Application of Support Vector Machine on Algorithmic Trading

Szklarz J[1]., Rosillo R[2]., Alvarez N[2]., Fernández I[2]., and Garcia N[2].

[1]Programmer, Izertis S.L, Gijón, Asturias, Spain
[2]Business Management Dept., University of Oviedo, Gijón, Asturias, Spain. rosillo@uniovi.es

Abstract - *The following research provides a thoughtful analysis regarding the use of machine learning techniques applied to algorithmic trading using common indexes such as the S&P500 and the Chicago Board Options Exchange Market Volatility Index (VIX). A trading simulation is carried out in order to test the efficiency of the algorithms in up trending and down trending periods. Statistical and economic performance measures are obtained and compared in order to discuss the most effective technique. The inputs used in the analysis are well-known quantitative indicators such as the Relative Strength Index and the Moving Average Convergence-Divergence. The relevance of the results lies in the use of separated training models for each kind of trend.*

Keywords: Support Vector Machines, Quantitative trading, VIX, Machine learning, ADX, RSI.

1 Introduction

Trading is one of the most ancient ways of improving the personal economic situation, either by exchanging certain goods in favorable situations or using currency, by purchasing by a low amount, and selling by a higher one. Nowadays, trading system have become well-regulated and information is spread around the world, making it possible to gather the necessary data in order to perform the most successful operation [1]. However, the immense amount of information provided becomes unbearable by humans, who need to rely in computers in order to maximize the decision-making process. Using predefined systems in this process in order to avoid manual trading is called *Algorithmic Trading*.

There are many trading techniques which are suitable to be used in financial markets. From a global point of view, they can be classified in two main families, the technical trading and the fundamental trading. Fundamental analysis uses the social information as a source of knowledge in the decision-making process [2]. On the other hand, technical analysis relies in the price movements to forecast the future situation.

Algorithmic trading is especially useful in this kind of techniques, which are the ones that will be under use in this research. Regarding technical analysis, there will be two main group of techniques that will be discussed: quantitative techniques and machine learning. Quantitative techniques use fixed rules in order to trigger the purchases and sales operations, producing a solid system that, if well designed, can provide massive profit to the user. On the other hand, machine learning, which is increasingly gathering popularity in our society, relies on artificial intelligence techniques, and by extending quantitative techniques, they can become adaptive in changing situations and provide earnings even when the initial configuration is not valid anymore [3].

As a way to carry out the research, a programming language is required, presenting a variety of options to choose from. *Matlab* is selected as the most convenient software to prepare and fulfil the analysis thanks to the different toolboxes that are provided to support machine learning and quantitative analysis, as well as direct linkage to Interactive Brokers API in order to test the results in a real environment.

The proposed analysis considers basic quantitative algorithms using indicators such as RSI, MACD and Momentum, as well as machine learning techniques such as Naïve-Bayes and Random Forest to be compared versus the use of a Support Vector Machine in order to present the best decision-making system for trading, using two different trainings for up trending and down trending periods based on the indexes data of S&P500 and VIX. Metrics such as the Sharpe Ratio, and the Maximum Drawdown are used as comparative vertexes.

2 Background

In this section, the relevant literature regarding technical analysis and machine learning is presented.

2.1 Technical Analysis

Technical analysts argue that their methods take advantage of market psychology. In particular, technical textbooks such as Murphy (1986) and Pring (1991) outline three principles that guide this behavior [4]. The first is price and volume “discounts” everything, which means that an asset’s price history incorporates all relevant information, removing the need of research asset “fundamentals.” Recent findings by Engel and West (2005), Murphy (1986) claims that asset price changes often precede observed changes in fundamentals [5]. The second principle is that asset prices move in trends. This is essential to the success of technical analysis because trends imply predictability. The third principle is that history repeats itself. This implies that the future could be forecasted relying on past behavior of price.

The technical indicators that are used are the following ones.

The Relative Strength Index (RSI) is an oscillator that compares the magnitude of a stock's recent gains to the magnitude of its recent losses and turns that information into a number that ranges from 0 to 100. It was created by J. Welles Wilder [6]. Eq. 1 displays the calculation of this oscillator, being *n* the number of periods which is a user defined parameter, although wilder has suggested to use 14 by default.

$$RSI = 100 - \frac{100}{1 + RS} \qquad RS = Avg.Gain \; / \; Avg.Loss \tag{1}$$

$$Avg = [(prev.Avg) * (n-1) + curr.Avg] \; / \; n$$

The Moving Average Convergence Divergence (MACD) is an oscillator that turns two moving averages (Eq. 2a), into a momentum oscillator by subtracting the longer moving average from the shorter moving average. It also relies in a third moving average called the MACD line in order to trigger the operation signals (Eq. 2b). It was designed by Gerald Appel and the proposed basic configuration of this set of moving averages are 12 and 26 periods for the two moving averages that get subtracted, and 9 periods for the line [7].

$$MA\,(n) = \frac{1}{n} x \sum_{i=0}^{n-1} Pm - i \tag{2}$$

$$MACD = MA(n_{fast}) - MA(n_{slow}) \;\; | \; Line = MA(n_{line})$$

The momentum is an oscillator designed to identify the speed (or strength) of price movement. It simply deducts the current closing price minus the closing price *n* days ago, being *n* a user-defined parameter. It is unknown who has created this indicator but there is extensive literature about it by the hand of Pring [8]. It is also common to normalize this indicator between 0 and 100, and with n-value of 12 (Eq. 3).

$$Mom = Price\,(today) - Price\,(n\ days\ ago) \tag{3}$$

The Bollinger Bands are a couple of volatility bands that v developer by John Bollinger and are placed above and below a moving average. They are calculated using the standard deviation (*std*) of the price from the same amount of days that the moving average is pointing [9]. By default, two *stds* (M = 2) are used in the calculation (Eq. 4).

$$L.Band = M.Band - std\big(MA(n)\big) * M \tag{4}$$

$$U.Band = M.Band + std\big(MA(n)\big) * M$$

2.2 Machine Learning

Several authors have discussed the use of machine learning techniques in quantitative trading. The techniques that are used in this paper are the following ones.

Random Forest (RF) is a supervised classification algorithm which is based in the generation of a random number of decision trees. As opposing to a single decision tree method, the information gain ratio is not taken in consideration, but the result of a voting algorithm using random test data is obtained instead. Tsai, Lin, Yen, and Chen (2011) investigated the prediction performance that utilizes the random forest classifier method to analyze stock returns [10].

Naïve-Bayes (NB) method consist in using a probabilistic classifier based on applying Bayes' theorem (Eq. 5a) with strong independence assumptions between the features and combining it with a decision rule, most likely the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule (Eq. 5b). Prof. Chate P.J (2016) researched the use of Naïve-Bayes techniques in financial markets optimization [11].

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{5}$$

$$\theta_{MAP}(x) = \arg_{\theta} max\, f(\theta \mid x) = \frac{\arg_{\theta} max\, f(x \mid \theta) * g(\theta)}{\int f(x|\vartheta) * g(\vartheta) * d\vartheta}$$

Neural Networks (NN) are computing systems based on a collection of nodes with inputs and outputs. Each of those nodes contains parameters called *weights* (the w in Eq 6), which are modified each time data travels across them. This behavior results in a change of the chosen output, which ends up conditioning the result of the whole network. Chen, Leung, and Daouk (2003) investigated the probabilistic neural network to forecast the direction of index after trained [12].

$$y = g(a) = g(\sum w_i\, x_i) \tag{6}$$

Finally, Support Vector Machines (SVM) are classification or regression methods of supervised learning based in the construction a hyperplane as the decision surface. This hyperplane can be built by the use of kernel functions, which are functions that transform a vector space in another one of superior dimension. Huang, Nakamori, and Wang (2005) investigated the predictability of financial movement direction with SVM by fore-casting the weekly movement direction of NIKKEI 225 index [13]. Equation 7 represents the problem that SVM must resolve, being *w* the normal vector to the hyperplane, and C the cost of misclassification.

$$\min\{\frac{1}{2}\|w\|^2 + C\sum \xi_i\} \tag{7}$$

$$y_i(w * x_i - b) \geq 1 - \xi_i \qquad \xi_i > 0$$

3 Strategy Scope

The design process uses the following three phases: scope choice, trade logic design and optimization. In this section, the first one is described in detail.

3.1 Indexes and Data Sources

The proposed system has been designed as a mid-term strategy, rather than an intraday system, due to the cost of the real time data needed for the intraday one. The strategy presented in this paper is recommended for the index S&P500 due to the relationship with the VIX index which is also studied. S&P500 is composed of 505 stocks issued by 500 large companies with capitalizations of at least $6.1 billion. It is seen as a leading indicator of U.S. equities and a reflection of the performance of the large-cap universe [14]. A large number of researches has been conducted using this index.

The Chicago Board Options Exchange Market Volatility Index (VIX) is an index that shows the market's expectation of 30-day volatility. It is constructed using the implied volatilities of a wide range of S&P 500 index options. This volatility is meant to be forward looking, is calculated from both calls and puts, and is a widely used measure of market risk, often referred to as the "investor fear gauge." As researched by R. Rosillo in [15] there is a short relationship between this index and S&P500, which treated accordingly could provide an accurate forecast of the S&P500 movement, especially in down trending periods.

The data from both indexes is obtained from Bloomberg Service. It is used as a table containing the following fields: close, high, low, open, volume and change for each of the days of the training and testing periods. This data is imported into *Matlab* program and stored as matrix.

3.2 Analyzed Periods

The presented strategy requires at least four periods for the testing phase, two training and two testing periods, although it is advisable to use more periods, the research has been conducted using only two due to time constraints.

The periods selected are presented in the table I. and the line plots are shown in figure 1. The chosen periods match other researches such as the one from R. Rosillo [16].

Period	Start Date	End Date
Up trending training	14/02/2003	09/10/2007
Down trending training	01/09/2000	13/02/2003
Up trending test	09/04/2009	26/03/2010
Down trending test	09/10/2007	26/09/2008

Table I: Selected periods used in the research.

4 Algorithm Design

Firstly, the quantitative trading algorithms will be presented. In second place, the machine learnings algorithm will be described, as they require an additional configuration step. After that, the SVM algorithm will be described in detail. Finally the *backtesting* process will be illustrated.

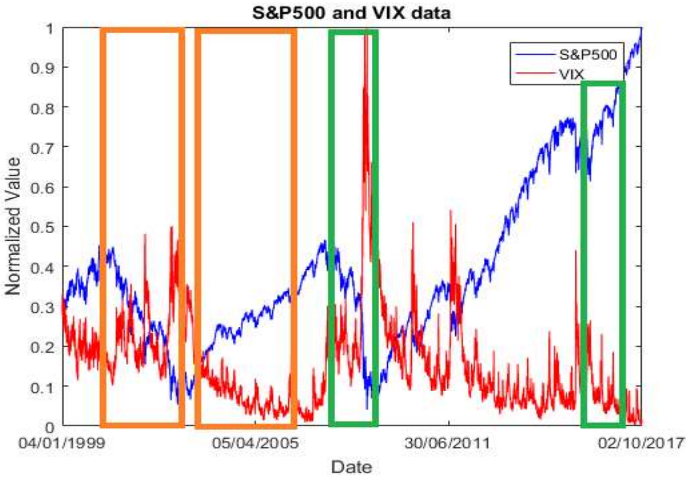

Fig. 1. Closing value of the S&P500 and VIX. Yellow frame correspond to training data and green ones designate test data.

4.1 Quantitative Algorithms

The quantitative algorithms follow the same structure: each one of them returns an indicator with a value inside a specific range. The selling (buying) orders will get triggered when that value crosses above (or below) a threshold. The algorithm will finally output a vector of trading orders (+1) for purchases and (-1) for sales that will be later evaluated by a *backtester* module. The threshold as well as the different parameters that define the algorithm are based on the default parameters used in the papers that they were taken from, passed through an optimization module which will be described in a later section. Table II shows the trading rules for each one of the quantitative algorithms.

Indicator	Purchase TH.	Sale TH.	Other value
RSI	30	70	N = 14
MACD	>0	<0	-
Momentum	15	85	N = 12
Bol. Band.	<L. Band	>U. Band	STD = 2

Table II: Trading rules for each quantitative algorithm.

4.2 Machine Learning Algorithms

Machine learning algorithms require a different approach. In first place, a training process is performed. This training is equal for every machine learning algorithm used in this paper, including SVM one. This training requires a set of inputs and an expected output for the training period. The 10 inputs to the training are the following ones [15]:

- RSI (14) at close.
- RSI (14) at previous day's close.
- RSI (14) at previous two day's close.
- MACD (26, 12, 9) at close.
- MACD (26, 12, 9) at previous day's close.
- MACD (26, 12, 9) at previous two days' close.
- VIX data at close.
- VIX data at previous day's close.
- VIX data at previous two day's close.
- S&P500 change at close.

An expected output is also needed in order to feed the training process. This output is obtained by designing a special algorithm which runs taking in consideration the whole training data vector (making it non-realistic for a test period). The trading rules used by the algorithm are as follows: "*Purchase orders are generated when the price from five days later has increased, whereas, sell orders are generated when it has decreased*".

The training requires another set of parameters that differ in each of the methods, however they are not mentioned here because the default values offered by *Matlab* are used.

Each training is stored as a model using the corresponding function. This model is evaluated during the algorithm processing phase. The output data is composed of two variables: forecast output by the model and likelihood of the forecast. Some algorithms such as the NN only returns the likelihood towards one of the values (+1) for purchase and (-1) for sale. Using this likelihood and normalizing it from 0 to 100, will make the algorithm capable of forecasting the market. Finally the optimization process is run once again in order to select the most suitable purchase and sale thresholds that will be later evaluated in the *backtester* [17].

Table III shows the selected thresholds that were obtained during the optimization process, these thresholds are the minimum percentage of likelihood for each order type.

Technique	Purchase TH.	Sale TH.	Other value
Random Forest	55%	30%	Number of trees = 100
Naïve-Bayes	40%	20%	-
Neural Network	25%	20%	Hidden layers = 10

Table III: Trading rules for each ML algorithm.

4.3 Support Vector Machine Algorithm

The SVM algorithm is the core of the research, thus it will be described in detail. The SVM implementation used in this research is known as SVM-KM being a SVM that combines a clustering technique known as K-means [18]. This added step provides a more solid output from the SVM, benefiting from the advantages of clustering methods such as the identification of outliers or non-useful values in the data.

The training phase use the same inputs and training outputs as the other algorithms. The training of the SVM requires three parameters: C, lambda and a kernel. According to R. Rosillo's research, best results for this algorithm requires the use of the parameters showed in table IV [16].

Parameter	Value
C	100
Lambda	0.0000001
Kernel	High Tail Radial Basis Function (HTRFB)

Table IV: Training parameters for SVM algorithm.

The training is performed for both up trending and down trending period and is stored in two separated matrixes.

Once the algorithm process starts, the input data is fed to the model. The evaluation of this model returns two vectors: the forecasted suitable operation for each evaluated day and the degree of belonging for each of the analyzed classes (being only two: a purchase or a sale). Combining these two outputs returns a vector of values from the range [-1, 1] that designates the selected order for each testing day. This vector is optimized using the optimization module. Results from this optimization are shown in the discussion section.

4.4 *Backtesting* Process

To evaluate the proposed algorithms a *backtester* system is designed. *Backtester* is a tool that can output performance measures receiving as inputs the price data and the algorithm order vectors [19]. *Backtester* must be configured with an initial capital input and commissions, which are chosen to be 10000$ and 0.35% of the cost from each operation, as it is an average value for brokerage fees. Also the *backtester* requires a logic in order to perform the operations. In this research, the chosen logic is as follows:

"*In up trending markets, there can be only one long operation ongoing that must be exited in order to start a new one, in down trending markets there can be only one short operation on going that must be exited in order to start a new one. In either situation, the operation must be opened with the total capital available in the account at each moment*".

This module also provides the performance of the Buy and Hold (BH) strategy as a benchmark for the evaluation. BH strategy uses the following rule: "*A purchase is made during the first day of evaluation and sale is made at last day*" [20].

5 Optimization

This section will describe the optimization phase. Firstly, quality metrics will be introduced. In second place, the optimization module will be described. Finally the post-filter module will be explained and justified.

5.1 Quality Metrics

The measures used in this paper are the following ones.

The *return* (R) is obtained subtracting the equity of each day from evaluation period. It is also presented as a percentage over the base capital, adding the sum of returns and subtracting them from the initial capital set (Eq. 8a).

The *Sharpe Ratio* (SR) is a measure of risk of the investment which is calculated as the returns divided by their standard deviation (Eq. 8b).

The *Maximum Drawdown* (MDD) is a measure of risk which displays the biggest performance fall of the return (Eq. 9). It is calculated by subtracting the peak minus the valley after each fall, and getting the maximum from them.

The *volatility* (Vol) is a measure of risk which displays the variation of the returns over time (Eq. 8c). It is calculated as the standard deviation of the return divided by their mean.

R and SR are required to be as high as possible, with a negative cipher denoting loss, and a positive one denoting benefit. MDD and Vol are required to be as low as possible, and must be necessarily positive numbers. They are usually presented in percentage.

5.2 Optimization Module

The optimization module takes as inputs a range of values that must be optimized, such as the *n* parameter from the RSI algorithm. The module will iterate along this range of values and perform a *backtest* phase for each one of them, producing a matrix that stores the metrics from the evaluation.

A condition must be set in order to rank the evaluated iterations. In this paper, the chosen condition is: *highest SR*.

The module will finally output the most successful iteration and will perform a full evaluation using the most suitable configuration of parameters [21]. It will also output a heat map of the process, in which, the axis will be the purchase and sale thresholds, and the values of the map are the SR for each iteration, where the areas with high value use hotter colors.

$$R = Cap_i - Cap_{i-1}. \qquad SR = \frac{R}{\sigma_R} \qquad Vol = \frac{\sigma_R}{E(R)} \tag{8}$$

$$MDD = Peak\ Before\ Lowest\ Valley - Valley \tag{9}$$

$$UpM = High_i - High_{i-1} \quad DownM = Low_{i-1} - Low_i \tag{10}$$

$$ADX = 100 * \frac{MA_{UpM-DownM}(14)}{UpM - DownM} \tag{11}$$

5.3 Post-Filtering

The post-filtering module is designed in order to refine the output from the algorithm as it provides a step that is usually skipped in researches. In this paper two filters will be used: Gain / Stop Loss (G/SL) and Average Directional Index (ADX), both will be analyzed and justified.

The G/SL ratio is a filtering measure which will analyze the algorithm return on each iteration and will cancel the algorithm exit operation if the gain doesn't match the fixed percentage, or will create an exit operation that was not previously planned if the price goes below the stop loss percentage that was fixed [22].

The ADX filter uses the indicator of the same name in order to check the strength of the trend for each day (Eq. 10). If the strength doesn't match the required threshold, any order that was planned for that day is cancelled. The ADX requires a *n* value that is once again set as 14, as advised by Schaap [23].

This research has been conducted, using both these filters and without using them. Comparative results are taken from this analysis and presented in the following section.

Figure 2 shows a schema of the entire process that has been explained in the previous sections.

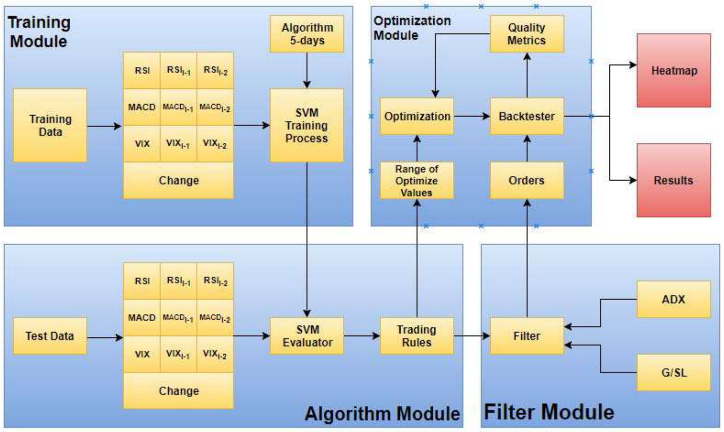

Fig. 2. Schema of the proposed algorithm system.

6 Results

The results from the research are separated in two sections for each one of the analyzed periods, each of them containing the result of the optimization and *backtester* modules for the SVM and a table containing the *backtester* results for each experiment. There is a third section that presents an exhaustive discussion of these results, as well as future work suggestions in order to extend the research.

6.1 Up Trending Analysis

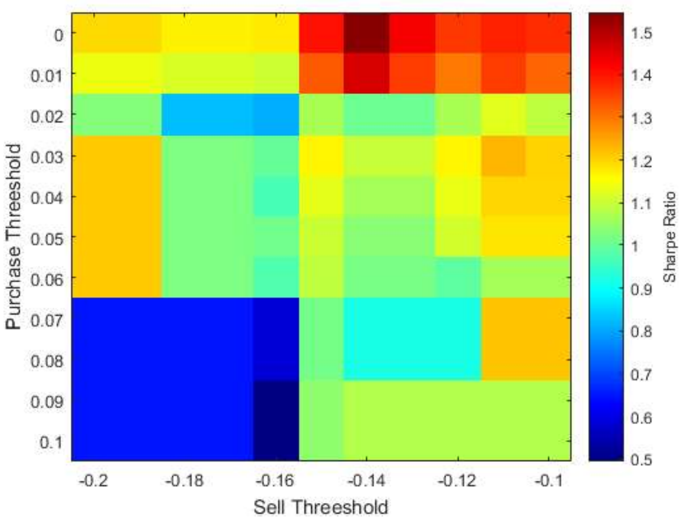

Fig. 3. Heat map of the trading rules analysis in the up trending period, using the SR as the value and the purchase and sale threshold as the axis.

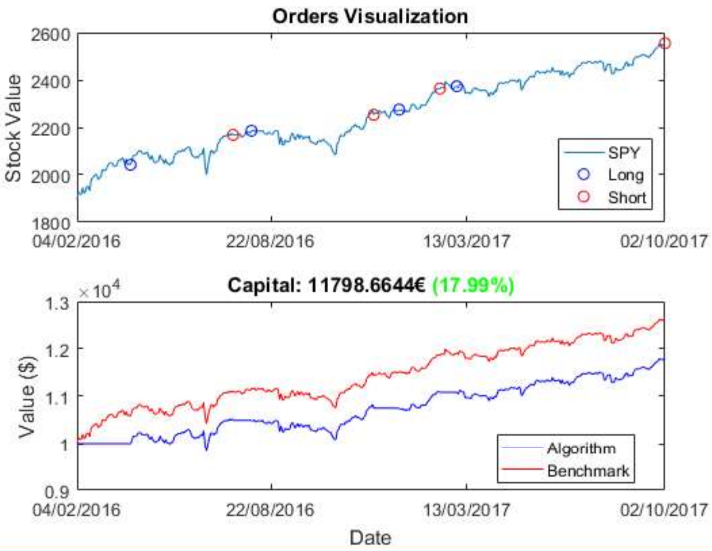

Fig. 4. *Backtester* output from SVM algorithm in up trending period, showing in upper chart the selected orders alongside the close price of S&P500, and in the lower chart the equity during the testing period vs the equity that would be obtained using Buy and Hold strategy.

Algorithm	R (%)	SR	MDD (%)	Vol (%)
RSI	19.47%	1.39	3.45%	5.9%
MACD	14.28%	0.88	4.69%	3.9%
Momentum	18.41%	1.39	2.52%	5.76%
Bol. Bands	14.57%	1.09	3.12%	4.34%
Random Forest	12.86%	1.04	3.46%	4.08%
Naïve-Bayes	14.37%	1.03	3.65%	4.42%
Neural Network	16.56%	0.94	4.36%	4.68%
SVM	11.78%	1.03	2.13%	3.18%
G/SL filter	12.7%	0.77	5.02%	4.23%
ADX filter	17.99%	0.92	4.6%	4.87%

Table V: Results for each algorithm in up trending period.

6.2 Down Trending Analysis

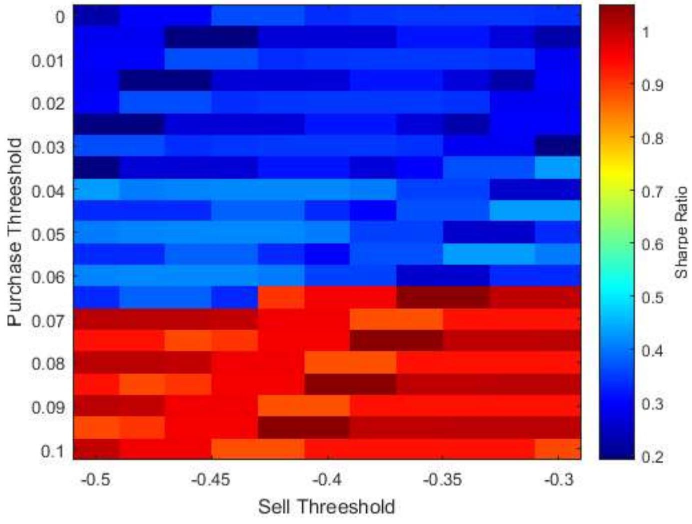

Fig. 5. Heat map of the trading rules analysis in the down trending period, using the SR as the value and the purchase and sale threshold as the axis.

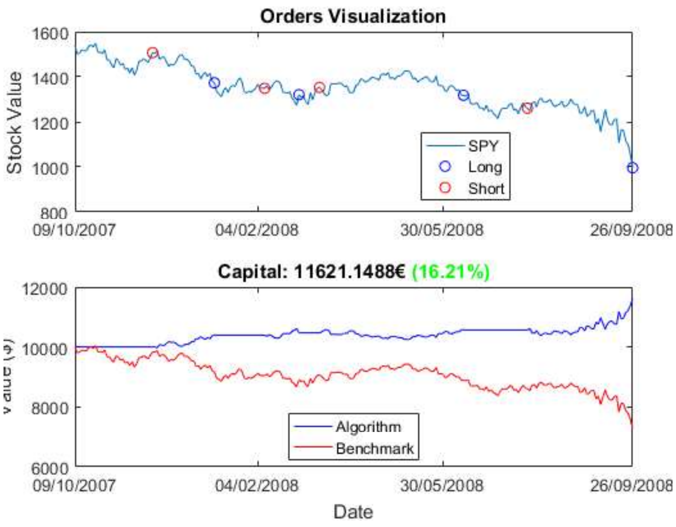

Fig. 6. *Backtester* output from SVM algorithm in down trending period, showing in upper chart the selected orders alongside the close price of S&P500, and in the lower chart the equity during the testing period vs the equity that would be obtained using Buy and Hold strategy.

Algorithm	R (%)	SR	MDD (%)	Vol (%)
RSI	6.98%	0.95	2.90%	2.54%
MACD	6.58%	0.61	3.6%	2.29%
Momentum	6.27%	0.86	2.95%	1.84%
Bol. Bands	7.41%	1.02	2.63%	2.32%
Random Forest	5.26%	0.65	3.02%	1.7%
Naïve-Bayes	8.93%	0.56	7.04%	1.89%
Neural Network	8.05%	0.57	4.51%	2.68%
SVM	22.53%	1.67	2.09%	4.48%
G/SL filter	13.56%	0.85	4.43%	1.91%
ADX filter	16.17%	1.13	3.59%	2.55%

Table VI: Results for each algorithm in down trending period.

6.3 Discussion

As shown in table V and VI, most algorithms output a positive income, except MACD one, which incurs in losses. So far, the returns from all algorithms tend to be similar, which is caused due to the short size of the testing period However, this size was required due to the short duration of the trends that were required to analyze.

The B&H strategy outperforms every algorithm in the up trending period, which is the expected result due to the strong dependence with price movement presented by this strategy, which is also a weakness in a down trend.

The SVM algorithm has a pretty high performance in the up trending period, although not being able to beat the B&H. However it is noticeable that in up trending periods SVM becomes a risky option as the MDD and Vol results show. There are other algorithms which present a higher SR such as RSI and NN. RSI's performance goes as expected as it appears in other researches such as the one from R. Rosillo [19]. Also NN presents a great performance, reinforcing the research by Chootang [24].

In down trending periods, the first fact to notice is the greatly performance loss that all quantitative algorithms presents compared to the up trending periods. This fact supports the initial premise regarding the need of machine learning techniques in order to adapt to changing situations in the stock market. Furthermore, the SVM shows a strong performance, which coincides with the expected result as researched by R. Rosillo in [18] and supporting other researches that point the adaptive traits from the SVM algorithm.

Regarding the filters, it is noticeable that the gain and stop loss filter improves the result in both periods. This is a result of using security measures such as the Stop Loss in order to protect the assets, which is strongly recommended by many investors, and as shown in the results, it is perfectly compatible with the SVM. On the other hand, the ADX filter reduces the performance of the SVM. This is due to the fact that this filter restricts the amount of outgoing operations from the algorithm, making it warier from the risks. Safer algorithms are usually expected to have less income.

Regarding the optimization process it is noticeable that the SVM tends to have better results the more sure it is in the predictions in down trending periods, however the phenomena gets reverted in the up trending one. This confirms once again the hypothesis proposed by R. Rosillo [18] about the use of this strategy in down trending periods rather than up trending.

7 Conclusion and Future Work

The SVM algorithm has been trained using different trainings for each period, and fed with common technical indicators such as RSI and MACD, and the VIX index. It has been also optimized with a Gain / Stop Loss and an ADX filter, and it has gone under a parameter-optimization process.

The results show that the SVM doesn't return the best results in up trending periods, being outperformed by RSI and B&H systems. However, it shows a strong performance in down trending periods making it more suitable than quantitative techniques and other machine learning systems. The results improve further when applying the Stop Loss filter.

As further work, it would be advisable to reproduce the current research in a higher amount of. Another interesting improve would be to develop a decision-logic to make the SVM autonomous in the training selection step.

8 References

[1] R. Oka C. M. Kusimba. (2008) The Archaeology of Trading Systems.

[2] D. Seng (2012) Fundamental Analysis and the Prediction of Earnings.

[3] R. Huerta (2013) Nonlinear support vector machines can systematically identify stocks with high and low future returns.

[4] Murphy J. J. (1999) Technical analysis of the financial markets.

[5] C. Engel and K. D. (2001) West Exchange Rates and Fundamentals.

[6] J. W. Wilder (1978) New Concepts in Technical Trading Systems.

[7] Gerard Appel (2008) Understanding Macd (Moving Average Convergence Divergence).

[8] M. Pring (2009) Definitive guide to momentum indicators.

[9] J. Bollinger (2001) Bollinger on Bollinger Bands.

[10] Tsai, Lin, Yen, and Chen (2011) Predicting stock markets return by classifier ensembles.

[11] Chate P.J (2016) Stock Market Prediction and Analysis Using Naïve Bayes.

[12] Chen, Leung, and Daouk (2003) Application of neural networks to an emerging financial market: Forecasting and trading the taiwan stock index.

[13] Huang, Nakamori, and Wang (2005) Forecasting stock market movement direction with support vector machine.

[14] D. Indrani (2013) Understanding the S&P 500: This Index Offers a Lot of International Exposure.

[15] R. Rosillo (2013). The effectiveness of the combined use of VIX and Support Vector Machines on the prediction of S&P 500.

[16] R. Rosillo (2012). Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies.

[17] Y. Nevmyvaka (2006) Reinforcement Learning for Optimized Trade Execution.

[18] D. Weiguo W. Li W. Yijang (2012) An Improved SVM-KM Model for Imbalanced Datasets.

[19] S. D. Campbell (2005) A Review of Backtesting and Backtesting Procedures.

[20] D. M. Muriuki (2015) A comparative analysis of stop loss and buy and hold strategies at the Nairobi Securities Exchange.

[21] R. Pardo (2008). The evaluation and optimization of trading strategies.

[22] A. W. Lo (2013). When do stop-loss rules stop losses?

[23] C. B. Schaap [2006] ADXcellence: Power Trend Strategies.

[24] C. Chootong (2012) Trading Signal Generation Using A Combination of Chart Patterns and Indicators.