UNIVERSIDAD Đ SEVILLA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Doctoral Thesis

Forecasting and optimization of stock trades

Gerardo Alfonso Pérez

Thesis submitted for the degree of Doctor of Philosophy in the University of Seville

---

Supervised by:

Daniel Rodríguez Ramírez

To my father, Antonio, and my mother, Pilar

Acknowledgements

I would like to thank my thesis supervisor professor Daniel R. Ramirez for the help during the long process of studying for a PhD and for his support and encouragement.

I would also like to thank professor Teodoro Alamo and Daniel Carnerero.

Finally, but very importantly, I would like to thank my wife, Wandy, and my kids, Leo and Alyssa.

Shanghai, 2021.

Abstract

The stock market is a complex and challenging field of research that has attracted researchers from several fields, such as for instance engineering. This dissertation approaches stock market analysis from an engineering point of view. It is empirically shown that techniques, such as neural networks, can be applied in many stock markets, as a tool creating reasonably accurate stock forecasts. This approach was also analyzed in the context of narrow markets. In this dissertation a broad definition of narrow market was followed, encompassing not only stock markets with a small trading volumes, which can potentially distort stock prices, but also markets that while having large daily trading volumes have some other features, such as a relatively large proportion of retail investors compared to institutional investors, that might result on price distortions. It is shown that neural networks can be applied, for forecasting purposes, even in narrow markets. However, some practical issues, such as stale prices, should be taken into account. The topic of technical indicator selection is also discussed in this dissertation. There is an ever increasing amount of technical indicators that are used in an attempt to discern future stock prices trends. Some of those indicators can generate contradicting signals. Therefore, it is important to choose the right combination of technical indicators when taking stock investment decisions. In this dissertation it is presented a new combinatorial algorithm for stock selection. It is shown, using this algorithm, that predictions are more accurate than using all the technical indicators together. It should also be noticed that using all the possible combinations it is not feasible given the enormous amount of potential combinations. Therefore, it is of clear importance to have algorithms that can generate adequate combinations of those technical indicators.

The adaptation and application to stock forecasting of a forecasting technique based on local data is also shown in this thesis. It has been shown that this technique generates forecasts that are better than some commonly used benchmarks. At the core of this approach there is the assumption that stock prices then to follow, at least to some degree, historical patterns. Besides accuracy, an advantage of this technique is that it generates forecasts relatively fast, not requiring a time consuming training phase. Having a better understanding of the likely losses of the trade can help the investor to have a more complete investment decision process. Thus, it is also important, particularly from a risk management point of view, to have techniques that generate probability distributions for the predictions, that can be used to obtain intervals that are guaranteed to contain the future real prices with a prescribed probability. In this dissertation a probabilistic price interval estimation strategy has been adapted and applied to the problem of finding

k-step ahead price intervals, getting better results than those obtained with well known techniques.

Having good forecasting techniques is only a part of the investment process. After a buy or sell decision has been made it is important to carry out that trade efficiently in what it is commonly referred as “best execution”. There are multiple factors to take into account after the purchase (or sell) of a stock is decided, for instance in which period of the day (or over how many days) to carry out the trade. This is also related to the idea of avoiding distorting the market with the trade, i.e., a large order might unintentionally distort market prices. In this dissertation it is presented an approach that generates better results than frequently used benchmarks, such as for instance the Time Weighted Price (TWAP) or the Volume Weighted Price (VWAP). The approach is based on the concept of receding horizon optimization, well known in the predictive control community, but not used in the optimization of the execution of large trade orders. The technique can accommodate both market and limit orders and has shown a great potential economic impact in the execution of such large orders.

Resumen

El mercado de valores es un campo de investigación complejo y desafiante que ha atraído a investigadores de varios campos, como por ejemplo la ingeniería. Esta disertación aborda el análisis bursátil desde el punto de vista de la ingeniería. Se mostrará empíricamente que técnicas, como las redes neuronales, se pueden aplicar en muchos mercados bursátiles, como una herramienta que crea estimaciones del precio de las acciones razonablemente precisas. Este enfoque también se analizó en el contexto de un mercado estrecho. En esta disertación se siguió una definición amplia de mercado estrecho, abarcando no solo mercado de valores con pequeños volúmenes de negociación, que potencialmente pueden distorsionar los precios de las acciones, sino también los mercados que, si bien tienen grandes volúmenes de negociación diarios, tienen algunas otras características, como una proporción relativamente grande de inversores minoristas en comparación con los inversores institucionales, que podrían dar lugar a distorsiones en los precios. Se mostrará que redes neuronales se puede aplicar incluso en mercados estrechos. Sin embargo, deben tenerse en cuenta algunas cuestiones prácticas, como los precios obsoletos.

El tema de la selección de indicadores técnicos también se discute en esta disertación. Hay una cantidad cada vez mayor de indicadores técnicos que se utilizan en un intento de discernir las tendencias futuras de los precios de las acciones. Algunos de esos indicadores pueden generar señales contradictorias. Por lo tanto, es importante elegir la combinación correcta de indicadores técnicos al tomar decisiones bursátiles. En esta disertación se presenta un nuevo algoritmo combinatorio para la selección de indicadores técnicos. Se mostrara que las predicciones, usando este algoritmo, son más precisas que las estimaciones generadas usando todos los indicadores técnicos juntos. También debe tenerse en cuenta que usar todas las combinaciones posibles no es factible dada la enorme cantidad de combinaciones posibles. Por lo tanto, es de clara importancia tener algoritmos que puedan generar combinaciones adecuadas de esos indicadores técnicos.

También se presenta en esta disertación una aplicación de una técnica estadística basada en datos locales para fines de estimación del precio de las acciones. Se mostrará que esta técnica genera previsiones que son mejores que algunos bechmarks de uso común. En el núcleo de este enfoque está la suposición de que los precios de las acciones siguen, al menos hasta cierto punto, patrones históricos. Además de la precisión, una ventaja de esta técnica es que genera pronósticos relativamente rápidamente, ya que no requiere una fase de entrenamiento que consume mucho tiempo. Tener una mejor comprensión de las pérdidas probables de la operación puede ayudar al inversor a tener un proceso de decisión de inversión más completo. Por lo tanto, también es importante, especialmente desde el punto de vista de la gestión de riesgos, contar con buenas técnicas de previsión que generen distribuciones de probabilidad para las predicciones, que puedan utilizarse para obtener intervalos que garanticen contener los precios reales futuros con una probabilidad prescrita. En esta disertación se ha adaptado y aplicado una estrategia de estimación de intervalos de precios probabilísticos al problema de encontrar intervalos de precios por delante de k-paso, obteniendo mejores resultados que los obtenidos con técnicas bien conocidas.

Tener buenas técnicas de previsión es sólo una parte del proceso de inversión. Después de que se haya tomado una decisión de compra o venta es importante llevar a cabo esa operación de manera eficiente en lo que coménmente se conoce como "mejor ejecución". Hay múltiples factores a tener en cuenta después de que la compra (o venta) de una acción se haya decidido, por ejemplo, en qué período del día (o durante cuántos días) se va a llevar a cabo la operación. Esto se relaciona con la idea de evitar distorsionar el mercado con la operación, es decir, una orden grande podría distorsionar involuntariamente los precios de mercado. En esta disertación se presenta un enfoque que genera mejores resultados que los puntos de referencia utilizados con frecuencia, como por ejemplo el precio ponderado por tiempo (TWAP) o el precio ponderado por volumen (VWAP). El enfoque se basa en el concepto de retroceso de la optimización del horizonte, bien conocido en la comunidad de control predictivo, pero no se utiliza en la optimización de la ejecución de grandes órdenes comerciales. La técnica puede acomodar órdenes de mercado y límite y ha demostrado un gran impacto económico potencial en la ejecución de órdenes tan grandes.

Contents

Chapter 1

Introduction

This thesis deals with some associated problems found when trying to devise investment and trading strategies. Successful trading strategies require not only tools to generate reasonably accurate forecasts, but also techniques to execute trading orders in an adequate way. The stock market has attracted researchers from very different fields creating multiple different types of investment strategies and techniques. The purpose of this chapter is not only to show the motivation, objectives and general layout of the thesis, but also to state a minimal background in some of the most basic concepts in finance, that are required by engineers and scientists that want to research in stock forecasting and trading. There are two major approaches to investing: fundamental investing and quantitative investing. The core concept behind fundamental investment is that there exists an intrinsic value for a security [Lee et al., 1999, Lai and Wong, 2015, Tiwari, 2016]. Quantitative investing [DeFusco et al., 2015], on the other hand, relies on historical prices and statistical tools to find the value of a security.

1.1 The value of a security

A key problem in investing is to be able to estimate the value of a company or, in general, the value of a security[1]. Two main styles of investing rely on two different interpretations on what constitutes the value of a security. In fundamental investing the value of a security is identified as its intrinsic value, following assumption 1.

Assumption 1 (Fundamental investing - Intrinsic value)**.** *There is a relationship between intrinsic value and actual securities prices [Lee et al., 1999] and in the long term securities prices, tend to their intrinsic value.*

[1]In this thesis a security is any financial instrument related to a company or institution that can be traded on the financial markets and produces an income for the investor. The securities used in this thesis are mainly stock shares, but all the methods and concepts can be applied to any traded security, e.g., debt bonds.

Once assumption 1 is accepted, fundamental investor seek to buy securities priced under their intrinsic value, so that they can profit by selling them once the price reaches the intrinsic value. This style of investing is also called value investing.

This intrinsic value is, in simple terms, the theoretical real value of a security or, in other words, the discounted value of the future cash flows of this security [Shrieves and Wachowicz Jr, 2001, Lundholm and O'keefe, 2001]. In practice, there are several ways to estimate this intrinsic value [Bask, 2020, Budagaga, 2020] that generates certain degree of uncertainty. A usual way to calculate the intrinsic value of a security is

$$P_{intrinsic} = \sum_{i}^{n} DF_i \cdot CF_i \tag{1.1}$$

where $P_{intrinsic}$ is the intrinsic price, the CF_i are the cash flow and DF_i the discount factor of cash flow i. Discount factor (DF) are typically express as a number $DF \in [0, 1]$, with DF=1 denoting that there is no discount while a DF=0 denotes that there is a 100% discount. Note that this intrinsic value does not necessarily coincide with the price of that security in the market.

There are several issues regarding the intrinsic value approach, such as the need to forecast the cash flows of the company for several years into the future [Sougiannis and Yaekura, 2001] which makes estimates rather challenging. The average length of this explicit forecast of future stream of cash flows varies from analyst to analyst, but in most cases they are forecasted for 10 to 30 years. Needless to say that this is an extremely difficult task. For example, very few analysts, if any, would have forecasted the stream of cash flows for Amazon in 1994 when it was founded. Amazon still does not have 30 years in business. Similarly, very few analyst would have forecast in 2005 the demise of Lehman Brothers a few years later. Furthermore, calculating the intrinsic value of a stock not only requires estimating (explicitly) the cash flows of that company for a long period of time but it also entails to estimate the terminal value i.e., and estimate of the cash flows after the explicit forecast period. There are multiple approaches to calculate the terminal value but they typically assume one (or multiple) growth rates after the initial explicit forecast. This can lead to very significant difference in the values obtained for the intrinsic value.

Another important factor to take into consideration when estimating the intrinsic value of a security is that the cash flow are discounted. This discounting accounts for the time value of money. The idea is that the same amount of money is likely more value now than in the future. This reflects not only inflation but also tries to account for the certainty of the cash flows. The longer the forecast time the less certain that those cash flows are. This can be clearly illustrated with a relatively extreme example. A 100 USD now are clearly worth more than the promise of 100 USD to be received 100 years later. Therefore, it is reasonable to add some type of discounting to the cash flows that are used to estimate the intrinsic value of a stock. This idea is commonly referred as the Time Value of Money.

There are several ways in which this discounting can be done in practice with several factors that can potentially affect it such as expected inflation, real inflation, actual interest rates, expected interest rate and even expectations on the broad monetary policy. It is also frequent to use indirect ways to calculate these discounts such as for instance the value of futures contracts[2], for instance, using a 12 months futures contract and comparing it with the values for two six months futures contracts. This can also be done annually with for instance a two year futures contract and two one year futures contracts. Theoretically, and assuming that the no arbitrage assumption holds[3], there should be no profit to be gain (or loss) from choosing between holding a two year futures contract or two one year futures contracts. Otherwise an arbitrage could be done exploiting this risk-free opportunity. The discount factor would be then estimated from the difference between the total price of two consecutive one year future contract and the price of a two year future contract.

Another disturbing aspect of this valuation process is that, while in theory the intrinsic value of a security should be an objective value, given the different approaches that are followed in practice to estimate it, as well as the discretion that have the analysts when estimate some of the inputs to the model, it is unlikely that two investors reach independently identical values. Furthermore, it is common to have rather different estimates even among professional stock price forecasters from the main investment banks such as JP Morgan, Goldman or Credit Suisse. These forecasters usually follow a bottom up approach, in which the forecaster builds its model starting with the sales and cost of the company rather than looking at overall macro trend in the economy. But, it is also possible, although less frequent in the field of stock forecasting, to follow a different approach based on a top down analysis, in which the forecaster starts by analyzing the macro economic conditions and then moves gradually to the company level. Either approach finds usually a different price and there is not an agreement on which is best. In a bottom up approach the usual criticism is that the analyst might miss overall macro tendencies when doing the analysis of the company. On the other hand, a common criticism of a top down approach is that it is less specific i.e., less accurate, for a company level analysis. Regardless of the approach used, the intrinsic value of a security should be understood as a theoretical value which a well informed and rational investor would pay for the stock. This concept is at the core of fundamental investing, more precisely of value investing, which is the approach followed by some of the most renowned investors like Warren Buffet. It should be noted that the intrinsic value of a security changes over time, as more information is gradually released, such as for instance the annual or quarterly reports of the company but that, as assumption 1 states, in the long term security prices trend to their intrinsic value.

The other major investing style is that which rely on a quantitative pricing scheme of the value of a security, known as quantitative investment [DeFusco et al., 2015]. In this type of investment strategy, the objective is not to estimate the intrinsic value (true value) of a security, but the price (or price trend) that the stock will follow over a certain time horizon, which tends to be short to midterm investment [Atsalakis and Valavanis, 2009, Ince and Trafalis, 2008]. In this investing style, rather than assuming that there is an intrinsic (real) value of the security, it is assumed that security prices are dictated by

[2]Futures contracts are contracts to buy or sell a security at a future date. For instance, an investor might decide that wants to purchase a certain amount of stock A but rather than doing it immediately enters a contract for those shares to be delivered (ownership change) six months later. In fact there is a Spot Market, where securities are currently traded and the Futures market in which the securities are traded at a later date in the future with a contract arranged in the present.

[3]Basically this assumption states that the value of two securities with the same price must be the same.

demand and supply forces and that the value is identified by the price of the security. This imply that statistical and machine learning tools are applicable for its analysis. In simple words, in quantitative analysis the investor uses statistical and learning tools applied to historical prices and other quantitative indicators in order to estimate future prices and trends, without having to know the internals of the companies that are represented by the security, i.e., without having to look for the book value. Quantitative investment involves a large set of different strategies, based on statistical analysis and machine learning. Thus, this investment approach has attracted researches from the fields of mathematics, statistics and engineering.

Fundamental investing is the traditional way of approaching investments and there are a large number of investors and financial institutions following this approach. Quantitative investment on the other hand is relatively more recent, and has experienced a considerable expansion in the last few decades as computers became more adapt to process large amount of information. Currently there is a large amount of financial institutions following this type of investment approach. Usually, the investment time frame of a fundamental investor tends to be longer than the investment time frame of a quantitative investors. In fact, it is common for quantitative investor to have daily or intraday time horizons while fundamental investors tend to have monthly, yearly or even decades time horizons, in hope that assumption 1 will finally hold.

1.2 Main financial theories related to stock forecasting

There are two, competing, financial theories that underscore the differences between quantitative and fundamental analysis and that have implications in stock forecasting. These theories are:

- Efficient market hypothesis.
- Behavioural finance.

In the following sections a brief introduction of these two competing theories are presented as well as their implications for stock investing.

1.2.1 The theory of the market efficiency

This theory is based on the market efficiency hypothesis that has, in turn, three variants: the strong, semi-strong and weak market efficient hypothesis. These ideas basically related to the degree in which current information is reflected on the price of a security. This degree affects to the feasibility of stock forecasting based on different tools.

Definition 1.1. ***Weak form market efficiency***. *All the information contained in historical prices is already contained in security prices.*

If the weak form market efficiency theory is right then it is not possible to outperform the market using historical data, such as stock prices and volume. Therefore quantitative analysis is not a valid approach for stock investing.

Definition 1.2. ***Semi-strong form market efficiency***. *All the information contained in historical prices as well as in the analysis of the company fundamentals is already priced in equity prices.*

If the semi-strong form market efficiency theory is correct then both quantitative and fundamental analysis are both inappropriate tools for stock investing. In this way no amount of analysis based on historical data or in the analysis of the company fundamentals, such as for instance its accounting data, financial statements and business model, can be used to create an accurate stock forecasting model.

Definition 1.3. ***Strong form market efficiency hypothesis***. *All the information, both public and private, is already priced in equity levels.*

If the strong from market efficiency hypothesis is correct then no amount of analysis, regardless of the type, can be used to generate accurate forecasting models. Not even insiders of the company, such as for instance a CEO, can generate an accurate stock forecast and benefit from it using all the public and private information available to them.

Clearly, depending on which hypothesis, if any, is true, the stock forecasting, already perceived as a very difficult task, can be, in fact, impossible at all. However, there are other approaches to financial theory that suggest that the markets are not completely efficient because investors are subject to biases and do not have access to all the information that can potentially impact the price of the stock of a company [Shiller, 2000]. Thus, forecasting the stock price using historical data would be a difficult but possible task.

1.2.2 The behavioral finance theory

Behavioral finance is based in the idea that stock prices are dictated by the buy or sell decisions of individuals that are dominated, not by a perfect analysis of all the information available that can possibly impact the price of a stock, but by human biases. There are a large amount of human biases related to stock investing, such as for instance confirmation bias, loss aversion, overconfidence or irrational information processing. These are terms precisely defined and understood in the context of behavioral finance. For instance, confirmation bias relates to the idea that an investor, when he/she has decided what type of trade to do will look for information that confirms that trade i.e., looking for confirmation, and will unconsciously discard information that puts into question the trade. Overconfidence is frequently referred as a bias, common among male investors, based on not reassessing the investment thesis, assuming that it is undoubtedly correct. Irrational information processing relates to the idea of having a situation in which the investor cannot properly process the facts related to a trade. This term should not be confused with the issue of information asymmetry. Information asymmetry occurs when two different investors have substantially different levels of information related to a potential trade. A classical example of information asymmetry could be the difference between a professional investor and a retail (individual) investor. A professional investor is likely to have access to more information, such as for instance analyst research reports or better databases, than an individual investor but this is however not necessarily an example of irrational information processing as the retail investor could, at least in principle, analyze all the available information presented to her/him in a rational way (regardless of how limited that information might be).

Similarly, it is important to make the distinction between loss aversion and risk aversion. Every rational investor should be risk adverse. In this context risk adverse means that the investor will prefer to achieve certain return (X) taking certain level or risk (R_1), rather than achieving the same return (X) taking a higher level or risk (R_2) (with $R_2 > R_1$). It should be noted that behavioral finance does allow for the existence of investor that prefer to take higher levels of risk for the same return (thrill seekers), due to psychological characteristics. Loss aversion on the other hand, is different from risk aversion, and it is related to the reward/physiological pain relationship. In the context of investing loss aversion, in simple term, means than an investor experiences more physiological pain when losing in an investment certain amount (A) then physiological pleasure when gaining the same amount (A). This might lead to overly cautious investment approaches, as well as holding to losing position for longer than expected. An example of this is an investor which, after experiencing a significant loss in an investment, rather than selling the stock (making the loss real but avoiding potential additional losses) keeps the position in an (illogical) hope that the stock will recover its value because the amount of emotional pain is too large to cut the losses.

An implied assumption in behavioral finance is that not all the information available is effectively logically and immediately analyzed. and therefore, contrary to the efficient market hypothesis, investment techniques can generate accurate forecasts. [Ritter, 2003] mentioned that, in the context of behavioral finance, it is assumed that some market participants do not behave in a fully rational manner. In [Thaler, 2005], it is mentioned that example of the internet bubble at the beginning of the twenty first century, in which, according to the author, it is difficult to argue that all market participants were behaving in a fully rational way. [Glaser et al., 2004] showed how some of these physiological biases, intrinsic to behavioral finance, can be modelled. Furthermore, as human behavior follows patterns it is potentially possible to use statistical techniques.

1.2.3 Implications of the efficient market hypothesis and behavioral finance theories

It should be noted that there is currently no market consensus regarding the validity of either the efficient market hypothesis or behavioural finance, with two major universities, the University of Chicago and the University of Yale, both in the US, representing two opposite schools of thought in this regard. Nobel laureate professor Eugene Fama, from the University of Chicago, and several of his colleagues at the same university are the main proponents of the market efficient hypothesis. In fact, professor Fama received the

Nobel Prize in economics in 2013 for the “empirical analysis of asset prices”. Interestingly, the same year, professor Robert J. Shiller, from the University of Yale, also received the Nobel Prize (they shared the Nobel Prize) for his work on behavioral finance, which is an opposite theory to the market efficiency.

The acceptance in the academic community of the efficient market hypothesis has fluctuated considerably. The theory was rapidly accepted in the seventies when it was first introduced becoming the dominant theory during the following few decades. This started to change by the end of the twenty century with the advent of new theories, such as behavioral finance, with authors such as [Malkiel, 2003] mentioning the change in status quo. There is an extensive literature regarding the topic of market efficiency. However, as just mentioned, there is less academic consensus about its validity than in the previous decades. There are for example mixed results when analyzing the stock market of different countries during different periods, see for instance [Kim and Shamsuddin, 2008].

1.3 The random walk

Another approach that is frequently followed in finance is modelling the stock price as a random walk. The objective of this approach is not so much trying to forecast stock price, as in a random walk it is assumed that forecasting the stock price in the next period (T+1) is not possible, but having a model that roughly gives an indication of the possible trends of the stock. This is sometimes used as a benchmark to measure the accuracy of forecasts generated by other techniques, such as illustrated in [Pemy, 2012]. A popular random walk model is the Brownian motion model [Pemy, 2012] that can be described by

$$\begin{aligned} X(k+1) =& X(k) + X(k)\mu t + \sigma X(k)\zeta(k)\sqrt{t} \\ X(0) =& X, \end{aligned} \tag{1.2}$$

where $X(k)$ is the stock price in step k, $\zeta(k)$ is a random input with zero mean and unity variance, σ the standard deviation, t the time fraction and μ the mean return. In figure 1.1 it can be seen an example of an actual stock modeled as a random walk and its real prices. This figure shows how the Brownian motion (random walk) it is not strictly speaking a forecasting model. In other words, the Brownian motion approach does not try to generate at time (T) an accurate forecast for the value of the stock in the next time step $(T+1)$ but simply tries to provide an indication of possible values for the stock according to historical volatility and returns.

The implied basic assumption in a random walk is that the stock price follows a random process that cannot be forecast with accuracy. The random walk approach is followed in some well known financial techniques such as for instance option valuations. In fact the 1997 Nobel Prize in economics was obtained by Robert C. Merton and Myron S. Scholes for their work on the Black-Scholes-Merton[4] model on option valuation. However, it should be noted that, similar to the previous section, there is no scholar consensus [Cooper,

[4]Professor Black, who died before 1997, was not awarded with the Nobel as this prize can only be granted to researchers that are alive at the time of the prize decision.

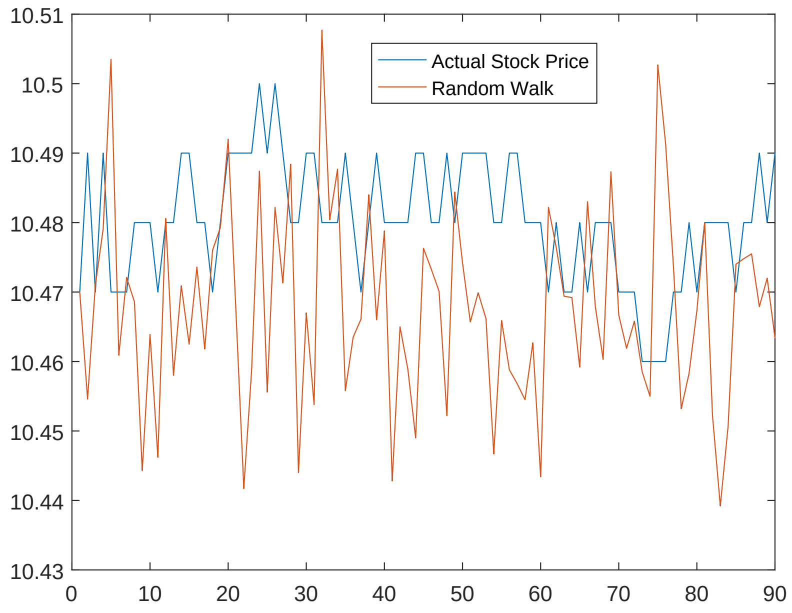


Figure 1.1: Sample of a stock price modeled as a random walk.

1982, Dupernex, 2007] regarding the validity of the random walk theory with multitude of papers both supporting it, such as [Fama, 1995], and strongly rejecting it, such as [Lo and MacKinlay, 1988, Frennberg and Hansson, 1993, Chaudhuri and Wu, 2003, Poshakwale, 2002, Hoque et al., 2007]. The discrepancies in views in the existing literature regarding the random walk hypothesis is rather substantial. Much of the discussion seems to be related with the specific market (country) and period analyzed with, overall, more support for the stock market following a random walk in the mid to late twenty century in the United States and, to some degree in the United Kingdom as well. There is however considerable amount of literature supporting that stock markets in several European markets as well as emerging markets, particularly in the twenty first century. Other factors, such as the liquidity of the market (narrow markets compared to deeper markets) has been proposed as factors impacting the random walk hypothesis on stock prices.

The random walk hypothesis and the efficient market hypothesis are not unrelated. If market are fully efficient, therefore all the information (both private and public) is already incorporated in stock prices and hence stock should follow a random walk, i.e., a random process in which historical information (both quantitative and fundamental) cannot be use to develop an efficient trading strategy. The diverging views over the last few decades regarding the market efficient hypothesis has caused many scholar to question the, once widely accepted, random walk hypothesis with, as previously mentioned, that in at least some markets stocks do not appear to follow random walks. This is in fact one of the major open questions in finance and has been debated since the 1960s. Some proponents of random walks have equated stock investing with monkeys randomly throwing darts to the financial pages (figure 1.2) of the newspaper as there is no possible way of picking stocks in a sustainable way. In other words, if stock markets can be accurately described as a random walk, then the strong form market efficiency hypothesis is likely true, and it would be extremely difficult, likely impossible, to generate accurate stock forecasts and hence trading strategies. It should be noted, however, that there appear to be investors, such as for instance Warren Buffet that seem to consistently outperform the stock market during a multi-decade investment career.

Perhaps the main takeaway of the literature review regarding both the efficient market hypothesis as well as random walk is that, while in previous decades they were financial dogmas, there is currently an increasing amount of literature against both of these ideas. The difference in methodologies, as well as on the data analyzed, make direct comparison among papers rather challenging. Furthermore, it is possible that the validity of these two concepts depends on the time period analyzed, i.e., with possible regime changes from one period to the next, as well as of the stock market analyzed. As previously mentioned there seems to be more support for the validity of these ideas in the US market while, there appears to be less empirical data supporting them in some European and emerging markets. A factor that should be also be taken into account is the potential for data mining, i.e., looking for a data set that corroborates the assumptions of the researcher, rather than trying to validate (or disprove) assumption with multiple datasets.

Another important factor that will be addressed later in this dissertation is the quality of the underlying data, with some narrow markets having issues such as stale prices. If the underlying data is not reliable, then the conclusions from the models will not be reliable either. Stale data is usually related to lightly traded markets. In some narrow markets there is no trading in all securities everyday with the data showed in some data providers been the data from previous days. Less extreme cases are also important, for example, there might be only a few trades carried out by a few retail (individual) investors with the price recorded by data providers obtained from such trades. However, this price might not be representative of the price level at which an institutional investor can actually carry out a (large) trade. These are just some examples of data issues that can impact the results of the analysis, and that can potentially generate erroneous conclusions regarding the validity of the efficient market hypothesis and/or the random walk hypothesis.

1.4 Motivation and objectives

While there is a large number of equity forecasting tools as well as trading techniques there is no consensus in the literature of the best possible approach. Therefore, developing techniques that improve, even is slightly, in the current techniques or that add some benefit, such as for example an algorithm that can be trained quicker than existing approaches, or with better forecasts, can have valuable practical applications. Stock trading is a highly complex process which can be impacted not only by multiple factors, such as human biases. Nevertheless, the potential rewards of even a small improvement over current techniques can be substantial as the frequency of the trading and the amount of shares traded magnify potential gains.

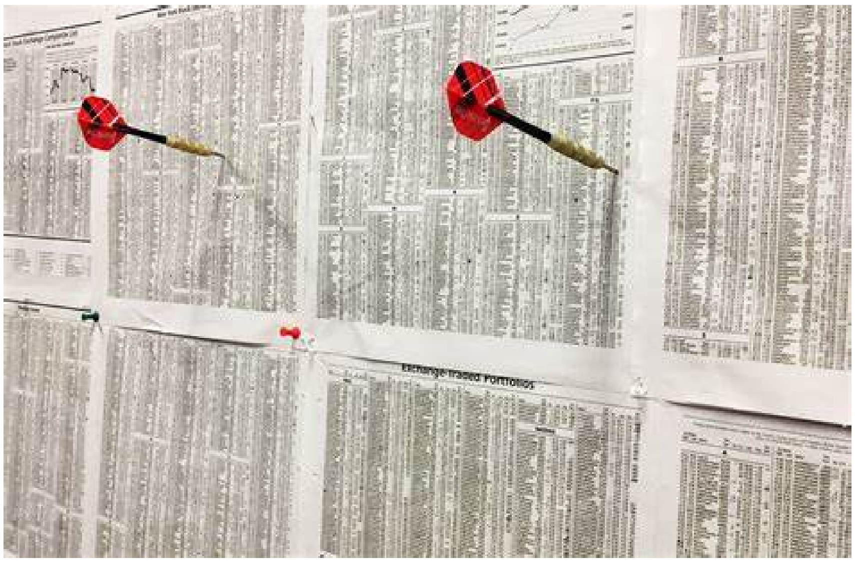


Figure 1.2: Dart throwing to the financial pages.

Statistical analysis of the stock market can be used not only to try to increase profits but also as a risk management tool. Risk management is becoming an increasingly important field of research as practice. In every trade there is some risk that the investor will lose money. Developing techniques that help managing or, at least realistically quantify the exposure of the investor to the market, is of clear practical value. All major international banks have large department with multimillion dollar budget allocated to such function.

Another motivation behind this dissertation is the need to developed better trading models. Besides improvement in stock price forecasting there are other considerations to be taken into account when trading. In fact, the term "best execution" is increasingly important in the financial sector. Best execution refers to providing to the client an adequate trading techniques, after the buy or sell decision has been taken. The trading quality (after investment decision is made) is typically measured by comparing the average price obtained with a benchmark. Frequently used benchmarks are the Time Weight Average (TWAP) and the Volume Weighted Average (VWAP), which will be used in chapter 5.

Based on these motivations the main objectives of this dissertation are as follows:

- Create forecasting techniques for stock prices.
- Develop techniques with applications in trading risk management.
- Devise trading techniques based on optimization and forecasting.
- Asses the application of learning techniques to stock forecasting.
- Develop algorithms for automated selection of technical indicators,
- Evaluate the applicability of forecasting techniques in stock markets with peculiarities, such as for instance narrow markets.

1.5 Main results

- Neural networks were successfully applied to markets such as the Indian stock market, generating reasonably accurate forecasts.
- The applicability of neural networks in narrow markets has been tested, concluding that they are suitable tools, identifying, however, practical issues such as stale prices, that could generate forecasts that superficially appear to be accurate, but that do not realistic represent the price at which an actual trade can be placed .
- A new algorithm to select an adequate combination of technical indicators for stock forecasting purposes has been presented. The combination of technical indicators obtained in the simulations generated more accurate forecast than when using directly all the technical indicators available.
- A technique, based on local data, was applied for the first time, to the task of stock forecasting. The technique does not need training and the computational burden is quite low. Furthermore, the estimates were more accurate than some frequent benchmarks.
- Interval forecasting based on dissimilarity functions has been applied for the first time to the problem of price interval forecasting, adapting the technique to this particular problem, using only local data to better handle big datasets. This type of application can have direct practical applications in the field of risk management, as having a better understanding of the potential losses, with a related probability distribution, can help managing investments in a more balanced way. The results have been favourably compared to a well known technique.
- Techniques for trading optimization. After the purchase or sell of a stock is decided by the investor it is important to carry out in an efficient way in what is frequently described as "best execution". There are multiple factors to take into account such as for instance the potential impact of the order in the market and the need to divide the order into smaller trades. In this dissertation a technique to carry out such type of execution has been presented. The technique is based on forecasting and optimization and can accommodate different types of stock orders. The results, from the application to the Chinese Stock Market, show significant savings over two well known price benchmarks.

1.6 Thesis overview

This dissertation is structured in six chapters plus appendixes. Beyond this first introductory chapter, the second chapter of the dissertation is devoted to show how neural networks can be applied for stock forecasting purposes, generating accurate forecasts in several markets. A section of this chapter is dedicated to analyze the applicability of neural networks to narrow markets.

In chapter three it is presented an algorithm for the automated selection of technical indicators. There is a large amount of technical indicators available with some of those indicators generating conflicting signals. Given this large amount of indicators it is not feasible to check all possible combinations. It is also shown empirically in this chapter that it is possible to obtain a combination of indicators that produce more accurate trading signals than using directly the entire set of indicators.

In chapter four a technique, based on local data, has been applied for stock forecasting purposes. This technique is based on the idea of using historical data that best approach the current stock situation. It has been illustrated by applying it to the Dow Jones Industrial Index. For completeness purposes other techniques, such as neural networks, were also used to determine a base benchmark for the technique. In this chapter it is also show how to apply to price interval forecasting a technique that build a probability distribution around a stock price forecast. This approach can have applications on risk management techniques.

In chapter five it is presented a technique for the implementation of trades. After the investor has decided to do a trade, this trade has to be executed in an adequate way, in pursuit of a “best execution”. Factors such as not distorting the market price with an excessive large trade (which can be potentially be divided into smaller trades) must be taken into consideration. The proposed technique, based on forecasting and optimization, has proved to obtain better results than the TWAP and VWAP prices in several stocks from the Chinese Stock Market.

Finally, in chapter six the conclusions, in the form of a summary of contributions, and some directions for future work are presented.

1.7 Publications

The list of research papers in indexed journals as well as conference papers carried out during this dissertation are as follows:

Journal papers

- Alfonso, G.; Ramirez, D.R. A Nonlinear Technical Indicator Selection Approach for Stock Markets. Application to the Chinese Stock Market. Mathematics 2020, 8, 1301. Impact factor 1.747, JCR ranking: Q1 (2019).
- Alfonso, G.; Ramirez, D.R. Neural Networks in Narrow Stock Markets. Symmetry 2020, 12(8), 1272. Impact factor 2.645, JCR ranking Q2 (2019).
- Gerardo Alfonso, A. Daniel Carnerero, Daniel R. Ramirez, Teodoro Alamo. Stock forecasting using local data. IEEE Access 2021, vol.9, pp.9334-6344. Impact factor 3.745, JCR ranking Q1 (2019)

International conference papers

- Alfonso, G.; Ramirez, D.R. Forecasting the Indian Stock Market by Applying the Levenberg Marquardt and Scaled Conjugate Training Algorithms in Neural Networks. CSAE 2017. The International Conference on Computer Science and Application Engineering. Shanghai. China.

Unpublished papers

- Gerardo Alfonso, A. Daniel Carnerero, Daniel R. Ramirez, Teodoro Alamo. Receding horizon optimization of large trade orders. Submitted to IEEE Access (2021).

Chapter 2

Neural networks in stock prediction

This chapter is based on the following journal publication:

- Alfonso, G.; Ramirez, D.R. Neural Networks in Narrow Stock Markets. Symmetry 2020, 12(8), 1272

As well as from a conference presented at an international conference

- Alfonso, G.; Ramirez, D.R. Forecasting the Indian Stock Market by Applying the Levenberg Marquardt and Scaled Conjugate Training Algorithms in Neural Networks. CSAE 2017. The International Conference on Computer Science and Application Engineering. Shanghai. China.

2.1 Introduction

A large amount [Granger, 1992, Shively, 2003, Wang et al., 2012b] of stock forecasting techniques have developed overtime, with an increase in the number of such techniques in recent years, as asset prices became easily available and computer power, not only significantly increased, but also became more affordable. Among the different quantitative and machine learning techniques applied to stock forecasting, one of the most popular are neural networks [Zhang et al., 1998, Zhang and Suganthan, 2016, Cao et al., 2018, Refenes et al., 1994, Ballings et al., 2015, Kanas, 2001, Lahmiri, 2016, Li et al., 2016, Li et al., 2016, Quah and Srinivasan, 1999, Guresen et al., 2011, Moghaddam et al., 2016, Qiu and Song, 2016, Kimoto et al., 1990, Yoon and Swales, 1991, Zhang, 2007, Naeini et al., 2010, Chen et al., 2016, Qiu et al., 2020]. Neural networks are very flexible tools that do not require any previous modeling of the underlying system. However, there is a large amount of factors to take into consideration when building a network. Among these, the most important basic characteristics necessary to define a neuronal network are, the number of neurons and layers and the learning algorithm (supervised learning), that will be used to train the network. Artificial neural networks are composed by artificial neurons that

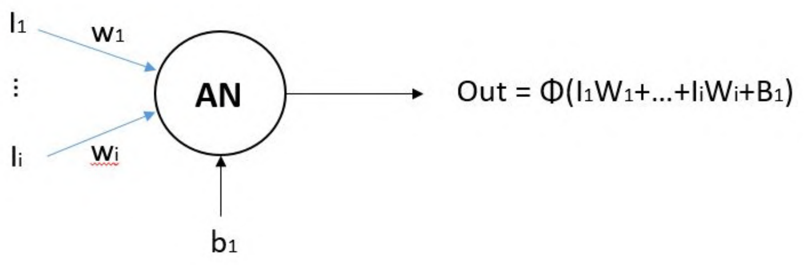


Figure 2.1: Typical Artificial Neuron. Where I_1 to I_i are the inputs, W_1 to W_i the related weights, b_1 the bias and Out the output.

are mathematical expression inspired on the function of biological neurons. A sample of a typical artificial neuron can be seen in figure 2.1.

There are typically at least three steps when creating a neural network. In the first step the basic network architecture is chosen. In the second step, when the basic structure is already in place the network is trained. The training process entails some type of algorithm that iteratively adjust the weights of each neuron in an attempt to generate forecast that are as close as possible to the actual value. In order to do this it is necessary to have an error function, measuring the distance between the forecasted and actual values. This error is what the supervised learning algorithm tries to minimize. After that, in the last step, the network is used to create forecasts from previously unseen data (set aside during the training phase), and the forecasting accuracy is calculated. This last step is typically done to avoid the issue of over-fitting which can cause the network to generalize poorly or, in other words, perform poorly when applied to new (unseen) data.

The remaining of the chapter is organized as follows: section 2.2 presents a brief review of the training algorithms used in the chapter. The application of neural networks to stock forecasting in narrow markets is discussed in section 2.3. Finally, section 2.4 presents the application of neural networks to forecast one of the main indexes of the Indian stock market.

2.2 Training Algorithms

The choice of the learning algorithm as well as the number of neurons used can have a very significant impact on the results. In this section a very brief description of the training algorithms used for the neural networks is shown. All of them are well-known algorithms with applications in several areas. All the actual implementation of the algorithms used in this chapter are those of the Matlab Neural Network Toolbox, so, for clarity purposes, its notation and acronyms are followed.

Quasi Newton back propagation

The quasi Newton (BFG) [Dennis Jr and Schnabel, 1996] is a powerful and relatively fast training algorithm as it skips the exact computation of the second derivatives that it is necessary in the Newton's method. The objective is to minimize a performance index, $f(X)$, which measures forecasting accuracy as a function of X, the network bias and weights. The basic rule to iteratively adjust the value of X is

$$X_{i+1} = X_i - \gamma_i \frac{H_i}{\nabla f(x_i)}, \tag{2.1}$$

where the parameter γ_i is selected to minimize the performance along the search direction, X_i are the weights and bias at iteration i, H_i is an approximate Hessian of $f(X_i)$ and $\nabla f(x_i)$ is its gradient [MathWorks, 2010].

Conjugate gradient

Conjugate gradient methods, are popular training algorithms that update the weights according to the basic rule

$$X_{i+1} = X_i + \gamma_i d_i, \tag{2.2}$$

where γ_i plays the same role as in BFG method and d_i is a vector in the direction of the gradient, which is in turn updated according to

$$d_i = -\nabla f(X_i) + \phi_i d_{i-1},$$

being ϕ_i a constant with a different value depending on the conjugate gradient method considered. In the case of the Fletcher Reeves (CGF) algorithm [Al Baali, 1985] the scalar ϕ is calculated as

$$\phi_i = \frac{\nabla f^T(X_i)\nabla f(X_i)}{\nabla f^T(X_{i-1})\nabla f(X_{i-1})}. \tag{2.3}$$

On the other hand, in the case of the [Khoda et al., 1992, Zhang et al., 2006] Polak Ribiere (CGP) variant the parameter is computed using

$$\phi_i = \frac{(\nabla f(X_{i-1}) - \nabla f(X_{i-2}))^T \nabla f(X_i)}{\nabla f^T(X_{i-1})\nabla f(X_{i-1})} \tag{2.4}$$

Another of the conjugate gradient variant, and more complex, is the Fletcher Powell (CGB) algorithm [Powell, 1977]. Powell describes the idea as automatically resetting the search direction (restart) and it is done according to some conditions on the orthogonality of the gradient at successive steps. First, the algorithm uses a test to determine when to reset, i.e., the reset is performed if

$$|\nabla f^T(X_{i-1})\nabla f(X_i)| \geqslant 0.2||\nabla f(X_i)||^2. \tag{2.5}$$

If this condition hold, the search direction is reset to the negative of the gradient. Second, the search direction is computed from the negative gradient, the previous search direction, and the last search direction before the previous reset.

Gradient Descent

Gradient decent algorithms are among the most popular training techniques for neural networks due to their relative simplicity, and they use the gradient as the main component for the changes in each iteration. Three different types of gradient descent algorithms are used in this chapter. The first one was the gradient descent with momentum (DM), as implemented by Matlab function. This algorithm adds momentum considerations in an attempt to avoid the issue of local minima. The rule to compute dX_i, i.e., the change of X_i is

$$dX_i = m_i dX_{i-1} + l_i(1 - m_i)\frac{\nabla f(X_i)}{dX_i}, \tag{2.6}$$

where l_i and m_i are respectively the learning rate and the momentum constant. The momentum constant, which is in the $[0, 1]$ range, is used as the pole of a low pass filter that allows the network to ignore small changes in the performance function, thus avoiding local minima. The learning rate on the other hand is increased as long as the performance function decreases and decreased if it increases.

Another training algorithm used was the gradient descent with adaptive learning (DA) variant. In this approach the learning rate is dynamically adjusted, taking into consideration the accuracy of the forecasts in each iteration. The resulting updating rule is

$$dX_i = l_i\frac{\nabla f(X_i)}{dX_i}, \tag{2.7}$$

where l_i is the learning rate which is adjusted to higher values as long as the performance function decreases and decreased otherwise.

A third variant used was a combination of the previous two. This method is called gradient descent with momentum and adaptive learning (DX) and its update rule is

$$dX_i = m_i dX_{i-1} + l_i m_i\frac{\nabla f(X_i)}{dX_i}, \tag{2.8}$$

which can be seen as a rule that combines both the momentum and adaptive learning concepts of the previous rules.

Other methods

Other methods, such as the Levenberg Marquardt (LM) approach, have been also used in this chapter. This is a relatively simple approach with moderate computational requirements. The basic rule to iteratively update the weights and bias is

$$X_{i+1} = X_i + (J^T J + \mu_i I)^{-1} J^T \xi, \tag{2.9}$$

where μ_i is a parameter which is adjusted to ensure that the performance function decreases at each iteration, J is the Jacobian of $f(X_i)$ and ξ is the network error. When the μ_i is zero, the algorithm is equivalent to Newton's method. When μ_i is large, the update is that of gradient descent with a small step size. Newton's method is faster so the parameter μ_i is decreased after each successful step (the performance function decreases) and is increased only if the update would increase the performance function. In this way, the performance function always decreases at each iteration of the algorithm.

Another method used was the [Battiti, 1992] Secant approach (OSS). This is a frequently used technique in which the iterative change in values is dictated by a function of the current gradient and the gradient in the previous iteration. In this approach the update rule is

$$dX_i = -\nabla f_i(X_i) + \kappa_A dX_{i-1} + \kappa_B \left(\nabla f(X_i) - \nabla f(X_{i-1})\right), \tag{2.10}$$

where κ_A and κ_B are constants.

The last technique used was resilient backpropagation (RP). This is another frequently used training algorithm in which the update at each iteration is defined by

$$dX_i = \Delta X^T \text{sgn}(\nabla f(X_i)), \tag{2.11}$$

where sgn denotes the signum function. At each iteration the elements of ΔX are modified. If an element of the gradient changes sign from one iteration to the next, then the corresponding element of ΔX is decreased by a predefined amount. Conversely, those elements of the gradient that maintain the same sign from one iteration to the next, indicate that the corresponding element of ΔX must be increased by another predefined amount.

2.3 Neural networks for stock forecasting in narrow markets

A narrow stock market can be defined in several ways. Narrow markets tend to be defined in the literature in the sense of a thin market, or in other words markets with low liquidity. It should be noted that in the context of this dissertation, narrow market is understood as, not only, encompassing relatively illiquid markets but also markets that, while having relatively ample liquidity might present significant price distortions due to structural factors, such as having a large percentage of the traded volume done by individual investors, making price discovery more difficult[1]. An example of a thin, or in this context context narrow, market could be the equity market in Switzerland. Switzerland has a very large, particularly when compared to its overall GDP, financial sector but its domestic stock market is relatively small. Bruand [Bruand and Gibson-Asner,

[1]In fact, markets with a large proportion of institutional investors are usually considered as reflecting prices in a more rational way than in markets were the predominant investors is retail. The underpinning of this idea is that institutional investors have better information and more training and, hence, would make, on average, more reasonable investment decisions.

1998] selected this market as representative of a thin market[2]. In this chapter moderately narrow markets such as the one in Switzerland and very narrow markets, for instance Namibia are analyzed separetely. There is ample existing literature describing the use of neural network as a forecasting tool in deep stock markets. Narrow stock markets, perhaps because they tend to be located in less developed economies, have attracted considerably less academic research. Having tools that can generate acceptably accurate stock forecasts can be useful for the development of the stock market in those narrow markets. In turn, the development of the stock market can also potentially help the development of the economy of that country. Therefore, it seems of importance to analyze if well-known techniques, such as neural networks, are actually applicable for stock market forecasting purposes on narrow markets.

One of the objectives of this chapter is to get a better sense of the feasibility [Wang et al., 2011], under relatively realistic assumptions, of the applicability of neural networks as a forecasting tool in narrow markets. Even in this type of narrow market, it will showed that neural networks are robust enough to generate relatively accurate forecasts. The forecasts obtained in this type of market were comparable to those of deeper market but, at least in most cases, of moderately lower accuracy. These results are consistent across a wide range of learning algorithms and other network features such as the number of neurons. However, practical considerations such as potentially suboptimal trading infrastructure and stale prices should be taken into considerations.

It should be taken into account that some of the most narrow markets might have stale prices. The reason is that some quoted prices are not representative of an actual trade in the day analyzed but of trades on previous days as there was no or very little trading activity on the day analyzed. For instance, data providers companies might use the latest price traded for that stock, even if in the day analyzed that security did not actually traded. Similarly, it is possible that there is only a very small trade, perhaps by a limited amount of retail investors. This type of trade would generate a misleading price, particularly for an institutional investor trading relatively large amounts. Stale prices might cause the price level of the index to have an estimated volatility lower than its real volatility as a frequently used measure of the volatility of the stock is the standard deviation of the price of the stock. If the stock is not traded for a period of time, i.e., a few days, and the data provider uses a constant price for all those days, then the volatility of the stock (or index) will be artificially low and not representative of the actual market values. This should be taken into considerations when developing investment strategies. As the price does not appear to change (stale prices) the forecast, at least on paper, might look particularly precise but in practice the investor following an strategy based on such forecasts might obtain undesired results.

Two important conclusions frequently cited in the literature [Barnes, 1986] regarding thin markets are that price discovery is more difficult, as prices do not necessarily reflect the actual price of the stock, and that thin markets are more easily manipulated than deep markets. For instance, an unscrupulous investor with a relatively small amount of capital could "corner the market" becoming the dominant player in a security, thus

[2]This paper illustrates that narrowness is something that can be changed, e.g., by introducing new financial instruments like derivatives, that in the case of Switzerland helped in improving liquidity.

distorting prices. Such a type of malpractice would be much more difficult in a deep liquid market where the investor represents just a very small part of the total traded volume and becomes in practice, at least to some degree, a price taker. A price taker in this context is simply an investor which trades are too small, compared to the overall market, to impact in a material way the price of the security traded. As an example, a retail (individual) investor might well trade a few hundred shares in a large cap stock that can potentially have millions of shares traded daily i.e., retail investor buying a few shares of Coca Cola. It is unlikely that such trade would impact the price. On the other hand, a large institutional investor, such as a pension fund, mutual fun or sovereign wealth fund, might have position on a stock worth hundreds of million of dollars. Trading in or out of those position can, at least in principle, have a significant impact on the share price.

The question if the stock market can be forecast using techniques, such as neural networks, using historical prices is not a trivial one, regardless if the particular market analyzed is considered narrow or otherwise. The efficient market hypothesis, created by Fama [Fama, 1991], supports the idea that stock prices cannot forecasted using only inputs such as historical prices and trading volumes. One of the underlying implied assumptions of the efficient market hypothesis is that information flows, almost immediately, as stock prices reprice, basically, instantly reacting to all new (private and public) information. In this context, narrow markets are particularly interesting because it is conceivable that the information flow in narrow stocks markets, like for instance in Tanzania, being slower and arguably less efficient than in markets, such as the United States, that have a better telecommunication infrastructure.

Assuming that markets are not completely efficient, in which case there is no point in using any type of stock forecasting tool, then finding tools that generate relatively accurate forecasts is of a topic of clear importance. As previously mentioned, narrow markets have not received the same level of interest than deeper markets, with much less existing literature covering those markets. Nevertheless, there are some interesting articles in the topic. For instance, [Virtanen and Yli-Olli, 1987] treated the Helsinki Stock exchange as a thin market and tried to determine if accurate forecasts could be done. They concluded that forecasts were doable in this thin market, obtaining better results for a one-month time horizon than for quarterly predictions. On the other hand, [Idowu et al., 2012] found that neural networks are applicable tool for forecasting stock prices in the Nigerian stock market, which is an example of a narrow markets. The neural network used in this article was a feedforward network. An example of an application of neural network in a moderately narrow market can be found in [Mostafa, 2010]. In this article the author analyzed the stock market of Kuwait, concluding that neural networks are an appropriate tool for that market. Another interesting article is from [Senol and Ozturan, 2009] that analyzes the stock forecasting abilities of neural networks in the stock market of Turkey, which is another narrow market. [Senol and Ozturan, 2009] concluded that their results seem to contradict the efficient market hypothesis. Similar results were found by [Samarawickrama and Fernando, 2017] in the case of the stock market of Sri Lanka. The similarity of these papers is that they tend to analyze one specific country without considering similarities, such as classifying countries according to their level on narrowness. They also tend to use a relatively small number of learning algorithms, and they usually do not compare those results with the ones obtained in other markets, such as the US or Europe.

2.3.1 Stale prices and data availability

Stale prices and data availability are typically not real concerns in the stock markets of many developed economies, but some equity markets, such as Namibia, do present some data issues. In some periods there was no, or very little, trading in the Namibian index. This leads to the classic issue of stale price, as the quoted price might not reflect the current "true" price but the latest transaction that might have happened on a previous day. A related issue is when trading has occurred on a stock on that day, but the amount traded is too small to be useful as an indication of the current price for a trade. This is of particular importance for institutional investors that tend to trade larger amounts than retail investors.

Stale prices are likely to produce artificially good forecasts, as the estimate for the volatility of the stock is likely underestimating the real volatility. In Figure 2.2 the normalized traded volume for the 2012 to 2017 period can be seen. The normalization was done by dividing the daily traded volume in the market by the maximum traded volume in a single day during that period. During 120 days, representing approximately 8.0% of the days, the traded volume did not reach 0.05% of the total peak traded volume. In 401 days, accounting for 26.8% of the total days, the traded volume did not reach 0.5%.

Figure 2.3 shows an extreme example for illustrative purposes. For the three months period from 5 January 2011 to 31 March 2011, there were four days in which there appears an ending price level for the index but there is no recorded traded volume. It should be noted that this period was not included in the analysis, and that it is shown only as an example. For the majority of the other indexes the issues of stale prices and data availability were not as apparent as in the case of Namibia.

2.3.2 Methods

In this chapter equity indexes representative of countries that are classic examples of markets with different levels of narrowness are considered. This will range from extremely deep markets such as the U.S., represented with the Dow Jones index, to very narrow markets such as the Tanzanian case. These ten countries were grouped into four categories according to the perceived narrowness of its equity capital market. Those four categories were:

1. Very deep
2. Deep
3. Moderately narrow
4. Narrow

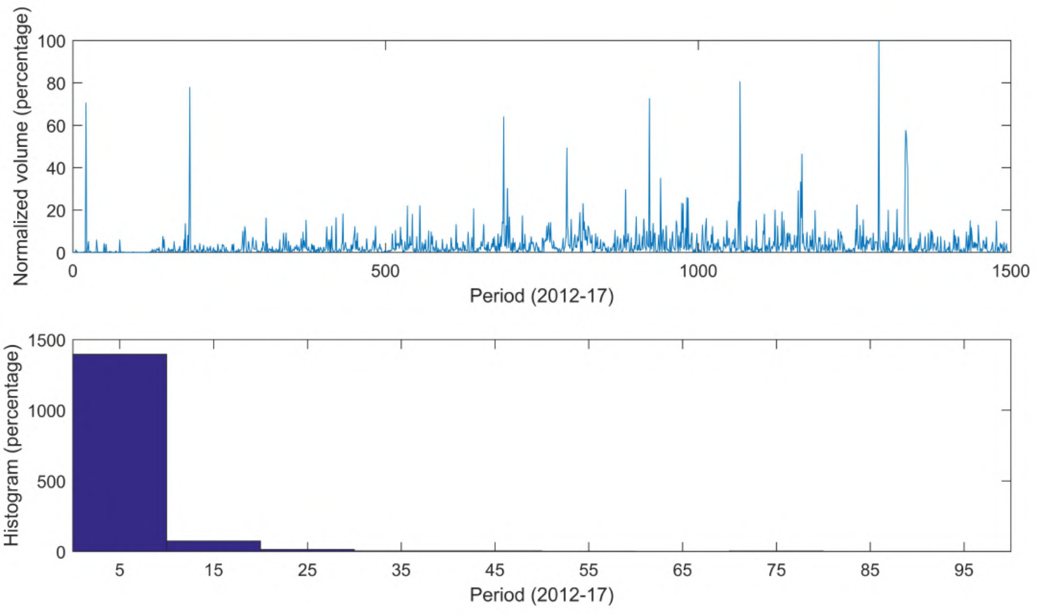


Figure 2.2: Daily traded volume for the Namibian index (2012 to 2017).

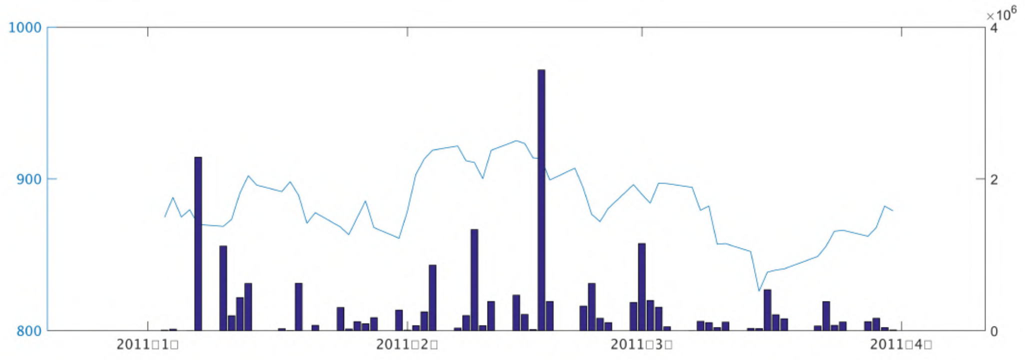


Figure 2.3: Traded volume and index value for the Namibian index (5 January to 31 March 2011).

The US equity market is so large and deep that it is really on a category of its own and was the only country selected for the very deep category. The Dow Jones index was selected as a representative index for the US equity market. The deep category is composed of the FTSE (UK), DAX (Germany) and CAC (France) indexes. The moderately narrow category contains the CSI (China), IBEX (Spain) and RIGSE (Latvia) indexes. These are vastly different markets. The Chinese equity market is one of the largest in the world with large daily trading volumes, but it is typically associated with narrow markets because of the high proportion of retail (individual) investors compared to other markets in which institutional investors account for the bulk of the trading volume. The Chinese equity market is a liquid market, but it is likely not very efficient from a pricing point of view due to this large proportion of individual investors. The market indexes for Tunisia, Tanzania and Namibia were included in the narrow category. A list of all the indexes is showed in Table 2.1.

Table 2.1: Country equity indexes

Indexes	Country	Abbreviation
Very deep market		
Dow Jones Industrial Average	U.S.	DJ
Deep market		
FTSE 100 Index	U.K.	FTSE
Deustche Bourse DAX	Germany	DAX
CAC 40 Index	France	CAC
Moderately narrow market		
IBEX 35	Spain	IBEX
CSI 300	China	CSI
OMX Rige	Latvia	RIGSE
Narrow market		
Tunisia Stock Exchange Index	Tunisia	Tusise
FTSE Namibia	Namibia	Namibia
Tanzania All Share Index	Tanzania	Tanzania

When forecasting stocks or equity indexes one of the most important factors, together with the chosen algorithm, is deciding what input to use. In this case it was decided to use several moving averages. Moving averages are some of the most frequently used indicators for stock performance [Adrian, 2015, Johnston et al., 1999, Gencay and Stengos, 1998, Larson, 2007, Raudys et al., 2013] and they are easily obtained. The moving average at any given time is just the average price over a predetermined number of previous days, that is,

$$MA_N(t) = \frac{1}{N}\sum_{i=0}^{N-1} p(t-i) \tag{2.12}$$

where $p(t)$ is the stock price at time t. Thus, the 50 day moving average will be the sum of the closing prices over the last 50 days divided by 50. In Figure 2.4 an example for the RIGSE index and its 50-day, 100-day and 200-day moving averages can be seen. The neural network inputs used are the 250 most recent values of the moving average. The target used in the training of the network is the price to be forecasted, thus the output of the network is the price forecast, i.e.,

$$\hat{p}(t+1) = \mathrm{NN}(MA_N(t), MA_N(t-1), \ldots, MA_N(t-249))$$

On the other hand, as the forecasting accuracy will depend on the structure of the neural network, a relatively large amount of configurations will be tested, including ten different learning algorithms as well as varying number of neurons.

The forecasting capability of neural networks for all the previously mentioned ten indexes, representing very deep, deep, moderately narrow and narrow markets were estimated. Daily closing prices of the indexes for the period from 2012 to 2017 were obtained from Bloomberg. Also different moving averages will be considered in the experiments, thus the 50, 100 and 200 day moving averages were also computed for each of

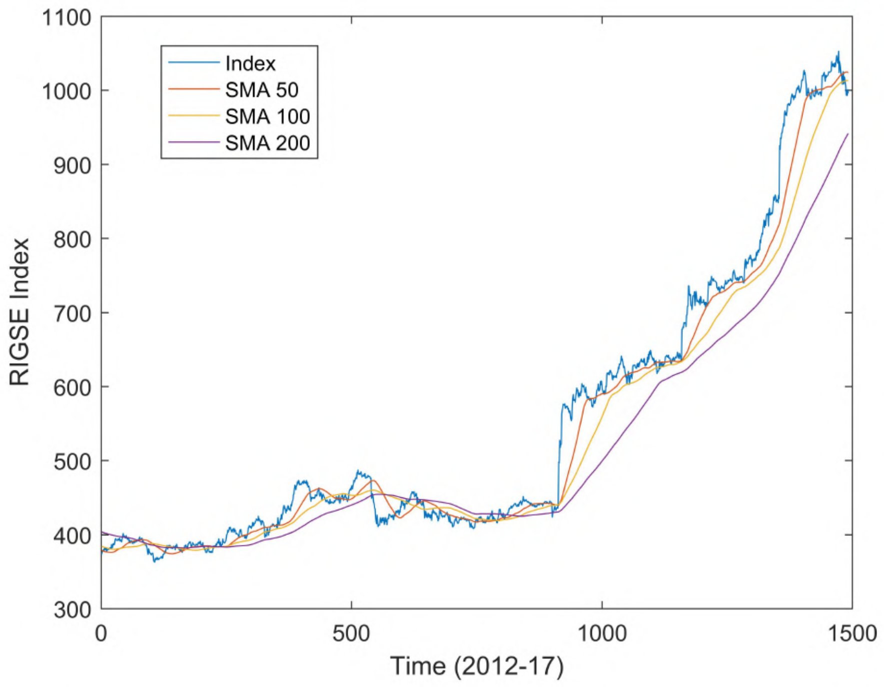


Figure 2.4: RIGSE index and moving averages (stock moving average (SMA)).

the previously mentioned indexes. The structure followed for the neural network consisted of one hidden layer. The amount of neurons was increased from 10 to 100 in steps of 5. Then, the network was created using an input of one of the moving averages and the index as the target value. The process was repeated for all three moving averages (50-day, 100-day and 200-day). Due to different randomly generated initial conditions two neural networks with the same inputs, output and structure are likely to generate slightly different results (forecasts). In order to account for that 100 networks were estimated for each configuration. Each of these forecasts was regressed against the actual target. In this way a probability distribution for the R-square value was obtained. The standard procedure of setting aside 15% of the data for testing purposes was followed. It will be shown that increasing the number of neurons, for most cases, did not increase the accuracy of the forecasts. The opposite was actually true in many cases with accuracy gradually declining. A typical outcome can be seen in Figure 2.5.

In total 10 learning algorithms were used, see Table 2.2, as a forecasting tool for the previously mentioned ten country stock indexes. The moving average was used as an input for the neural network (the process was repeated for all the three moving averages considered). As previously mentioned, 15% of the data was designated as testing data and therefore not used during the training period. The R-squared values showed in the results are the R-squared obtained when regressing the actual value with the estimated generated for the testing data set. This is the standard procedure to try to ensure that the trained neural network generalizes reasonably well when faced with new data, which is a critical step for neural networks.

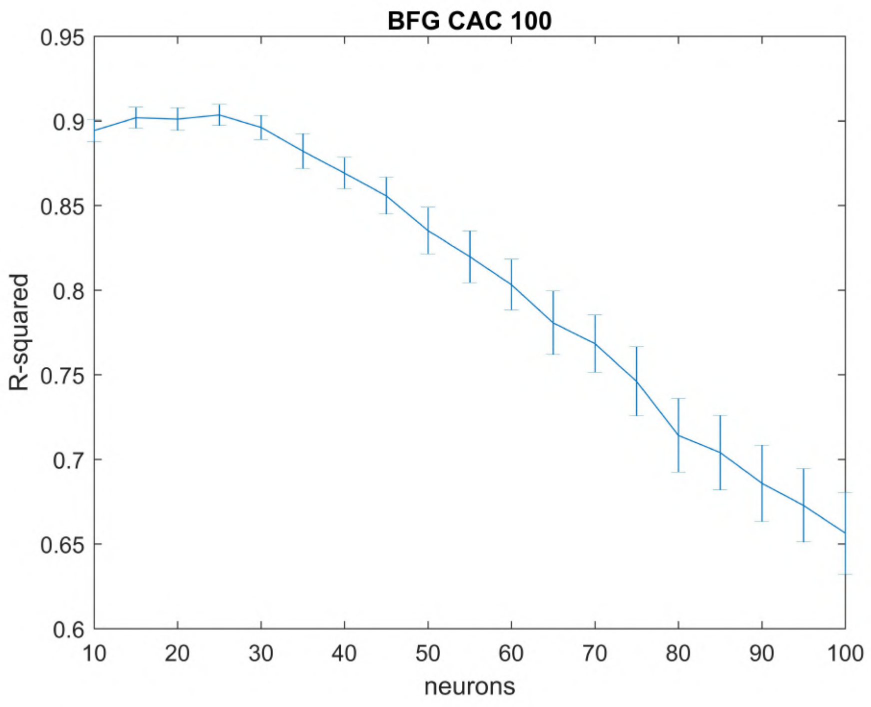


Figure 2.5: Results for CAC index (France) using the 100-day moving average (MA) and quasi Newton (BFG) training algorithm. After a certain point, around 30 neurons, the forecasting accuracy, measured as the R-squared of the regression between the actual and forecasted values, decreases as the number of neurons increases.

Choosing the appropriate learning algorithm for the neural network is of clear importance but it will be showed that there appear to be some general trends for all the ten learning algorithms analyzed, supporting the hypothesis that neural networks are actually an appropriate forecasting tool for stock prices in narrow markets. Calculating 100 networks per index and per configuration allowed for the estimation of confidence intervals. This was done in order to avoid having results that are relatively accurate but that can be obtained because a one off or relatively unlikely events.

As an example, in Figure 2.6 it is shown the forecasting accuracy measured as the R-squared of the regression between the actual and forecasted values, using the OSS training algorithm, for all the indexes increasing the number of neurons and using the 50-day, 100-day and 200-day moving averages, which are denoted in the figures as MA50, MA100 and MA200. This analysis was carried out for all the previously mentioned 10 training algorithms. The results for all the other training algorithms can be seen in figures A.1 to A.9 (see appendix A).

It could be also useful to compare how, given a specific configuration and choice of moving average, would the multiple neural network perform for each country, rather than just comparing several networks and moving averages for the same country index. In Figure 2.7 an example of this approach can be seen. This figure showed the accuracy of forecasts using the BFG training algorithm for all the indexes analyzed. The results for all other training algorithms can be seen in figures A.10 to A.18.

Table 2.2: Training algorithm and standard Matlab abbreviation

Abbreviation	**Algorithm**
BFG	Quasi Newton training algorithm
CGB	Conjugate Gradient (with restarts)
CGF	Conjugate Gradient Fetcher Powell
CGP	Conjugate Gradient Polak Ribiere
DA	Gradient Descent (adaptive learning)
DM	Gradient Descent (momentum)
DX	Gradient Descent (momentum and adaptive learning)
LM	Levenberg Marquardt
RP	Resilient backpropagation
OSS	Secant training algorithm

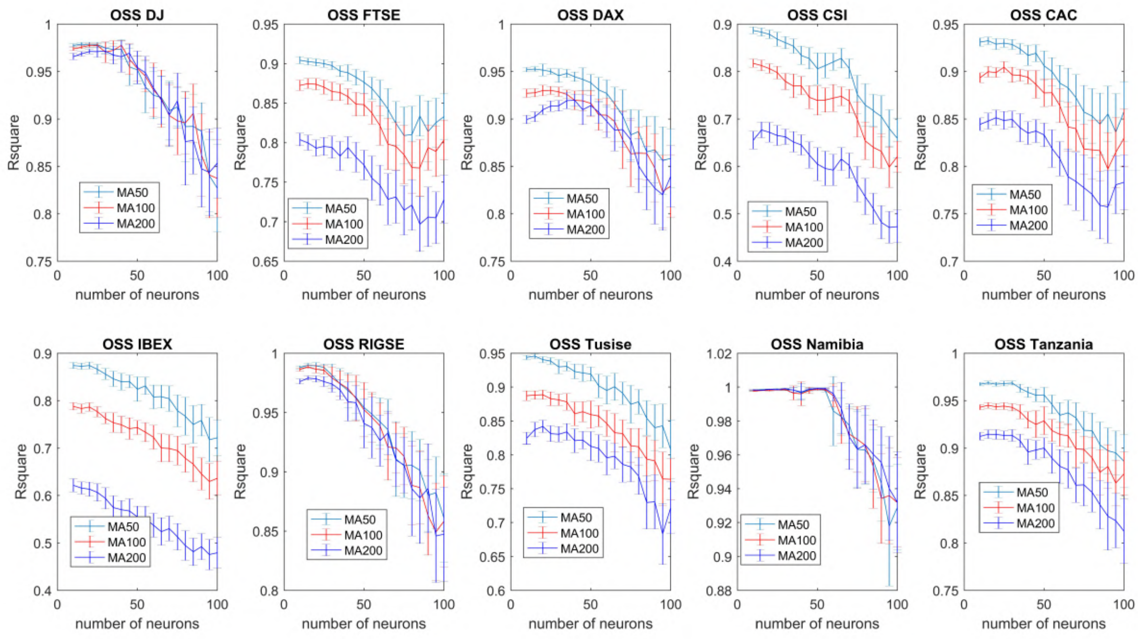


Figure 2.6: Forecasting accuracy comparison of different moving averages using the Secant training algorithm (OSS). After a certain number of neurons, and regardless of the index analyzed, the forecasting accuracy of the algorithms decreases when additional neurons are added.

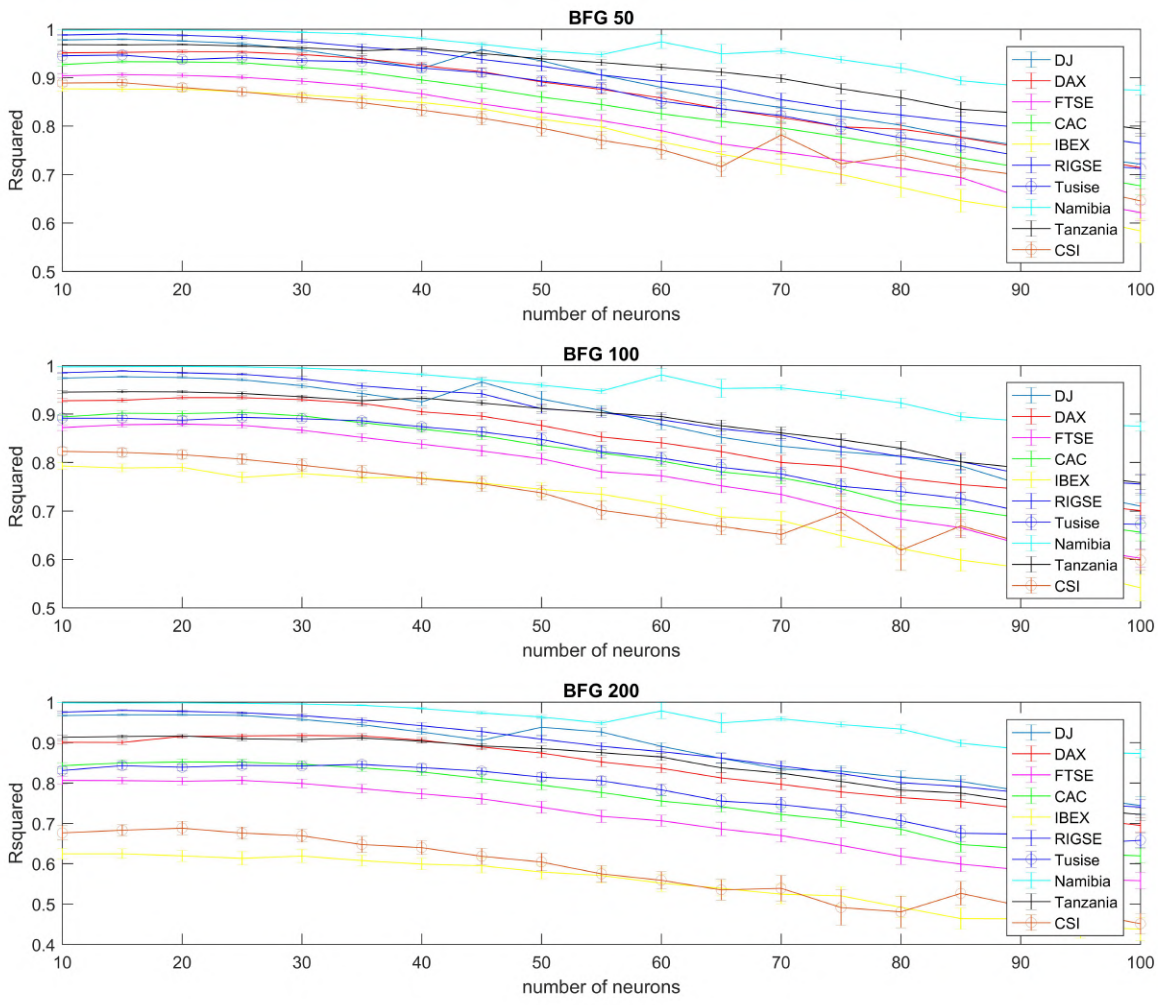


Figure 2.7: Results using BFG training per country (using the 50, 100 and 200 days moving average).

2.3.3 Results

Of the models analyzed the best results, from a forecasting accuracy point of view, were those using the 50-day moving average and a relatively small number of neurons, typically 10. Increasing the number of neurons in the network did not increase the forecasting accuracy. This result was relatively consistent among most of the indexes, regardless if they belong to very deep, deep, moderately narrow or narrow categories, as well as across most of the training algorithms analyzed. It is noteworthy that the Namibian case appears to be the one with the best forecasting accuracy regardless of the choice of moving average. The results in the Namibian case, a particularly narrow market, should be taken with caution and present some unique characteristics. The forecasts for the Namibian case appear to be remarkably accurate, but this could be related to the stale price issue and the accuracy hence overstated. The result in this way could appear to be very accurate but a practical application of such a model for trading purposes could lead to disappointing results if the price quoted does not match the price at which a transaction can be actually done. It is also interesting to notice that in the case of Namibia, using the Levenberg Marquardt method the forecasting accuracy appears to increase as the number of neurons increases, which is not the case for the vast majority of other indexes and configurations. In this case there also appears to be no statistically significant difference between using the 50-day, 100-day or 200-day moving averages, which is again in direct contrast with the results from most of the other indexes.

The case of Tanzania and Spain (IBEX) are examples of typical results with the 50-day moving average generating better results than any of the other indexes analyzed and forecasting accuracy gradually decreasing as the number of neurons increases. It should also be noted that the forecasting accuracy of the Dow Jones index, using the same configuration, did marginally increase when the number of neurons were increased. Another peculiarity showed by the Dow Jones index is that the selection of the moving average (50-day, 100-day or 200-day) did not have a significant impact on the forecasting accuracy for most of the configurations analyzed. On the other hand, in some other cases like the Spanish (IBEX), there is a significant difference between using one moving average or the other. This would translate in differences in the forecasting error. As an example, the RMSE values for the case of the IBEX and the Dow Jones are showed in table 2.3.

Table 2.3: RMSE for the IBEX and the Dow Jones using a neural network with 20 neurons. The RMSE values showed are the average of 100 simulations for each case.

Technical indicator	**IBEX**	**Dow Jones**
Moving Average (50 days)	422.9	363.1
Moving Average (100 days)	576.9	385.6
Moving Average (200 days)	775.0	399.2

In this table it can be seen that there is a relatively large difference of the RMSE obtained using the IBEX index for the various moving averages while in the case of the Dow Jones the values of the RMSE appear to be relatively close, regardless of the moving average used. These different behaviours highlight the importance of input variable selection, and that forecasting performance is affected in a case dependant way. While the forecasting accuracy is good for all the indexes, regardless if it is a narrow market or not, there are some apparent differences with for instance moderately narrow markets, particularly in the case of the Spanish (IBEX) and Chinese (CSI) generating for most of the configurations analyzed slightly poorer forecasts than in the case of the other indexes. The difference appears to be particularly large in the case of the gradient descent with adaptive learning.

In general terms the accuracy of the training algorithms was comparable. One of the noticeable exceptions was the gradient descent momentum that generated the worst results among all the training algorithms used. These poor results were obtained for all the 10 indexes and regardless of the number of neurons used. All the other training algorithms provided comparable results for the indexes regardless on the classification of the index. All the narrow and moderately narrow markets analyzed, with the previously mentioned caveat for the Namibian market, generated forecasts that are comparable to deep and very deep markets.

2.3.4 Discussion

Neural networks appear to be an applicable tool for stock forecasting purposes on narrow markets with performance that is comparable but typically lower than in deeper markets. Forecasts in some particularly narrow markets might appear to be very accurate but that could be related to stale prices. This phenomenon appear when the quoted price is not representative of an actual transaction on the analyzed day but of some transaction on a previous day. This is typically associated with illiquid markets. Besides, for this type of extreme case, it would appear that neural networks do a relatively good job forecasting stock performance in the countries analyzed. The 50-day moving average provided results that were at least statistically not worse than the 100-day or 200-day moving averages for most of the neural network configurations analyzed. In fact, the 50-day moving average was the best choice in some of the indexes, like the CSI.

For other, deeper, markets, such as the U.S. market, there appears to be less statistically significant differences between these different moving averages regarding forecasting accuracy. It should also be noted that increasing the number of neurons did, in most cases, not only not increase forecasting accuracy but it decreased it. This was a general trend observed when using virtually all of the training algorithms with basically all the ten stock indexes analyzed. This might be related to the issue of local minima in neural networks. As the number of neurons increases the neural network might get stuck in a local minimum, basically losing generalization power. Therefore, an important takeaway is that naively increasing the level of complexity of a neural networks, by adding large amounts of neurons to the network, is not likely to translate into more accurate stock forecasts.

The fact that neural networks appear to be applicable for stock forecasting in narrow markets suggest that while there are clearly very big differences between narrow and deep markets they might also share some features that allow the successful use of the same forecasting technique, such as neural networks, in both types of markets. As previously mentioned the flow of information is likely very different in some of the narrow markets analyzed, with relatively poor telecommunication and trading infrastructure, compared to countries such as the United States, but, interestingly, it appears that regardless of these obvious differences neural networks have comparable levels of applicability, for stock forecasting purposes, in narrow and deep markets.

2.4 Forecasting the Indian Stock Market

The Indian stock market is experiencing fast growth and has peculiarities that differentiate it from many other stock markets, particularly those in developed markets. Given the potential size of this stock market and the growing importance of the Indian economy it seemed reasonable to look at what tools provide reasonably accurately stock prices forecast. It is of clear practical and theoretical relevance to determine which type of algorithm to use in order to try to forecast stock market trends. In recent times there has been an increasing interest in applying neural networks to such aim.

In this section it is shown that a neural network using as an input the closing price in the previous day and two different learning algorithms produce accurate forecast. This simple but efficient approach generated a one day forecasting accuracy, measured as the mean value of R^2, of [0.9952, 0.9972] for the Levenberg-Marquardt training algorithm and [0.9965, 0.9970] for the Scaled Conjugate learning algorithm. These results were obtained using only 10 neurons in the neural network. Increasing the number of neurons to 500 did not improve the out of sample forecasting accuracy. It actually decreased, for both algorithms. It would seem that at least for the period of time analyzed, from 2010 to 2016, neural networks did a reasonably good job forecasting stock prices.

In order to forecast share prices it is necessary to decide what variables to use as inputs as well as the forecasting technique to use. In this case it was decided to use only historical values, rather than more complex indicators derived from historical values. On the other hand, neural networks with one hidden layer and two different learning algorithms, Levenberg-Marquardt and Scaled Conjugate Gradient, have been chosen as the forecasting tool.

2.4.1 Methodology

The NSE Nifty 50 index is one of the most frequently quoted stock indexes in India and it is composed by the top 50 companies by market capitalization in the Indian stock market. The index is owned and maintained by India Index Services Products Ltd. and according to their data represent approximately 63% of the total free float market capitalization of the National Stock Exchange of India (NSE). All the share prices were obtained from the data provider Bloomberg Professional. The data consist of closing daily prices for the NSE nifty 50 for the period from 2010 to 2016.

The closing price of the index in the previous 250 days was used as the input for forecasting the closing price in the following day, i.e.,

$$\hat{p}(t+1) = NN(p(t), p(t-1), \dots, p(t-249))$$

As previously mentioned the LM and Scaled Conjugate gradient were used as the learning algorithms. The neural network built had one hidden layer. Initially the number of neurons was set to 10. The number of neurons was then increase, in steps of initially 10 neurons, until reaching 100 neurons. From that point onwards the number of neurons were increased in steps of 50 neurons until reaching total of 500 in an attempt to determine the effect of such neural network parameter in the out of sample forecasting accuracy.

The forecasting accuracy for each neural network (learning algorithm plus number of neurons) was tested in a series of 100 tests. The objective of this process was to estimate an average value and a 95% confidence interval for the mean value of R^2 as well as of its volatility. As prediction is the main objective of this analysis the values R^2 were calculated for the out of sample data, i.e., the data not previously seen by the neural network. All the calculations were performed in Matlab using the neural network toolbox.

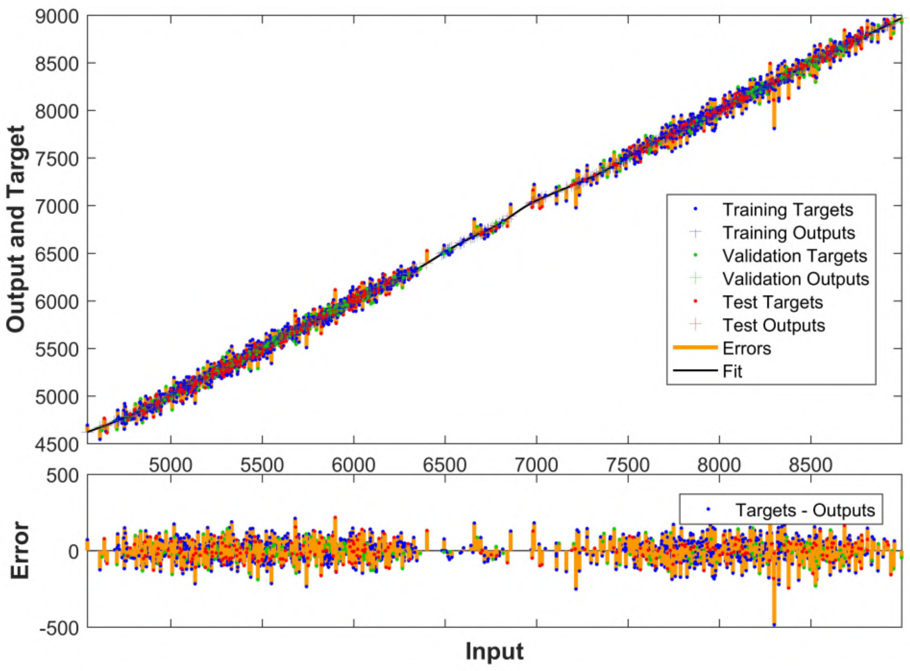


Figure 2.8: Output from NN using LM algorithm

2.4.2 Results

The analysis was performed using the NSE Nifty 50 index. The results support the idea that neural networks are applicable to the Indian stock market, for the period from 2010 to 2016, for forecasting purposes. At a 95% confidence interval there was no statistically significant difference between the forecasting accuracy of a neural network with 10 neurons using the LM training algorithm and the Scaled Conjugate algorithm (table 2.4 and 2.5) using the mean value of R^2 as the metric for comparisons. Always as the same confidence level, the standard deviation for the mean R^2 was statistically significantly lower for the Scaled Conjugate approach than for the Levenberg-Marquardt approach. Visually, the goodness of the model can be appreciated in figures 2.8 and 2.9. The histogram of the error can be seen in figures 2.10 and 2.11.

As previously seen for other markets, increasing the number of neurons by an order of magnitude did not help improving out-of-sample forecasting accuracy. In fact, for both the Scaled Conjugate and the LM approach the mean value, at a 95% confidence level, for the mean value of R^2 was lower when using 100 neurons than when using 10 neurons. Similarly, the value for the standard deviation of the mean value of R^2 was higher for the case with 100 neurons when compared to the 10 neurons approach for both learning

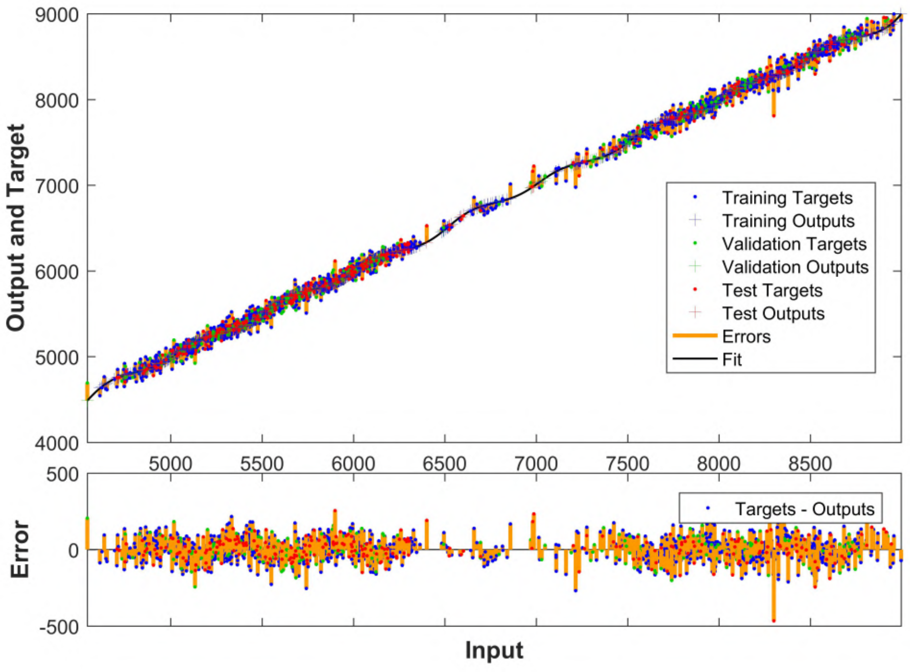


Figure 2.9: Output from NN using scaled conjugate algorithm

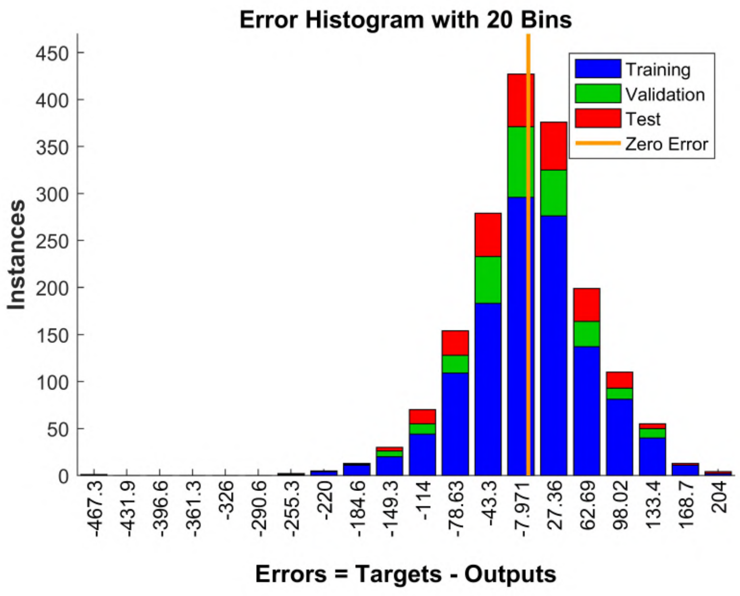


Figure 2.10: Error histogram from NN using LM algorithm

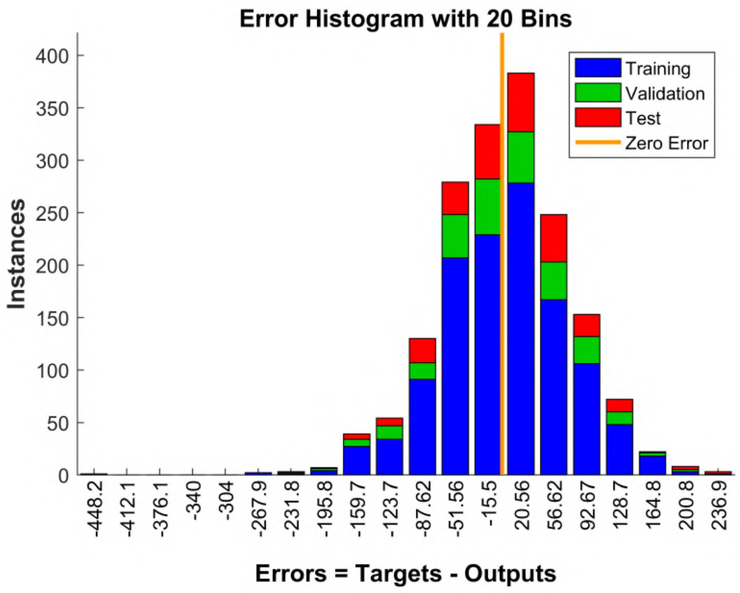


Figure 2.11: Error histogram from NN using scaled conjugate algorithm

algorithms. The forecasting accuracy of neural networks in the Indian market, during the period analyzed, is remarkable. It is possible that as neural networks become more popular in the future and more investors use them their value as an investment tool might decline but at the present stage there seems to be a useful tool[3]. It should also be noticed that trading costs and liquidity considerations were not included in this analysis and they are an area of future possible work.

The sensitivity of the approach was also studied increasing the number of layers. The number of layers was increased from 1 to 50 with each layer containing 10 neurons. In figure 2.12 it can be seen that the after certain number of layers the RMSE increases substantially. This is most likely related to the issue of overfitting. Due to the rapidly increasing magnitude of the RMS the first section of figure 2.12 (roughly from 1 to 12 layers) appears roughly flat while in fact there was a clear tendency for the error to increase as the number of layers increased. In order to show this visually a zoom of the first section of the graph was also presented. In this zoom it can be clearly be seen that the RMSE appears to increase as the number of layers increases.

It is also important to consider the computational time required for each of the different neural networks analyzed. In figure 2.13 it can be seen the time required for the neural network to generate estimations for the Nifty 50 index as the number of layers is increased. As expected, the general trend is for the time to increase as the number of layers increases.

A characteristic of neural networks is that they are initialized with a random set of initial weights. This random initialization can have a very significant impact on the training process as well as on the accuracy of the forecasts. For instance it is likely that, by chance, one random weight initialization might lead to a better (more accurate) forecast)

[3]From market efficiency considerations, as the number of investors that use a forecasting tool rises, the tool's capability for exploiting the market inefficiencies decreases.

Table 2.4: R^2 values using LN training (95% confidence)

N. of neurons	**mean** R^2	σ R^2
10	[0.9952, 0.9972]	[0.0043, 0.0056]
20	[0.9961, 0.9988]	[0.0058, 0.0076]
30	[0.9944, 0.9978]	[0.0073, 0.0095]
40	[0.9821, 0.9950]	[0.0277, 0.0361]
50	[0.9700, 0.9830]	[0.0280, 0.0364]
60	[0.9648, 0.9689]	[0.0088, 0.0115]
70	[0.9631, 0.9695]	[0.0138, 0.0179]
80	[0.9945, 0.9777]	[0.0714, 0.0930]
90	[0.9345, 0.9709]	[0.0783, 0.1019]
100	[0.9233, 0.9779]	[0.0783, 0.1019]
150	[0.9256, 0.9781]	[0.1129, 0.1470]
200	[0.9178, 0.9832]	[0.1406, 0.1831]
250	[0.9211, 0.9648]	[0.0940, 0.1224]
300	[0.9137, 0.9678]	[0.1163, 0.1515]
350	[0.8801, 0.9457]	[0.1410, 0.1837]
400	[0.8715, 0.9356]	[0.1378, 0.1795]
450	[0.8706, 0.9224]	[0.1140, 0.1450]
500	[0.8679, 0.9314]	[0.1365, 0.1778]

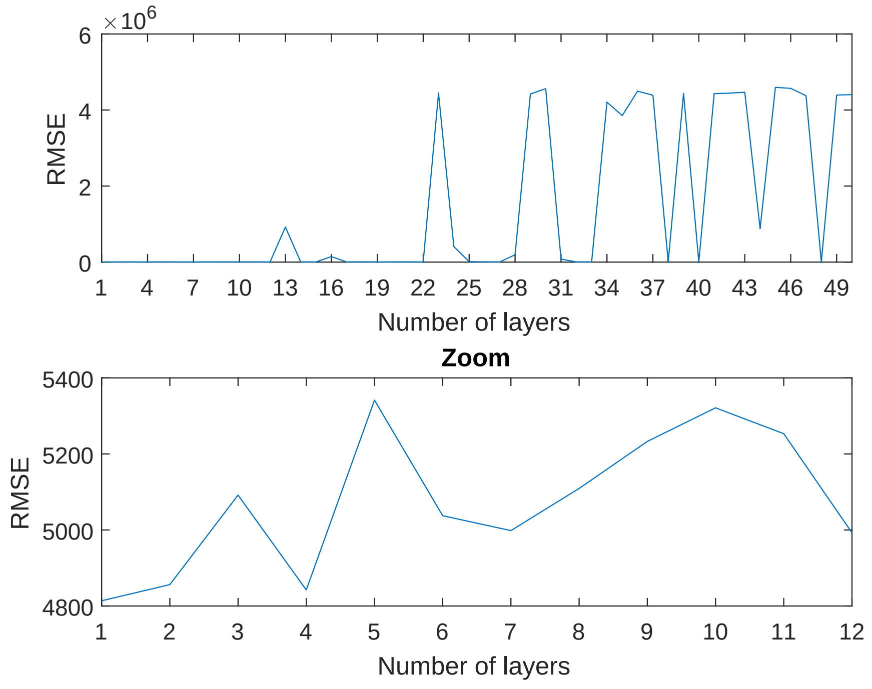


Figure 2.12: Output error from NN using scaled conjugate algorithm (includes a zoom of the results from 1 to 12 layers)

Table 2.5: R^2 values using Scaled Conjugate training (95% confidence)

N. of neurons	**mean R^2**	σ R^2
10	[0.9965, 0.9970]	[0.0010, 0.0014]
20	[0.9974, 0.9982]	[0.0016, 0.0022]
30	[0.9921, 0.9961]	[0.0080, 0.0112]
40	[0.9903, 0.9945]	[0.0084, 0.0118]
50	[0.9888, 0.9917]	[0.0058, 0.0081]
60	[0.9838, 0.9912]	[0.0148, 0.0207]
70	[0.9822, 0.9856]	[0.0068, 0.0095]
80	[0.9850, 0.9901]	[0.0102, 0.0143]
90	[0.9816, 0.9899]	[0.0166, 0.0232]
100	[0.9845, 0.9890]	[0.0098, 0.0130]
150	[0.9753, 0.9881]	[0.0256, 0.0359]
200	[0.9777, 0.9854]	[0.0154, 0.0216]
250	[0.9759, 0.9833]	[0.0148, 0.0207]
300	[0.9619, 0.9845]	[0.0452, 0.0633]
350	[0.9532, 0.9732]	[0.0400, 0.0560]
400	[0.9552, 0.9711]	[0.0318, 0.0445]
450	[0.9555, 0.9728]	[0.0346, 0.0484]
500	[0.9432, 0.9632]	[0.0400, 0.0560]

just by random luck. It is also likely that some random initialization might lead to less accurate forecasts as well as more computational time. This random behavior can be seen in figure 2.13. While the trend is clear, suggesting that the computational time increases as layers are added, there is also a random component that generate for some neural network architecture particularly high computational time (higher than for neural networks with higher number of layers).

For this application it would appear that increasing the number of layers does not only increase the required computational time but it also does not improve the accuracy of the forecast. Furthermore, it also appears that the accuracy of the forecast decreases as the number of layers increases. It is likely that this is related to the issue of overfitting. Overfitting might occurs when there is an excessive large amount of neurons or an excessively complex neural network which is unable to properly extrapolate the model generated using the training data when faced with new (unseen) testing data.

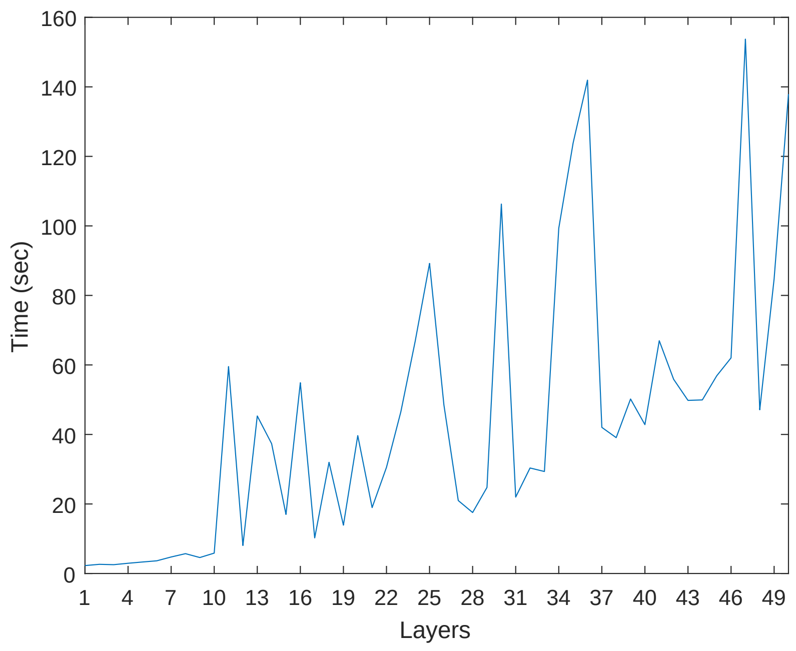


Figure 2.13: Neural network training time vs. number of layers

Chapter 3

A technical indicator selection approach with application to the Chinese stock market

This chapter is based on the following journal publication:

- Alfonso, G.; Ramirez, D.R. A Nonlinear Technical Indicator Selection Approach for Stock Markets. Application to the Chinese Stock Market. Mathematics 2020, 8, 1301.

3.1 Introduction

Trend indicators are quantitative time series, typically related with historical prices or trading volumes, which are used as a tool for forecasting future stock prices. The basic idea of using technical trend indicators is investing in a systematic way based on some quantitative rules. This is a rather different approach from traditional investment. In traditional stock investment the investor picks a company which he/she thinks that have strong fundamentals and it is going to outperform. This typically involves longer investment time horizon as the stock prices moves to some value that the investors considers as the objectively true value of the stock [Wafi et al., 2015]. Such assumptions are not made in technical trading. Technical trading is based on finding typically short term trends in the stock market. This is hence typically related to short term, rather than long term, investment. Some scholars, such as [Lo et al., 2000, Bettman et al., 2009, Wong et al., 2003, Chong et al., 2014] have found that technical indicators provide investment value but there seems to be no strong consensus. Over the years investors have found a large amount of indicators ranging from simple moving averages [Chitra, 2011] over a specific period of time to more exotic and complex indicators [Khaidem et al., 2016]. Some of the common features found in these indicators are that they tend to use historical prices as well as in some cases historical trading volumes. The implied assumption is that, at least to some degree, historical performance can be used to forecast future stock prices.

It should be noted that, as mentioned in the context of the market efficiency hypothesis, there is not a wide consensus in agreeing with such an assumption.

Over the last few decades there has been an increase in non-linear forecasting techniques, such as for instance neural networks. The non-linearity of the stock market performance has been mentioned by many researchers such as for instance [Vrbka and Rowland, 2017]. In fact, the non-linearity of the stock market was one of the reasons behind choosing neural networks (a non-linear approach) by [Vrbka and Rowland, 2017] as a model to forecast stocks prices in Prague Stock Exchange. In this paper the authors successfully applied multilayer perceptron and radial basis networks to that particular stock market. The nonlinearity of these models makes the choice of the input variables an even more important question than in the case of linear models. Thus, the impact of the selection of technical indicators in the forecasting performance is something that must be considered.

An important development in recent years is the very large increase in data available in many disciplines. It is relatively frequent to have a high number of variables that can potentially have an impact in non-linear processes, see [Guyon and Elisseeff, 2003]. There is substantial research covering the topic of variable selection using linear methods, such as for instance [Hocking, 1976], but these techniques might not be ideal when intended for non-linear modelling. There is relatively little literature covering the issue of variable selection of non-linear processes from a combinatorial approach. The basic assumption is that in non-linear processes the way in which different independent variables interact with each other can be highly complex. Identifying which combination of variables work better for a non-linear problem is clearly not trivial, see [Yuan and Lin, 2006]. There are some interesting methods, such as for instance [Rech et al., 2001], using polynomial approximation. The drawback of this type of approach is that it is only applicable when there is a relatively small number of variables. On the other hand, [Ye and Sun, 2018] proposed an iterative method in which starting from all the variables considered one, of the variables is dropped and the resulting set of variables is used, using neural networks, and then the results compared.

The stock market, as many other fields, has seen a large increase in the number of data available. More specifically, many researchers and practitioners have developed large amount of technical indicators. Getting into the details of these indicators is beyond the scope of this chapter but it is important to mention that they are typically constructed using historical data such as for instance the closing price of a stock. A moving average is a well-known technical indicator [Qin et al., 2014, Glabadanidis, 2015, Bruni, 2017, Stanković et al., 2015, de Souza et al., 2018, Lin, 2018, Wang and Kim, 2018]. A simple moving average can be constructed as the average of the closing price of a certain stock or index over a certain period of time. There are multiple different technical indicators and strategies based on those indicators with diverse levels of profitability [Brown and Jennings, 1989, Gencay, 1998, Bokhari et al., 2005, Park and Irwin, 2007].

The issue of technical indicator selection is immediately posed when facing the task of finding a good strategy for stock price forecasting. As stated in chapter 2, neural networks are a very popular tool for such task. One of the frequently mentioned drawbacks of neural networks is the issue of local minima [Baldi and Hornik, 1989, Gori and Tesi, 1992, Bianchini et al., 1994, Yu and Chen, 1995] which can cause the neural network to generalize poorly or in other words, generate poor forecast when faced with new data. In this regard there has been a focus on reducing the dimensionality of the data to avoid having the neural network stuck in this local minima [Hinton and Salakhutdinov, 2006, Wang et al., 2012a, Becker et al., 2020, Kiarashinejad et al., 2020, Breger et al., 2017]. A strategy focused on finding the combination of technical indicators with the best forecasting accuracy would be very useful in this context.

In this chapter it is proposed a combinatorial approach for variable selection applied to stock market technical indicators. Given the number of possible technical indicators that are available today, is clear that not all possible combinations can be tested. A pool of 35 technical indicators is used, so the number of possible combinations is staggering. Because of that the proposed approach resorts to randomization. The algorithm starts by generating a preset number of combinations each with a size of half the number of available indicators. The reason for using such size is because the maximum number of combinations for a given n and k, that is the binomial coefficient n choose k, is attained for $k = \lfloor \frac{n}{2} \rfloor$. This is evident by inspecting the Pascal's Triangle, but can be proved using the properties of the Newton binomial. From this initial set of combinations, new combinations are generated randomly retaining the best ones in an iterative process that ends when the stop condition is met.

The proposed method is shown to improve the baseline approach, that is using all the available indicators, when applied to the task of predicting market trends. As a case study for the proposed method, this chapter focuses on the Chinese stock market. China stock market is a very dynamic and of increasingly importance due to the economic grow of China. The most relevant Chinese stock indexes have been considered. The strategy described in this chapter has obtained good results and better combinations of technical indicators have been identified for such indexes.

The remaining of the chapter is organized as follows: Section 3.2 presents the proposed approach. Section 3.3 shows the application of the proposed approach to the China stock market. Finally, the results are discussed in Section 3.5.

3.2 Technical Indicator Selection Approach

Let $X_T^i(t)$ be tuple of T values of the i-th technical indicator from a pool of up to N nonlinear technical indicators computed at the time period t (usually the time period is measured in days, but could be weeks, months or whatever), such as for instance a moving average:

$$X_T^i(t) = \{X^i(t-(T-1)), X^i(t-(T-2)), ..., X^i(t)\}, \tag{3.1}$$

where $X^i(t)$ is the value of the i-th technical indicator evaluated at time t.

Let $R_T(t)$ be a vector that groups the direction of the change of the price of a stock or index, in the period from $t-(T-1)$ to t, that is $R_T = \{0,1\}^T$ with 0 meaning that the stock increases or remains constant at the end of a period, 1 meaning otherwise.

Additionally, let $\hat{\phi}$ define a nonlinear mapping from $X_T^i(t)$ to $R_T(t)$

$$\hat{\phi}(X_T^1(t), X_T^2(t), \ldots, X_T^N(t)) = \hat{R}_T(t), \tag{3.2}$$

where $\hat{R}_T(t)$ is an estimation of $R_T(t)$. In order to guarantee that an estimator $\hat{\phi}$ can be found is necessary to assume the following:

Assumption 2 (Ground truth function existence)**.** *It is assumed that, ϕ, a mapping from $X_T^i(t)$ to $R_T(t)$ exists.*

Technical indicators in the stock market can produce contradictory signals and some non-linear techniques can have local minima issues when using high dimensionality input variables, (large N or T value). Therefore it might be convenient to find a combination of the technical indicators X_T^i rather than using all N available indicators. A combinatorial selection approach for technical indicators, which do not have to be linear variable combination in non-linear processes, is presented.

The steps are as follows:

1. Split the available instances of $X_T^i(t)$ and $R_T(t)$ data into two subsets, an estimation subset $S_e \triangleq \{X_{T,e}^i(t), R_{T,e}(t)\}$ and a validation[1] subset $S_v \triangleq \{X_{T,v}^i(t), R_{T,v}(t)\}$.
2. Generate C_s, a set of M combinations of $\lfloor \frac{N}{2} \rfloor$ random numbers in the range $\{1, 2, \ldots, N\}$ denoted from i_1 to $i_{\lfloor \frac{N}{2} \rfloor}$. Check for repetition within each combination. If there is repetition, leave out the repeated numbers and keep generating random numbers in the aforementioned range until there is no repetition. Similarly, check that there are no repeated combinations and proceed to replace the redundant ones.
3. For each of the M combinations in C_s, denoted as $I \in C_s$, perform the following steps:
   (a) Compute $\hat{\phi}$, i.e., the estimated non-linear classification mapping, using any technique of choice. In this case, the nonlinear mapping will have as arguments $X_T^i(t), \forall i \in I$. Using neural networks as an example, this step involves training a neural network with the training data set $\{X_{T,e}^i(t), R_{T,e}(t)\}$ with $i \in I$.
   (b) Evaluate $\hat{\phi}$ over the validation set. Let $\hat{R}_{T,v}$ denote the estimated classification output of $R_{T,v}$.
   (c) Calculate the error $\xi_T(t)$ of the non-linear classification approach for each instance in the validation set, so that

$$\xi_T(t) = \begin{cases} 0 & \text{if } \hat{R}_{T,v}(t) = R_{T,v}(t), \\ 1 & \text{Otherwise} \end{cases}$$

   with the resulting total error being:

$$\xi^{Total} = \frac{\sum_{\forall t \in S_v} \xi_T(t)}{\text{card}(S_v)}, \tag{3.3}$$

[1]To keep the notation as clear as possible here the term validation is used with the meaning of test

where card(S_v) denotes the cardinality of S_v. This is therefore the total error for the initial random combination of technical indicators chosen in step 2.

(d) Randomly generate a single value $i_a \in \{1, 2, ..., N\}$ that would denote a technical indicator candidate to be added to the combination chosen in step 2. Check for repetition with the previously generated $\frac{N}{2}$ values. If there is repetition, randomly generate another value i_a. Repeat this step until there is no repetition.

(e) Randomly generate a single value $i_r \in \{i_1, \ldots, i_{\frac{N}{2}}\}$. This value will be used later to denote a technical indicator to be removed from the combination chosen in step 2.

(f) Form a new combination of technical indicator indexes I^{up} as $I^{up} \triangleq I \cup \{i_a\}$. In the same fashion, form a new combination I^{down} as $I^{down} \triangleq I - \{i_r\}$. If some of these combinations are already in C_s repeat either step 3d or 3e until a combination different from those of C_s is obtained.

(g) As in step 3a, compute two new mappings, denoted $\hat{\phi}^{up}$ and $\hat{\phi}^{down}$ using this case as inputs arguments the technical indicators given by the index combinations I^{up} and I^{down} respectively.

(h) As in step 3c compute the resulting total error of evaluating the two mappings computed in the previous step over the validation set. Denote these total errors as ξ^{Total}_{up} and ξ^{Total}_{down}.

(i) Given the three previous index combinations I, I^{up}, I^{down} and their resulting total errors ξ^{Total},ξ^{Total}_{up} and ξ^{Total}_{down} pick the two combinations with the smallest error.

(j) Substitute the initial combination I by the two combinations picked in the previous step. This will double the number of combinations in set C_s, i.e., C_s will contain $2M$ combinations after this step.

4. Retain the M combinations in C_s with the smallest error and discard the remaining M combinations with greatest error.

5. Find the minimum ξ^{Total} of all the combinations $I \in C_S$.

6. Repeat step 3 starting from 3d until a certain stop condition is met. Here it is proposed to check if a certain objective error is achieved or a maximum number of iterations is met, so that an infinite loop is avoided.

Remark 1. *This strategy can be easily parallelized as the tasks in step 3 can be done independently for each combination, hence can be done in parallel, with one exception. The latter part of step 3f, that is the rejection of repeated combinations, cannot be done in parallel and must be done sequentially in a separate step outside of step 3.*

Remark 2. *The number of combinations in C_s, that is M, and the number of iterations are related in the sense that similar performances can be achieved by using a greater M and fewer iterations or vice versa. Besides the differences in performance due to the randomized nature of the proposed strategy, a practical difference can lead to one choice or the other. With a greater M one can exploit the parallelizable nature of the algorithm, whereas in the case of a high number of iterations that cannot be made.*

To illustrate the proposed approach consider a simple example with just two combinations (i.e., $M = 2$) and one iteration. If for instance, it is assumed that there are a total of 6 technical indicators to choose from, i.e., $\{X_5^1, X_5^2, X_5^3, X_5^4, X_5^5, X_5^6\}$, where the sub-index 5 means that each of the indicators are evaluated over five periods of time. There is also a related classification vector R_5 identifying up and down movements in the stock at each time t (in this case $R_5 \in \{0,1\}^5$). First, the algorithm selects randomly 2 combinations of 3 initial technical indicator indexes. For example:

$$C_s = \{I_1, I_2\} = \{\{3,5,6\},\{1,2,3\}\}$$

Then the non-linear classification error is estimated for this configuration; let us assume that the obtained value is (with a slight abuse of notation the parameters of each combination will denote in this example the set of the parameters of both combinations):

$$\xi^{Total} = \{0.6, 0.5\}$$

Then the indexes i_a and i_r are computed for each combination:

$$i_a = \{1,4\} \;\; i_r = \{6,2\}.$$

The addition of the new technical indicator index is done randomly, ensuring that there is no repetition, i.e., each input variable (technical indicator) is used only once. The index removed is also computed randomly. Then the combinations I^{up} and I^{down} are formed:

$$I^{up} = \{I_1^{up}, I_2^{up}\} = \{\{3,5,6,1\},\{1,2,3,4\}\}$$

$$I^{down} = \{I_1^{down}, I_2^{down}\} = \{\{3,5\},\{1,3\}\}$$

Note that in every combination generated so far no index is repeated and that there are not repeated combinations. Once the new combinations are formed, their total classifying errors are computed:

$$\xi_{up}^{Total} = \{0.7, 0.3\}$$

$$\xi_{down}^{Total} = \{0.4, 0.2\}$$

$$\xi^{Total} = \{0.6, 0.5\}$$

For every combination in C_s two combinations (from I, I^{up} and I^{down}) with the smallest error are chosen. From the first combination, I_1^{down} and I_1 are chosen. On the other hand, from the second combination I_2^{up} and I_2^{down} are chosen. Therefore, the augmented C_s will consists of:

$$C_s = \{I_1^{down}, I_1, I_2^{up}, I_2^{down}\}$$

with their errors:

$$\xi = \{0.4, 0.6, 0.3, 0.2\}$$

The combinations are ordered using the error as sort criterion:

$$C_s = \{\{I_2^{down}, I_2^{up}, I_1^{down}, I_1\}$$

$$\xi = \{0.2, 0.3, 0.4, 0.6\}$$

Now the numbers of combinations can be reduced to its original number ($M = 2$) picking the two with the smallest error:

$$C_s = \{\{I_2^{down}, I_2^{up}\}$$

with errors:

$$\xi = \{0.2, 0.3\}$$

Then at the end of this first iteration the algorithm would pick the combination I_2^{down} as the best technical indicator combination, as it has the lowest error (0.2). Thus, the technical indicators proposed to forecast the direction of the change of price will be $\{X_5^1, X_5^3\}$.

In practice, the number of combinations and iterations will be determined by the computing power available. The process thus will be repeated until a stop criteria is reached (that is, a maximum number of iterations or a target classification error).

3.3 Chinese Equity Market Application

Data and Methodology

The Chinese equity market is an increasingly important market propelled by the large economic expansion of the Chinese economy over the last few decades. The Chinese equity market is divided into two major stock exchanges. The Shanghai and the Shenzhen Stock Exchange with several major stock indexes describing the performance of those markets. It should be noted that there is no overlap between the Shanghai Stock market and the Shenzhen stock market i.e., there are no companies listed in both markets simultaneously. There are two main type of stocks. A-share and B-share. A-share is the main type of stocks while B-share are stocks that were originally created to allow foreign investors participate in the China onshore equity market. There hasn't been any new IPO in the B-share in recent years and it is unlikely that there will be any in the future as it is a market been phased out. Therefore, it seem reasonable to focus the analysis on the A-share market. 6 different stock indexes describing the Chinese stock were used.

These indexes reflect a wide spectrum of the Chinese equity market including small, large and medium caps. For example, the A50 is one of the most frequently used indexes for Chinese large cap stocks while the CSI 800 is a mid to small cap index describing companies listed in mainland China. For completeness purposes, and to limit some form of regional bias in the results, other international (non-Chinese) stock indexes (see Table 3.1) such as the Euro Stoxx 50, which is a frequently used index to describe the equity market

Table 3.1: Stock indexes.

Indexes	
A50	FTSE China A50 index composed of the largest Chinese companies.
CSI 300	Largest 300 companies listed in the Shenzhen and Shanghai Stock Exchanges
CSI 800	Largest 800 companies listed in the Shenzhen and Shanghai Stock Exchanges
Shanghai Composite	Index composed of all the stocks listed in the Shanghai Stock Exchange
Shenzhen Component	Index composed of the largest 500 companies in the Shenzhen Stock Exchange
SSE 50	Index composed of the largest 50 companies in the Shanghai Stock Exchange
Euro Stoxx 50	Index composed by 50 of the largest companies from 11 Euro zone countries
New York Composite	Index composed of all the stocks listed in the New York Stock Exchange

in Europe, and the New York Composite index which I used as a representative of the equity market in the United States were also used.

As described in the previous section the selection approach is used to forecast the direction of the movement of the stock index (up or down) in the next time period rather than the exact end price for that period. Daily closing prices for all the indexes mentioned in Table 3.1 were collected from the Bloomberg database for the period from 14 February 2007 to 30 March 2020. The returns of those index can be seen in Figure 3.1. Positive or zero returns will result in an R_T value of 0 and 1 otherwise.

The proposed approach was used to identify an appropriate combination of technical indicators trying to describe the performance of the equity market. Stock technical indicators are indicators typically based on historical performance of the stock as well as the traded volume of that stock. There is an ever increasing amount of technical indicators in the existing literature which can generate contradictory signals. 35 commonly used stock technical indicators (see Table 3.2) extracted from the database Bloomberg were used.

There is a very large amount of technical indicators ranging from simple moving averages (see figure 3.2) to proprietary indexes in which the actual algorithm describing the indexes is not publicly disclosed. It is also common to add volume information (see figure 3.3) as high or low volume can give an indication of possible stock price movements.

In figure 3.2 it can be seen that the moving average acts as a smoothing of the actual stock price. The longer the time frame for the moving average the smoother output that this technical indicator will generate. It is common to use more than one moving average. The intuition behind this approach is using different time windows as indicators for mid

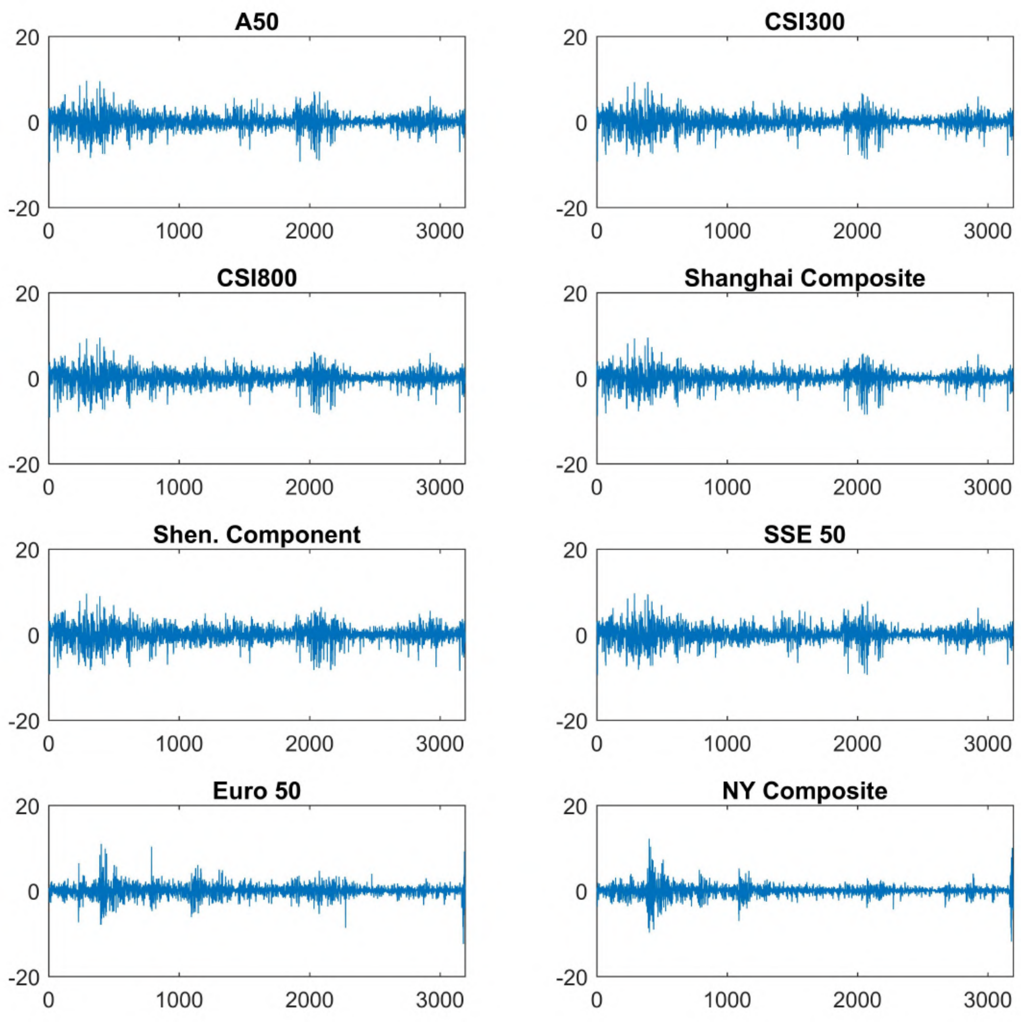


Figure 3.1: Returns of the indexes (measured in percentages).

Table 3.2: Technical indicators. C_t denotes the current price and L_t and H_t are the minimum and maximum prices in period t and MA means moving average. The exact formulas for some of the indexes are proprietary with the indicators for specific stocks obtainable from databases such as Bloomberg. Source: Blomberg, Kim [Kim, 2003].

N	Indicator	Description
1	SMAVG (50)	50 days simple MA. $\frac{\sum_{i=0}^{49} C_{t-i}}{50}$
2	SMAVG (100)	100 days simple MA. $\frac{\sum_{i=0}^{99} C_{t-i}}{100}$
3	SMAVG (200)	200 days simple MA. $\frac{\sum_{i=0}^{199} C_{t-i}}{200}$
4	Volume	Total traded volume
5	Stochastic %K (14-day)	$\frac{C_t - min(L_{t-n, \forall n \in \{1,\ldots,14\}})}{max(H_{t-n, \forall n \in \{1,\ldots,14\}}) - min(L_{t-n}, \forall n \in \{1,\ldots,14\})}$
6	%D (3-day)	3 days MA of stochastic %K. $\frac{\sum_{i=0}^{2} K_{t-i}}{3}$
7	%D (5-day)	5 days MA of stochastic %K. $\frac{\sum_{i=0}^{4} K_{t-i}}{5}$
8	Slow %D (3-day)	MA of the 3 days %D. $\frac{\sum_{i=0}^{2} D_{t-i}}{3}$
9	Slow %D (5-day)	MA of the 5 days %D. $\frac{\sum_{i=0}^{4} D_{t-i}}{5}$
10	Momentum (10-day)	Change in price over a 10 days period. $C_t - C_{t-10}$
11	Moving average (5D) of Momentum (10-day)	5 days MA of 10-day Momemtum. $\frac{\sum_{i=0}^{4} M_{t-i}}{5}$
12	ROC (daily)	Change in price in % over 1 day period. $\frac{C_t}{C_{t-1}}100$
13	Moving average (5D) of ROC	5 days MA of ROC. $\frac{\sum_{i=0}^{4} ROC_{t-i}}{5}$
14	Williams (14)	14 days Williams' indicator. $\frac{max(H_n, \forall n \in \{1,\ldots,14\}) - C_t}{max(H_n, \forall n \in \{1,\ldots,14\}) - min(L_n, \forall n \in \{1,\ldots,14\})}$
15	A/D oscillator	Accumulation distribution indicator. $\frac{H_t - C_{t-1}}{H_t - L_t}$
16	MA (5)	5 days moving average. $\frac{\sum_{i=0}^{4} C_{t-i}}{5}$
17	MA (10)	10 days moving average. $\frac{\sum_{i=0}^{9} C_{t-i}}{10}$
18	Disparity 5	$\frac{C_t}{MA_t}$
19	Disparity 10	$\frac{C_t}{MA_t}$
20	OSCP	$\frac{MA_t - MA_n}{MA_t}$
21	CMCI (13)	$\frac{(H_t+L_t+C_t)/3 - \frac{\sum(H_{t-i+1}+L_{t-i+1}+C_{t-i+1})}{3n}}{0.015(\frac{\sum\|H_{t-i+1}+L_{t-i+1}+C_{t-i+1} - \sum((H_t+L_t+C_t)/3)\|}{n})}$
22	RSI (14) on Close	Relative strength index. $100 - \frac{100}{1+\frac{Average\ up}{Average\ down}}$
23	MA (250)	250 days simple MA. $\frac{\sum_{i=0}^{249} C_{t-i}}{250}$
24	Z-score	Distance in standard deviations from MA
25	AMAVG(14,2,30)	Exponential MA
26	EMAVG (5) on Close	5 days exponential moving average with exponential term $\frac{2}{t+1}$
27	TMAVG (5) on Close	5 days triangular moving average with factor $\frac{t+1}{2}$
28	VMAVG (5) on Close	Variable MA with smoothing based in volatility
29	MACD(12,26)	T_1 exponential MA (fast) minus T_2 exponential MA (slow)
30	Hurst(25)	Exponential MA with mean reversion assumption.
31	FG(5)	Ratio of buying and selling strength (5 days)
32	Kairi (Simple,25)	Deviation of prices from its MA
33	Elder Force Index	Relationship between price and trading volume movements
34	JKHL Index	Relationship between new high and lows smoothed by two MA.
35	RMI(Close,14,10)	Relationship between overbought and oversold levels

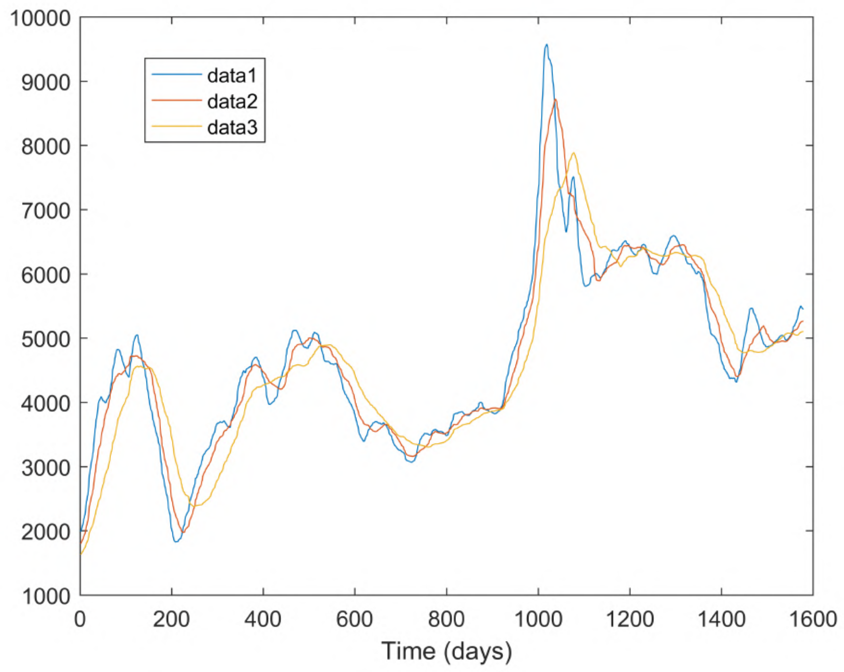


Figure 3.2: 50, 100 and 200 days moving average (CSI 500 Index).

and long term performance. It also should be notice that, by definition, the moving average will roughly follow the trend of the actual stock price. This is clearly different of other technical indicators such as for instance, oscillators.

The Chinese equity market has a large proportion of retail investors compared to institutional investors. Therefore it would seem reasonable to assume that indicators, such as momentum indicators, that are related to trend following investors (like retail investors usually are) is a reasonable technical indicator to consider. Therefore the 10-day momentum indicator (see figure 3.4) was included in the analysis. There is a large amount of momentum indicators available with most of them based on the idea that the market tend to follow, at least to some degree, the recent direction. It is easy to see that this type of technical indicator can generate a signal that it is rather different from the one obtained from a contrarian technical indicator, which are designed to try to indicate when an existing trend is likely to end. It seems reasonable to include both types of indicators as the market in fact shows these two types of behaviours.

As previously mentioned there is a large amount of momentum indicators, being some of the basic distinctions among them the time frame over which the indicator is estimated. For example, in some momentum technical indicators for daily forecasts the indicator is built using historical data ranging from a couple of days to 300 days or even longer. Some momentum indicators use information only about the recent minimum and maximum while other use the entire stock prices time series for a predetermined period of time. There is a very large amount of potential combinations that can be used to build momentum technical indicators using historical data. The Williams indicator, seen in figure 3.5, and

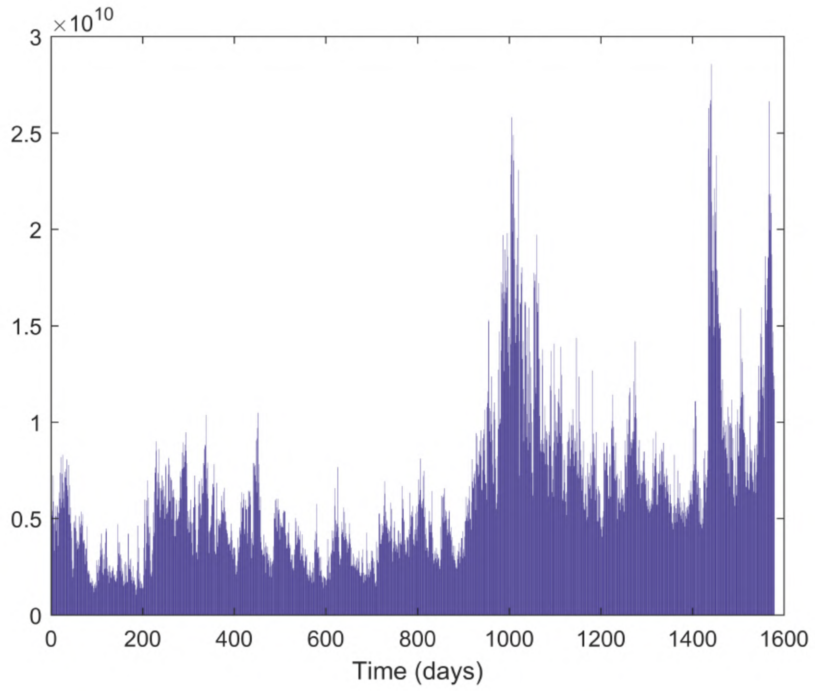


Figure 3.3: Volume (CSI 500 index).

the Relative Strength Index (RSI), seen in figure 3.6, are some other frequently used momentum indicators included in the analysis in this chapter.

Some other technical indicators follow a completely different approach from momentum indicators (trend following) and try to detect when a stock is overbought or oversold and are hence defined as contrarian indicators. In figure 3.7 it can be seen one such indicator (OSCP) that was included in the analysis. The Z-score, showed in figure 3.8, is also typically used in stock forecasting as an indicator of the distance to typical values.

From the graphs it is clear that the behavior of some of these technical indicators are clearly different and that can generate conflicting signals when used for stock forecasting or trend signaling.

Another frequently used technical indicators are oscillators, such as the A/D oscillator showed in figure 3.9. These are perhaps among some of the most frequently used technical indicators, mentioned for instance in [Ahn et al., 2018, Ye and Huang, 2008], together with moving averages.

Strictly speaking oscillators are a subtype of momentum indicators. The existing literature seems to support their use, particularly in markets with a high proportion of retail investors [Ni et al., 2015]. The idea, similarly to the case of overall momentum indicators, is that individual investors tend to follow the direction of the market (rather than taking contrarian views) in what is frequently called "Heard behavior" i.e., retail investors tend to buy when they see other retail investor buying a vice versa with little analysis regarding the future of the current trend. This type of trading can be potentially

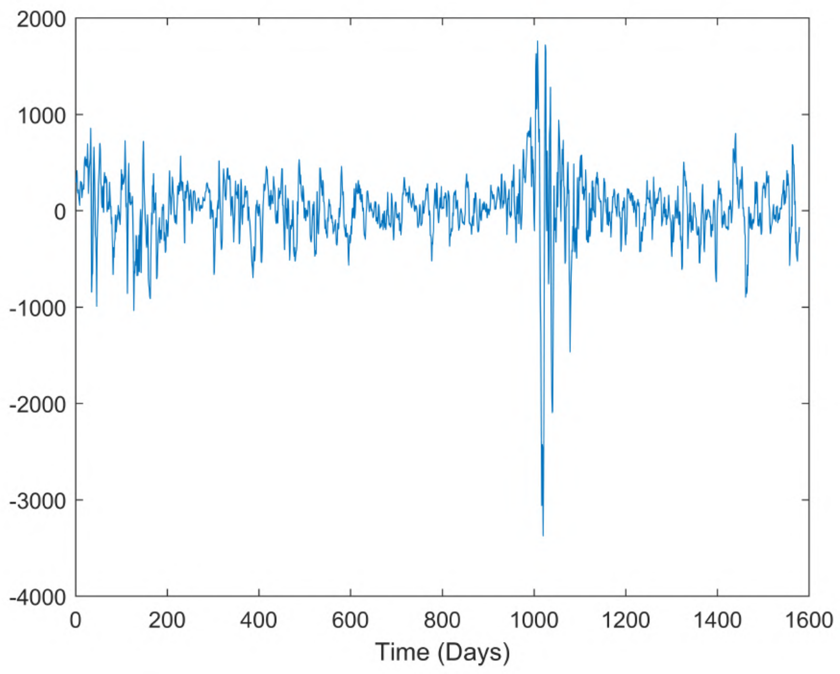


Figure 3.4: Momentum (CSI 500 index).

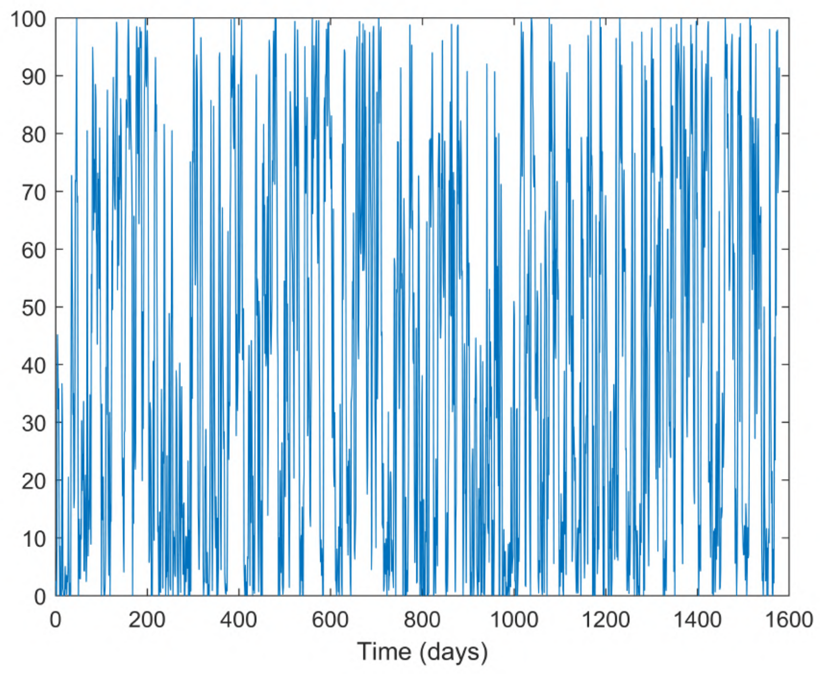


Figure 3.5: Williams (CSI 500 index).

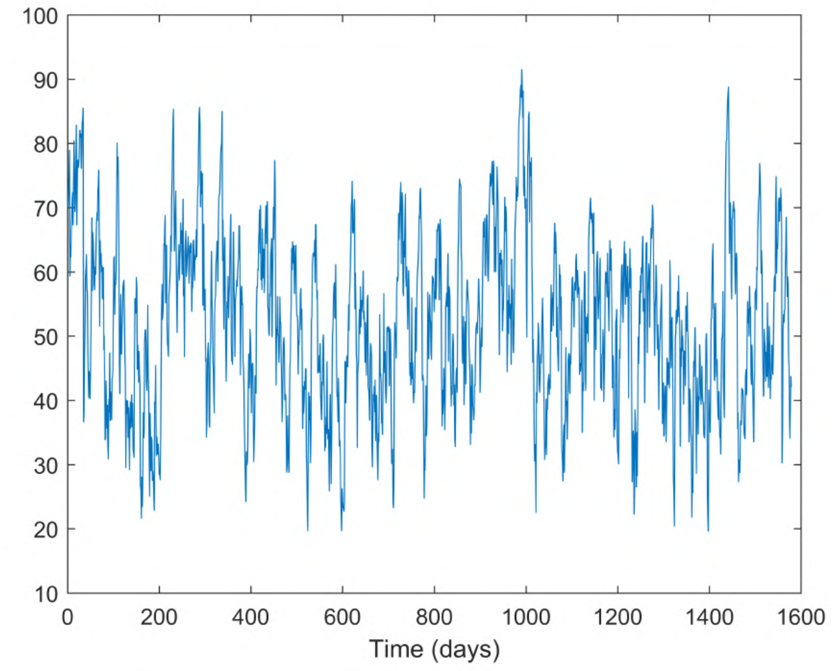


Figure 3.6: RSI (CSI 500 index).

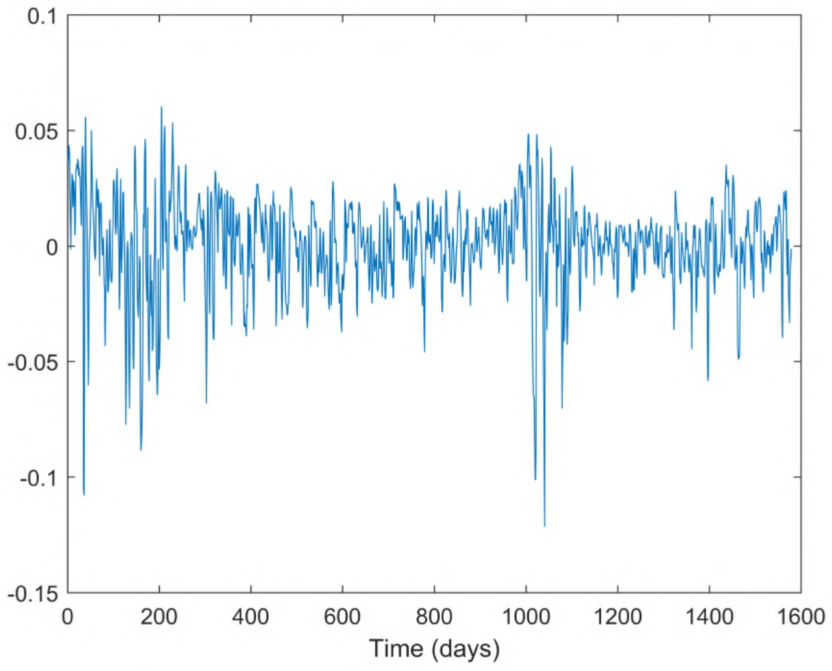


Figure 3.7: OSCP (CSI 500 index).

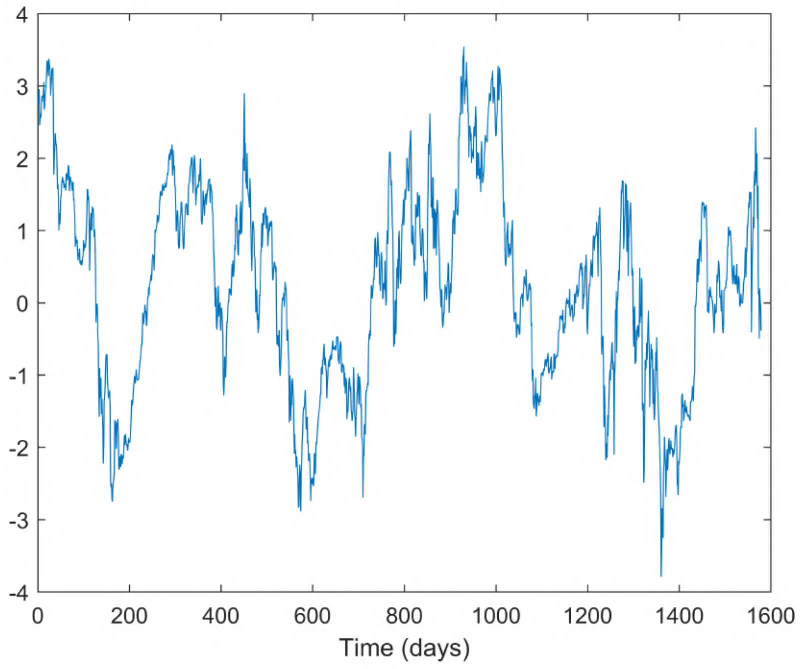


Figure 3.8: Z-score index (CSI 500 index).

risky as the investor might buy at the peak or sell at the bottom which are clearly not desirable outcomes.

There is also a relatively large amount of proprietary, or at least not fully publicly disclosed, indicators such as for instance the Kairi index, the Elder Force Index and the JKHL index. These three technical indicators can be seen in figures 3.10, 3.11 and 3.12.

The proposed approach was implemented in Matlab using neural networks as classifiers. A total of 100 initial configurations times 2500 iterations were carried out for each index, which translates in 250,000 neural networks per index and a total of 2,000,000 neural networks (8 indexes). The neural networks used were back propagation classification neural networks with one hidden layer with 25 neurons and trained with the Levenberg-Marquardt rule. The value for the number of neurons was chosen as a result of the preliminary sensitivity analysis on the number of neurons presented in Section 3.4.

3.4 Results

Previously to testing the proposed strategy, a preliminary sensitivity analysis on the number of neurons has been carried out to find the most convenient number of neurons in the hidden layer. The full set of 35 technical indicators was used, and the number of neurons in the hidden layer was increased from 25 to 25,000 in steps of 25 neurons. Simply increasing the number of neurons did not appear to increase the classification accuracy of neural networks for the performance on stock index in the following period $(t + 1)$ (see Figure 3.13) for any of the six indexes analyzed.

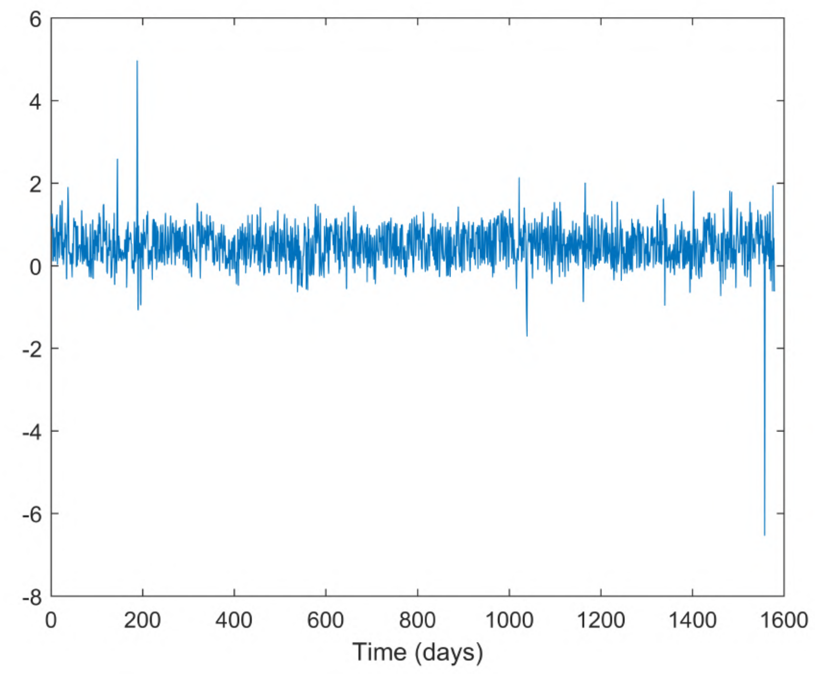


Figure 3.9: A/D Oscillator (CSI 500 index).

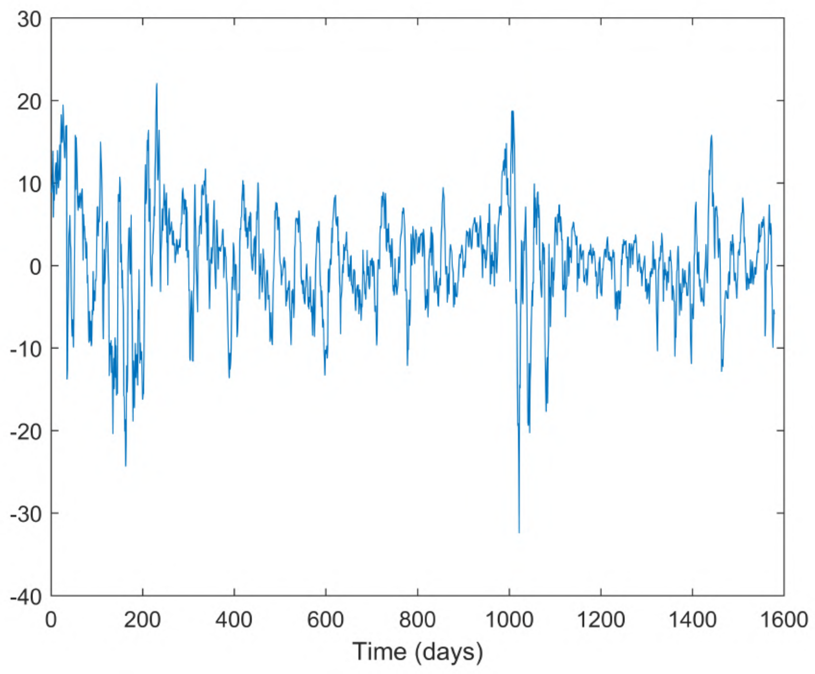


Figure 3.10: Kairi (CSI 500 index).

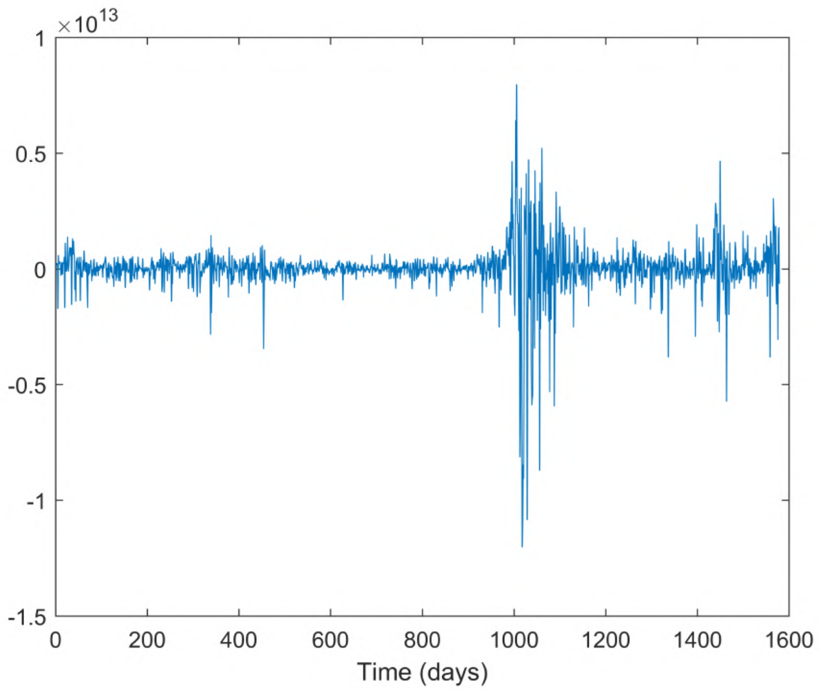


Figure 3.11: Elder Force Index (CSI 500 index).

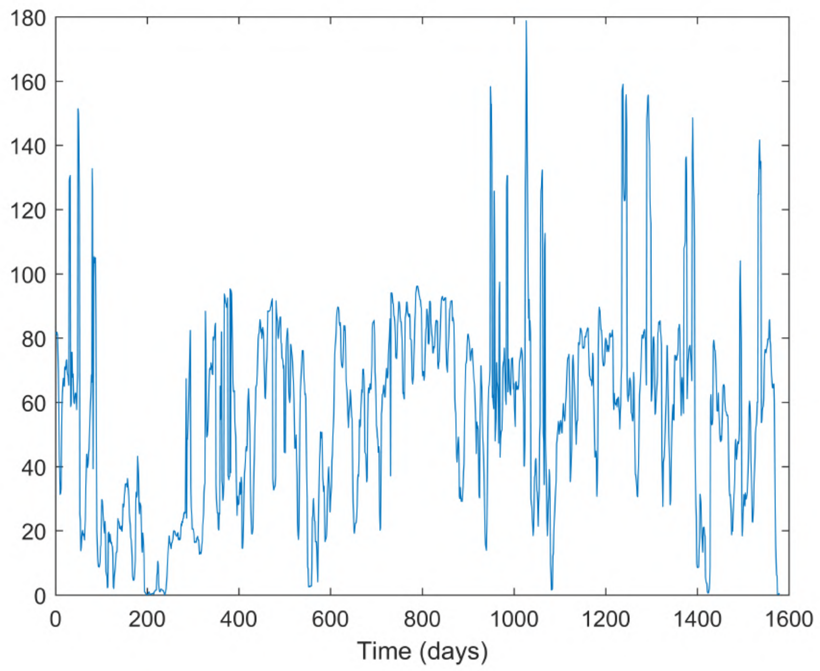


Figure 3.12: JKHL Index (CSI 500 index).

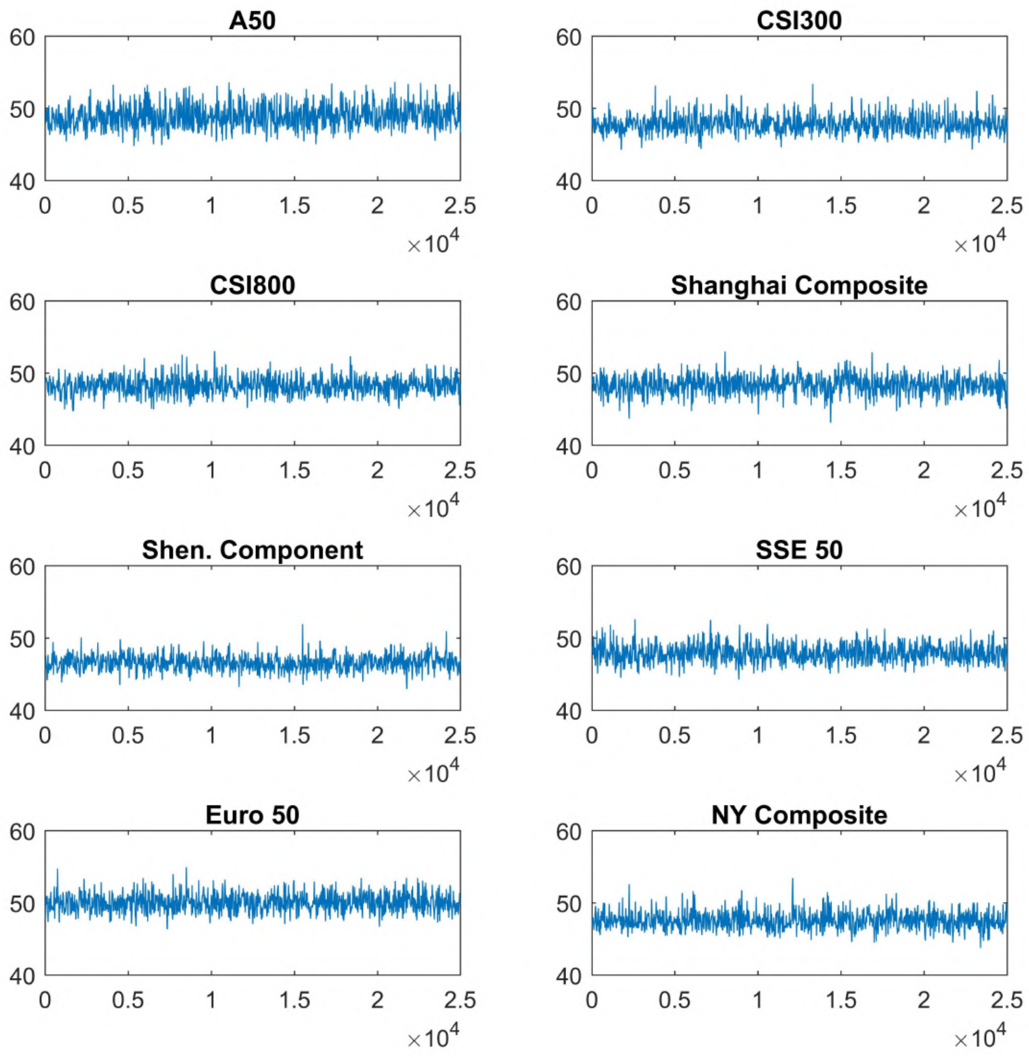


Figure 3.13: Sensitivity analysis of the error rate (using directly neural networks) as the number of neurons is increased.

On the other hand, Figure 3.14 shows an example of the evolution of the error rate (misclassification of up/down days) in the training of one the neural networks. It can be seen that training improves the fitting to the data up to a significant number of training iterations, showing that the NN is really learning the ground truth function ϕ.

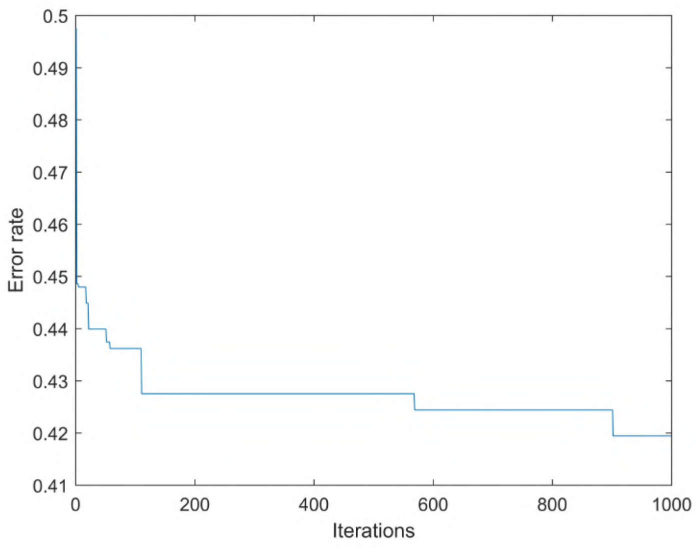


Figure 3.14: Example of the evolution of the error rate as the number of iterations increases.

The indicator selection approach is a combinatorial approach that can be used to select an appropriate combination of variables in non-linear models, using techniques such as neural networks. In the example illustrated the proposed approach was implemented in six different Chinese stock indexes plus two world indexes.

The error rate obtained using the algorithm in combination with neural networks was lower than the error rate obtained using directly neural networks including all the 35 available variables, see Table 3.3. The average improvement in the error rate (over all the considered indexes) was 9.1%. It turns out that this is a very good result considering how difficult is the task of predicting the movements of the market and the great amount of benefits that can be realized even with a small improvement in the forecasting. Moreover, in some of the indexes the improvement is quite high, in excess of 11% (A50, Shanghai Composite and SSE50). A very interesting finding is that the baseline approach, that is, considering all the technical indicators available for forecasting yields error rates greater than 50% in some cases (A50, CSI800, Shanghai Composite, SSE 50 and Euro Stoxx 50). This means that tossing a coin produces better results than using the full pool of indicators. The reason for that is that, as pointed out in Section 3.3, some of the technical indicators produce contradictory signals.

The histogram indicating the frequency of appearance of the technical indicators in the output of the algorithm for the various indexes shown in Table 3.3 can be seen in Figure 3.15. It is evident that some of the technical indicators are picked more frequently than others, thus they are more likely to provide better accuracy in the prediction of the price change direction.

Table 3.3: Classification errors (up and down index movements) of the proposed approach using neural networks compared to the direct application of neural networks using all the 35 technical indicators.

Index	**Direct (Error %)**	**Proposed (Error %)**	**Improv. (%)**	**Combinations**
A50	54.0	41.2	12.8	(1,2,4,6,7,10,12,13,14,15 19,21,23,25,34,35)
CSI 300	46.9	40.3	6.6	(2,3,4,7,8,11,12,13 15,19,21,25,26,29,34)
CSI 800	50.7	41.1	9.6	(2,4,5,6,9,12,13,16 17,20,22,28,30,32,33,34)
Shanghai Composite	52.3	40.1	12.2	(1,2,4,5,6,9,10,12,13,15 16,18,19,20,21,22,24,33)
Shenzhen Component	45.6	40.7	4.9	(2,4,5,8,10,12,14,19,21 22,25,28,29,30,33,34,35)
SSE 50	51.0	39.6	11.4	(1,2,3,7,9,10,11,12,20,21 22,24,25,27,28,30,32,35)
Euro Stoxx 50	50.1	41.8	8.3	(3,4,7,11,17,18,20,22 25,27,28,31,33,34)
New York Composite	47.2	40.7	9.0	(1,3,7,8,9,11,14,15,17,19 20,25,27,28,29,33,34)
Average	49.7	41.0	9.1	

Nevertheless, this does not imply that using only these more relevant indicators will lead to a better forecasting. As an example the combination formed by the indicators with highest frequency of occurrence in Table 3.3 and Figure 3.15, i.e., $\{2, 4, 12, 25, 34\}$, gives an average improvement of 2.3% over the baseline approach, which is significantly worse than that achieved by using the proposed combinations.

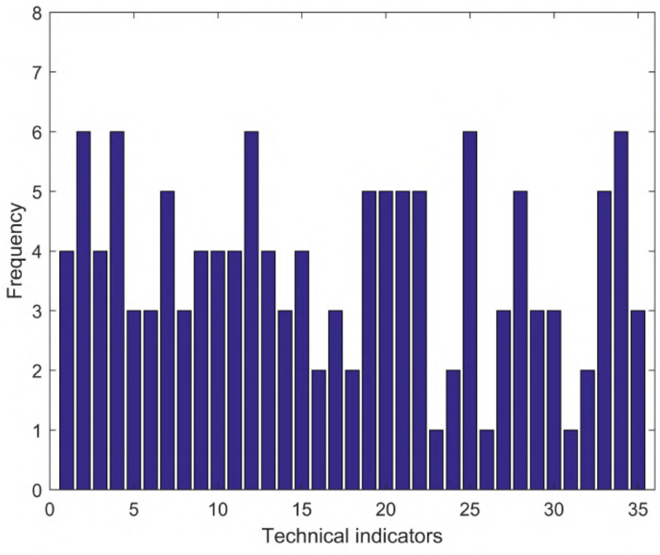


Figure 3.15: Histogram of the technical indicators appearance in the output of the proposed algorithm.

Table 3.4 shows the improvement along the iterations on the algorithm. The average improvement from the first iteration to the last was 7.1%. While this value clearly evinces that the algorithm improves the initial combinations, a more rigorous test has been done.

A Wilcoxon test was carried out comparing the distribution of error rates obtained in the first and last iterations for each index. The Wilcoxon test rejects the hypothesis, for all the indexes analyzed, that the median of the error rates for the initial and final distributions are statistically the same (Table 3.5), suggesting that the iterative process does significantly improve accuracy.

The same approach was followed to compare the error rate using neural networks directly (with all 35 technical indicators) with the error rate obtained using the technical indicator selection approach (Table 3.6). The Wilcoxon test rejects the null hypothesis that the median of the error rates obtained using these two methods are statistically equivalent, suggesting that the proposed method statistically significantly improved the forecasting accuracy of up/down stock index movements.

Remark 2 has been also taken into account, and the algorithm has also been used with $M = 2$ and a total $125,000$ iterations which should be roughly equivalent to the parameters used previously, that is 100 combinations and 2500 iterations. The average improvement in this case was 8.7% which is marginally worse than that previously achieved. The difference could be due to the lower diversity of the set of candidate solutions, but also to the randomized nature of the strategy.

Table 3.4: Classification errors (up and down index movements) of the selection approach using neural networks.

Index	**Error (Initial) %**	**Error (Final) %**	**Improvement %**
A50	48.4	41.2	7.2
CSI 300	48.1	40.3	7.8
CSI 800	48.3	41.1	7.2
Shanghai Composite	46.9	40.1	6.8
Shenzhen Component	46.1	40.7	5.4
SSE 50	47.3	39.6	7.7
Euro Stoxx 50	49.7	41.8	7.9
New York Composite	47.5	40.6	6.9
Average	47.5	40.7	7.1

Table 3.5: Wilcoxon test comparing the error rate in the first and last iterations (p-values).

Index	***p*-Value**
A50	5.5467E-127
CSI 300	6.3542E-31
CSI 800	2.8497E-126
Shanghai Composite	6.4421E-115
Shenzhen Component	7.8451E-51
SSE 50	6.5487E-121
Euro Stoxx 50	4.6587E-114
New York Composite	7.6257E-112

Table 3.6: Wilcoxon test comparing the error rate using neural networks directly with all the 35 technical indicators and the algorithm output.

Index	***p*-Value**
A50	6.7894E-172
CSI 300	5.6284E-161
CSI 800	3.5487E-155
Shanghai Composite	4.6672E-167
Shenzhen Component	3.4837E-175
SSE 50	4.45728E-156
Euro Stoxx 50	4.6524E-156
New York Composite	2.7845E-165

The average total time per index (100 initial configurations times 2500 iterations) was 157,691 s. The calculation time for each index can be seen in Table 3.7. The calculations were carried out in Matlab 2016 in an Intel, i5-3470, 3.2 GHz, 64 bit computer. The selection approach requires a significant amount of computation time but it clearly is faster than calculating all the possible combinations of technical indicators, which is for the example presented not a feasible calculation in a normal computer.

Table 3.7: Calculation time.

Index	**Total Time (sec)**
A50	159,132
CSI 300	177,876
CSI 800	144,195
Shanghai Composite	150,710
Shenzhen Component	169,498
SSE 50	156,739
Euro Stoxx 50	142,816
New York Composite	160,566
Average	157,691

3.5 Discussion

The proposed method can be a feasible approach, when trying to determine a combination of variables or features to be used when forecasting the behaviour of non-linear processes. In the particular example of the stock market, there is a very large number of technical indicators that are intended to give the investor some indication of the future performance of the stock. These indicators can generate contradictory signals and selecting the appropriate combination of technical indicators can become a difficult task.

Reducing the dimensionality of the problem is also important to avoid issues such as local minimum, that can cause poor generalization when applying techniques such as neural networks. It is showed in this chapter that it is possible to use this approach in the Chinese stock market (generating an appropriate combination of independent variables for non-linear models), obtaining better results than directly applying neural networks to all the available independent variables. This was tested using 6 Chinese stock index (as well as two international indexes) and 35 technical indicators.

There was an average 9.1% improvement when using the combinatorial approach with neural networks over the results using directly all the technical indicators and neural networks as the non-linear forecasting technique. The formal statistical analysis comparing the results using neural networks directly (all technical indicators) with the results from the combinatorial approach using neural networks shows that there are statistically significant difference for the error rates obtained at a 1%, 5% and 10% significance level, supporting once more the hypothesis that the combinatorial approach using neural networks is a more appropriate tool for forecasting the direction of the stock market movement, at least for the 8 indexes analyzed, than using neural networks directly. Thus, for eight different indexes, better combinations of the technical indicators have been found, offering a practical choice to improve the forecasting accuracy and hence the expected benefits.

The total calculation time per index (100 initial configurations time 2500 iterations) was 157,691 seconds. While this is a substantial amount of time it is a calculation that can be done with a normal laptop computer. Moreover, many of the operations of the proposed algorithm can be done in parallel further shortening the computation times.

Although a direct comparison is challenging, the approach of using the combinatorial approach with neural networks seems to be generate better results for stock forecasting purposes than other approaches used in the existing literature, such as for instance the Box–Jenkins approach used by Groda and Vrbka [Groda and Vrbka, 2017], which the authors considered not suitable. A more comparable paper is Kim and Han [Kim and Han, 2000] that achieved a hit rate of 61% in the Korean market using genetic algorithm in combination to neural networks which is comparable to the 59% rate obtained in the Chinese market. Nevertheless comparison across different stock markets should be taken with caution. For instance, it should be naive to believe that the same approach would generate the same results in two markets as different as South Korea and China with China being an open stock market dominated by institutional investors while the Chinese market is a market dominated by local retail investors.

The selection approach was illustrated in the context of the stock market and using neural networks but the approach is easily applicable to other fields. This is increasingly important as the amount of data available in many fields has increased substantially over the last few decades with an ever increasing need for tools to process large databases. Besides neural networks other non-linear models, such as for instance support vector machines or the forecasting techniques presented in the next chapter, can be used in the proposed approach. This could be an interesting area of future work.

Chapter 4

Stock forecasting using local data

This chapter is based on the following journal publication:

- Gerardo Alfonso, A. Daniel Carnerero, Daniel R. Ramirez, Teodoro Alamo. Stock forecasting using local data. IEEE Access 2021, vol.9, pp.9334-6344. Impact factor 3.745, JCR ranking Q1 (2019)

4.1 Introduction

Stock price forecasting is a challenging field that has attracted researchers from different fields including engineers and scientists. It is likely fair to say that there is not yet an approach for stock forecasting that is accepted as superior, with the existing approaches having their strengths and weaknesses.

Due to the financial relevance of stock price forecasting, many different techniques have been applied to the problem. The almost random nature of the market have made brownian motion [Osborne, 1959] and martingale models [Danthine, 1977, Barnett and Serletis, 2000] one of the first choices.

Since the efficient market hypothesis is not proved, more elaborate techniques have been used trying to exploit the market inefficencies. Among these techniques, in the literature can be found applications with linear models [Zheng and Zhu, 2017], support vector machines [Lin et al., 2013], genetic algorithms [Mahfoud and Mani, 1996] or more frequently neural networks [Baba and Kozaki, 1992, White, 1988, Guresen et al., 2011] and deep learning methods [Cao and Wang, 2019, Yu and Yan, 2020, Chen et al., 2020] (for a recent survey on the topic see [Rao et al., 2020]). While neural networks are an important stock forecasting technique it should be noted that, as in any other technique, it has limitations. [Horák and Krulický, 2019] did in this regard an interesting comparison between the exponential time series alignment method and the time series alignment with neural networks. The authors highlighted the importance of neural networks in the field of stock forecasting mentioning that generally neural networks provide better forecast than traditional methods. However, they also concluded that in their example, applied to a volatile stock, the traditional forecasting method generated better results than the neural network. This highlights the importance of using the appropriate stock forecasting techniques with papers in the existing literature finding less than optimal results for some popular techniques. For instance, [Groda and Vrbka, 2017] concluded that the widely used Box–Jenkins model is not an appropriate method in the case of stocks listed in the Prague Stock Exchange.

In this chapter two quantitative techniques that have not been applied to the problem of stock price and price interval forecasting are presented. One is an approach derived from the predictive control strategy presented in [Salvador et al., 2020, Salvador et al., 2018], which in turn can be related to the direct weight optimization approach [Bravo et al., 2016, Roll et al., 2005b, Jianhong, 2015]. Direct weight optimization uses linear estimators and convex optimization [Roll et al., 2005a], and has been applied in different fields like predictive control [He et al., 2010], nonlinear system identification [Bai and Liu, 2007] or electron density analysis [Wu and Van Voorhis, 2005].

The proposed approach uses local data, that is, only a subset of the whole data available, chosen among those past stock market states that are close to the current state. With such subset, the approach computes an optimal linear combination of past states that equals the current state, using such combination then to compute the price forecast. Unlike other methods, like neural networks, the proposed approach do not use a training phase as the subset and the linear combination is computed each time a forecast is needed. This allows an easy adaptation to different market situations and also the updating of the database as new data are available without having to retrain the estimator. Furthermore, the use of local data results in a lower computational burden, as the cardinality of the subset will be much lower than that of the whole data set.

The other technique proposed in the chapter is a probabilistic price interval strategy previously presented in a more general context in [Carnerero et al., 2020]. This strategy can be used to forecast stock prices but its main application is to provide price intervals with a guaranteed probability of containing the real price. In this sense is complementary to the first approach as it provides the guarantees that the previous one lacks. On the other hand, although the algorithm is highly parallelizable, the computational burden is higher, thus it does not replace the first approach if no guarantees are required. This approach uses dissimilarity functions evaluated on local data to build an empirical probability distribution of the predicted price. Thus using such distribution it is possible to build price intervals using the desired percentiles and also predict the forecast using the median of the distribution. This can be very useful for risk management purposes [Berkowitz, 2001], a field of increasing importance in finance.

Finally, as a case study, the techniques have been applied to the task of predicting future values of the Dow Jones Industrial Average Index up to 5 days (i.e., a full week), and also in the intraday market, validating the results in relation to two baseline approaches, a persistence (martingale) predictor and a neural network based predictor. Furthermore, quantile regression [Koenker and Hallock, 2001, Pradeepkumar and Ravi, 2017] has been used as a third baseline approach to validate the predicted price intervals. The results prove that the proposed techniques are a valuable tool that can be added to the portfolio of existing techniques for stock price forecasting.

The chapter is organized as follows: section 4.2 presents the first strategy for stock price forecasting using local data. The probabilistic price interval strategy is shown in section 4.3. Section 4.4 presents the results of these two strategies when used to forecast the Dow Jones Average Industrial Index.

4.2 Stock Price Forecasting using Local Data

The first approach used in this chapter to forecast stock prices is based on the technique presented for predictive control in [Salvador et al., 2020, Salvador et al., 2018].

Consider the evolution of the price of a stock as a time series $p(t) \in \mathcal{P}$, being t the time index expressed in the proper time unit, usually days in the case of the daily market, and $\mathcal{P}$ the possible range of values for the stock price. The state of the price time series is described as the value at time t of series of technical indicators, i.e.,

$$z(t) = (Z_1(t), Z_2(t), \dots, Z_{nz}(t)) \in \Re^{nz}.$$

These technical indicators can be past values of $p(t)$, stock price returns or more complex metrics like moving averages or the relative strength indicator amongst others (see chapter 3). The objective is to be able to predict k-steps ahead the price of a stock, that is, to obtain $\hat{p}(t+k)$ at time t in such a way that it is as close as possible to $p(t+k)$.

The approach presented here uses a database of past values of $z(t)$ and k-step ahead stock prices. The database DB_k for predicting k-steps ahead the price of $p(t)$ will be implemented as a table with N_{DB} entries (i.e., rows), in which each entry contains a past value of $z(t)$ and the corresponding $p(t+k)$, as shown in table 4.1.

$z(i_1)$	$p(i_1+k)$
$z(i_2)$	$p(i_2+k)$
$z(i_3)$	$p(i_3+k)$
$\vdots$	$\vdots$

Table 4.1: Structure of the database DB_k.

Note that the time indexes i_j of the past states do not have to be ordered in any way or be consecutive, thus they are not required to form a proper time series. The only requisite is that the price associated to $z(i_j)$ is the one corresponding to k steps after time i_j.

The proposed approach does not use the database to train or fit a predictor, as in the training of a neural network. Thus, the database is not considered a training set (except for the tuning of a reduced number of hyperparameters). Instead, it is used every time a prediction is needed in an oracle fashion. Furthermore, the approach considered in this section uses only a subset of the database, denoted as $\Omega(z(t))$, to compute the prediction $\hat{p}(t+k)$. In this sense it is also different from techniques like neural networks in which the estimator is fitted using the whole training set. More precisely, given a distance measurement function $d(z(t), z(i_j))$, and cardinality parameter N, the elements of $\Omega(z(t))$ are obtained selecting the N states $z(i_j)$ closer to $z(t)$. Thus, the prediction is computed using only local data.

Once the data that are to be included in $\Omega(z(t))$ are selected, the proposed approach proceeds to compute an optimal combination of all the $z(i_j)$ in $\Omega(z(t))$ that matches $z(t)$, using the weights of such combination to compute $\hat{p}(t+k)$ as the corresponding combination of all the $p(i_j+k)$ in $\Omega(z(t))$. Furthermore, a regularization term, weighted by a scalar $\gamma \geq 0$, is included in the computation of the optimal combination. Algorithm 1 gives a formal description of the proposed approach.

Algorithm 1 k-step ahead stock forecasting using local data

```
Input: $DB_k$, $z(t)$, $N$ and $\gamma$.
Output: $\hat{p}(t+k)$ (estimation of price at $t+k$).
1: Compute the distance $d(z(t), z(i_j))$ for all $z(i_j)$ in the database $DB_k$.
2: Create a list of the entries in $DB_k$ sorted according to the distances $d(z(t), z(i_j))$.
   Denote as $z_l$ and $p_{l,k}$ the state $z(i_j)$ and $k$-step ahead price $p(i_j+k)$ of the $l$-th entry
   in this ordered list.
3: Build $\Omega(z(t))$ using the first $N$ entries in the ordered list, that is,
   $$\Omega(z(t)) \triangleq \{(z_l, p_{l,k})\} \quad \forall l \in \{1, \dots, N\}.$$
4: Solve the following QP problem:
   $$\begin{aligned} \min_{\lambda_1, \lambda_2, \dots, \lambda_N} \quad & \sum_{l=1}^{N} \lambda_l^2 + \gamma|\lambda_l| \\ \text{s.t.} \quad & \sum_{l=1}^{N} \lambda_l = 1, \\ & \sum_{l=1}^{N} \lambda_l z_l = z(t). \end{aligned}$$
5: Compute $\hat{p}(t+k)$ as:
   $$\hat{p}(t+k) = \sum_{l=1}^{N} \lambda_l p_{l,k}.$$
```

Remark 1. *The distance $d(\cdot,\cdot)$ can be any measure of how close are the states $z(i_j)$ stored in DB_k to $z(t)$. A typical choice would be the Euclidean distance, i.e., $d(z(t), z(i_j)) = \|z(t) - z(i_j)\|$, but also could consider other aspects like the time span between states, i,e.,*

$$d(z(t), z(i_j)) = \|z(t) - z(i_j)\| + \rho|t - i_j|,$$

where the non negative scalar ρ would be a weighting factor. In this way recent data would be prioritized in the selection process of step 3 in algorithm 1. Other aspects like seasonality *could also be taken into account using the modulus operator, and, in general, many of the resemblance measures used in cluster analysis [Anderberg, 1973].*

The optimization problem in step 4 of algorithm 1 can be easily solved, specially when $\gamma = 0$ as it results in a QP problem with equality constraints whose solution is that of a system of linear equations. Moreover, steps 1 and 2 can be easily parallelized, thus efficient implementations of algorithm 1 can be obtained.

Note that the fact that only local data is used to compute $\hat{p}(t+k)$ makes the strategy adaptive, being the definition of $d(\cdot,\cdot)$ the way to change how the strategy adapts to the current price variations. Finally, the proposed approach does not require a training phase (except for the possible tuning of the hyperparameter $\gamma \geq 0$), thus new data can be included in the database as they are available, without having to retrain the predictor. As in Lasso approaches [Tibshirani, 1996], larger values of γ tend to maker a larger fraction of the weights equal to zero, providing an enhanced local approach approximation. Thus, γ is an hyperparameter that potentially improves the quality of the predictions.

4.3 Probabilistic price interval forecasting

The price forecasting approach presented in the previous section provides an easy and convenient way of forecasting stock prices k-steps ahead. However, this approach does not provide any measure on how the real price could deviate from the forecasted one and also it does not have any guarantee on that deviation. In this section the approach presented in [Carnerero et al., 2020] is adapted for stock price interval forecasting with probabilistic guarantees. The proposed methodology computes an interval prediction for the price $p(t+k)$ in which the lower and upper bound of the interval are computed taking into account given probabilistic specifications. The use of local data is introduced here in the strategy to better handle large databases. The reader is referred to [Carnerero et al., 2020] for a full description of the procedures involved in the original strategy. Here it will be shown the main concepts and implementation details.

The proposed strategy is based on building an empirical conditional probability distribution for $p(t+k)$ subject to $z(t)$. Let $p_{\underline{\alpha}}$ be the $\underline{\alpha}$-th percentile and $p_{\overline{\alpha}}$ the $\overline{\alpha}$-th percentile of such distribution. Then, the interval $[p_{\underline{\alpha}}, p_{\overline{\alpha}}]$ will contain the price with a probability of $\frac{\overline{\alpha}-\underline{\alpha}}{100}$. Thus, finding the intervals amounts to compute the lowest and highest percentile that define the desired interval for a given probability, e.g., for a probability of 0.8 the desired interval will be $[p_{10}, p_{90}]$. On the other hand, if a forecast $\hat{p}(t+k)$ is also needed, it can be chosen as the 50th percentile of the distribution.

The key concept in this approach is that of dissimilarity function, a generalization of the optimization problem in algorithm 1. A dissimilarity function measures how similar the given pair $(z(t), p)$ is to the set of pairs $(z_l, p_{l,k})$ of $\Omega \subseteq DB_k$. The formal definition of dissimilarity function is given in the following.

Definition 4.1. *Given $\Omega \subseteq DB_k$, a scalar $\gamma \geq 0$, $z(t)$ and price p then the dissimilarity function $J_\gamma(\cdot,\cdot,\cdot)$ will be defined as:*

$$\begin{aligned} J_\gamma(z(t),p,\Omega) = \min_{\lambda_1,\lambda_2,\ldots,\lambda_N} \quad & \sum_{l=1}^{N} \lambda_l^2 + \gamma|\lambda_l| \\ s.t. \quad & \sum_{l=1}^{N} \lambda_l = 1, \\ & \sum_{l=1}^{N} \lambda_l z_l = z(t), \\ & \sum_{l=1}^{N} \lambda_l p_{l,k} = p, \end{aligned}$$

where, as in section 4.2, the N pairs $(z_l, p_{l,k}) \in \Omega$ denote the state $z(i_j)$ and k-step ahead price $p(i_j + k)$ of the l-th entry in Ω.

The dissimilarity function J_γ has a lower value when it is easy to represent $(z(t), p)$ as a combination of the N pairs $(z_l, p_{l,k})$ of Ω and a higher value otherwise, for further details see [Carnerero et al., 2020]. Notice that given $z(t)$, the value of p that minimizes the dissimilarity function $J_\gamma(z(t), p, \Omega)$ is equal to the k-step ahead forecast of Algorithm 1.

The other key concept in the approach is the empirical conditional probability density function (ecp) [Carnerero et al., 2020], that uses the dissimilarity function of definition 4.1, and approximates the real distribution of $p(t+k)$ conditioned to $z(t)$.

Definition 4.2. *For a given $\Omega \subseteq DB_k$, $\gamma \geq 0$, $c > 0$, $z(t)$ and price p, the empirical conditional probability density function (pdf)* ecp *is defined as:*

$$\text{ecp}_{\gamma,c}(z(t),p,\Omega) = \frac{e^{-cJ_\gamma(z(t),p,\Omega)}}{\int\limits_{\mathcal{P}} e^{-cJ_\gamma(z(t),\breve{p},\Omega)} d\breve{p}},$$

where $\mathcal{P}$ is the set of all possible values of $p(t+k)$ for all $t+k$.

Note that according to this definition, given $z(t)$, the probability of $p(t+k)$ being in a certain interval $[p_a, p_b]$ is approximately equal to

$$\int\limits_{p_a}^{p_b} \text{ecp}_{\gamma,c}(z(t),p,\Omega)dp.$$

In order to obtain the interval prediction $[p_{\underline{\alpha}}, p_{\overline{\alpha}}]$, the hyperparameters γ and c are chosen to make the approximation as sharp as possible for $p_a = p_{\underline{\alpha}}$ and $p_b = p_{\overline{\alpha}}$.

The hyperparameter c affects how the prices are distributed around its expected values. Higher values of c yield a more narrow pdf. Lower values of γ are appropiate when the real pdf is close to a normal distribution whereas higher values of γ are to be used if the ditribution is flat and close to a uniform distribution. Thus, the family of empirical distributions parameterized with c and γ encompasses a broad range of distributions [Carnerero et al., 2020].

Note however that, in practice, the integral in definition 4.2 should be computed numerically over a finite set of possible values of $p(t+k)$, denoted as $\mathcal{P}_s \subset \mathcal{P}$ obtained from a grid of N_p values $\bar{p}_i$ in the interval $[p_{min}, p_{max}]$[1] with $\bar{p}_1 = p_{min}$ and $\bar{p}_{N_p} = p_{max}$. Denote the increment between two sucessive prices $\bar{p}_i \in \mathcal{P}_s$ as:

$$\Delta\bar{p} = \bar{p}_{i+1} - \bar{p}_i$$

Then, the approximation of the ecp will be computed as

$$\mathrm{ecp}_{\gamma,c}(z(t), p, \Omega) \approx \frac{e^{-cJ_\gamma(z(t),p,\Omega)}}{I_S}, \tag{4.1}$$

where the approximation of the integral can be computed using the trapezoidal rule[2] obtaining

$$I_S = \Delta\bar{p} \sum_{i=1}^{N_p-1} \frac{e^{-cJ_\gamma(z(t),\bar{p}_{i+1},\Omega)} + e^{-cJ_\gamma(z(t),\bar{p}_i,\Omega)}}{2}.$$

Once the empirical distribution of $p(t+k)$ is obtained, computing the desired percentiles requires to find the value $p_{\overline{\alpha}}$ for which

$$\int_{p_{min}}^{p_{\overline{\alpha}}} \mathrm{ecp}_{\gamma,c}(z(t), p, \Omega) dp = \frac{\overline{\alpha}}{100}.$$

holds and repeating the operation for $\underline{\alpha}$ to obtain $p_{\underline{\alpha}}$. As in the previous case, these integrals should be computed numerically. In the case of finding $p_{\overline{\alpha}}$ and using the trapezoidal approximation, it reduces to solve:

$$\begin{aligned} i_{\overline{\alpha}} = \arg\min_i \quad & i \\ \text{s.t.} \sum_{j=1}^{i} \frac{\varphi_{j+1} + \varphi_j}{2} & \geq \frac{\overline{\alpha}}{100\Delta\bar{p}}, \end{aligned} \tag{4.2}$$

where $\varphi_j = \mathrm{ecp}(z(t), \bar{p}_j, \Omega)$ computed as in (4.1), and then

$$p_{\overline{\alpha}} = \bar{p}_{i_{\overline{\alpha}}+1} \in \mathcal{P}_s. \tag{4.3}$$

[1]The choice of p_{min} and p_{max} can be done arbitrarily conservative, as the only requisite is that with a high probability any $p(t+k)$ verifies that $p(t+k) \in [p_{min}, p_{max}]$. However it is better to use reasonably tight bounds that require a lower N_p to sample the interval correctly.

[2]More accurate methods can be used instead of the trapezoidal rule, which has been chosen here because of its simplicity and low requeriments on the function to be integrated.

On the other hand, finding the lower percentile $p_{\underline{\alpha}}$ requires to solve:

$$\begin{aligned} i_{\underline{\alpha}} = \arg\max_{i} \quad & i \\ \text{s.t.} \sum_{j=1}^{i} \frac{\varphi_{j+1} + \varphi_j}{2} &\leq \frac{\underline{\alpha}}{100\Delta\bar{p}}, \end{aligned} \tag{4.4}$$

and then

$$p_{\underline{\alpha}} = \bar{p}_{i_{\underline{\alpha}}+1} \in \mathcal{P}_s. \tag{4.5}$$

The procedure to obtain the empirical distribution and price intervals can be outlined as follows. First, it is assumed that some values for γ and c denoted as γ^* and c^* have been chosen previously. Also, the value of the current market state $z(t)$ and the desired percentiles $\underline{\alpha}$ and $\overline{\alpha}$ are known. The procedure starts by computing the value of the dissimilarity function for the given γ^* and $z(t)$ for all the possible values of $p(t+k)$ (i.e., $\forall \bar{p}_i \in \mathcal{P}_s$). These values of J_{γ^*} are then used to compute the empirical probability density function for the given γ^*, c^* and $z(t)$ for all the possible values of $p(t+k)$. Using these computations it is possible to build the aforementioned empirical distribution of $p(t+k)$, and then find the desired percentiles to build the price interval, and also the median to be used as the price forecast. These steps are formally described in algorithm 2.

Algorithm 2 k-step probabilistic price interval forecasting.

```
Input: $DB_k$, $\mathcal{P}_s$, $\gamma^*$, $c^*$,$z(t)$, $\underline{\alpha}$ and $\overline{\alpha}$.
Output: $\hat{p}(t+k)$ and the price interval $[p_{\underline{\alpha}}, p_{\overline{\alpha}}]$.
1: Build $\Omega(z(t))$ as in Algorithm 1.
2: Compute the dissimilarity function of definition 4.1 $J_{\gamma^*}(z(t), \bar{p}_i, \Omega(z(t))$ for all $\bar{p}_i \in \mathcal{P}_s$.
3: Using the previously computed values of the dissimilarity function, build the empirical distribution by computing $\text{ecp}_{\gamma^*,c^*}(z(t), \bar{p}_i, \Omega(z(t))$ for all $\bar{p}_i \in \mathcal{P}_s$ using the approximation given in (4.1).
4: Find the desired upper percentile $p_{\overline{\alpha}} \in \mathcal{P}_s$ using the approximations given in (4.2) and (4.3) with $z(t)$, $\Omega(z(t))$,$\gamma^*$ and $c^*$. In the same way, find the desired lower percentile $p_{\underline{\alpha}} \in \mathcal{P}_s$ using (4.4) and (4.5). Finally, using both methods with $\overline{\alpha} = \underline{\alpha} = 50$ obtain $p_{\overline{50}}$ and $p_{\underline{50}}$ and compute the median as $p_{50} = \frac{p_{\overline{50}} + p_{\underline{50}}}{2}$.
5: Return the desired interval $[p_{\underline{\alpha}}, p_{\overline{\alpha}}]$ and the price forecast $\hat{p}(t+k) = p_{50}$.
```

Note that the database can be updated as new market data is available. This, together with the use of local data, makes this strategy adaptive as in section 4.2.

There are different ways of chosing the values γ^* and c^*. A possibility could be to implement some form of local search that would find the values of c and γ that minimize the prediction error in a validation set or even maximize the revenue when using the forecast and price intervals in a trading strategy. However, these strategies would not give the desired probabilistic guarantees on the computed price intervals. Thus here it is proposed to use a maximum likelihood estimation procedure presented in [Carnerero et al., 2020] and modified to use local data.

The algorithm needs the sets of possible values of γ and c, denoted as Γ and C. The sets can be chosen as sets of N_γ and N_c numbers from a grid in the intervals $[\gamma_{min}, \gamma_{max}]$ and $[c_{min}, c_{max}]$ where the extreme points of these intervals can be chosen directly as tuning parameters (e.g, they could be chosen using cross-validation with a test set). On the other hand, N_γ and N_c should be set in relation to the computing power available.

The procedure starts by computing the dissimilarity function for all the possible combinations of values of Γ and $\mathcal{P}_s$ and for every entry in the database using local data. Then, with these values of the dissimilarity function, the ecp is used to compute the empirical distribution for all the combinations of γ, c and market states in DB_k. After this, the desired percentiles are computed for each of the previously built distributions. Then, for every combination of $\gamma \in \Gamma$ and $c \in C$, the number of prices in DB_k that fall outside the quantiles of its distribution (that is, the number of quantile violations) are computed. These numbers are used to associate to every $\gamma \in \Gamma$ the greatest $c_\gamma \in C$ for which the percentage of violations of both lower and upper quantile is less than $\underline{\alpha}$ and greater than $\overline{\alpha}$ respectively. Finally, the optimal γ^*, c^* are chosen as the one combination among all the previously computed (γ, c_γ) that maximizes the likelihood ratio. Algorithm 3 describes formally this procedure.

Algorithm 3 Computation of γ^* and c^*.

```
Input: $DB_k$, $\mathcal{P}_s$, $\Gamma$, $C$, $\underline{\alpha}$, $\overline{\alpha}$.
Output: $\gamma^*$ , $c^*$.
1: For all the possible combinations of $\gamma \in \Gamma$, $z_l \in DB_k$ and $\bar{p}_i \in \mathcal{P}_s$ compute the
   dissimilarity function $J_\gamma(z_l, \bar{p}_i, \Omega(z_l))$.
2: For all possible combinations of $\gamma \in \Gamma$, $c \in C$ and $z_l \in DB_k$ build its associated empirical
   distribution computing $\mathrm{ecp}_{\gamma,c}(z_l, \bar{p}_i, \Omega(z_l))$ for all $\bar{p}_i \in \mathcal{P}_s$ using the approximation
   given in (4.1).
3: For all $\gamma \in \Gamma$, $c \in C$, $z_l \in DB_k$ and their associated empirical distribution find the
   desired percentiles as in algorithm 2, i.e, for every combination find $p_{\overline{\alpha}} \in \mathcal{P}_s$ using (4.2)
   and (4.3) and $p_{\underline{\alpha}} \in \mathcal{P}_s$ using (4.4) and (4.5). Save the $p_{\underline{\alpha}}$ values in a vector denoted as
   $\underline{\phi}_{\gamma,c} \in \Re^{N_{DB}}$ and the values $p_{\overline{\alpha}}$ in a vector denoted $\overline{\phi}_{\gamma,c} \in \Re^{N_{DB}}$.
4: For every $\gamma, c$ compute the number of prices $p_l \in DB_k$ falling outside the interval
   defined by $[\underline{\phi}_{\gamma,c}(l), \overline{\phi}_{\gamma,c}(l)]$. Denote such numbers as $\underline{v}_{\gamma,c}$ and $\overline{v}_{\gamma,c}$.
5: For each $\gamma \in \Gamma$, select the greatest $c_\gamma \in C$ for which
   $$\frac{100\underline{v}_{\gamma,c_\gamma}}{N_{DB}} \leq \underline{\alpha} \text{ and } \frac{100\overline{v}_{\gamma,c_\gamma}}{N_{DB}} \geq \overline{\alpha}.$$
6: Compute
   $$\gamma^* = \arg\max_{\gamma} \sum_{l=1}^{N_{DB}} \log\Big(\mathrm{ecp}_{\gamma,c_\gamma}(z_l, p_l, \Omega(z_l))\Big).$$
   using for every $\gamma$ considered the $c_\gamma$ selected in the previous step. The optimal value of
   $c^*$ is the $c_\gamma$ selected in step 3 for $\gamma^*$.
```

There are other ways to compute the optimal γ^* and c^* with probabilistic guarantees, such as algorithm 2 in [Carnerero et al., 2020] in which the interval length is penalized aiming to smaller intervals.

Finally, note that although algorithms 2 and 3 have a higher computational burden than algorithm 1, both are highly parallelizable as many of the operations performed on every combination of data and parameters are independent of each other. In this way large data sets, which are readily available by the stock market data providers, can be used. Furthermore, the computation of the optimal γ^* and c^* does not have to be repeated if the database is updated until the amount of updated data becomes a significant fraction of the database.

4.4 Forecasting the Dow Jones Industrial Average index

The proposed approaches have been used in the problem of predicting the daily closing prices and price intervals for the Dow Jones Industrial Average index. The dataset was obtained from the data provider Bloomberg and is composed of the daily closing price of the Dow Jones Index from 2005 to mid-2016. The data was divided into a database DB, from 2005 to 2014, and a testing dataset, from 2015 to mid-2016. This latter period has been chosen because there is not a clear market trend (bullish or bearish) that would make forecasting easier. To lower the noise, all the raw prices in the database have been smoothed using a 5-day Exponential Moving Average (EMA), which can be computed as:

$$p^d_{EMA}(t) = \frac{2}{d+1}p(t) + (1 - \frac{2}{d+1})p^d_{EMA}(t-1),$$

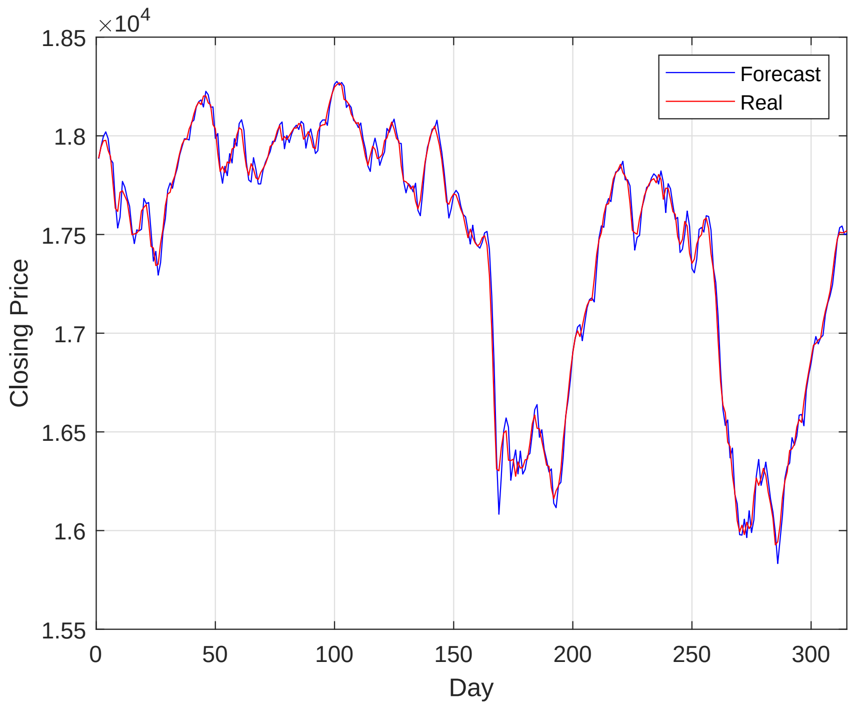
with $EMA_D(0) = p(0)$ and being $d = 5$ in this case. Note that the smoothing applied here is very light as the usual values of d for short-term forecasting are the 12 and 26 day EMA [Upadhyay et al., 2016]. This would preserve fast price fluctuations although makes forecasting more difficult.

The market state $z(t)$ has been chosen to be composed of the last ten days prices smoothed using the 5-day EMA approach, as well as the 5-day and 10-day relative difference percentage of unsmoothed prices (RDP) [Thomason, 1999] i.e.,

$$RDP_d(t) = 100\frac{p(t) - p(t-d)}{p(t)},$$

being d equal to 5 and 10 respectively.

The approach of section 4.2 has been applied to the case study to forecast the closing prices for up to 5 days, that is a full week of market sessions. The size of $\Omega(z(t))$ was $N = 250$ and $\gamma = 0$. The forecast and real prices are seen in figures 4.1 to 4.3. It can be seen that the forecast is quite accurate for the first sessions and, as expected, it becomes worse as k grows.



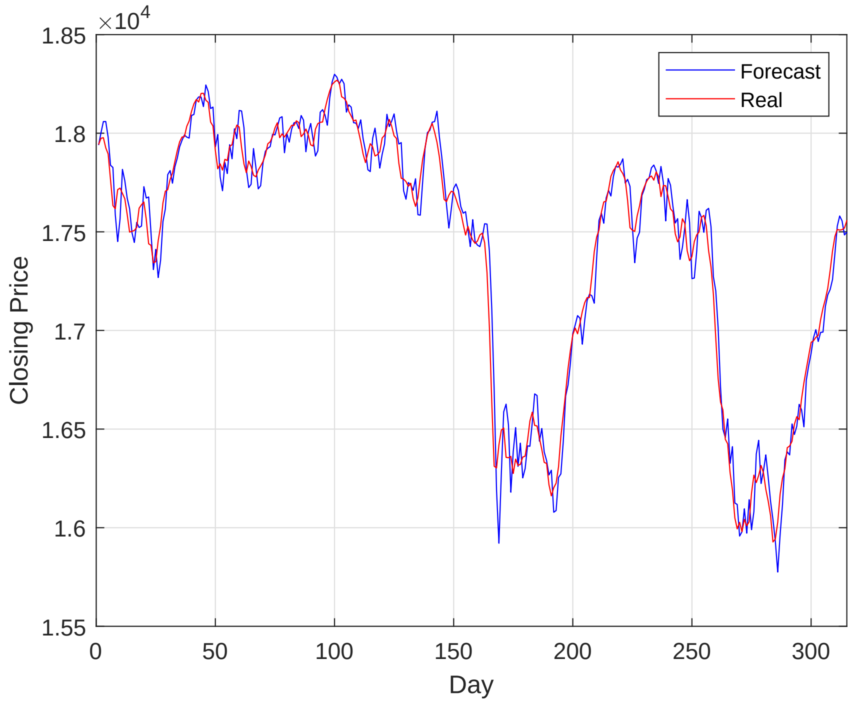


Figure 4.1: Forecasted and real prices (5-day EMA) for 1 day and 2 days forecasting.

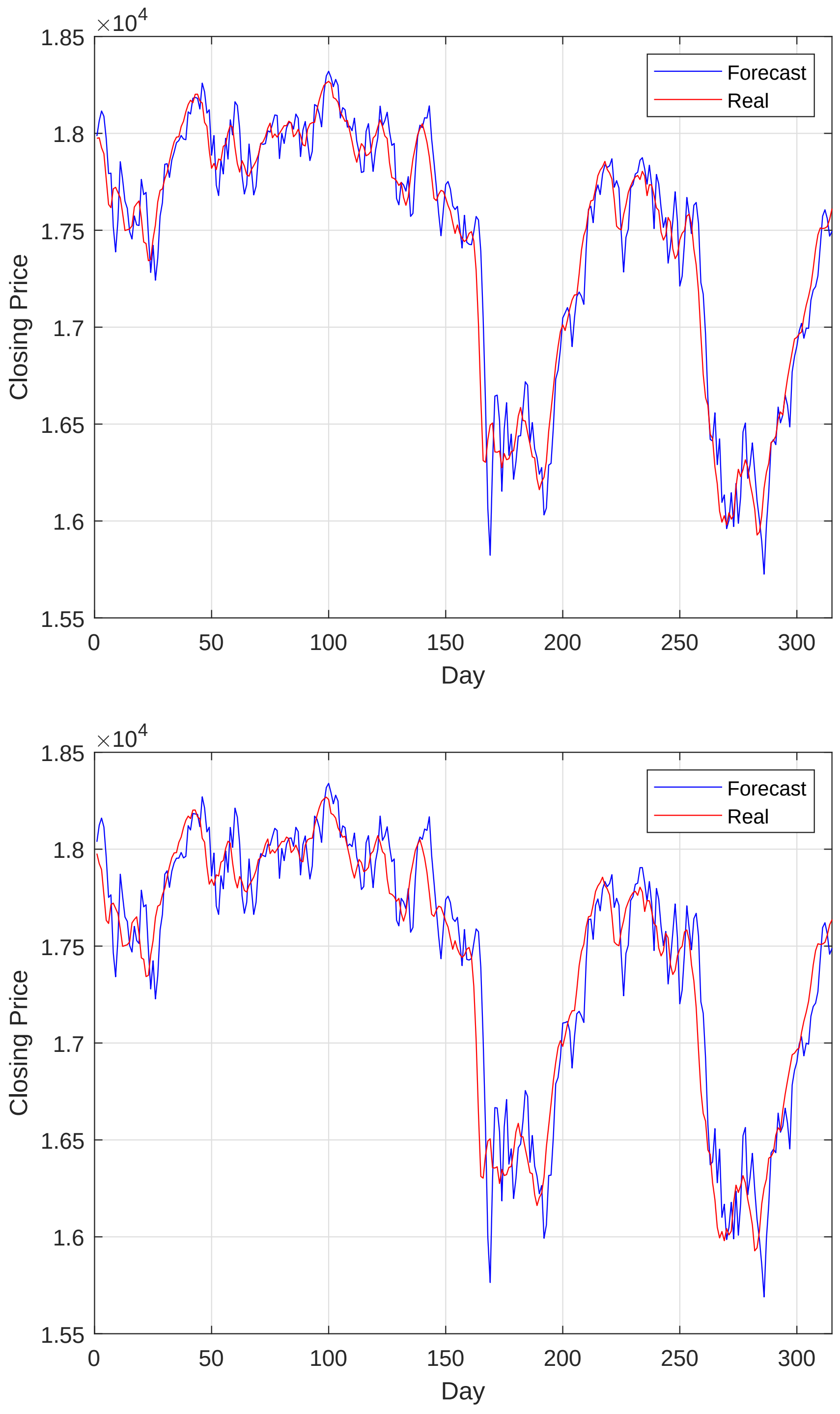


Figure 4.2: Forecasted and real prices (5-day EMA) for 3 and 4 days forecasting.

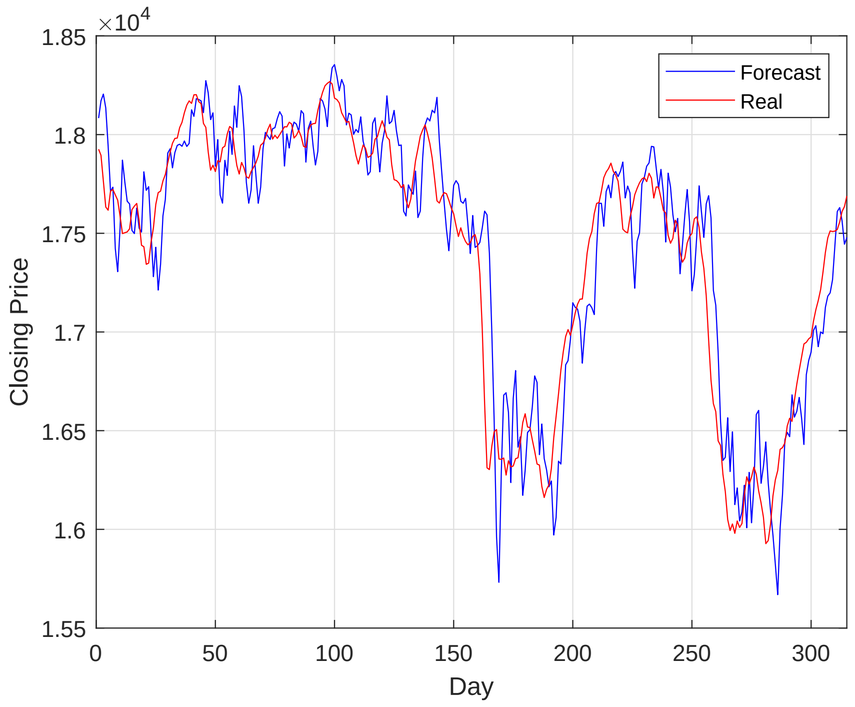


Figure 4.3: Forecasted and real prices (5-day EMA) for 5 days forecasting.

The approach for price interval forecasting has been also applied to the case study. The parameters have been $N = 250$, $N_p = 1000$, $p_{min} = 6684.3$, $p_{max} = 19445$, $N_\gamma = 10$, $\gamma_{min} = 0$, $\gamma_{max} = 5$, $N_c = 60$, $c_{min} = 0.25$, $c_{max} = 15$. The 10-th and 90-th percentiles were chosen for the price intervals, thus the probabilistic specification is that the intervals contain the real price is 0.8. The results obtained are shown in figures 4.4 to 4.6 in which the price intervals are represented as envelopes. It can be seen that although the price intervals are quite tight for $k = 1$, they grow as k rises. This is congruent with the fact that for farther prediction horizons the uncertainty on the forecasting is greater. Note also that sometimes the real price is not inside the computed price interval. This is also congruent to the fact that it should fall outside of the interval about 20% of the times.

Even if the results obtained seem correct at a glance, it is necessary some form of validation. Thus, the results have been validated in relation to a persistence predictor, i.e., martingale, that has been used as a baseline approach forecasting the prices as $\hat{p}(t + k) = p(t)$. Furthermore, a multi-layer perceptron (MLP) with 20 neurons in the hidden layer and trained with the Levenberg-Marquardt rule has been also used as a baseline approach. Table 4.2 shows the mean squared errors (MSE) for proposed and baseline approaches. It can be seen that the approach proposed in section 4.2 presents the lower MSE of all approaches and that the forecast using the strategy of section 4.3 is the second best for k up to 4. Another parameter to be studied is the dispersion of errors. Table 4.3 shows the standard deviations of the errors for all approaches. It can be seen that, as in the case of the MSE, the approach of section 4.2 has the tighter errors in all cases and that the forecasting using the median of the price distribution is the second best for k up to 4. Thus, the errors are expected to be smaller with the proposed

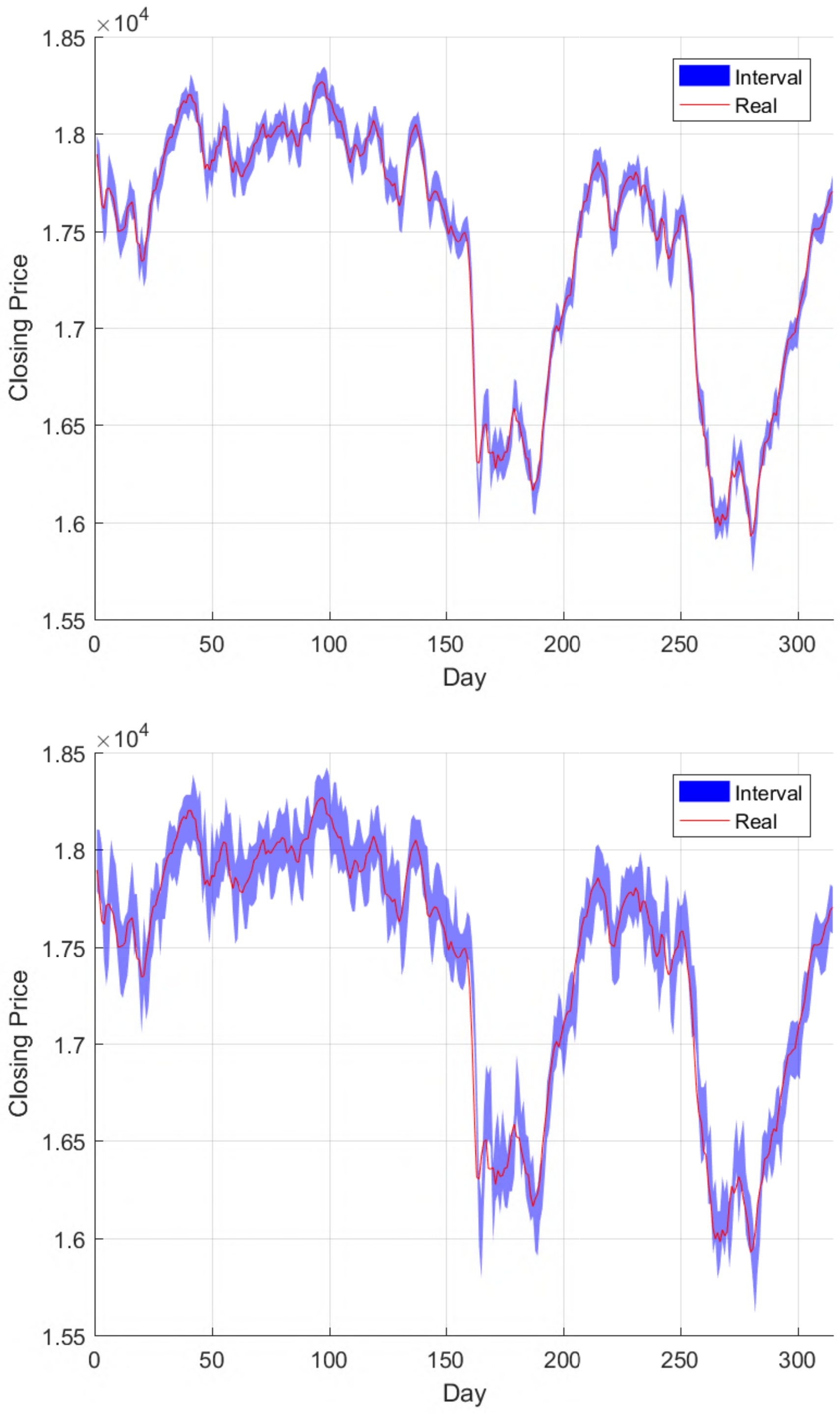


Figure 4.4: Price intervals for 1 to 2 days (5-day EMA).

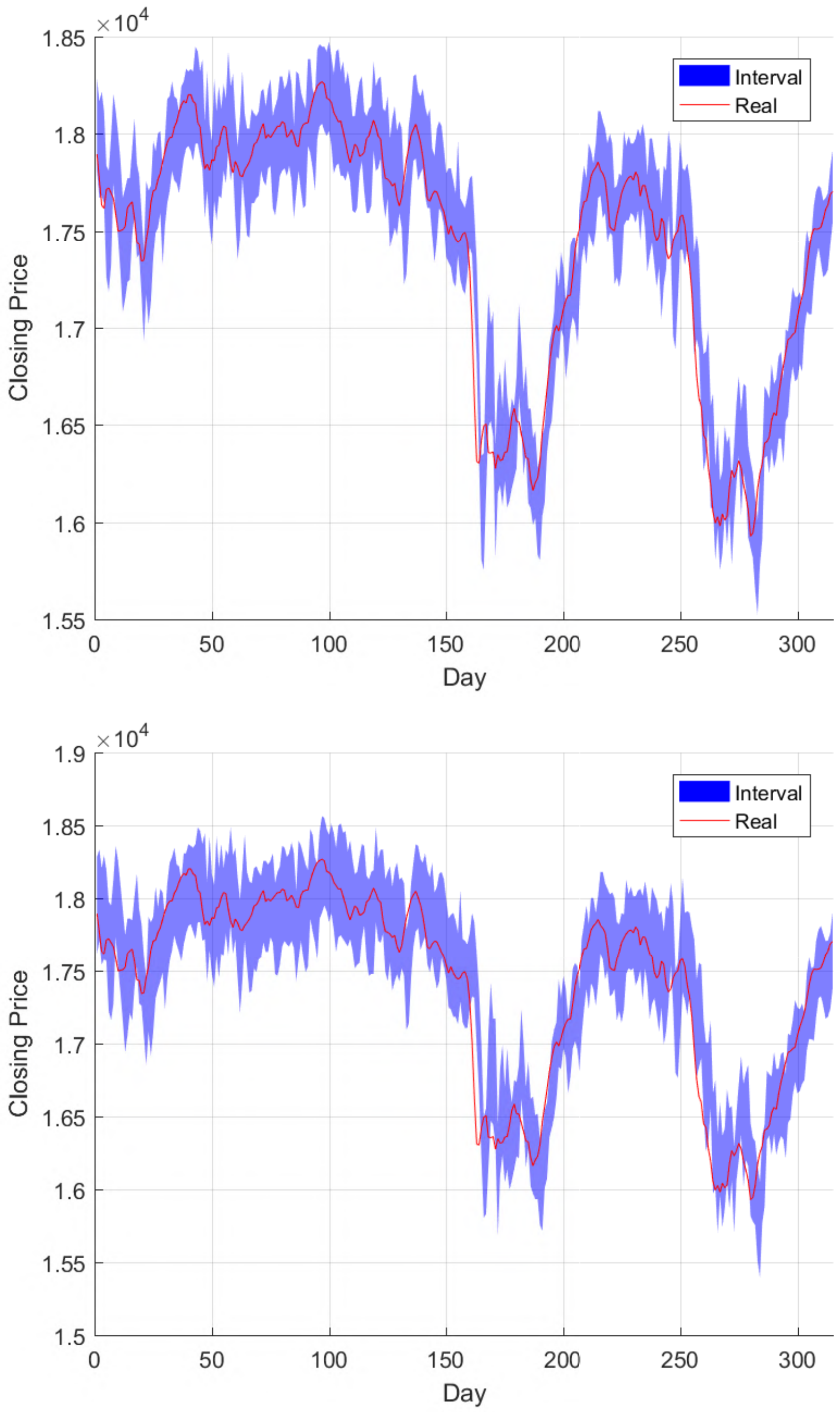


Figure 4.5: Price intervals for 3 to 4 days (5-day EMA).

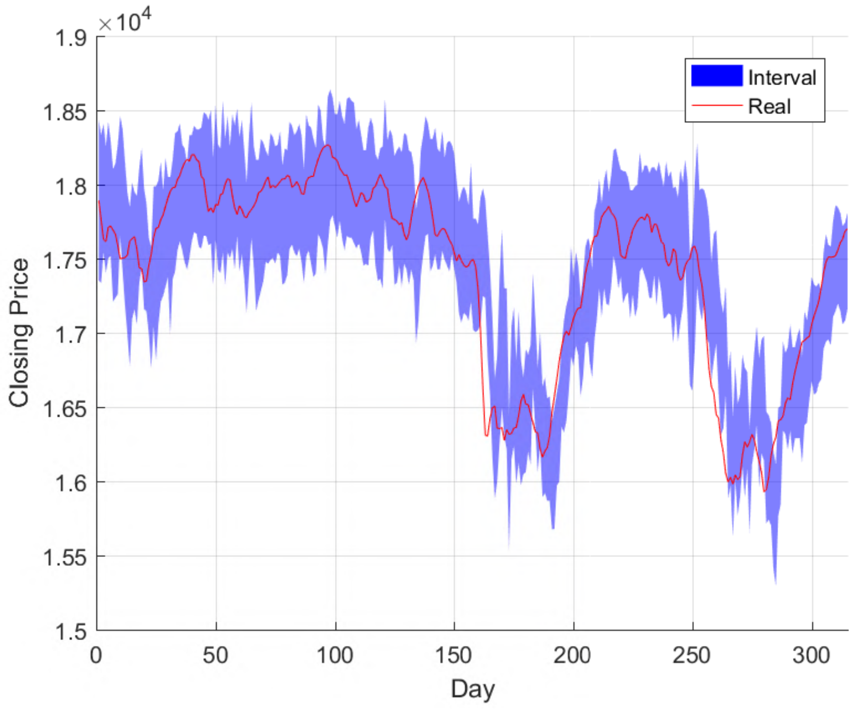


Figure 4.6: Price intervals for 5 days (5-day EMA).

strategies and more close to their mean values. This also results in lower uncertainty on the quality of the prediction. Furthermore, the results show that the approach of section 4.2 is complementary to that of section 4.3 producing better forecasts with a much lower computational burden. In fact, when implemented in Matlab on an Intel Core i7-4790 CPU computer, the computation time for the strategy of section 4.2 was 0.0037 seconds. On the other hand, the implementation of Algorithm 2 took 0.3997 seconds on the same computer, whereas the implementation of algorithm 3 required 4.5 hours to find γ^* and c^*. Note, however, that γ^* and c^* are computed only once provided that the database does not suffer major changes.

Table 4.2: MSE obtained using the proposed and baseline approaches (5-day EMA)

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	3,294.8	3,520.5	3,577.0	5,296.4
2	12,433.2	13,607.3	14,190.8	16,867.4
3	26,659.0	28,574.7	30,829.6	31,975.0
4	45,475.9	46,931.0	50,794.9	49,072.4
5	66,859.6	71,300.9	77,017.5	66,943.4

Although the MSE and standard deviation is better for the approach of section 4.2, it is practically equal to the MSE of the persistence predictor for $k = 5$. The reason for this is that as the prediction horizon k grows, the price time series becomes more similar to a random walk, making persistence predictors a good choice for price forecasting. This is more evident when the smoothing of the prices is quite light, like in the previous

Table 4.3: Standard deviation of the errors (σ); 5-day EMA

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	57.4	59.4	59.9	72.8
2	111.5	116.4	119.3	130.0
3	163.1	168.2	175.6	178.9
4	212.9	215.7	224.9	221.6
5	257.8	266.4	276.0	258.8

simulations. More typical periods in the EMA smoothing (12 and 26 days are common in stock trading for short term forecasting) show how the proposed approaches make a better job predicting the price trend than persistence predictors. Tables 4.4 and 4.5 show the MSE and standard deviations for all the approaches when using a 15-day EMA smoothing. These tables show that in this case the proposed approaches have always lower MSE and tighter errors. Furthermore, the MLP is also better than the persistence for all k, whereas in the case of the lighter smoothing it was worse for k equal to 4 and 5.

Table 4.4: MSE obtained using the proposed and baseline approaches (15-day EMA)

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	469.2	486.8	606.8	1,531.7
2	2,129.4	2,313.2	2,492.3	5,642.4
3	5,386.7	5,907.5	7,538.6	11,944.6
4	10,596.6	11,554.2	13,629.2	20,151.2
5	17,745.7	19,856.5	24,768.5	29,953.1

Table 4.5: Standard deviation of the errors (σ); 15-day EMA

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	21.6	22.1	24.6	42.1
2	45.9	47.9	49.9	78.7
3	72.8	76.0	85.6	113.2
4	101.8	105.9	114.2	145.9
5	131.4	138.5	152.8	177.1

The proposed strategies can also be used to forecast intraday stock prices. The Dow Jones Industrial Average prices from 06/03/2020 to 11/03/2020 have been considered as an example. Tables 4.6 and 4.7 show the MSE and standard deviation values when forecasting half-hourly prices from this period. In these tests the longest forecasting horizon was 3.5 hours, thus k varies from 1 to 7. A 2.5-hour EMA has been used to smooth the prices and the structure of market state $z(t)$ is the same as before but changing daily prices and RDP for their half-hourly counterparts. It can be seen that the strategy of section 4.2 performs as expected and that, in this case, there is no need for further smoothing to keep the performance better than the persistence predictor. On the other hand, the strategy of section 4.3 has a higher MSE than the MLP (but lower than the persistence predictor). Note, however, that the strategy of section 4.3 is used to produce interval forecasts rather than price forecasts.

Table 4.6: MSE obtained using the proposed and baseline approaches with intraday half-hourly prices (2.5-hour EMA)

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	1,034.9	1,153.1	1,037.7	2,103.9
2	4,406.9	4,816.6	4,498.1	7,335.2
3	10,233.0	11,527.9	10,589.7	14,584.5
4	17,423.8	19,859.2	18,599.3	22,933.7
5	25,453.4	29,444.1	27,880.3	31,796.9
6	34,002.5	39,619.5	37,676.9	40,947.9
7	43,086.0	49,990.4	49,331.7	50,124.3

Table 4.7: Standard deviation of the errors (σ); 2.5-hour EMA

k	**Proposed Section II**	**Proposed Section III**	**MLP**	**Persistence**
1	32.2	33.5	32.1	45.9
2	66.4	68.2	66.9	85.7
3	101.2	104.6	102.6	120.9
4	132.0	136.8	135.9	151.6
5	159.5	165.8	166.4	178.5
6	184.3	191.6	193.3	202.5
7	207.4	215.4	221.2	224.1

On the other hand, the previous results and baseline approaches are not useful to validate the price interval forecasting obtained using the strategy of section 4.3. In order to do so, the well-known quantile regression has been chosen as a baseline approach to validate the interval forecasting. Table 4.8 shows the empirical probabilities and interval width of the proposed strategy and quantile regression using the same data and theoretical probability (0.8). It can be seen that the intervals computed using the proposed approach contains the real price with a higher probability than the specified one, whereas the quantile regression produces tighter intervals that do not meet the specified probability for any k. Thus, the quantile regression fails in this case. However, in the case of the intraday dataset both strategies work well (see table 4.9), being the proposed strategy a bit more conservative. Note that some form of tightening could be used with the proposed approach to make the intervals narrower while meeting the probability in practice, but the resulting intervals will not have the probabilistic guarantee of the approach.

Table 4.8: Empirical probability of the real price to be contained in the computed intervals (the theoretical one is 0.8) and average width of the price intervals using the proposed approach and quantile regression.

	Empirical probability		**Average interval width**	
k	**Proposed**	**Q. regression**	**Proposed**	**Q. regression**
1	0.8679	0.6006	162.4802	99.8091
2	0.8459	0.6101	315.7218	190.5443
3	0.8553	0.6321	477.8004	287.5774
4	0.8648	0.6761	638.8747	397.3780
5	0.8805	0.6950	813.5661	489.7816

Table 4.9: Empirical probability and average width of the price intervals using the proposed approach and quantile regression (intraday dataset).

	Empirical probability		**Average interval width**	
k	**Proposed**	**Q. regression**	**Proposed**	**Q. regression**
1	0.8820	0.8614	88.5662	74.3958
2	0.8846	0.8373	178.6064	159.0279
3	0.8635	0.8279	275.3878	238.7693
4	0.8571	0.8214	378.3202	313.8665
5	0.8358	0.8119	443.9647	389.4412
6	0.8293	0.8084	516.9591	457.2965
7	0.8348	0.8138	576.9186	516.4660

Finally, the growing values of the forecasting error and its dispersion together with the effect of smoothing the prices suggest, that although not a perfect random walk for very short term forecasting, as the prediction horizon grows, the price time series becomes more difficult to forecast, gradually approaching to a random walk.

Chapter 5

Receding horizon optimization of large trade orders

5.1 Introduction

Stock trading is becoming an increasingly complex field as investors try to maximize their profits, and reduce their risks with increased emphasis in recent years on optimal execution. After an investor has decided to buy or sell stocks, there remains many options on how to execute this trade. There are two basic types of trading orders: 1) market orders and 2) limit orders. A market order is an order to be executed immediately at the prevailing market price. The focus on this type of order is speed rather than price optimization. Limit orders on the other hand focuses on price efficiency. A limit order necessarily has an associated price over (below) which the buy (sell) order cannot be executed. It should be noted that the usual convention for limit prices orders is described by the expression "or better". In other words, if the limit price on a buy limit order is 15 that the maximum price that the investor would pay, but the actual transaction price could be any price below that number. There is a tradeoff between speed of execution and price optimization.

It is also very frequent to split orders into smaller orders in an attempt to not to impact the market. For instance, a buy order to purchase a large amount of stock could push the price of the stock up if executed in one single block. The market impact of that order can be potentially de reduced by splitting that single order in smaller orders, and execute them over time rather than in one go. There are clearly many ways in which this can be accomplished. One of the simplest approaches is called "Time Weighted Average Price", commonly referred as TWAP. A TWAP order will split the order in blocks of shares of the same size, that are then executed at regular time intervals. For example, a TWAP order to buy 1,000 shares could be divided into 10 orders of 100 shares, each to be executed every 5 minutes. Therefore the complete trade takes 50 minutes to complete. An investor using a TWAP order needs to specify the time period over which the order needs to be executed. In the previous case this would be the previously mentioned 50 minutes. A TWAP trade can use market or limit price orders. In a TWAP market order there is almost certainty of execution as the transactions at each time interval is carries out at the prevailing market price. In the case of a limit order there is no certainty of execution. This is better illustrated with an example. Let assume than a TWAP limit price order is received at 10:00 with a time window of one 60 minutes. Let also assume that the order is to buy 600 shares of a given stock and that the limit price is 15 RMB and the current price is 14 RMB and that the approach followed uses 6 time windows of 10 minutes each. The trader executing the transaction will send a buy order every 10 minutes. The trader will continue executing trades every 10 minutes, as long as the price remain below the 15 RMB price. If the price however, exceeds 15 RMB no more trades will be executed until the price comes back within the initially specified limit. Clearly, with a TWAP limit order there is a risk that the trade cannot be completed. A TWAP trade is one of the simplest orders. It implicitly assumes that there is a constant trading volume for the stock which will not be typically the case.

A more sophisticated trade is a “Volume Weighted Average Price”, commonly referred as VWAP. This type of order is very common as it represents around 50% of all the institutional investors trading [Bofa, 2007]. In a VWAP order, rather than slicing the original order in smaller trades of equal size (equal number of shares), a forecasted traded volume for the desired interval is estimated. The trade is sliced into smaller trades, executed at regular time intervals. The size of each (sliced) transaction is proportional to the forecasted volume for that time interval. A critical step in this approach is to be able to generate accurate volume forecasts. Something to take into account is that stock market conditions can substantially change from historical values so it is important to adjust the volume market forecast, likely obtained using historical data, with the actual trading values obtained in that day.

A common variation of VWAP is VWAP Max, which is similar but with the further constraint of not exceeding a maximum percentage of the total traded volume for that stock, at any point in time during the trading. For example, a VWAP Max can have a 10% maximum volume meaning that the trades executed on behalf of the client cannot exceed 10% of that total traded volume (during that time) for that stock. Finally, Enhanced VWAP and Enhanced VWAP Max are not at fully standardized terms and refer to variations from the VWAP Max trying to further optimize the price and/or tracking error. This could entail modifications to the volume and/or price forecasts to try to improve the speed or to try increase the chance of completing the transaction.

The objective of this chapter is to design trade algorithms to execute the trade at the best possible price, while following the investor trade instruction, and subject to the constraints that the different order types impose. The focus of this chapter is on large orders that must be splitted to lower their impact on the market, as the Enhanced VWAP and Enhanced VWAP Max orders which are also relatively sophisticated requiring volume as well as price forecasts.

There is relatively limited existing literature in the topic of optimal execution of splitted trade orders using learning techniques, with more papers covering stock forecasting by means of different techniques like neural networks [White, 1988] or deep learning [Wang et al., 2018], support vector machines [Hasibuan et al., 2019, Trafalis and Ince, 2000], adaptive line combiners [Seidy, 2016] or local data based techniques like those of chapter 4.

There are however some interesting articles in the field of trade execution optimization such as for instance [Cui et al., 2009]. In this article the authors proposed a genetic algorithm to optimize a limit order book used for price formation in an artificial stock market. Genetic algorithms are also used in [Lim and Coggins, 2005] to generate trading strategies, not based on forecasting, that are back-tested against historical data of Australian Stock Exchange. Recurrent neural networks have been used in [Dixon, 2018] to predict price-flip events in limit order books, by classifying sequences of observations of the book depths and market orders.

Forecasting the traded volume has also been used in [Białkowski et al., 2008] to improve the execution of VWAP orders. That is, by forecasting the traded volume one can track the VWAP price matching it at the end of the chosen time window. On the other hand, [Konishi, 2002] derives analytical solutions of a static optimal execution strategy of a VWAP trade, in which the optimal execution strategy can be calculated by an iteration of a single variable optimization, rather than by a multivariable optimization. In that work the market is modelled using non-anticipating and brownian motion processes.

[Crawford et al., 2018] presented a high frequency trading system based on moving averages and particle swarm optimization, used to determine the trading sequence that maximizes the net returns over a series of consecutive time steps. Particle swarm optimization has been also used by [Ding et al., 2015] to train a kernel based nonlinear predictor that was also applied to forecast the VWAP price in the Shanghai market. Optimal control methods have been also considered for generating the sequence of suborders in splitted large orders. For instance [Pemy, 2012] presented an optimal VWAP algorithm based on the linear quadratic regulator (LQR) subject to limits in the size of the suborders. The linear model for the stock prices is based on brownian motion, a type of model that has also been used by [Stace, 2007] to find the stock prices also in the context of VWAP operations. Hamilton-Jacobi-Belmann methods and in general variational calculus have been used in [Bertsimas and Lo, 1998, Forsyth, 2011, Forsyth et al., 2012] based on brownian motion and random walks models. In a recent paper, [Mitchell et al., 2020] studied optimal VWAP strategies using unconstrained optimization on models based on the assumption that the stock prices can be modelled as martingales and the traded volume as autoregressive processes. [Ye et al., 2014] used a dynamic time series approach ARFIMA to forecast intraday trading volumes in the Chinese equity market, applied to VWAP tracking, obtaining better results than static approaches. VWAP tracking has also been tackled in [Liu and Lai, 2017], where an interesting combination of historical averages and SVM to forecast intraday trading volumes in the gold and S&P 500 futures markets was used. Another interesting approach to forecast trading volumes in a VWAP tracking context has been presented in [Li and Ye, 2013]. In that paper the authors used the fast Fourier transform algorithm to identify the periodic and the non-periodic part of the trading volume, using historical values of 50 stocks contained in the Shanghai 50 stock index. A similar approach is followed in [Song et al., 2014].

In this chapter it is proposed the use of dynamic optimization over a finite horizon and based on price and volume forecasts, to obtain optimal sequences of suborders to fulfill large trade orders. The optimal sequence is computed by minimizing a cost index, in which several terms are taking into account. The technique is based on solving a optimization problem each time bucket of the time window in which the order has to be executed. Thus at each time bucket a complete sequence of suborders for the remaining time window is obtained, but only the first component of the sequence is effectively used. This strategy is similar to the feedback receding horizon or predictive control strategy used in automatic control [Rawlings et al., 2017], but with a shrinking prediction horizon. In the optimization, besides the price and volume forecasts, the trading impact factor is considered along with terms related to forecasting accuracy and the number of suborders (which can be useful when trade commissions and fees are taking into account). The optimization, which uses integer variables, is carried out by means of a particle swarm algorithm tailored to the receding horizon nature of the proposed strategy. The forecasts are based on the local data forecasting methods of chapter 4. Market and limit orders both in regular and Max formats are considered in the chapter. Finally, as a case study, stocks from the onshore Chinese A-share market are used to show the performance of the proposed strategy.

The rest of the chapter is outlined as follows: section 5.2 describes the problem statement for each type of order. Section 5.3 presents the optimization algorithm. Section 5.4 presents the results of applying the proposed technique to the case study.

5.2 Problem statement

The objective of this chapter is to design a strategy to execute large stock orders that, in order to limit its impact on the market, have to be splitted in a number of smaller suborders. Thus, an order to buy $M \in \mathbb{N}_0$ shares will be executed by splitting the order into up to N_p suborders $m(t+k) \in \mathbb{N}_0$ with $k \in \{1, \ldots, N_p\}$ such that

$$\sum_{k=1}^{N_p} m(t+k) = M. \tag{5.1}$$

The proposed strategy will compute the splitting of the original order in an optimal way, that is, achieving the lowest price. The strategy will be based on forecasting both the price of the stock and its total traded volume over the time window, defined by N_p, i.e., from $t+1$ to $t+N_p$. Both forecasts will be used to compute a performance index that should be optimized. Thus the strategy will rely on an optimization problem in which a performance index V will be minimized. The formulation of V must weigh certain aspects of concern such as the total cost of the order, but also the impact on the market of each of the suborders. Besides that, given that forecasting accuracy is worse for long prediction horizons, a term that would penalize the placement of big suborders far into the time window will be included in V . Finally, a term that will penalize higher number of suborders will also be considered. This can be useful if there are commissions and fees that are paid per order independently of the traded volume. It can be also useful to avoid a large dispersion of the suborders.

Let $\hat{p}(t+k|t)$ be the price forecast for $t+k$, $\hat{v}(t+k|t)$ the total traded volume forecast for $t+k$ and $m(t+k|t)$ the amount of shares to be bought at $t+k$, meaning the notation

$t+k|t$ that these values are computed at time t. Then the proposed performance index is:

$$\begin{aligned} V(\mathbf{m}(t),\hat{\mathbf{p}}(t),\hat{\mathbf{v}}(t)) \;=\; & \sum_{k=1}^{N_p}\left(\hat{p}(t+k|t)+\alpha\left(\frac{m(t+k|t)}{\hat{v}(t+k|t)}\right)^{\beta}\right)m(t+k|t) \\ & +\mu\sum_{k=1}^{N_p}\sigma^k m(t+k|t)+\iota\sum_{k=1}^{N_p}(m(t+k|t)>0), \end{aligned} \tag{5.2}$$

where α, β, μ, σ and ι are nonnegative tuning parameters and

$$\begin{aligned} \mathbf{m}(t) &= [m(t+1|t),\ldots,m(t+N_p|t)] \\ \hat{\mathbf{p}}(t) &= [\hat{p}(t+1|t),\ldots,\hat{p}(t+N_p|t)] \\ \hat{\mathbf{v}}(t) &= [\hat{v}(t+1|t),\ldots,\hat{v}(t+N_p|t)] \end{aligned}$$

the sequences of suborders, and price and traded volume forecastings respectively. Note that the first term represents the forecasted cost of executing the order, and that in this term a impact factor correction has been included. Impact factor correction represents the influence of the suborder volume in the price for that time bucket. Impact factors can be modelled as linear terms, like in [Pemy, 2012] or be more elaborate like the exponential form used here based on [Almgren, 2003]. The second term assign a greater cost to suborders that are far in the future because the prediction error grows with the prediction horizon. Note that for this effect the exponential weight σ^k must have $\sigma > 1$. Finally, the last term penalizes those orders that are splitted into many suborders more than those which are executed with a lower number of suborders. This can helpful if some fees or commissions grow as the number of suborders rises.

The proposed strategy aims to find the optimal sequence of suborders

$$\mathbf{m}^*(t) = [m^*(t+1|t), m^*(t+2|t),\ldots,m^*(t+N_p|t)], \tag{5.3}$$

that minimizes the performance index V over the time window N_p. The sum of all suborders must be equal to the total number of shares to be traded (M) thus the equality constraint

$$\sum_{k=1}^{N_p} m(t+k|t) = M. \tag{5.4}$$

must be taken into account. Also, as a measure to prevent the impact of big suborders, the size of each suborder must satisfy:

$$0 \le m(t+k|t) \le \frac{M}{c} \quad \forall k \in \{1,\ldots,N_p\} \tag{5.5}$$

with $c > 1$, being a typical value $c = 10$ (i.e., a suborder cannot excess a 10% of the whole order). Then, the optimal sequence of suborders will be obtained by solving the optimization problem:

$$\begin{aligned} \mathbf{m}^*(t) = \arg\min_{\mathbf{m}(t)} \; & V(\mathbf{m}(t), \hat{\mathbf{p}}(t), \hat{\mathbf{v}}(t)) \\ \text{s.t.} \;\; & (5.4) \text{ and } (5.5). \end{aligned} \tag{5.6}$$

Once problem (5.6) is solved one could apply the entire optimal sequence $\mathbf{m}(t)$ executing the suborders $m(t+1), \ldots, m(t+N_p)$ at the corresponding time buckets. This approach suffers from two related issues: first, as the time index k grows, the forecastings for $\hat{p}(t+k)$ and $\hat{v}(t+k)$ have a higher prediction error. Thus the suborders $m(t+k)$ will rely on progressively more inaccurate forecastings and thus the computed optimal value will differ from the ideal optimal value that could be computed if the real values of $p(t+k)$ and $v(t+k)$ would be known in advance. Second, as time advances, new real values of $p(t+k)$ and $v(t+k)$ are available, but they are not used. These new values could be used to obtain better predictions of the remaining time window, allowing the computations of suborders more closely to their ideal optimal values. Thus, in this chapter it is proposed to use a feedback receding horizon optimization strategy, typical of predictive control techniques [Rawlings et al., 2017], in which at each time bucket t problem (5.6) is solved to obtain $\mathbf{m}^*(t)$ using only $m^*(t+1|t)$, and discarding the rest of the sequence $\mathbf{m}^*(t)$. Algorithm 4 summarizes the proposed strategy for the case of market orders.

Algorithm 4 Optimal execution of large market orders.

```
Input: $M, N_p, \alpha, \beta, \mu, \sigma, \iota, c$.
 1: repeat
 2:    Compute $\hat{\mathbf{p}}(t)$ and $\hat{\mathbf{v}}(t)$.
 3:    Solve (5.6) to obtain $\mathbf{m}^*(t)$.
 4:    Wait for next time bucket.
 5:    if $m^*(t+1|t) > 0$ then
 6:       Send $m^*(t+1|t)$ to the market.
 7:       $M \leftarrow M - m^*(t+1|t)$.
 8:    end if
 9:    $N_p \leftarrow N_p - 1$.
10: until $M = 0$
```

Remark 2. *Note that the time window shrinks at each step, reducing the prediction horizon, thus allowing to work with progressively better forecasts. Thus, not only new real values of $p(t)$ are taken into account, but also the task of forecasting prices and volume is less demanding as time goes by.*

Limit orders can be also executed with the proposed strategy provided that some changes are taken into account. The main characteristic of limit orders is that the execution of a suborder is conditioned to being it under the price limit stated by the client. The result of this constraint is that limit orders do not have the guarantee of completion and therefore, the degree of execution of the original order after the time window expires can be lower than 100%. The changes needed to implement limit orders start with restricting the time instants in which the optimization problem will be solved, to those in which the price forecast is under or equal the stated price limit, discarding those that do not meet that limit. Let R be the set of time indexes for which the price forecast meet the price limit p_l, i.e.,

$$R = \{i_1, \dots, i_r\} \text{ such that } \hat{p}(t + i_j) \leq p_l \;\; j \in [1, r]. \tag{5.7}$$

Then the restricted suborder sequence and price and volume forecast sequences are defined as:

$$\begin{aligned}
\mathbf{m}_{\mathbf{R}}(t) &= [m(t + i_1|t), \dots, m(t + i_r|t)] \\
\hat{\mathbf{p}}_R(t) &= [\hat{p}(t + i_1|t), \dots, \hat{p}(t + i_r|t)] \\
\hat{\mathbf{v}}_R(t) &= [\hat{v}(t + i_1|t), \dots, \hat{v}(t + i_r|t)] \,.
\end{aligned}$$

Constraint (5.5) must be modified:

$$0 \leq m(t + i_j|t) \leq \frac{M}{c} \quad \forall i_j \in R \tag{5.8}$$

Furthermore, as stated before, it can happen that the order cannot be completed because the price forecast do not meet the price limit in enough time instants, i.e., if every suborder must be at least 10% of original order and the price forecast is only under the limit in 7 time instants, the original order would be completed at most at a 70%. Thus the equality constraint (5.4) would be substituted by

$$\sum_{i_j \in R} m(t + i_j|t) = M \;\; \text{if } r \geq c, \tag{5.9}$$

or by

$$\sum_{i_j \in R} m(t + i_j|t) = \frac{r}{c} M, \;\; \text{if } r < c \tag{5.10}$$

to take into account the case in which the order cannot be completely fulfilled. The optimization problem (5.6) would be then modified to

$$\begin{aligned}
\mathbf{m}^*{}_R(t) = \arg \min_{\mathbf{m}_R(t)} & \; V(\mathbf{m}_R(t), \hat{\mathbf{p}}_R(t), \hat{\mathbf{v}}_R(t)) \\
\text{s.t.} & \;\; (5.8) \;\text{ and} \\
& (5.9) \text{ or } (5.10).
\end{aligned} \tag{5.11}$$

Note that the price forecasts for the complete time window are known in advance to the solution of (5.11) so the choice between (5.9) or (5.10) can be easily made.

Algorithm 4 must be modified, because the price limit must be imposed not only in the computation of $\mathbf{m}^*{}_R(t)$ but also in the execution of the order. Furthermore, as suborders must be also applied only in the time instants in R, problem (5.11) must solved only if $i_1 = 1$. Algorithm 5 shows the steps required for limit orders.

Algorithm 5 Optimal execution of large limit orders.

```
Input: $M, N_p, \alpha, \beta, \mu, \sigma, \iota, c, p_l$.
 1: repeat
 2:    Compute $\hat{\mathbf{p}}(t)$ and $\hat{\mathbf{v}}(t)$.
 3:    Form $R$ as in (5.7), and $\hat{\mathbf{p}}_R(t), \hat{\mathbf{v}}_R(t)$.
 4:    if $i_1 = 1$ then solve (5.11) to obtain $\mathbf{m}^*{}_R(t)$.
 5:    end if
 6:    Wait for next time bucket.
 7:    if $i_1 = 1$ & $m_R^*(t+i_1|t) > 0$ & $p(t+1) \leq p_l$ then
 8:       Send $m_R^*(t+i_1|t)$ to the market.
 9:       $M \leftarrow M - m_R^*(t+i_1|t)$.
10:    end if
11:    $N_p \leftarrow N_p - 1$.
12: until $M = 0$
```

Finally, market and limit orders described so far can easily be executed under the "max" policy described in section 5.1 by imposing the additional constraint

$$m(t+k|t) \leq \frac{\hat{v}(t+k|t)}{c_{\max}} \quad \forall k \in \{1, \dots, N_p\} \tag{5.12}$$

to (5.6) or

$$m(t+i_j|t) \leq \frac{\hat{v}(t+i_j|t)}{c_{\max}} \quad \forall i_j \in R \tag{5.13}$$

to problem (5.11), being $c_{\max} > 1$ the parameter that adjust the limit on the volume fraction.

5.3 Optimization algorithm

In this chapter, it is proposed to use a Particle Swarm Optimization (PSO) algorithm [Kennedy and Eberhart, 1995] to solve problems (5.6) and (5.11) defined in section 5.2. The reasons for such a choice are the potentially high number of decision variables and the fact that they are integer. Specifically, the implementation of the PSO algorithm is of the PSO with "Constriction coefficients" or "Constriction factor" type [Clerc and Kennedy, 2002], with a tailored treatment of the suborder constraints. In a PSO algorithm, particles represents potential solutions of the optimization problem to be solved. That is, they are compounded of feasible values of the decision variables. Also, the effectiveness of the particles is evaluated by means of a cost function or fitness function (as it is known in the PSO literature). After computing the cost for each particle, particles evolve to another generation following certain rules that are specific for each PSO algorithm. Usually, the worst particles are attracted to better solutions whereas the best particles tend to stay in their place. In the case of the proposed PSO algorithm, the particles are attracted to the best solution found in their evolution and to the best global solution found among all the particles. This process is repeated until a sufficiently good solution is found, or an iteration or time limit constraint is met.

In the proposed algorithm, every particle s_i stores the following data:

- Current suborder sequence, $\mathbf{m_i} \in \mathbb{N}_\mathbf{0}^{\mathbf{N_p}}$.
- Current cost function value, i.e., $V_i = V(\mathbf{m_i}, \hat{\mathbf{p}}(\mathbf{t}), \hat{\mathbf{v}}(\mathbf{t}))$. Note that in the algorithm the forecasts of price and traded volumes are considered as input data, and therefore must be computed in advance.
- Best past suborder sequence that has been found through the evolution of s_i denoted as $\mathbf{m_i^b}$, and its associated cost function value, that is $V_i^b = V(\mathbf{m_i^b}, \hat{\mathbf{p}}(\mathbf{t}), \hat{\mathbf{v}}(\mathbf{t}))$.
- The computed update velocity, $\Delta\mathbf{m_i}$. Velocities will determine how the particles are updated through the iterations of the algorithm.

The steps of the proposed algorithm will be exposed in the the following. First it is necessary to give initial values to all the L particles in the particle set. This is done by means of a random generation of the vectors $\mathbf{m_i}$ of each particle s_i and the computation of its associated costs. In this initial values the best suborder sequence and cost of each particle will be obviously set to the initial $\mathbf{m_i}$ and its associated cost. Also. the algorithm needs to know the best cost so far, V^{best}, that it is initially set to infinity, so that it gets updated in the first iteration. Then the algorithm iterates through the following steps:

1. Compute the cost V_i for every particle s_i.
2. Update the best so far cost V_i^b and the best so far sequence $\mathbf{m_i^b}$ of each particle s_i if necessary (that is, if $V_i < V_i^b$).
3. Find the particle with the lowest cost and update the best so far global cost V^{best} if necessary, also saving the associated sequence of suborders in a variable denoted by $\mathbf{m^{best}}$.
4. Compute the update velocity $\Delta\mathbf{m_i}$ for every particle and update the particles. The computation takes into account mainly two terms, one attracts the particle to $\mathbf{m_i^b}$ and the other attracts the particle to $\mathbf{m^{best}}$, i.e., the velocity of a particle i is computed as

$$\Delta\mathbf{m_i} \leftarrow \chi\left(\mathbf{\Delta m_i} + \phi_\mathbf{1}\mathbf{r_1}\left(\mathbf{m_i^b} - \mathbf{m_i}\right) + \phi_\mathbf{2}\mathbf{r_2}\left(\mathbf{m^{best}} - \mathbf{u_i}\right)\right), \tag{5.14}$$

   where χ, ϕ_1 and ϕ_2 are constants with values 0.7298, 2.0500 and 2.0500 respectively, and $r_1 \in [0,1]$ and $r_2 \in [0,1]$ random numbers [Poli et al., 2007].

5. Update the sequence of orders of every particle s_i according to $\mathbf{m_i} \leftarrow \mathbf{m_i} + \mathbf{\Delta m_i}$, rounding the resulting $\mathbf{m_i}$ component-wise to the nearest natural number.

6. Feasibility verification and particle modification. As the update of particles do not take into account the problem constraints, feasibility verification and restoration have to be performed after updating the particles. Handling inequality constraints is straightforward, as it is only needed to limit the values of the suborders, according to the maximum and minimum allowed. Equality constraints like (5.4), on the other hand, require a more complex treatment, using the following strategy based on random perturbations of the sequence of suborders:

   (a) First, for every particle, compute

   $$d_i = M - \sum_{k=0}^{N_p} \mathbf{m_i(k)}, \tag{5.15}$$

   that is, the amount of shares for which constraint (5.4) is violated.

   (b) Then, for every particle, apply random perturbations to some randomly chosen time buckets in the prediction horizon N_p.

   (c) Check that all particles verify the inequality constraints and correct if necessary.

   (d) Compute again d_i, and repeat from 6b for those particles in which $d_i \neq 0$.

Once a stop condition is met (e.g., a total number of iterations or a limit in the change of V^{best}), the algorithm returns $\mathbf{m^{best}}$ as the solution. Algorithm 6 formally describes the steps of the proposed PSO strategy.

Finally, constraints (5.12) or (5.13), required for "max", can be handled easily by the algorithm in the same way as constraints (5.5) and (5.8).

5.4 Case Study: Chinese stock market

The proposed approach has been validated using stocks from the Chinese Stock Market. A brief description of some market rules of the Chinese Stock Market is given in the following. The Chinese A-share market (main board) has several practical restrictions. Stocks traded in the onshore Chinese A-share market can only be purchased in increments of 100 shares. The portfolio of an investor can actually hold what is commonly referred as "odd lots". For example, an investor can actually own 217 shares of a given stock. This is because of the result of corporate actions, such as stock dividends or conversion of convertible bonds. This odd lots can be disposed of, but there are limitations. The investors can sell the entire position (217 shares) or the odd lots (17 shares). The investor can also sell round lots, i.e., 100 shares or 200 shares. However, the investor will not be able to sell a different odd lot, such as for instance 8 shares. While this are limitations specific to the Chinese A-share market there are similar limitations in many other stock markets.

The focus of this chapter is on the continuous trading period. Every working day has two continuous trading sessions. The morning session, from 9:30 to 11: 30, and the afternoon session. Additionally, there are two auctions. An opening auction, from 9:15

Algorithm 6 PSO for large orders trading

```
Input: $L, c, M, N_p, \hat{\mathbf{p}}(\mathbf{t}), \hat{\mathbf{v}}(\mathbf{t})$
 1: Initialize the variables and particles with random generation of the $\mathbf{m_i}$.
 2: repeat
 3:   for $i = 1$ to $L$ do
 4:     Compute $V_i$ using (5.2).
 5:     if $V_i < V_i^b$ then
 6:       $V_i^b \leftarrow V_i$.
 7:       $\mathbf{m_i^b} \leftarrow \mathbf{m_i}$.
 8:     end if
 9:   end for
10:   Find $m = \arg\min_i V_i$.
11:   if $V_m < V^{best}$ then
12:     $V^{best} \leftarrow V_m$.
13:     $\mathbf{m^{best}} \leftarrow \mathbf{m_m}$
14:   end if
15:   for $i = 1$ to $L$ do
16:     Compute $\Delta\mathbf{m_i}$ according to (5.14).
17:     $\mathbf{m_i} \leftarrow \text{round}(\mathbf{m_i} + \boldsymbol{\Delta}\mathbf{m_i})$.
18:     for $k = 1$ to $N_p$ do
19:       if $\mathbf{m_i(k) > \frac{M}{c}}$ then $\mathbf{m_i(k) \leftarrow \frac{M}{c}}$.
20:       end if
21:       if $\mathbf{m_i(k) < 0}$ then $\mathbf{m_i(k) \leftarrow 0}$.
22:       end if
23:     end for
24:   end for
25:   for $i = 1$ to $L$ do
26:     $f \leftarrow 0$
27:     while $f = 0$ do
28:       Apply random perturbation to $\mathbf{m_i}$.
29:       for $k = 1$ to $N_p$ do
30:         if $\mathbf{m_i(k) > \frac{M}{c}}$ then $\mathbf{m_i(k) \leftarrow \frac{M}{c}}$.
31:         end if
32:         if $\mathbf{m_i(k) < 0}$ then $\mathbf{m_i(k) \leftarrow 0}$.
33:         end if
34:       end for
35:       Compute $d_i$ as in (5.15).
36:       if $d_i = 0$ then $f \leftarrow 1$
37:       end if
38:     end while
39:   end for
40: until a stop condition is fulfilled.
```

to 9:25, and a closing auction, from 14:57 to 15:00. The opening and closing auction set the opening and closing prices for the day, and follow different rules than the continuous trading session. This chapter focuses on the continuous trading session. Another important rules to take into account in the Chinese equity market is the up/down limit of 10%. This rules prevents any stock traded in the main board for increasing (decreasing) by more than 10%, compared to the previous day closing price. Please notice that there are different rules for stocks listed in the Chinext and STAR Board, which are outside of the scope of this thesis.

To validate the proposed strategy, nine stocks from the Chinese Stock Market have been chosen, splitting the choices between small, mid or large capitalization stocks. Table 5.1 show the names and tickers of the nine stocks ordered according to their capitalization. The validation consists on the optimization of market orders of $2,000,000$ shares over

Table 5.1: Stock dataset. The following levels of market capitalization were followed for stock classification: small (<80 bn RMB), mid (from 80 to 300 bn) and large (>300 bn).

Name	**Ticker**	**Capitalization**
Nanjing Port	002040	Small
Baotou Dongbao Biotech	300239	Small
Royal Flush Information Network	300033	Small
Advanced micro-fabrication equipment	688012	Mid
Tianqi Lithium	002466	Mid
Shangahi Pudong Development Bank	600000	Mid
Ping An Bank	000001	Large
Midea Group	000333	Large
Kweichow Moutai	600519	Large

a validation set of 20 two hour sessions randomly chosen from the range 1/4/2017 to 12/29/2020 (except the newer stock 688012 which is 1/2/2020 to 12/29/2020). The validation data sets are shown in figures 5.1 to 5.9. It can be seen that the sessions in each set show bearish, bullish and sideways trends. Furthermore, some of the sets, shown in figures 5.2 and 5.3, include sessions in which the 10% limit was activated. Note that in this situation the trading continues, albeit at most at that saturated price and, usually, with small traded volumes.

The proposed trading strategy relies on forecasting of prices and traded volumes, but it is not linked to any particular forecasting technique. Here it is proposed to use the forecasting strategy of section 4.2. This predictor has proved to be more accurate than other methods like modelling the stock as a brownian motion process or forecasting with neural networks. In figure 5.10 it can be seen a sample of the obtained forecasted values, for the stock Shanghai Pudong Development Bank (600000) during a 120 minutes time interval (intraday), using the strategy of section 4.2 as well as neural network and brownian motion process (random walk). The neural network employed had a single hidden layer with 30 neurons. The random walk simulation was obtained following equation 1.2, and using the average return as the expected return (2.8%) and the historical standard deviation (0.1%) as the measure of the volatility of the process. As expected, the Brownian motion was the

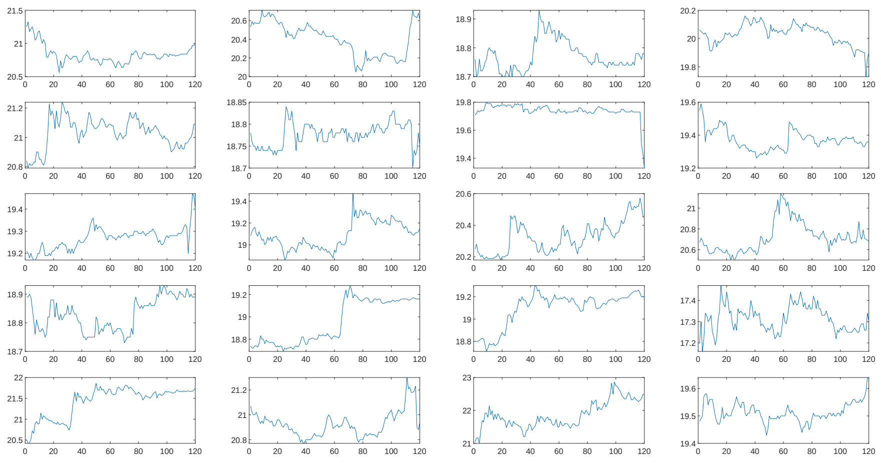


Figure 5.1: Data set for 002040.

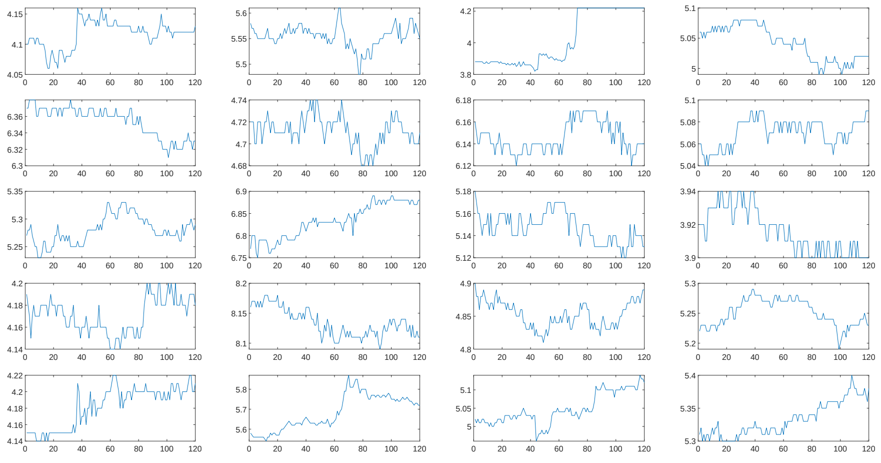


Figure 5.2: Data set for 300239.

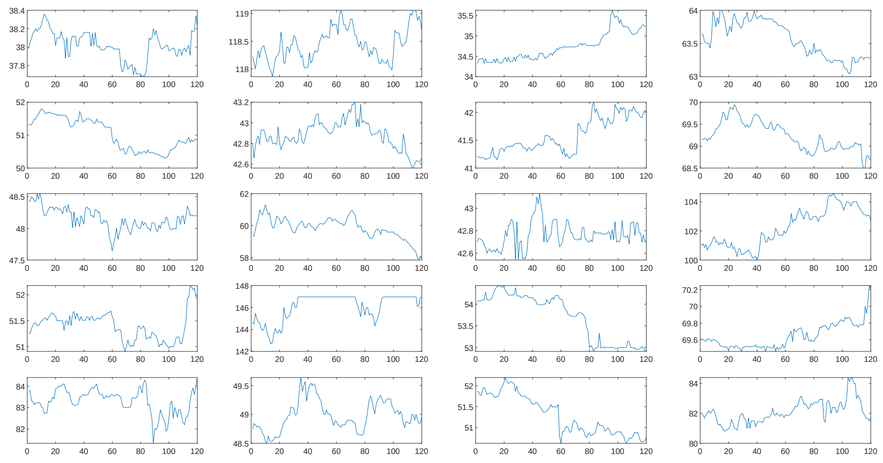


Figure 5.3: Data set for 300033.

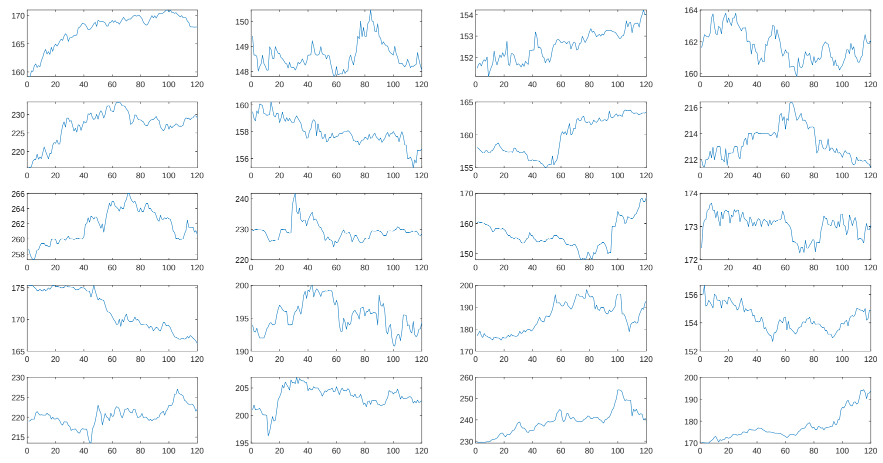


Figure 5.4: Data set for 688012.

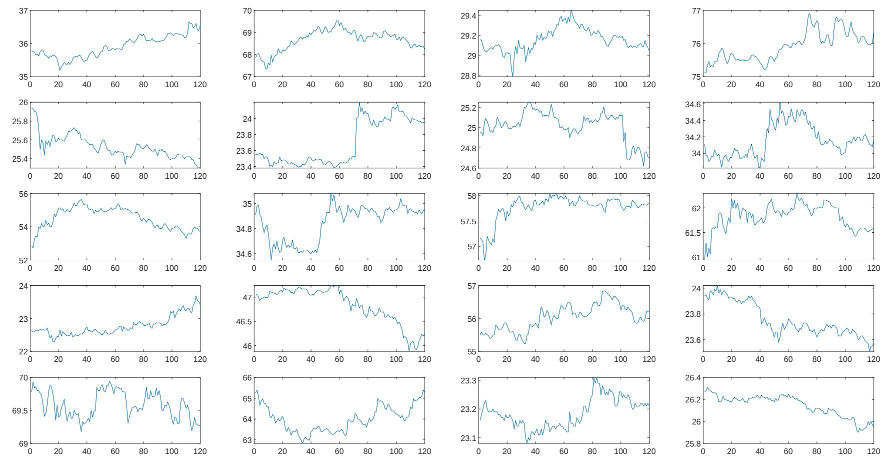


Figure 5.5: Data set for 002466.

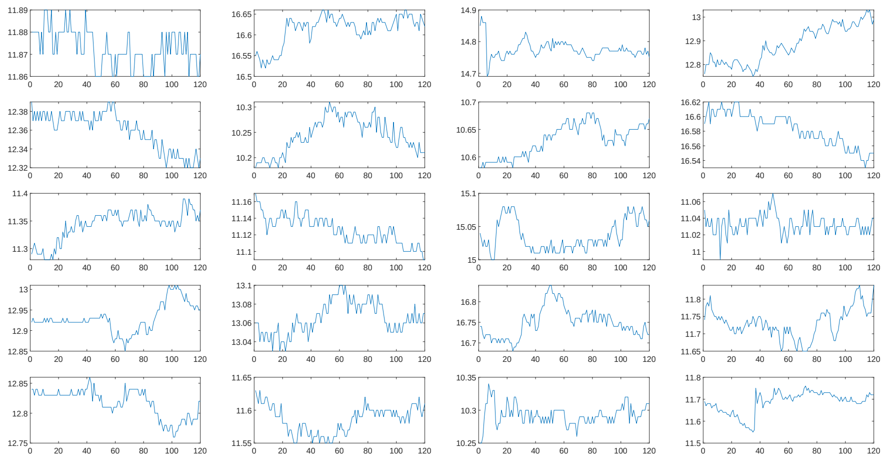


Figure 5.6: Data set for 600000.

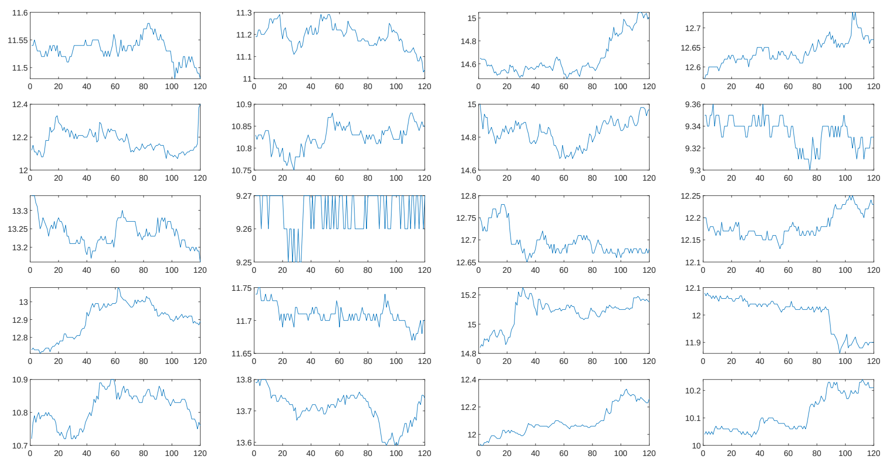


Figure 5.7: Data set for 000001.

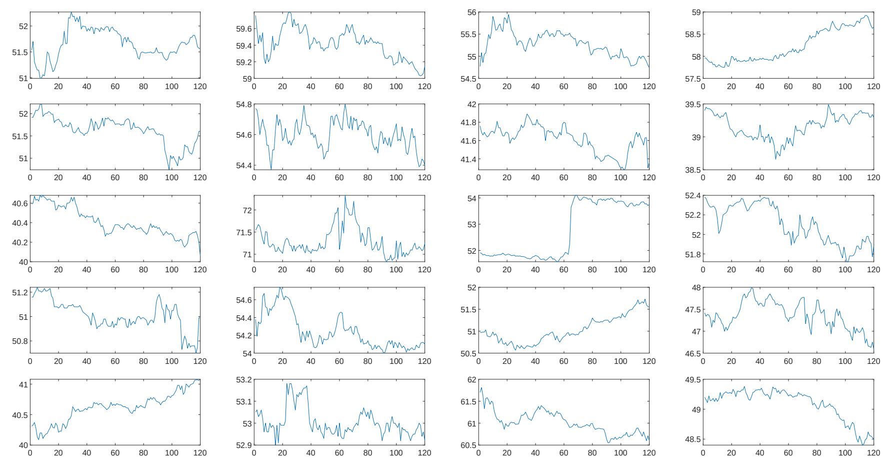


Figure 5.8: Data set for 000333.

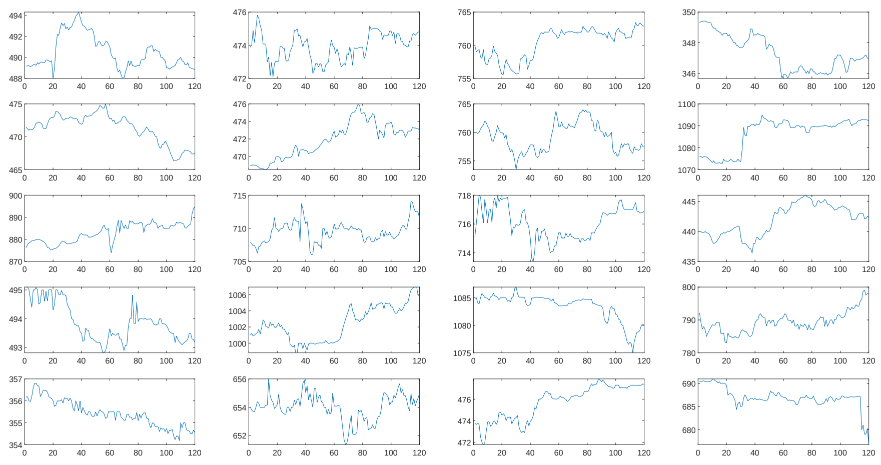


Figure 5.9: Data set for 600519.

less accurate of all the forecasting techniques used. In order to forecast prices using the

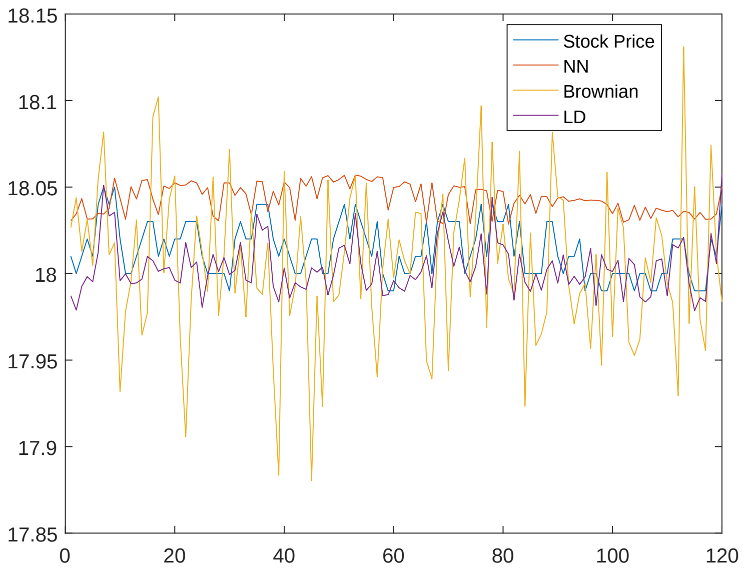


Figure 5.10: Sample of forecasts with the strategy of section 4.2, neural networks and brownian motion.

strategy of section 4.2, the market state, denoted as $z_p(t)$ has been described as composed by the last 100 prices and the last traded volume, i.e.,

$$z_p(t) = [p(t-1), p(t-2), \ldots, p(t-100), v(t-1)].$$

On the other hand, to forecast the traded volume, the market state, denoted by $z_v(t)$ is formed in this case by

$$z_v(t) = [v(t-1), v(t-2), \ldots, v(t-100), p(t-1)].$$

The components of both state vectors are smoothed with an EMA5 (see chapter 4) but the target values (i.e., prices or volumes depending on which is to be forecasted) are unsmoothed. The number of market states in the local data set used in the algorithm 1 of section 4.2 has been chosen to be $N = 250$ and the weighting factor $\gamma = 0$.

The objective is to split the large stock order over each of the sessions of the validation set, optimizing the buying price. The time bucket is 1 minute, thus the time window is 120 time buckets. Thus the proposed strategy must solve (5.6) with an initial prediction horizon of $N_p = 120$, that will shrink down to $N_p = 1$ as the receding horizon optimization is applied through the entire time window. Other parameters of V used in this case study are $\alpha = 0.1$, $\beta = 1.2$, $\mu = 0.5$, $\sigma = 1.07$ and $\iota = 0$. Note that in the Chinese Stock Market there is no difference regarding to fees between splitting the order into many suborders, few suborders or no splitting at all, thus $\iota = 0$ reflects this. On the other hand, the PSO algorithm has been used with 1000 particles of N_p suborders. The stop condition was to limit the algorithm iterations to 50.

To validate the resutls, two well known trading strategies will be considered for comparison purposes. The first one is the VWAP price assuming knowledge of the traded volume (which has to be forecasted in practice). This price is computed as

$$\text{VWAP(t)} = \sum_{k=1}^{N_p} \frac{v(t+k)p(t+k)}{v(t+k)}. \tag{5.16}$$

The second baseline strategy is the TWAP price, which is the average price of the stock over the time window, that is,

$$\text{TWAP}(t) = \sum_{k=1}^{N_p} \frac{p(t+k)}{N_p}. \tag{5.17}$$

Table 5.2 shows the average prices attained on the validation set of each of the 9 stocks proposed in this case study. The average price attained by the proposed algorithm assuming that the real future values of price and traded volume is shown in column “Proposed (hypothetical)”. Obviously, this price cannot be attained in practice, but it is used to illustrate the maximum hypothetical performance, i.e., the minimum price, that could be obtained by using the proposed approach. In practice, forecastings have to be used, and the corresponding prices are shown in column “Proposed (forecasting)”. The last two columns show the average VWAP and TWAP prices. It can be seen that the proposed strategy obtains a lower price than VWAP or TWAP in all the validation sets. Clearly, the prediction error in the price and volume forecasting reduces the difference, but even with forecasted prices and volumes the proposed strategy obtains lower prices than the baseline approaches. Regarding the capitalization of the stocks, it appears that

Table 5.2: Average prices in the validation set (RMB)

Stock	**Proposed (hypothetical)**	**Proposed (forecasting)**	**VWAP**	**TWAP**
002040	19.649	19.809	19.837	19.814
300239	5.216	5.228	5.245	5.242
300033	64.343	64.948	65.013	64.950
688012	183.609	187.181	188.341	187.799
002466	42.609	42.905	42.965	42.962
600000	12.741	12.758	12.771	12.769
000001	12.098	12.127	12.161	12.150
000333	51.505	51.697	51.806	51.774
600519	654.008	657.139	657.608	657.630

the proposed strategy gets worse results in the case of small cap stocks, but this could be related to the differences in the price per share, getting better results in stocks with a higher price per share. On the other hand, it is quite remarkable that for some of the stocks, even the "hypothetical prices" are very close to VWAP or TWAP, e.g., stocks with ticker 600000 and 000001. In this cases, there is clearly a very small margin of improvement, meaning that in these cases it proved very difficult to beat the market on the validation set. There are other stocks, like 688012, in which the hypothetical benefit that could be obtained is greatly wasted by the prediction errors. Regarding this, it is noteworthy that the prediction horizon is quite large in this case study, forcing the forecast up to 120 steps ahead. This is much longer than usual, given the fact that most of the stock forecasting applications focus in one step ahead predictions. On the other hand, higher prediction errors do not always imply a worse result as long as the price trend is correctly predicted. The reason for this is that in the proposed strategy price forecasting accuracy is not what really allow better results, but the ability to forecast the position of the lower prices in the time window. Finally, even in the price difference attained in practice seem small, given the high number of shares traded in this type of orders, the overall benefit can be significant. To illustrate this, consider table 5.3 in which the savings of the proposed strategy over VWAP and TWAP are shown for the case of using the real or forecast prices and volumes. The TWAP and VWAP values are some of the most frequently used benchmarks for execution by institutional investors hence it seems reasonable to compare the results using these benchmarks.

It can be seen that the savings are substantial in most cases and very substantial in some cases like the stocks with tickers 688012 and 600519. Furthermore, considering the savings that can be obtained over the many large orders that are traded over a year, it is evident that even a modest improvement on the price can justify the use of strategies like the proposed in this chapter. It should be noted that a typical institutional investor can trade even hundreds of different stocks, further amplifying the saving effect.

Table 5.3: Savings on the validation set over VWAP and TWAP (RMB).

	Savings over VWAP		**Savings over TWAP**	
Stock	**(hypothetical)**	**(forecasting)**	**(hypothetical)**	**(forecasting)**
002040	7,520,000	1,120,000	6,600,000	200,000
300239	1,160,000	680,000	1,040,000	560,000
300033	26,800,000	2,600,000	24,280,000	80,000
688012	189,280,000	46,400,000	167,600,000	24,720,000
002466	14,240,000	2,400,000	14,120,000	2,280,000
600000	1,200,000	520,000	1,120,000	440,000
000001	2,520,000	1,360,000	2,080,000	920,000
000333	12,040,000	4,360,000	10,760,000	3,080,000
600519	144,000,000	18,760,000	144,880,000	19,640,000

Chapter 6

Conclusions and future work

This final chapter presents the conclusions of this dissertation, in the form of a summary of the main contributions of the thesis in the fields of stock forecasting, as well as trading optimization, and some ideas for future works.

6.1 Summary of contributions

Besides the many differences between narrow and deep stock markets it appears that neural networks are an efficient forecasting tool for both types of markets. Although this appear to be expected as the price trends reduce to a time series in each case, the differences between those markets could be rather large, with the basic expectation being that their behavior and, hence, the appropriate tool for forecasting the dynamics of its stock markets being rather different. This does not appear to be the case with neural networks generating relatively accurate forecasts for narrow, moderately narrow, deep and very deep markets , even after accounting for the issue of stale prices. Just to put it into perspective, the results suggest that the same technique (neural network) of stock forecasting is applicable to stock markets as different as the ones in Namibia, Tanzania and the United States. It is interesting that the forecast for moderately narrow markets are slightly less accurate than those for deep markets. This might be due to differences in quality and reliability of the information and trading platform in those markets. It is also interesting to observe that the results support the idea that the analyzed markets are not perfectly efficient, as forecasting tools such as neural network are able, using only historical data, to generate relatively accurate forecasts, which would seem to contradict the efficient market hypothesis.

The applicability of neural networks as a forecasting tool in other markets, such as for instance the Indian market, was also analyzed. In the case of the Indian market, represented by the Nifty 50 index, several neural network architectures were tested. As expected, simply increasing the number of neurons did not result in better, out-of-sample, forecasts. It was also observed that for a relatively large period of time (2010-2016) there was no statistically significant difference between the accuracy of the forecasts generated using the Levenberg-Marquardt and the Scaled Conjugate learning algorithms.

Another contribution of this dissertation is developing an algorithm for the selection of technical indicators. There is a large amount of technical indicators developed for stock market forecasting purposes. In some cases these indicators give contradicting signals. It is also important to take into account that it is unfeasible to test all the potential combinations of technical indicators, given the extremely large number of potential combinations. Thus, it is necessary to use some form of search to find the best combination, without having to test all of them. The proposed combinatorial method for variable selection is applicable to the problem of forecasting the direction of stock market movements using non-linear techniques such as neural networks. This approach aims to find combinations of technical indicators that generates better results than directly using all the available variables. Another relevant result is that better indicator choices have been researched for 8 different indexes from the Chinese, US and European stock market. While the approach was tested using neural networks it can be easily applied to other forecasting techniques. It can also be generalized to other forecasting problems besides stocks. The calculation time for the combinatorial approach is another factor to take into account as it is a computationally demanding, but clearly more efficient than estimating the forecasts for all the possible combinations.

Another contribution of this dissertation is the application to the task of stock forecasting of a technique based on local data. Additionally, it has been applied another technique that generates not only a stock forecast but also a probability distribution related to this stock forecast. Having a probability distribution related to the stock forecast can be a very useful tool for risk management purposes.

The first technique is related to direct weight optimization techniques and obtains the forecast by using local data close to the current market state. The computational burden is quite low and does not require a training phase, except for the tuning of γ. Moreover, its results when applied to a well known case study have been validated in relation to two well-known techniques, such as neural networks and persistence predictors. The second approach computes the price intervals using a probabilistic approach, in which the empirical conditional probability density function of the forecasted price is computed using local data. The algorithm for doing this is easily parallelizable, making its computational burden manageable. This approach has been also validated and compared favourably to the well-known quantile regression approach. Both techniques have been proved to be useful for the investors in terms of accuracy, and have other advantages like adaptation to the current market situation. Thus, the proposed techniques have proved that they can be added to the toolbox of stock market traders.

In the last chapter a trading strategy was proposed for large orders that have to be splitted to minimize the impact on the market. The strategy is based on the well known receding horizon optimization scheme used in predictive controllers. The strategy can be adapted to a variety of market and limit orders and, although relies on forecasting the future prices and traded volumes, is independent of the forecasting method used. The strategy has been validated with an assorted set of Chinese stocks of different capitalization levels. The results have been very positive despite the fact that the prediction horizon is quite large. More precisely, the savings over two well known price benchmarks (TWAP and VWAP) are quite significant.

6.2 Future work

There are several areas of potential future work. In the field of stock forecasting, it would be interesting to further analyze how the type of forecasting tool changes as the forecasting time interval varies. The basic idea is that potentially a stock forecasting tool that performs well with daily data might have a different performance when applied to high frequency (intraday data) or when applied to a longer time horizon like weeks or even months.

Another interesting avenue of future work would be to analyze the impact on the presented forecasting tools of black swan events. Black swan event refer to unpredictable but important events that move the stock market significantly. Trading forecasting tools based on historical data can struggle when faced by an environment that has not occurred in the time series used to train the algorithm.

The algorithm of chapter 3 could be improved by introducing some concepts borrowed from particle swarm optimization like the updating speed used in chapter 5 or genetic operators. However, there is no guarantee that these additions could have a relevant impact in the performance of the algorithm. Keeping the random nature of the search method, it could be interesting to use resampling strategies like that of [Kitagawa, 1996], or updating rules like that of PSO algorithms.

Regarding the strategies applied in chapter 4, among the open questions that could be considered as future work, one would be the study of which technical indicators work best as market state, i.e., a feature selection study tailored for the proposed approaches. This could be done using the algorithm of chapter 3, but it would be difficult because the computational burden of the hyperparameter training phase of algorithm 3. In this regard, the number of possible technical indicators is quite high, and clearly, some of them are redundant, but from the results of 3, it can be inferred that among the most promising technical indicators would be moving averages, either simple or exponential, the daily return on capital, the traded volume and the proprietary JKHL index. Nevertheless, the suitability of these technical indicators with the proposed techniques is something that needs to be studied in future works.

The proposed strategy in chapter 5 will certainly benefit from future research on how to reduce the impact of prediction errors. In this regard, most forecasting techniques are focused on one-step price prediction, whereas the needs of the proposed strategy are quite different. The topic of research would be to develop methods to forecast the position of the c lower prices on a given time window, rather than predicting the prices themselves. Also, it would be interesting to modify the strategy to be able to use prediction horizons shorter than the time window without affecting the performance of the strategy.

Finally, it would be also interesting seeing how the trading optimization techniques presented in this dissertation work in real practice. In order to do this, it would likely required to have a collaboration with a financial institution with the infrastructure required to carry out actual trading.

Appendix A

Additional figures of chapter 2

A.1 Figures with results of testing different training algorithms and number of neurons.

This section contains the figures for the results of using different training algorithms and number of neurons when using neural networks to forecast the stock indexes proposed in chapter 2. Each graph contains the evolution of the forecasting accuracy as the number of neurons is increased for a given index. For instance, figure A.1 shows how the forecasting accuracy, measured as R-squared, tends to decrease as the number of neurons increases for all the equity index (see table 2.1) for the quasi Newton learning algorithm (BFG). The abbreviations for each training algorithm can be seen in table 2.2.

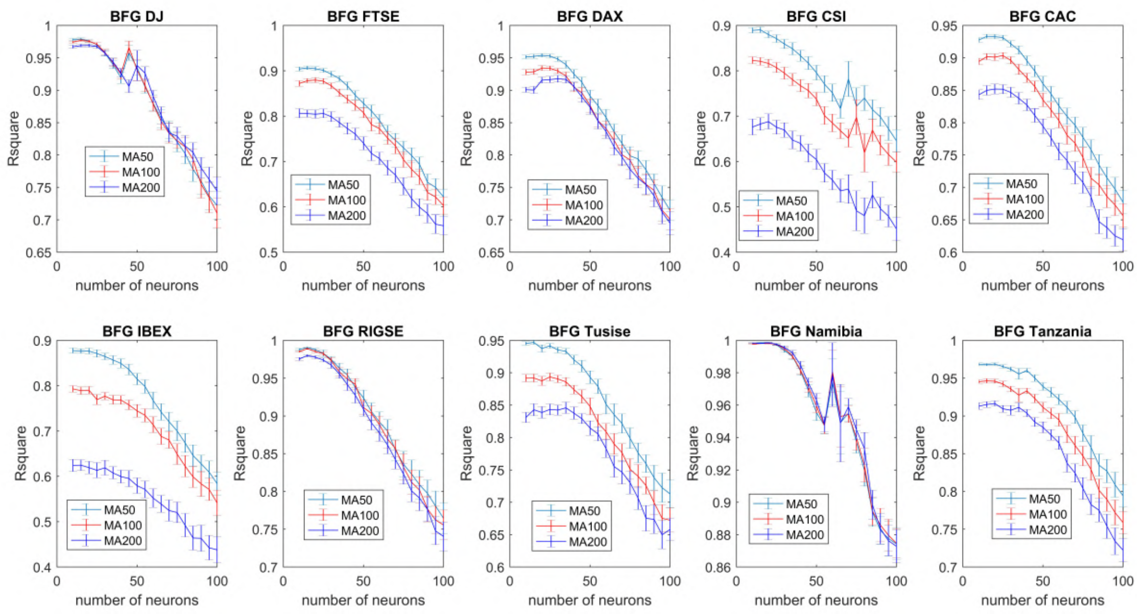


Figure A.1: Forecasting accuracy comparison of different moving averages using the quasi Newton (BFG) training algorithm (99% confidence interval).

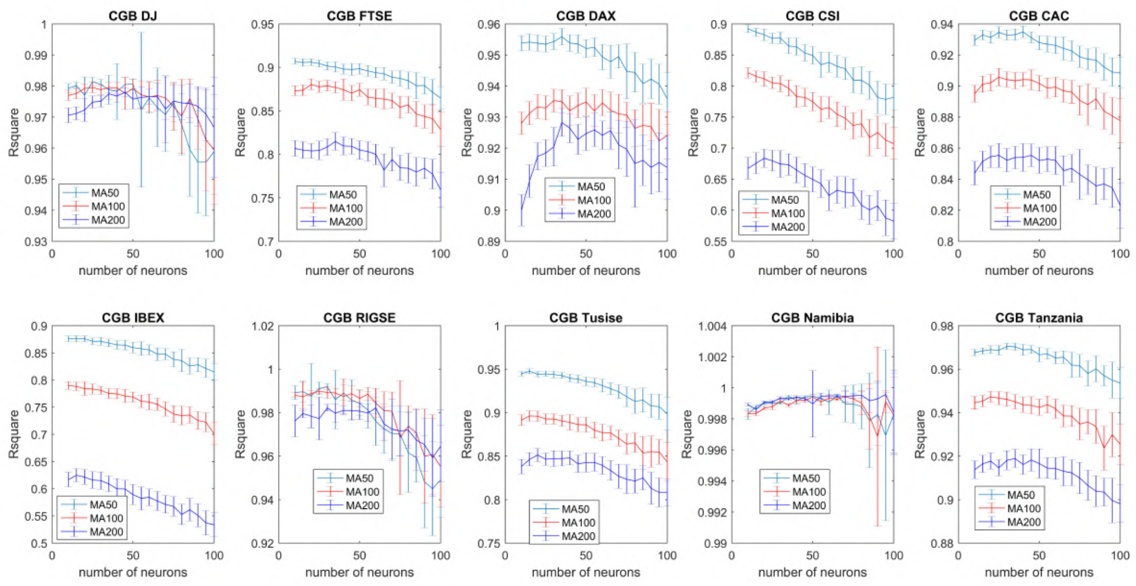


Figure A.2: Forecasting accuracy comparison of different moving averages using the conjugate gradient (with restarts) training algorithm (99% confidence interval).

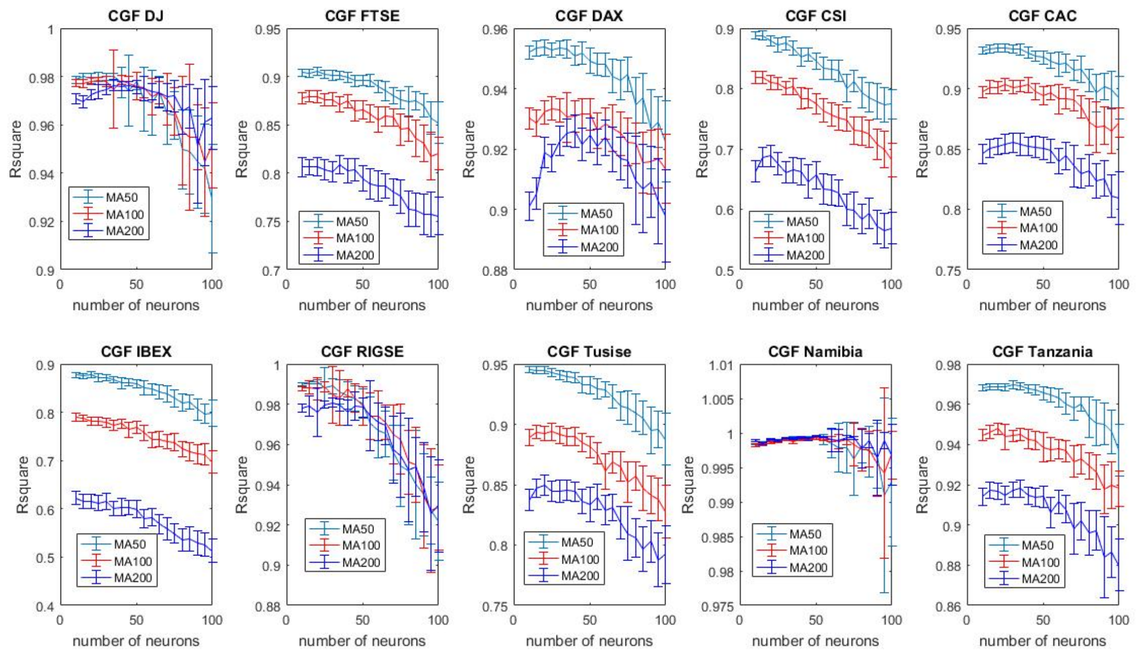


Figure A.3: Forecasting accuracy comparison of different moving averages using the conjugate gradient Fetcher Powell training algorithm (99% confidence interval).

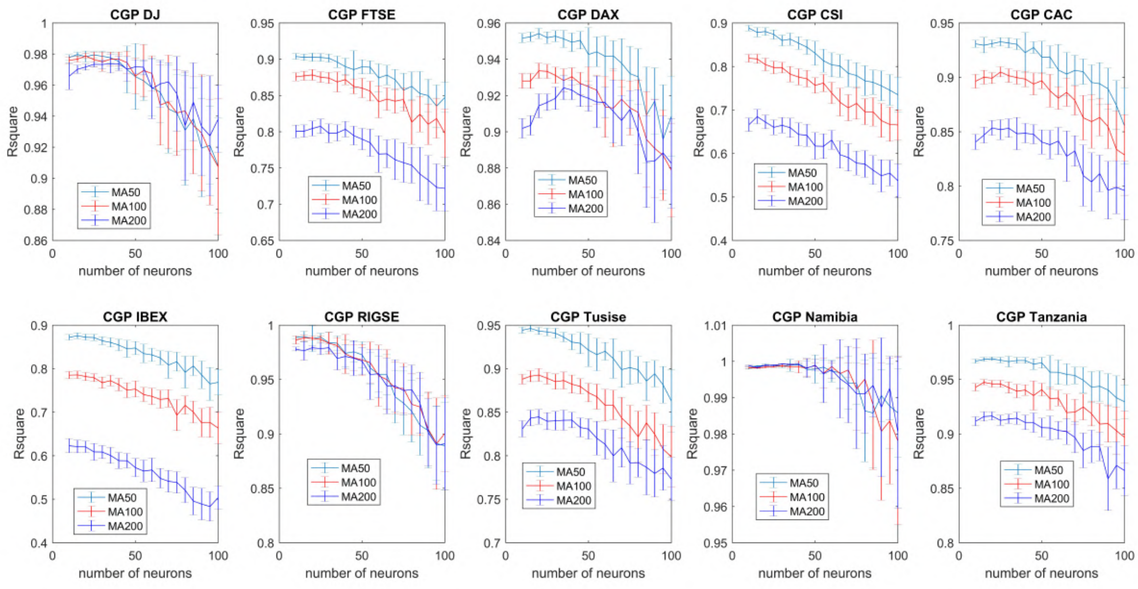


Figure A.4: Forecasting accuracy comparison of different moving averages using the conjugate gradient Polak Ribiere training algorithm (99% confidence interval).

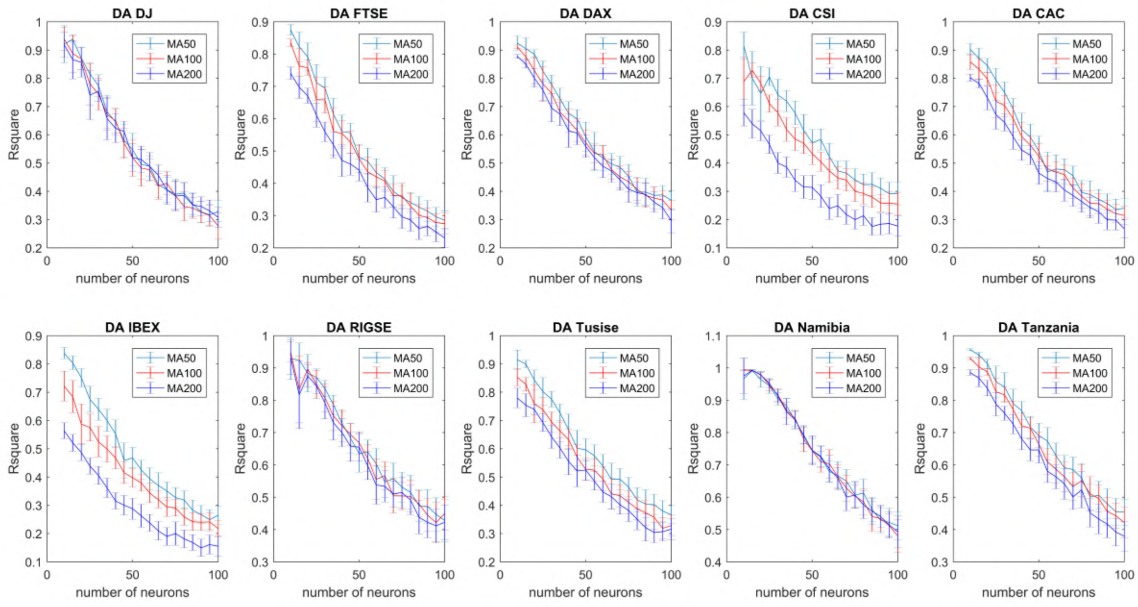


Figure A.5: Forecasting accuracy comparison of different moving averages using the gradient descent (adaptive learning) training algorithm (99% confidence interval).

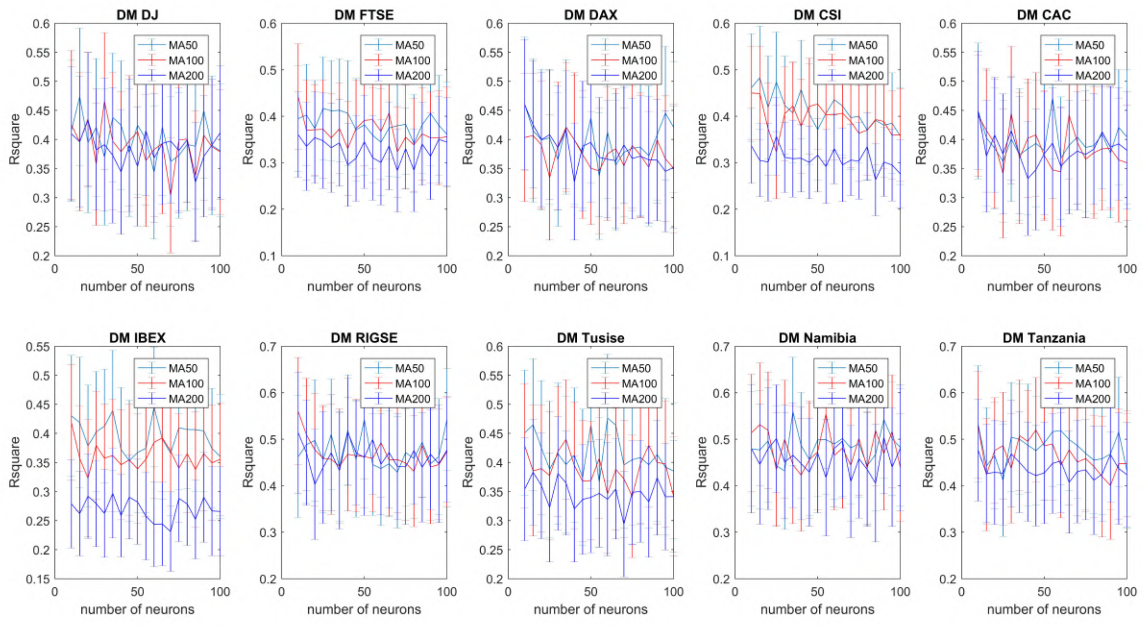


Figure A.6: Forecasting accuracy comparison of different moving averages using the gradient descent (momentum) training algorithm (99% confidence interval).

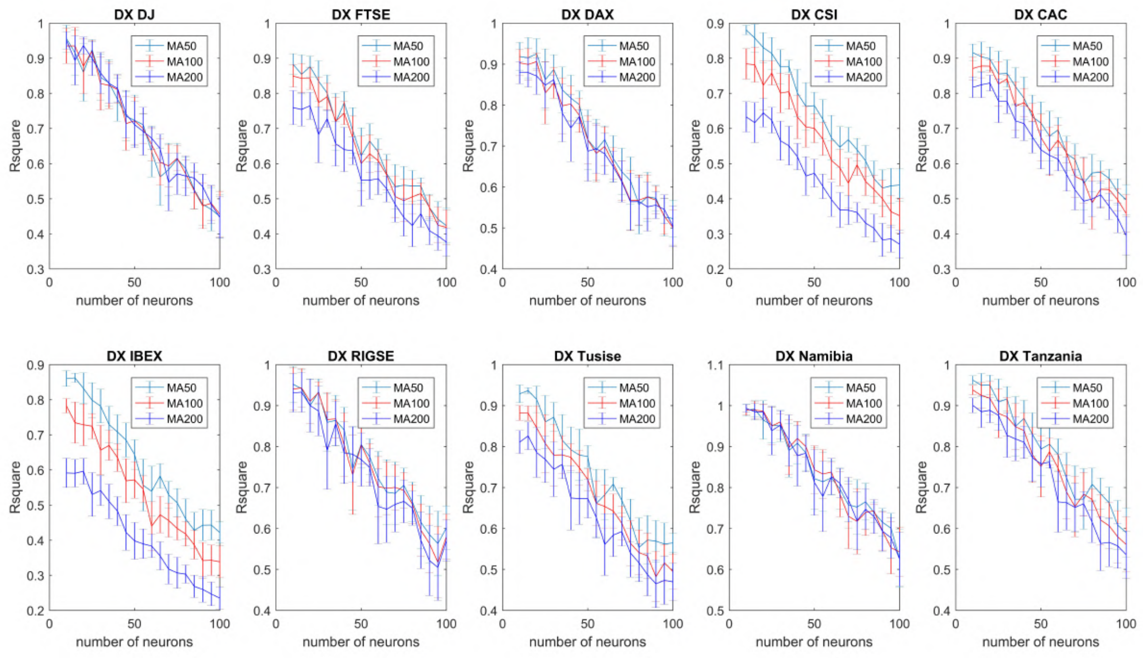


Figure A.7: Forecasting accuracy comparison of different moving averages using the gradient descent (momentum) training algorithm (99% confidence interval).

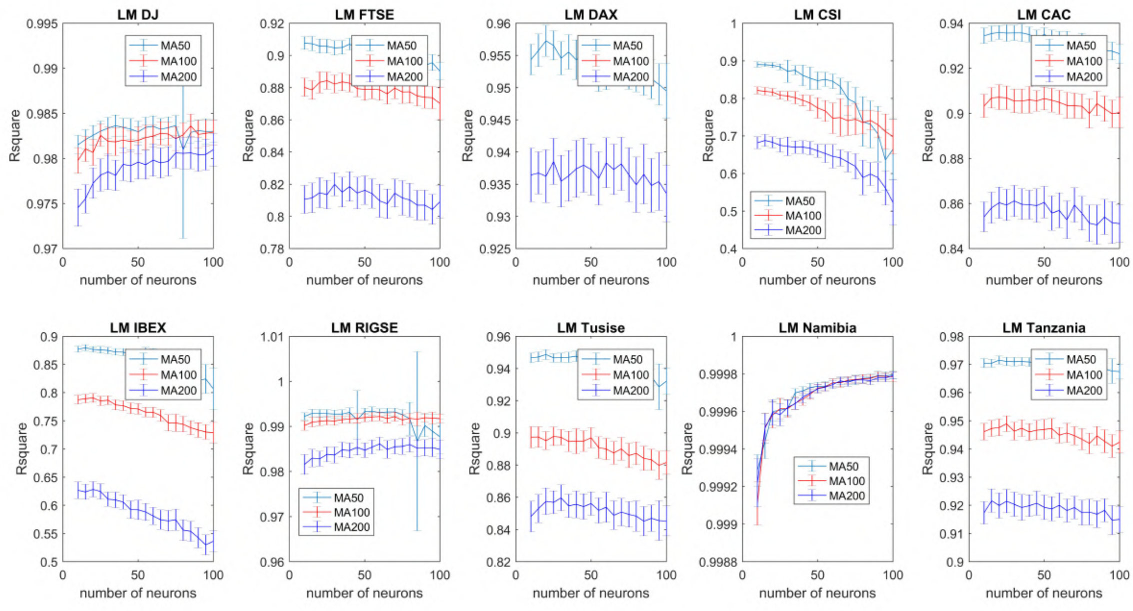


Figure A.8: Forecasting accuracy comparison of different moving averages using the Levenberg Marquardt training algorithm (99% confidence interval).

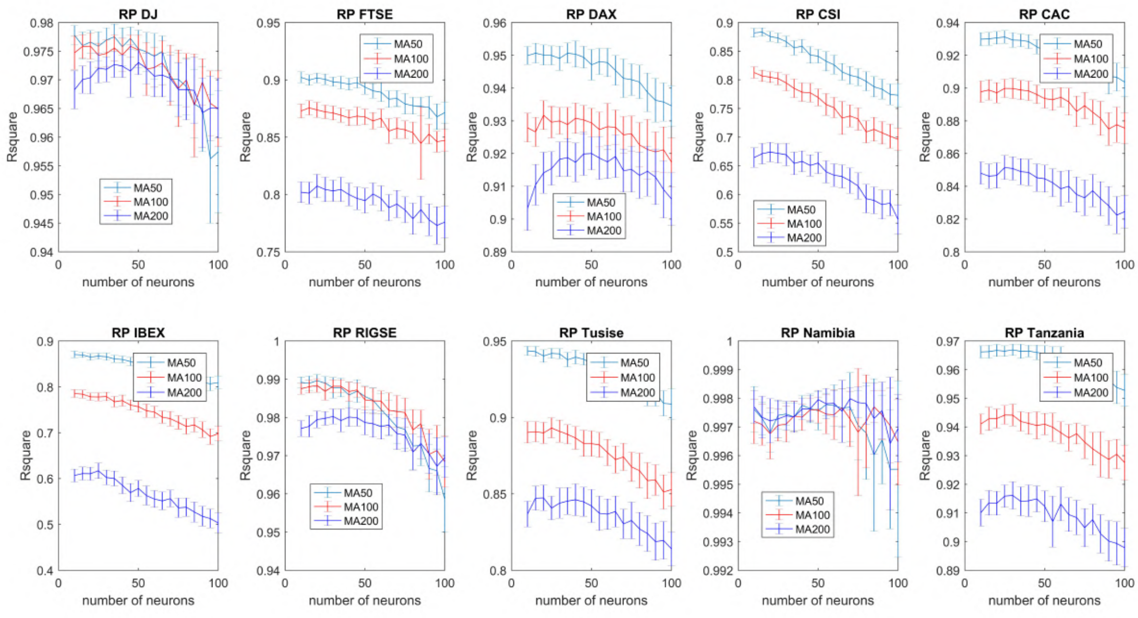


Figure A.9: Forecasting accuracy comparison of different moving averages using the resilient backpropagation training algorithm (99% confidence interval).

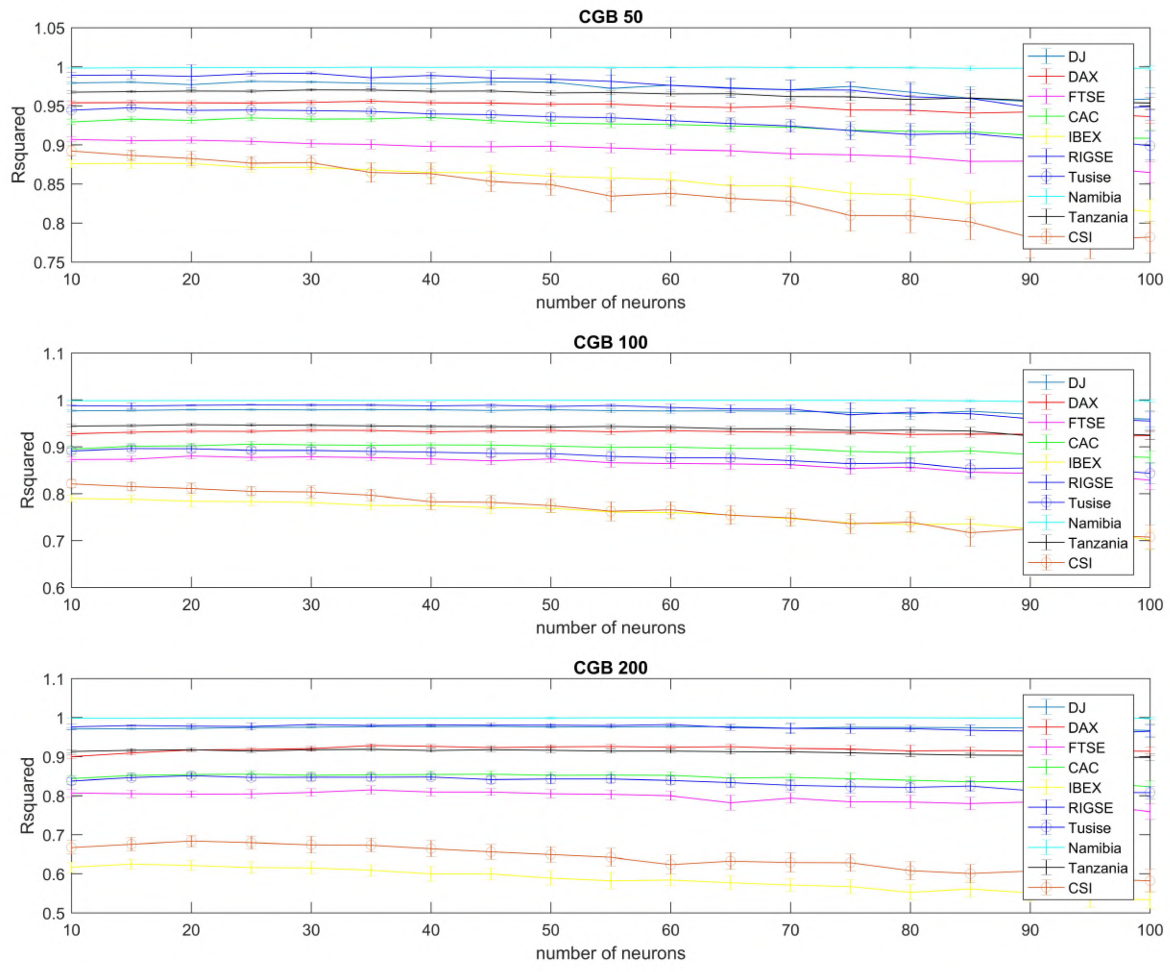


Figure A.10: Results using conjugate gradient (with restarts) training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

A.2 Figures of results using different training algorithms and different moving averages

In this section every figure is divided into three subplots. From top to bottom this graph represents the analysis for the 50, 100 and 200 days moving averages showing how the forecasting accuracy evolves as the number of neurons is increased. For instance, figure A.10 illustrates the evolution of the forecasting accuracy of the conjugate gradient (with restarts) learning algorithm for all the ten stock indexes considered.

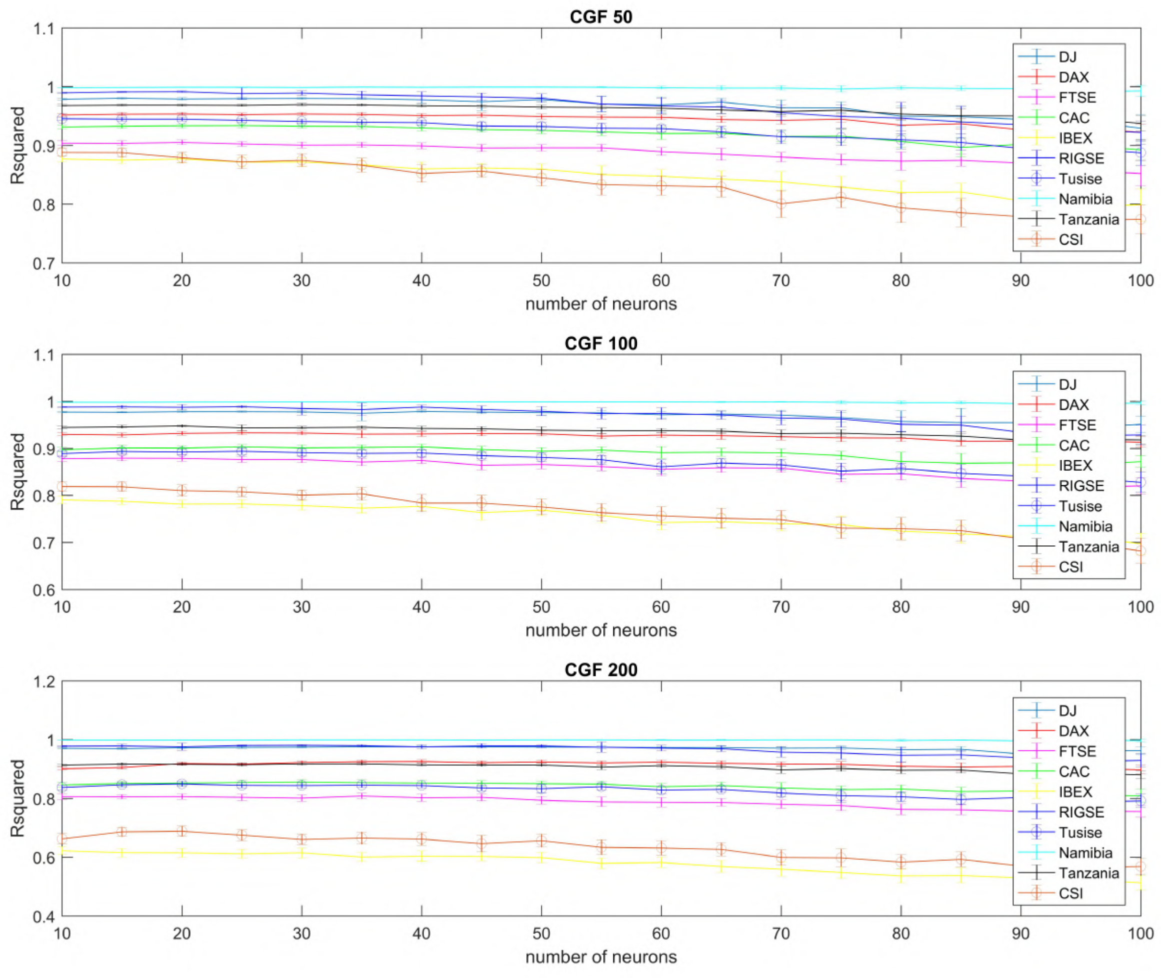


Figure A.11: Results using conjugate gradient Fetcher Powell training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

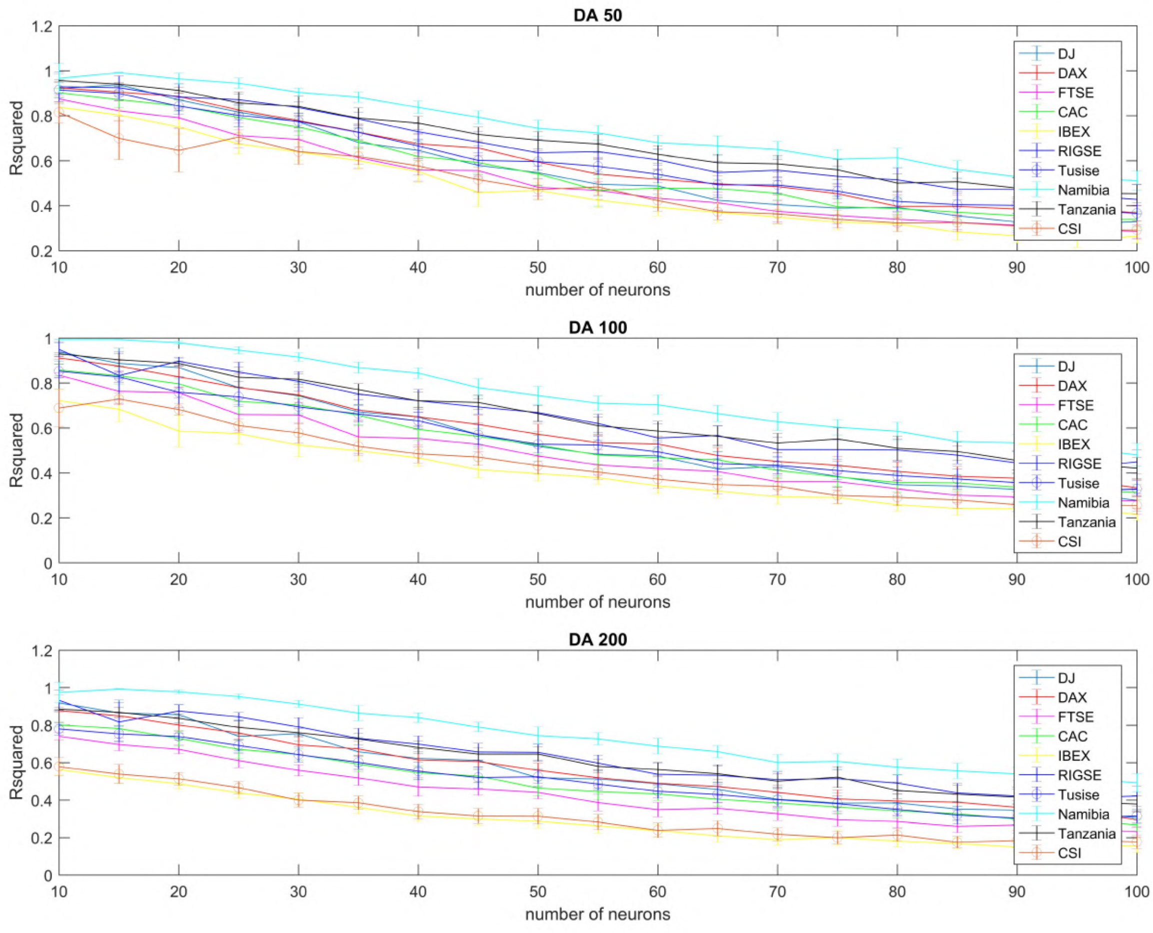


Figure A.12: Results using gradient descent (adaptive learning) training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

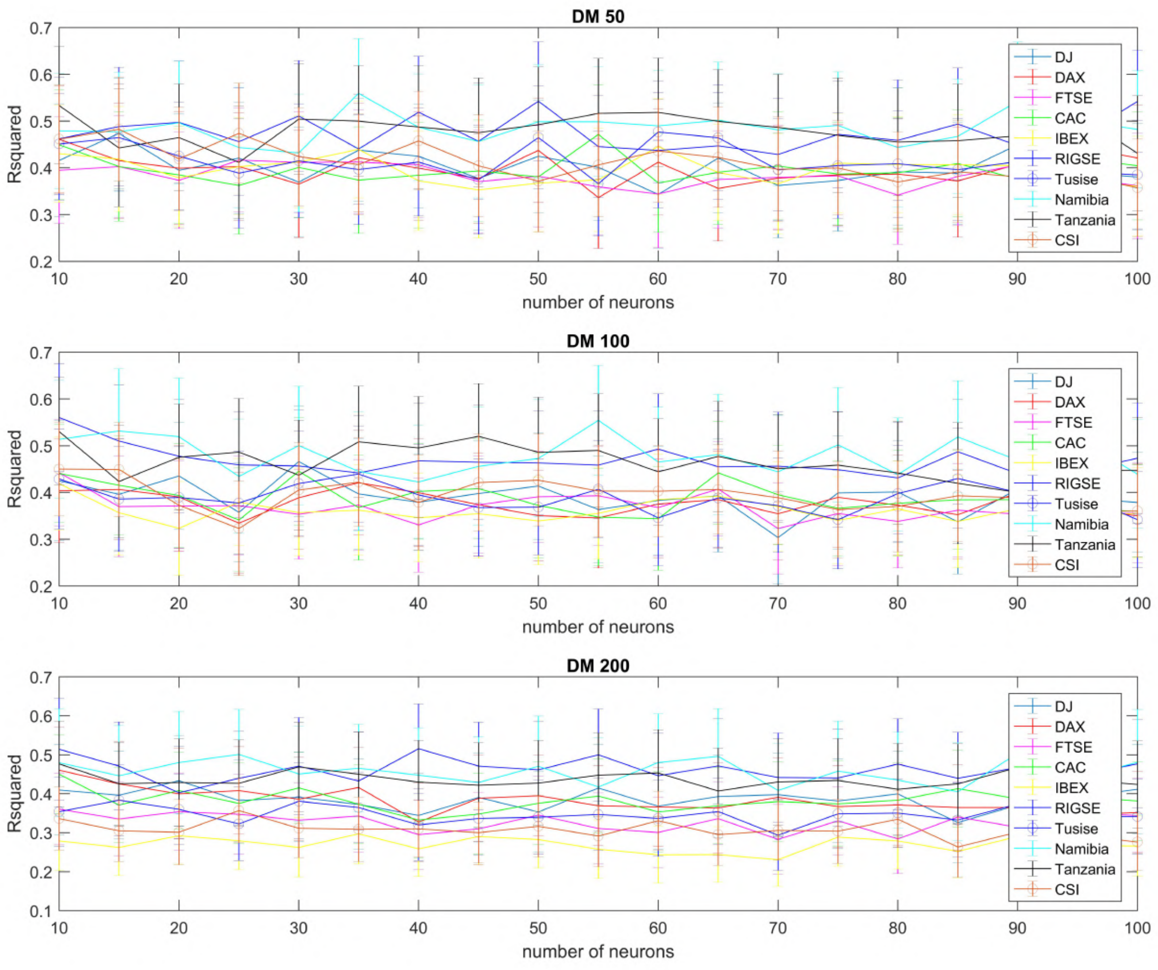


Figure A.13: Results using gradient descent (momentum) training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

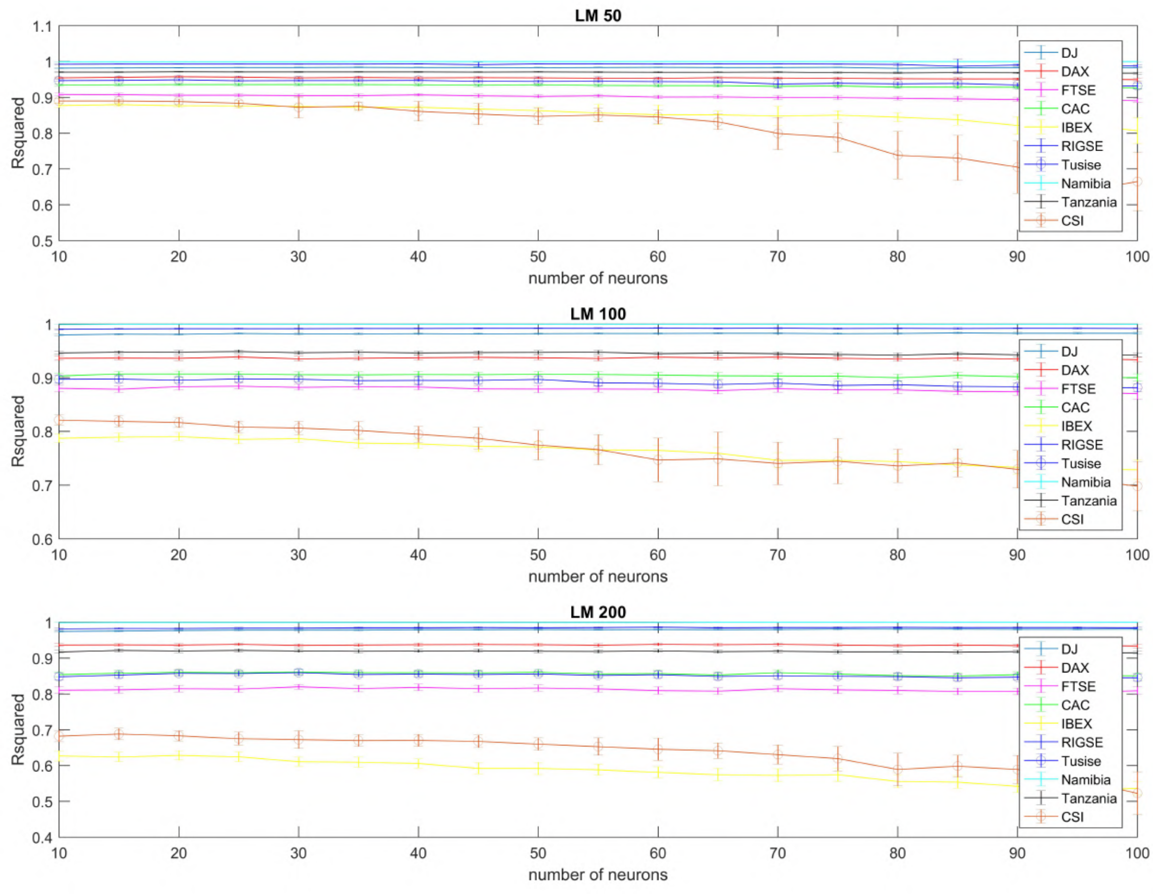


Figure A.14: Results using Levenberg Marquardt training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

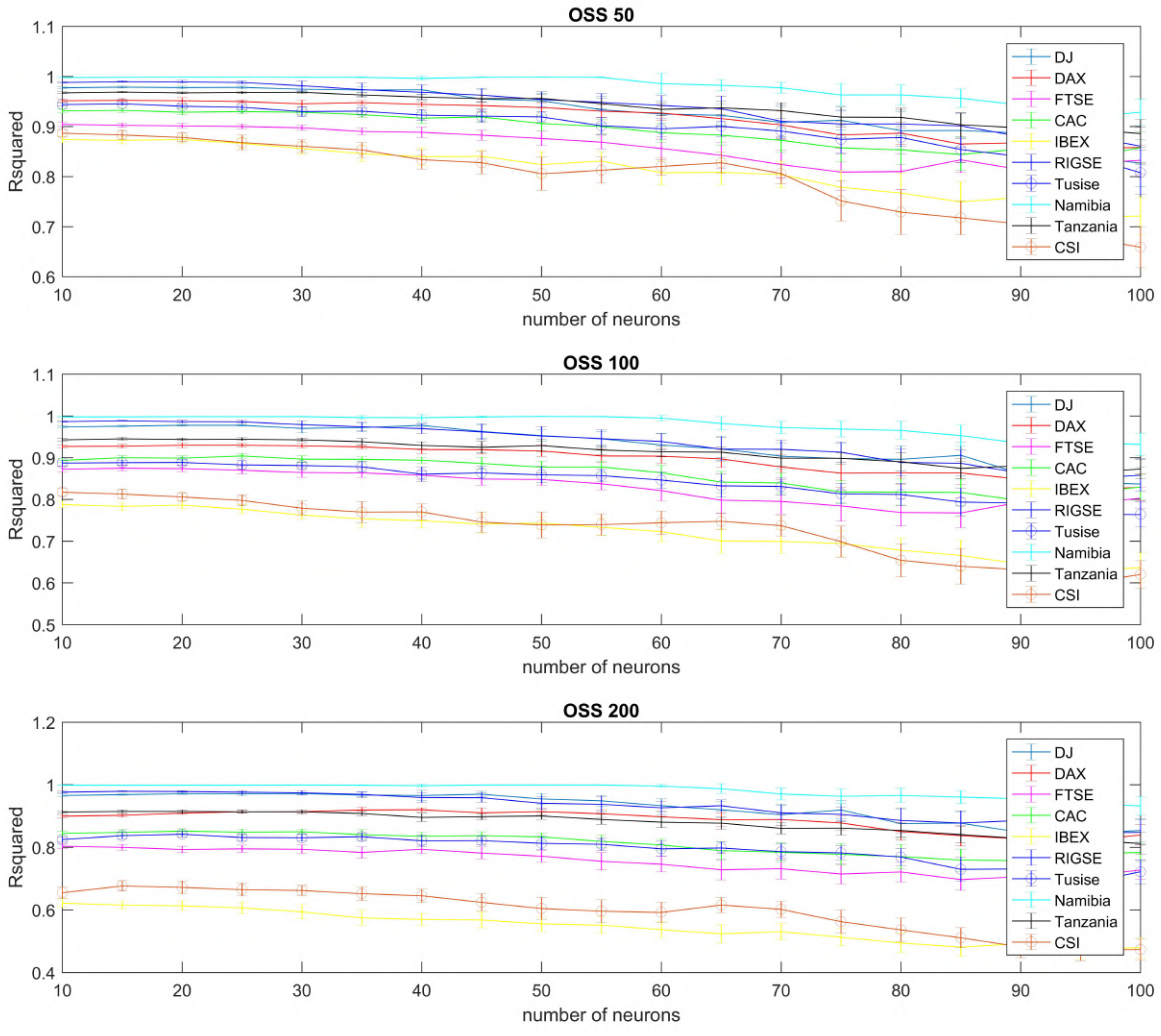


Figure A.15: Results using Secant training per country, 99% confidence interval, using the 50, 100 and 200 days moving average .

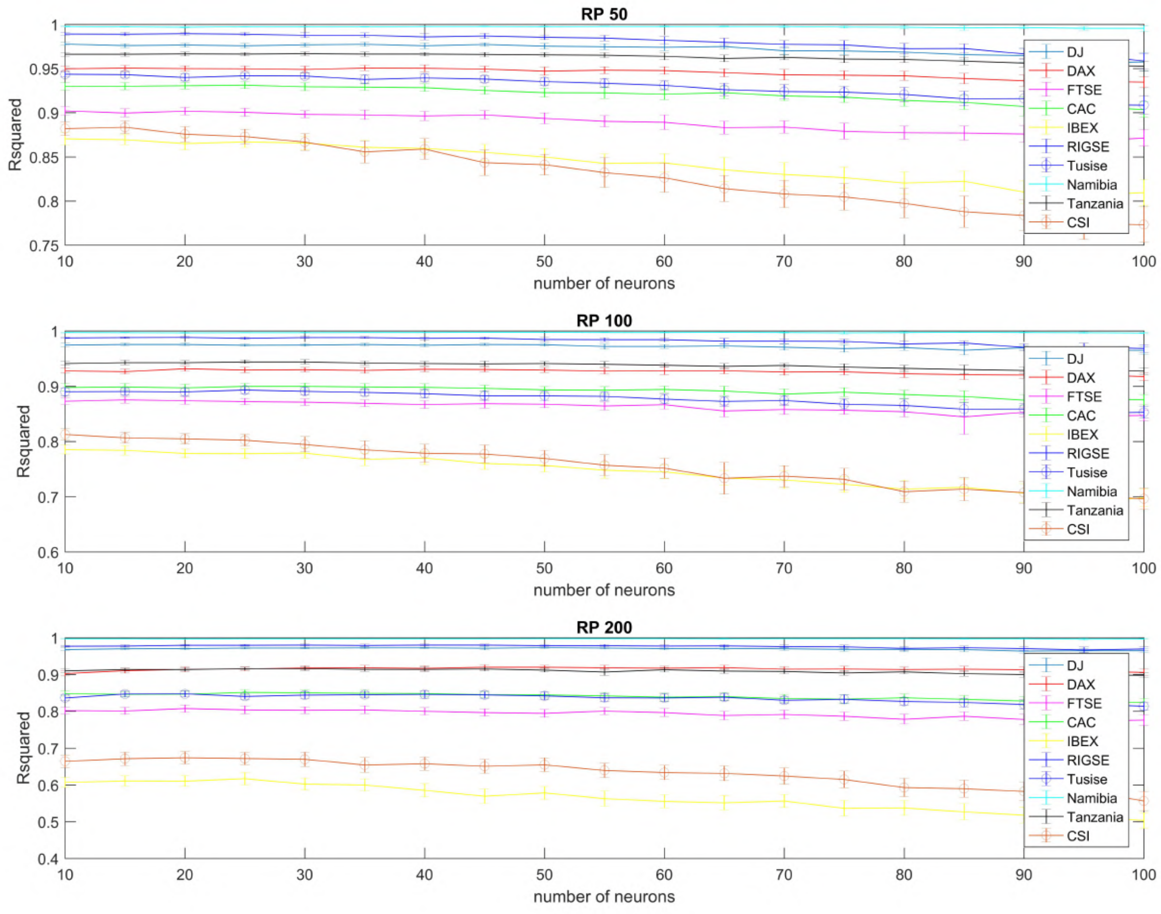


Figure A.16: Results using resilient backpropagation training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

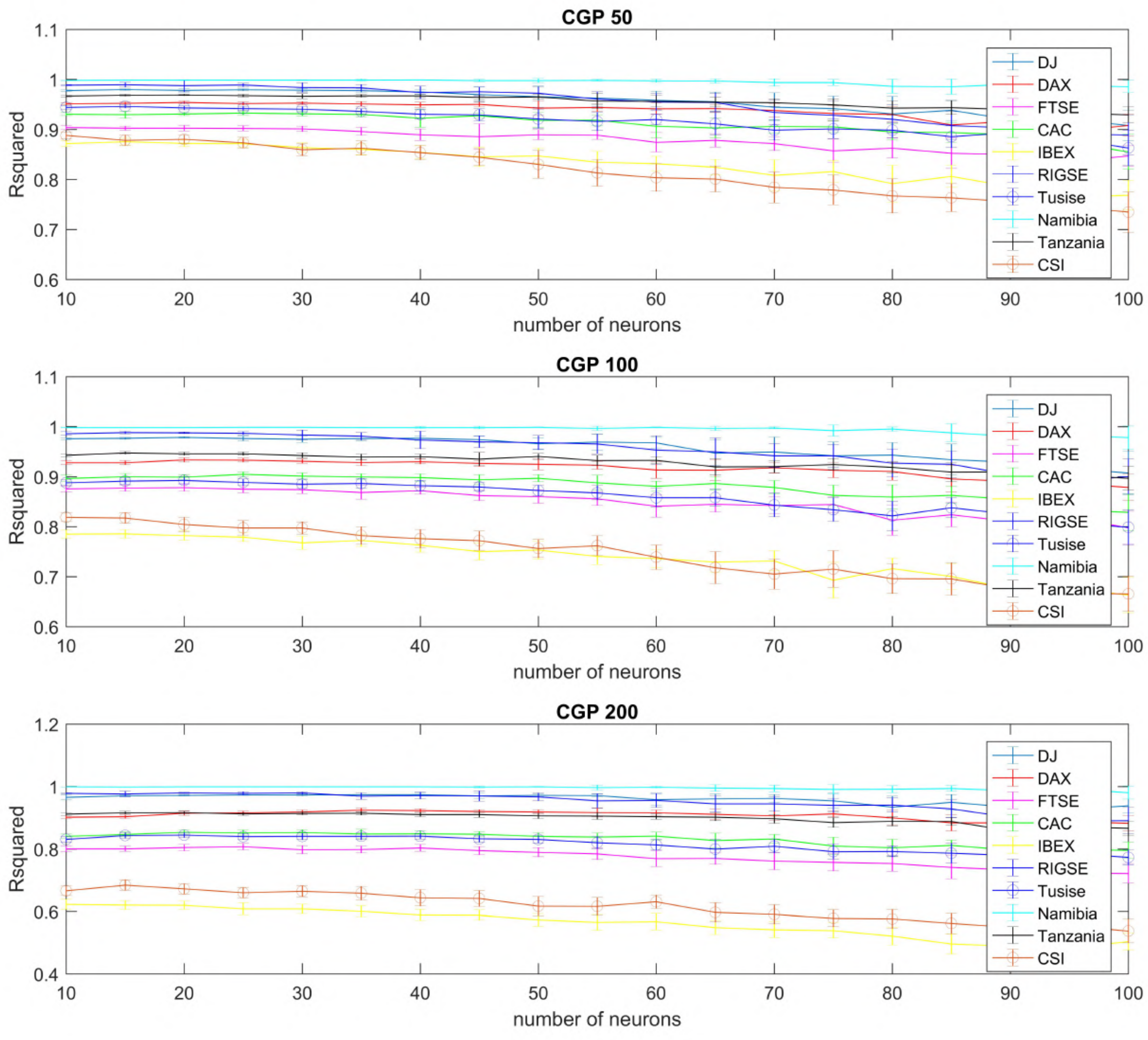


Figure A.17: Results using Polak Ribiere training per country, 99% confidence interval, using the 50, 100 and 200 days moving average.

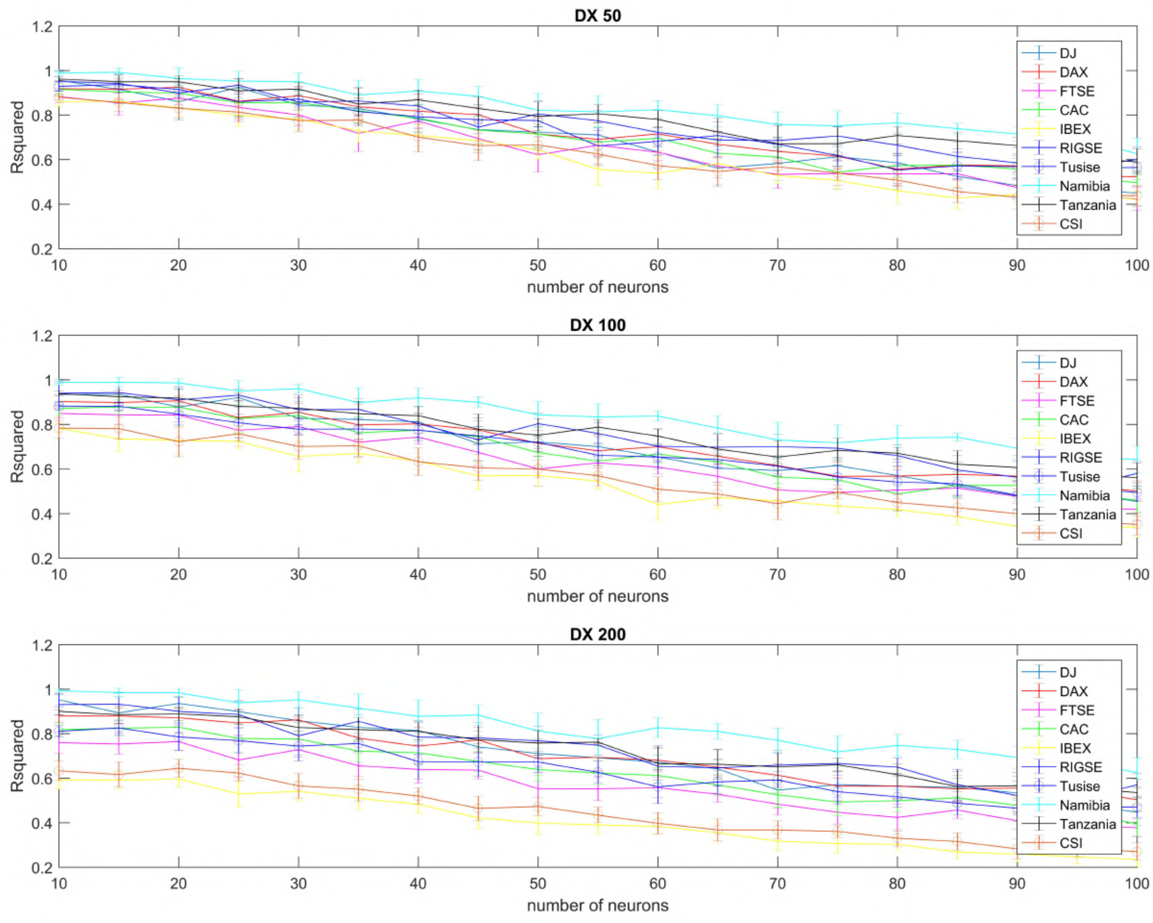


Figure A.18: Results using gradient descent with momentum and adaptive learning training per country, 99% confidence interval, using the 50, 100 and 200 days moving average..

List of Figures

List of Tables

Bibliography

[Adrian, 2015] Adrian, Z.-I. (2015). The sensitivity of moving average trading rules performance with respect to methodological assumptions. *Procedia Economics and Finance*, 32:1353–1361. (Cited on page 23).

[Ahn et al., 2018] Ahn, K., Choi, M., Dai, B., Sohn, S., and Yang, B. (2018). Modeling stock return distributions with a quantum harmonic oscillator. *EPL (Europhysics Letters)*, 120(3):38003. (Cited on page 48).

[Al Baali, 1985] Al Baali, M. (1985). Descent property and global convergence of the fletcher reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5(1):121–124. (Cited on page 16).

[Almgren, 2003] Almgren, R. F. (2003). Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied mathematical finance*, 10(1):1–18. (Cited on page 84).

[Anderberg, 1973] Anderberg, M. R. (1973). *Cluster analysis for applications*. Academic press. (Cited on page 65).

[Atsalakis and Valavanis, 2009] Atsalakis, G. S. and Valavanis, K. P. (2009). Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert systems with Applications*, 36(7):10696–10707. (Cited on page 3).

[Baba and Kozaki, 1992] Baba, N. and Kozaki, M. (1992). An intelligent forecasting system of stock price using neural networks. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 371–377. IEEE. (Cited on page 61).

[Bai and Liu, 2007] Bai, E.-W. and Liu, Y. (2007). Recursive direct weight optimization in nonlinear system identification: A minimal probability approach. *IEEE Transactions on Automatic Control*, 52(7):1218–1231. (Cited on page 62).

[Baldi and Hornik, 1989] Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58. (Cited on page 38).

[Ballings et al., 2015] Ballings, M., Van den Poel, D., Hespeels, N., and Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056. (Cited on page 14).

[Barnes, 1986] Barnes, P. (1986). Thin trading and stock market efficiency: The case of the kuala lumpur stock exchange. *Journal of Business Finance & Accounting*, 13(4):609–617. (Cited on page 19).

[Barnett and Serletis, 2000] Barnett, W. A. and Serletis, A. (2000). Martingales, nonlinearity, and chaos. *Journal of Economic Dynamics and Control*, 24(5-7):703–724. (Cited on page 61).

[Bask, 2020] Bask, M. (2020). Pure announcement and time effects in the dividend-discount model. *The Quarterly Review of Economics and Finance*, 77:266–270. (Cited on page 2).

[Battiti, 1992] Battiti, R. (1992). First-and second-order methods for learning: between steepest descent and newton's method. *Neural computation*, 4(2):141–166. (Cited on page 18).

[Becker et al., 2020] Becker, M., Lippel, J., Stuhlsatz, A., and Zielke, T. (2020). Robust dimensionality reduction for data visualization with deep neural networks. *Graphical Models*, 108:101060. (Cited on page 39).

[Berkowitz, 2001] Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics*, 19(4):465–474. (Cited on page 62).

[Bertsimas and Lo, 1998] Bertsimas, D. and Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*, 1(1):1–50. (Cited on page 82).

[Bettman et al., 2009] Bettman, J. L., Sault, S. J., and Schultz, E. L. (2009). Fundamental and technical analysis: substitutes or complements? *Accounting & Finance*, 49(1):21–36. (Cited on page 37).

[Białkowski et al., 2008] Białkowski, J., Darolles, S., and Le Fol, G. (2008). Improving vwap strategies: A dynamic volume approach. *Journal of Banking & Finance*, 32(9):1709–1722. (Cited on page 82).

[Bianchini et al., 1994] Bianchini, M., Gori, M., and Maggini, M. (1994). On the problem of local minima in recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(2):167–177. (Cited on page 38).

[Bofa, 2007] Bofa (2007). Marching up the learning curve: The second buy-side algorithmic trading survey. Bank of America. (Cited on page 81).

[Bokhari et al., 2005] Bokhari, J., Cai, C., Hudson, R., and Keasey, K. (2005). The predictive ability and profitability of technical trading rules: does company size matter? *Economics letters*, 86(1):21–27. (Cited on page 38).

[Bravo et al., 2016] Bravo, J. M., Alamo, T., Vasallo, M., and Gegundez, M. (2016). A general framework for predictors based on bounding techniques and local approximation. *IEEE Transactions on Automatic Control*, 62(7):3430–3435. (Cited on page 62).

[Breger et al., 2017] Breger, A., Ehler, M., Bogunovic, H., Waldstein, S., Philip, A.-M., Schmidt-Erfurth, U., and Gerendas, B. (2017). Supervised learning and dimension reduction techniques for quantification of retinal fluid in optical coherence tomography images. *Eye*, 31(8):1212–1220. (Cited on page 39).

[Brown and Jennings, 1989] Brown, D. P. and Jennings, R. H. (1989). On technical analysis. *The Review of Financial Studies*, 2(4):527–551. (Cited on page 38).

[Bruand and Gibson-Asner, 1998] Bruand, M. and Gibson-Asner, R. (1998). The effects of newly listed derivatives in a thin stock market. *Review of Derivatives Research*, 2(1):59–86. (Cited on page 18).

[Bruni, 2017] Bruni, R. (2017). Stock market index data and indicators for day trading as a binary classification problem. *Data in brief*, 10:569–575. (Cited on page 38).

[Budagaga, 2020] Budagaga, A. R. (2020). Dividend policy and market value of banks in mena emerging markets: residual income approach. *Journal of Capital Markets Studies*. (Cited on page 2).

[Cao and Wang, 2019] Cao, J. and Wang, J. (2019). Stock price forecasting model based on modified convolution neural network and financial time series analysis. *International Journal of Communication Systems*, 32:e3987. (Cited on page 61).

[Cao et al., 2018] Cao, W., Wang, X., Ming, Z., and Gao, J. (2018). A review on neural networks with random weights. *Neurocomputing*, 275:278–287. (Cited on page 14).

[Carnerero et al., 2020] Carnerero, A. D., Ramirez, D. R., and Alamo, T. (2020). Probabilistic interval predictor based on dissimilarity functions. *arXiv e-prints*, page arXiv:2010.15530. (Cited on pages 62, 65, 66, 67, 68, and 70).

[Chaudhuri and Wu, 2003] Chaudhuri, K. and Wu, Y. (2003). Random walk versus breaking trend in stock prices: Evidence from emerging markets. *Journal of Banking & Finance*, 27(4):575–592. (Cited on page 8).

[Chen et al., 2016] Chen, J.-F., Chen, W.-L., Huang, C.-P., Huang, S.-H., and Chen, A.-P. (2016). Financial time-series data analysis using deep convolutional neural networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 87–92. IEEE. (Cited on page 14).

[Chen et al., 2020] Chen, Q., Zhang, W., and Lou, Y. (2020). Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. *IEEE Access*, 8:117365–117376. (Cited on page 61).

[Chitra, 2011] Chitra, R. (2011). Technical analysis on selected stocks of energy sector. *International Journal of Management & Business Studies*, 1(1):42–46. (Cited on page 37).

[Chong et al., 2014] Chong, T. T.-L., Ng, W.-K., and Liew, V. K.-S. (2014). Revisiting the performance of macd and rsi oscillators. *Journal of risk and financial management*, 7(1):1–12. (Cited on page 37).

[Clerc and Kennedy, 2002] Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73. (Cited on page 87).

[Cooper, 1982] Cooper, J. C. (1982). World stock markets: Some random walk tests. *Applied Economics*, 14(5):515–531. (Cited on page 7).

[Crawford et al., 2018] Crawford, B., Soto, R., San Martín, M. A., de la Fuente-Mella, H., Castro, C., and Paredes, F. (2018). Automatic high-frequency trading: An application to emerging chilean stock market. *Scientific Programming*, 2018. (Cited on page 82).

[Cui et al., 2009] Cui, W., Brabazon, A., and O'Neill, M. (2009). Efficient trade execution using a genetic algorithm in an order book based artificial stock market. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2023–2028. (Cited on page 82).

[Danthine, 1977] Danthine, J.-P. (1977). Martingale, market efficiency and commodity prices. *European Economic Review*, 10(1):1–17. (Cited on page 61).

[de Souza et al., 2018] de Souza, M. J. S., Ramos, D. G. F., Pena, M. G., Sobreiro, V. A., and Kimura, H. (2018). Examination of the profitability of technical analysis based on moving average strategies in brics. *Financial Innovation*, 4(1):3. (Cited on page 38).

[DeFusco et al., 2015] DeFusco, R. A., McLeavey, D. W., Pinto, J. E., and Runkle, D. E. (2015). *Quantitative investment analysis workbook*. John Wiley & Sons. (Cited on pages 1 and 3).

[Dennis Jr and Schnabel, 1996] Dennis Jr, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM. (Cited on page 16).

[Ding et al., 2015] Ding, Y., Cheng, L., Pedrycz, W., and Hao, K. (2015). Global nonlinear kernel prediction for large data set with a particle swarm-optimized interval support vector regression. *IEEE transactions on neural networks and learning systems*, 26(10):2521–2534. (Cited on page 82).

[Dixon, 2018] Dixon, M. (2018). Sequence classification of the limit order book using recurrent neural networks. *Journal of computational science*, 24:277–286. (Cited on page 82).

[Dupernex, 2007] Dupernex, S. (2007). Why might share prices follow a random walk. *Student Economic Review*, 21(1):167–179. (Cited on page 7).

[Fama, 1991] Fama, E. F. (1991). Efficient capital markets: Ii. *The journal of finance*, 46(5):1575–1617. (Cited on page 20).

[Fama, 1995] Fama, E. F. (1995). Random walks in stock market prices. *Financial analysts journal*, 51(1):75–80. (Cited on page 8).

[Forsyth et al., 2012] Forsyth, P., Kennedy, J., Tse, S., and Windcliff, H. (2012). Optimal trade execution: A mean quadratic variation approach. *Journal of Economic Dynamics and Control*, 36(12):1971 – 1991. (Cited on page 82).

[Forsyth, 2011] Forsyth, P. A. (2011). A hamilton jacobi bellman approach to optimal trade execution. *Applied Numerical Mathematics*, 61(2):241 – 265. (Cited on page 82).

[Frennberg and Hansson, 1993] Frennberg, P. and Hansson, B. (1993). Testing the random walk hypothesis on swedish stock prices: 1919–1990. *Journal of Banking & Finance*, 17(1):175–191. (Cited on page 8).

[Gencay, 1998] Gencay, R. (1998). The predictability of security returns with simple technical trading rules. *Journal of Empirical Finance*, 5(4):347–359. (Cited on page 38).

[Gencay and Stengos, 1998] Gencay, R. and Stengos, T. (1998). Moving average rules, volume and the predictability of security returns with feedforward networks. *Journal of Forecasting*, 17(5-6):401–414. (Cited on page 23).

[Glabadanidis, 2015] Glabadanidis, P. (2015). Market timing with moving averages. *International Review of Finance*, 15(3):387–425. (Cited on page 38).

[Glaser et al., 2004] Glaser, M., Nöth, M., and Weber, M. (2004). Behavioral finance. *Blackwell handbook of judgment and decision making*, pages 527–546. (Cited on page 6).

[Gori and Tesi, 1992] Gori, M. and Tesi, A. (1992). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86. (Cited on page 38).

[Granger, 1992] Granger, C. W. (1992). Forecasting stock market prices: Lessons for forecasters. *International Journal of Forecasting*, 8(1):3–13. (Cited on page 14).

[Groda and Vrbka, 2017] Groda, B. and Vrbka, J. (2017). Prediction of stock price developments using the box-jenkins method. In *SHS Web of Conferences*, volume 39, page 01007. EDP Sciences. (Cited on pages 60 and 62).

[Guresen et al., 2011] Guresen, E., Kayakutlu, G., and Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38(8):10389 – 10397. (Cited on pages 14 and 61).

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182. (Cited on page 38).

[Hasibuan et al., 2019] Hasibuan, D., Jaya, I. K., Rumahorbo, B., Naibaho, J., Napitupulu, J., and Rajagukguk, E. (2019). Time series financial market forecasting based on support vector regression algorithm. In *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM)*, pages 1–4. IEEE. (Cited on page 81).

[He et al., 2010] He, Q., Tan, S., and Wanshan, D. (2010). DWO based predictive control for nonlinear systems. In *2010 International Conference on Measuring Technology and Mechatronics Automation*, volume 2, pages 38–41. IEEE. (Cited on page 62).

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507. (Cited on page 39).

[Hocking, 1976] Hocking, R. R. (1976). A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics*, 32(1):1–49. (Cited on page 38).

[Hoque et al., 2007] Hoque, H. A., Kim, J. H., and Pyun, C. S. (2007). A comparison of variance ratio tests of random walk: A case of asian emerging stock markets. *International Review of Economics & Finance*, 16(4):488–502. (Cited on page 8).

[Horák and Krulickỳ, 2019] Horák, J. and Krulickỳ, T. (2019). Comparison of exponential time series alignment and time series alignment using artificial neural networks by example of prediction of future development of stock prices of a specific company. In *SHS Web of Conferences*, volume 61, page 01006. EDP Sciences. (Cited on page 61).

[Idowu et al., 2012] Idowu, P. A., Osakwe, C., Aderonke, A. K., and Adagunodo, E. R. (2012). Prediction of stock market in nigeria using artificial neural network. *International Journal of Intelligent Systems and Applications*, 4(11):68. (Cited on page 20).

[Ince and Trafalis, 2008] Ince, H. and Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, 37(6):677–687. (Cited on page 3).

[Jianhong, 2015] Jianhong, W. (2015). Improvement on direct weight optimization identification. *Journal of Control and Systems Engineering*, 3:1–9. (Cited on page 62).

[Johnston et al., 1999] Johnston, F., Boyland, J., Meadows, M., and Shale, E. (1999). Some properties of a simple moving average when applied to forecasting a time series. *Journal of the Operational Research Society*, 50(12):1267–1271. (Cited on page 23).

[Kanas, 2001] Kanas, A. (2001). Neural network linear forecasts for stock returns. *International Journal of Finance & Economics*, 6(3):245–254. (Cited on page 14).

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE. (Cited on page 87).

[Khaidem et al., 2016] Khaidem, L., Saha, S., and Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*. (Cited on page 37).

[Khoda et al., 1992] Khoda, K., Liu, Y., and Storey, C. (1992). Generalized polak-ribiere algorithm. *Journal of Optimization Theory and Applications*, 75(2):345–354. (Cited on page 16).

[Kiarashinejad et al., 2020] Kiarashinejad, Y., Abdollahramezani, S., and Adibi, A. (2020). Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures. *npj Computational Materials*, 6(1):1–12. (Cited on page 39).

[Kim and Shamsuddin, 2008] Kim, J. H. and Shamsuddin, A. (2008). Are asian stock markets efficient? evidence from new multiple variance ratio tests. *Journal of Empirical Finance*, 15(3):518–532. (Cited on page 7).

[Kim, 2003] Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319. (Cited on pages 46 and 121).

[Kim and Han, 2000] Kim, K.-j. and Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, 19(2):125–132. (Cited on page 60).

[Kimoto et al., 1990] Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *1990 IJCNN international joint conference on neural networks*, pages 1–6. IEEE. (Cited on page 14).

[Kitagawa, 1996] Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25. (Cited on page 102).

[Koenker and Hallock, 2001] Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, 15(4):143–156. (Cited on page 62).

[Konishi, 2002] Konishi, H. (2002). Optimal slice of a vwap trade. *Journal of Financial Markets*, 5(2):197–221. (Cited on page 82).

[Lahmiri, 2016] Lahmiri, S. (2016). Intraday stock price forecasting based on variational mode decomposition. *Journal of Computational Science*, 12:23–27. (Cited on page 14).

[Lai and Wong, 2015] Lai, P.-f. B. and Wong, W. K. (2015). An empirical study of relationship between share price and intrinsic value of company. *Financial Studies*, 19(4). (Cited on page 1).

[Larson, 2007] Larson, M. (2007). Moving average convergence/divergence (macd). (Cited on page 23).

[Lee et al., 1999] Lee, C. M., Myers, J., and Swaminathan, B. (1999). What is the intrinsic value of the dow? *The Journal of Finance*, 54(5):1693–1741. (Cited on page 1).

[Li and Ye, 2013] Li, H. and Ye, X. (2013). A dynamic, volume-weighted average price approach based on the fast fourier transform algorithm. *Asia-Pacific Journal of Financial Studies*, 42(6):969–991. (Cited on page 82).

[Li et al., 2016] Li, Y., Li, X., and Wang, H. (2016). Based on multiple scales forecasting stock price with a hybrid forecasting system. *American Journal of Industrial and Business Management*, 6(11):1102–1112. (Cited on page 14).

[Lim and Coggins, 2005] Lim, M. and Coggins, R. J. (2005). Optimal trade execution: an evolutionary approach. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1045–1052. IEEE. (Cited on page 82).

[Lin, 2018] Lin, Q. (2018). Technical analysis and stock return predictability: An aligned approach. *Journal of financial markets*, 38:103–123. (Cited on page 38).

[Lin et al., 2013] Lin, Y., Guo, H., and Hu, J. (2013). An svm-based approach for stock market trend prediction. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE. (Cited on page 61).

[Liu and Lai, 2017] Liu, X. and Lai, K. K. (2017). Intraday volume percentages forecasting using a dynamic svm-based approach. *Journal of Systems Science and Complexity*, 30(2):421–433. (Cited on page 82).

[Lo and MacKinlay, 1988] Lo, A. W. and MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *The review of financial studies*, 1(1):41–66. (Cited on page 8).

[Lo et al., 2000] Lo, A. W., Mamaysky, H., and Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of finance*, 55(4):1705–1765. (Cited on page 37).

[Lundholm and O'keefe, 2001] Lundholm, R. and O'keefe, T. (2001). Reconciling value estimates from the discounted cash flow model and the residual income model. *Contemporary Accounting Research*, 18(2):311–335. (Cited on page 2).

[Mahfoud and Mani, 1996] Mahfoud, S. and Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied artificial intelligence*, 10(6):543–566. (Cited on page 61).

[Malkiel, 2003] Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1):59–82. (Cited on page 7).

[MathWorks, 2010] MathWorks (2010). Matlab. *Mathworks.* (Cited on page 16).

[Mitchell et al., 2020] Mitchell, D., Bialkowski, J., and Tompaidis, S. (2020). Volume-weighted average price tracking: A theoretical and empirical study. *IISE Transactions*, 52(8):864–889. (Cited on page 82).

[Moghaddam et al., 2016] Moghaddam, A. H., Moghaddam, M. H., and Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21(41):89 – 93. (Cited on page 14).

[Mostafa, 2010] Mostafa, M. M. (2010). Forecasting stock exchange movements using neural networks: Empirical evidence from kuwait. *Expert Systems with Applications*, 37(9):6302–6309. (Cited on page 20).

[Naeini et al., 2010] Naeini, M. P., Taremian, H., and Hashemi, H. B. (2010). Stock market value prediction using neural networks. In *2010 international conference on computer information systems and industrial management applications (CISIM)*, pages 132–136. IEEE. (Cited on page 14).

[Ni et al., 2015] Ni, Y., Liao, Y.-C., and Huang, P. (2015). Momentum in the chinese stock market: Evidence from stochastic oscillator indicators. *Emerging Markets Finance and Trade*, 51(sup1):S99–S110. (Cited on page 48).

[Osborne, 1959] Osborne, M. F. (1959). Brownian motion in the stock market. *Operations research*, 7(2):145–173. (Cited on page 61).

[Park and Irwin, 2007] Park, C.-H. and Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, 21(4):786–826. (Cited on page 38).

[Pemy, 2012] Pemy, M. (2012). Optimal algorithms for trading large positions. *Automatica*, 48(7):1353 – 1358. (Cited on pages 7, 82, and 84).

[Poli et al., 2007] Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57. (Cited on page 88).

[Poshakwale, 2002] Poshakwale, S. (2002). The random walk hypothesis in the emerging indian stock market. *Journal of Business Finance & Accounting*, 29(9-10):1275–1299. (Cited on page 8).

[Powell, 1977] Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical programming*, 12(1):241–254. (Cited on page 16).

[Pradeepkumar and Ravi, 2017] Pradeepkumar, D. and Ravi, V. (2017). Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. *Applied Soft Computing*, 58:35–52. (Cited on page 62).

[Qin et al., 2014] Qin, Q., Wang, Q.-G., Ge, S. S., and Ramakrishnan, G. (2014). Chinese stock price and volatility predictions with multiple technical indicators. *Soft-Computing in Capital Market: Research and Methods of Computational Finance for Measuring Risk of Financial Instruments*, 109. (Cited on page 38).

[Qiu et al., 2020] Qiu, J., Wang, B., and Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1):e0227222. (Cited on page 14).

[Qiu and Song, 2016] Qiu, M. and Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PloS one*, 11(5):e0155133. (Cited on page 14).

[Quah and Srinivasan, 1999] Quah, T.-S. and Srinivasan, B. (1999). Improving returns on stock investment through neural network selection. *Expert Systems with Applications*, 17(4):295 – 301. (Cited on page 14).

[Rao et al., 2020] Rao, P. S., Srinivas, K., and Mohan, A. K. (2020). A survey on stock market prediction using machine learning techniques. In *ICDSMLA 2019*, pages 923–931. Springer. (Cited on page 61).

[Raudys et al., 2013] Raudys, A., Lenčiauskas, V., and Malčius, E. (2013). Moving averages for financial data smoothing. In *International Conference on Information and Software Technologies*, pages 34–45. Springer. (Cited on page 23).

[Rawlings et al., 2017] Rawlings, J. B., Mayne, D. Q., and Diehl, M. (2017). *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI. (Cited on pages 83 and 85).

[Rech et al., 2001] Rech, G., Teräsvirta, T., and Tschernig, R. (2001). A simple variable selection technique for nonlinear models. *Communications in Statistics-Theory and Methods*, 30(6):1227–1241. (Cited on page 38).

[Refenes et al., 1994] Refenes, A. N., Zapranis, A., and Francis, G. (1994). Stock performance modeling using neural networks: a comparative study with regression models. *Neural networks*, 7(2):375–388. (Cited on page 14).

[Ritter, 2003] Ritter, J. R. (2003). Behavioral finance. *Pacific-Basin finance journal*, 11(4):429–437. (Cited on page 6).

[Roll et al., 2005a] Roll, J., Nazin, A., and Ljung, L. (2005a). A general direct weight optimization framework for nonlinear system identification. *IFAC Proceedings Volumes*, 38(1):178–183. (Cited on page 62).

[Roll et al., 2005b] Roll, J., Nazin, A., and Ljung, L. (2005b). Nonlinear system identification via direct weight optimization. *Automatica*, 41(3):475–490. (Cited on page 62).

[Salvador et al., 2018] Salvador, J. R., de la Peña, D. M., Alamo, T., and Bemporad, A. (2018). Data-based predictive control via direct weight optimization. *IFAC-PapersOnLine*, 51(20):356–361. (Cited on pages 62 and 63).

[Salvador et al., 2020] Salvador, J. R., Muñoz de la Peña, D., Ramirez, D., and Alamo, T. (2020). Predictive control of a water distribution system based on process historian data. *Optimal Control Applications and Methods*, 41(2):571–586. (Cited on pages 62 and 63).

[Samarawickrama and Fernando, 2017] Samarawickrama, A. and Fernando, T. (2017). A recurrent neural network approach in predicting daily stock prices an application to the sri lankan stock market. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6. IEEE. (Cited on page 20).

[Seidy, 2016] Seidy, E. E. (2016). A new particle swarm optimization based stock market prediction technique. *International Journal of Advanced Computer Science and Applications*, 7(2). (Cited on page 81).

[Senol and Ozturan, 2009] Senol, D. and Ozturan, M. (2009). Stock price direction prediction using artificial neural network approach: The case of turkey. *Journal of Artificial Intelligence*. (Cited on page 20).

[Shiller, 2000] Shiller, R. C. (2000). Irrational exuberance. *Philosophy and Public Policy Quarterly*, 20(1):18–23. (Cited on page 5).

[Shively, 2003] Shively, P. A. (2003). The nonlinear dynamics of stock prices. *The Quarterly Review of Economics and Finance*, 43(3):505–517. (Cited on page 14).

[Shrieves and Wachowicz Jr, 2001] Shrieves, R. E. and Wachowicz Jr, J. M. (2001). Free cash flow (fcf), economic value added (eva), and net present value (npv):. a reconciliation of variations of discounted-cash-flow (dcf) valuation. *The engineering economist*, 46(1):33–52. (Cited on page 2).

[Song et al., 2014] Song, J. H., de Prado, M. L., Simon, H. D., and Wu, K. (2014). Exploring irregular time series through non-uniform fast fourier transform. In *2014 Seventh Workshop on High Performance Computational Finance*, pages 37–44. IEEE. (Cited on page 82).

[Sougiannis and Yaekura, 2001] Sougiannis, T. and Yaekura, T. (2001). The accuracy and bias of equity values inferred from analysts' earnings forecasts. *Journal of Accounting, Auditing & Finance*, 16(4):331–362. (Cited on page 2).

[Stace, 2007] Stace, A. W. (2007). A moment matching approach to the valuation of a volume weighted average price option. *International Journal of Theoretical and Applied Finance*, 10(01):95–110. (Cited on page 82).

[Stanković et al., 2015] Stanković, J., Marković, I., and Stojanović, M. (2015). Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Procedia Economics and Finance*, 19:51–62. (Cited on page 38).

[Thaler, 2005] Thaler, R. H. (2005). *Advances in behavioral finance, Volume II*, volume 2. Princeton University Press. (Cited on page 6).

[Thomason, 1999] Thomason, M. (1999). The practitioner methods and tool. *Journal of Computational Intelligence in Finance*, 7(3):36–45. (Cited on page 70).

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288. (Cited on page 65).

[Tiwari, 2016] Tiwari, R. (2016). Intrinsic value estimates and its accuracy: Evidence from indian manufacturing industry. *Future Business Journal*, 2(2):138–151. (Cited on page 1).

[Trafalis and Ince, 2000] Trafalis, T. B. and Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 6, pages 348–353. IEEE. (Cited on page 81).

[Upadhyay et al., 2016] Upadhyay, V. P., Panwar, S., Merugu, R., and Panchariya, R. (2016). Forecasting stock market movements using various kernel functions in support vector machine. In *Proceedings of the International Conference on Advances in Information Communication Technology —& Computing*, AICTC '16, New York, NY, USA. Association for Computing Machinery. (Cited on page 70).

[Virtanen and Yli-Olli, 1987] Virtanen, I. and Yli-Olli, P. (1987). Forecasting stock market prices in a thin security market. *Omega*, 15(2):145–155. (Cited on page 20).

[Vrbka and Rowland, 2017] Vrbka, J. and Rowland, Z. (2017). Stock price development forecasting using neural networks. In *SHS Web of Conferences*, volume 39, page 01032. EDP Sciences. (Cited on page 38).

[Wafi et al., 2015] Wafi, A. S., Hassan, H., and Mabrouk, A. (2015). Fundamental analysis models in financial markets–review study. *Procedia economics and finance*, 30:939–947. (Cited on page 37).

[Wang et al., 2012a] Wang, J., He, H., and Prokhorov, D. V. (2012a). A folded neural network autoencoder for dimensionality reduction. *Procedia Computer Science*, 13:120–127. (Cited on page 39).

[Wang and Kim, 2018] Wang, J. and Kim, J. (2018). Predicting stock price trend using macd optimized by historical volatility. *Mathematical Problems in Engineering*, 2018. (Cited on page 38).

[Wang et al., 2018] Wang, J., Sun, T., Liu, B., Cao, Y., and Wang, D. (2018). Financial markets prediction with deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 97–104. IEEE. (Cited on page 81).

[Wang et al., 2012b] Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., and Guo, S.-P. (2012b). Stock index forecasting based on a hybrid model. *Omega*, 40(6):758–766. (Cited on page 14).

[Wang et al., 2011] Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., and Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, 38(11):14346–14355. (Cited on page 19).

[White, 1988] White, H. (1988). Economic prediction using neural networks: The case of ibm daily stock returns. In *ICNN*, volume 2, pages 451–458. (Cited on pages 61 and 81).

[Wong et al., 2003] Wong, W.-K., Manzur, M., and Chew, B.-K. (2003). How rewarding is technical analysis? evidence from singapore stock market. *Applied Financial Economics*, 13(7):543–551. (Cited on page 37).

[Wu and Van Voorhis, 2005] Wu, Q. and Van Voorhis, T. (2005). Direct optimization method to study constrained systems within density-functional theory. *Physical Review A*, 72(2):024502. (Cited on page 62).

[Ye and Huang, 2008] Ye, C. and Huang, J. (2008). Non-classical oscillator model for persistent fluctuations in stock markets. *Physica A: Statistical Mechanics and its Applications*, 387(5-6):1255–1263. (Cited on page 48).

[Ye and Sun, 2018] Ye, M. and Sun, Y. (2018). Variable selection via penalized neural network: a drop-out-one loss approach. In *International Conference on Machine Learning*, pages 5620–5629. (Cited on page 38).

[Ye et al., 2014] Ye, X., Yan, R., and Li, H. (2014). Forecasting trading volume in the chinese stock market based on the dynamic vwap. *Studies in Nonlinear Dynamics & Econometrics*, 18(2):125–144. (Cited on page 82).

[Yoon and Swales, 1991] Yoon, Y. and Swales, G. (1991). Predicting stock price performance: A neural network approach. In *Proceedings of the twenty-fourth annual Hawaii international conference on system sciences*, volume 4, pages 156–162. IEEE. (Cited on page 14).

[Yu and Yan, 2020] Yu, P. and Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, 32:1609–1628. (Cited on page 61).

[Yu and Chen, 1995] Yu, X.-H. and Chen, G.-A. (1995). On the local minima free condition of backpropagation learning. *IEEE Transactions on Neural Networks*, 6(5):1300–1303. (Cited on page 38).

[Yuan and Lin, 2006] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67. (Cited on page 38).

[Zhang et al., 1998] Zhang, G., Patuwo, B. E., and Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62. (Cited on page 14).

[Zhang, 2007] Zhang, G. P. (2007). A neural network ensemble method with jittered training data for time series forecasting. *Information Sciences*, 177(23):5329–5346. (Cited on page 14).

[Zhang and Suganthan, 2016] Zhang, L. and Suganthan, P. N. (2016). A survey of randomized algorithms for training neural networks. *Information Sciences*, 364:146–155. (Cited on page 14).

[Zhang et al., 2006] Zhang, L., Zhou, W., and Li, D.-H. (2006). A descent modified polak–ribière–polyak conjugate gradient method and its global convergence. *IMA Journal of Numerical Analysis*, 26(4):629–640. (Cited on page 16).

[Zheng and Zhu, 2017] Zheng, C. and Zhu, J. (2017). Research on stock price forecast based on gray relational analysis and armax model. In *2017 International Conference on Grey Systems and Intelligent Services (GSIS)*, pages 145–148. (Cited on page 61).