Mixture of Random Forest Experts in Options Portfolio Management

Rodrigo Manuel Lopes da Silva Lousada
rodrigo.lousada@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

September 2020

Abstract

The financial markets prediction have been studied in many research studies. Although many Machine Learning solutions try to predict the stock market, the derivatives market still has a lot to explore. This Master thesis main goal is to develop an algorithm able to trade financial products. To achieve this main goal it was used a modern approach to trade in the Options Market, through an incremental validation, aiming a portfolio with high profits and controlled risk associated. The architecture includes an approach based on Technical Analysis and Machine Learning to develop a system that goes from collection to process, forecast and trade the options data. It uses a Mixture of Experts (MoE) composed of Random Forests (RaF) to optimize the forecasting performance of the S&P500 index (SPX). The final work overcomes the previous literature with an Annual Rate of Return (ROR) of 100% and Sharpe Ratio of 1.90, on the year of 2015 when a simple Buy&Hold on the SPX only achieves 0.22% of ROR and 0.09 of Sharpe Ratio. On the second half of 2017, when the market was bullish and the options market unstable, it also achieved more than 42%, and a single RaF was able to triplicate the invested money. The main implications of this work provide a solid strategy to accomplish higher profit margins in the options market, especially in periods of high volatility, providing one of the best Return Risk Ratios (RRR) reported, without resorting to Fundamental Analysis data.
Keywords: Machine Learning, Ensemble Learning, Technical Analysis, Options, VIX, Mixture of Experts, Portfolio Management, Random Forests, Derivatives

1. Introduction

Nowadays, financial freedom has become one of the primary goals of the young generations. Although all the warnings about the multiple challenges it presents and the risk of catastrophic loss, new fanatics arrive every year. Trading in the financial markets is hard, and it has so many factors to have in consideration that exceeds human capacity [11]. The Options market is an example of a prosper market which, although famous, it is still underexplored. The field of ML has been researching this question for years, and developed solutions able to do automated trading in very short-term periods. These strategies, aligned with the usage of multiple technical indicators, allowing us to extract more information and achieve a deeper understanding of the financial outcomes [22]. Unlike most of the ML applications, values like precision and accuracy say little about the success of using an algorithm that will manage a trader's portfolio. Despite the forecasting performance of an algorithm, a single wrong prediction can make the trader lose all his money or even create debt. The usage of the VIX index was also referred to as an essential tool for forecasting improvement [23, 25]. Finally, in recent literature, RaF has revealed to be an excellent constituent when ensembled in a combined solution [21, 17, 3, 2]. The concept of a MoE is still growing, having very few applications on the financial markets [5]. The proposed architecture was, to the best of our knowledge, never used before in Options trading. Apart from that, research on Ensemble Learning methods is arising. The MoE allows better performance on large datasets like the ones related to financial markets [5]. Extra motivation will be the exploration of a solution to distribute the price signal through experts, and the development of an approximation to an optimal gating function. Literature indicates the need to explore more in-depth the use of ML to predict the options market behaviour [16].

During the development of the proposed work, this thesis pretends to develop a solution based on Ensemble Learning methods, RaF and Technical Analysis to create a profitable and reliable system. This solution has the main goal of maximizing profit and minimizing risk in an autonomous trading simulation. We pretend to create a reference work for ML applications on the Options Market, as well as improve the literature on MoE. This thesis developed a scalable trading system which exploits ML and Technical Analysis to interpret past data and forecast the direction of the market while simulating trading decisions to generate profit and reduce risk. It compares the usage of a single RaF algorithm, with a MoE [34] composed of a set of RaF experts.

This thesis contributed with for science and industry by generating an impartial and reliable system for choosing the best Call and Put Options to trade in the six months to expiration, developing an innovative and ground-breaking ensemble algorithm which gather predictions from different learners and weights their predictions based on their expertise. Also, we created personalized functions which split the data into different groups to focus each learner on a specific part of the dataset, and developed a Trading System which simulates trading for a given period based on the predictions of a classifier. Finally, we designed a ML system that is available for Finance specialists to give their input without requiring programming knowledge.

2. Background

2.1. Financial Concepts

Financial Markets describe a variety of marketplaces responsible for trading securities or assets such as equities, currencies, derivatives, commodities or bonds. These markets set prices which allow buyers and sellers to trade globally in an open and decentralized system, with the intention of raising capital and relocate risk and liquidity.

The maximization of success prospects, while investing in financial markets, is firmly associated with a robust market analysis. Most of the strategies we will cover try to win better results, generally having as a common benchmark the B&H strategy. The investor has the possibility of adopting one of three positions: long, short or neutral. If the investor considers that the market is hard to predict, he stays out, adopting a "neutral position". If the investor buys a specific asset, hoping for the price to go up, he is taking a "long position". On the other hand, if the investor expects the price to fall, it is possible to have a "short position", selling the asset before actually owning it. The B&H is an approach confident that a certain asset will be more valuable in the future, therefore it is a simple adoption of an extended long position. Those three investment positions are strongly related to the concept of trend, which market analysists use to define the direction of the market. An uptrend is described as a prevailing increase of peaks and throughs, revealing a direction of the stock price going up. A downtrend denotes that those peaks and troughs will have a descending direction. If the market is considered "trendless" because of the horizontality of the peaks and throughs we call it a sideways trend.

2.2. Options

Options are a derivative product from the financial markets. Writing an option designates an investment contract between an option writer and an option buyer (or holder). This exchange gives the buyer the right, but not the obligation, to buy (call options) or sell (put options) an underlying asset in a later date, for a certain fee. This fee will be higher for a more extended period. After the expiration date, the contract will have no value and will no longer exist.

An option contract works as insurance for buyers, as the buyers will pay a small fee to safeguard their money if they take the wrong position in the market. In the worst case scenario, the investor would let the option expire, ending up without the money spent on the fee. Three aspects will determine this fee:

1. **Time to expiration** - as further in time the expiration date is, more time the investor has to profit. Therefore, more valuable is the option. Due to the passage of time, the value of an option tends to drop while we get closer to its expiration date.

2. **Underlying stock price** - both parties will look at the underlying price of a stock and agree on a fixed price for the contract. This price is known as the strike price, which is how much will those shares cost if the option is exercised.

3. **Volatility** - represents the risk for the stock owner, based on the magnitude of fluctuations on an asset's price. An option will be more worthwhile if associated with a volatile asset.

While buying an option, the investor starts the venture with an expense of the fee cost. This cost implies that the investor, to take profit, should expect that the underlying price of the asset will rise or decrease more than the cost associated with this fee. Figure 1 illustrates the possible payoff of an option depending on the intrinsic value of the option.

The investor has three possible approaches: exercising the option, letting the option expire, or selling the valid option to another entity.

In figure 1 it is possible to see an Intrinsic Value line (in blue) and an Option Value line (in red). The Option Value represents the price an option is being exchanged and it is generated by the sum of two values: the Intrinsic Value and the Extrinsic

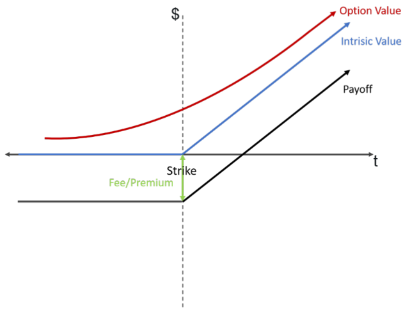

Figure 1: Option payoff range

Value. The Intrinsic Value is the direct value of an asset at a certain moment, which in an option, is how much an investor would profit from exercising the option. The Extrinsic Value of the option is a speculative calculation of how much more the option is probable to value.

The largest and most famous US options exchange market is the CBOE. Most studies on options explored data collected from this exchange market.

2.3. Mixture of Experts

The concept of MoE proposes to use the multiple learners at its disposal in a "divide-and-conquer" strategy. It breaks up a complex task into various smaller and simpler subtasks. Each base learner is considered an expert and is trained for a different subtask. A component called "Gating" generally manages the connection between experts. Image 2 represents the structure of a typical MoE

MoE differentiates from the majority of ensemble methods since in other methods each base learner is trained for the same problem. A MoE, while distributing the problem in smaller problems, benefits the diversity of learners, preventing them from being too much correlated.

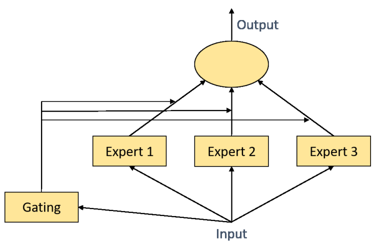

Figure 2: Example of MoE architecture (based on similar figure from Jacobs[12] and Zhou[35])

The primary concern is finding the natural division of the task, and determine the final solution from the set of subsolutions. This distribution allows reducing the processing time due to the reduced data attributed to each expert. Researchers believe that, in order to achieve good performance, it is essential to turn the experts local, focused in only one subtask. A possible solution is to assign each expert to a distribution selected by the gating function instead of the original training data [35].

In a collection of N experts, given an input $\boldsymbol{x}$ and a set of parameters Ψ. The final output of a MoE is a weighted sum of all the local output E_i generated by each ith expert as described in function 1a. The gating function G_i returns the weight of the ith expert.

$$H(\boldsymbol{x}, \Psi) = \sum_{i=1}^{N} G_i(\boldsymbol{x}) E_i(\boldsymbol{x}) \tag{1a}$$

In order to accept the sparsity of experts important for a specific decision, it is possible to save significant computational time by not computing $E_i(x)$ whenever $G_i(x) = 0$ [26].

In the same level of generalization, it is possible to extend this function a little bit more [35], as shown in equation 1b. While θi is the parameter of the ith expert, α is the parameter of the gating function. Also, considering binary classification in which a possible output y is a discrete variable with possible values 0 and 1.

$$H(y|\boldsymbol{x}, \Psi) = \sum_{i=1}^{N} G_i(\boldsymbol{x}, \boldsymbol{\alpha}) \cdot E_i(y|\boldsymbol{x}, \boldsymbol{\theta}_i) \tag{1b}$$

In a simple approach, the gating function is modeled by the $Softmax$ function, as described in functions 2a and 2b

$$G_i(\boldsymbol{x}, \boldsymbol{\alpha}) = Softmax(\boldsymbol{x}, \boldsymbol{\alpha}) \tag{2a}$$

$$Softmax(\boldsymbol{x}, \boldsymbol{\alpha}) = \frac{\exp\left(\boldsymbol{v}_i^\top \boldsymbol{x}\right)}{\sum_{j=1}^{N} \exp\left(\boldsymbol{v}_j^\top \boldsymbol{x}\right)} \tag{2b}$$

The weight vector of the ith expert is represented as v_i, and α contains all the elements in every v.

There is a difference in the application of G_i if applied to the training or the test level. During the test stage, G_i defines how much each expert contributes to the overall forecast. However, while in the training step, this function declares the probability of an instance x appearing in the training set of each expert. The training period of a MoE not only trains each expert based on the distribution specified by G_i, but also seeks the optimal gating function for the overall output.

2.4. Related Work

2.4.1 Works on Single Algorithms

Rosillo et all. [25] used VIX with MACD and RSI, indicators, as inputs to forecast the S&P500. The VIX seemed to improve the algorithm performance on bearish phases, but not as much the bullish ones. Pinto et al. [23] introduced indicators based on VIX and other TI, for their Multi-Objective GA.

Table 1: Overview over literature

Paper	Market Tested	Period of Simulation	Ensemble Method	Algorithms Used	Best Performance	Observations
[3]	DAX	2000 - 2012	Perfomance Weighted	Random Forests (RAF)	Annualised Return: 0.09 Sharpe Ratio: 1.27	Outperformed Models in profitability and accuracy
[20]	Hong Kong Options market	2 years	-	Neural Networks (NN) Support Vector Regressions (SVR)	Average Absolute Error: 0.0620 with SVR and L=12	SVR outperformed NN
[25]	S&P500	2000 - 2011	-	Support Vector Machine (SVM) using VIX, MACD and RSI	-	Overcomes Buy&Hold and SVM w/ VIX
[4]	S&P500 Futures	1997 - 2004	-	Extended Classifier System (XCS) using VIX, Put/Call Ratio and Traders Index	Accuracy: 62.28% Mean Profit: 2456.602	Overcomes profits of Buy&Hold, Mean Reversion and Trend-Following
[2]	5767 publicly listed European Companies	2009 - 2014	AdaBoost	Support Vector Machine (SVM) K-Nearest Neighbour (KNN) Logistic Regression Neural Networks (NN) Random Forest (RAF) Kernel Factory	Random Forest with AUC: 0.9037	All classifieres above 0.5 RF overcomes all others
[27]	S&P500	2010-2014	-	Multi-Objective Evolutionary Algorithms (MOEA)	ROE: 50,24%	Best chromosome with 50,24% of return
[23]	NASDAQ, DAX indexes	2006-2014	-	Multi-Objective Genetic Algorithm (MOGA) using VIX and RSI	Annualised Return: 10%	Outperform Buy&Hold and Sell&Hold
[31]	AMEX ticker: DIA	2001-2003	Mixture of Experts	Genetic Algorithms (GA) Neural Networks (NN)	Accuracy: 73,4%	73.4% correct up/down predictions
[9]	37 Companies on Tehran Stock Exchange	2005-2007	Mixture of Experts	Neural Networks (NN) Adaptive Network-Based Fuzzy Inference System (ANFIS)	Recognition Rate: 86.35%	
[17]	S&P500	1992-2015	Equal-Weighted, Performance-Based, and Rank-Based ensembles	Deep Neural Networks (DNN) (SVM) Gradient-Boosted-Trees (GBT) Random Forests	Annualised Basis: 73%	Equal-weighted ensemble outperformed base learners. RAF was the best base learner
[18]	S&P CNX NIFTY Market Index	2000-2005	-	Support Vector Machines (SVM) Random Forests (RAF)	SVM Hit Ratio: 68.44% RAF Hit Ratio: 67.40%	SVM and RAF outperformed NN and others

Their solution allowed them to avoid multiple falls in the stock market, and achieve more than 10% of annualized return in a period which included the 2008's crash. Kumar and Thenmozhi [18] examined how predictable would be the direction of the S&P CNX NIFTY Market Index. This study concluded that the SVM and the RaF outperformed the remaining models, NN, Logit Model and Discriminant Analysis. The SVM performed slightly better than the RaF due to the ability to minimize the generalization error.

2.4.2 Works on Ensemble Learning

Ballings et al.[2] made a comparison between base classifier models (Artificial NN, logistic regression, SVM and KNN) and ensemble methods (RaF, AdaBoost and kernel factory), for prediction of stock price movements. Ensemble methods proved themselves better than individual learners, having RaF as the best performer. Research using MoE has been used for applications such as risk estimation [34], financial forecasting [5][32][6] and direction variation prediction [31]. Booth et al. [3] developed a performance weighted solution based on RaF which only invested in specific seasons. They stated that although same papers based on reinforcement learning [13], evolutionary bootstrapping [19] and PCA [29] affirmed to beat benchmarks, they also proved to be overfitted showing large drawdowns in profits as well as unnecessary switching behavior. The authors developed an approach based on three layers: expert generalization, expert weighting, and risk management. In the first layer, they generated repeatedly RaF to make predictions on the magnitude of stock prices fluctuations. Secondly, based on each current learner performance, the experts generated an overall output. Finally, the last layer analyzed the decisions made and eradicates the weak signals and liquidates positions which are challenging to predict. Krauss et al. [17] used Deep NN, Gradient-Boosted-Tree, and RaF, to build multiple ensemble algorithms. They developed three ensemble approaches: equal-weighted, performance-based, and rank-based. When considering transaction fees, the first solution achieved annualized returns of 0.73%. This study also confirmed that the best performer was the RaF base learner.

2.4.3 Works on Mixture of Experts Architecture

Since Jacobs and Jordan[12, 15] published the MoE, much research has been emerging. There were proposals for new architectures based on SVM [7], Gaussian Processes [30, 8, 28], and NN [10, 26]. Eigen et al. [10] proposed the concept of multiple MoE, each of them with a specific gating network as a component of a deep model.

Other investigation work converged to changing the expert's configurations. Rasmussen and Ghahramani [24] proposed an infinite number of experts, while Aljundi et al. [1] proposed to add experts sequentially. Jordana, Jacobs [14], and further Yao et al. [33] introduced a hierarchical structure for MoE.

2.5. Works on Derivatives

Liang et al.[20] introduced a non-parametric method of forecasting the option prices in the Hong Kong option market. This research employed NN and SVR to decrease the forecasting error of the parametric methods. The NN and the SVM displayed higher forecast accuracy, having the SVR outperformed the NN. Chen et al. [4] argue that are some disadvantages regarding the use of GA and NN. Both types of algorithms struggle to consistently fit the dynamic financial environment, decreasing their performance on out-of-sample testing. Chen et al. [4] explore an XCS, a variation form LCS, which uses sentiment indicators such as VIX, PCR and Traders Index. The algorithm outputs if it should take a Long or Short position on the market. This approach allowed the authors to overcome the B&H, Mean Reversion and Trend-Following strategies.

3. Methodology

3.1. Decisions based on literature

Reviewing the literature, it seems not to exist a consensus over the benefits of using GA and NN. Although both algorithms displayed great results, they generally have large drawdowns on out-of-sample testing [4]. Although they appear to have great results on stocks forecasting, they might not be the best algorithms to rely on when applying for money, specially on a derivative product since its price changes based on the reaction to the index price. On the other hand, solutions based on RaF and SVM appear to show significant results[3, 17, 2, 21], with RaF showing off as the best base learner when ensembled [2] which indicates us the benefits of using two layers of ensembles, a MoE with RaF as base learners. In the literature review, it was also clear that ensemble methods, when applied, always overperformed the individual learners' achievements [17].

Regarding the Market Analysis, Technical Indicators (TIs) are reported to be used in every research project, therefore we assume it has a fundamental impact on the predictor's results. To have a complete analysis of the market situation, all TIs' types should be generated with different short and long periods. Another remark was the impact of the VIX to perform on bearish signals [25, 23], as this index gives a useful estimation of the volatility for the following 30 days.

Knowing what was mentioned before, we decided to develop a Mixture of Random Forest Experts (MoRFe) to trade SPX options. The market analysis will use Technical Analysis by applying TIs and using the VIX. We chose not to use any sort of Fundamental Analysis as the necessary data would be harder to collect in a real case scenario. This thesis also intends to confirm the veracity of the technicians' belief that we only need to look at the price to get the necessary information to trade on it.

3.2. Architecure

The architecture of our portfolio manager consists on a system that collects data from the CBOE options market and simulates trading SPX options to generate the maximum profit possible with less risk associated. It used data extracted from two notorious databases, Delta Neutral and WRDS, and is based on three layers: the Data Layer, the Logic Layer, and the Presentation Layer. Figure 3 illustrates an overview of the overall architecture.

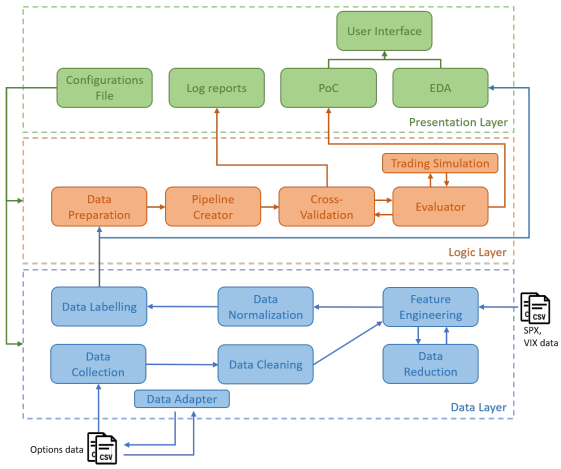

Figure 3: General View of the System

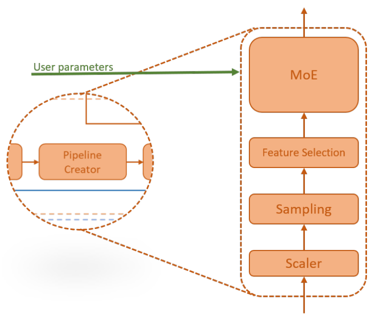

Figure 4: Pipeline Creator close-up

The bottom layer is responsible for managing the original data and preparing it for the Logic Layer, it does it while using the parameters defined by the user in the Presentation Layer. The second layer is responsible for the predictive process where it will train our model, predict the direction of the price and simulate the investments in the market. Finally, the presentation layer contains the UI for the user, with a set of parameters that can be twitched, and all the relevant information displayed, such as the success metrics and the characteristics of the collected data.

The architecture represented in figure 3 is a bottom-up approach, which was developed accordingly. The first step was to collect past data from reliable sources. Data relative to the SPX and VIX, have high demand in the market, therefore it was possible to acquire them for free. However, options' trustworthy data is still short in supply due to the complexity of this instrument. Therefore we had to resort to paid database services. After gathering the data, we were ready to develop our solution. First, we created the Data Layer, starting by collecting the data from the different files, and joining all in one DataFrame, ordering it in records and features in the Data Collection module. As the two sources where we gathered the options' data had different schemas (name of features, and categories' formats), we also developed a Data Adapter which converts a third party data into the same schema of our solution.

The Data Cleaning module receives the data and fixes problems of missing values or inaccurate data. After this, we apply the steps of Features Engineering and Data Reduction. The Feature Engineering process is responsible for generating new features with relevant information which may help in getting better predictions from the algorithm. The Data Reduction module applies a set of rules to filter the data into a chosen set of options. It uses rules related to the type of contract (put or call), liquidity of the option, and average price some months before expiration. The data Feature Engineering module is divided into two steps, one before, and another after the data reduction process. The first step generate features that need to be calculated before the Data Reduction module, such as the price and time to expiration of the option that will be used in that process, or the PCR in case we filter only for one type of contract. The second step, calculates the remaining features in a shorter dataset, avoiding unnecessary calculations for the excluded options. This second step is when the Feature Engineering module will read the SPX and VIX data to calculated Technical Indicators, Intrinsic and Extrinsic Value, if the option is ITM or OTM, as well as, the percentage change in price.

Most of the generated features are unlimited continuous values, which may be difficult to predict for our algorithm if unseen values appear. Therefore, the Data Normalization module grades most of the TI based on a set of rules, and normalizes other features in a fixed range between 0 and 1. Finally, the data is sent to the Data Labelling module which determines the labels that our classification algorithm will try to predict. In this case, the values will be -1 for selling, 0 for holding or staying neutral, and 1 for buying.

In the Logic Layer, the Data Preparation module will read the data from the Data Layer and arrange it as input of a ML algorithm, casting features types, enconding categorical variables, and splitting the data into train and test sets, following the specific rules for Time Series problems. After this, the Pipeline Creator will be responsible for assembling in a pipeline all the elements which will transform the received data into a set of predictions. These elements include processes like standardizing the data into a normal distribution, sampling the data to balance the number of occurrences of each label, ranking the features importance and select only the most relevant, and finally creating the ML algorithm. The ML algorithm will vary from a single RaF to a MoE with different gating functions, in order to compare the performance of the further explained classifiers.

Henceforth, the CV module splits the training data into multiple pairs for train and validation so we can determine the best hyperparameters for the Pipeline. Each set of training and validation will be sent to the Evaluator which will try to predict the defined label and send it's prediction to the Trading Simulator. The Trading Simulator uses those suggestions to test its trading performance. The Trading Simulator will send its results to the Evaluator which send the Classification and Trading performance results back to the CV module. This CV module generates a report in the form of a log, for further analysis in the Presentation Layer. After determining the best hyperparameters the same process is repeated with the Evaluator saving the model, predictions and results, and sends it to the PoC module.

In the Presentation layer, the User has the Configuration File to determine and adjust some of the hyperparameters, rules, and ranges used in the full solution. Apart from having the Log reports mentioned before, it contains a UI based on Jupyter Notebooks that contains a set of visualizations for the EDA and PoC of our solution. This notebooks allow the user and the developer to further investigate the current results, interpret the data, and think on alternative solutions.

In the end, we test the explained architecture with three Calssification algorithms: a single RaF, a MoRFe using equal weight on the gating function, and a MoRFe using a function called BinSplit as gating function. The BinSplit function discretized its continuous values into a specified number of bin, and assigns each bin to one expert.

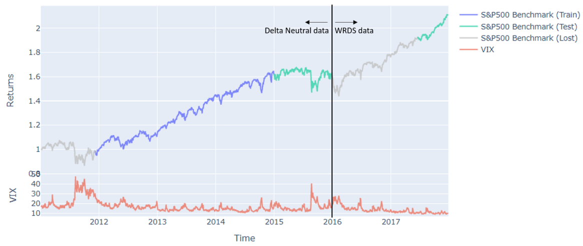

Figure 5: SPX and VIX in the complete time frame of the available data

3.3. Methodologies of Work Evaluation

The stated results were tested in a laboratory environment where the available data was split into training, validation and test periods. An iterative approach was used to maximize the performance of the algorithm by experimenting with multiple hyperparameter combinations on the validation sets. This validation follows the CV technique when applied to Time Series forecasting. The different strategies were tested based on the same evaluation metrics, which analysis their forecasting capacity and trading performance, compared to the literature [3, 18, 2] and each other. Results for each algorithm implementation will be compared with each other to determine the best system. These results are the approximation of an optimized configuration based on a finite number of combinations for the hyperparameters. Due to time complexity, we can't assure that the combination used for the classifier will be the best for this problem but at least the best for a set of reasonably considered combinations. After achieving the best results for our algorithm, its profits, risk and accuracy will be compared against strategies in the next section.

3.4. Starting Point and Benchmarking

Regarding benchmarks for the testing time periods, our first benchmark will be to adopt a B&H strategy in the SPX. This strategy is the simplest and most common approach when people invest in the market. The SPX represents the US economy, which presents a long term bullish trend. If the economy is healthy, the price should be in an uptrend. It is a strategy that takes very low effort, and in some cases can represent a better return than the retirement pension.

In figure 5, it is clear that the two test periods are very different, which help us explore the algorithm's performance in multiple situations. The first period is from 31th of December of 2014 to the 30th of December of 2015. The second period is from the 20th of June of 2017 to the 28th of December of 2017. The first period corresponds to the 30% of the test of the Delta Neutral data, while the second one is the resulting data from the WRDS data after applying it to the Data Layer. For simplicity reasons, we will refer to them as 2015 and 2017, respectively. In terms of trend, the year 2017 is in a clear uptrend, ending with ROR around 9.5%, and 2.84 of Sharpe Ratio. On the other end, 2015 is sideways, ending with ROR around 0.22%, and a 0.09 Sharpe Ratio. While in 2017 the highest value of the VIX was 15.55, in 2015, the minimum value is 11.95, and the highest is 40.74. The results for both periods are stated in table 2 as Benchmark.

Figure 5 illustrates the SPX and the VIX over all the data we had available. The periods were the SPX line is grey, represent the time where the data was "lost" by the Data Layer when creating the TI and applying the Data Reduction process. This lost happens since, to calculate a TI with period X. To test a real case scenario, and to avoid compatibility issues, we consider the WRDS data as a separated dataset only for testing. The purple line represents the period used for training the data, and the two green periods are the test periods. As we can see, the training period diverges from the two testing periods. While the training period presents a moderated VIX and a bullish price with the peaks and throughs successively rising, the period of 2015 is a sideways period with high levels of VIX. At the same time, the year of 2017 represents very low values for the VIX, and, mostly, a strong bullish trend on price almost without any loss days. The year of 2017 was justifiably uncommon with the market rising too quickly. Although it is not represented in figure 5, from January to March 2018, this price had a correction, with a depreciation around 10%. A correction is mentioned when the price of an asset drops, from its last peak, 10% or more. This tends to happen when the asset is overvalued at that time, and a drop in demand occurs.

4. Thesis Results

4.1. Results Overview

This thesis has two case studies which are represented by two different time windows: 2015, and the second half of 2017. Each case study tests the trading Results if a B&H strategy was applied to the SPX (Benchmark), the trading results if the algorithm used had 100% accuracy on the selected data (Best Case), the trading results if the algorithm used had the inverse decision of the Best Case (Worst Case), the B&H and S&H to the selected set of SPX options, and the application of the three algorithms. The SPX prices are used as the main benchmark for comparison on how could we benefit from just using a B&H strategy in the market. The Best Case, Worst Case, B&H and S&H give us a glance of the quality of the data selection, and will give us more context to understand the trading performance of our algorithms. After running all simulations, table 2 joins all the metrics for these approaches. It is important to remember that the Best and Wort Cases represent the best and worst cases in term of predictions, which doesn't mean that it applies in terms of return or risk.

In all cases, the precision value for each algorithm is higher than the recall and accuracy. This means that the number of times a positive identification was actually correct is higher than the number of times an actual positive was identified correctly. Higher Precision is good when we want to enter a position, while higher Recall is good when we want to exit a position. As our trading algorithm has Stop Loss and Take Profit rules to exit a position, in the predictive process, it will be a priority to have better Precision than Recall. The F-measure represents the mean of both metrics (Precision and Recall), and we can see a significant improvement of our values from choosing to always use one of the positions at all times. It is possible to see that the MoE algorithms tend to have solid returns and better risk management. Therefore the RRR shows solid improvements.

4.2. Case Studies
4.3. Testing on 2015 data

This test period corresponds to the range between the 31 of December of 2014 and the 30 of December of 2015. This period is sideways, ending with a low ROR of around 0.22%, and Sharpe Ratio of 0.09, when applying a B&H on the index. The VIX indicates values between 11.95 and 40.75, which represent very high volatility. High volatility in both directions will result in very volatile options prices, as most traders will opt to stay out of the market or to assure a maximum loss on their investment by buying an option. In terms of selected data, if it was possible to predict all the daily directions correctly, our algorithm could profit an ROR of 61 times the money invested. On the other hand, if would always take the wrong decision, it would only lose seven times the money invested. This difference happens from the account of the Trading Simulator, which uses Stoploss and Take Profit strategies to stop wrong positions and assure profits in good ones. This 61 to 7 ratio give us great confidence to test our algorithm.

In table 2, the RaF achieved an accuracy of 49% in 2015, which exceeds both the B&H (24%) and S&H (38%). However its results do not get close to Versace et al. [31] (73.4%). This may happen because of the increased volatility of this period, which will comprehensively be harder to predict. It achieved 91% of return. This result already overcomes all the ROR reported in the literature. Although in table 2 we conclude that the B&H strategy had better ROR, we can see in figure 6 that the S&H and B&H suffered a peak over their returns around March. Apart from that event, the B&H was losing money slowly, while the S&H strategy was getting profit most of the time, except when it had sudden losses. That event happens as both strategies assume a position on an options that rose from $1.5\$$ to $40\$$ in a day. Looking to figures 6 and 5, it is possible to infer that its biggest loss occured during the fast recovery of the SPX during the month of October, which the algorithm failed to predict. Comparing to our Benchmark (B&H on the SPX), it is possible to analyse how much leverage we can take from using options instead of only buying the index. In terms of risk, the previous approaches presented Sharpe Ratio levels below 1, while the RaF gets 1.37, which is already a good indicator of risk management from this algorithm. This indicator also surpasses the 1.27 reported in the literature.

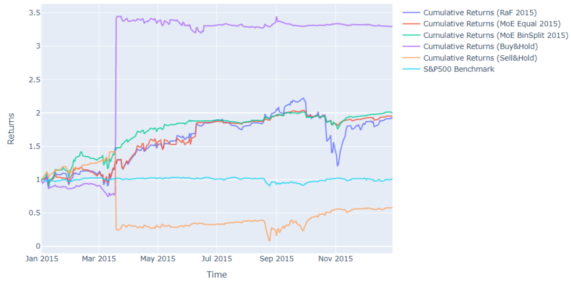

Figure 6: Returns of all the strategies during the 2015 testing period

For comparison purposes, we chose to maintain the configuration of the RaF for the MoRFe algorithms, as it would allow us to see, for the same hyperparameters, the progress of results following the improved architecture. In terms of predictive performance, the results of the MoE with the Equal Weight gating function were pretty close to the single RaF, which was expected. The whole dataset was sent to each expert, giving them exactly the same expertise. The trading results showed some improvements from the single RaF. The ROR improved from 91% to 95%, while the Sharpe Ratio went from 1.37 to 1.69. If we compare the performances of the RaF and the MoE on figure 6 we can observe more stable returns from this algorithm. While the single RaF had a higher maximum ROR than the MoE around September, if we examine the months of February and October, the major losses have decreased to lower values. A higher number of Stop Loss orders seem to be activated during the month of June, which may explain the slower growth in ROR during the following month. The number of losing options from July to November reduced from using only one base learner. Thus, the major benefit of using the MoE (Equal) over a single RaF seems to be the improvement in risk

Table 2: Results for all case scenarios

		Precision	Recall	Accuracy	F Measure	ROR	ROI	Annualised ROI	Mean Daily Return	Sharpe Ratio	Risk Return Ratio	Max Drawdown	Sortino Ratio
Train	RaF	77.06%	75.19%	75.19%	75.80%								
	MoE (Equal)	77.68%	75.74%	75.74%	76.38%								
	MoE (Bin Split)	77.22%	75.35%	75.35%	75.96%								
2015	Benchmark					0.22%	0.22%	0.22%	0.01%	0.0926	0.58%	-12.35%	0.1311
	Best Case	100.00%	100.00%	100.00%	100.00%	6170.92%	6170.92%	6192.11%	18.56%	1.0245	6.45%	-13.16%	207.9627
	Worst Case	0.00%	0.00%	0.00%	0.00%	-702.03%	-702.03%	-704.44%	-20.98%	-0.9365	-5.90%	-3342.85%	-0.9764
	S&H	30.12%	54.88%	54.88%	38.90%	-42.07%	-42.07%	-42.21%	-0.84%	0.9194	5.79%	-94.26%	1.6838
	B&H	17.06%	41.30%	41.30%	24.15%	229.61%	229.61%	230.40%	1.24%	0.9424	5.94%	-28.92%	16.0484
	RaF	**50.28%**	**49.26%**	**49.26%**	**49.63%**	**91.49%**	**91.49%**	**91.80%**	**0.35%**	**1.3706**	**8.63%**	**-45.94%**	**1.9980**
	MoE (Equal)	**50.26%**	**49.10%**	**49.10%**	**49.52%**	**95.11%**	**95.11%**	**95.44%**	**0.26%**	**1.6904**	**10.65%**	**-19.28%**	**2.8591**
	MoE (Bin Split)	**50.19%**	**48.34%**	**48.34%**	**49.01%**	**100.20%**	**100.20%**	**100.55%**	**0.28%**	**1.8983**	**11.96%**	**-19.76%**	**2.9298**
2017	Benchmark					9.54%	9.54%	18.15%	0.08%	2.8423	17.90%	-2.11%	4.4650
	Best Case	100.00%	100.00%	100.00%	100.00%	87.02%	87.02%	165.54%	0.45%	2.5023	15.76%	-1.22%	85.4127
	Worst Case	0.00%	0.00%	0.00%	0.00%	9.02%	9.02%	17.16%	0.05%	0.6547	4.12%	-38.23%	1.3437
	S&H	38.41%	61.98%	61.98%	47.43%	16.42%	16.42%	31.24%	0.09%	0.9128	5.75%	-53.04%	2.4444
	B&H	11.75%	34.27%	34.27%	17.49%	99.17%	99.17%	188.65%	0.52%	1.9946	12.56%	-18.59%	6.3167
	RaF	**48.34%**	**43.97%**	**43.97%**	**45.74%**	**212.37%**	**212.37%**	**404.00%**	**1.11%**	**2.4665**	**15.54%**	**-12.22%**	**15.2212**
	MoE (Equal)	**48.12%**	**43.97%**	**43.97%**	**45.69%**	**52.92%**	**52.92%**	**100.68%**	**0.28%**	**1.4011**	**8.83%**	**-18.11%**	**3.2661**
	MoE (Bin Split)	**49.93%**	**47.09%**	**47.09%**	**47.57%**	**42.51%**	**42.51%**	**80.86%**	**0.22%**	**-1.6387**	**-10.32%**	**-110.08%**	**-0.1700**

management. It increased the Sharpe Ratio, from 1.37 to 1.69, the Sortino Ratio, from 1.99 to 2.86, and the RRR from 8.63% to 10.65%.

Finally, we tested the MoE with the Bin Split function, which enabled the creation of experts with different subsets of data in their knowledge base. This split used the most relevant feature, determined by the SelectKBest, to generate equally ranged bins. Using this gating function drop 1% on the accuracy. Even though it had slightly worse accuracy, we understand that using a feature to break the data into different learners may have a beneficial effect as the algorithm shows faster comebacks from its drawdowns. A possible improvement for this function would be to reference a biased feature such as the VIX or a TI. Nevertheless, that capability might limit the usage of some preprocessing steps such as sampling or PCA. If we explore figure 7 and the results in table 2, it is possible to verify an improvement in the trading performance by using this gating function. The ROR hits the 100% of annual ROR, and registers the highest Sharpe Ratio, increasing from 1.69 to 1.90.

Each of the three algorithms performed very well during this period of high volatility. The B&H strategy only works if we catch these big swings in price, that should compensate for the remaining losses, while the S&H works if we avoid those same swings. From the single RaF we understand improvement in risk management, where the months of September to November were less volatile. On the MoRFe gating functions, there is an improvement in the capability of recovering from losses, since we see higher profits after a Drawdown. Three events when this ability was evident, where the recoveries from February, March and November.

Although less accurate, the MoRFe presented a substantial improvement in profit/risk ratio, as we can see by examining the RRR values. This algorithm also beats the reported metrics in the literature, in a period were investing in the SPX directly would result in hardly any profit. Most of the return was achieved until the month of June where it diminished the number of positions, but that fact can also be related to the decisions made on the architecture, where he amount of money invested is lower, the more we have allocated in other options.

4.4. Testing on the second half of 2017 data

The second test period is from the 20 of July of 2017 to the 28 of December of 2017. It has an extremely bullish trend with low volatility, and seems like one of the moments when investors would be more confident to invest with a B&H on the index. Nonetheless, this period is probably overvalued, as it is rising too fast with very few loss days. Even though its price could continue rising, it would eventually, sooner or later, suffer a correction. In terms of options selection, our Trading Algorithm was able to make profit even on the Worst Case. Although the B&H has an exemplary Sharpe Ratio of 2.84, it presents a low ROR of 9.5% in six months, compared to the results reported by our algorithms in the previously studied period. The VIX indicates values between 9.36 and 15.55, which is lower than the last years. This situation may symbolise that the prices are too unstable but changing inside the same range. This variation allows us to make a profit in either direction. Therefore, the Take Profit order makes the difference so we can exit at the right time.

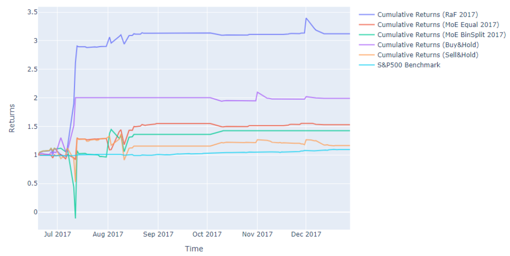

Figure 7: Returns of all the strategies during the 2017 testing period

The single RaF performed exceptionally well for this period. Despite the accuracy values dropped to 43.97%, the algorithm achieved an incredible 212.37% of ROR and 2.4665 of Sharpe Ratio. The relation between those metrics is represented in the 15.54% of RRR, which almost equals the result of the Best Case. The reason for not winning the Best Case in RRR is the Max Drawdown of -12.22%. We can also comprehend that the reason for this high profit relied on the interval between 6 and 13 of July when the SPX price was starting a bullish trend. Nevertheless, the algorithm continued to make a profit, rising its profit from 190% to the final 212%. Once more, we also prove the leverage of trading options, from a simple B&H on the index.

The first application of the MoE had slowly changed the prediction results, similarly to the 2015 period. However, in this case, instead of improving results, the MoE didn't profit from the 6 to the 12 of July. It has worse performance than the S&H before July, and a close valorization until mid-August, when it increased the number of winning positions significantly achieving around 52.92% of ROR, in half the time of the 2015 period.

When using the BinSplit gating function, the results improved in accuracy up to 47.09%, what shows us that the divide-and-conquer strategy may improve our performance. However, the predictions didn't have a better result in terms of trading performance, as the previous algorithms. This may be normal as the hyperparameters were tunned for the single RaF during the CV period.

Notwithstanding, the algorithm achieved 42.51% of return in six months. However, the Sharpe Ratio presents a negative value of -1.6387 with the max Drawdown of -110.08%. This means that at a specific point, the algorithm was not only losing the money invested but also in debt. This loss only occurred in one day of July, and bounces back right in the next day. Even so, this loss represents a significant threat for a portfolio, affecting the RRR with -10.32%. During this period when the single RaF increased dramatically, this algorithm registered a spike downwards, as it invested in the same options in the opposite direction.

In conclusion, the three algorithms had very different results, as one single position could change drastically the profit generated. One of the main events for this difference is the period from 6 to the 13 of July, where the right prediction could completely change the outcome. For the same reason, a lot of Take Profit and Stop Loss orders were activated in response to this high swings in price, justifying why all strategies had much flatter ROR differences after September. One common issue is the few amount of options considered to be traded for this period, leaving some parts of the period with the algorithm in a neutral position in any option.

An excellent indicator was that our Trading Simulator was able to have profit, either adopting a B&H, or a S&H strategy. The single RaF was the top performer as it got one spike on the right direction. For this reason, there was some Take Profit and Stop Loss orders activated in the early beginning. On the other hand, the MoE with the Bin Split caught one of those. Nevertheless, this system was more profitable in this period, as it was applied to a shorter time span.

4.5. Results Conclusions

The solution in hands achieved accuracy scores between 40% and 50% in both test periods and overcame the literature results in terms of trading performance. In a sideways period of high volatility, this work achieved an Annual ROR around 100% and Sharpe Ratio 1.89. On the other hand, on a period of very low volatility, the solution was able to exploit the unstable prices and achieve profit in all strategies, having a single RaF with 212% profit in only six months. In this period, the algorithm achieved profit even when the market was uncertain due to its overvalued price.

5. Conclusions

This work delivers a complete ML trading solution, from the collection of data to the trading simulation, based on forecasting the price directions. It was developed to become an Open Source project and to establish a base for a future thesis on ML applied to Finance. Although most theses on this topic end up not sharing the code online, we firmly believe that sharing our project would enable others to go further on their research, without the necessity of coding the whole system from scratch. Since this thesis is related to two fields, ML and Finance, the input from specialists in both areas would be welcome. Therefore, our architecture delivers a straightforward interface which allows people with a finance background, to collaborate without a Computer Science degree or any code skills. We also prove that the application of this methodology accomplishes better predictions than its base learners, mostly on topics that simple ML metrics may not mirror. Lastly, this thesis applies an innovative strategy and completed the primary goal: achieving high profits with controlled risk, reflected on the stated RRR values.

We can conclude that the thesis goals were attained. Besides using options to leverage profits which beat the SPX exceedingly, our trading simulation overcame the benchmarks stated in the literature in terms of profit and risk. The developed architecture of MoRFe proved to determine useful insights which were reflected on the metrics as accuracy or ROR. Although the ML metrics were below the literature, after seeing the contrast between the training set and the two test sets, we feel that the results were pretty satisfactory.

The Data Reduction process surpassed our expectations as it created a reliable foundation to trade in the selected option, even with a simple B&H strategy. The Trading system proved to exploit the ML predictions properly, allowing to mitigate the risk and achieve higher profits. The system amazed through its performance on high volatility periods, assuring stable and high returns. Regarding risk, it also mitigated the effect of volatility, ensuring low risk. On the other hand, in times of weak volatility, options with high swings in prices should be avoided as it represents a very high risk on our portfolio. The low volatility values should be further studied and understood to create a more mindful rule which would allow our system to stay out of the market during these moments. Overall, we believe this work can encourage others to continue to explore the Options Market and possible MoE applications.

5.1. Limitations

This work is not without limitation. The gathered data is from 2011 to 2017, and it would be interesting to analyse real-time data. Also, similarly to the literature, it was considered a mid-point price between Ask and Bid, instead of creating two isolated Classifiers. In case of applying this strategy on a non-commission-free broker, the commission rate should also be considered.

5.2. Future Work

With such results, we can envision more iterations to make the project even more interesting, possible variations from our decision, and some challenges to overcome. As challenges to overcome, we believe that increasing the amount of options data will bring better results, improving the grading rules for each TI, and optimizing the cash allocation for each investment would help improve the solution. One could vary from our decision, and experiment using Regression instead of Classification, using Genetic Algorithm for tunning Hyperparameters, or adopting a S&H strategy with the classifier only detecting when to switch for a B&H strategy. Finally, as incremental work on our solution it would be possible to increase in three fronts. For the Data Layer, it would be interesting to automate the data extraction, or create a labeling function that would also consider the asset price. On the Logic Layer, we would focus future work on generating new gating functions, design a scoring function that would balance Accuracy, ROR and RRR, or implement Options Spreads. Also, in the Presentation Layer, the current work could be displayed in a customizable Dasboard available online, with an available API for external forecasting requests, and a friendly customizable UI for financial user may tweak the hyperparmeters.

References

[1] R. Aljundi, P. Chakravarty, and T. Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 7120–7129, 2017.

[2] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056, 2015.

[3] A. Booth, E. Gerding, and F. Mcgroarty. Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*, 41(8):3651–3661, 2014.

[4] A.-P. Chen and Y.-H. Chang. Using extended classifier system to forecast s&p futures based on contrary sentiment indicators. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2084–2090. IEEE, 2005.

[5] Y. M. Cheung, W. M. Leung, and L. Xu. Application of mixture of experts model to financial time series forecasting. In *Proc. Int. Conf. Neural Netw. Signal Process*, pages 1–4, 1995.

[6] A. L. Coelho, C. Lima, and F. J. Von Zuben. Hybrid genetic training of gated mixtures of experts for nonlinear time series forecasting. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 5, pages 4625–4630. IEEE, 2003.

[7] R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of svms for very large scale problems. In *Advances in Neural Information Processing Systems*, pages 633–640, 2002.

[8] M. P. Deisenroth and J. W. Ng. Distributed gaussian processes. *arXiv preprint arXiv:1502.02843*, 2015.

[9] R. Ebrahimpour, H. Nikoo, S. Masoudnia, M. R. Yousefi, and M. S. Ghaemi. Mixture of mlp-experts for trend forecasting of time series: A case study of the tehran stock exchange. *International Journal of Forecasting*, 27(3):804–816, 2011.

[10] D. Eigen, M. Ranzato, and I. Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[11] A. Gabriela ğiĠan. The efficient market hypothesis: Review of specialized literature and empirical research. *Procedia Economics and Finance*, 32:442–449, 2015.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[13] O. Jangmin, J. Lee, J. W. Lee, and B.-T. Zhang. Adaptive stock trading with dynamic asset allocation using reinforcement learning. *Information Sciences*, 176(15):2121–2147, 2006.

[14] M. I. Jordan and R. A. Jacobs. Hierarchies of adaptive experts. In *Advances in neural information processing systems*, pages 985–992, 1992.

[15] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[16] G. Kou, X. Chao, Y. Peng, F. E. Alsaadi, and E. Herrera-Viedma. Machine learning methods for systemic risk analysis in financial sectors. *Technological and Economic Development of Economy*, 25(5):716–742, 2019.

[17] C. Krauss, X. A. Do, and N. Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259(2):689–702, 2017.

[18] M. Kumar and M. Thenmozhi. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[19] B. LeBaron. An evolutionary bootstrap method for selecting dynamic trading strategies. In *Decision Technologies for Computational Finance*, pages 141–160. Springer, 1998.

[20] X. Liang, H. Zhang, J. Xiao, and Y. Chen. Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing*, 72(13-15):3055–3065, 2009.

[21] M. Maragoudakis and D. Serpanos. Towards stock market data mining using enriched random forests from textual resources and technical indicators. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 278–286. Springer, 2010.

[22] J. J. Murphy. *Technical Analysis of the Futures Markets: A Comprehensive Guide to Trading Methods and Applications, New York Institute of Finance*. Prentice-Hall, 1999.

[23] J. M. Pinto, R. F. Neves, and N. Horta. Boosting trading strategies performance using vix indicator together with a dual-objective evolutionary computation optimizer. *Expert Systems with Applications*, 42(19):6699–6716, 2015.

[24] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pages 881–888, 2002.

[25] R. Rosillo, J. Giner, and D. de la Fuente. The effectiveness of the combined use of vix and support vector machines on the prediction of s&p 500. *Neural Computing and Applications*, 25(2):321–332, 2014.

[26] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[27] A. Silva, R. Neves, and N. Horta. A hybrid approach to portfolio composition based on fundamental and technical indicators. *Expert Systems with Applications*, 42(4):2036–2048, 2015.

[28] L. Theis and M. Bethge. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems*, pages 1927–1935, 2015.

[29] N. Towers and A. N. Burgess. Implementing trading strategies for forecasting models. In *Y. Abu-Mostafa et al.(eds.) Computational finance. Proceedings of the Sixth International Conference on Computational Finance*, pages 313–325, 1999.

[30] V. Tresp. Mixtures of gaussian processes. In *Advances in neural information processing systems*, pages 654–660, 2001.

[31] M. Versace, R. Bhatt, O. Hinds, and M. Shiffer. Predicting the exchange traded fund dia with a combination of genetic algorithms and neural networks. *Expert systems with applications*, 27(3):417–425, 2004.

[32] A. S. Weigend and S. Shi. Predicting daily probability distributions of s&p500 returns. *Journal of Forecasting*, 19(4):375–392, 2000.

[33] B. Yao, D. Walther, D. Beck, and L. Fei-Fei. Hierarchical mixture of classification experts uncovers interactions between brain regions. In *Advances in Neural Information Processing Systems*, pages 2178–2186, 2009.

[34] M. S. Yumlu, F. S. Gurgen, and N. Okay. Financial time series prediction using mixture of experts. In *International Symposium on Computer and Information Sciences*, pages 553–560. Springer, 2003.

[35] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.