\ 50 - \frac{SL - MACD_{n,m}}{MACD_{n,m}} * 25, & \text{for } MACD_{n,m} < SL \end{cases} \quad , \text{n} < \text{m}, 0 < \text{sl\_MACD} < 100 \tag{3.8}$$

A positive MACD indicates that the short EMA has a higher value than its long counterpart and so that the signal value is increasing. If, on the other hand, the MACD has a negative value, the long EMA is bigger than the short EMA and the signal is therefore falling. MACD is, for this reason, a momentum signal.

This new sl_MACD is therefore a momentum signal of the momentum signal of the original signal as is bigger than 50 if the momentum of the MACD is increasing and smaller than 50 otherwise. For example if a signal's value and its rate of growth are both increasing, then the sl_MACD will return a value above 50. If after that, the signal continues increasing but the at a lower rate of growth, than it will tend to a possible turning point. At this point, the sl_MACD will return a value below 50.

XEMA

Finally the Crossing Exponential Moving Averages (XEMA) is usually used as a visual indicator. When the short EMA has a bigger value than the long EMA then the signal is rising. Contrary to this, if the short EMA has a lower value than the long EMA, the signal is falling.

In order to use this idea as an input for the GA, the condition demonstrated in 3.11 is implemented originating a signal called XEMA.

$$XEMA_{n,m} = \begin{cases} 100, & \text{for } EMA_n > EMA_m \\ 0, & \text{for } EMA_n < EMA_m \end{cases} \quad , \text{n} < \text{m} \tag{3.9}$$

3.6 Training Phase

During the training phase the GA will use the input signals described in section 3.5 to find the combination of weights that better forecasts the movement of the implied volatility's signal. To this end a rolling window is used to select the training period.

Rolling Window

If the training data sample is too small, the machine learning algorithm's result might be a good predictor for the data in the training window but fail to forecast data in the test window. In other words, when dealing with machine learning systems, small data may lead to over-fit. For that reason data augmentation techniques play an important role in consolidate the training phase.

the augmentation technique used In this work is the rolling window. This consist in, instead of training the system with a single training window, setting a smaller window size that moves over the whole training data until all data is used. Taking as an example figure 3.4, that represents the rolling window selected for this work: Firstly the system is trained using the window *A* that encloses the first two years of the whole training data. Then, the window "rolls", in this example one year, and we get the window *B*. This process repeats until all training data has been covered by the rolling window, ending with the window *C*. This way, the training data can be used as six years worth of data with a wider diversity of signal shapes

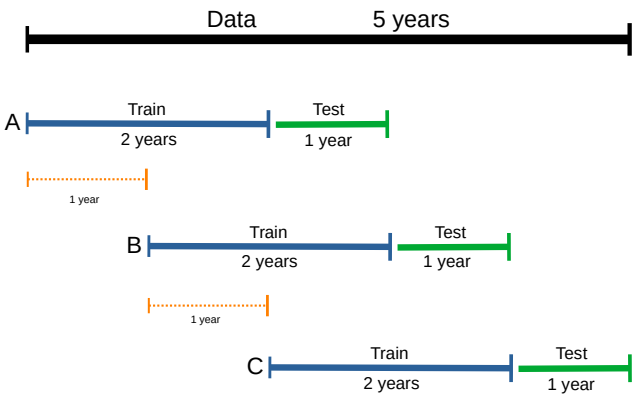

Figure 3.4: Rolling Window example with a window with size of 100 days

Second GA

The second GA is the most high level one, It serves the purpose of trying to optimize the hyper parameters of the first GA.

Population generator

In this first phase the a sequential method is used. This method assigns the same value to all genes of a chromosome, starting low in the first chromosome of the population an increasing sequentially as demonstrated in figure 3.5.

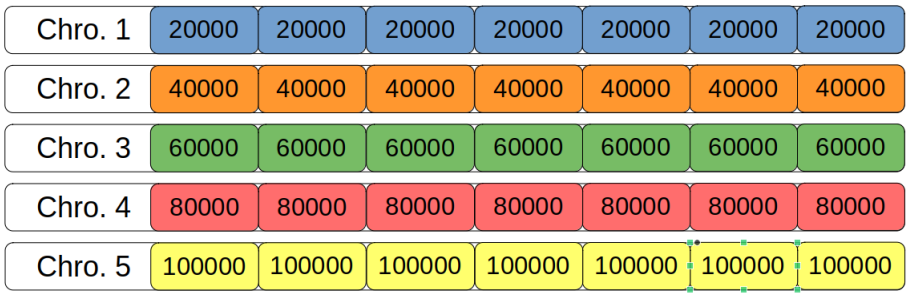

Figure 3.5: Sequential population generation method example

This Population is comprised of ten chromosomes, each one having eight genes corresponding to eight first GA's parameters: The number of parents of the parent selection phase; the number of children of the crossover phase; the *w* factor used in the intermediate method, one of the crossover methods used in the crossover phase; the mutation rate which represents each gene's probability of mutating in the mutation phase; the mutation standard deviation which dictates the degree of change of a gene's value when a mutation occurs; the parent selection method; the crossover method and the mutation generation method. This structure is shown in figure 3.6.

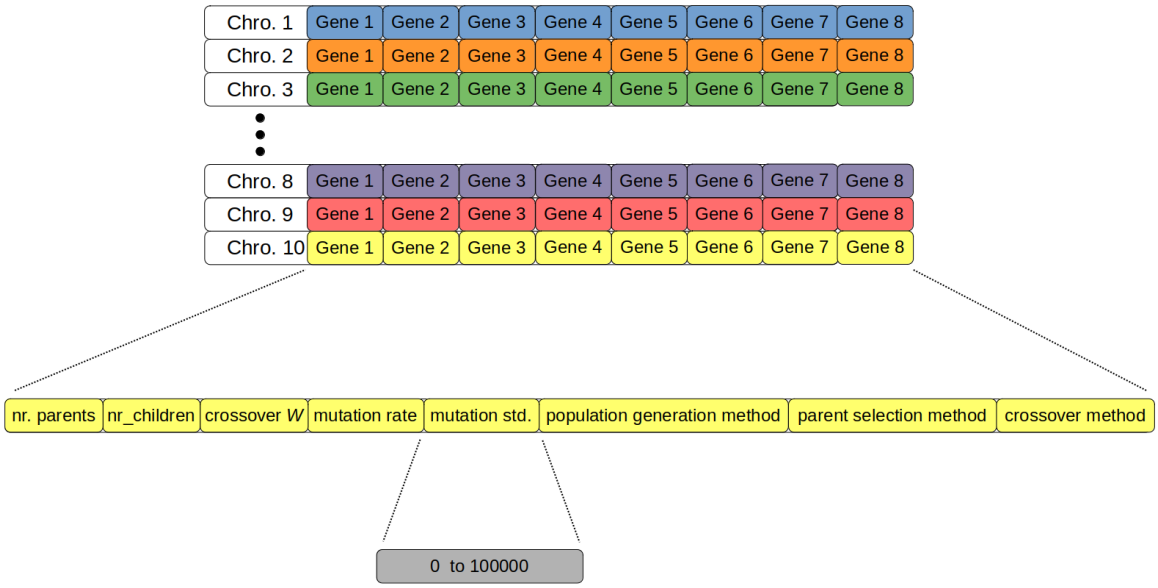

Figure 3.6: Structure of the Second GA's population configuration

All genes are single values between 0 and 100000 for ease of calculation in the crossover and mutation phases but are later translated to the corresponding value for each of their applications. For example the mutation rate is a percentage so the gene value is divided by 100000 to produce a value between 0 and 1 with five decimal numbers. The complete transformation for all genes can be seen in the figure 3.7.

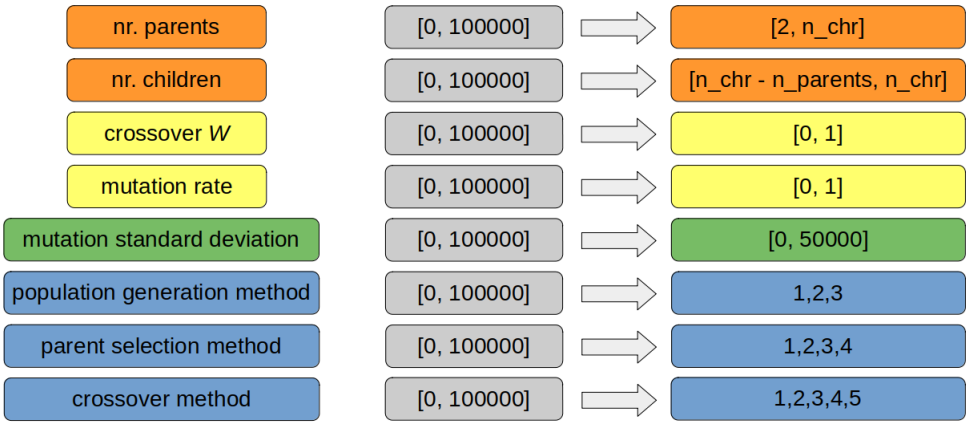

Figure 3.7: Second GA's genes transformation

Evaluation

During the evaluation phase the GA completes a full run of the first GA. This run returns, among all the necessary information from the first GA, the highest score of the run. This score ranges from 0 and 1. Each second GA's chromosome score will be the highest score from the corresponding first GA.

After evaluating all chromosomes, the hall of fame is updated, and the configuration of the five all time best chromosomes of that population is saved.

Stopping criteria

After each evaluation phase the algorithms checks if the run is complete. There are three different stopping criteria: If a chromosome has a score higher than 0.9; If the population has reached the tenth generation; If there has not been a a score increase in the last two generations.

Parent selection

So as to create a new generation the parents of the new chromosomes must be selected. Four parents are selected by the *Roulette wheel selection*, which consists in giving each individual of the population a probability of being selected. This probability is proportional to its relative score and can be computed in various ways, with the condition that the probabilities of the population must sum to 1. In figure 3.8 is illustrated an example of this method. Here the selection probability of each chromosome is given by $p_i = score_i / \sum_{k=1}^{n} score_k$, where n is the number of chromosomes in the population.

Reletive position	1º	1º	2º	3º	4º	4º
Chromossome	2	5	6	1	3	4
Evaluation score	0,6	0,6	0,5	0,4	0,3	0,3
Selection Probability	0,22	0,22	0,19	0,15	0,11	0,11

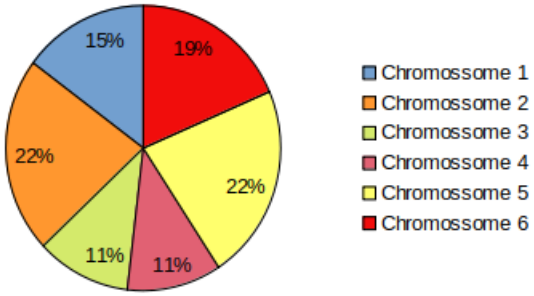

Figure 3.8: Roulette wheel selection method

Crossover

After selecting all parent chromosomes the child chromosomes are created through a crossover method. For this GA the *Random* method is applied. Each gene of the new chromosome is randomly chosen between the two corresponding genes from the two parent chromosomes.

Mutation

The final step before the new population is ready for evaluation is the mutation phase. This GA uses a Gaussian probability distribution with the mean in the gene's value and a standard deviation of 15000. Each gene has a 30% probability of occurring a mutation.

First GA

The purpose of this GA is to forecast implied volatility signals. This forecast does not need to include the value of the signal but merely the direction of the movement. The algorithm is then expected to assess if the signal's value will, in ten days time, be higher, lower, or considered static.

Population generator

As the population generation method is a parameter that depends on one of the chromosomes of the second GA it can be one of three techniques: random, where each gene is given an random value between 0 and 100000; sequential, as shown in the figure 3.5; and lastly parallel. This last method divides the search space (0 to 100000) into equal sized parcels, the same number as chromosomes in the population. Each chromosome is assign a range and each of its genes is randomly chosen from this range. An example is demonstrated in figure 3.9.

The population of this GA is comprised of a hundred chromosomes. Each chromosome has ten genes, five for the choice of the n factor present in the technical indicators values, as explained in section 3.5, and five for the weights associated to each one of the indicators. The use of these weighs is explained in the evaluation section of this GA . Similarly to the second GA, the genes of the chromosomes are single values ranging from 0 to 100000. This structure is represented in figure 3.10

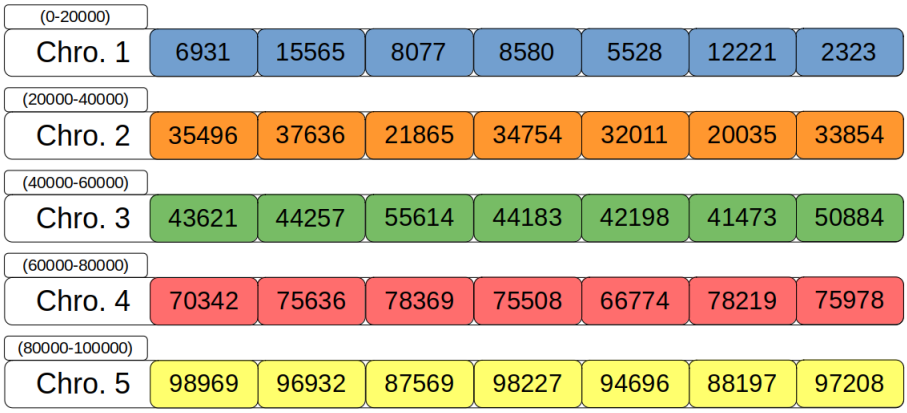

Figure 3.9: Parallel population generation method example

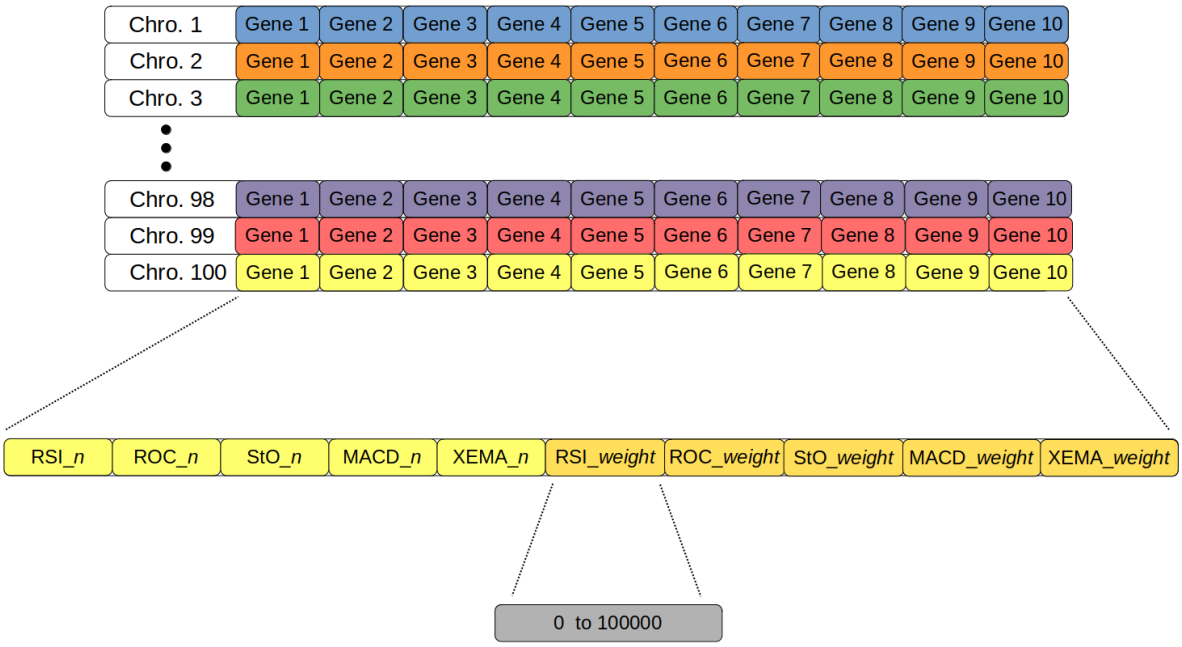

Figure 3.10: Structure of the First GA's population configuration

Evaluation

During the evaluation step of the algorithm, a predictive score *F* is computed using the weighted mean seen in equation 3.10, where n is the number of indicators, five in this work.

$$F = \frac{\sum^{n} IndicatorValue_i * IndicatorWeigh_i}{\sum^{n} IndicatorValue_i} \tag{3.10}$$

The values of the technical indicators, being original or normalized, are so that if $F = 50$ the foretasted signal is considered to be in a perfect standstill *i.e.* the value of the signal is predicted to stay the same in the forecast time of ten days. Using a threshold of 10, points three ranges were created with an associated forecast:

$$forecast = \begin{cases} up, & \text{for } F > 60 \\ stay, & \text{for } 60 > F > 40 \\ down, & \text{for } F < 40 \end{cases} \tag{3.11}$$

After acquire a prediction for every day, a ground truth is needed so to evaluate the correctness of the prediction. To this end a comparison between the implied volatility value of the ”current” day and of the one ten days later was made. If the value had increased over 3 points the real forecast was of an up day; if the value had decrease 3 points or more the real forecast was of a down day; if, on the other hand, the value had not move more than three points in either direction, the real forecast was of a stay day. The forecast set by the algorithm was then compared with this ground truth and saved as a correct or incorrect forecast.

This was reproduced for each day in the training period, for each of the selected companies. It is also worth to mention that for each company the algorithm used the technical indicator signals applied to the corresponding implied volatility.

This procedure resulted in the return of the total number of correct and incorrect foretasted days. The evaluated chromosome was then given a score corresponding to the percentage of correct ones, displayed in equation 3.12, ranging between 0 and 1.

$$score = \frac{NrCorrectDays}{NrCorrectDays + NrIncorrectDays} \tag{3.12}$$

Moreover, whenever a new generation was fully evaluated and did not have a new higher score, the mutation standard deviation would increase by 2500. As a stagnation in the population’s score could mean that the algorithm has reached a local maximum, the increase of the mutation standard deviation should allow for increasingly different solutions to be found. This technique is called hyper mutation.

Stopping criteria

The stopping criteria for this GA were the same of the second GA but differentiating in the values. The run would end if a score of 0.9 was achieved by any of the chromosomes, if the maximum score in the hall of fame had not increase for twenty generations and if the population reached the end of its hundredth generation. Reaching one of this criterion would result on the termination of the first GA's run, returning its necessary information to the second GA's evaluation of one of its chromosomes, starting a new first GA's run for the evaluation of the next second GA's chromosome.

Parent selection

After evaluation every chromosome of the population, if the stopping criteria had not been reached, new parents needed to be selected in order to create a new generation. The number of parents, unlike the second GA was not pre selected. This number could not be lower that two nor bigger than the number of chromosomes in the population meaning that in extreme conditions every chromosome could be used as a parent for the next generation. This dynamic value was linked to one of the existing genes of the second GA's chromosomes.

Also contrary to the second GA, where only one method for the parent selection was applied, in this first GA the method through which the parent chromosomes were selected varied. The value of the corresponding gene of the second GA's chromosome responsible for that particular first GA's run, dictated which method was applied. This method could be one of the following:

-*Roulette method*, already explained in the parent selection section of the second GA, where each chromosome is given a probability of being selected based on their score [11].

-*Top method*, where the chromosomes were selected by the highest score first until all parent's slots had been filled [11].

-*Tournament method*, mentioned in section 2.2.10, randomly selects a group of chromosomes. From that group the parents are the chromosomes with the highest scores [11].

-*Roulette/Top method* sees the merge of these two methods. Firstly a pre-established number of chromosomes are selected by their score -top method-. Then, the rest of parent slots are filled using the roulette method [11].

Crossover

Similarly to the previous phase. there is no pre-assign method for the first GA crossover. Instead, the method depends once again on the value of the second GA corresponding chromosome's gene.

-The *Random method* consists in randomly selecting, for each gene, the values of one of the parents corresponding gene [11].

-In order to use the *Geometric method* one has to apply the equation 3.13 where the value of a new chromosome's gene is the square root of the two parents' corresponding genes multiplication [11].

$$value_{G_3} = \sqrt{value_{G_1} * value_{G_2}} \tag{3.13}$$

-In the *intermediate method* an extra parameter is needed. It is here where the factor w, value of one the first GA chromosomes' genes, is used. following equation 3.14, the value of the new chromosome's gene is a weighted mean between the two parents' corresponding genes value. The factor w is the weight of the first parent [11].

$$value_{G_3} = w * value_{G_1} + (1 - w) * value_{G_2} \tag{3.14}$$

-The *One point method* randomly chooses the position of one of the new chromosome' genes. The genes prior to the chosen position receive the value of the corresponding genes from the first parent. The remaining genes are attributed the values of the second parent's corresponding genes [11]. Figure 3.11 shows an example of this method.

Figure 3.11: One point crossover method example

-The *Two point method* is very similar to the *One point method*, but this time the gene list is divided in three groups split by the two randomly selected points. To the genes of the first and third groups are assign the values of the first parent's corresponding genes. The second group's genes receive the values o the second parent's corresponding genes as can be seen in figure 3.12 [11].

Figure 3.12: Two point crossover method example

Mutation

Finally the mutation phase is the only one with no changes. Since both the first and second GAs' genes have the same range, the same method can be used in the two algorithms. This abstraction is the reason why the second GA's genes are kept with the standard range and associated value and only translated when really needed.

3.7 Test Phase

After completing the training phase, which implies a full run of the second GA, the fittest solution needs to be tested. Besides having two sets of genes as the solution to the training phase, one for each GA, the only important to test is the chromosome of the first Genetic Algorithm (GA) with the highest score.

Building on the conjecture presented in the introduction of this work, that the value of an asset's implied volatility has a direct correlation to the price of any of that asset's options, the test phase evaluates the feasibility of the proposed solution to predict the implied volatility and thus the aptitude to buy and sell options for profit.

Case studies

The test phase of this work is divided into four case studies. In each case study the simulator trades different types of options, either call or put, to better analyse which yields better results. Another aspect that changes between case studies is the type of positions. These can be long, where the option is bought from the marker and later sold, or short, where sold options are later bought back. An extended explanation of this characteristics can be seen in sections 2.2.1 and 2.2.2.

The four case studies consist then on:

1. Making **long** positions of **call** options.
2. Making **long** positions of **put** options.
3. Making **short** positions of **call** options.
4. Making **short** positions of **put** options.

All transacted options are in-the-money and in between 90 days to 40 days until maturity. As options approximate maturity their prices get more susceptible to variations of corresponding stock. As maturity draws closer, price percentage changes become steeper. Closing positions forty days to maturity decreases some of the risk from the trade.

Trading Simulator

The trading simulator keeps record of a series of structures for later analysis: a dictionary with every trade, containing both the open and close price, number of options of the trade and the option's root; a portfolio with all open positions not yet closed and their corresponding options; a record of the capital throughout the test time period; and a record of the Return on Investment (ROI) of the trades.

The first step of the simulator is to check if any company had a stock split in that day. If there is an occurrence and there are options of that company in the portfolio, the options' root and quantity have to

be corrected -as explained in the stock split section in 2.2.1-. The open positions must also be corrected: both option price, quantity and root.

The next step is to check for the end of the test period. If indeed is the last day all positions are closed: for long positions all options are sold and for short positions they are bought back. The last check before addressing the orders from the GA's chromosome is to check if any options in the portfolio has reached the forty day to maturity boundary. If that happens the position associated with those options is closed (the options are either sold or bought back in case of a long or short position respectively).

The simulator can now analyse the orders created by the solution chromosome of the trading phase. This consists in a signal for each company that can take three values depending on the forecast made by the solution chromosome:

$$orders = \begin{cases} 1, & \text{implied volatility increase} \\ 0, & \text{implied volatility stationary} \\ -1, & \text{implied volatility decrease} \end{cases} \tag{3.15}$$

Depending on the type of position and thus on the case study, the same order can lead to different actions. The table 3.2 demonstrates this relationship.

position \ order	1	0	-1
long	open	-	close
short	close	-	open

Table 3.2: Actions depending on the type of positions and order value

This disparity can be explained by the following example: If an order has a value of 1 the the prediction is for the implied volatility to increase which, by the assumption of this thesis, will lead to and increase of the option's price. Now there are two options, if the position is long, then the action should be to open a position and buying options. If, on the other hand the case study uses short position, then the action, if there is any open position, should be to close and buy the options back. The opposite happens if the order has a value of -1.

The simulator now makes three verification before opening a position: The first one is the check if the VIX value is below twenty points since a high VIX value may be consider the result of an unstable market.

The Second verification uses the value of the option's specific XEMA. This signal is computed for every option of every traded company. The behaviour of this signal is as explained in section 3.5 but similarly to the order signal, it takes the values of 1, 0 and -1. In order to control any possible false forecasts by the GA, the two signals are compared and if their values do not coincide the simulator does not go through with the order.

The third verification checks if the maximum investment per company has been reached. In order

to decrease risk in investments is important that the capital is distributed in a diverse portfolio. For this reason each position has a maximum investment of 0.5% of the initial capital with each company having a maximum of investment of 5% of the initial capital.

Once the simulator has all the ”approvals” it buys, sells or does nothings according to the order. If the order is to open a position, by either buying or selling transactions, a single option is chosen, the first in-the-money options that satisfies the case study requirements. The number of options bought or sold is determine by a maximum capital per transaction 0.5% of the initial capital, in this work this was five thousand dollars (5000$). If on the other hand the order is to close then all open positions of that company are closed, depending on the case study the options are sold or bought back.

After each trading the capital, net value, ROI are updated as is the portfolio and trade dictionary.

There are two important signals that will be return once the simulation has finished: The profit and the net value. The net value consists on the sum of the capital and the market value of every option in the portfolio at that particular moment, in case of long positions or the capital value minus the market value of every option in the portfolio at that particular moment, in case of short positions. This way the true evolution of the simulator's portfolio value can better be perceived.

Finally the third signal by which the solution chromosome will be evaluated is ROI following the formula 3.16. This signal is widely used in financial analysis to quantify the success of a trading strategy.

$$ROI = \frac{\text{return of investment}}{\text{cost of investment}} \tag{3.16}$$

4

Results and Discussion

Contents

4.1 Overview

In this chapter the result of this work will be introduced and discussed. Starting with the training phase, for each of the three training periods, the evolution of the genetic algorithms score will presented as well as the composition of the best scoring chromosome. In the test phase result section the trading simulator results for the four case studies will be presented and analysed.

4.2 Train phase results

Even though the train phase is divided in three time periods corresponding to the trading days of 2011, 2012 and 2013, its result section will be divided by a different structure. Firstly the composition of the GA2's solution chromosome for the three time periods will be presented and analysed, followed by the composition of the GA1's solution chromosome of the three different periods, and ending with the score evolution graphics. This structure is used to enable an easier comparison between the results of the time periods.

GA2's chromosome composition

The composition of the GA2's solution chromosome of the first, second and third period can be seen in tables 4.1, 4.2 and 4.3 respectively. For ease of comprehension these are not the actual genes' values but the variables values that they represent. The original genes' values are integers between 0 and 100000.

As GA2's evaluation is not tied to a financial instrument as implied volatility, the expectation was that the composition of the three solution chromosomes would be similar to one another. This expectation is met, as can be seen by comparing the three compositions. Starting by the population generation method, parent selection method, and crossover method, its noticeable that, even though the actual genes' values differ, the same methods are selected in the three time periods. As the preferable crossover method was not the intermediate method, the crov. w variable has no significant meaning for the solution since this was a parameter only used in this specific crossover method. The rest of the variables take standard values, for example the mutation rate varies from 0.1 to 0.2 which is predictable seen as a high mutation rate can prevent the algorithm convergence towards a maximum. Another example is the number of parents being always less than half of the population as a bigger percentage of parents would propagate bad solutions into the next generations.

First period's GA2 chromosome			
number of parents 39	number of children 68	crov. w 0.2	mutation rate 0.141
population gen. method random	parent selection method top	crossover method 1 point crossover	mutation std. 10000

Table 4.1: GA2's best scoring chromosome composition in the first training period

Second period's GA2 chromosome			
number of parents 21	number of children 80	crov. w 0.37657	mutation rate 0.2
population gen. method random	parent selection method top	crossover method 1 point crossover	mutation std. 10000

Table 4.2: GA2's best scoring chromosome composition in the second training period

Third period's GA2 chromosome			
number of parents 11	number of children 90	crov. w 0.1	mutation rate 0.1
population gen. method random	parent selection method top	crossover method 1 point crossover	mutation std. 5000

Table 4.3: GA1's best scoring chromosome composition in the third training period

GA1's chromosome composition

In tables 4.4, 4.5 and 4.6 the compositions of the GA1's solution chromosomes of the three time periods can be seen respectively. Similar to the previous section some of the genes displayed in the following tables are not associated with their actual value but with the variable they represent. This is happens in the case of the last four genes. Although the genes' values range from 0 to 100000 the values the displayed variable is the corresponding *n* factor of the technical indicators.

Contrary to the GA2, the GA1's evaluation directly interacts with a financial instrument, implied volatility. As the financial market is a very complex system and contrary to physical systems, is influenced by human sentiment and actions, the shape of a company's implied volatility signal can be completely different in two distinct time periods. For this reason the expectation was that depending on the time period of the train phase, the solution chromosome composition would have a significant variation. As can be seen in the three following tables, this does not occur. The case might probably be that, being the train periods three consecutive years, the time between them is not enough so that the financial market suffers a fundamental change that would require a completely different configuration of technical indicators. In terms of chromosome composition, from the three solution chromosomes can be seen that that RSI and MACD are the most impactful technical indicators in the forecast computation, followed by XEMA.

GA1 chromosome				
RSI weight 88543	ROC weight 266	StO weight 29	MACD weight 63814	XEMA weight 926
RSI n 60	ROC n 46	StO n 56	MACD n 51	XEMA n1,n2 2-20

Table 4.4: GA1's best scoring chromosome composition in the first training period

GA1 chromosome				
RSI weight 62624	ROC weight 1986	StO weight 55	MACD weight 86145	XEMA weight 1235
RSI n 60	ROC n 5	StO n 7	MACD n 51	XEMA n1,n2 2-20

Table 4.5: GA1's best scoring chromosome composition in the second training period

GA1 chromosome				
RSI weight 99834	ROC weight 56	StO weight 23	MACD weight 17188	XEMA weight 1784
RSI n 60	ROC n 53	StO n 41	MACD n 5	XEMA n1,n2 2-20

Table 4.6: GA1's best scoring chromosome composition in the third training period

Score evolution

The score evolution of the three different train periods are presented in figures 4.1, 4.2 and 4.3 with the maximum final scores of 0.5120, 0.5910 and 0.6104 respectively. These values means that 51% to 61% of the trading days the machine learning algorithm can correctly predict the direction of the implied volatility signal in a period of ten days. This result meets the expectations as implied volatility is a very complex signal and its forecast is a highly discussed, ever changing problem, as explained in the state of the art.

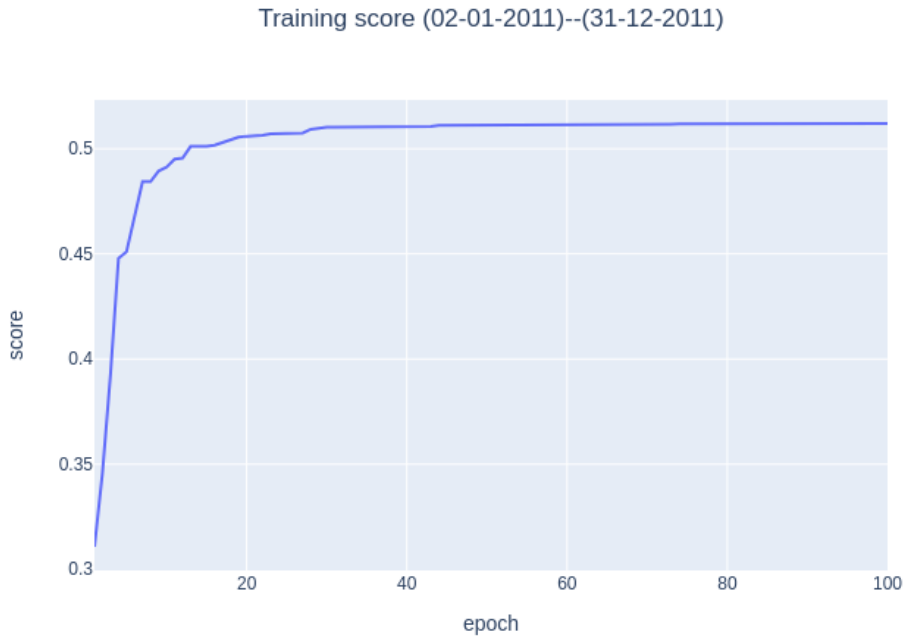

Figure 4.1: Score evolution during the first train period

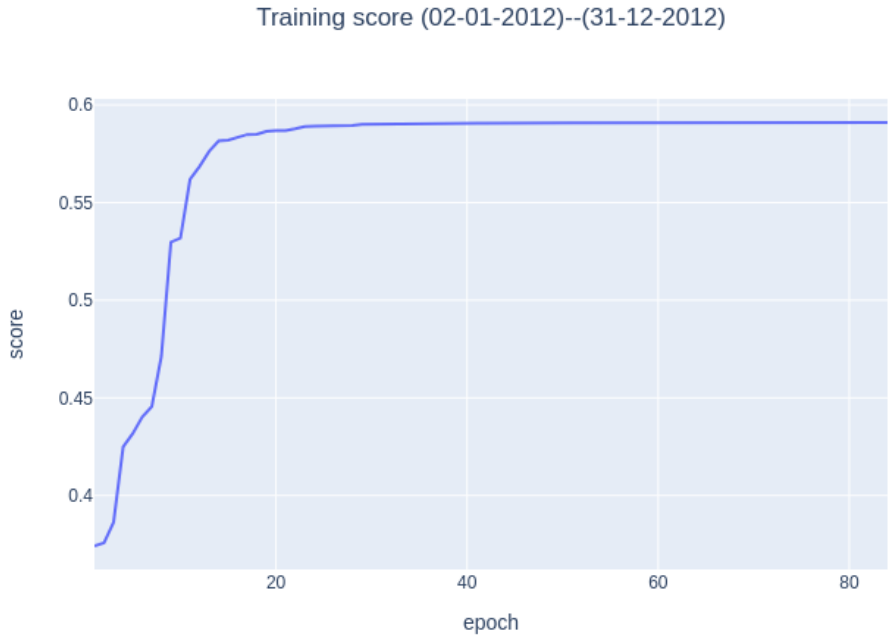

Figure 4.2: Score evolution during the second train period

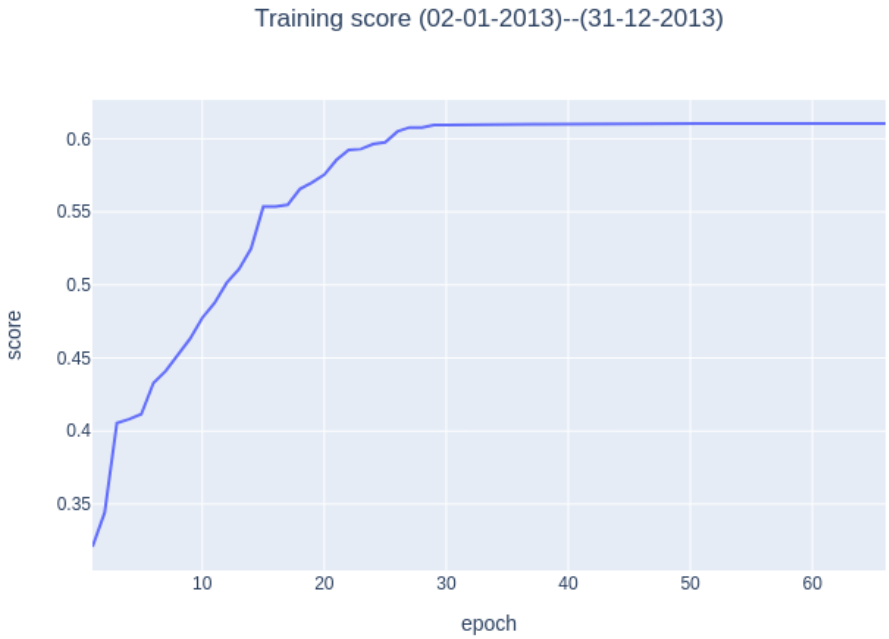

Figure 4.3: Score evolution during the third train period

4.3 Test phase results

Table 4.7 presents the trade statistics for the four case studies in the three different test periods. It can be seen that the two case studies with the higher percentage of positive trades are the long calls and short puts with this value ranging from 60% to 65%. This comes as no surprise as the financial market and in particular the S&P 500, despite short term fluctuations, tend to have a positive growth in the long term. These upward tendency signifies that put options usually lose value as the companies increase theirs. This, combined with the fact that options loose value as they reach maturity makes put options the best choice to open short positions, as can be seen in figure 4.4

Case study	Time period	total trades	positive trades	negative trades	% positive trades
Long Calls	1st period	25	16	9	64%
	2nd period	31	17	14	64,84%
	3rd period	26	16	10	61,54%
Long Puts	1st period	441	147	294	33,33%
	2nd period	372	121	251	32,53%
	3rd period	437	132	305	30,21%
Short Calls	1st period	88	42	46	47,73%
	2nd period	79	42	37	53,16%
	3rd period	26	22	4	84,62%
Short Puts	1st period	1270	805	465	63,39%
	2nd period	914	586	328	64,11%
	3rd period	325	203	122	62,46%

Table 4.7: Trades comparison for the different case studies and test periods

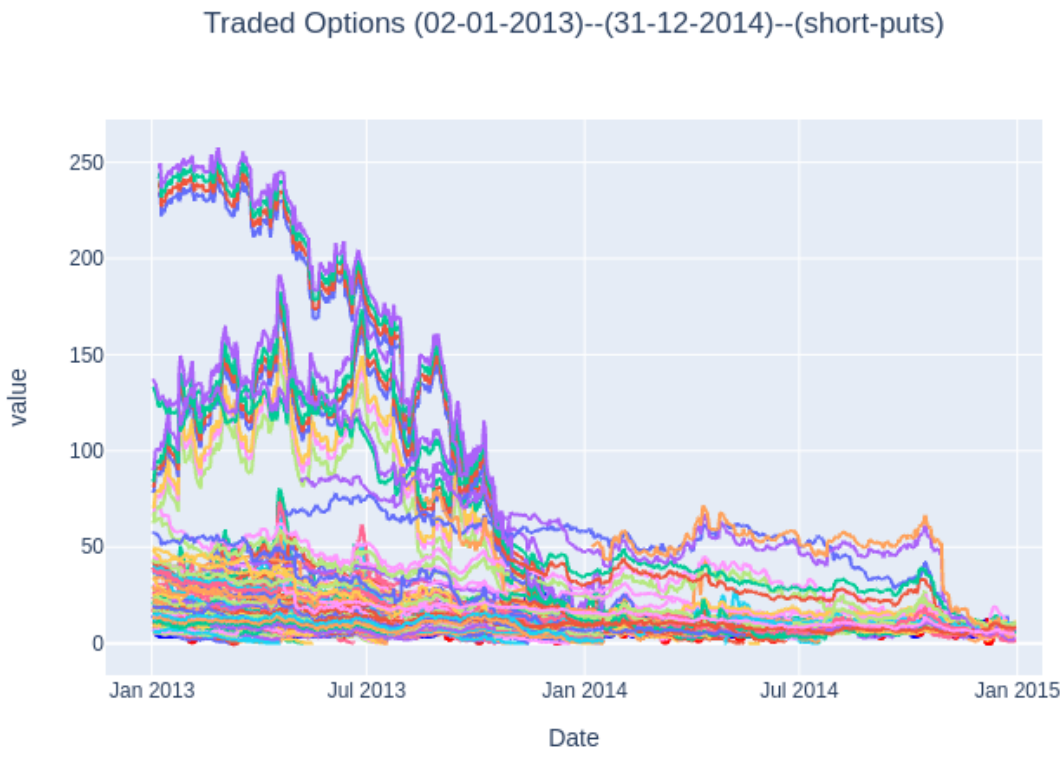

Figure 4.4: Traded options' value during the second test period for the short puts case study

On the other hand, and despite the continuously decrease of options value, call options increase their price as the corresponding stock value increase. This opposition makes for the type of signals seen in figure 4.5. This tendency to have more upward movements that put options value, makes call options better choices for long positions than put options. The other two case studies, long puts and short calls, are somehow contradictory in its nature. As already explained in the current financial market puts tend to loose value as calls tend to increases theirs. By this reasoning, opening long positions (where the value is expected to increase) with a put option (where the value tends to decrease by the behaviour of the market) has a higher risk as implied volatility is not the only conditioning in option pricing and even with a near perfect implied volatility forecast this two case studies would be less reliable than short puts and long calls.

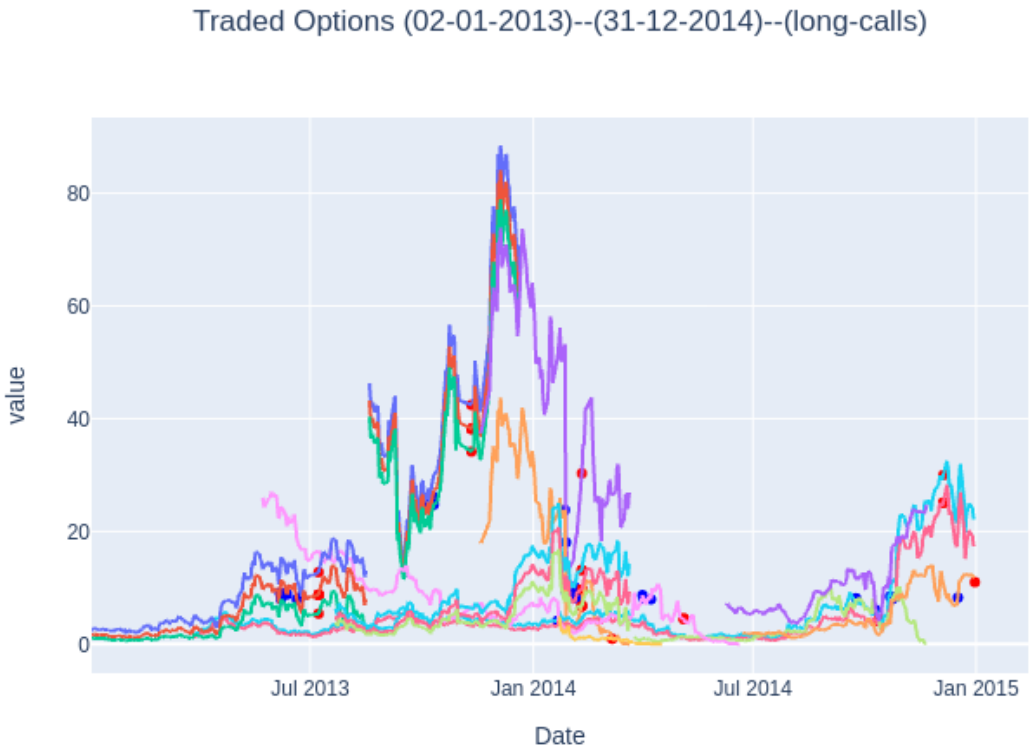

Figure 4.5: Traded options' value during the second test period for the long calls case study

The difference between the number of trades of short puts and long calls seen in table 4.7 can be explained by the fact that, as options value intrinsically decreases with time, there are more situations in which a short position is advantageous than with long positions, that are perpetually fighting against the ”natural” movement of options value. Besides this occurrence, throughout the test periods long calls have showed to yield a bigger profit for trade, and thus a bigger ROI, that short puts, as can be seen in table 4.8. Besides this occurrence by the end of each of the test periods the short puts case study yielded a higher absolute profit that long calls. This happens because of the increased number of trades this case study makes. As already could be predicted by the previous table, the long puts and short calls case studies have negatives profits, with the long puts being the case study with the worst result as it has both the lowest ROI and bigger number of trades of the two.

Case study	Time period	ROI	Profit	Profit/trade	Avg. Profit
Long Calls	1st period	21,72%	27.081k$	1083,24$	41.622k$
	2nd period	23,57%	36.49k$	1177,01$	
	3rd period	47,21%	61.295k$	2357,5$	
Long Puts	1st period	-7,43%	-163.678k$	-371,15$	-221.591k$
	2nd period	-10,93%	-203.002k$	-545,70$	
	3rd period	-12,28%	-268.092k$	-613,49$	
Short Calls	1st period	-1,92%	-8.585k$	-97,56$	-3.031k$
	2nd period	-3,77%	-15.507k$	-196,29$	
	3rd period	34,35%	33.185k$	1276,35$	
Short Puts	1st period	06,53%	415.921k$	327,50$	289.561k$
	2nd period	08,65%	363.648k$	397,86$	
	3rd period	05,81%	89.113k$	274,19$	

Table 4.8: ROI and profit analytics for the different case studies and test periods

4.3.1 ROI

The ROI evolution of the four case studies for the three test periods can be found in figures 4.6, 4.7 and 4.8. Different from table 4.8, now, not only the final ROI value can be seen, but the whole evolution throughout the test periods. From these figures it can be seen that some periods are better than others depending on the case study. For example in 4.6, short puts ended with a negative ROI value but by the end of the test period it was already positive. Even in 4.7, where the short put ROI reaches 24% and has a decline in the second half of 2013, the final ROI value is positive. This shows that the duration of the test period was not too short as the algorithm has time to compensate for eventual bad periods. In the case of the two worst case studies, short calls and long puts, the opposite occurs, even though there are some periods with a positive ROI value, as the majority of trades have a negative performance, the ROI value tends to negative. This can be seen in figure 4.8 where even thought there is a lucrative period by the end of 2014, the overall performance is negative. The test period is also not too long as there is no correlation between the duration of the test and the ROI value. Is expected that if a trained algorithm was applied to a longer test period the returns would decrease, as the time distance between test and train period would result in too different market behaviours.

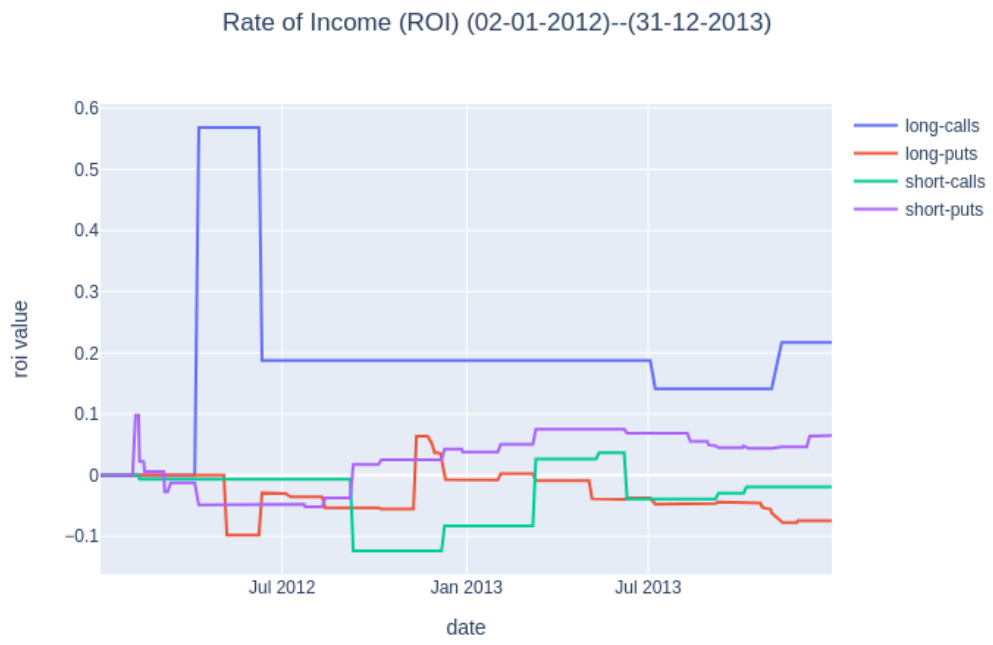

Figure 4.6: ROI evolution for the four case studies during the first test period

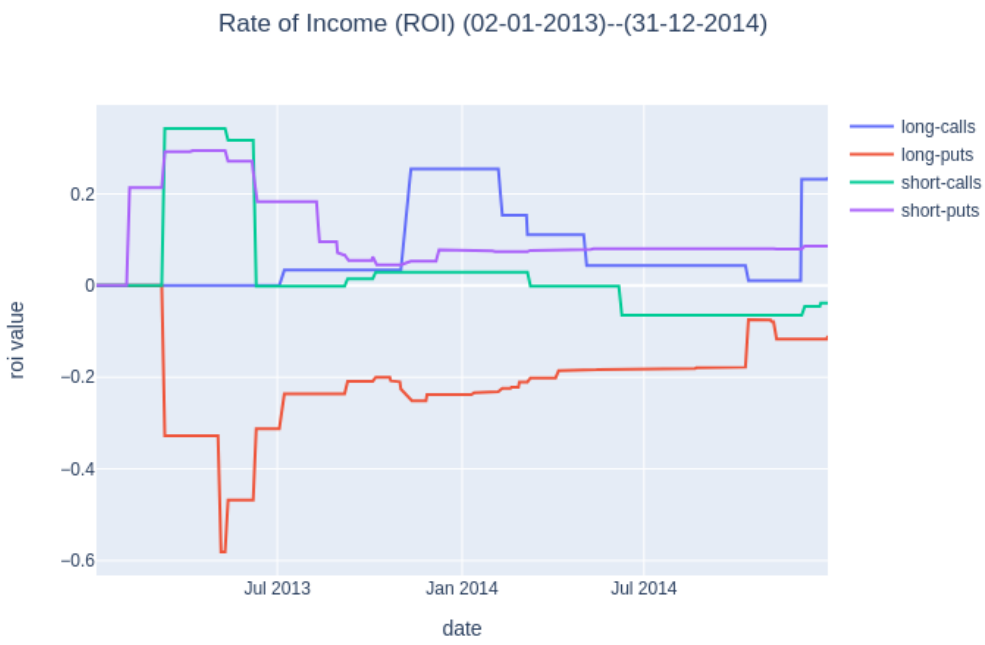

Figure 4.7: ROI evolution for the four case studies during the second test period

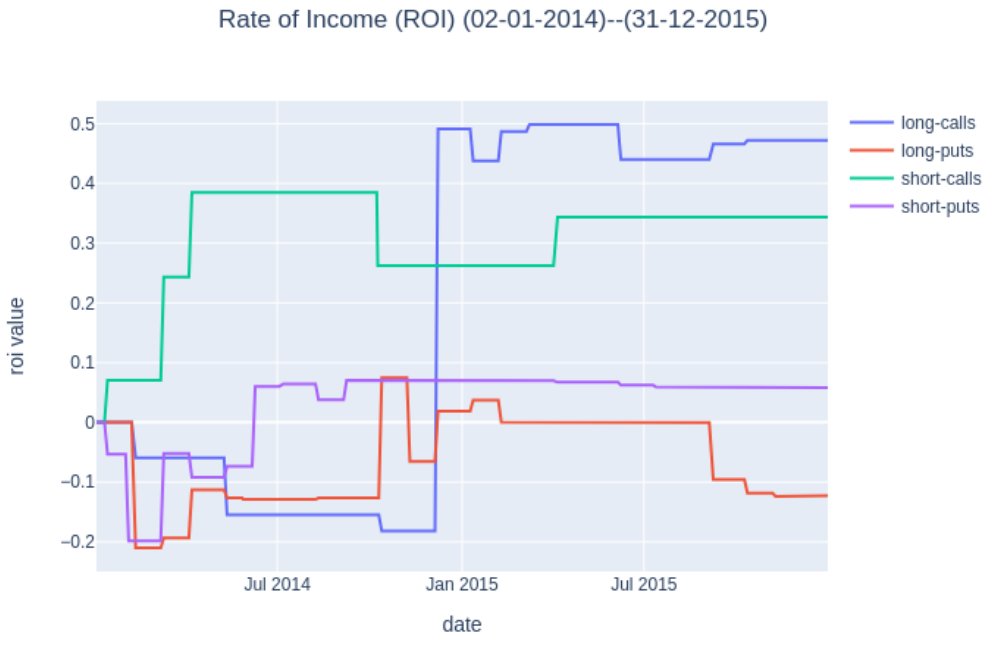

Figure 4.8: ROI evolution for the four case studies during the third test period

4.3.2 Net Value

Figures 4.10, 4.11 and 4.12 represent the evolution of the net value of the portfolio throughout the respective test period. This value calculation depended on the case study. In a case study with long positions, is the sum of the capital and the market value of all options in the portfolio. In a case study with short positions, is the capital value minus the market value of all options in the portfolio. This is a better parameter to evaluate the results of the case studies than pure capital as doesn't treat investments as losses of money. For example, in figure 4.9 the traded instruments are long calls. If the capital was the analysed parameter, by the end of 2014 one could read that the algorithm had placed bad position and later recuperated, but by looking at the net value it can be seen that the capital was used to open a position where the option value increased.

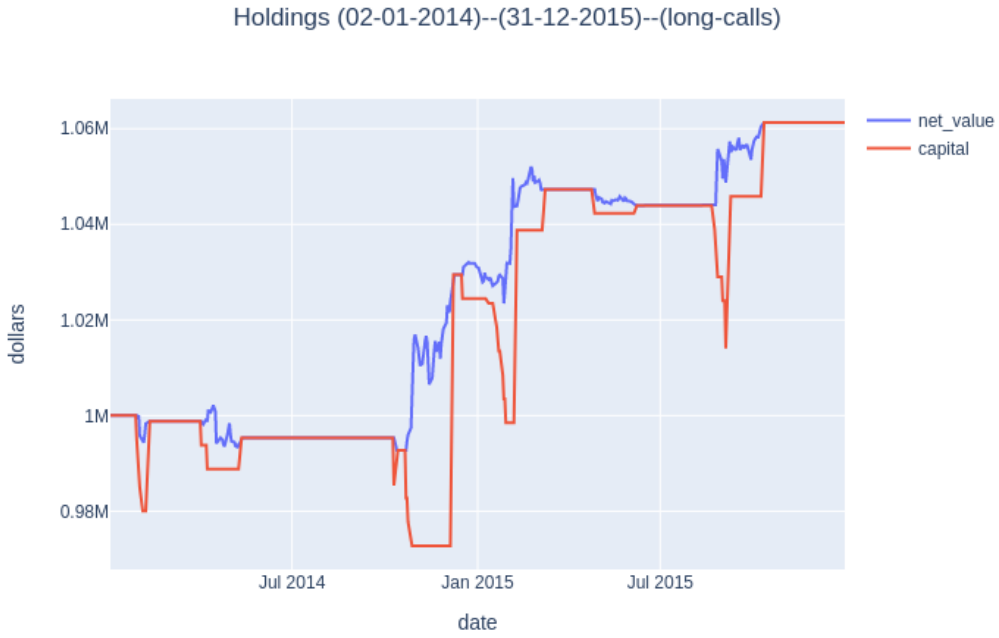

Figure 4.9: Holdings of the long calls case study in the third test period

After looking at the ROI graphs, the following figures show that besides having a lower ROI value than long calls, the increased number of trades makes trading short puts the best case study in terms of absolute profit.

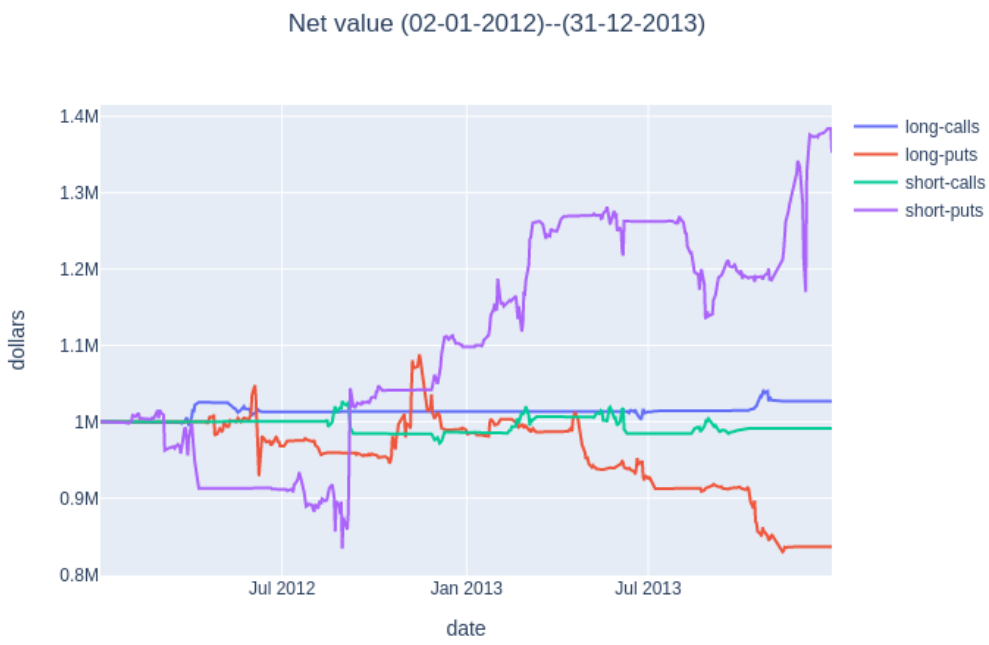

Figure 4.10: Net value evolution for the four case studies during the first test period

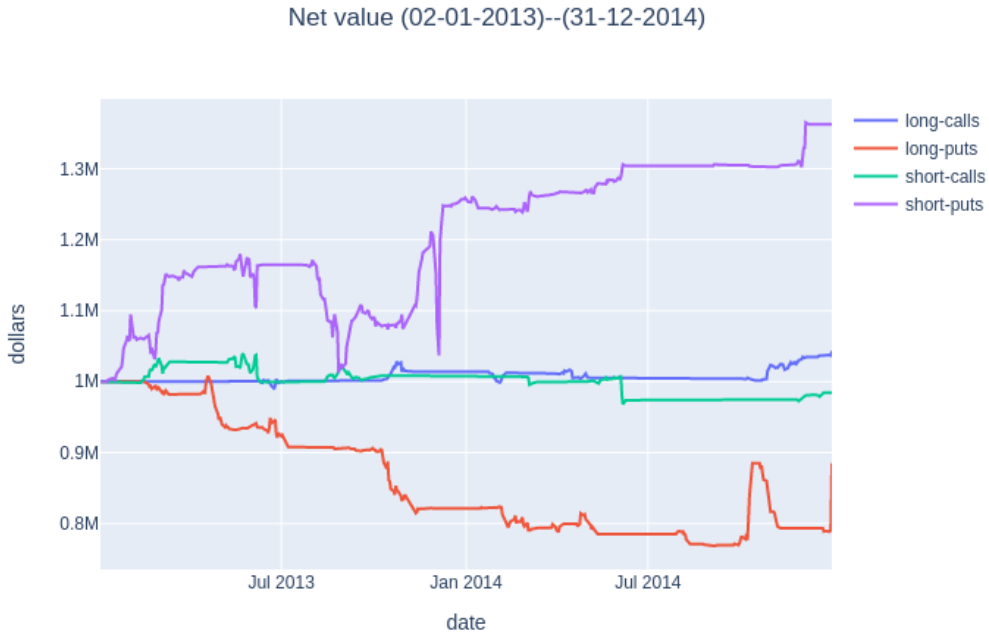

Figure 4.11: Net value evolution for the four case studies during the second test period

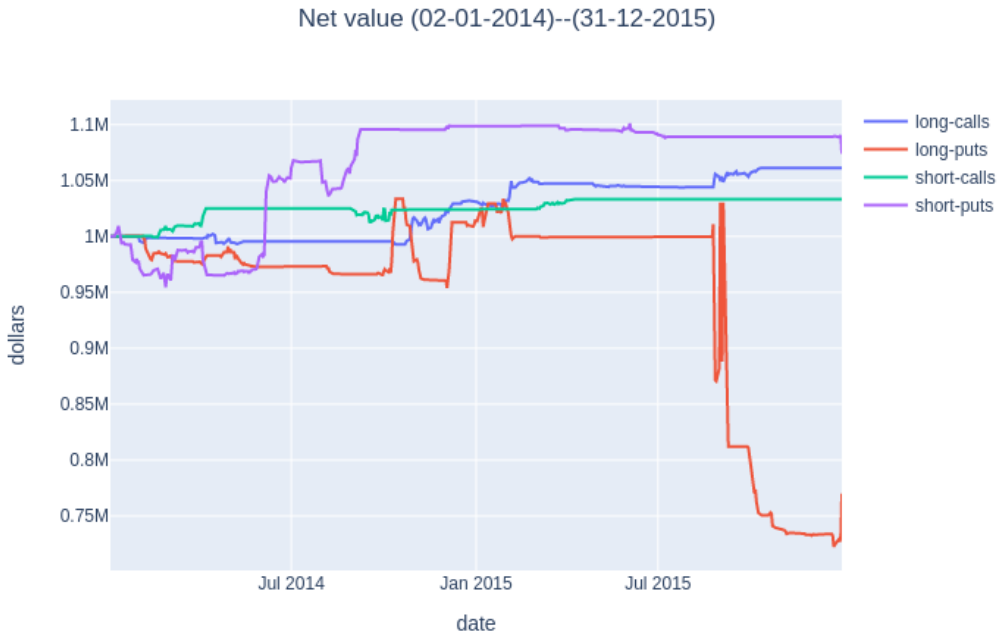

Figure 4.12: Net value evolution for the four case studies during the third test period

4.3.3 Profit

The profit evolution for the four case studies in the three test periods can be seen in figures 4.13, 4.14 and 4.15. Even though the long calls case study has the biggest ROI value of the four, the short puts case study managed to open more positions and thus be the most profitable case study. In the first test period it ended with a profit of 415.921k$ which corresponds to a total growth of 41,59% of the initial investment, or 20,795% per year. In the second test period the total growth was of 363.648k$ and 36,36% and 18,18% of percentile growth in two and one year respectively. In the final test period these values where 89.113k$ total, 8,91% in two years and 4,455% yearly. It is noticeable that having a maximum investment per company as a percentage of the initial investment is very important. In two of the three time periods the short puts case study started with a negative profit but as the losses where controlled, the algorithm managed to revert the situation and make a good profit. If the investments where not controlled the portfolio might had run out of capital in the start of the test period and would not be able to compensate.

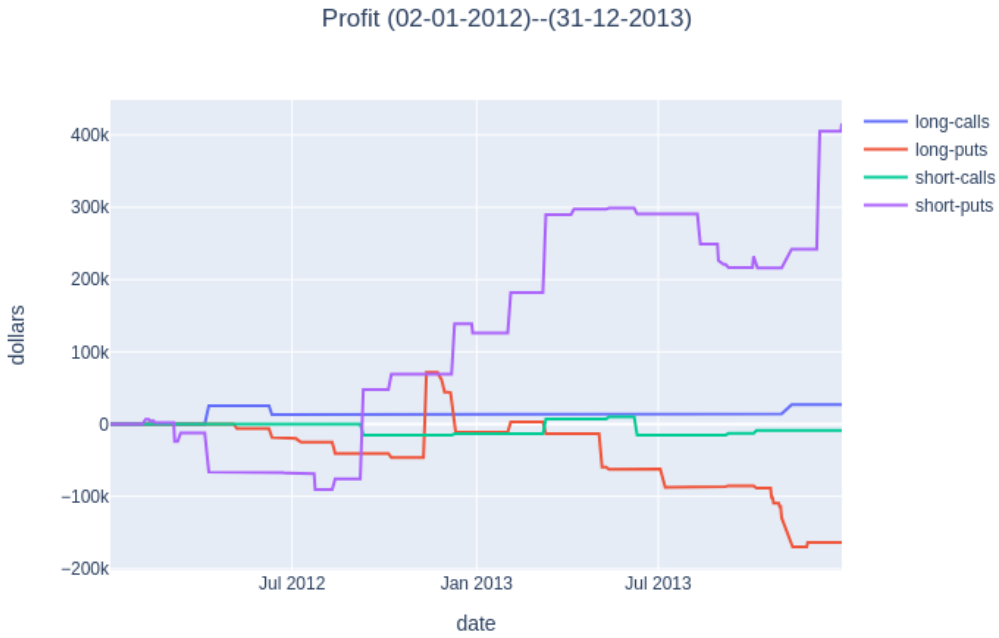

Figure 4.13: Profit evolution for the four case studies during the first test period

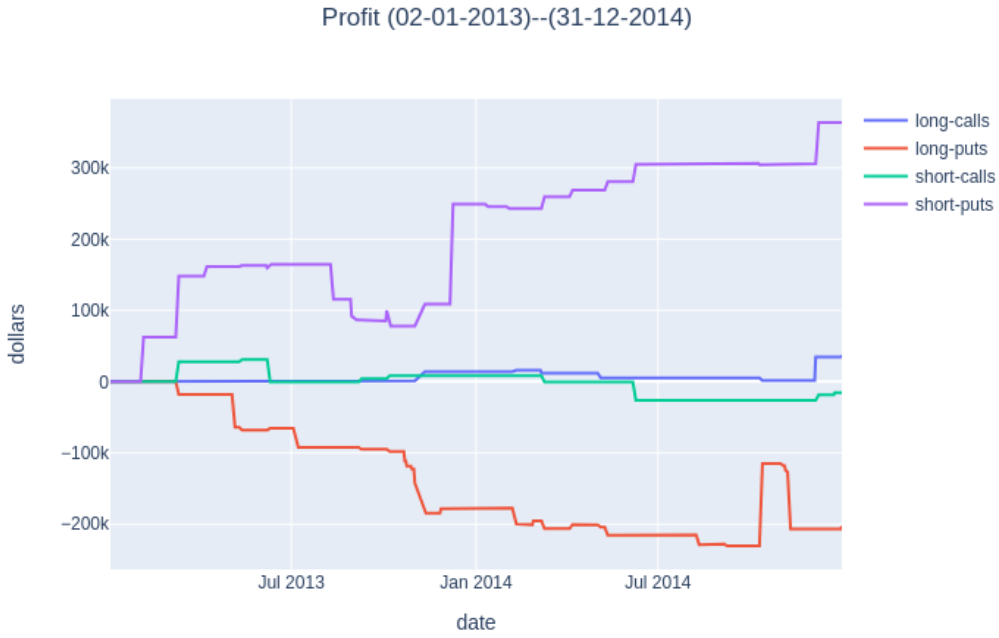

Figure 4.14: Profit evolution for the four case studies during the second test period

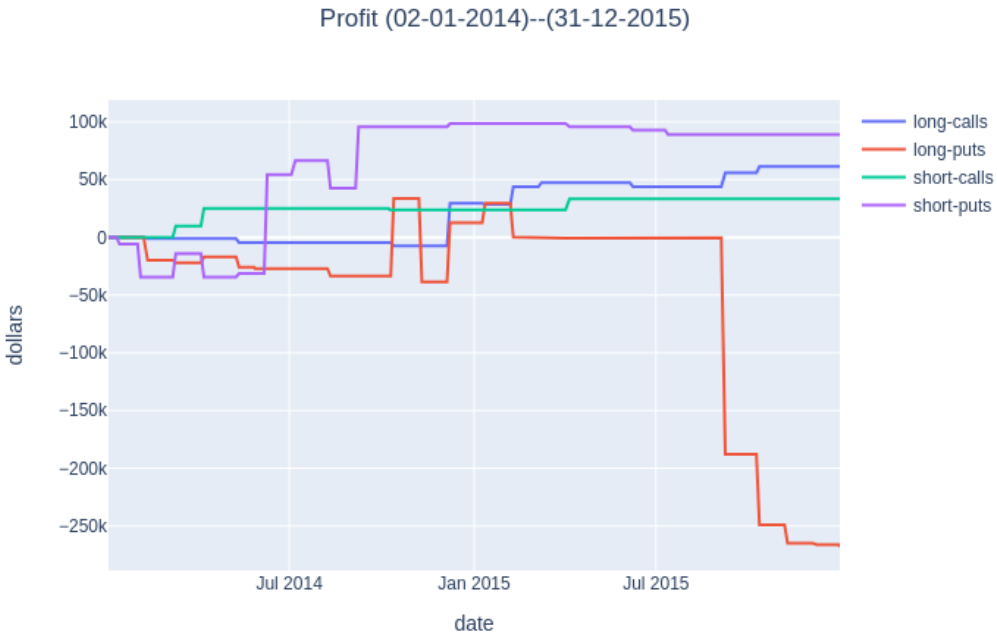

Figure 4.15: Profit evolution for the four case studies during the third test period

5

Conclusion

Contents

5.1 Overview

This work presents a method of investing in the financial market, in particular option market, different from what is currently commonly used. Using a conjugation of a machine learning algorithm, called genetic algorithm, and financial techniques, to control investment risk, the proposed approach manages to yield good and stable returns in option trading.

Starting with the machine learning algorithm, two interlinked genetic algorithms were implemented with the objective of forecasting implied volatility signals of companies in the american financial market. The first uses technical indicators applied to implied volatility signals to culminate in a solution capable of forecasting the movement of these signals. The second genetic algorithm specializes in improving the first one by finding the best configuration of the the first algorithm's hyper parameters.

Finally, after finding the best solution to forecast the behaviour of implied volatility signals, the trading simulation trades options based on the assumption that option prices are directly correlated to the respective company's implied volatility value. This trades are monitored by a series of financial techniques to diminish investment risk. The simulator tests four different case studies, in the form of type of positions (long/short) and type of options (call/put) combinations, for three differed test periods of two years each.

The conclusions drawn from this work results are presented in this chapter as well as some limitations that may have been found during its development. Lastly, some directions that future work can take to surpass this work limitations, are given.

5.2 Conclusion

The results analysed in the previous chapter prove that option trading based on implied volatility forecasting is a valid approach for profitable investment in the financial market. Implied volatility is a very complex signal that has a multitude of outside influences which makes accomplishing near perfect forecast of its movement close to impossible. A clear limitation in the algorithm is the choice of technical indicators. As two of the five technical indicators are responsible for the majority of the forecast computation, the solution suffers a limitation in its forecast complexity. Besides this fact, the machine learning step of this work accomplishes a good enough prediction that allows for the trading simulator to produce satisfying results. This does not mean that further improvements are not recommended. A more reliable forecast would benefit the outcome of this work as this is the biggest constrain for improving overall results,

Following the machine learning step, the trading simulator yielded promising results. Out of the four case studies two stood out: Long calls and short puts. Long calls repeatedly presented the biggest Rate of Investment of the four making it the most capital efficient case study. The only problem with this case study is the low number of adequate investments found by the machine genetic algorithm's solution. For

this reason, in spite of the high ROI values, the long calls case study did not yield the most profitable results. This was achieved by the short puts case study. Besides having a lower ROI than long calls, the increased number of opened positions resulted in the most profitable case study, with an average profit per year of 14,48% of the initial capital.

Altogether this work demonstrates that, according with the assumptions made in the introductory chapter, Implied Volatility forecasting can be used to trade options in the financial market, being a valid strategy, capable of yielding satisfactory results.

5.3 Future work

In spite of the good results produced by this work, there are some aspects that, if addressed in the future, could substantially improve its performance and results. Furthermore, the already mentioned limitations of this work can easily be addressed in the future. The following are some of the solutions and approaches that could be later implemented.

- As the option database is divided in trading days, each file consists in a .csv file with millions of entries, each for every option traded in that day. If a different structure was adopted, for example dividing it also by company, the access time of this data would diminish, resulting in a much faster trading simulator.
- The time the training phase takes could also be improved. A future work could try to run the genetic algorithm responsible for the implied volatility forecast with the hyper parameters of this work GA2 solution chromosome. Eliminating the second genetic algorithm might decrease the forecasting score but would also significantly decrease the training phase running time.
- In order to improve the genetic algorithm's forecast capability, different technical indicator should be tested and fundamental analysis should also be implemented.
- Other machine learning algorithms, like a neural network, could be implemented, rather than genetic algorithm, to compare the the capability of forecasting implied volatility signals movements.
- In the trading simulator, loss controlling mechanisms could be applied like trailing stop-losses.
- To better analyse which options represent the best investments, one should experiment with different ranges of in-the-money and out-of-the-money.

Bibliography

[1] F. Black, “How we came up with the formula,” *The Journal of Portfolio Management*, 1989.

[2] J. F. d. E. U. N. d. L. . Matos, “Derivatives,” no. October, 2008.

[3] J. C. Hull, “Options, Futures and Other Derivatives,” *Pearson Education Inc, Boston,*, pp. 9–26, 2012. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-9230-7_2

[4] J. B. Belo, “Innovative Options Data Signal for Trading with Technical Indicators and VIX optimized by a GA,” no. May, 2019.

[5] F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *Journal of Political Economy*, vol. 81, no. 3, pp. 637–657, 1973.

[6] D. Sornette and J.-P. Bouchaud, “The Black-Scholes option pricing problem in mathematical finance: generalization and extensions for a large class of stochastic processes,” *Journal de Physique I*, 1994. [Online]. Available: http://jp1.journaldephysique.org/articles/jp1/abs/1994/06/jp1v4p863/jp1v4p863.html

[7] G. Cohen, *Options made easy : your guide to profitable trading*, 2005. [Online]. Available: http://www.loc.gov/catdir/toc/fy0601/2005923851.html

[8] CBOE, “VIX - White Paper,” 2019.

[9] R. Engle, “GARCH 101: The use of ARCH/GARCH models in applied econometrics,” *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 157–168, 2001.

[10] D. A. Coley, “An Introduction to Genetic Algorithms for Scientists and Engineers,” *An Introduction to Genetic Algorithms for Scientists and Engineers*, 1999.

[11] M. E.-g. Talbi, *Metaheuristics : from Design to Implementation Single solution-based metaheuristics*, 2009, vol. 2009, no. 479. [Online]. Available: http://www.wiley.com/go/permission.

[12] M. F. Dicle and J. Levendis, “Historic risk and implied volatility,” *Global Finance Journal*, no. June 2018, p. 100475, 2019. [Online]. Available: https://doi.org/10.1016/j.gfj.2019.100475

[13] H. Wang, “VIX and volatility forecasting: A new insight,” *Physica A: Statistical Mechanics and its Applications*, vol. 533, p. 121951, 2019. [Online]. Available: https://doi.org/10.1016/j.physa.2019.121951

[14] L. V. Ballestra, A. Guizzardi, and F. Palladini, “Forecasting and trading on the VIX futures market: A neural network approach based on open to close returns and coincident indicators,” *International Journal of Forecasting*, vol. 35, no. 4, pp. 1250–1262, 2019. [Online]. Available: https://doi.org/10.1016/j.ijforecast.2019.03.022

[15] W. Kristjanpoller, A. Fadic, and M. C. Minutolo, “Volatility forecast using hybrid Neural Network models,” *Expert Systems with Applications*, vol. 41, no. 5, pp. 2437–2442, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.09.043

[16] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, “Forecasting volatility with a stacked model based on a hybridized Artificial Neural Network,” *Expert Systems with Applications*, vol. 129, pp. 1–9, 2019.

[17] B. K. Grace, “Black-Scholes option pricing via genetic algorithms,” *Applied Economics Letters*, vol. 7, no. 2, pp. 129–132, 2000.

[18] G. Santamaría-Bonfil, J. Frausto-Solís, and I. Vázquez-Rodarte, “Volatility Forecasting Using Support Vector Regression and a Hybrid Genetic Algorithm,” *Computational Economics*, vol. 45, no. 1, pp. 111–133, 2015.

[19] “Stock Options Trading Tools - Trader Information, Resource.” [Online]. Available: https://marketchameleon.com/

[20] “Cboe Global Markets.” [Online]. Available: https://www.cboe.com/

[21] “Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” [Online]. Available: https://finance.yahoo.com/

[22] “Stock Split History.” [Online]. Available: https://www.stocksplithistory.com/