Option Trading Based on Implied Volatility Forecasts using Genetic Algorithm

Extended Abstract of the MSc. Dissertation

Francisco Sá Reis Machado e Costa
Instituto Superior Técnico
Supervisor: Prof. Rui Fuentecilla Neves

***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 S&P500 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.**

I. Introduction

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

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.

II. Related Work

A. 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 [2]. This formula takes into consideration several factors that influence an option value [3]. 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

B. Volatility research

Some authors have theorized a correlation between implied volatility and other volatility related signals. In [4], 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 Volatility Index (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 Chicago Board Options Exchange (CBOE) in 1993 to measure the expected market 30-day implied volatility using Standard & Poor's 100 (S&P 100) option prices [5]. In 2003 CBOE changed this signal to start using 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 [6], 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 [7] the authors model the Volatility Index Futures (VXF) dynamics using a multilayer 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 [8] tried a different approach. Instead of comparing machine learning models to volatility models like in [7], their approach focused in using NN to improve the forecast capability of Generalized Autoregressive Conditional heteroskedasticity (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 [9] 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 Genetic Algorithm (GA)’s. In [10] 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 [11] 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.

III. Methodology

A. Structure

The structure of this program, as can be seen in figure 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 2 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.

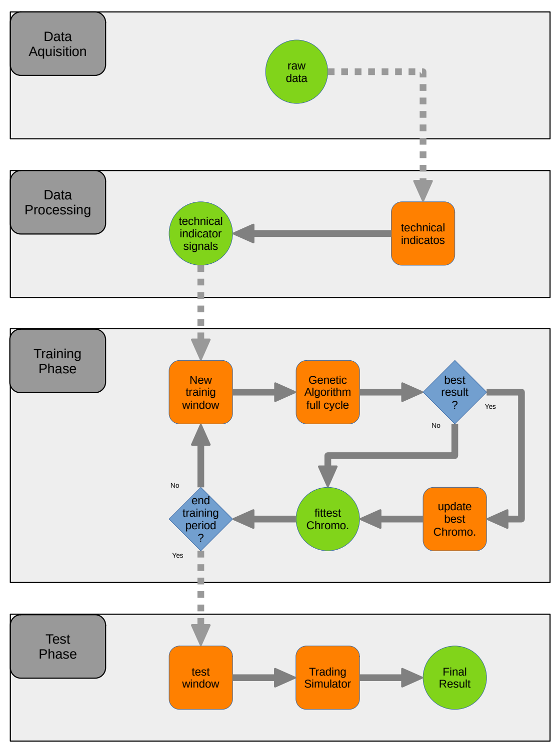

Fig. 1. Structure of the program architecture

B. 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 capitalization during the time period this work focus on, where chosen for the signals of all data types except VIX. This companies can be seen in the table I

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

TABLE I
TRADED COMPANIES

Implied volatility: The implied volatility signal has a direct correlation to option prices and so it was the subject of the machine learning algorithm forecast. This data was obtained from [12].

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 [13]

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 [14]

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 [15].

C. 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: Relative Strength Index (RSI); Rate of Change (ROC); Stochastic Oscillator (StO); Moving Average Convergence Divergence (MACD); and Crossing Exponential Moving Averages (XEMA). 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.

D. Training Phase

During the training phase the GA will use the input signals described in section III-C 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: In this work the rolling window is used as a data augmentation technique. This consists in, instead of training the system with a single training window and a subsequent test window, setting smaller, two year training windows with one year test windows, that shift over the whole training data. This behaviour is demonstrated by figure 2.

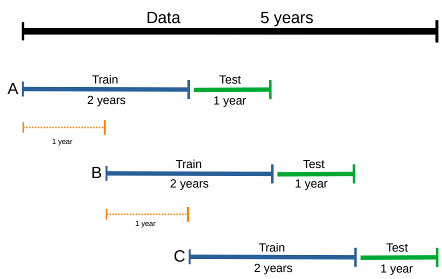

Fig. 2. Rolling Window example with a window with size of 100 days

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.

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.

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.

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; 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.

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 III-C, and five for the weights associated to each one of the indicators.

Evaluation: During the evaluation step of the algorithm, a predictive score F is computed using the weighted mean seen in equation 1, where n is the number of indicators, five in this work.

$$F = \frac{\sum^n IndicatorValue_i * IndicatorWeigh_i}{\sum^n IndicatorValue_i} \quad (1)$$

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:

$$forecast = \begin{cases} up, & \text{for } F > 60 \\ stay, & \text{for } 60 > F > 40 \\ down, & \text{for } F < 40 \end{cases} \quad (2)$$

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, ranging between 0 and 1.

$$score = \frac{NrCorrectDays}{NrCorrectDays + NrIncorrectDays} \quad (3)$$

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 [16].

-*Top method*, where the chromosomes were selected by the highest score first until all parent's slots had been filled [16].

-*Tournament method*, randomly selects a group of chromosomes. From that group the parents are the chromosomes with the highest scores [16].

-*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 [16].

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 [16].

-In order to use the *Geometric method* one has to apply the equation 4 where the value of a new chromosome's gene is the square root of the two parents' corresponding genes multiplication [16].

$$value_{G_3} = \sqrt{value_{G_1} * value_{G_2}} \quad (4)$$

-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 5, 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 [16].

$$value_{G_3} = w * value_{G_1} + (1 - w) * value_{G_2} \quad (5)$$

-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 [16].

-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. [16].

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.

E. 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 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.

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 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. 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:

$$orders = \begin{cases} 1, & \text{implied volatility increase} \\ 0, & \text{implied volatility stationary} \\ -1, & \text{implied volatility decrease} \end{cases} \tag{6}$$

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

positionorder	1	0	-1
long	open	-	close
short	close	-	open

TABLE II

ACTIONS DEPENDING ON THE TYPE OF POSITIONS AND ORDER VALUE

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 III-C 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 7. This signal is widely used in financial analysis to quantify the success of a trading strategy.

$$ROI = \frac{\text{return of investment}}{\text{cost of investment}} \tag{7}$$

IV. Results

A. Test phase results

Table III 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 3

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 III
TRADES COMPARISON FOR THE DIFFERENT CASE STUDIES AND TEST PERIODS

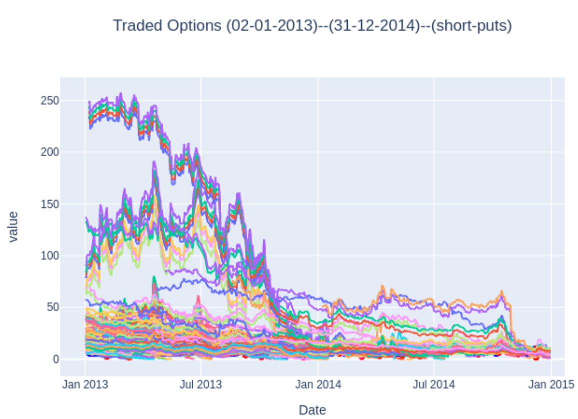

Fig. 3. Traded options' value during the second test period for the short puts case study

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. 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.

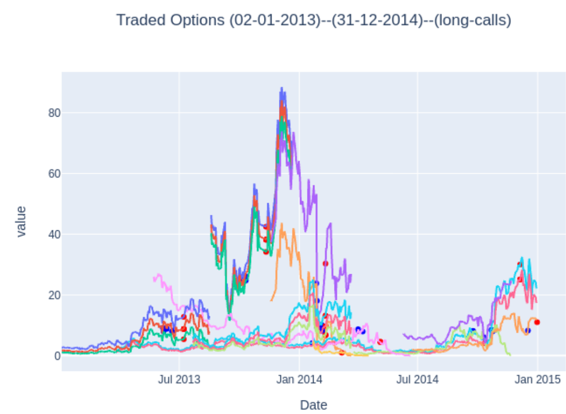

Fig. 4. Traded options' value during the second test period for the long calls case study

The difference between the number of trades of short puts and long calls seen in table III 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 IV. 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 IV
ROI AND PROFIT ANALYTICS FOR THE DIFFERENT CASE STUDIES AND TEST PERIODS

1) ROI: The ROI evolution of the four case studies for the three test periods can be found in figures 5, 6 and 7. Different from table IV, 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 5, short puts ended with a negative ROI value but by the end of the test period it was already positive. Even in 6, 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 7 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.

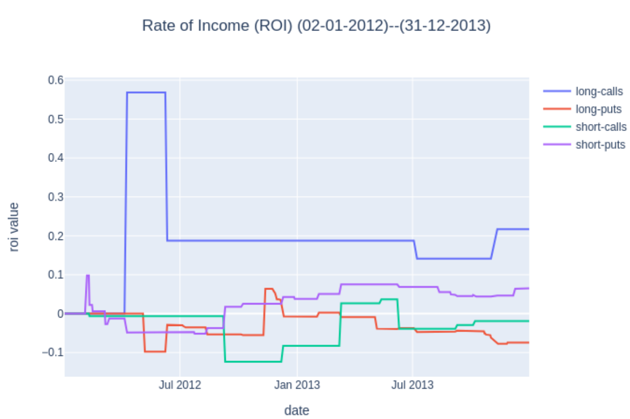

Fig. 5. ROI evolution for the four case studies during the first test period

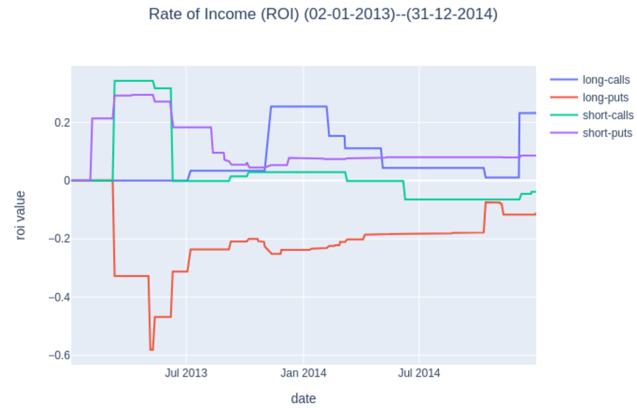

Fig. 6. ROI evolution for the four case studies during the second test period

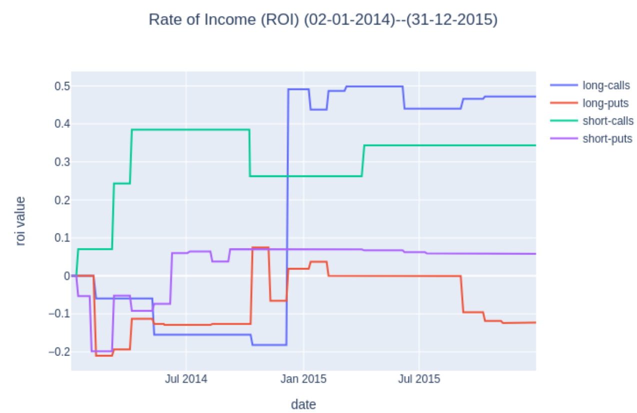

Fig. 7. ROI evolution for the four case studies during the third test period

2) Net Value: Figures 9, 10 and 11 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 8 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.

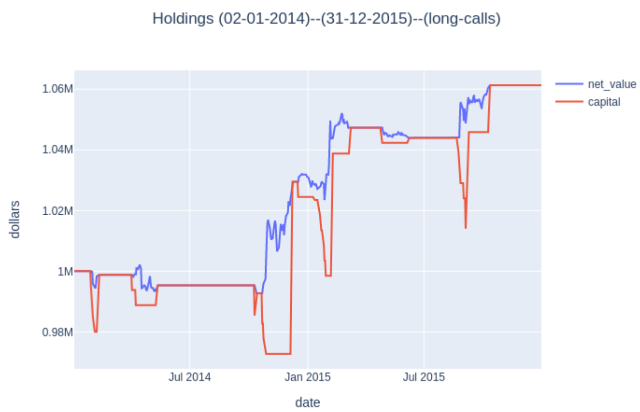

Fig. 8. Holdings of the long calls case study in the third test period

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.

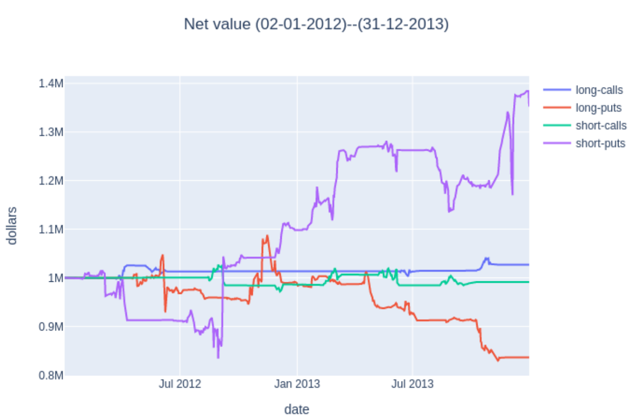

Fig. 9. Net value evolution for the four case studies during the first test period

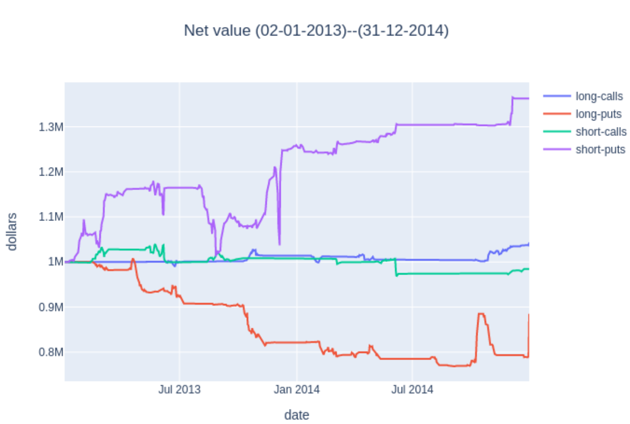

Fig. 10. Net value evolution for the four case studies during the second test period

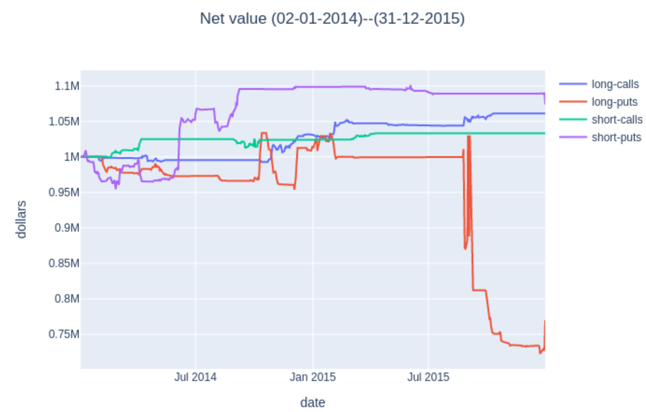

Fig. 11. Net value evolution for the four case studies during the third test period

3) Profit: The profit evolution for the four case studies in the three test periods can be seen in figures 12, 13 and 14. 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.

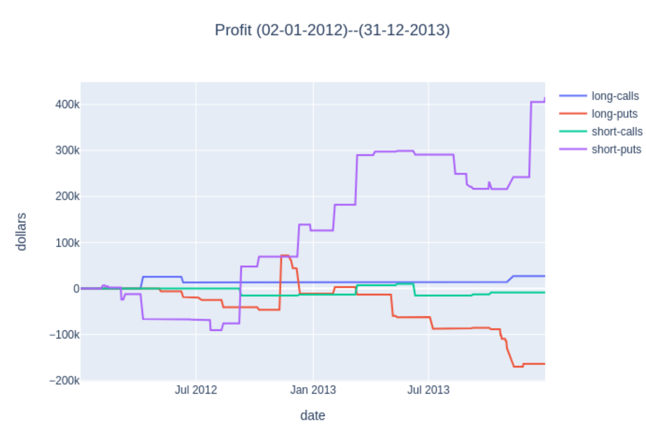

Fig. 12. Profit evolution for the four case studies during the first test period

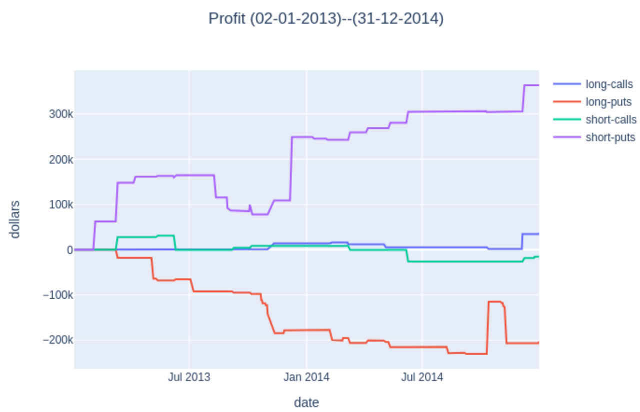

Fig. 13. Profit evolution for the four case studies during the second test period

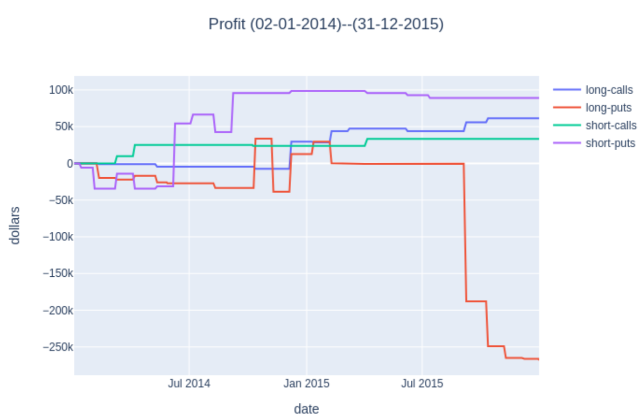

Fig. 14. Profit evolution for the four case studies during the third test period

V. Conclusion

The analysed results 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.

References

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

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

[3] 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

[4] 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

[5] CBOE, "VIX - White Paper," 2019.

[6] 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

[7] 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

[8] 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

[9] 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.

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

[11] 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.

[12] "Stock Options Trading Tools - Trader Information, Resource." [Online]. Available: https://marketchameleon.com/

[13] "Cboe Global Markets." [Online]. Available: https://www.cboe.com/

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

[15] "Stock Split History." [Online]. Available: https://www.stocksplithistory.com/

[16] 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.