Predicting Economic Releases using Genetic Support Vector Machines

MSc Thesis
Extended Summary

Tomás Valbordo Carvalho[1]

***Abstract*— This work's goal is financial markets forecasting. More specifically, it aims at predicting exchange rate market behaviour during isolated periods: the period after the release of macroeconomic indicators. Soft computing techniques (Support Vector Machines and Genetic Algorithms as a hybrid model) are used to predict Foreign Exchange market returns. The innovation lies not only in the used hybrid model, but also in a combination of predictive factors (including fundamental, technical and emotional) and the focus on proximity to real world practices rather than a theoretically-leaned overview. The model includes variables to increase the ability to trade under different scenarios, including unpredictable and unstable times. The Genetic Algorithm is used to pinpoint three different goals: Support Vector Machine hyperparameter tuning; finding the optimal periodicity of the technical indicators used and selecting the back-testing variables. The model was also tested under different conditions, namely trading a different instrument (future contract of the Standard & Poor's 500 index) and without period isolation (trading continuously). The results were positive and consistently beat the corresponding benchmarks in the analyzed periods, in and out-of-sample. The success is verified not only in the macroeconomic release isolation but also (and with even better performance across all the used metrics) when applied to different conditions.**

I. INTRODUCTION

As stated by Grinblatt [6], "Finance is the study of trade-offs between the present and the future". What players do in financial markets is try to forecast the movement of financial assets' value and invest according to their beliefs, capital restrictions and timings. Different investors define beliefs based on access to information and ability to infer from it, and the difference between investors' rationale and environment creates a market where supply and demand sustain trading.

There are many aspects to be considered when trading, as our perception cannot control and track information with the necessary efficiency to always be in position to perform profitable trading. The escalation of computing power in the last decades came to revolutionize the trader-developer relationship. Decision making and parameter optimization are some of the many use cases for machine learning techniques, and the present, fast-paced technological atmosphere fueled the already existing potential for this approach to problem-solving by providing affordable and powerful hardware, together with increased access to more reliable data.

Departamento de Engenharia Informatica
[1]IST-Taguspark, Universidade Técnica de Lisboa, 2744-016 Porto Salvo, Portugal *tomas.carvalho@tecnico.ulisboa.pt*

II. RELATED WORK

A. Financial Markets

Financial Markets are one of the most challenging areas to be involved in. It is constantly evolving and adapting, it is fast-paced, and offers opportunity for very distinct approaches. Players have different objectives and behaviors, and can range from an individual potentiating the growth rate of their savings to corporations diversifying their income sources or simply using different financial instruments to mitigate risk.

Efficient Market Hypothesis According to Malkiel [12], "A capital market is said to be efficient if it fully and correctly reflects all relevant information in determining security prices". In practice, this means that it is impossible to profit based on a given information set about the valuation of an asset, as its implications on what would be a fair price for the financial instrument are already reflected in the price. Whenever new information is released, prices react to new information quickly and to the right extent. Under this hypothesis, the only way to get higher returns (or Profit and Loss - P&L) is by taking on more risk, implying that active money management does not work.

Technical Analysis is one of the most used tool-sets in the trading world. Even if unconscious, most players in the Financial Markets use or have used Technical Indicators to drive their trades. The base concept is simple: to trade based on history, finding and drawing conclusions from price and volume behavior in the past in order to try to predict what will happen in the future.

Fundamental Analysis is another tool-set available to most market players who are willing to understand what drives markets and try to price their assets (or seek opportunities). Nowadays, a lot of public information is available to track financial health and future perspectives, be it in a macroeconomic environment through reports released by the government (this topic in addressed later with more detail), or a microeconomic one, through companies' annual reports and balance sheets, for example, that enlighten market players about what would be the intrinsic value of an asset.

Futures Trading Futures are a type of financial derivative that reflect the price of a good, either tradeable (oil, corn, wheat, ...) or non-tradeable (interest rate, ...), in a future date. On paper, a future is a contractual agreement made between two parties through a regulated futures exchange. Each contract specifies the quantity and quality of the item, expiration date, and all the details of the transaction except the price, which is set in the trading process.

The futures market is a *Zero-Sum Game*, meaning that there are no net winners or losers. Every-time a contract is bought, it means there is a seller on the other side of the trade, so all losses suffered by one trader are wins of others. It is also extremely liquid[1], risky[2] and complex by nature.

Foreign Exchange Trading The Foreign Exchange Market (FOREX) is the largest and most important financial market in the world. In today's globalized world, fluctuation between different currencies affects the majority of the corporative activity and even most people's everyday life without them even noticing it. With an average daily trading volume of around five trillion dollars per day [15], FOREX is traded for different purpose ranging from commercial reasons (to mitigate currency risk) to speculative ones (short term trading, trying to profit from fluctuations in price).

According to Dolan[2], the following are the main building blocks of currency prices:

- ***Economic Data Reports*** As drivers of government's economic decisions and monetary policy, these reports serve as input for both policy makers for decision-making and for the market participants for gauging the state of the economy.
- ***Interest Rate Levels*** are the single most important determinant of a currency's value. Not only their present status, but also its future perspectives: direction, expected limits, and pace. Typically, lower interest rates point to a weaker economic outlook and the probability of lower interest rates ahead, having a negative impact on the currency; higher interest rates point to economic optimism, usually supporting the currency.
- ***Monetary Policy*** consists on the actions of a central bank, currency board or other regulatory committee that determines the size and rate of growth of the money supply, which in turn affects interest rates.
- ***International Trade and Investment Flows*** also determine (and are affected by) the relative strength of a currency. Companies transact goods and set payment schemes in different currencies to, for example, make the deal more attractive for the counterpart.
  Another relationship between International Trade and markets corresponds to the Currency Reserve Management. Countries with large trade surpluses will accumulate reserves of foreign currency over time.
- ***Geopolitical Fundamentals*** including territory disputes, trade conflicts, terrorist attacks, wars, and political elections in major economies.
- ***Investors Sentiment*** is also a key factor. Humans are emotional. Emotion is uncertain. Emotions spread, usually part ways with rationality, and can have an overwhelming effect on the markets. More often than not, traders see their predictions and valuations missing the target because of collective fear or excessive optimism.

[1]traded volume is high

[2]because of its leveraged nature, variations in future price may result in high gains or losses, see the concept of *margin* in futures market context, explained in [9]

B. Data Processing

As defined by [16], Data Mining is an analytic process designed to explore large-scale data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. It is an interdisciplinary field merging concepts from allied areas such as database systems, statistics, machine learning, and pattern recognition.

Classification is the task of attributing a class (or categorical label) to an unlabeled point Y_t given dimensions $X_{t1}...X_{tc}$. To build a model, it is required to have a dataset of classified points and the corresponding dimensions, train the model using one of the many existing prediction approaches (different approaches suit different problems, but there is a clear overlap and wide range to choose, from Machine Learning algorithm's such as Support Vector Machines, that will be used in the present work; to more statistical-focused approaches such as the Autoregressive Integrated Moving Average (ARIMA)[3] model). An example of class labels, according to financial time series, could be "Positive daily returns" or "Negative Daily Returns", and could be used to understand if a trader's position should be long, short or flat[4].

C. Soft Computing

Genetic Programming is a class of meta-heuristic optimization algorithms aimed at solving problems without a deterministic solution, and that comes from the theory of evolution's concept of natural selection. Through processes such as ***crossover***, ***mutation***, ***fit*** and ***selection***, the algorithm starts from an hypothesis space (each hypothesis defined by a set of parameters, assigned randomly from a pool of possibilities) and works its way to get to the set of parameters that deliver the best result for the designated problem.

According to Goldberg [4], the procedure of a simple, generic Genetic Algorithm can be as follows:

Algorithm 1 Genetic Algorithm

1: Set $t = 1$. Randomly generate N solutions to form the first population, P_1. Evaluate the fitness of solutions in P_1.
2: *Crossover* - Generate an offspring population Q_t as follows:
   1) Choose two solutions, x and y from P_t based on the fitness values.
   2) Using a crossover operator, generate offspring and add them to Q_t.
3: *Mutation* - Mutate each solution x_{Qt} with a predefined mutation rate.
4: *Fitness assignment* - Evaluate and assign a fitness value to each solution Q_t based on its objective function value and infeasibility.
5: *Selection* - Select N solutions from Q_t based on their fitness and copy them to P_{t+1}.
6: If the stopping criterion is satisfied, terminate the search and return to the current population, else, set $t = t + 1$ go to Step 2.

Support Vector Machines are a binary classifier model that use decision boundaries to separate points belonging to each class or category. This optimal decision boundary, which is the classifier with the best accuracy and best

[3]Generalization autoregressive model used for information extraction and forecasting in time series. The purpose of each of these features is to make the model fit the data as well as possible." [17]

[4]*Long* refers to buying the stock; *Short* refers to short-selling the financial asset (selling the asset and buying it only later); *Flat* refers to having no open position

generalization ability, is also called ***Maximum Margin Hyperplane***, and it is the one that maximizes the distance from the plane to any of the points that are positioned closest to the boundary (called support vectors) - all the other points are irrelevant for the hyperplane selection process. As the separation of the points may not be exact, error can be added to the model in order to control how many points are misclassified[11].

The hyperplane can be defined, given an n dimensional feature vector $x = (X_1, \ldots, X_n)$, as:

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n = \beta_0 + \sum_{i=1}^{n} \beta_i X_i = 0$$

If the hyperplane equation is re-written using inner products:

$$y = \beta_0 + \sum \alpha_i y_i x(i) \cdot x$$

As for classification, imagine a binary solution space represented by two labels, namely $y \; \varepsilon \; \{-1,1\}$:

$$y = \begin{cases} 1, & \text{if } \beta_0 + \sum_{i=1}^{n} \beta_i X_i > 0 \\ -1, & \text{if } \beta_0 + \sum_{i=1}^{n} \beta_i X_i < 0 \end{cases}$$

One of the hurdles of this approach is that, in its essence, the model is limited to linear boundaries. This problem was overcome by introducing the concept of Kernel Functions[5] into the equation, hence transforming a linear problem into a higher dimensional one, where you can classify linearly, to then get a non-linear solution in the original dimensionality.

The *Kernel Trick* is a mathematical method applied to get linear learning algorithms (such as Support Vector Machines) to learn a nonlinear function or decision boundary.

By replacing all dot products with a kernel function, a higher-dimensional space R^M *(where* $M > N$*)* can be used, without explicitly building the higher-dimensional representation. Thus, the algorithm can learn a nonlinear decision boundary in the original R^N space, which corresponds to a linear decision boundary in R^M. This allows for significant time and resource savings without losing too much accuracy (even though Kernel function parameters still need to be tweaked).

$$x_i, x_j \varepsilon \Re^N, K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_M$$
$$\textit{where } \langle \cdot, \cdot \rangle_M \textit{ is an inner product of } \Re^M, M > N$$
$$\textit{and } \phi(x) \textit{ transforms } x \textit{ to } \Re^M \; (\phi : \Re^N \rightarrow \Re^M)$$

Notice that *K(x,y)* needs to satisfy a technical condition (*Mercer* condition, see [8] for more detail) in order for $\phi(.)$ to exist.

To control and adapt the model to the desired results, a penalty parameter is often included (called C parameter), that sets the SVM optimization misclassification scope. when choosing the best margin hyperplane, for large values of C the optimization will choose a hyperplane that has the highest accuracy in terms of correctly classified points; conversely, small values of C will cause the optimizer to select the largest margin hyperplane, even if that hyperplane misclassifies more points.

Among the advantages of the Support Vector Machines are the low tendency for overfitting, the robustness in high-dimensional spaces, and the fact that it is computationally efficient.

D. Prediction in Financial Markets: Examples

A lot of authors have tried Machine Learning and other Optimization methodologies to try to predict stock market movements. From Artificial Neural Networks, to Support Vector Machines and Genetic Algorithm, several papers can be found testing a unique (or several) combination of parameters, kernels and classifiers applied to different types of financial instruments.

While some authors focus exclusively on using technical indicators as triggers and classifiers, others analyze the importance of fundamental data, or even include external Index's information as a driver to predict the price of a certain asset (for example, looking at S&P indicators and use them as a gauge to understand overall market sentiment and how it could be influencing the traded asset's price).

Among the problems addressed, there is kernel picking, prediction model comparison, profit seeking approaches, where some authors focus on the research side of the matter (just track if the market goes up or down at certain points, "feed" the parameters to the model and directly analyzing the outputs), others try to include a setting that mimics reality and a practical point of view of trading as much as possible (by including transaction costs, for example).

In the researched solutions, there is usually a common gap: real world variables, present in a trader's everyday life, are disregarded. Aspects such as risk control (through, for example, stop-loss[6] mechanisms), exposure, among others that are a must in a professional trader's activity are seldom considered, and the model ends up being oblivious to the existence of multiple setup components. This is one of the holes that this paper will try to fill in within the already vast literature in the successful "joint-venture" between Financial Markets and Computer Science.

In the analyzed works, prediction or profit seeking models were also commonly approached using a GA. Mendes et al (2012)[13] used GA to generate trading rules for the EUR/USD (Euro / United States Dollar) and GBP/USD (Great Britain Pound / United States Dollar) exchange rates. Using ten technical trading rules (five to enter, five to exit positions), the problem is set to optimize a total of thirty one parameters within the technical indicators present in the rules to generate signals. Each parameter is a gene of each individual.

[5]As explained in the next topic.

[6]Trigger that tracks the amount of money a trader is willing to risk when entering a position. If the current price corresponds to a loss of less than that value, the trader will hold the trade and hope for it to invert and either profit or close the position at the price it entered (*aka "Scratch"*); if the price movement sends it to a point where the trader is causing a loss close to or slightly higher than the stop-loss, the trader will close the position and accept the current losses, to mitigate the dimension of poor performing trades.

The results display positive returns if transaction costs are disregarded, but shows difficulty in obtaining out-of-sample profits when accounting for these costs.

III. MODEL

A. Data Importing

Event Information As mentioned, to access information about different economic releases, Bloomberg was used as a data source. Bloomberg has a relative importance metric, built based on how many Wall Street investors are subscribing alerts for a specific number release - the higher the number of subscriptions, the higher the number's importance, hence more likely it is to stir up the markets [18] - from which the most relevant releases were selected to be included in the model.

Blpapi (Bloomberg Application Programming Interface) was the interface used to import fields from Bloomberg. The input parameters for this data pull are: list of names of the economic releases to account for; start date of the analysis and end date of the analysis. From these inputs, the model possesses a routine to query Bloomberg to get a list of the releases that occurred within the analyzed period, their dates and times, the expected value and the actual released number.

The selected events were the following [7]:

- The Consumer Price Index (**CPI**) is a measure of prices paid by consumers for a market basket of consumer goods and services. The yearly (or monthly) growth rates represent the inflation rate.
- The ADP National Employment Report (**ADP**) is a monthly measure of the change in total U.S. non-farm private employment derived from actual, anonymous payroll data of client companies served by ADP, a leading provider of human capital management solutions.
- The Non-Farm Payrolls (**NFP**) indicator measures the number of employees on business payrolls. It is also sometimes referred to as establishment survey employment to distinguish it from the household survey measure of employment.
- The United States Unemployment Rate Total in Labor Force Seasonally Adjusted (**USURTOT**) tracks the number of unemployed persons as a percentage of the labor force (the total number of employed plus unemployed). These figures generally come from a household labor force survey.
- The United States Adjusted Retail and Food Services Sales Total Monthly Percentage Change (**RSTAMOM**) tracks the resale of new and used goods to the general public, for personal or household consumption. This concept is based on the value of goods sold.
- The United States New Privately Owned Housing Units Started by Structure Total (**NHSPSTOT**) track the number of new housing units (or buildings) that have been started during the reference period.

[7]The following information was retrieved directly from Bloomberg's website [10], with the exception of ADP, which was returned directly from ADP's website [7]

Conjunctural Time Series (VIX and S&P500 Futures) Conjectural time series were stored differently. After downloading the .RData files from Reuters, an R script was created to export them to a MySql database. The idea was to build a whole data set that was easy to use and that had some flexibility in how it was pulled and transformed (which was accomplished by the use of MySql Views created to handle the roll).

Main Data (Dollar Index Futures) As mentioned in the introductory part of this work, this study will be focused on the United States of America, one of the strongest economies in the world with the most traded capital markets, the epicenter of market activity.

The evaluated ***instrument*** is the United States Dollar Index (DX) [3], a leading benchmark for the international value of the USD and the world's most widely-recognized traded currency index (mediated by the Intercontinental Exchange), which tracks the overall strength of the US Dollar compared to a basket of other currencies (instead of comparing to a specific currency, as an exchange rate, this index does an in-depth evaluation of the USD by comparing it to multiple currencies). The train and test data come from the time series around the macroeconomic release dates and times.

The data for the DX data set was stored in .hdf5, as it was the format that offered the best performance to load without increasing file size drastically. Rdata files (raw, as they were imported from Reuters) were very compact, buy their read speed was extremely slow compared to hdf5, hence the migration.

B. Pre-processing

Two types of time series are used for the analysis: VIX and S&P500 daily data (the respective futures contracts, with the tickers ES and VX); and tick data for the traded instrument, the dollar index future (DX1). Both time series needed to undergo severe transformation to get to a point where they are more robust and ready to be used.

Building front contracts When the data is raw, all that was available was individual, daily files for all the traded contracts at a given point. For example, in August 2017, there are two DX contracts that are traded: the contract expiring soon, Sep17; and the one expiring in the next quarter, Dec17.

As adopters of the speculator posture, the ideal contract is the one that can guarantee the easy closing of our positions and avoid holding the contracts until expiration due to lack of opportunity to trade. To do so, volume is used to gauge which contract is going to be traded (Front Contract). As the files are organized by day and have tick data[8], the file size is a good proxy of the volume to understand, for a given day, which contract is the one to be traded.

Cleaning outliers Data cleaning procedures are required, to make sure that there is no bias in the results caused

[8]tick data is characterized by the recording of a new data point every time one of the following fields changes its value: bid, ask, last, bid quantity, ask quantity, last quantity. Its granularity can be as narrow as milliseconds. There is strong correlation between volume and the change rate of the forementioned fields, and more ticks are recorded, increasing the file size.

by time series of data points that did not occur, thus not executable. One of the inconsistencies found in the data sets was timestamps where the bid was bigger than the ask which, by definition, cannot happen, as the bid represents the highest price available on the buy side, and the ask reflects the lowest price that is being offered by sellers; if these two cross, a trade occurs.

A second method to remove outliers was to remove points where the bid and the ask were too wide (far from each other). As the front contract is being traded, where volume is usually not an issue, and periods where market gaps would seldom occur, data points where the distance between bid and ask was above a certain threshold were removed.

The final cleaning procedure was removing extreme values, using percentiles. For each day, in the tick data, the prints that are above the 99th percentile and below the 1st percentile are removed. On this step, just like in the previous one, real prints can be excluded, but the cost of having an outlier in your data is higher than losing a few extreme values around which, given that this is done at a tick-data granularity stage, there are probably values close to it.

Building Technical Indicators – for this step, TALIB library was used. This was one of the last steps before the preparation of data for training the SVM model. The selected indicators were the following:

- ***Simple Moving Average (SMA)*** is an arithmetic average of the data points comprised between current and a number of periods before current.
- ***Relative Strength Index (RSI)*** is a momentum indicator that measures the magnitude of recent price changes to analyze if the instrument is close to inverting its trend (overbought or oversold conditions).
- ***Exponential Moving Average (EMA)*** has the same fundamentals as the Simple Moving Average, the difference being that the Exponential Moving Average differentiates how it weights different periods based on distance to current period, with the objective of giving more importance to the nearest past.
- ***Momentum (MOM)*** simply measures the difference between current price and the price that was a certain number of periods before.
- ***Percentiles (PCT)*** separate the N sorted prices into 100 chunks of about N/100 observations each. Here, an input defines how many periods before current to include in the percentile calculation. The threshold parameter controls when the price is considered to be overbought (lowest percentile) or oversold (highest percentile).

C. Classification

Classes The classification of each data point was divided into three categories (where *SVM - Time Ahead* is a gene of each individual, see table I for more detail):

- ***Buy signal*** (1) if the price of the traded product was significantly above current in *SVM - Time Ahead* future time periods.
- ***Sell signal*** (-1) if the price of the traded product was significantly below current in *SVM - Time Ahead* future time periods.
- ***Hold signal*** (0) if the difference between the current price and price in *SVM - Time Ahead* periods is not significant (being *significant* measured by the parameter *SVM - Minimum Variation for Signal* of the GA).

The Hold class was added and shaped to improve the proximity to real world conditions and to mitigate the transaction cost damage coming from the increase in the number of trades.

Training vs Testing The train and test split is done at two different moments: first, all the releases are listed and split, according to a percentage set by one of the individual's genes; the second moment is after the aggregation of all the training data from the first step, and before training the SVM Model (A fixed percentage of 50% was used in this second stage of splitting, coming from trial and error).

The first moment, is an effectively chronological division and setup of how much past data the model will be using to train and classify future data points, while the second point is just a randomized way to downsize the training sample, finding the ideal percentage to increase performance while not losing accuracy.

Features The features are the variables that will be used to predict the outcome (class). In this work, the types of features that were used can be divided into three:

- ***History of the traded product***, among which lie the signals explained in the previous subsection. They measure direction (Buy, Sell or Hold) and intensity (distance from metric, which may indicate stronger or weaker signals).
- ***Conjectural variables***, to analyze the status of the market compared to its medium-term history. Percentiles of two different financial instruments for this were used: the VX (VX1), and the ES (ES1).
- ***Event-specific variables***, to understand if the release information was different than what was expected by market agents (according to Bloomberg), represented by a Boolean stating True if the there is a difference between what was surveyed and the actual release, False otherwise.

Scaling The usual process of scaling is used to transform all the features into small, similar (or close to) values, not only to increase the performance (due to the avoidance of numerical difficulties during the calculations), but also to avoid domination of attributes with a greater numerical range. In this work, the standardization was done through SciKit's [14] RobustScaler function, a method created for this purpose. Among the scaling solutions, this approach was selected because it standardizes all values to a small scale (around [0, 1], although some points fall outside the fore-mentioned range due to the use of quantiles instead of maximum and minimum values).

The scaling function is as shown in equation 1:

$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \tag{1}$$

As stated, quantiles are used instead of max-min range, making it more robust to outliers. Default Q_1 is 25 and Q_3 is 75.

This scaler (more specifically, the quantiles) are set during the training phase, and the test data is scaled using the trained scaling model. For example, if $Q_1 training$ = -10 and $Q_3 training$ = 10, when scaling a value of 17, the scaled value would be 1.35.

D. Support Vector Machines

The Machine Learning methodology selected for the prediction of financial time series in this work was the SVM model. The SciKit's Python package SVC (Support Vector Classification) function was used to train and elaborate the model.

Principal Component Analysis (PCA) In order to decrease the size of the problem and increase the performance, PCA was used. For the purpose, SciKit's Decomposition *PCA* function was used. According to the documentation, this function uses "Linear dimensionality reduction using Singular Value Decomposition[9] of the data to project it to a lower dimensional space".

The number of dimensions the whole feature space is reduced to is a gene of each individual, within a predefined range.

Kernels and Parameters Among the tested kernels, the most popular ones were used: Linear, Radial-Basis Function, Polynomial and Sigmoid functions. It is part of the genetic optimization routine to understand which of these kernels deliver the best results, together with the tuning of the hyperparameters of each trained model.

The tuned parameters are either kernel-agnostic (the penalty parameter *C* and the *time_ahead* variable that defines how far long each point will try to predict) or kernel-specific (the *Degree* parameter is only used in the Polynomial kernel, and the *gamma* parameter is not used in the Linear kernel).

E. Back-testing

The back-test routine consists in the simulation of trading activity to try to understand how a specific set of parameters would perform in the markets. It receives a set of parameters (genes for a given individual) and a prediction package (composed by three elements: a Scaler, a Principal Component Analysis trained model and an Support Vector Machine trained model), that it uses to generate Buy or Sell signals based on present data and act accordingly.

A set of rules were included to approach this simulated moment of trading to reality:

- **Initial lag.** The trading activity only starts after a predefined amount of time passes after the release of the economic data (for example, the first buy/sell for a number that was released at 14:00:00 in a given will only be possible after 14:00:10).
- **Execution delay.** After the model generates the signal, the action takes 1 period to be undertaken. If the used got a Buy signal at 14:15:00, he will only act upon it in the next data point (for 30 second data, it would be at 14:15:30).
- **Stop-loss.** When the trader has an open position, one of the possible ways for the trader to close it is to define a threshold and close it to avoid further losses.
- **Conservative execution.** The price at which the trader is able to execute it's market actions is not the market price (buying the best bid, selling the best ask), but the mid price (average between best bid and best ask).

Another important part of the developed back-tester are the exit and entry conditions. The individual has three possible states: he has an open long position, an open short position, or no open position (using trading jargon, he is *flat*).

For the individual ***to open a position***, two conditions need to be met: first, the individual needs to be in the *no open position* state; when in that state, the SVM model signal will determine what type of entry position the individual will open.

For the individual ***to close a position***, three conditions need to be met: first, the individual needs to have an open position of either of the types (*long* or *short*); second, the SVM model signal needs to be the opposite of the already opened position; finally the mid price needs to be higher or equal to the associated entry price (this last condition can be disregarded depending on the gene set of the individual).

F. Genetic Algorithm

In this Genetic Algorithm, an *Individual* the chromosomes of the individual can be divided into four generic types:

- *Sizing* parameters
- *Trading* parameters
- *SVM* hyperparameters
- *Pre-processing* parameters.

The following table enlighten how these individual are structured: figure I shows all the 29 chromosomes complemented by a description of each and the range of valid values.

Selection - For the selection process, a hybrid approach was selected between a Truncate Selection (Individuals with the highest fit would breed) and a randomized lesser number of individuals to be included in the breeding process. Elitism was also implemented, so breeders would not only generate children, but also move on to the next generation to compare fitness with the new individuals. The selected Elite was 30 percent of the top individuals of the previous generation, the randomly picked individuals would account for 5 percent of the next generation (and around 15 percent of the breeders), and the remainder of the generation would be offspring from the crossover/mutation process. The selected breeders would all move on to the next generation.

Crossover - From the individuals selected to breed, the offspring for the new generation is created. The breeding (or crossover) process consists on the generation of a new individual based on the characteristics of two other individuals

[9] factorization of a real or complex matrix, see [5] for more detail.

TABLE I: Individual Structure

Chromosome	Description	Unit	Range
Time after release	Minutes after the release with trading activity.	Integer (Minutes)	30 to 120
Tick data re-sampling size	Tick size to re-sample time-series.	Integer (Seconds)	60
Stop Loss	How much the trader is willing to lose until he closes a position and accepts the loss.	Percentage	3% to 7%
Loss Close Prevention	Allows (or forbids) the trader to close positions that have a negative P&L when the model generates a cover signal.	Bool	True/False
PCA - Number of Components	Number of components that the model reduces the features to.	Integer	3 to 19
SVM - Kernel	Kernel to be used when training the SVM model.	String	Linear, Poly, RBF, Sigmoid
SVM - C	Penalty parameter of the SVM model.	Decimal	$2^{(-3,...,10)}$
SVM - Gamma	Kernel coefficient for "rbf", "poly" and "sigmoid" functions.	Decimal	$2^{(-5,...,4)}$
SVM - Using Probability?	Whether to enable probability estimates.Slows down model but may increase accuracy.	Bool	True/False
SVM - Shrinking?	Eliminates variables to reduce size of the problem and increase performance [1].	Bool	True/False
SVM - Max iterations	Hard limit on iterations within solver. Introduced to improve efficiency.	Integer	15000
SVM - Degree	Degree of the polynomial kernel function.	Integer	1 to 4
SVM - Time Ahead	Determines how far ahead the model will try to predict.	Integer (Seconds)	600 to 1800
SVM - Train percentage	Percentage of the total sample that will be used for training.	Percentage	66.6%
SVM - Minimum Variation for Signal	Minimum variation of price that triggers a Buy or Sell Signal.	Percentage	0.1% to 1.5%
EMA - Number of Periods (x2)	Number of periods to be used in the calculation of the EMA.	Integer (Seconds)	120 to 720 ST; 3000 to 4800 LT
SMA - Number of Periods (x2)	Number of periods to be used in the calculation of the SMA.	Integer (Seconds)	120 to 720 ST; 3000 to 4800 LT
MOM - Number of Periods (x2)	Number of periods to be used in the calculation of the MOM.	Integer (Seconds)	120 to 720 ST; 3000 to 4800 LT
RSI - Number of Periods (x2)	Number of periods to be used in the calculation of the RSI.	Integer (Seconds)	120 to 720 ST; 3000 to 4800 LT
DX - Number of Periods (x2)	Number of periods to be used in the calculation of the DX percentile.	Integer (Seconds)	120 to 720 ST; 3000 to 4800 LT
RSI - Threshold	Threshold value to trigger decision from RSI calculations.	Integer	1 to 50
DX - Percentile Threshold	Threshold value to trigger decision from RSI calculations.	Integer	1 to 50
VIX - Percentile Threshold	Threshold value to trigger decision from VIX percentile.	Integer	1 to 50
ES - Percentile Threshold	Threshold value to trigger decision from ES percentile.	Integer	1 to 50

(parents), inheriting one gene from one parent or the other with 50% probability.

This is a good way to find if there is gene set between two already good performing individuals that would get you into a higher fitness boundary, discovering a better solution for the problem.

Mutation - The mutation rate determines the probability of gene randomization given an individual that resulted from breeding. The objective is to avoid local extremes and randomly seeking better global solutions for the optimization problem. When setting the mutation rate, one also needs to be careful with its value being set too high, as it will make it harder for the model to converge within a specified region. It will look for spread solutions across the total search space, but it will be less likely (or take more time) to find the most optimal solution in a specific region of the search space. A mutation rate of 10% was used.

Fitness Function The Fitness function is a key element for the GA. It defines what to search for, sets what is reasonable or not in terms of comparison, and it is fully customizable from a developing perspective, as long as it results in a metric that is uniform and comparable throughout all the analyzed individuals in a population. The selected fitness function was a weighted score that would include three action avenues: raw Profit & Loss, Risk and Accuracy. Four metrics were selected to account for the three action avenues: the Sharpe Ratio, Accuracy, P&L and P&L to Max Drawdown Ratio (more detail in the EVALUATION section of this article). Each of the metrics was normalized and weighted to create a comparable score.

IV. EVALUATION

In this section, results are analyzed, not only from the main focus of this dissertation, but also an alternative scenario (referred to as *case study*).

A. Performance Evaluation

To evaluate and compare results, a list of metrics was selected. The used formulas and measuring methodologies pinpoint not only understanding and directly comparing the main result (profit), but also to analyze consistency, risk/reward relationship and the behaviour of the model throughout the training and testing period. Notice that, in the following descriptions, the term "trade" is used to describe not a single transaction, but the opening and closing of a position. Table II summarizes the used metrics, together with a brief description and formula explaining how these metrics work.

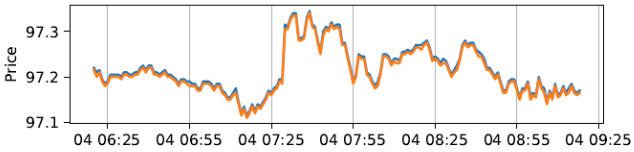
TABLE II: Performance Metrics

Metric	*Formula*	*Description*
Annualized Sharpe Ratio	$\frac{Avg(DailyReturns)*\sqrt{\#TradingDays}}{StDev(DailyReturns)}$	Risk adjusted returns - a ratio between the average value of the model's P&L curve and the corresponding Standard Deviation. The higher the Sharpe ratio, the better the risk-adjusted returns
Total P&L	$\sum DailyP\&L$	Sum of the P&L of all trades
Used Capital	–	Used capital during the trading period.
Cumulative Returns	$\frac{TotalP\&L}{UsedCapital}$	Ratio between the Total P&L and the Used Capital
Annualized Returns	$(1+CumRet)^{\left(\frac{252}{\#TradingDays}\right)}-1$	Adjustment to Cumulative Returns to make the value comparable with different periodicity strategies
P&L / Max Drawdown	$\frac{TotalP\&L}{MaxDrawdown}$	Risk adjusted measure to compare return (via Total P&L) with risk (via Max Drawdown). The higher, the better risk-adjusted returns.
Accuracy	$\frac{\#CorrectClassifications}{\#TotalClassifications}$	Percentage of right classifications of the SVM algorithm
Average Profit	$\frac{\sum ProfitPerTrade}{\#Trades}$	Mean of every trade
Number of Trades	$\#Trades$	Total number of trades
Winning Trade Rate	$\frac{\#TradesWhereP\&L>0}{\#Trades}$	Ratio between the trades with positive P&L and the total number of trades
Number of stop-loss Trades	$\#ExitsByStopLoss$	Number of trades that triggered the stop-loss criteria
Most Profitable Trade	$Max(ProfitPerTrade)$	Trade with the highest P&L
Most Costly Trade	$Min(ProfitPerTrade)$	Trade with the lowest P&L
Highest Winning Streak	$Max(ConsecutiveTradesWhereP\&L>0)$	Highest number of consecutive winning trades
Highest Losing Streak	$Max(ConsecutiveTradesWhereP\&L<0)$	Highest number of consecutive losing trades
Maximum Drawdown	$P-L$ where P = Peak value before largest drop and L = Lowest value before new high established	Maximum loss (in USD) from a peak to a trough of the model's P&L curve, before a new peak is attained
Positive Days	$\#DayswhereP\&L>0$	Number of days in the trading period where P&L >0
Negative Days	$\#DayswhereP\&L<0$	Number of days in the trading period where P&L <0

B. Case Study I - GSVM for economic release trading

This is the main case study of this dissertation: it uses an hybrid model called Genetic Support Vector Machines (GSVM, comprised by a combination of a GA and SVM), to generate forecasting signals to trade a FOREX instrument (in this case, DX) in a short period after the disclosure of macroeconomic release information. These periods are known as very active trading-wise (depending on the type and value of the release, the recent history of market volatility) as shown in figure 1, and our main goal is to use a powerful forecasting framework to try to profit from the market movement around them.

Fig. 1: Economic Release example Chart - DX price before and after the NFP release, 2016-11-04 07:30



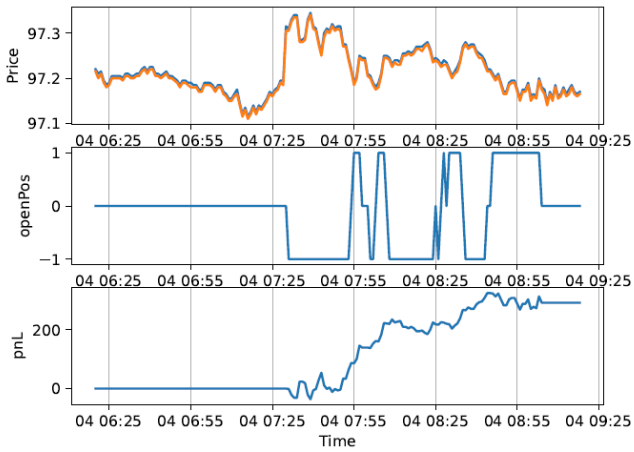
There are four separate moments in a day when a relevant macroeconomic indicator is released:

- **Before the release.** Usually slow periods, with low volume and volatility. Only minor adjustments to the market agent's positions. Seldom does a relevant agent take action during this period, as there is no certainty about what the after-release period will look like, only conjectures based on guesses.
- **During the release.** Volume reduces even more than in the previous moment. Most electronic traders hold off until the market is thinner (meaning, the distance between the best bid and best ask has reduced), as the algorithms may not be able to keep up with the quick and sudden price movements characteristic to these moments.
- **Shortly after the release.** Electronic traders return. The market is overflowed with orders, traders adjusting and acting based on the most up-to-date information about the economic release and its implications. Wide executing price range and sudden movements, hard to predict, and multiple inversions while the equilibrium price is being corrected.
- **Rest of the day.** Either slow or active, depending on the the release value and on the overall economic sentiment. Most of the action has already taken place, but under certain conditions may keep some of its pace until the market closes.

In this case study, trading activity in the third moment (*Shortly after the release*) is captured, knowing that there is a high risk that this type of movement is too random and hard to predict.

An example of the trading activity around one macroeconomic release can be found in figure 2:

Fig. 2: Trading after an Economic Release example Chart - NFP release, 2016-11-04 07:30



The top subplot represents DX price with time (as in figure 1). The middle subplot represents the trader's open position overtime. When *openPos* is 1, it means that the trader has a long open position (bought one contract, is waiting for a cover sell signal); when it is -1, the trader has a short position (sold one contract, is waiting for a cover buy signal); when it is zero, the trader has no open position. As the trader opens and closes positions, the P&L curve (bottom subplot) updates and reflects how much money the trader is making at any given time.

Results. The algorithm was able to yield positive cumulative returns and to beat the benchmark in all runs (in and out of sample, but the focus of the analysis is in the out-of-sample). One of the runs stood out, achieving positive 100% returns (doubling the initial capital) during a period when the Buy and Hold return was around -20%. Notice that this was also the best in-sample performer. In fact, the worst in-sample performer is the second best out-of-sample P&L-wise, showing less signals of overfitness than some the other individuals. Notice also that all the runs are much less volatile than the benchmark: the Buy and Hold strategy ranges 120 percentage points from -40 % to +80 % in terms of cumulative returns, while the algorithmic traders have a maximum range of 100 percentage points in the best out-of-sample individual and around 45 percentage in the other individuals.

Figure 3 displays the out-of-sample performance of the best individual in each of the five simulations performed. Table III has a more discriminated overview with the best individual of a single run in detail, including its genome and performance metrics' values.

Fig. 3: Cumulative Return chart for the different out-of-sample runs of Case Study I.

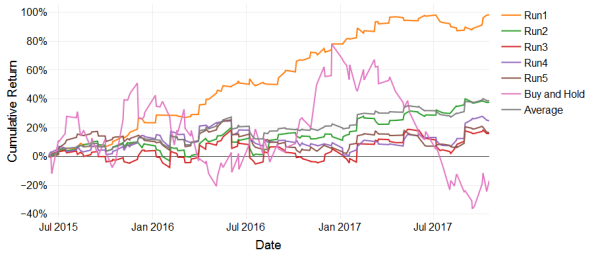


TABLE III: Case I - Run1 Best Individual

(a) Genome

Chromosome	*Gene*
Stop-loss	0.035
Loss Close Prevention	True
Time after Release	120
Tick Data Re-sampling size	60
PCA Components	15
SVM kernel	RBF
SVM C	32
SVM gamma	8
SVM using probability	True
SVM degree	4
SVM max iter	15000
SVM shrinking	True
SVM time ahead	10
SVM min profit	0.011
Train Percentage	0.66
EMA LT number of periods	70
SMA LT number of periods	60
MOM LT number of periods	50
RSI LT number of periods	50
DX LT number of periods	50
EMA ST number of periods	6
SMA ST number of periods	8
MOM ST number of periods	2
RSI ST number of periods	2
DX ST number of periods	6
RSI threshold	46
DX percent rank threshold	0.26
VX percentile to trade	0.01
ES percentile to trade	0.06

(b) Performance

Metric	*Run1*
Accuracy (%)	37.03
Average Profit	10.76
# Trades	943
Winning Trade Rate (%)	72
# Stop-loss	209
Max Profit	427.2
Min Profit	-93.2
Max Win Streak	17
Max Loss Streak	6
Max Drawdown	1288.8
Annualized Sharpe	27.67
P&L	10042.8
Used Capital	10000
Cumulative Returns (%)	100.4
Annualized Returns (%)	3.18
Positive Days	64
Negative Days	48
P&L / Max Drawdown	7.79

C. Case Study II - GSVM for continuous ES trading

The second case study is a test to the algorithm under different conditions. The following conditions changed:

- **Traded Instrument.** Instead of the DX future, the ES future contract is traded. Tick size and margin were adjusted accordingly. Notice that the ES is a directional product, with tendency to increase its value.
- **Trading Periods.** In this case study, the agent trades continuously, with no distinction between periods. To adjust to the much greater data-set size, the granularity shifted from 60 to 1800 second bars (30 minutes per bar).
- **Features.** The macroeconomic (distance to VX and ES long term percentile) and release-specific features (the type of release and difference to expected) were removed, to both simplify the calculations or because they no longer applied. It uses a total of 12 features, around half the 22 features used in the previous case.

The predictive model remained the same: the GSVM, hybrid GA with SVM in its fitness function.

Results. As in the previous case study, out-of-sample performance was favorable. All the individuals surpassed the Buy and Hold strategy, showing positive returns in all cases. Only one individual (5^{th} simulation's best performer) was able to perform positively in terms of return.

Curiously, the best performer (2^{nd} simulation's best individual) is the second least accurate individual and one of the individuals with the highest stop-loss trigger. On the other hand, this individual has the highest winning trade rate (metric at which it also performed very well in-sample).

On the other end, the worst performer (best individual of the 5^{th} run) is the most active trader and one of the individuals with the best out-of-sample accuracy. It has the best single trade profit-wise, but also the worst in terms of loss and risk adjusted returns (P&L to Max Drawdown ratio).

As in the previous case study, figure 4 displays the out-of-sample performance of the best individual in each of the five simulations performed. Table IV has a more discriminated overview with the best individual of a single run in detail, including its genome and performance metrics' values.

Fig. 4: Cumulative Return chart for the different out-of-sample runs of Case Study II.

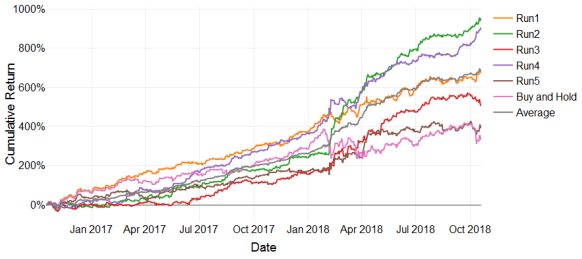


TABLE IV: Case II - Run2 Best Individual

(a) Genome

Chromosome	*Gene*
Stop-loss	0.045
Loss Close Prevention	True
Time after Release	92
Tick Data Re-sampling size	1800
PCA Components	13
SVM kernel	RBF
SVM C	32
SVM gamma	2
SVM using probability	True
SVM degree	4
SVM max iter	15000
SVM shrinking	True
SVM time ahead	15
SVM min profit	0.011
Train Percentage	0.66
EMA LT number of periods	50
SMA LT number of periods	60
MOM LT number of periods	60
RSI LT number of periods	50
DX LT number of periods	60
EMA ST number of periods	6
SMA ST number of periods	4
MOM ST number of periods	2
RSI ST number of periods	2
DX ST number of periods	2
RSI threshold	31
DX percent rank threshold	0.41

(b) Performance

Metric	*Run2*
Accuracy (%)	59.39
Average Profit	59.47
# Trades	1597
Winning Trade Rate (%)	73.51
# Stop-loss	412
Max Profit	2993.2
Min Profit	-465.5
Max Win Streak	17
Max Loss Streak	6
Max Drawdown	3822.3
Annualized Sharpe	11.18
P&L	94975
Used Capital	10000
Cumulative Returns (%)	949.8
Annualized Returns (%)	11.13
Positive Days	293
Negative Days	218
P&L / Max Drawdown	24.85

V. CONCLUSIONS

A. Summary of contributions

This work presents two case studies on financial market trading optimization:

- **Case Study I:** Hybrid model using Genetic Algorithm with Support Vector Machines to trade the US Dollar Index instrument in a short period after economic releases.
- **Case Study II:** Hybrid model using Genetic Algorithm with Support Vector Machines to trade the ES (SP500 future contract) instrument continuously.

In both case studies, a SVM model was included in the fitness function of a GA to generate the trading signals based in which the algorithm would trade, included in a hybrid framework (named Genetic Support Vector Machines). The GA influences not only the hyper-parameters of the SVM (like which kernel, C parameter and Gamma to use, see the previous section of this paper for more information) but also the calculations of the Technical Indicators to be used by the SVM as classifiers in the signal-generating process.

In the first case study that was addressed, the GSVM algorithm successfully obtained positive results in and out-of-sample. The pinpointed solutions performed very well in-sample, and always above the Buy and Hold strategy in train and test periods.

The second case study was raised by the need to understand how the algorithm would perform under different conditions, changing the problem statement but keeping the used solution as close as possible to the release prediction model. It was accomplished by maintaining the base of the algorithm structure while changing from economic release prediction to continuous trading, by changing from DX to ES as the traded instrument, and time-series granularity from 60 to 1800 seconds. In this case study, the results were even more favorable than the previous scenario: the algorithm obtained monetary gains both in and out-of-sample, always better than the Buy and Hold strategy. The out-of-sample accuracy was also much higher in this last case study than in the previous. The algorithm was not only learning better in terms of right forecast, but it also increased the financial performance of the trading agent. The explanation of the results lies on the fact that the weight of longer term settings is greater than the noise associated with short term events such as economic releases, making it more suited to fit prediction frameworks.

B. Future Work

One of the biggest challenges throughout the development process was the complexity of the problem. The inclusion of the SVM training inside the fitness function of a GA is rather intense in terms of computing power usage, and the considerable run time of the routine made it harder to tweak and test, as each modification would lead to running the routine again, which would take a long time.

The inclusion of parallel programming within Python greatly increased the performance (the population fitness calculation was split into processes, one process per individual, and all the used cores of the processor would share the load, resulting in a close to linear speed up). What would be considered to tackle this problem would be to test alternative programming languages and/or include more computing power to increase size and performance of the algorithm. Modifications such as an increase in population size and number of generations, in the GA fraction of the algorithm or the removal of the limiter of the number of iterations in the SVM fraction of the algorithm would improve the quality of the solution.

Another possible tweak to the model would be to revisit and/or find different ways to classify each data point to generate signals. A significant scope of variable types and information was included, but additional information may always add relevant predictive capabilities to the model. Together with this, the inclusion of additional economic releases would also be relevant, as there are still important releases to be included.

Finally, even though the objective of this thesis was specifically to test the proposed model, a different combination of optimization and learning algorithms could be used. This work proves the power of combining frameworks, but there could be different results if the structure is maintained but the moving parts are changed.

REFERENCES

[1] BOTTOU, L., AND LIN, C.-J. Support vector machine solvers. *Large scale kernel machines 3*, 1 (2007), 301–320.

[2] DOLAN, B. *Currency trading for dummies*. John Wiley & Sons, 2011.

[3] EXCHANGE, I. Us dollar index® futures, 2017. [Online; accessed 21-November-2017].

[4] GOLDBERG, D. E., ET AL. Genetic algorithms in search optimization and machine learning, 1989.

[5] GOLUB, G. H., AND REINSCH, C. Singular value decomposition and least squares solutions. *Numerische mathematik 14*, 5 (1970), 403–420.

[6] GRINBLATT, M., AND TITMAN, S. *Financial markets & corporate strategy*. 2016.

[7] HACKMAN, A. Adp national employment report. *ADP National Employment Report* (2018), 4.

[8] HAZEWINKEL, M. Encyclopedia of mathematics, 1997.

[9] INVESTOPEDIA. What does margin mean in investing?, 2017. [Online; accessed 20-November-2017].

[10] L.P., B. Bloomberg website. [Online; accessed 21-March-2018].

[11] MADGE, S., AND BHATT, S. Predicting stock price direction using support vector machines. *Independent Work Report Spring* (2015).

[12] MALKIEL, B. G. Efficient market hypothesis. *The New Palgrave: Finance. Norton, New York* (1989), 127–134.

[13] MENDES, L., GODINHO, P., AND DIAS, J. A forex trading system based on a genetic algorithm. *Journal of Heuristics 18*, 4 (2012), 627–656.

[14] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., ET AL. Scikit-learn: Machine learning in python. *Journal of machine learning research 12*, Oct (2011), 2825–2830.

[15] SETTLEMENTS, B. F. I. Bis papers no 90 - foreign exchange liquidity in the americas.

[16] TEXTBOOK, E. S. Statsoft. *Inc., Tulsa, OK, USA* (2011).

[17] WIKIPEDIA. Autoregressive integrated moving average — Wikipedia, the free encyclopedia, 2017. [Online; accessed 15-November-2017].

[18] YAMARONE, R. The economic indicator handbook: How to evaluate economic trends to maximize profits and minimize losses, 2016.