EVALUATING EFFICIENCY OF ENSEMBLE CLASSIFIERS IN PREDICTING THE JSE ALL-SHARE INDEX ATTITUDE

Shaun Ramsumar

A research report submitted to the Faculty of Commerce, Law and Management, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Management in Finance and Investment.

Johannesburg, 2016

Declaration

I declare that this research report is my own unaided work. It is being submitted to the Degree of Master of Management in Finance and Investment to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

_____

Shaun Ramsumar

Signed this _____ day of _____ , _____

Abstract

The prediction of stock price and index level in a financial market is an interesting but highly complex and intricate topic. Advancements in prediction models leading to even a slight increase in performance can be very profitable. The number of studies investigating models in predicting actual levels of stocks and indices however, far exceed those predicting the direction of stocks and indices. This study evaluates the performance of ensemble prediction models in predicting the daily direction of the JSE All-Share index. The ensemble prediction models are benchmarked against three common prediction models in the domain of financial data prediction namely, support vector machines, logistic regression and k-nearest neighbour. The results indicate that the Boosted algorithm of the ensemble prediction model is able to predict the index direction the best, followed by k-nearest neighbour, logistic regression and support vector machines respectively. The study suggests that ensemble models be considered in all stock price and index prediction applications.

Acknowledgements

I would like to thank Dr Thabang Mokoaleli-Mokoteli from the Wits Business School of the University of the Witwatersrand who inspired the topic and provided her undivided attention and support during the research. I would also like to thank Dr Yarish Brijmohan for his suggestions, guidance and continued support.

Table of Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

List of Tables

Chapter 3

Chapter 4

Appendix A

Appendix B

Nomenclature

ADO	Accumulation Distribution Oscillator
AR	Auto Regressive
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
CCI	Commodity Channel Index
DJIA	Dow Jones Industrial Average
EMH	Efficient Market Hypothesis
FN	False Negative
FP	False Positive
JSE	Johannesburg Stock Exchange
KNN	k-Nearest Neighbour
LR	Logistic Regression
MACD	Moving Average Convergence Divergence
MOEA	Multi-Objective Evolutionary Algorithm
MOM	Momentum
PCC	Percentage Correctly Classified
ROC	Receiver Operating Characteristic
RSI	Relative Strength Index
SMA	Simple Moving Average
STCD	Stochastic D Indicator
STCK	Stochastic K Indicator
SVAR	Structural Vector Auto Regression
SVC	Support Vector Classification
SVM	Support Vector Machines
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
WILLR	Larry Williams %R Indicator
WMA	Weighted Moving Average

Chapter 1

Introduction

1.1 Introduction

The purpose of this chapter is to provide a broad introduction to the research report including the motivation for the research. The chapter is organised as follows: Section 1.2 presents the research background. Section 1.3 discusses the research problem. Section 1.4 follows from the research problem and identifies the gap in the literature. Section 1.5 outlines the two research objectives. Section 1.6 presents the benefits of the study. Section 1.7 presents the organisation of the entire report and chapter summary concludes the chapter.

1.2 Background

The prediction of stock price direction is a topic of significant interest in the field of modern finance and investments. A great deal of literature presents arguments as to whether stock price behaviour is predictable or not. The task of predicting such movements however, prove both challenging and intricate in nature. The challenging nature of the prediction is as a result of multiple non-predictable factors like natural disasters, political instabilities, varying economic climates etc. Reasonably accurate predictions in stock price movements can however, result in high financial gains for speculators and arbitrageurs and can also assist to hedge against potential market risks (Kumar & Thenmozhi, 2006). In the area of automated trading, stock price predictive models often serve as the foundation of such intricate algorithms (Manojlovic & Stajduhar, 2015).

The Efficient Market Hypothesis (EMH) has, for a long time, been an accepted hypothesis by investors globally. The hypothesis states that no abnormal returns can be achieved by knowledge of the stock prices past behaviour and that all information about a stock is already incorporated in its price. It also states that one cannot constantly achieve returns in excess of the market average. At the dawn of the 21st century, some economists presented arguments to support the fact that stock prices are at least partially predictable (Malkiel, 2003). Since then, many researchers have explored a myriad of prediction algorithms, models and techniques in the quest to create a model that can accurately predict stock price behaviour.

Investors generally have three options when it comes to analysis of a stock prior to buy or sell decisions. The first option is the fundamental analysis where the intrinsic value of the stock is determined in conjunction with the industry's performance, the political and economic climate. The second option is a technical analysis where a stock's value is determined by studying its detailed statistics. Technical analysis makes no attempts to measure a stocks intrinsic value. It merely uses vast quantities of statistical data to help identify trends and patterns that may be interpreted as predictors for future performance. To achieve such predictions, the historical time series data is used as an input to complex algorithms which attempt to model and then forecast the future time series e.g. auto regressive (AR), moving average (MA) and auto regressive integrated moving average (ARIMA) models. The third option involves machine learning and data mining (Hellstrom & Holmstromm, 1998). This option is often desired among analysts because one of the main challenges around stock price prediction involves working with large masses of data.

1.3 Research Problem

Investors use various models to predict stock price direction. Single classifier models such as logistic regression (LR), neural networks (NN), k-nearest neighbour (KN) and support vector machines (SVM) are currently the most common and widely used machine learning models (Ballings et al., 2015). Although attempting to predict stock prices is in contravention of the EMH, many researchers e.g. Malkiel (2003) and Lo, Mamaysky & Wang (2000), reject the EMH and continue to explore complex machine learning algorithms with the aim of accurately modelling the complex dynamics that characterise financial data. de Oliveira, Nobre, & Zárate (2013) suggest that combining prediction models can achieve better performance than standalone models.

There exist a large number of single classifier machine prediction models for the purpose of stock or index direction and level prediction. The ensemble prediction models, however, are relatively unexplored in the domain of stock market trend prediction (Kumar & Thenmozhi, 2006; Kara, Boyacioglu, & Baykan, 2011; Ballings et al., 2015). Thus, we do not know whether ensemble prediction models can accurately predict the daily direction of the stock market especially in a relatively volatile emerging market like South Africa.

1.4 Gap in the Literature

It was found that emerging markets are generally more predictable than developed markets and that emerging market returns are more influenced by local information than developed markets (Harvey, 1995). This finding motivated this research into determining the predictability of the South African stock market index by evaluating prediction models that proved to perform in developed markets. This study is of significance as it evaluates prediction models that are relatively new to the field of technical analysis in the stock market.

Machine learning and data mining make available two ways of predicting stock market behaviour. The first way is to predict the actual future price of the stock. This is referred to as discrete analysis and a way of predicting exact stock prices (Ballings, Van Den Poel, Hespeels, & Gryp, 2015). The second way is based on predicting the actual future direction of the stock. This is where a prediction is made as to whether the future price of the stock will rise or fall in relation to the current known price. The models used to predict the price direction of stocks is commonly referred to as classification models. There are considerably fewer studies around stock price direction prediction than actual price prediction (Manojlovic & Stajduhar, 2015). In recent years, there has been a significant increase in the number studies looking at the direction or trends in financial markets (Imandoust & Bolandraftar, 2014). Literature also reveals that prediction of stock price direction is sufficient in producing profitable trading strategies (Cheung, Chinn, & Pascual, 2005).

In comparison to single classification prediction models, ensemble prediction models are far less utilised in stock market trend prediction (Kumar & Thenmozhi, 2006; Kara, Boyacioglu, & Baykan, 2011; Ballings et al., 2015). Ensemble models proved to perform the best in predicting European stocks (Ballings et al., 2015) but failed to perform in the Indian stock market (Kumar & Thenmozhi, 2006) where SVM proved to perform the best. No published literature was found that evaluates the performance of ensemble prediction models in any financial time series from the African continent. It would therefore prove useful to evaluate the performance of ensemble prediction models in the South African market by attempting to predict the daily trend of the JSE All-Share index.

1.5 Research Objectives

1. To investigate whether ensemble prediction models are able to predict the daily trend of the JSE All-Share index and determine the prediction accuracy.
2. To compare the performance of ensemble prediction models with the performance of three most popular models (SVM, KNN and LR) in predicting the daily trend of the JSE All-Share index

1.6 Benefits of the Study

The results from this study will help market analysts in making better decisions regarding choice of prediction models in their technical analysis. This will translate to better decision making in hedging against risks, developing efficient market trading strategies and even profiting from more accurate forecasts. The study will also provide insight into the technical characteristics of the South African JSE All-Share Index and establish the extent of its predictability using the models that worked best for developed markets and other developing markets.

1.7 Organisation of the Report

This report is divided into five chapters. Chapter 2 reviews the extant literature relevant to the research. Chapter 3 presents the research methodology including the research design. Chapter 4 presents the research results and Chapter 5 discusses the research findings and concludes the report.

Chapter Summary

This chapter provides a background to the field of financial time series prediction with regard to stock price and market indices. An introduction to the basic concepts of fundamental analysis and technical analysis is presented. Two research objectives is established and supported by a presentation of the gaps in the literature. The benefits of the study is then presented by establishing how financial analysts can capitalise from prediction models. In Chapter 2, the extant literature in the field of financial prediction models is presented.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents a summary of recent literature in the domain of stock price and stock index trend prediction. The chapter is organised as follows. Section 2.2 discusses the literature on the efficient market hypothesis. Section 2.3 presents the literature on fundamental analysis. Section 2.4 presents the literature on technical analysis. Section 2.5 discusses machine prediction algorithms including the different models used. Chapter summary concludes the chapter.

2.2 The Efficient Market Hypothesis

Market efficiency remains one of the most debated and controversial topics in modern investment theory. An efficient market is defined as one where the current market price of assets fully reflect the available information within the market (Firer et al., 2012). The efficient market hypothesis therefore states that one cannot consistently, beat the market. Fama (1991) argued that asset prices in an efficient market is subject to random behaviour. Jordan, Miller, & Dolvin (2015) state three economic forces that lead to market efficiency: one, investor rationality, two, independent deviations from rationality and three, arbitrage. At the start of the 21st century, Malkiel (2003) presented arguments by various economists that stock prices are indeed, partially predictable. This set the stage for prediction models and algorithms in the dynamic, complex and interelated financial markets.

There are numerous studies that aim to dispute the efficient market hypothesis by presenting empirical evidence from various financial markets. Hu et al., (2015) developed a hybrid trend following algorithm which combines the information inherent in the trend of a stock and extended classification theory. This results in a trading rule that identifies stocks via key indicators. Kao et al., (2013) suggested a stock forecasting model based on wavelet transforms, support vectors and regression splines to improve forecasting accuracy. Patel et al. (2015) introduced the concept of trending technical indicators to improve prediction model accuracy in predicting the CNX NIFTY and S&P BSE index. Booth, Gerding, & McGroarty (2014) proposed a machine learning technique using random forest to predict both risk and returns when considering seasonal events. Liao & Chou (2013) proposed association rules and cluster theory in describing the co-movements between the stock markets of China and Taiwan.

2.3 Fundamental Analysis

Fundamental analysis is a type of analysis where the intrinsic value of an asset is determined. This is performed by studying various financial and economic indicators. Equity prices generally follow movements in macroeconomic variables since firms dividends are directly linked to the macroeconomic environment. There are various authors that debate whether stock prices are reflected by fundamental factors. Black, Fraser, & Groenewold (2003), Becchetti, Rocci, & Trovato (2007) and Laopodis (2011), conclude that equity prices consistently deviate from their fundamental values. These findings corroborate the findings of Coakley & Fuertes (2006) and Manzan (2007) who found that stock prices do deviate from their fundamental values in the short run but revert to their fundamentals in the long run. In contrast, studies by Yuhn, Kim, & Nam (2015), Chen & Fraser (2010) and Pan (2007), all conclude that stock prices are priced in line with their fundamentals.

Velinova & Chen (2015) examined the role of macroeconomic fundamentals in relation to stock prices for six major industrialised countries using data from 1960 to 2013. One of the main research questions of the study was to determine how stock prices reacted relative to its fundamentals directly after the 2008 global financial crisis. The analysis in the study was based on the conventional bivariate structural vector autoregressive (SVAR) model in order to differentiate between fundamental and non-fundamental shocks to stock prices. The study revealed that stock prices increased steeply during the mid-1990s due to an undervaluation situation in the preceding period. After this mid-1990 period, the stocks became slightly overvalued with respect to their fundamentals. After the 2008 global financial crisis however, their value reverted back in line with their fundamentals. This reversion was particularly prominent in the US stock market. The study concluded that stock prices for the countries examined self-corrected toward their fundamental value in the long run.

2.4 Technical Analysis

Technical analysis is a process that attempts to predict the movement of stock or any other financial series based on an interrogation of the quantitative characteristics of the data available. This interrogation of the financial data involves methods such as graphic analysis, various techniques of averaging or combinations of both. From a methodological standpoint, technical analysis often incorporates models from

econometrics, statistics and artificial intelligence (Cervelló-Royo, Guijarro, & Michniuk, 2015). Technical analysis is based on the assumption that past data may contain important information about the future behaviour of the data (Zhu & Zhou, 2009). There are three assumptions presented by Murphy (1999) underlying the technical analysis and these are: One, prices reflect market events. Two, change in prices move in trends and last, historical prices tend to repeat.

Although technical analysis is in direct contravention of the EMH, many researchers have rejected this hypothesis on the basis of technical analysis. Silva, Neves, & Horta (2015) used a Multi-Objective Evolutionary Algorithm (MOEA) to optimise return and minimise risk. The study concluded that stocks with high valuation potential are characterised by low or average market capitalisation, low price earnings ratio, high revenue growth and high operating leverage. Cervelló-Royo et al. (2015) introduced a new trading rule based on a new breakout and consolidation flag pattern version which further challenge the efficacy of EMH. The trading rule defines when to buy and sell, the amount of profit pursued in each operation and maximum bearable loss. Cervelló-Royo et al. (2015) found that the returns generated when using the new trading rule were higher for the European indices compared to that of the US and, therefore, concluded the European markets suffered greater inefficiency than US markets.

There are also studies however, that reveal the low power of technical analysis. da Costa et al. (2015) analysed the performance of various averaging techniques in predicting stock behaviour in the Brazilian market. The study evaluated the performance of simple and exponential moving averages, moving average convergence divergence and triple screen techniques in actual trading of 198 Brazilian stocks. The study concluded that the investigated averaging and triple screen techniques had low power in predicting the Brazilian stock market and that the standard buy-and-hold strategy was responsible for the majority of the returns achieved in the investigation.

2.5 Machine Prediction Algorithms

The literature review reveals that there are many machine prediction models and algorithms used to predict stock price direction and levels. These models can be classified according to their level of complexity and performance characteristics. The simpler prediction models such as the single decision tree, discriminant analysis and

Naïve Bayes have been replaced by newer and better performing single classifier models such as logistic regression, neural networks, support vector machines and k-nearest neighbour. Ensemble models such as Random Forest (RF) or Bagged trees, Kernel Factory (KF) and AdaBoost (AB) or Boosted trees are still very much unexplored in the domain of stock price direction prediction (Ballings et al., 2015). Ballings et al. (2015) found that there is inadequate literature on ensemble prediction models in the domain of stock price direction prediction.

Figure 2.1 below illustrates the number of studies since 1990 for the four single classifier models and the three ensemble prediction models in the field of financial time series prediction (Ballings et al., 2015). From Figure 2.1, it is clear that ensemble prediction models are far less utilised in the domain of stock price and market index prediction compared to that of the single classifier prediction models.

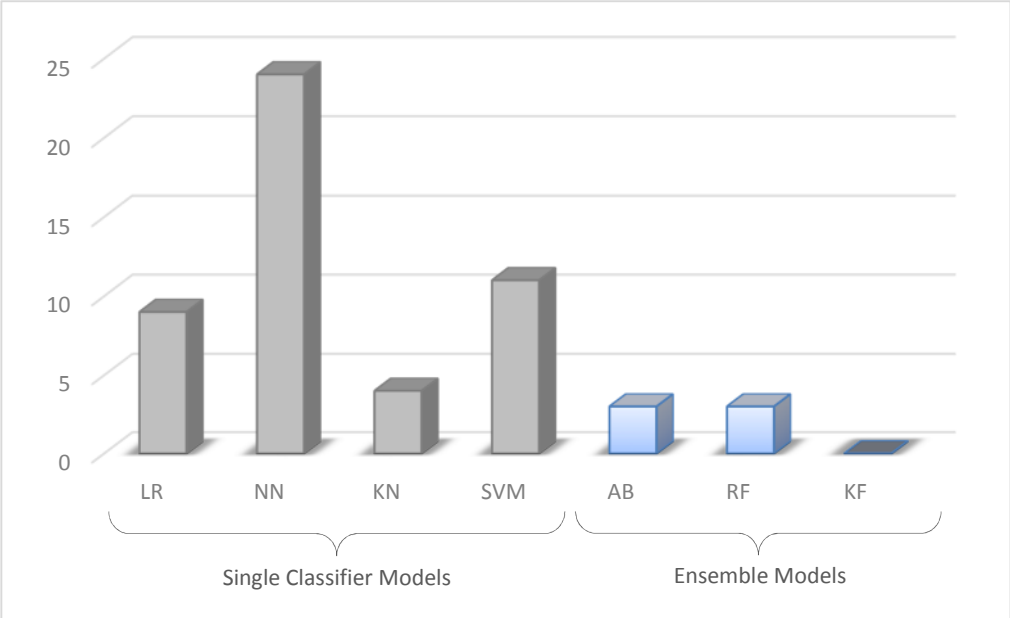

Figure 2.1: Number of studies since 1990 for stock price prediction (Ballings et al., 2015).

2.5.1 Ensemble Prediction Models

Ensemble prediction models solve problems that are statistical, computational and representational in nature. According to Ballings et al. (2015), only four studies exist in the domain of stock price direction prediction that utilises ensemble methods. Ensemble methods basically merge results from multiple weak methods in order to form one high quality prediction model. Many types or algorithm variations of ensemble prediction models exist and can be used in stock price and stock index direction prediction. Three of the most common types of ensemble algorithms are: One, Random Forest or Bagged trees. Two, Boosted trees and three, Subspace KNN.

The random forest ensemble method builds a collection of trees to improve upon the weak predictive capabilities of individual trees. Random forest was introduced by Leo Breiman in 2001 and proposed as an effective tool in any prediction application (Breiman, 2001). Rodriguez & Rodriguez (2004) predicted the daily movements of the Mexican, Malaysian and Brazilian market indices. Rodriguez & Rodriguez (2004) found that the ensemble prediction models performed the best among all the seven prediction models. Kumar & Thenmozhi (2006) however, found that the Random Forest ensemble prediction model underperforms against SVM in predicting the S&P CNX NIFTY index. Random Forest however, outperformed the other models (logistic regression, neural networks and Discriminant Analysis) in the study.

Patel et al. (2015) concluded that random forest outperforms the SVM, neural network and Naïve-Bayes models in predicting the CNX NIFTY and S&P BSE index movement. These results are in contrast with Kumar & Thenmozhi (2006) who found that SVM outperforms the ensemble models in predicting the CNX NIFTY market index. The reason for this difference in performance lies in the trend deterministic processing of the input data used by Patel et al. (2015) and the varying time periods of the index data used for each study. Ballings et al. (2015) findings ranked the prediction performance of three ensemble model algorithms in the following order: Random Forest, Kernel Factory and Boosted Trees, out of the seven models in the study in predicting stock price direction from 5767 publically listed European companies. The study was the first to make such a benchmark and highly encourage the use of ensemble methods in stock price direction prediction.

2.5.2 Support Vector Machines

SVMs are one of the most common machine learning algorithms in the field of stock price and stock index direction prediction. It appears to be one of the best performing algorithms in the financial world (Ballings et al., 2015). It is a specific type of machine learning algorithm that is characterised by the capacity control of the decision function, the use of kernel functions and sparsity of solution (Huang, Nakamori, & Wang, 2005). SVM uses a linear model to implement non-linear class boundaries via non-linear mapping of the input variables into a high-dimensional feature space (Kim, 2003). The accuracy of SVMs in financial forecasting and predictions are often measured by the deviations of the estimated values from the actual values. Predicting the actual values however, and using the errors as indicators of accuracy is of lesser importance and not as profitable to financial practitioners as accurate direction prediction.

Huang et al. (2005) investigated the predictability of the weekly movement of the NIKKEI 225 index, which indicated that the SVM outperformed other two classification models used in the study viz. the discriminant analysis model and back-propagation neural network model. Huang et al. (2005) further recommended that all models be combined in the study to achieve the best performance. In a 2003 Korean study, the direction of the daily change of the Korean composite stock price index (KOSPI) was predicted using SVM. A total of 12 technical indicators were used in the prediction model. The results also confirmed that SVM performed better than the neural network model and the case based reasoning models that were benchmarked in the study (Kim, 2003).

Lee (2009) investigated the predictability of the NASDAQ index using SVM with a hybrid feature selection. The SVM prediction models performance was compared to that of the back-propagation neural network model using three common feature selection methods. The study revealed that the SVM prediction model outperformed the back-propagation neural network model. A similar investigation by Kumar & Thenmozhi (2006) in the evaluation of the models that best predicts the direction of the S&P CNX NIFTY index of the National Stock Exchange showed that the SVM also outperformed the neural network, ensemble and discriminate models benchmarked in the study.

2.5.3 Logistic Regression Prediction Models

The logistic regression prediction is a common technique used in stock price and market index direction prediction. It arises from the need to model class probabilities via linear functions in the explanatory variables. The logistic regression model will only produce accurate predictions in stock price or stock index movements if and only if the parameterised equation resembles that of the true function (Rodriguez & Rodriguez, 2004). Logistic regression is similar to linear regression with the main difference being that linear models are characterised by a continuous response variable whereas the logistic model is characterised by a binary response variable. The result of logistic models, thus, uses maximum likelihood as opposed to least squares (Huang, Yang, & Chuang, 2008).

Although logistic regression models are one of the most popular prediction models in literature, they were found by Ballings et al. (2015) to perform the worst among all models in the evaluation of prediction models for European stock market. An investigation by Senol & Ozturan (2008) compared the performance of the logistic regression prediction model to a neural network model using data from the Istanbul Stock Exchange. It was found that the neural network prediction methodology statistically outperformed the logistic regression methodology in predicting the direction of stock prices in the market. In a study by Subha & Nambi (2012), the BSE-SENSEX and NSE-NIFTY stock index movement was predicted using logistic regression and its performance was compared to the k-nearest neighbour prediction model. The study concluded that k-nearest neighbour model far outperformed the logistic regression model in classifying the movement of the BSE-SENSEX and NSE-NIFTY stock indices. The study also found that the k-nearest neighbour outperformed the logistic regression model for all possible model evaluation parameters.

Ou & Wang (2009) compared the performance of ten classification techniques in predicting the price movement of the Hang Seng index of the Hong Kong stock market. In this study, the logistic regression model for predicting the stock price movement ranked third among the ten prediction models. The authors also argue that different stocks behave differently and recommend that all approaches and prediction models be explored in forecasting stock index movement.

2.5.4 Neural Network Prediction Models

The neural network is a common tool in the financial analysis arena particularly in the financial time series domain due to its broad applicability to business problems and its preeminent learning ability (Kumar & Thenmozhi, 2006). The neural network basically emulates the functioning of the human brain by forming a network of neurons. They are characterised by their learning capability and the ability to adjust their parameters by the use of a training set. A large number of successful financial applications have used neural networks in time series forecasting and stock market prediction. Neural networks, however exhibit inconsistent and unpredictable performance on noisy financial data and suffers in selecting from a large number of input financial variables (Huang et al, 2008).

According to de Oliveira et al. (2013), the first neural network model for predicting stock prices was by White (1988), where daily returns of the IBM stock was analysed in order to test the efficient market hypothesis. Although the model did not produce good predictive results, the research set the platform for further development of stock market predictive models using neural networks. The performance and comparisons of the traditional methods of stock prediction models to that of neural networks then began in the 1990s. Ballings et al. (2015) found that neural networks performed the fifth best out of a total of the seven stock price prediction models.

de Oliveira et al. (2013) found neural networks to be a feasible alternative to conventional techniques in predicting stock market direction and behaviour. The authors further suggest that neural networks prediction models be used in combination with other prediction models to achieve better predictive performance. In the study by Lee (2009), the NASDAQ index direction was predicted by comparing a hybrid version of an SVM model to a back-propagation neural network model. The study however favoured the SVM model over the neural network model in predicting the trend of the NASDAQ index. The authors suggest that their poor performance can be attributed to neural networks requiring large amounts of training data in order to formulate the distribution of the input data pattern. A neural networks over fitting nature also results in difficulties in generalising predictions.

Chapter Summary

In this chapter, the extant literature in the field of financial prediction models is presented. The literature relating to technical analysis, fundamental analysis and the EMH were reviewed. Machine prediction algorithms in the field of stock price and market index were discussed in detail. It was found that ensemble prediction models are not as popular in the fields of financial time series prediction as prediction models such as SVM, logistic regression, neural networks and k-nearest neighbour. Chapter 3 presents a detailed description of the data and methodology used in evaluating the performance of the ensemble prediction model. It also presents the methodologies for evaluating and comparing the performance of the ensemble prediction model to the SVM, logistic regression and k-nearest neighbour prediction models.

Chapter 3

Data and Methodology

3.1 Introduction

This chapter presents the methodologies used to test the efficacy of ensemble prediction models compared to other models. The chapter is organised as follows: Section 3.2 presents the data and the data sources by providing a detailed description of the JSE All-Share index time series data to be used in the study. Section 3.3 provides a presentation and a description of the ten technical indicators that is computed directly from the index time series dataset. The technical indicators are pre-processed, resulting in model input data used in the training and evaluation of the prediction models. Section 3.4 presents the pre-processing of the ten technical indicators which results in two forms of representation of them i.e. the continuous and the trend deterministic representation. These two forms of representation of the input data are used directly to train and evaluate the prediction models. Section 3.5 presents the process of prediction model training and evaluation. Section 3.6 presents a description of the ensemble prediction model for the index trend prediction. Section 3.7 presents a description of the three most common prediction models in the domain of stock price and index trend prediction applications i.e. SVM, k-nearest neighbour and logistic regression.

3.2 Data and Data Sources

The research data used in this empirical study is the daily data of the JSE All-Share index obtained from Bloomberg. The data comprise of the open, high, low and close daily index values that spanned from 1st August 2002 to 15th July 2016, totalling 3489 trading days of the JSE. Since this research evaluates prediction model performance in predicting the indices daily trend, each trading day closing index is compared to the previous day closing index. This comparison then yields a daily trend response assuming one of two values, up or down. As an example, the closing index value for the JSE All-Share on the 1st and 2nd of September 2011 was 31088.12 and 30518.92 respectively. This is regarded as a down trend response for trading the day, 2nd of September 2011, and this down trend is to be predicted on the 1st of September 2011.

Table 3.1 presents the number of up and down movements for each year in the time period of the research data. The table also presents the percentage of up and down movements for each year giving an indication of the volatility of the index on

an annual basis. In the 3489 trading days of the research data, 1869 days recorded up movements while 1619 days recorded down movements. The number of up trends is, on average, 8% more than the number of down trends over the 15 years of the index data.

This result is coherent considering the index closing value increased from 9216.30 on the 1st of August 2002 to 53088.46 on the 15th of July 2016, resulting in a 5.8 fold increase in the above time period.

Table 3.1: Annual number of up and down movements on the All-Share index.

Year	Down	%	Up	%	Total
2002	54	52%	50	48%	104
2003	123	49%	127	51%	250
2004	115	46%	136	54%	251
2005	105	42%	146	58%	251
2006	108	44%	140	56%	248
2007	108	43%	142	57%	250
2008	133	53%	118	47%	251
2009	114	46%	136	54%	250
2010	116	46%	135	54%	251
2011	121	49%	128	51%	249
2012	107	43%	143	57%	250
2013	114	46%	136	54%	250
2014	124	50%	125	50%	249
2015	121	48%	130	52%	251
2016	56	42%	77	58%	133
Total	1619	46%	1869	54%	3488

3.3 The Technical Indicators

The raw index data described in Section 3.2 above was used to generate a set of technical indicators that served as inputs to the various index trend prediction models, i.e., ensemble, SVM, LR and KNN. A total of ten technical indicators were used in this study as described in Kara et al. (2011), Patel et al. (2015), Kim (2003) and Kumar & Thenmozhi (2006). These technical indicators are relavant in stock prediction as fund managers and investment professionals often use them in their analysis and predictions of levels and trends in financial data. Table 3.2 presents the ten technical indicators with their respective equations. A description of each technical indicator is as follows:

3.3.1 Moving Averages

The simple moving average (SMA) and the weighted moving average (WMA) are basic technical analysis tools that are commonly used to smoothen out time series serial data by computing average levels of the serial data on a daily basis. In the current study, a ten day average on the closing index values were taken as in Kara et al. (2011) and Patel et al. (2015). Thus, the value of n as used in Table 3.2 for computing all technical indicators is ten days.

3.3.2 Momentum

Momentum (MOM), in the context of technical analysis on financial time series, is the difference between two price levels that is separated by a given number of periods, n. It is an indication of the rate of rise and fall of the market index. The momentum was calculated on the daily closing index values.

3.3.3 Stochastic Oscillators

Stochastic K (STCK), Stochastic D (STKD) and Larry Williams R% (WILLR) are all stochastic oscillator technical indicators. These oscillators are used to indicate trends in serial data. Increasing stochastic oscillators for closing index levels generally indicate an expected increase in future levels and vice-a-versa (Patel et al., 2015). The stochastic technical indicators use the lowest low and highest high index levels for a given time period as well as the high and low index levels for a particular day.

3.3.4 Relative Strength Index

Relative Strength Index (RSI) is a momentum indicator that measures a stock's price relative to itself and its past performance. The RSI function requires the index movements that are based on closing index values. When applied to common stocks, the RSI can be used to identify overbought and oversold points. If the RSI exceeds 70, it can be interpreted that the stock is overbought and its price is highly likely to drop in the near future. If the value falls below 30, it can be interpreted as the stock being oversold and its price is likely to go up in the near future (Patel et al., 2015).

Table 3.2: Technical indicators, their formulas and variable definition.

Simple Moving Average (SMA)	$\frac{C_t + C_{t-1} + \cdots + C_{t-n+1}}{n}$
Weighted Moving Average (WMA)	$\frac{(n) \times C_t + (n-1) \times C_{t-1} + \cdots + C_{t-n+1}}{n + (n-1) + (n-2) + \cdots + 1}$
Momentum (MOM)	$C_t - C_{t-n+1}$
Stochastic K% (STCK)	$\frac{C_t - LL_{t-n+1}}{HH_{t-n+1} - LL_{t-n+1}} \times 100$
Stochastic D% (STCD)	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{n}\%$
Relative Strength Index (RSI)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} UP_{t-i}/n)/(\sum_{i=0}^{n-1} DW_{t-i}/n)}$
Moving Avg. Convergence Divergence (MACD)	$MACD(n)_{t-1} + \frac{2}{n+1}(DIFF_t - MACD(n)_{t-1})$
Larry Williams R% (WILLR)	$\frac{H_n - C_t}{H_n - L_n} \times 100$
Accumulation Distribution Oscillator (ADO)	$\frac{H_n - C_{t-1}}{H_t - L_t}$
Commodity Channel Index (CCI)	$\frac{M_t - SM_t}{0{,}015 D_t}$

C_t, L_t, H_t is the closing price, low price and the high price respectively at time t.
$DIFF_t = EMA(12)_t$ - $EMA(26)_t$ where EMA is an exponential moving average given by the equation:
$EMA(k)_t = EMA(k)_{t-1} + \alpha(C_t\text{-}EMA(k)_{t-1})$
where a smoothing factor $\alpha = 2/(k+1)$ and k is the time period of k-day exponential moving average.
LL_t and HH_t is the lowest low and highest high level in the last t days.
$M_t = (H_t+L_t+C_t)/3$; $SM_t = (\sum M_{t\text{-}i\text{-}1})/n$; $D_t = (\sum|M_{t\text{-}i\text{-}1} - SM_t|)/n$
UP_t and DW_t is the upward and downward price change at time t respectively.

3.3.5 Moving Average Convergence Divergence

Moving Average Convergence Divergence (MACD) is a technical indicator that follows the trend of a stock. If MACD increases then the stock prices are expected to go up and vice-a-versa. MACD is computed from the closing index levels of the financial time series via the exponential moving average values.

3.3.6 Accumulation Distribution Oscillator

The Accumulation Distribution Oscillator (ADO) is another oscillator technical indicator that follows the trend of an index. The ADO is computed using high, low and closing index levels of the index time series.

3.3.7 Commodity Chanel Index

The Commodity Chanel Index (CCI) is another oscillator introduced in 1980 by Donald Lambert. It is widely used to identify cyclical trends in various financial time series applications. It does this by measuring the variation of a securities price from its statistical mean (Kim, 2003). The CCI is computed by first determining the typical price, the simple moving average and the mean absolute deviation of the typical price. The CCI is normally scaled by an inverse factor to ensure that about 80% of the values fall within the bounds of -100 and +100.

All technical indicators were computed from the JSE All-Share index daily data. Table 3.3 presents the summary statistics of the ten technical indicators generated from the JSE All-Share index data using the equations presented in Table 3.2. Each of the ten technical indicators are associated with a minimum value, a maximum value, a mean and a standard deviation. These statistical values of each technical indicator provide technical analysts insight into the technical characteristics of each technical indicator as well as that of the financial time series under study.

Table 3.3: Summary statistics of the technical indicators for the All-Share index.

No.	Technical Indicator	Min.	Max.	Mean	Std. Dev
1	SMA	7561,85	54538,24	28916,00	13719,58
2	WMA	7567,07	54631,17	28934,46	13722,84
3	MOM	-4870,80	4252,02	110,83	954,58
4	STCK	0,00	100,00	58,79	31,56
5	STCD	1,88	99,58	58,78	27,66
6	MACD	-1439,19	1076,04	86,88	326,88
7	RSI	0,00	100,00	54,85	19,24
8	WILLR	-100,00	0,00	-41,21	31,56
9	ADO	-6,21	100,00	53,54	31,74
10	CCI	-285,08	299,62	11,59	83,17

3.4 The Model Input Data

Technical indicators generated from the raw index data as described in Section 3.2 is pre-processed before being used as the input data to train and evaluate the prediction models viz. ensemble, SVM, logistic regression and k-nearest neighbour. Two methods of pre-processing the technical indicators are used in this study. The first method is the continuous representation of the input data and the second method is the trend deterministic representation of the input data. Since this study involves predicting the direction of the next days index level, a binary categorical response variable form part of the input data. The response variable assumes one of two values, up or down.

3.4.1 Continuous Input Representation

The continuous representation involves down-scaling of the technical indicators as in Kumar & Thenmozhi (2006), Kara et al. (2011) and Ballings et al. (2015). In this input representation method, the technical indicators are linearly normalised to the values in the range [-1; +1] and thereafter, used as inputs to the prediction models. This input method ensures that the higher magnitude technical indicators do not overpower smaller magnitude indicators within the various prediction algorithms. The max-min normalisation formula is presented in equation (3.1).

$$x' = \left(\frac{x - min(x)}{max(x) - min(x)} \times (Lim_U - Lim_L) \right) + Lim_L \qquad (3.1)$$

where

- x' is the linearly normalised technical indicator variable.
- x is the original, non-normalised technical indicator.
- $max(x)$ and $min(x)$ is the maximum and minimum values of the non-normalised technical indicators.
- Lim_L and Lim_U is the lower and upper limit respectively of the required normalised values i.e. [-1; +1].

Figure 3.1 illustrates the prediction model training process for a single training record using the continuous value representation of the ten linearly normalised technical indicators. The linear normalisation of each technical indicator is explained in Section 3.4.1 and computed using equation (3.1). Each of the linearly normalised technical indicators are computed using the index data of that particular day together with previous index data as described by the set of equations presented in Table 3.2. In Figure 3.1, all the technical indicators are computed using index data up to and including the 1st of September 2011. As this is the training process, a response variable must also be input to the model together with the continuous, linearly normalised technical indicators for that particular training record. The response variable will be computed using the next day's closing index value. As an example, the closing value of the index on the 2nd of September 2011 was 30518.92. Since this was lower than the closing index value on 1st of September 2011 (i.e. 31088.12), the training response variable will assume the value down and this will be used as the input in the training record for the 1st of September 2011.

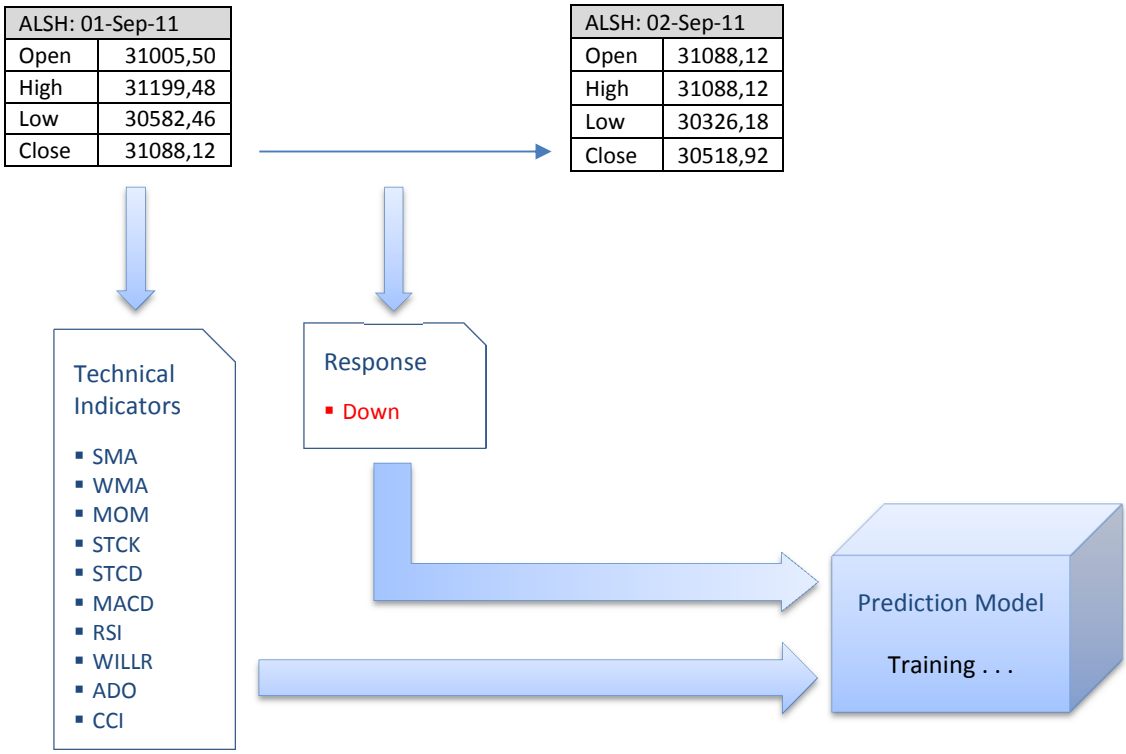

Figure 3.1: Illustration of the training process using continuous value technical indicators.

3.4.2 Trend Deterministic Input Representation

The trend deterministic input representation of the technical indicators involve discretising each of the ten technical indicators in order for them to represent a trend. In this form of input representation, the technical indicators assume either a value of +1, representing an upward trend for that technical indicator, or -1, representing a downward trend for the technical indicator. This input method is in contrast with the continuous input method where each of the technical indicator inputs can assume an infinite set of values in the range [-1; +1]. Patel et al. (2015) compared this method of input against the continuous variable input method. The study revealed that the trend deterministic method outperformed the continuous variable method when tested with market index data from India.

A 10 day moving average was used in the computation of SMA and WMA. If the current closing price is higher than the current value of the SMA, the trend deterministic value for SMA is +1. If the current closing price is lower than the current value of the SMA, the trend deterministic value for SMA is -1. The same logic applies in computing the trend deterministic values for WMA.

The three stochastic oscillators including STCK, STCD and WILLR are general trend indicators for financial time series data. When the value of these osillators at time t is greater than their value at time t–1, their trend deterministic value will be +1. If the value of the oscillator at time t is less than that of time t–1, then their trend deterministic value will be -1.

The MACD is a technical indicator that follows the trend of a stock. An increase in MACD is associated with an increase in the value of time series. Thus, if the value of MACD at time t is greater than its value at time t–1, its trend deterministic value will be +1. If the value of the MACD at time t is less than that of time t–1, then its trend deterministic value will be -1.

The RSI is generally used to identify overbought and oversold points. Values of RSI exceeding 70 indicate that stocks are overbought and as a result, prices are likely to decrease in future. In this case, the trend deterministic value for RSI will be -1. Values of RSI below 30 indicate that stocks are oversold and prices are likely to go up in future. In this case, the trend deterministic value for RSI will be +1. For values of RSI in the range [30; 70], if the value of RSI at time t is greater than its value at time t–1, its trend deterministic value will be +1 and vice-a-versa (Patel et al., 2015).

The CCI is also used to identify overbought and oversold levels. In this study, a CCI value exceeding +200 was used to indicate an overbought stock and is respresented by a trend deterministic value of -1. If the value of CCI is less than -200, the indication is that the stock is oversold and is respresented by a trend deterministic value of +1 (Patel et al., 2015).

The ADO is another oscillator that identifies trends. If its value at time t is greater than its value at time t–1, its trend deterministic value will be +1. If the value of ADO at time t is less than that of time t–1, then its trend deterministic value will be -1.

MOM is an indicator of the rate of rise and fall of stock prices. Trend deterministic values of MOM is determined from the sign of the MOM indicator only. If the value of MOM at time t is positive, then its trend deterministic value will be +1. If its value at time t is negative, then its trend deterministic value will be -1.

Figure 3.2 illustrates the model training process using trend deterministic technical indicators. The training process is similar to the training process for the continuous

value technical indicators. The difference here is that the previous day technical indicator and the current day technical indicator is used in determining the discrete trend indicator i.e. the trend indicator assuming a value of either -1 or +1. As an example, if the WMA on the 31st of August 2011 is 29984.79 and the WMA on the 1st of September 2011 is 30229.85, the WMA trend indicator for the 1st of September 2011 will then be +1 indicating an upward trend in WMA. If, however, the WMA on the 1st of September 2011 was less than the WMA on the 31st of August 2011, the WMA trend indicator for the 1st of September 2011 will then be -1 as this represents a downward trend in WMA for that particular trading day. The computations for the response variable is the same as for the response variable in the continuous input value training process where the closing index value of the following day was compared to that of the current day. Thus, for the trend deterministic indicator training model, both the technical indicators and the response variable are binary, i.e., each technical indicator assumes only one of two values, -1 indicating an downward trend and +1 indicating an upward trend. The response variable assumes only one of two values, up or down.

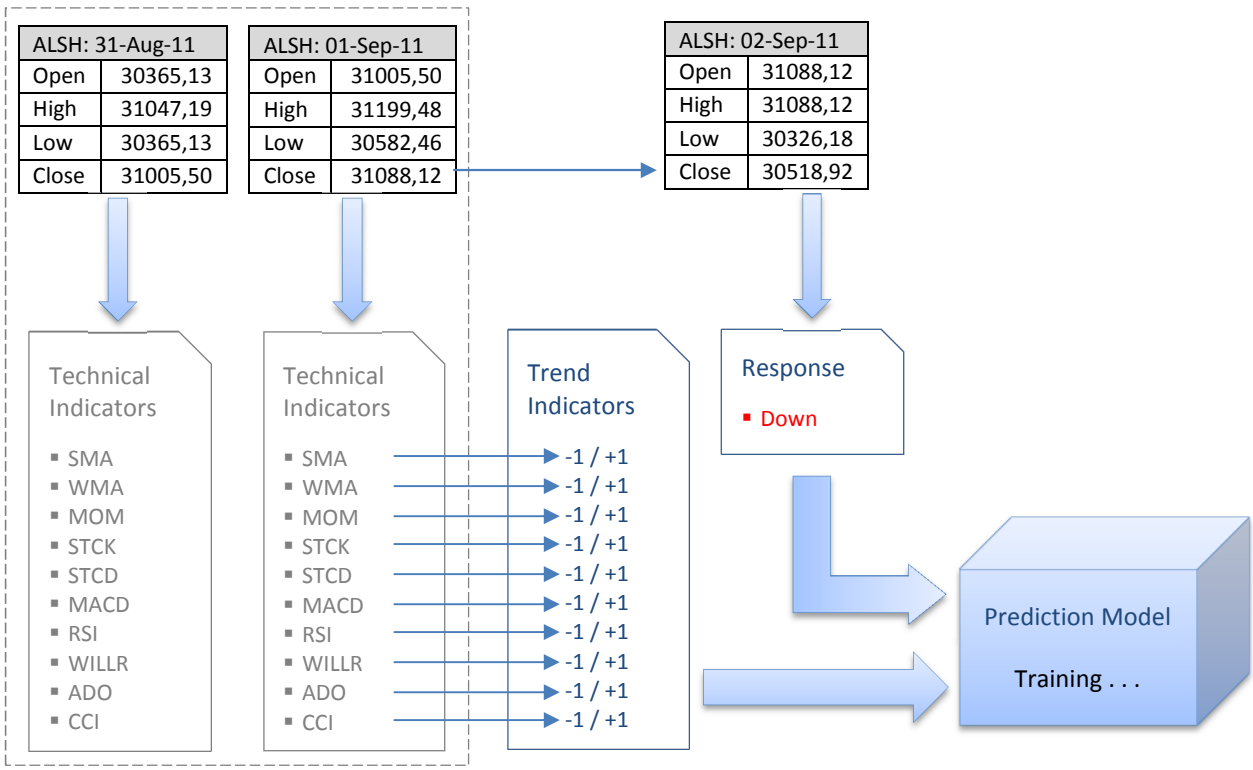

Figure 3.2: Illustration of the training process using trend deterministic technical indicators.

3.5 Training and Evaluation

The process of training and evaluating the prediction models both require model input data as described in Section 3.4. The training process uses both the technical indicators and the response variable as inputs while the evaluation process uses just the technical indicators as inputs to the trained model and thereafter compares the models predicted response to the actual response. The general training and evaluation process is illustrated in Figure 3.3 and Figure 3.4 respectively.

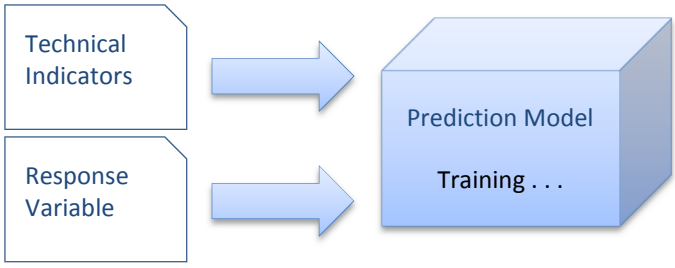

Figure 3.3: General prediction model training process.

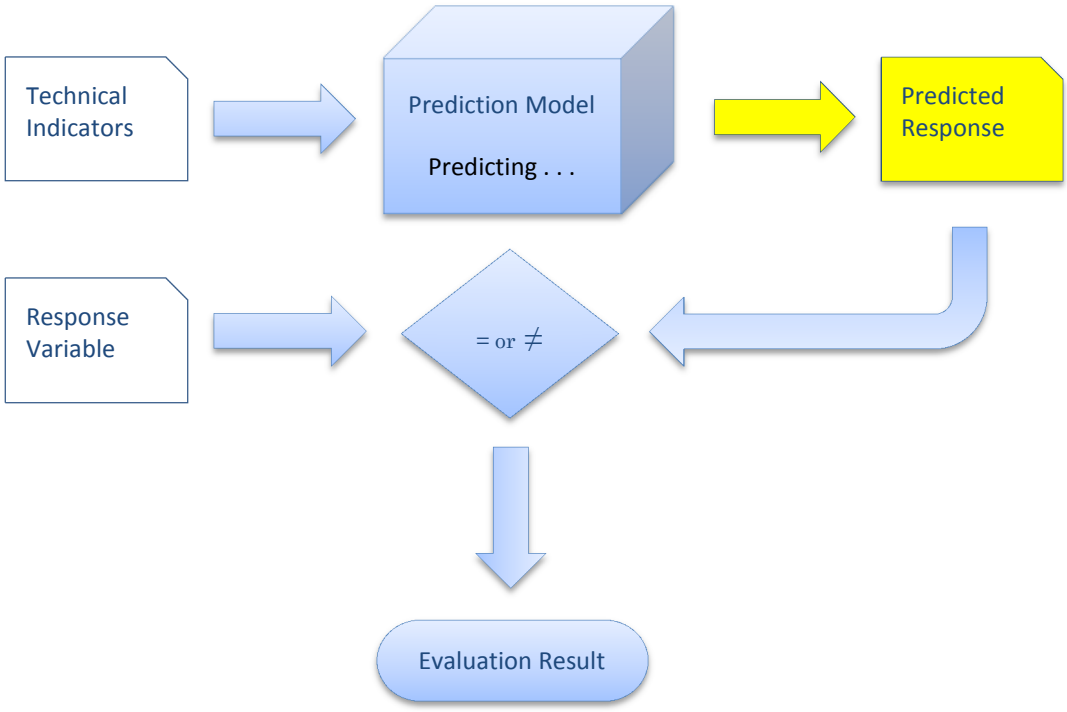

Figure 3.4: General prediction model evaluation process.

The training and evaluation datasets were each equal portions of the total data available. The total dataset as presented in Table 3.1 was used to create three separate training and evaluation datasets each with a different thresholds for the response variable: 0%, 0.5% and 1.0%. As an example, the 0.5% threshold dataset will have the response variable equal to up only if the increase in the next day index value exceeds 0.5%, else the response variable will be equal to down. Each dataset for the three thresholds were divided equally into training and evaluation portions for each year as presented in three tables in Appendix A. Figure 3.5 below graphs the number of up and down movements for each threshold dataset. Figure 3.5 also shows that the proportion of down movements increases as the response threshold increases for the fixed total dataset size of 3488. This is due to the index value increasing 5.8 fold in the timespan of the dataset and thus a higher threshold level would result in more responses being regarded as down instead of up.

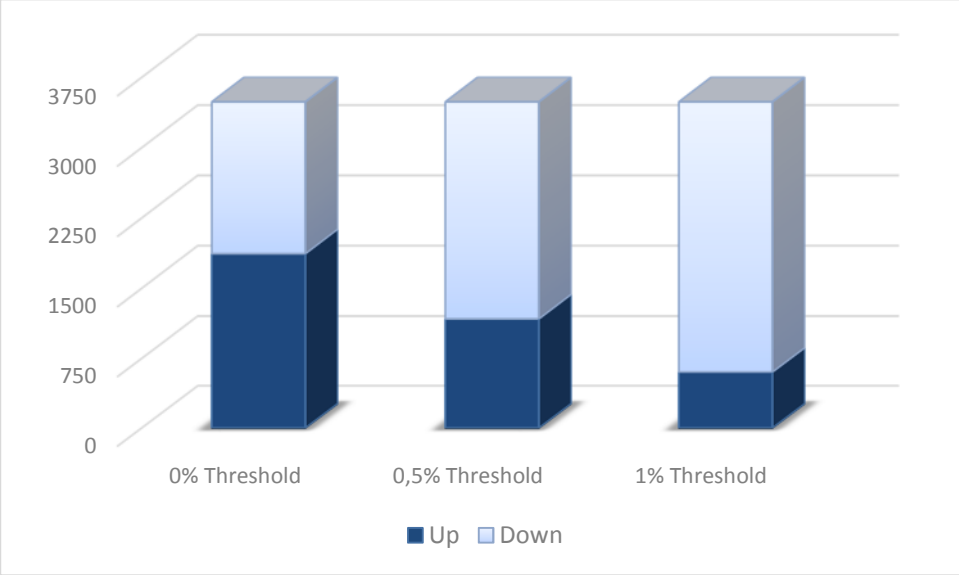

Figure 3.5: Response variable ratio for the three levels of threshold.

The area under the receiver operating characteristic (ROC) curve is seen as an adequate and accurate measure of prediction model performance as used in Ballings et al. (2015), Patel et al. (2015), Kara et al. (2011) and Rodriguez & Rodriguez (2004). The ROC assumes values in the range of [0.5; 1.0] where 0.5 indicates that the prediction is no better than random and a value of 1.0 indicates a perfect predictor. The ROC is computed using equation (3.2).

$$ROC = \int_0^1 \frac{TP}{TP + FN} d\left(\frac{FP}{FP + TN}\right) = \int_0^1 \frac{TP}{P} d\left(\frac{FP}{N}\right) \quad (3.2)$$

where:

- TP is the true positive (i.e. up) rate.
- TN is the true negative (i.e. down) rate.
- FP is the false positive rate.
- FN is the false negative rate.
- P is the number of positive events.
- N is the number of negative events.

Another commonly used measure for prediction performance is the percent correctly classified (PCC) as in studies by Kumar & Thenmozhi (2006), Kim (2003) and Manojlovic & Stajduhar (2015). The PCC is computed using equation (3.3).

$$PCC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

In this study, both the ROC and the PCC are computed but the ROC will be used as the performance evaluator and to rank the predictor models. Ballings & van den Poel (2013) stated than an advantage of the ROC over the PCC is that ROC includes all cut-off values in the computations of accuracy.

3.6 The Ensemble Prediction Model Performance

Ensemble prediction models are decision tree based models that are able to predict outcomes by averaging the outcomes from multiple decision trees. Problems that are statistical, computational and representational in nature can be readily solved by ensemble prediction models (Dietterich, 2000). The rationale behind ensemble prediction models is that a single decision tree alone is insufficient in accurately predicting an outcome based on a subset of available data. As only a subset of the data is used to train a single decision tree, it may not be able to distinguish between

noise and a definite pattern in the data. Hence, the majority decision from n decision trees is considered as the final output of the ensemble prediction model.

In the financial index trend prediction application, each node of a decision tree in the ensemble is split with a technical indicator from a random sample of the ten available technical indicators. The choice of technical indicator performing the actual split is the one that generates the highest information gain, i.e., the one that separates the up and down movements in the most effective manner. Each tree alone is trained from a random sample with replacement of the available data. The general training process for an ensemble training process is shown in Figure 3.6 below. There are many variation algorithms for ensemble predictions models each differing in the way data is selected and used to train the decision trees within the collection. Five ensemble algorithms are evaluated in this study namely: Boosted, RUS-Boosted, Sub-Space Discriminant, Bagged Trees and Sub-Space KNN. Details of these ensemble prediction algorithms remain beyond the scope of this study, however, differences in the prediction speed, memory usage, interpretability and model flexibility is presented in detail in Table B.1 of Appendix B.

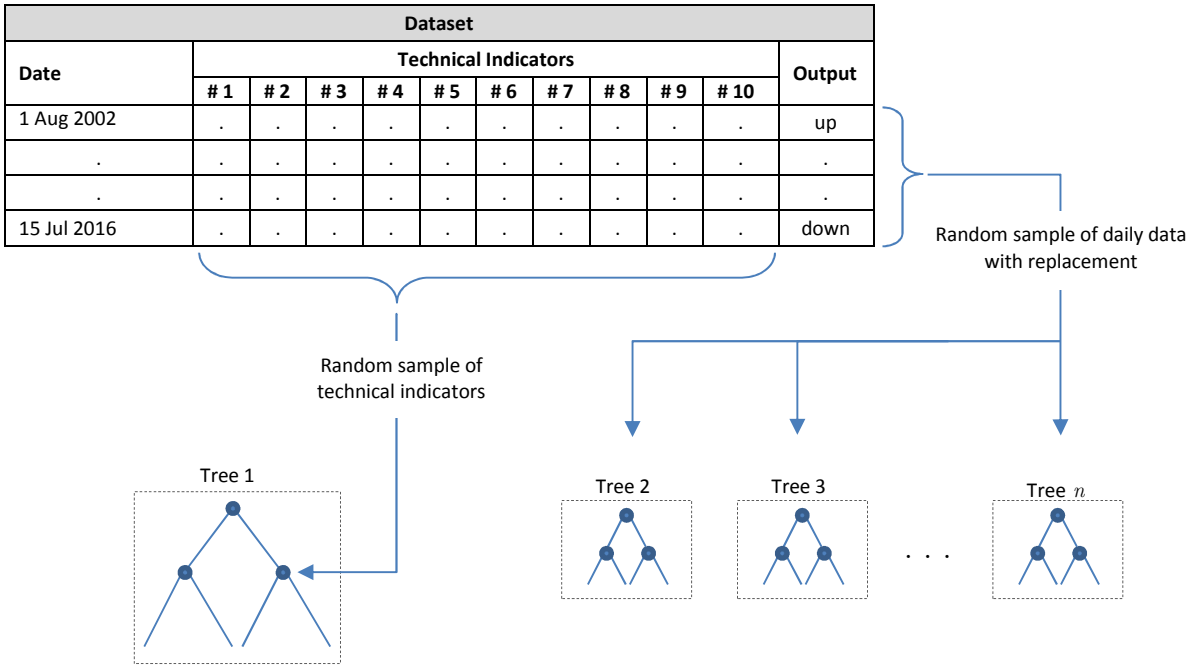

Figure 3.6: General training process for ensemble prediction models.

3.7 Prediction Performance of Other Models

This study primarily aims to evaluate the efficiency of ensemble prediction models in predicting the daily trend of the JSE All-Share index compared to other three most common prediction models in recent literature. The same training and evaluation datasets are used for all four prediction models to ensure an unbiased evaluation and comparison. The three other prediction models used in the comparison are SVM, logistic regression and KNN. An array of algorithms for each prediction model are evaluated in order to provide a comprehensive evaluation of all models in predicting the JSE All-Share index. The details of the algorithms for each of the four prediction models are presented in Appendix B. These details are in terms of prediction speed, memory usage, model interpretability and model flexibility.

3.7.1 SVM Prediction Models

The SVM prediction model was first developed by Vapnik (1999). It comprises of two categories: support vector classification (SVC) and support vector regression (SVR). The SVM model is characterised by high dimensional variable space and points are either classified as one of two disjoint half spaces or a higher dimensional feature space. The primary objective of the SVM algorithm is to establish a hyperplane that separates the data and maximises the margin of the hyperplane. The best hyperplane would therefore be one with the largest margin between the two classes of data. Support vectors are the points of data that lie closest to the hyperplane that separates the data. In the index trend scenario, the separating hyperplane would separate the up and down movements in two dimensions via any two of the ten available technical indicators.

Figure 3.7 below illustrates the two dimension case where two technical indicators are able to linearly separate the two classes of output, namely, the up and down daily trends. A separating margin is created from the datapoints of a class that is closest to the separating hyperplane. The support vectors are generated by the datapoints that lie on the margin at a variable distance away from the separating hyperplane.

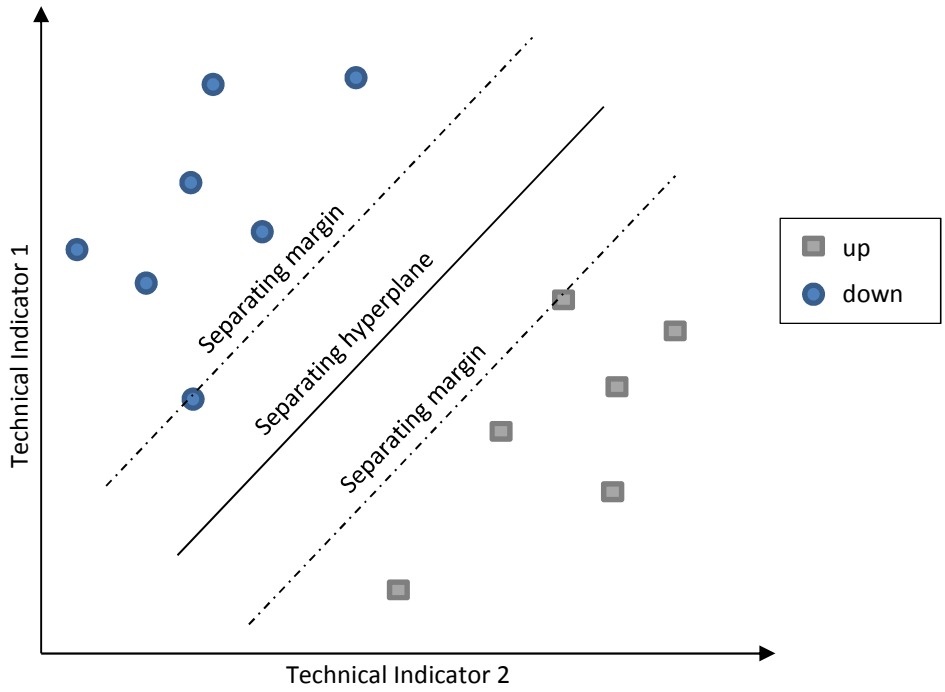

Figure 3.7: SVM classification principle using the separating hyperplane.

A kernel function performs the mapping of the feature space in the SVM prediction model. In the index trend prediction application, a ten dimension feature space exists. A regularization parameter of the SVM algorithm accounts for misclassifications resulting from the trade-off arising between the separating margin and misclassification of the classes. The kernel function can be set in order to generate variations in the SVM models. Variation in the kernel functions result in linear, cubic, quadratic and gaussian variations of the SVM algorithm. The gaussian algorithm contains three sub-variations namely: fine gaussian, medium gaussian and coarse gaussian. Each of these variations differ in the way distinctions between classes are made within the algorithm.

In the two class prediction application, the set of input vectors are represented as $x_i \in \mathrm{R}^{\mathrm{d}}$ where $i = 1,2,\ldots,N$ with the corresponding class labels represented by $y_i \in \{\text{up}, \text{down}\}$. The SVM attempts to generate a decision function that would result in a binary classifier from the available sample data. The SVM maps the input vectors into a high dimensional feature space $\Phi(x_i) \in H$ thus creating an optimal seperating hyperplane that maximises the separating margin of the two classes within the feature space H (Kara, Boyacioglu, & Baykan, 2011; Patel et al., 2015).

The mapping performed by the kernel function $K(x_i,x_j)$ results in a classifier decision boundary as described in equation (3.4). Quadratic programming is then used to solve for the coefficients α_i subject to the conditions given by equation (3.5) and equation (3.6) (Kara, Boyacioglu, & Baykan, 2011).

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i \cdot K(x, x_i) + b\right) \tag{3.4}$$

$$Maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \cdot y_i y_j \cdot K(x_i, x_j) \tag{3.5}$$

where $0 \leq \alpha_i \leq c$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.6}$$

The regularization parameter is defined by the constant c. The degree of kernel function is given by d (in the case of polynomial kernel function) and γ (in the case of radial basis kernel function). The choice of kernel function directly affects prediction quality. The polynomial kernel function is described by equation (3.7) and the radial basis kernel function is described by equation (3.8).

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{3.7}$$

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{3.8}$$

3.7.2 KNN Prediction Models

The k-nearest neighbour classification involves categorising points based on their distance to neighbour points within a training dataset. This classification method proves effective in the domain of pattern recognition, data-mining and machine learning and is comparable with even the state-of-the-art methods while still requiring unassuming computations (Atkeson, Moore, & Schaal, 1997). The KNN algorithm is based on the closest training example feature space (Huang et al., 2008; Kelly & Davis, 1991). A single datapoint is classified to a class that is most common among its k nearest datapoints. During model training, the KNN algorithm stores both the feature vectors and the classification variables.

In the index trend prediction application, the feature vector is one consisting of the ten techncial indicators associated with the classification variable i.e. up or down. During the evaluation or classification phase, a new feature vector consisting of ten technical indicators is input to the prediction model. Distances from the new vector and the existing stored vectors is calculated and the k closest samples are then selected. The new vector is then classified according to the most frequent class within the vector set (Huang et al., 2008).

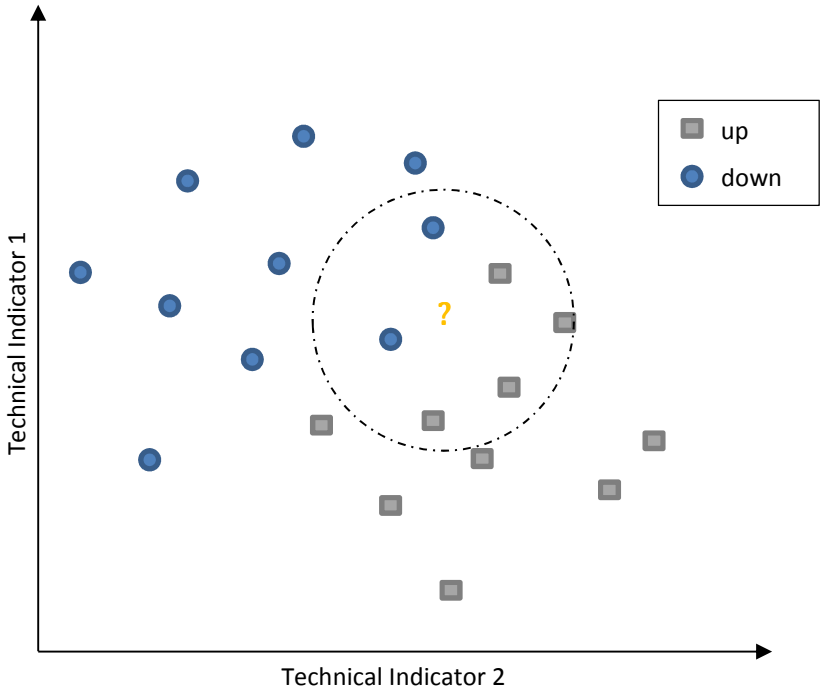

Figure 3.8: KNN classification principle using two technical indicators.

Figure 3.8 shows an example of the KNN prediction principle for the market index trend application. A two dimensional representation of the training data is illustrated via two technical indicators, similar to that of the SVM classification model in Figure 3.7. When a prediction is required, a new point within the ten dimensional technical indicator space is inserted. This new point can also be reflected in any set of the two dimensional technical indicator space as illustrated by an orange ? in Figure 3.8. Using a circular distance measure with the unknown point at the centre of the circle, the nearest six points is found with four of them belonging to the up class and two of them belonging to the down class. The unknown point is then classified into the up class by the prediction algorithm.

Various metrics are used to define the number of neighbours allowed within the prediction algorithm. The number of neighbours can be chosen via fixed KNN algorithms. The fine, medium and coarse KNN algorithms sets the number of neighbours to 1, 10 and 100 respectively. The distance metric can also be varied using the cosine, cubic and weighted distance metrics of the KNN algorithms. The distance between points is generally determined by a Euclidean parameter which defines the dissimilarity or distance $d(i,j)$ between a point i and point j where $d(i,j)$ is defined by equation (3.9), R_q is the range of any classified point q and p is the unclassified point.

$$d(i,j) = \sqrt{\left(\frac{|x_{i1-j1}|}{R_1}\right)^2 + \left(\frac{|x_{i2-j2}|}{R_2}\right)^2 + \ldots + \left(\frac{|x_{ip-jp}|}{R_p}\right)^2} \tag{3.9}$$

A method is then selected that combines classifications generated from k nearest neighbours in classifying the point p. The most common method is the voting method where the class given to point p is the majority class within the defined neighbours. Another method that eliminates the effect of unequal class points is one that averages the distances between each class point. As an example, an unclassified point p is classified to class C_1 (in a binary classification problem with classes C_1 and C_2), if the condition in equation (3.10) is satisfied.

$$\frac{1}{k_1}\sum_{i \in C_1(p,k)} d(i,p) \quad < \quad \frac{1}{k_2}\sum_{i \in C_2(p,k)} d(i,p) \tag{3.10}$$

k_1 is the total points belonging to class C_1 and k_2 is the total points belonging to class C_2 and $k = k_1 + k_2$. A third method is one that compares the total number of class points of each class within the k nearest neighbours. Similarly, in a binary classification example with classes C_1 and C_2, the unknown point p will be classified into class C_1 if equation (3.11) is satisfied.

$$\sum_{i \in C_1(p,k)} d(i,p) \quad < \quad \sum_{i \in C_2(p,k)} d(i,p) \tag{3.11}$$

In instances where the classes are highly asymmetric, the class with the higher number of data points will naturally be favoured. In such a case, k_1 and k_2 can e user defined parameter in the k nearest neighbour algorithm (Huang et al., 2008).

3.7.3 LR Prediction Models

LR prediction models are statistical regression models which use binary dependant variables as the output. In the index trend prediction application, the dependant variable is the trend that assumes the value up or down. LR models applies the maximum likelihood estimation after the dependant variable is transformed into a logit variable (Ou & Wang, 2009). In this way, the LR prediction model estimates the probability of occurrence of the possible events. In the index trend prediction application, the goal of the LR model is to predict the following day trend into one of two classes. The regression output for the LR prediction model can be computed using equation (3.12) where Y represents the output of the prediction model, TI represents each of the ten technical indicators and the β is the regression coefficients.

$$Y = \beta_0 + \beta_1 TI_1 + \beta_2 TI_2 + \ldots + \beta_{10} TI_{10} \tag{3.12}$$

A logistic response function is thereafter used to convert Y into a probability value. The probability function is computed using equation (3.13). An output classification can be made by simply providing a cut-off probability. The ten technical indicators serve as the independent variables while the prediction model output serve as the dependant variable that is binary assuming one of two values, up or down.

$$P = \frac{\exp(\beta_0 + \beta_1 TI_1 + \ldots + \beta_{10} TI_{10})}{1 + \exp(\beta_0 + \beta_1 TI_1 + \ldots + \beta_{10} TI_{10})} \tag{3.13}$$

Chapter Summary

This chapter presented the methodologies used in the evaluation of the ensemble prediction models as well as the three comparative prediction models. A detailed description of the JSE All-Share index data was first presented. The index data was described in terms of the daily movement on a yearly basis. This was followed by the derivation and description of the ten technical indicators. These technical indicators were pre-processed to form two representations of the prediction model input data, i.e. the continuous and the trend deterministic representation. These two forms of representation of the technical indicators were then used to train and evaluate each prediction model using three threshold levels for the response variable. Chapter 4 presents the evaluation results of the ensemble prediction model together with the results from the three comparative models, i.e. SVM, k-nearest neighbour and logistic regression.

Chapter 4

Presentation of Results

4.1 Introduction

This chapter presents the results of the performance of ensemble prediction models in predicting the JSE All-Share index daily direction. The performance of the ensemble prediction model is then compared to the performance of three other commonly used prediction models used in financial time series prediction. The chapter is structured as follows: Section 4.2 presents the performance of the ensemble prediction models in predicting the daily direction of the JSE All-Share index. Section 4.3 presents the performance of the three comparative prediction models, i.e. support vector machine, k-nearest neighbour and logistic regression. Section 4.4 presents the overall evaluation results where each of the four prediction models is represented by its best performing algorithm.

4.2 Ensemble Prediction Model Performance

The performance results of all five ensemble prediction model algorithms is presented in Table 4.1. The results are presented in terms of PCC and ROC and presented for each of the six possible data input method combinations i.e. continuous and trend deterministic inputs each based on 0%, 0.5% and 1.0% thresholds of the trend response variable.

The Boosted and RUS-Boosted ensemble algorithms both have the highest ROC value of 0.65 with a PCC of 82.80% for the Boosted algorithm and a PCC of 60.10% for the RUS-Boosted algorithm. These results are achieved with the continuous data input method and a 1.0% threshold of the response variable. Thus, the Boosted ensemble algorithm managed to correctly predict 82.80% of the trends in the evaluation dataset while the RUS-Boosted algorithm only managed to correctly predict 60.10% of the trends in the evaluation dataset. From the evaluation results provided in Table 4.1, it can be seen that as the threshold for the response variable increases from 0% to 1.0%, the ROC also increases for all five ensemble algorithms using both trend and continuous input data.

Figure 4.1 illustrates a bar graph of the ROC values from all five ensemble algorithms using the continuous input data method and 1.0% threshold in the response variable. This plot was chosen as it represents the best performance from all six input methods.

Table 4.1: Evaluation results for the ensemble prediction model algorithms.

Input Method	Threshold	Boosted		Bagged		Subspace Disc		Subspace KNN		RUS-Boosted	
		PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC
Continuous	0,00%	52,20%	0,50	53,30%	0,52	53,10%	0,52	49,30%	0,50	50,20%	0,50
Continuous	0,50%	65,30%	0,55	63,00%	0,53	66,40%	0,57	61,80%	0,53	50,10%	0,54
Continuous	1,00%	82,80%	0,65	81,80%	0,61	82,80%	0,62	80,80%	0,58	60,10%	0,65
Trend	0,00%	53,10%	0,51	53,20%	0,51	53,60%	0,51	50,10%	0,52	51,00%	0,52
Trend	0,50%	63,70%	0,51	62,70%	0,51	66,40%	0,53	51,20%	0,50	52,60%	0,51
Trend	1,00%	82,60%	0,57	82,40%	0,53	82,80%	0,58	76,90%	0,55	62,90%	0,55

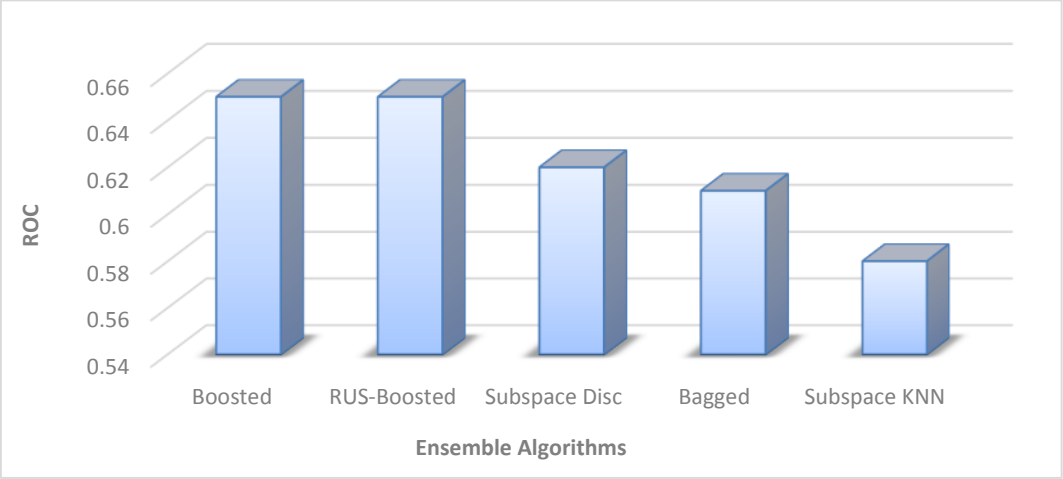

Figure 4.1: Ensemble model performance results for each ensemble algorithm.

4.3 Performance of Comparative Models

The detailed evaluation results for the three comparative prediction models, i.e., SVM, k-nearest neighbour and logistic regression is discussed below. The presentation of the results is similar to that of the ensemble prediction model where each models algorithm performance is represented by its ROC and PCC values.

4.3.1 SVM Model Performance

Six SVM algorithms are trained and evaluated using same dataset used in the training and evaluation of the ensemble prediction model algorithms. The six SVM algorithms used in the evaluation are the linear, quadratic, cubic, fine gauss, medium gauss and the coarse gauss. Table 4.2 presents the detailed evaluation results for all six SVM algorithms. The results are presented for all six possible data input method combinations, i.e., continuous and trend deterministic inputs each based on 0%, 0.5% and 1.0% thresholds of the trend response variable.

Details of each SVM prediction algorithm in terms of prediction speed, memory usage, interpretability and model flexibility is presented in Table B.2 of Appendix B. The best performing SVM algorithm is the fine gauss with an ROC of 0.59 and a PCC of 82.60%. This result is achieved with the continuous data input method and a 1.0% threshold of the trend response variable. From the evaluation results in Table 4.2, it can be seen that each of the SVM algorithms performance, in terms of ROC, does not necessarily increase as the threshold of the response variable increases from 0% to 1.0%. This is in contrast to the ensemble algorithms where the performance of all prediction algorithms increased as the response threshold increased. This best performing SVM algorithm, i.e. fine gauss with an ROC of 0.59, is only marginally better than the worst performing ensemble algorithm, i.e. Subspace KNN with an ROC of 0.58.

Figure 4.2 illustrates a bar graph of the ROC values from all six SVM algorithms using the continuous input data method and 1.0% threshold in the response variable. Similar to the ensemble algorithms, this was chosen as it represents the best performance from all six input methods.

Table 4.2: Evaluation results for the SVM prediction model algorithms.

Input Method	Threshold	Linear		Quadratic		Cubic		Fine Gauss		Med. Gauss		Coarse Gauss	
		PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC
Continuous	0,00%	53,60%	0,52	53,00%	0,52	54,00%	0,52	52,10%	0,51	53,60%	0,53	53,60%	0,51
Continuous	0,50%	66,40%	0,54	66,40%	0,53	64,60%	0,51	64,20%	0,54	66,40%	0,52	66,40%	0,50
Continuous	1,00%	82,80%	0,49	82,80%	0,55	82,70%	0,55	82,60%	0,59	82,80%	0,56	82,80%	0,58
Trend	0,00%	53,60%	0,49	52,20%	0,51	50,80%	0,51	51,60%	0,50	52,00%	0,50	53,60%	0,49
Trend	0,50%	66,40%	0,53	66,40%	0,49	64,90%	0,52	65,60%	0,50	66,10%	0,51	66,40%	0,49
Trend	1,00%	82,80%	0,46	82,80%	0,51	82,10%	0,50	82,50%	0,52	82,80%	0,51	82,80%	0,48

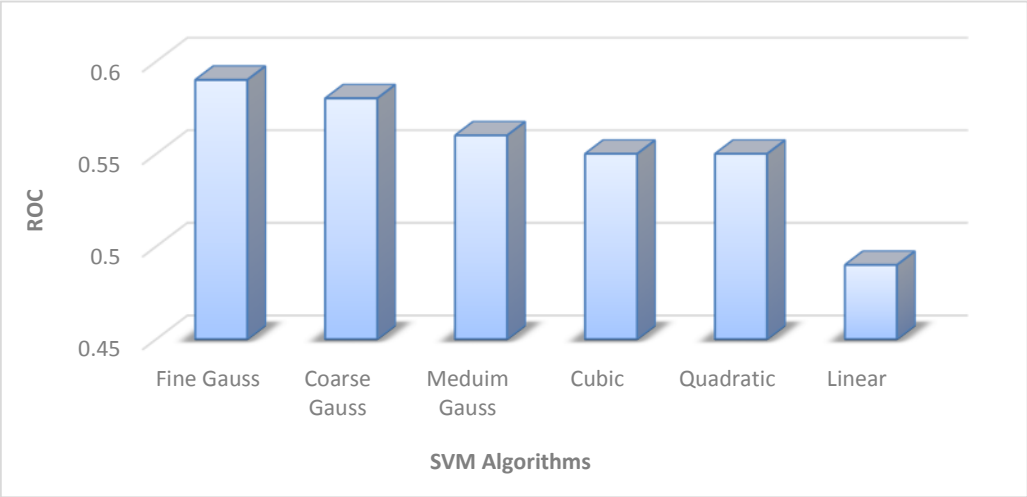

Figure 4.2: SVM model performance results for each SVM algorithm.

4.3.2 KNN Model Performance

Six KNN algorithms are also trained and evaluated using same dataset used in the training and evaluation of the ensemble and SVM prediction model algorithms. The six KNN algorithms used in the evaluation are the fine, medium, coarse, cosine, cubic and the weighted. Table 4.3 presents the detailed evaluation results for all six KNN algorithms. The results are presented for all six possible data input methods as for the ensemble and SVM algorithms. Details of each KNN prediction algorithm in terms of prediction speed, memory usage, interpretability and model flexibility is also presented in Table B.3 of Appendix B. The best performing KNN algorithm is the coarse algorithm with an ROC of 0.62 and a PCC of 82.80%. This best result is also achieved with the continuous data input method and a 1.0% threshold of the trend response variable, similar to that of the ensemble and SVM models.

From the evaluation results in Table 4.3, it can be seen that each of the KNN algorithms performance, in terms of ROC, increase as the threshold of the response variable increases from 0% to 1.0%. This increase in performance however, is only applicable to the continuous data input method. For the trend deterministic input method, an increase in the threshold of the response variable does not necessarily translate to an increase in model performance. As an example, in the fine KNN algorithm, the ROC for the trend deterministic input at a 0% response variable threshold is 0.52. At a 0.50% response variable threshold using trend deterministic input, the ROC decreased to 0.49. This result is similar to that of the SVM algorithms and in contrast to the ensemble algorithms where the performance increased as the response variable threshold increased. The exception is in the case of the coarse KNN algorithm where the ROC did not decrease as the response variable threshold increased from 0% to 1.0% with trend deterministic inputs. The ROC for the coarse KNN algorithm actually increased from 0.52 to 0.53 as the response threshold increased from 0% to 0.50% and remained at 0.53 for the response threshold increasing to 1.0%.

Figure 4.3 illustrates a bar graph of the ROC values from all six KNN algorithms using the continuous input data method and 1.0% threshold in the trend response. Similar to the ensemble and SVM algorithms, this was chosen as it results in the best performance from all six data input methods.

Table 4.3: Evaluation results for the KNN prediction model algorithms.

Input Method	Threshold	Fine		Medium		Coarse		Cosine		Cubic		Weighted	
		PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC	PCC	ROC
Continuous	0,00%	50,00%	0,50	50,40%	0,50	51,80%	0,50	49,00%	0,49	49,20%	0,50	51,20%	0,50
Continuous	0,50%	56,80%	0,51	64,10%	0,55	66,60%	0,57	64,20%	0,55	64,20%	0,55	63,40%	0,55
Continuous	1,00%	73,20%	0,52	82,30%	0,60	82,80%	0,62	82,60%	0,58	82,20%	0,60	81,00%	0,60
Trend	0,00%	51,60%	0,52	48,20%	0,51	51,60%	0,52	48,10%	0,51	48,20%	0,51	47,80%	0,50
Trend	0,50%	50,80%	0,49	63,60%	0,51	66,40%	0,53	63,50%	0,51	63,60%	0,51	63,50%	0,51
Trend	1,00%	72,20%	0,52	82,40%	0,50	82,80%	0,53	82,40%	0,50	82,40%	0,50	81,70%	0,50

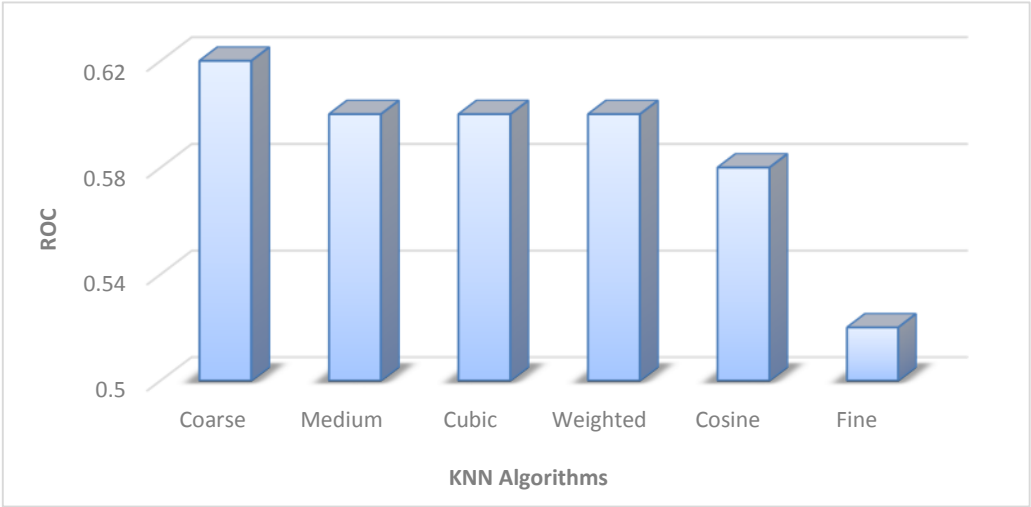

Figure 4.3: KNN model performance results for each KNN algorithm.

4.3.3 LR Model Performance

Since the LR model does not contain any algorithm variants or parameters to adjust the model, only a single set of results in terms of PCC and ROC is computed for each of the six forms of input data as presented in Table 4.4 below. The best performance of the LR prediction model is also achieved with the continuous input data representation and 1.0% threshold on the response variable. This yields an ROC of 0.61 and a PCC of 82.80%. It is also evident from Table 4.4, that the LR prediction model performance increases as the response threshold increases when using both the continuous and trend deterministic input data representation. This behaviour is similar to that of the ensemble prediction model algorithms. Figure 4.4 illustrates a bar graph of the ROC values for all six data input methods of the LR prediction model.

Table 4.4: Evaluation results for the LR prediction model.

Input Method	Threshold	LR	
		PCC	ROC
Continuous	0,00%	53,80%	0,51
Continuous	0,50%	66,40%	0,55
Continuous	1,00%	82,80%	0,61
Trend	0,00%	51,80%	0,51
Trend	0,50%	66,40%	0,53
Trend	1,00%	82,80%	0,58

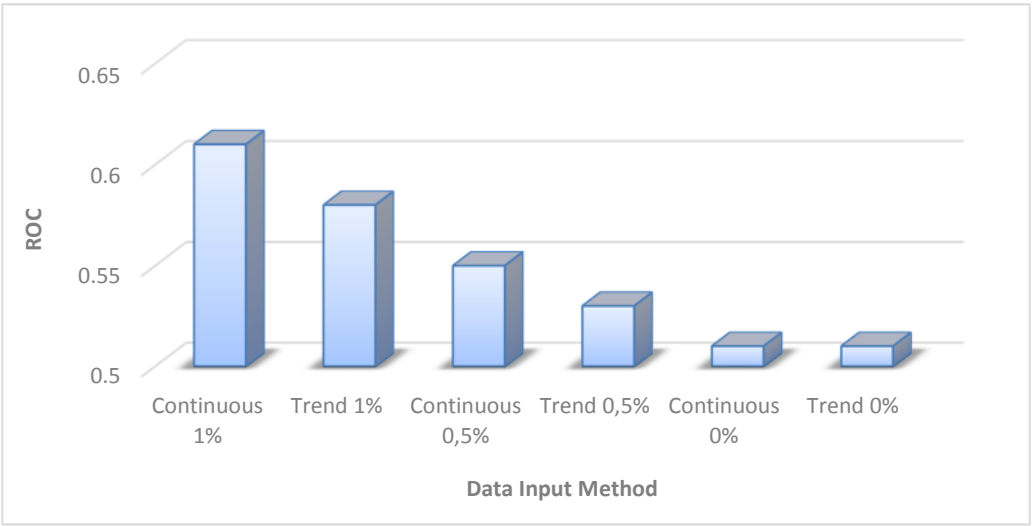

Figure 4.4: LR model performance results for the six data input methods.

4.4 Overall Model Evaluation

Figure 4.5 presents the overall evaluation result where each prediction model is represented by it best performing algorithm. The ROC for each model is based on continuous input technical data with 1.0% threshold of the response variable as this input and response combination resulted in the best overall performance in all prediction models. Figure 4.5 shows that the ensemble prediction model is the best performer of the four models, followed closely by KNN, LR and SVM. The ensemble prediction model is represented by the Boosted tree algorithm with an ROC of 0.65. The KNN model is represented by the coarse boundary algorithm with and ROC of 0.62. The LR prediction model yields and ROC of 0.61 and the SVM prediction model represented by the fine gauss algorithm yields the lowest ROC of 0.59.

Figure 4.5: Overall prediction model performance.

Chapter Summary

This chapter presented the detailed evaluation results of all algorithms of each of the four prediction models. The evaluation results were presented in terms of PCC and ROC. Each prediction algorithm was evaluated with six data input methods thus providing a comprehensive evaluation of each prediction model. The ensemble prediction model represented by the boosted algorithm was found to be the best performer from the four models in predicting the JSE All-Share daily direction.

Chapter 5

Discussion and Conclusion

5.1 Introduction

This chapter discusses the results obtained in Chapter 4. The chapter is organised as follows. Section 5.2 discusses the prediction model evaluation results. Section 5.3 concludes the study and make suggestions for future work.

5.2 Discussion

This research examines the predictability of the JSE All-Share index daily trend using ensemble prediction models. The prediction performance of ensemble models is then compared to that of commonly used models in the domain of stock index and stock price prediction, namely SVM, LR and KNN. Index data ranging from August 2002 to July 2016 is used in model construction and evaluation. Multiple variations of each prediction model in the form of prediction algorithms is used in the investigation to ensure a comprehensive, unbiased model comparison and evaluation. Ten independent technical indicators reflecting the JSE All-Share index characteristics are used as predictors for each model. The technical indicators are used in the linearly normalised continuous form and the trend deterministic form, while the index response variable is computed according to three threshold levels namely, 0%, 0.5% and 1.0%.

The results show that the Boosted algorithm of the ensemble prediction model performed the best out of the four models in the study in predicting the daily direction of the JSE All-Share index with an ROC of 0.65 and a PCC of 82.80%. This result is congruent to that achieved by Ballings et al. (2015) and Patel et al., (2015) where ensemble prediction models also outperformed the KNN, SVM and LR prediction models. It is also found that the linearly normalised continuous valued technical indicator inputs results in better prediction performance than the trend deterministic technical indicator inputs. This result is consistent in all four prediction models. This however, is in contrast to the results achieved by Patel et al., (2015) where the authors found that the trend deterministic data inputs resulted in better prediction performance compared to the continuous data inputs in all the prediction models investigated. The study also reveals that the Boosted ensemble algorithms performance increased by 14% when continuous valued inputs were used in place of trend deterministic inputs while using a 1.0% threshold in the response variable. The best performing KNN, LR and SVM algorithms resulted in performance increases of

16.9%, 5.17% and 13.4% respectively when continuous valued inputs were used instead of trend deterministic inputs while using a 1.0% response threshold.

The KNN prediction model ranked second, the LR prediction model ranked third and the SVM prediction model is found to perform the worst among the four models. This result is in contrast to that achieved by Kumar & Thenmozhi (2006) where SVM performed the best among ensemble, LR and neural networks in predicting the trend of the S&P CNX NIFTY market index.

5.3 Conclusion and Future Research

Investors aim is always to profit from the stock market. However, various studies have indicated that this is impossible as, in terms of the EMH, the price of the stock is always valued. However, the findings in this study shows that investors can beat the market if they incorporate algorithms in their analysis to predict the direction of a stock or index with a prediction performance better than that of random. Ensemble prediction model outperforms the commonly used SVM, LR and KNN prediction models in predicting the daily trend of the JSE All-Share index. It is therefore strongly recommended that ensemble prediction models be included in the field of prediction and technical analysis on various financial time series such as market indices, stock prices and trends, exchange rates, etc.

The other three prediction models evaluated in this study should definitely not be excluded as their performance in this study is not significantly worse than that of the ensemble prediction models. It is also recommended that multiple prediction models be considered and evaluated in the field of financial time series prediction as each markets time series has its own unique technical and statistical characteristic and no single model should be regarded as a superior performer. Therefore, every market must evaluate and identify the best performing prediction model for each time series within that market. Using this approach, the best prediction model for the financial time series under investigation can be identified and utilised as one of the many tools in the domain of trading and investments.

In this study, a one day ahead prediction on the trend is made on a financial time series using thresholds of one percent and below. Future studies in predicting one month or one year ahead trends using technical indicators representative of an

appropriate period would certainly add value to the literature. Economic indicators instead of purely technical indicators in the above prediction models should also be explored in predicting trends in financial time series.

References

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, 11-73.

Ballings, M., & van den Poel, D. (2013). Kernel Factory: An ensemble of kernel machines. *Expert Systems with Applications*, 2409-2413.

Ballings, M., Van Den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications*, 7046-7056.

Becchetti, L., Rocci, R., & Trovato, G. (2007). Industry and time specific deviations from fundamental values in a random coefficient model. *Annals of Finance*, 257-276.

Black, A., Fraser, P., & Groenewold, N. (2003). US stock prices and macroeconomic fundamentals. *International Review of Economics & Finance*, 345-367.

Booth, A., Gerding, E., & McGroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*, 3651-3661.

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Cervelló-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems with Applications*, 5963-5975.

Chen, Y. H., & Fraser, P. (2010). What drives stock prices? Fundamentals, bubbles and investor behaviour. *Applied Financial Economics*, 1461-1477.

Cheung, Y. W., Chinn, M. D., & Pascual, A. G. (2005). Empiracle exchange rate models of the nineties: Are any fit to survive? *Journal of International Money and Finance*, 1150-1175.

Coakley, J., & Fuertes, A. M. (2006). Valuation ratios and price deviations from fundamentals. *Journal of Banking & Finance*, 2325-2346.

da Costa, T. R., Nazário, R. T., Bergo, G. S., Sobreiro, V. A., & Kimura, H. (2015). Trading System based on the use of technical analysis: A computational experiment. *Journal of Behavioral and Experimental Finance*, 42-55.

de Oliveira, F. A., Nobre, C. N., & Zárate, L. E. (2013). Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index – Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications*, 7596-7606.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*, 1-15.

Fama, E. F. (1991). Efficient Capital Markets. *Journal of Finance*, 1575-1617.

Firer, C., Ross, S. A., Westerfield, R. W., & Jordan, B. D. (2012). *Fundamentals of Corporate Finance.* New York: McGraw-Hill.

Harvey, C. R. (1995). Predictable risk and returns in emerging markets. *The Review of Financial Studies*, 773-816.

Hellstrom, T., & Holmstromm, K. (1998). *Predictable Patterns in Stock Returns.* Technical Report Series IMa-TOM-1997-09.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications*, 212-222.

Huang, C.-J., Yang, D.-X., & Chuang, Y.-T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 2870-2878.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 2513-2522.

Imandoust, S. B., & Bolandraftar, M. (2014). Forecasting the direction of stock market index movement using three data mining techniques: the case of Tehran Stock Exchange. *Journal of Engineering Research and Applications*, 106-117.

Jordan, B. D., Miller, T. W., & Dolvin, S. D. (2015). *Fundamentals of Investments: Valuation and Management.* New York: McGraw-Hill.

Kao, L. J., Chiu, C. C., Lu, C. J., & Chang, C. H. (2013). A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting. *Decision Support Systems*, 1228-1244.

Kara, Y., Boyacioglu, M. A., & Baykan, O. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of Istanbul Stock Exchange. *Expert Systems with Applications*, 5311-5319.

Kelly, J. D., & Davis, L. (1991). Hybridizing the genetic algorithm and the k-nearest neighbors classification algorithm. *Proceedings of the fourth international conference on genetic algorithms & applications*, 377-383.

Kim, K.-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 307-319.

Kumar, M., & Thenmozhi, M. (2006). *Forecasting stock index movement: A comparison of support vector machines and random forest.* Rochester NY: SSRN Scholarly Paper.

Laopodis, N. T. (2011). Equity prices and macroeconomic fundamentals: International evidence. *Journal of International Financial Markets, Institutions & Money*, 247-276.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 10896-10904.

Liao, S. H., & Chou, S. Y. (2013). Data mining investigation of co-movements on the Taiwan and China stock markets for future investment portfolio. *Expert Systems with Applications*, 1542-554.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of Technical analysis: Computational algorithms, statistical inference and empirical implementation. *Journal of Finance*, 1705-1770.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *The Journal of Economic Perspectives*, 59-82.

Manojlovic, T., & Stajduhar, I. (2015). Predicting stock market trends using random forests: A sample of the Zagreb Stock Exchange. *MIPRO*, 25-29.

Manzan, S. (2007). Nonlinear mean reversion in stock prices. *Quantitative and Qualitative Analysis in Social Sciences*, 1-20.

Murphy, J. J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide To Trading Methods and Applications.* New York Institute of Finance.

Ou, P., & Wang, H. (2009). Prediction of Stock Market Index Movement by Ten Data Mining Techniques. *Modern Applied Science*, 28-42.

Pan, M. S. (2007). Permanent and transitory components of earnings, dividends, and stock prices. *The Quarterly Review of Economics and Finance*, 535-549.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 259-268.

Pesaran, H. M., & Timmermann, A. (1995). Predictability of stock returns: Robustness and economic signiificance. *The Journal of Finance*, 1201-1228.

Rodriguez, P. N., & Rodriguez, A. (2004). Predicting stock market indices movements. *Computational Finance and its Applications.*

Senol, D., & Ozturan, M. (2008). Stock Price Direction Prediction using Artificial Neural Network Approach: The Case of Turkey. *Journal of Artificial Intelligence*, 70-77.

Silva, A., Neves, R., & Horta, N. (2015). A hybrid approach to portfolio composition based on fundamental and technical indicators. *Expert Systems with Applications*, 2036-2048.

Subha, M. V., & Nambi, S. T. (2012). Classification of Stock Index movement using k-Nearest Neighbours (k-NN) algorithm. *Information Science and Applications*, 261-270.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEEE Transactions on Neural Networks*, 988-999.

Velinova, A., & Chen, W. (2015). Do stock prices reflect their fundamentals? New evidence in the aftermath of the financial crisis. *Journal of Economics and Business*, 1-20.

White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. *IEEE International on Neural Networks*, 451-458.

Yuhn, K. H., Kim, S. B., & Nam, J. H. (2015). Bubbles and the Weibull distribution: Was there an explosive bubble in US stockprices before the global economic crisis? *Applied Economics*, 255-271.

Zhu, Y., & Zhou, G. (2009). Technical analysis: An asset allocation perspective on the use of moving averages. *Journal of Financial Economics*, 519-544.

Appendix A

Table A.1: Training and evaluation dataset quantities for 0% threshold of the response variable.

Year	Training			Evaluation		
	Down	Up	Total	Down	Up	Total
2002	27	25	52	27	25	52
2003	62	64	126	61	63	124
2004	58	68	126	57	68	125
2005	53	73	126	52	73	125
2006	54	70	124	54	70	124
2007	54	71	125	54	71	125
2008	67	59	126	66	59	125
2009	57	68	125	57	68	125
2010	58	68	126	58	67	125
2011	61	64	125	60	64	124
2012	54	72	126	53	71	124
2013	57	68	125	57	68	125
2014	62	63	125	62	62	124
2015	61	65	126	60	65	125
2016	28	39	67	28	38	66
Total	813	937	1750	806	932	1738

Table A.2: Training and evaluation dataset quantities for 0.5% threshold of the response variable.

Year	Training			Evaluation		
	Down	Up	Total	Down	Up	Total
2002	36	17	53	35	16	51
2003	84	42	126	83	41	124
2004	87	39	126	87	38	125
2005	83	43	126	82	43	125
2006	73	51	124	73	51	124
2007	76	50	126	75	49	124
2008	79	47	126	79	46	125
2009	78	48	126	77	47	124
2010	88	38	126	87	38	125
2011	82	43	125	82	42	124
2012	89	37	126	88	36	124
2013	89	37	126	88	36	124
2014	92	33	125	91	33	124
2015	83	43	126	83	42	125
2016	43	24	67	42	24	66
Total	1162	592	1754	1152	582	1734

Table A.3: Training and evaluation dataset quantities for 1.0% threshold of the response variable.

Year	Training			Evaluation		
	Down	Up	Total	Down	Up	Total
2002	43	10	53	42	9	51
2003	101	25	126	100	24	124
2004	108	18	126	107	18	125
2005	107	19	126	106	19	125
2006	98	27	125	97	26	123
2007	99	27	126	98	26	124
2008	88	38	126	87	38	125
2009	94	31	125	94	31	125
2010	107	19	126	107	18	125
2011	102	23	125	102	22	124
2012	115	10	125	115	10	125
2013	107	19	126	106	18	124
2014	115	10	125	115	9	124
2015	109	17	126	109	16	125
2016	53	14	67	53	13	66
Total	1446	307	1753	1438	297	1735

Appendix B

Table B.1: Algorithms for ensemble prediction model.

Ensemble Algorithm	Prediction Speed	Memory Usage	Interpretability	Ensemble Method	Model Flexibility
Boosted Trees	Fast	Low	Hard	ADABoost with Decision Trees	Medium-High
Bagged Trees	Medium	High	Hard	Random Forest	High
Subspace Discriminate	Medium	Low	Hard	Subspace with Discriminate	Medium
Subspace KNN	Medium	Medium	Hard	Subspace with k-Nearest Neighbour	Medium
RUSBoosted Trees	Fast	Low	Hard	RUSBoost with Decision Tree	Medium

Table B.2: Algorithms for SVM prediction model.

SVM Algorithm	Prediction Speed	Memory Usage	Interpretability	Model Flexibility
Linear	Fast	Medium	Easy	Low
Quadratic	Fast	Medium	Hard	Medium
Cubic	Fast	Medium	Hard	Medium
Fine Gaussian	Fast	Medium	Hard	High
Medium Gaussian	Fast	Medium	Hard	Medium
Coarse Gaussian	Fast	Medium	Hard	Low

Table B.3: Algorithms for KNN prediction model.

KNN Algorithm	Prediction Speed	Memory Usage	Interpretability	Model Flexibility
Fine	Medium	Medium	Hard	Fine details between classes $n_{\text{neigh}} = 1$
Medium	Medium	Medium	Hard	Medium distinction between classes $n_{\text{neigh}} = 10$
Coarse	Medium	Medium	Hard	Coarse distinction between classes $n_{\text{neigh}} = 100$
Cosine	Medium	Medium	Hard	Medium distinction between classes Cosine distance metric $n_{\text{neigh}} = 10$
Cubic	Slow	Medium	Hard	Medium distinction between classes Cubic distance metric $n_{\text{neigh}} = 10$
Weighted	Medium	Medium	Hard	Medium distinction between classes Weighted distance metric $n_{\text{neigh}} = 10$

Table B.4: Algorithm for LR prediction model.

LR Algorithm	Prediction Speed	Memory Usage	Interpretability	Model Flexibility
Logistic Regression	Fast	Medium	Easy	Low (No control parameters)