COPYRIGHT AND CITATION CONSIDERATIONS FOR THIS THESIS/ DISSERTATION

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

- NonCommercial — You may not use the material for commercial purposes.

- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

How to cite this thesis

Forecasting the one day ahead magnitude and direction of the South African Volatility Index using a Multilayer Perceptron Artificial Neural Network

A minor dissertation submitted to the College of Business and Economics in partial fulfilment of the requirements for the degree of Master of Commerce in Finance by Vusumuzi Moyo

Supervisor: Mrs Indrani O’Leary-Govender
Co-supervisor: Miss Roxanne Andrew

Department of Finance and Investment Management University of Johannesburg
April 2018
Abstract

Forecasting stock market volatility is an important subject in investment and risk management. By improving upon this task, investors can make sounder investment choices in re-balancing their portfolios, leading to greater profits. Artificial Neural Networks (ANNs) have been used extensively in many fields. They have proved to be one of the most effective tools in time series forecasting. The aim of this study is to forecast the South African Volatility Index (SAVI) using a Multilayer Perceptron ANN. Several experiments are conducted where the ANN is trained to forecast (i) the next trading day’s SAVI return and (ii) the movement direction in the next trading day’s SAVI return. The results of the experiments show that the ANN performed better in forecasting the movement direction in the next period’s volatility as opposed to the next period’s expected volatility.
Key Words

Artificial Neural Networks, stock market, machine learning, forecasting, South African Volatility Index
Declaration of original work

I, Vusumuzi Moyo, declare that this minor dissertation is my own, unaided work. Any assistance that I received has been duly acknowledged in the dissertation. It is submitted in partial fulfilment of the requirements for the degree of Master of Commerce at the University of Johannesburg. It has not been submitted before for any degree or examination at this or any other university.

___________________________  _________________________
Signature                      Date
Acknowledgements

I would like to sincerely thank my supervisor, Mrs Indrani O’Leary Govender and co-supervisor Miss Andrew Roxanne, for their invaluable guidance and assistance during this research. I would also like to thank the Department of Finance and Investment Management at the University of Johannesburg for giving me the opportunity to pursue this Master’s degree. Special thanks goes to the entire class of 2016, especially Mlibazisi Ngwenya for all the sleepless nights. To my family, thank you for the encouragement without which the completion of this thesis would not have been possible. Finally, I would like to thank the Almighty.
Table of Contents

Chapter 1: Introduction .............................................................................................................. 1
  1.1 Background ................................................................................................................ 1
  1.2 Problem statement ...................................................................................................... 3
  1.3 Research question ....................................................................................................... 3
  1.4 Objectives .................................................................................................................. 4
  1.5 Methodology ............................................................................................................... 4
  1.6 Data collection and analysis ....................................................................................... 4
  1.7 Limitations .................................................................................................................. 5
  1.8 Contribution of study ................................................................................................. 5
  1.9 Dissertation overview ................................................................................................. 6

Chapter 2: Literature review .................................................................................................... 7
  2.1 What is a stock market? ............................................................................................... 7
    2.1.1 Johannesburg Stock Exchange ........................................................................... 7
  2.2 Volatility .................................................................................................................... 7
    2.2.1 What is volatility? ............................................................................................... 7
    2.2.2 Why is volatility important? .............................................................................. 8
    2.2.3 South African Volatility Index ........................................................................... 8
    2.2.4 Volatility forecasting ......................................................................................... 9
  2.3 Artificial Neural Networks ......................................................................................... 10
    2.3.1 Biological inspiration ....................................................................................... 10
    2.3.2 Brief history of Artificial Neural Networks .................................................... 12
    2.3.3 Artificial Neural Network architecture ............................................................. 13
    2.3.4 Learning ............................................................................................................ 14
    2.3.5 Weaknesses and strengths of Artificial Neural Networks ............................. 14
    2.3.6 Artificial Neural Networks in time series forecasting .................................... 15
    2.3.7 Artificial Neural Networks in finance .............................................................. 16
    2.3.8 ANNs in stock market volatility forecasting ................................................ 17
  2.4 Conclusion .................................................................................................................. 18

Chapter 3: Data and methodology .......................................................................................... 19
  3.1 Problem statement ...................................................................................................... 19
  3.2 Research question ...................................................................................................... 19
  3.3 Research objectives .................................................................................................... 19
  3.4 Research design ......................................................................................................... 20
  3.5 Sampling strategy ....................................................................................................... 21
    3.5.1 Target population ............................................................................................. 21
    3.5.2 Sample selection ............................................................................................... 21
3.6 Data collection ........................................................................................................ 22
    3.6.1 Input variable identification ........................................................................... 22
3.7 Data pre-processing .............................................................................................. 28
    3.7.1 Linear interpolation ....................................................................................... 28
    3.7.2 De-trending .................................................................................................... 29
    3.7.3 Normalisation ................................................................................................. 30
3.8 Data partitioning ................................................................................................... 30
3.9 Input variable selection ......................................................................................... 31
    3.9.1 The Boruta algorithm ................................................................................... 33
3.10 Artificial Neural Network construction .............................................................. 33
    3.10.1 Number of hidden layers ............................................................................ 34
    3.10.2 Number of hidden neurons ......................................................................... 34
3.11 Artificial Neural Network training ...................................................................... 35
    3.11.1 Training algorithm ..................................................................................... 35
    3.11.2 Learning rate and momentum ...................................................................... 36
    3.11.3 Number of training iterations ..................................................................... 36
    3.11.4 Transfer function ......................................................................................... 37
3.12 Evaluation of the Artificial Neural Network ....................................................... 37
    3.12.1 Root Mean Square Error .............................................................................. 38
    3.12.2 Mean Absolute Percentage Error ............................................................... 38
    3.12.3 Accuracy ....................................................................................................... 39
    3.12.4 Sensitivity ..................................................................................................... 40
    3.12.5 Specificity .................................................................................................... 40
    3.12.6 Kappa ........................................................................................................... 40
3.13 Artificial Neural Network implementation .......................................................... 41
    3.13.1 Software ...................................................................................................... 41
3.14 Reliability of the study ........................................................................................ 41
3.15 Validity of the study ............................................................................................ 42
3.16 Ethical considerations of the study ..................................................................... 42
3.17 Limitations of the study ...................................................................................... 42
3.18 Significance of the study .................................................................................... 43
3.19 Conclusion ............................................................................................................ 43
Chapter 4: Empirical results ....................................................................................... 44
4.1 Experiments and results ....................................................................................... 44
    4.1.1 Scenario A experiments and results ............................................................ 44
    4.1.2 Scenario B experiments and results ............................................................ 49
4.2 Conclusion ............................................................................................................. 53
Chapter 5: Conclusion and future work ................................................................. 54
  5.1 Reason for undertaking the study .................................................................. 54
  5.2 Summary of findings ...................................................................................... 54
  5.3 Limitations ..................................................................................................... 55
  5.4 Future research .............................................................................................. 55
List of References .................................................................................................. 56
Appendix .................................................................................................................. 62
  Appendix A: ANN in R programming language .................................................. 62
List of Figures

Figure 2.1: The negative correlation between the SAVI and the JSE Top 40 Index .......... 9
Figure 2.2: The human brain .................................................................................. 11
Figure 2.3: Multilayer Perceptron............................................................................. 13
Figure 4.1: Actual versus predicted........................................................................... 48
Figure 4.2: Autocorrelation plot................................................................................. 49
Figure 4.3: Autocorrelation plot................................................................................ 53
List of Tables

Table 1.1: Dissertation overview.................................................................6
Table 3.1: Input variables..................................................................................26
Table 4.1: Boruta results..................................................................................44
Table 4.2: Training set results.........................................................................46
Table 4.3: Testing set results...........................................................................47
Table 4.4: Boruta results..................................................................................50
Table 4.5: Training set results.........................................................................51
Table 4.6: Testing set results...........................................................................52
Chapter 1: Introduction

A detailed theoretical background of the research is presented, followed by a comprehensive discussion of the research problem. The research question, aims and objectives are then stated, followed by a brief section which discusses the rationale of the research. The data techniques and methodological approaches that are used to address the research objectives are outlined. The final part of the chapter outlines the structure of the dissertation.

1.1 Background

The forecasting of volatility has been a subject of immense importance, both in industry and academia. Mzamane (2013, p.2) defines volatility as the “dispersion of returns for a given market index”. Volatility has long been considered an important metric in portfolio management, asset allocation, option pricing and risk management decisions. High volatility typically results in a large deviation from the mean, implying high risk, whilst low volatility typically implies low risk (Oratile, 2014). The volatility of financial returns can be characterised by five defining properties: clustered, heteroscedastic, leptokurtic, time-varying and asymmetrical (Adesina, 2013). These stylised properties of volatility have resulted in numerous studies which attempt to adequately define each of the characteristics.

The theoretical underpinnings of the modelling and forecasting of volatility can be traced to the work of Engle (1982), who attempted to capture the time-varying characteristic of volatility by introducing the Autoregressive Conditional Heteroscedastic (ARCH) model. The ARCH model is largely viewed as a success in capturing the time-varying aspect of volatility, however, it has been criticised for requiring many input variables to properly estimate model parameters (Chand, Kamal & Ali, 2012).

Bollerslev (1986) later improved upon the ARCH model by introducing the Generalized Autoregressive Conditional Heteroscedastic (GARCH) model. This sought to capture the leptokurtic and clustering characteristics of volatility. Although the GARCH model is a parsimonious version of the ARCH model, its inability to adequately capture the leverage effects of volatility have earned it widespread criticism (Lim & Sek, 2013). To address this, many variations of the GARCH family of models have been introduced. To accommodate leverage and asymmetry in volatility, Andersen and Lange (1996) introduced the Exponential Generalized Autoregressive Conditional Heteroscedastic (EGARCH) model. The Fractionally Integrated Generalized Autoregressive Conditional Heteroscedastic (FIGARCH) and the Asymmetric Generalized Autoregressive Conditional Heteroscedastic (AGARCH) models followed in later years (Brooks, 2008).
After the 1987 global stock market collapse, the need to accurately forecast stock market volatility increased. Widespread reports of terrorism, wars, political tensions, natural disasters and financial scandals in various parts of the world were a constant source of panic and instability. These precarious events led to a global stock market that is constantly subdued and prone to high levels of volatility (Oratile, 2014). Highly volatile markets hinder the ability of investors to make wise investment choices as speculative behaviour becomes the norm. Similarly, governments throughout the world are experiencing problems implementing policies that spur economic growth due to the volatility of the financial markets (Mzamane, 2013). This presents a challenge for researchers in academia and practitioners in industry to find efficient and robust methods of modelling and forecasting stock market volatility. Efficiently predicting stock market volatility would greatly assist investors to construct balanced portfolios and accurately value stocks, traders to design dynamic hedging strategies for options and futures and monetary authorities to monitor the vulnerability of their economies.

In recent times and due to technological advancements, machine learning models are slowly gaining traction within finance. Machine learning is a field of computer science concerned with the design and construction of algorithms that iteratively learn from and find hidden insights in data without being explicitly programmed. Machine learning models comprise a set of inputs, target responses and a learning algorithm. Artificial Neural Networks (ANNs) are the most widely used machine learning models in finance (Coakley & Brown, 2000). Haykin (2010, p.102) defines ANNs as “systems which can acquire, store and utilise knowledge”. ANNs have been used to solve several problems in various financial domains which include fraud detection in banks, stock screening, portfolio optimisation and asset allocation. Inspired by the biological design of the human brain, ANNs are made of neurons which are interconnected through various layers which enable them to learn complex patterns inherent in the data. Their popularity, particularly in the time series forecasting community, stems from the fact that ANNs are non-linear and data-driven models which can extrapolate patterns in complex data (Engelbrecht, Geldenhyus & Cloete, 1995).

Despite the popularity of ANNs in solving a myriad of problems in finance, very little is known about their performance in forecasting stock market volatility. Although some studies have been conducted by authors such as Ladokhin (2009), Chaudhuri and Ghosh (2015), most of these have focused on developed nations with little or no attention to developing markets. In the context of South Africa, one study conducted by Harrilall and Seetharam (2015) at the University of the Witwatersand has been published. In the study, the authors compare the performance of a Time Delay Neural Network (TDNN) to the traditional time series models in forecasting the South African Volatility Index (SAVI). Since its launch by the Johannesburg Stock Exchange (JSE) in 2007, the SAVI has generated significant research interest. It is used
as a measure of the three-month implied volatility in the South African equity market and is widely considered as an investor fear gauge (Kenmoe & Tafou, 2015). Oratile (2014) states that the SAVI tends to spike upwards in periods of high volatility, indicating fear in the markets and spikes downwards in periods of low volatility.

The research by Harrilall and Seetharam (2015), though the first of its kind in Africa, provides little insight by way of conclusive evidence on the performance of ANNs in forecasting stock market volatility. This is due to the poor choice of ANN architecture and the inadequate number of input variables used in the research. To address this shortcoming, this study seeks to expand on the work of Harrilall and Seetharam (2015) by forecasting the SAVI using a Multilayer Perceptron ANN. The Multilayer Perceptron is by far the most common and successful ANN architecture to date (Haykin, 2010). Although the SAVI is more specific to the equity market, it is widely considered the premier benchmark of stock market volatility in South Africa and as such the findings of the research will help to uncover the potential of ANNs as viable tools for volatility forecasting in the South African stock market (Joseph & Oosthuizen, 2009).

1.2 Problem statement

The Johannesburg Stock Exchange (JSE) is ranked as the 19th largest stock exchange in the world by market capitalisation and is thus the largest on the African continent (Johannesburg Stock Exchange, 2017). Despite South Africa’s high standing on the African continent, research on the modelling and forecasting of volatility in the South African stock market remains an under-researched area. The findings of studies conducted thus far have largely proven inconclusive. Further compounding this dearth of information is the reluctance of researchers to go beyond traditional econometric models. Most researchers such as Oratile (2014), Mzamane (2013) and Oberholzer and Venter (2015) continuously re-use the same tried, and in most cases, poorly performing models resulting in less than telling conclusions. Niyitegeka (2013) states that the modelling and forecasting of stock market volatility is made difficult by the complicated interconnections inherent in financial market data. This means that non-linear models in the mould of ANNs may be better suited to capture the dynamics of stock market volatility.

1.3 Research question

The research seeks to answer the following question:

Can a Multilayer Perceptron ANN forecast the SAVI?
1.4 Objectives

The aim of the research is to forecast the SAVI using a Multilayer Perceptron ANN.

The objectives of the research are:

➢ To evaluate the performance of a Multilayer Perceptron ANN in forecasting the next trading day’s SAVI return.
➢ To evaluate the performance of a Multilayer Perceptron ANN in forecasting the movement direction in the next trading day’s SAVI return.

1.5 Methodology

In this study, a Multilayer Perceptron ANN will be used to forecast the SAVI. The study is empirical in nature and takes the form of a quantitative, experimental research design. The study follows a post-positivist research paradigm. Several simulation experiments are conducted where the ANN is trained to forecast the next trading day’s SAVI return and the next trading day’s movement direction in the SAVI return. Performance evaluation of the ANN is based on the results of the testing set and measured by various metrics such as the Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), accuracy, specificity and sensitivity. The computer simulation code for implementing the ANN is written using the statistical software R version 3.3.3.

1.6 Data collection and analysis

Secondary data consisting of twenty financial time series is collected from the I-Net BFA database. The data is daily and covers the period 8 January 2010 to 10 July 2017. Twenty-nine fundamental, technical and intermarket input (predictor) variables are derived from the data. The data is pre-processed, firstly through linear interpolation to fill in missing values, secondly by de-trending to remove seasonality, trends and outliers in the data and thirdly, by normalising the data to scale within the range of the ANN activation function. Finally, the data is partitioned into training and testing sets.
1.7 Limitations

The limitations of the research are outlined below.

➢ The first limitation of the research is related to the data. In order to avoid the structural breaks in time series associated with the 2007-2009 financial crisis and to incorporate new changes in the SAVI calculation, this research considers data from January 2010 onwards, which limits the data sample size.

➢ The second limitation pertains to the design of the ANN. Literature does not provide a method of determining the sufficient number of hidden neurons to be used in building an ANN model, hence this parameter is determined through trial and error which is largely time consuming.

1.8 Contribution of study

This aim of this research to forecast the SAVI using a Multilayer Perceptron ANN. The SAVI represents the three-month expected volatility in the South African equity market. A change in the SAVI corresponds to a change in the expected volatility (Kenmoe & Tafou, 2015). The findings of this research will therefore provide a different dimension to the existing literature on volatility forecasting in the South African stock market.
1.9 Dissertation overview

Table 1.1: Dissertation overview

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Chapter 1 introduces the research, presenting the background, problem statement, objectives, questions, rationale, data analysis, methodology and limitations of the research.</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Chapter 2 provides an extensive literature review of studies conducted within a similar research framework.</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Chapter 3 presents the data analysis, methodology and design of the research.</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Chapter 4 presents the results and findings of the research. An extensive discussion of the results is done with a focus on comparing the research findings to studies in literature.</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Chapter 5 presents the conclusions of the research. Potential areas of future exploration are highlighted</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Appendix A presents the R statistical software code for the implementation of the ANN.</td>
</tr>
</tbody>
</table>

Source: Author
Chapter 2: Literature review

This chapter provides a comprehensive literature review of the research topic. The chapter begins by introducing the concept of stock markets and then discusses volatility, highlighting its major defining characteristics. A comprehensive theoretical and empirical background on ANNs is then presented, focusing on their biological origins, architectural design and learning processes. A review of the applications of ANNs in the finance industry is then outlined. The chapter concludes by stating the research gaps identified in the field of forecasting stock market volatility using ANNs.

2.1 What is a stock market?

Odhiambo (2010) defines a stock market as "a public platform where the issuing and trading of stocks of publicly listed companies, bonds, derivatives and other securities takes place through formal exchanges or over-the-counter markets at an agreed price". The stock market provides an avenue for companies to generate additional capital by selling their shares to the public. A company that is deemed to be successful, increases in share price whilst one perceived to be failing, decreases in share price (Odhiambo, 2010). A healthy stock market contributes to the overall wellbeing of the economy by promoting the production of goods and services by lowering costs associated with production and enabling companies to raise sufficient capital to expand, create jobs, services and opportunities (Odhiambo, 2010).

2.1.1 Johannesburg Stock Exchange

Established in 1887, the Johannesburg Stock Exchange (JSE) became part of the World Federation of Exchanges in 1963. To date, the JSE comprises 62 equities, 92 commodity derivatives, 102 interest rate derivatives and 120 equity derivatives members licensed in South Africa (Johannesburg Stock Exchange, 2017).

2.2 Volatility

2.2.1 What is volatility?

Ladokhin (2009, p. 5) defines volatility “as the amount of uncertainty about the size of changes in a security’s value”. Volatility of asset returns has five defining characteristics (Oratile, 2014), (Mzamane, 2013), (Poon & Granger, 2003):
➢ It is not constant but reverts to the mean.
➢ It tends to cluster where periods of high and low volatility are followed by periods of high and low volatility.
➢ It is characterised by leverage effects in which volatility and price movements are negatively correlated.
➢ It is leptokurtic (i.e. comprises fat tails), hence shocks to the economy have a strong impact on volatility.
➢ The explanatory power of volatility is greater when measured at a high sample than compared to lower sampling periods.

A special feature of the volatility of asset returns presented by Tsay (2010) is the conditional variance. According to Tsay (2010), this feature of volatility is not observable but extremely useful for portfolio allocation and valuation of derivatives.

2.2.2 Why is volatility important?

Mzamane (2013) states that volatility of asset returns is important for three primary reasons:

➢ It is used as an input to the Black-Scholes pricing model to calculate the variance of a financial asset.
➢ It is used as an input to the Binomial Tree Model for pricing American and European options.
➢ It is used to quantify Value at Risk (VaR) which is critical for risk management in banks.

Poon and Granger (2003) present compelling reasons as to the importance of forecasting stock market volatility. They argue that forecasting stock market volatility impacts investment decisions, risk management, valuations of security, risk and monetary policy-making. They state that the predictability of stock market volatility would have a direct impact on efficient portfolio selection and the smooth functioning of a country’s financial system. Furthermore, it would give policy-makers a means to handle potential shocks and variances in the financial markets.

2.2.3 South African Volatility Index

In 2007, the JSE launched a volatility index called the South African Volatility Index (SAVI). The SAVI is a measure of the three-month rolling volatility in the South African equity market (Kenmoe & Tafou, 2015). The index is calculated by averaging the daily three-month polled at-the-money volatilities of the JSE Top 40 Index. A low SAVI (i.e. below 22) indicates a bearish investor sentiment whilst a high SAVI (i.e. 28 and above) indicates a bullish investor
sentiment (Kenmoe & Tafou, 2015). The SAVI has been dubbed an investor ‘fear gauge’ for the South African equity market because it tends to spike during periods of uncertainty (Kenmoe & Tafou, 2015).

In 2010, the SAVI was updated by incorporating a volatility skew term which proxies a market crash risk premium. The new SAVI is no longer calculated as the three-month polled daily averages of at-the-money volatilities but rather as the weighted average of calls and puts of various strike prices which are due to expire in the next three months, and an additional volatility skew premium (Oratile, 2014). The volatility premium skew is a fair measure of the market expectations of a crash as it is independent of model assumptions and subjective market expectations (Oratile, 2014).

![Figure 2.1: The negative correlation between the SAVI and the JSE Top 40 Index](image)

*Source: INET BFA, 2017*

### 2.2.4 Volatility forecasting

The various characteristics of volatility have resulted in a plethora of models developed to adequately capture each of the characteristics. Several models in the form of the Autoregressive Moving Average (ARMA) model, Autoregressive Conditional Heteroscedasticity (ARCH) model, Generalised Conditional Heteroscedasticity (GARCH) model, Stochastic Volatility (SV) model, regime switching model and threshold model have been developed over the years in an attempt to model and forecast volatility. Poon and
Granger (2003) review the findings of 93 published studies on the performance of various volatility models in forecasting volatility of asset returns. The authors state that financial volatility is forecastable and that the most attractive class of models in forecasting volatility is the GARCH type of model. They state, however, that despite their superiority, the GARCH class of models are faced with various limitations, chief amongst them being their inability to capture the non-linear dynamics of financial data. They conclude by suggesting a shift towards non-linear models which are not constrained by the limitations of linearity and heteroscedasticity.

In a similar vein, Harrilall and Seetharam (2015) state that linear models are a relatively poor choice of forecasting stock market volatility. They state that the complicated forces which drive economic and financial activities enforce non-linear behaviour into the financial markets. Forecasting should thus involve identifying and exploiting these non-linearities, which can only be achieved by using non-linear forecasting techniques such as Artificial Neural Networks (ANNs).

ANNs are a class of non-linear models which have been used to solve various real-world problems in finance. These include fraud detection in banks, portfolio optimisation, asset allocation and robo-advisors (Moyo & Sibanda, 2014). The models are extremely popular in the forecasting community because unlike the traditional forecasting models, ANNs are data-driven, self-adaptive and fault-tolerant (Haykin, 2010). ANNs have the ability to independently learn from the data and are highly suited for tasks which require knowledge that is not easily specified (Ladokhin, 2009). These models are not constrained by the limitations of heteroscedasticity and linearity in data, making them a better alternative for modelling and forecasting financial market dynamics (Engelbretch, 2002). Despite their popularity in financial forecasting, not much is known about the performance of ANNs in forecasting stock market volatility.

2.3 Artificial Neural Networks

2.3.1 Biological inspiration

The human brain is one of the most complex organisms known to science. With the size of the brain of a fully-grown adult measuring approximately 1,500 cubic centimetres, the human brain can perform highly complex tasks such as pattern recognition, perception and motor control (Cherkassky & Zhong, 2001). The brain is made up of neurons, nerve cells, dendrites and axons. The neuron is the basic unit of the brain that generates electrical signals and impulses which are transmitted to other parts of the brain (Moyo & Sibanda, 2014).
The neurons are interconnected through a dense network of tissue called synapses. These synapses connect the dendrites of one neuron to the axons of another neuron (Haykin, 2010). Synapses have two important attributes: they transmit the signal in one direction only and they are chemical neurotransmitters. The signal is transmitted from the dendrites to the cell body at speeds approximating $10^{-3}$ seconds (Haykin, 2010). When the signal reaches the cell body, it is ‘fired’ to the axon where it accumulates until it reaches a certain threshold (Anctil & Lauzon, 2004). The axon either inhibits or excites the signal before it is passed to the dendrites and subsequently to other neurons in the brain. This constant interaction between the different components of the brain generates brain activity which allows the individual to respond to external stimuli and learn from experience. Estimates put the number of neurons in the brain at 10-500 billion and the number of synapses at 60 trillion (Moyo & Sibanda, 2014).

Figure 2.2: The human brain

Source: Moyo & Sibanda (2014, p. 11)

The theory of Artificial Neural Networks (ANNs) is inspired by the biological brain. ANNs attempt to imitate the functionality and mechanisms of the human brain mathematically (Engelbretch, 2002). They are designed to learn and generalise patterns in a similar manner to the learning curve of a human brain. An ANN, just like the biological neural network, consists of a series of connections between layers of input and output units (Engelbretch, 2002). Each neuron receives inputs (signal) in the form of real valued data. The signal is excited or inhibited by the connection weights of the neuron (Engelbretch, 2002). Once all the incoming signals are received, the ANN computes a net output signal which is propagated as input to other
neurons in the network. The firing of the signal is governed by the activation function (Hamid, 2004).

2.3.2 Brief history of Artificial Neural Networks

The making of machines which mimic the behaviour of the human brain has always been a dream for Science. Several attempts have been made to emulate the learning process of the human brain. Early evidence of these can be traced to the seminal work of Alan Turing, Warren McCullough, Walter Pitts, Donald Hebb and James von Neumann in the early 1940s. Most of the pioneering work in machine learning was done prior to World War II (i.e. prior to 1945) (Binner & Graham, 2004). As early as 1834, Charles Babbage proposed the first design of the electronic computer. His proposal was later implemented by the International Business Machines (IBM) more than eight decades later in 1914 as the first computer.

During 1939-1950, Alan Turing, who is credited as the father of modern day Computer Science, experimented with computers which used the human brain as an archetype. His most famous experiment, known as the Turing test, was developed in 1950 as a test of a computer's ability to think in an intelligent, human-like manner (Bassis, 2015). The test proposes that a human interrogator sits with a typewriter in a different room to two correspondents; one is human and the other a computer. The interrogator asks text-based questions to both the human correspondent and the computer correspondent. Based on the answers, the interrogator tries to differentiate between the computer and the human. The computer is programmed to give deceptive answers to confuse the interrogator. If, at the end of the test, the interrogator cannot tell the difference between the human and the computer correspondent, the computer is said to have passed the test (Bassis, 2015). The work by Alan Turing was the precursor to most modern day artificial intelligence research.

In 1943, McCulloch and Pitts published the first article on ANNs entitled A logical calculus of the ideas immanent in nervous activity. The article proposed the design of a single layer ANN with a binary output of either '0' or '1'. Three years later in 1949, Donald Hebb wrote an article entitled The Organization of Behaviour, where he introduced weights to the ANN proposed by McCulloch and Pitts to allow for the updating of the learning process. After Donald Hebb's article, Marvin Minsky built the first neurocomputer in 1954. Three years later in 1957, Rosenblatt introduced the Perceptron model to improve the learning ability of ANNs. Initially, the Perceptron seemed quite promising, but it failed dismally in classifying patterns which were not linearly separable. This caused some stagnation in ANN research until Minsky and Papert introduced the Multilayer Perceptron in 1969. The Multilayer Perceptron had more processing
power than the Perceptron and could classify patterns which were not linearly separable. In 1974, Werbos introduced the backpropagation learning algorithm, the most powerful ANN learning algorithm to date. Werbos’ work remained largely unknown until it was re-published in 1986 by Rummelhart and McClelland.

2.3.3 Artificial Neural Network architecture

The architecture of an ANN describes how the neurons are arranged in the ANN. The most common ANN architecture is the Multilayer Perceptron (Bassis, 2015). A Multilayer Perceptron uses a non-linear activation function between its neurons. The Multilayer Perceptron was introduced by Minsky and Papert in 1969 in response to the failure of the Single Layer Perceptron to solve non-linearly separable problems (Majumder & Hussain, 2010). The development of the Multilayer Perceptron has expanded the application of ANNs to different fields, mainly due to its simplicity (Cogo & Corazza, 2015). Cybenko (1989), through the Universal Approximation theory, proved that an MLP with one hidden layer can model any function. In the present research, the Multilayer Perceptron is the architecture of choice.

![Multilayer Perceptron](image)

**Figure 2.3: Multilayer Perceptron**

*Source: Binner & Graham (2004, p. 29)*
### 2.3.4 Learning

Learning is the process through which the ANN adjusts its synaptic weights and biases to improve its performance. Haykin (2010) defines learning “as a process by which the parameters of an ANN are adapted through a continuing process of stimulation by the environment in which the ANN is embedded”. Engelbrecht (2002) defines learning as the process by which weights are adjusted in response to a certain input until a set goal is achieved. The most popular ANN learning algorithm is the backpropagation algorithm introduced by Rummelhart in 1967 (Haykin, 2010). The purpose of the learning algorithm is to define the extent to which connection weights are varied to achieve the desired target response. The backpropagation algorithm follows a gradient descent mechanism where weights are tuned towards the direction of low error (Haykin, 2010). The weights are continually adjusted until some stopping criteria is met, which is based on the global error of the ANN (Haykin, 2010). The backpropagation algorithm has been used in over 80% of ANN applications, mainly due to its speed of convergence and ease of use (Raul, 2016). The backpropagation algorithm consists of a two-stage process – the forward phase and the backward phase (Lahmiri, 2011). In the forward phase, the inputs are presented to the ANN and pass through each of the layers of the ANN until they reach the output layer where the output is revealed. The backward phase begins at the output layer, propagating the errors backwards until they reach the input layer. When the errors reach the input layer, the initial weights of the ANN are then updated to try to minimise the error before the inputs are forwarded again to the output layer using the updated weights (Lahmiri, 2011).

### 2.3.5 Weaknesses and strengths of Artificial Neural Networks

ANNs have advantages which distinguish them from the traditional time series models (Moyo & Sibanda, 2014). These are listed below.

- A key advantage of ANNs is that they can model non-linear and complex relationships in the data. This is an important attribute, particularly in financial markets time series forecasting where the data is mostly chaotic and non-linear in nature.
- A second advantage of ANNs is that they are not restricted by the assumptions of linearity and heteroscedasticity in the data unlike the traditional time series models. In traditional time series models, a researcher must ensure that the assumption of linearity and heteroskedasticity (high volatility and non-constant variance) in the data is not violated, however, these conditions do not hold for ANNs as they are able to learn hidden relationships without imposing any fixed relationships in the data.
➢ A third advantage is that ANNs have the ability of generalising (i.e. inferring unseen relationships on unseen data) and approximating complex dependencies no matter the size of the data.

➢ An important distinguishing characteristic of ANNs is their ability to tolerate noise in the data, mainly due to their parallel structures. Noise tolerance means that ANNs can work with chaotic and incomplete data.

Despite these advantages, ANNs have some inherent weaknesses (Moyo & Sibanda, 2014). These are outlined below.

➢ Their ‘blackbox’ nature, which refers to the difficulty in understanding how ANNs operate in approximating functions.

➢ The lack of clear heuristics on how to design an ANN. In the absence of a clear framework, the process of designing an ANN is largely empirical.

➢ They are prone to overfitting the data. Overfitting occurs when the ANN learns the noise better than the actual pattern in the data which affects its ability to generalise to new data. Several methods have been suggested to overcome the problem of overfitting. These include k-fold cross-validation, early stopping and regularisation, however, overfitting remains a challenge for ANN practitioners.

2.3.6 Artificial Neural Networks in time series forecasting

ANNs have been applied to solve several real-world problems in various application areas which include clustering, control systems, optimisation, classification and time series forecasting. Although there is extensive literature on the use of ANNs in each of the domains, the focus of this study is on time series forecasting. Forecasting is the art of predicting the future. Forecasting is a critical component in finance, production and facilities planning, sales and inventory control (Brooks, 2008). Forecasting is done to assist decision-makers to make informed decisions regarding an uncertain future (Pissarenko, 2002). A time series is a sequence of data points arranged in successive order. In a time, series, future values may be dependent or influenced by past values of the time series (Nelson, Hill & Remus, 1999). A time series is made up of four components (Pissarenko, 2002): trend component (i.e. long-term pattern of a time series), seasonal component (i.e. short-term seasonal pattern of a time series), cyclical component (i.e. pattern within a trend of a time series) and irregular component (i.e. the trendless component of a time series). Pissarenko (2002, p. 42) defines time series forecasting as “prediction based on the assumption that the trend of variations in the value of a variable will continue to recur in the future; that tomorrow will resemble yesterday and today”. ANNs have been used extensively for time series forecasting. Their popularity
stems from the fact that unlike the traditional time series models, ANNs are non-linear, data-driven, self-adaptive, can generalise, are universal function approximators and noise tolerant. These properties make them a natural choice for several real-world forecasting tasks where the data is usually highly non-linear, volatile and messy (Cortez, Rio, Rocha & Sousa, 2010). Time series forecasting can be divided into two parts: multi-step ahead forecasting and one step ahead forecasting. One step ahead forecasting uses previous observations to forecast the next observation into the future whilst multi-step ahead forecasting uses previous observations to forecast multiple observations into the future (Brooks, 2008). Several models tend to perform better in one step ahead forecasting than in multi-step ahead forecasting as the forecast horizon is much shorter and less prone to error (Pacelli, Bevilacqua & Azzollini, 2011).

2.3.7 Artificial Neural Networks in finance

The financial industry has been a prime application area for ANNs. Financial time series are by nature chaotic and non-linear, hence the need to find techniques that accurately model such dynamics (Soni, 2005). Most traditional time series models are unable to capture patterns if the underlying data generating process is not linear (Nelson et al., 1999). Although non-linear models in the form of GARCH models have been used to forecast financial time data, they too fail to capture non-linear dynamics in the data if the models are not adequately specified (Ocran & Biekpe, 2007). ANNs have been applied to solve various problems in finance. These tasks include, but are not limited to, stock market forecasting, corporate bonds rating, exchange rate forecasting, inflation forecasting, asset allocation, portfolio management and optimisation, credit risk classification, bankruptcy forecasting, derivatives pricing and interest rate modelling.

Fadlalla and Lin (2001) review over 200 articles which have applied ANNs in finance between the period January 1986 and December 1997. They find that most studies have applied ANNs to stock market forecasting and conclude that although ANNs are superior to the traditional time series forecasting models, they do not always outperform other models. Although ANNs in finance are widely used, this research focuses on their application in stock market volatility forecasting.
2.3.8 ANNs in stock market volatility forecasting

ANNs have been used to forecast volatility in the stock market. The first recorded study that applied ANNs to forecasting stock market volatility was done by Malliaris (1996) who used a Multilayer Perceptron ANN to forecast the one day ahead magnitude and movement of the S&P 100 implied volatility. As input variables into the ANN, the study used 15 variables which reflect various at the money option prices. At the end of the experiment, Malliaris (1996) reports a Root Mean Square Error (RMSE) of 0.01 on the test set in forecasting the magnitude and an accuracy of 80% in forecasting the directional movement of the S&P 100 implied volatility. The correlation between the forecasts and the future implied volatility was 85% with a significance level of 0.0001. The author concludes that ANNs can be relied upon to forecast future implied volatility.

Dixit and Roy (2013) conduct a study in forecasting the one day ahead movement of the Indian Volatility Index (VIX) returns. The Indian VIX index is based on the NIFTY Index Option prices. Dixit and Roy (2013) use a Multilayer Perceptron ANN trained on the backpropagation algorithm, 16 input variables, 1 hidden layer and 1 output layer. At the end of their investigations, Dixit and Roy (2013) report an accuracy of 70% in forecasting the one day ahead movement of the Indian VIX returns. They report the ANN as being more accurate in forecasting the downward movement in volatility (80%) than the upwards movement (60%). The authors conclude that a Multilayer Perceptron ANN is an effective model in forecasting the downward movement of the VIX index.

Another study that forecasts volatility in the Indian stock market was done by Chaudhuri and Ghosh (2015) who use an ANN to forecast the volatility of NIFTY and gold returns. They use an ANN with two output neurons and seven input variables. They experiment with two different ANN architectures and nine learning algorithms. The authors conclude that ANNs are accurate in forecasting the volatility of the NIFTY and gold returns.

In the context of Africa, the only study to be conducted that applies ANNs to forecasting stock market volatility was done by Harrilall and Seetharam (2015) at the University of the Witwatersrand. In their article, Forecasting changes in the South African Volatility Index: A comparison of methods, Harrilall and Seetharam (2015) compare the performance of a Time Delay Neural Network, Historic Average, Simple Exponential Smoothing, Exponential Weighted Moving Average (EWMA), ARMA, GARCH and ARCH models in forecasting the one day ahead returns of the South African Volatility Index. In implementing their ANN, they use six years' worth of data from 2007 to 2013. They use the historical values of the SAVI
returns as the sole input variable. The backpropagation algorithm is used as the training algorithm of choice. In their results, they report an RMSE of 0.029467 for the ANN and conclude that the ANN did not perform significantly better than the traditional time series models. Harrilall and Seetharam (2015) suggest that the results be taken with a ‘pinch of salt’ because they only used one input variable and propose the inclusion of more fundamental, technical and intermarket indicators.

2.4 Conclusion

After an extensive review of the literature on the application of ANNs in forecasting stock market volatility, two key issues emerged: (i) generally, ANNs seem a promising tool in forecasting stock market volatility and (ii) most of the studies conducted so far have been in developed economies, yet in the context of Africa such studies are scant. In South Africa, which has the largest stock exchange on the African continent, much of the research on the forecasting of stock market volatility has been conducted using the traditional econometric models leading to biased and incomplete conclusions. To date, only one study conducted by Harrilall and Seetharam (2015) has applied ANNs in forecasting stock market volatility in South Africa. This has left a huge research gap, hence the need for more South African studies which apply ANNs in forecasting stock market volatility.
Chapter 3: Data and methodology

This chapter presents the methodology and data analysis techniques used in the research. The chapter also presents the research question, objectives and the design of the research. The research methodology to be discussed is largely based on the work of Kaastra and Boyd (1996), described in their seminal paper, Designing a Neural Network for forecasting financial and economic time series. Steps related to the design of ANNs in financial time series forecasting are discussed extensively. These include data collection and pre-processing, input variable identification and selection, neural network construction as well as training, testing and evaluation of the ANN. Ethical considerations and the reliability, validity, limitations and contribution of the study are outlined.

3.1 Problem statement

The forecasting of volatility is a subject of immense importance both in industry and academia. ANNs have proven to be one of the most powerful tools in time series forecasting (Moyo & Sibanda, 2014). Despite their popularity, not much is known about their performance in forecasting stock market volatility, particularly in the context of Africa. Most of the studies that have applied ANNs in forecasting stock market volatility have been in developed economies. In South Africa, the majority of the research on forecasting stock market volatility has continuously re-used the same traditional econometric models despite suggestions by researchers such as Niyitegeka (2013) that financial market volatility is best captured by non-linear models such as ANNs. The only study to have attempted to apply ANNs in forecasting stock market volatility in South Africa to date, was conducted by Harrilall and Seetharam (2015). This presents a huge research gap in so far as studies related to forecasting volatility in South Africa using ANNs and other machine learning methods is concerned.

3.2 Research question

The research seeks to investigate and answer the following question: Can a Multilayer Perceptron ANN forecast the SAVI?

3.3 Research objectives

The aim of the research is to forecast the SAVI using a Multilayer Perceptron ANN. To achieve this aim several experiments are conducted where the ANN’s performance is evaluated in forecasting (i) the next trading day’s SAVI return and (ii) the movement direction in the next trading day’s SAVI return.
3.4 Research design

Research design is defined as the set of procedures that detail the process or guidelines a researcher follows to achieve the research objectives (Creswell, 2009). Research design can be broadly split into two categories: quantitative and qualitative (Chipaumire & Ngirande, 2014). Qualitative research entails exploratory research seeks to understanding reasons, opinions and motivations whilst quantitative research is concerned with the quantification of a problem through numerical analysis (Chipaumire & Ngirande, 2014). A quantitative approach allows the researcher to study the behaviour of variables through numerical analysis and transformation of the data to usable statistics (Chipaumire & Ngirande, 2014). Quantitative research is deductive and follows a post-positivist paradigm which subscribes to the school of thought that reality can be imperfectly discovered (Creswell, 2009).

There are four types of quantitative research methods, namely, descriptive, correlational, true experimental and quasi-experimental design (Creswell, 2009):

- The descriptive method indicates the status of a variable. The data collection is done through observation.
- The correlative method explores the correlation between variables; it does not look for cause and effect relationships between the variables. The data collection is observational in nature.
- True experimental method studies seek out cause and effect relationships between variables. The variables are controlled except for the independent variable that is being manipulated. The relationship between the independent variable and the dependent variable is analysed. True experimental design is predictive in nature.
- Quasi-experimental method studies examine the cause and effect relationship between two or more variables. In this case, the independent variable is not manipulated or controlled.

This study adopts a true experimental quantitative research design. The research is conducted within a post-positivist paradigm which uses deductive reasoning to analyse the data. Creswell (2009) defines a research paradigm as the general view of reality and the manner in which it should be studied. Positivism refers to drawing general conclusions based on social realisms (Thornhill, Lewis & Saunders, 2015).
3.5 Sampling strategy

Bryman and Bell (2015) define a sampling strategy as the “process of selecting a suitable sample of a population for determining characteristics of the whole population”. A population is the set of units from which data is sourced whilst a sample is the segment of the population which is studied to get information about the population (Bryman & Bell, 2015). Sampling can be divided into two types: probability sampling and non-probability sampling (Bryman & Bell, 2015). In probability sampling, all the units in the population have an equal chance of being selected into the sample whilst in non-probability sampling, units have an unequal chance of being selected into the sample (Bryman & Bell, 2015). In non-probability sampling, the sample units are selected based on the researcher’s assumptions with regards to the population of interest whilst in probability sampling the probability of selecting a unit into the sample can be calculated (Bryman & Bell, 2015). This study follows a non-probability sampling strategy which leads to a purposeful and non-random selection of a sample.

3.5.1 Target population

The target population in this study is the secondary data from McGregor BFA of all fundamental, technical and intermarket variables, considered to be predictors of the SAVI.

3.5.2 Sample selection

The sample comprises secondary data from McGregor BFA of 29 fundamental, technical and intermarket variables considered to be predictors of the SAVI from the period 8 January 2010 to 10 July 2017. The period of study is motivated by two reasons:

➢ To avoid the structural breaks in time series associated with the 2007-2009 financial crisis. Structural breaks may present a source of bias and error in the model (Gregory & Dunne, 2012).

➢ The period from January 2010 onwards corresponds to the introduction of the ‘New SAVI’, which incorporates volatility skew in measuring the expected three-month volatility.

The frequency of data is another important consideration when collecting the sample. Kaastra and Boyd (1996) state that the data must be of the same frequency. This research makes use of daily data since stock markets are sensitive to new information. At frequencies higher than daily, certain information pertaining to market dynamics is usually lost, particularly during crisis periods (Zhang, Patuwo & Hu, 1998). Using daily data also ensures that there are enough observations from which to infer valid statistical conclusions.
In this research, time series corresponding to the daily closing values of the following variables are downloaded as Microsoft Excel files from the McGregor BFA website: South African Volatility Index (SAVI), JSE Top 40 Index, VIX Index, Indian VIX Index (VIXI), Russian VIX Index (VIXR), S&P 500 Index, Bovespa Index, Micex Index, Sensex Index, Shanghai Composite Index, USD/ZAR exchange rate, Yuan/ZAR exchange rate, Real/ZAR exchange rate, Rupee/ZAR exchange rate, Rouble/ZAR exchange rate, platinum price (USD), oil price (USD), gold price (USD), Johannesburg Interbank average rate at three months discount rate (JIBAR3), Johannesburg Interbank average rate at nine months discount rate (JIBAR9) and JSE Top 40 Index trading volume.

3.6 Data collection

The data used in this research is obtained from the McGregor BFA website via the University of Johannesburg research portal (http://research.mcgregorbfa.com). Kaastra and Boyd (1996) state that data must be collected from a reputable source which regularly updates the information. McGregor BFA is an internationally acclaimed financial database, headquartered in Johannesburg South Africa and founded in 1986. McGregor BFA provides online financial and stock market data such as statutory information, dividend history, financial statements and ratios.

3.6.1 Input variable identification

One of the most important steps in designing an ANN is the identification of input (predictor) variables (Kaastra & Boyd, 1996). This is one of the most challenging tasks in building an ANN as it involves the identification of variables that are to be used as predictors of the target response (Lux & Kaizoji, 2004). Kaastra and Boyd (1996) state that fundamental and technical indicators from one or several markets should be considered as inputs to the ANN. They define technical inputs as “lagged values of the dependent variable” whilst fundamental inputs are defined as “economic variables which may influence the dependent variable”. They state that knowledge of economic theory can be quite helpful in identifying the input variables which are likely important predictors. Furthermore, Kaastra and Boyd (1996) suggest the use of intermarket data (e.g. currency exchange rates), having noted that financial markets are closely linked both domestically and internationally.

This study identified a combination of 29 technical, fundamental and intermarket input variables. These variables are identified based on their perceived influence on the SAVI. The variables are classified into eight groups as follows:
i. **Stock indices**

- **JSE Top 40 Index:** The JSE Top 40 Index is a weighted index of the largest 40 companies by market capitalisation listed on the JSE. The index was launched in June 2002 (Johannesburg Stock Exchange, 2017).

- **S&P 500 Index:** The Standard and Poor’s 500 Index, commonly called the S&P 500, is an American equity index based on the largest 500 companies listed on the NASDAQ and New York stock exchanges (Bloomberg, 2017). The companies are selected by finance experts based on several factors such as liquidity, industry sector and market size. The S&P 500 index is considered the sole benchmark of large cap US equities and represents the general wellbeing of the US stock market. The index was launched on 4 March 1957 and as of February 2017, the S&P 500 market capitalisation has exceeded $20 trillion (Bloomberg, 2017). The United States is South Africa’s third largest trading partner and the largest economy in the world (Department of Trade and Industry, 2017). South Africa and the US signed the African Growth and Opportunity Act (AGOA) in 2008 which further boosts trade between the two countries (Department of Trade and Industry, 2017).

- **Russian Micex Index:** The Russian Micex Index was launched on 22 September 1997. The index is a capitalisation-weighted composite index of the 50 most liquid Russian stocks on the Moscow Exchange (Bloomberg, 2017). Russia is a member state of the BRICS.

- **Indian BSE Sensex:** The Bombay Stock Index (Sensex) is a market-weighted index of the 30 most liquid, actively traded and financially stable stocks listed on the Bombay stock exchange (Bloomberg, 2017). The BSE Sensex Index was launched in January 1986 and has since grown to a market capitalisation of over $800 billion (Bloomberg, 2017). Currently South Africa is second after Nigeria in terms of exports to India on the African continent. India is a member state of the BRICS (Department of Trade and Industry, 2017).

- **Brazilian Bovespa:** The Brazilian Bovespa Index is a weighted capitalisation index of the most liquid stocks traded on the São Paulo stock exchange (Bloomberg, 2017). It was launched in January 1985. Brazil is currently South Africa’s largest trading partner in Latin America and a member of the BRICS (Department of Trade and Industry, 2017).

- **Chinese Shanghai Composite Index:** The Shanghai Composite Index is a capitalisation-weighted index. The index was launched in December 1990 and tracks all the stocks (A shares and B shares) traded on the Shanghai stock
exchange (Bloomberg, 2017). China is South Africa’s largest trading partner (Department of Trade and Industry, 2017). In 2010, South Africa was upgraded to the diplomatic status of Strategic Comprehensive Partner by the Chinese government (Department of Trade and Industry, 2017). China is a member of the BRICS.

ii. **Interest rates:** The Johannesburg Interbank Average Rate (JIBAR) is the short-term money market rate used by local and foreign banks in South Africa (Bloomberg, 2017). The JIBAR is available in 1, 3, 9 and 12 months’ rates. It is the average South African prime lending rate determined by various local and international banks calculated daily (Bloomberg, 2017). In this research, as a proxy of South African interest rates, the daily relative changes in the 3 and 9 month JIBAR rates are considered as input variables.

iii. **Exchange rates:** Muktadir-al-mukit (2012) states that when the local currency appreciates in a country which is export-oriented, it results in a negative impact on the exports. This affects the domestic stock market, causing great volatility as export-oriented companies listed on the local stock exchange lose competitiveness and become less profitable. This study considers the daily relative change in the South African currency against the US Dollar, Chinese Yuan, Indian Rupee, Russian Rouble and Brazilian Real as input variables.

iv. **Trading days:** In a study of the information content of the SAVI, Kenmoe and Tafou (2015) reveal that Monday is the only day of the week that has a positive and statistically significant effect on the SAVI. They also report that the SAVI shows signs of a seasonal design with an increase in volatility noted from Friday to Monday and a significant decrease noted from Tuesday to Thursday. In this research, as a proxy of trading activity on the JSE, dummy variables representing trading days Monday to Friday are considered as input variables.

v. **Trading volume:** Chipaumire and Ngirande (2014) state that trading volume contains important information about investor sentiment. When investor sentiment increases, overconfident investors underreact, causing security prices to go up, which leads to lower expected returns and resultantly to reduced trading volume. A further increase in investor sentiment causes overconfident investors to dominate the market, which leads to high trading volume as rational investors are unable to counteract the overconfident investors. An increase in investor sentiment reflects an increase in trading volume as overconfident investors gainfully participate in the markets. In this research, as a proxy for investor sentiment and liquidity of the stock market, the daily
relative change in the trading volume of the JSE Top 40 Index is considered as an input variable.

vi. **Commodities:** Creti et al. (2012) study the co-integration relationship between platinum and gold prices as well as the South African stock market returns and find a long run relationship between the variables. They conclude that commodity prices have an intrinsic effect on the performance of the South African stock market. Whilst not much can be found on the relationship between volatility and commodity prices, in this study the relative daily changes in prices of platinum, crude oil and gold are considered as input variables.

vii. **Implied volatility:** Kenmoe and Tafou (2015) present evidence of volatility spillover between the VXN, VIX and the SAVI implied volatility indices. This indicates the dynamic relationship between implied volatility indices across markets. The daily returns of the following implied volatility indices are considered as input variables.

- **VIX Index:** The Chicago Board Options Exchange Volatility Index measures the volatility of the of S&P 500 stock index option prices (Bloomberg, 2017). Launched in 1993, the VIX Index is considered as the leading global investor fear gauge.
- **Russian Volatility Index:** The Russian Volatility Index relays the Russian market’s expectation of the 30-day volatility based on the 30-day volatility calculated from near-at-the-money Russian Trading System index options. Launched in December 2010, it is an investor fear gauge of the Russian market (Bloomberg, 2017).
- **Indian Volatility Index:** The Indian Volatility Index is a measure of investors’ perception of the market’s expected volatility over the next 30 calendar days calculated from the implied volatility of the Nifty stock index options (Bloomberg, 2017). The Indian VIX was launched in April 2008.

viii. **Realised volatility:** Realised volatility is defined as the sum of squared daily return (Oratile, 2014). In this research, as proxy of realised volatility, the volatility of the JSE Top 40 Index returns is used. The formula for calculating realised volatility is defined (Oratile, 2014):

\[
RV = 100 \times \frac{252}{n} \sum_{t=1}^{N_t} r_t^2
\]  

(1)
Where,

252 is the approximate number of trading days in a year

$r_t$ is the continuously compounded daily rate of return

$N_t$ is the number of trading days in the time frame.

Table 3.1: Input variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAVI Technical indicators</strong></td>
<td></td>
</tr>
<tr>
<td>SAV1</td>
<td>The SAVI return on day t-1</td>
</tr>
<tr>
<td>SAV2</td>
<td>The SAVI return on day t-2</td>
</tr>
<tr>
<td>SAV3</td>
<td>The SAVI return on day t-3</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td></td>
</tr>
<tr>
<td>JTV</td>
<td>Relative change in the daily trading volume of the JSE Top 40 Index on day t-1</td>
</tr>
<tr>
<td><strong>Stock Indices</strong></td>
<td></td>
</tr>
<tr>
<td>SP</td>
<td>S&amp;P 500 Index return on day t-1</td>
</tr>
<tr>
<td>MIC</td>
<td>Russian Micex Index return on day t-1</td>
</tr>
<tr>
<td>BOV</td>
<td>Brazilian Bovespa Index return on day t-1</td>
</tr>
<tr>
<td>SEN</td>
<td>Indian Sensex Index return on day t-1</td>
</tr>
<tr>
<td>SCI</td>
<td>Chinese Shanghai Composite Index return on day t-1</td>
</tr>
<tr>
<td>JT</td>
<td>JSE Top 40 Index return on day t-1</td>
</tr>
<tr>
<td><strong>Commodities</strong></td>
<td></td>
</tr>
<tr>
<td>PLAT</td>
<td>Relative change in the daily price of platinum (US dollar) at time t-1</td>
</tr>
<tr>
<td>GLD</td>
<td>Relative change in the daily price of gold (US dollar) at time t-1</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>OIL</td>
<td>Relative change in the daily crude oil price (dollars/barrel) at time t-1</td>
</tr>
</tbody>
</table>

**Exchange Rates**

<table>
<thead>
<tr>
<th>USD</th>
<th>Relative change in the daily exchange rate between US dollar and the South African rand at time t-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRZ</td>
<td>Relative change in the daily exchange rate between the Brazilian real and South African rand at time t-1</td>
</tr>
<tr>
<td>CHN</td>
<td>Relative change in the daily exchange rate between Chinese yuan and South African rand at time t-1</td>
</tr>
<tr>
<td>RUS</td>
<td>Relative change in the daily exchange rate between Russian rouble and South African rand at time t-1</td>
</tr>
<tr>
<td>IND</td>
<td>Relative change in the daily exchange rate between Indian rupee and South African rand at time t-1</td>
</tr>
</tbody>
</table>

**Implied Volatility Indices**

<table>
<thead>
<tr>
<th>VIX</th>
<th>VIX return on day t-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIXI</td>
<td>Indian VIX return on day t-1</td>
</tr>
<tr>
<td>VIXR</td>
<td>Russian VIX return on day t-1</td>
</tr>
<tr>
<td>VL</td>
<td>JSE Top 40 Index volatility return on day t-1</td>
</tr>
</tbody>
</table>

**Interest rates**

<table>
<thead>
<tr>
<th>JB3</th>
<th>Relative daily change in the Johannesburg Interbank average rate at 3 months discount rate at time t-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>JB9</td>
<td>Relative daily change in the Johannesburg Interbank average rate at 9 months discount rate at time t-1</td>
</tr>
</tbody>
</table>

**Trading days**

<table>
<thead>
<tr>
<th>DAY1</th>
<th>Trading day 1 at time t-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAY2</td>
<td>Trading day 1 at time t-2</td>
</tr>
</tbody>
</table>
3.7 Data pre-processing

Data pre-processing is the process of transforming the data to assist the ANN to learn the underlying data-generating pattern. Kaastra and Boyd (1996) state that the first step in data pre-processing is to ensure that there are no missing or corrupt observations. Missing observations should either be completely discarded or replaced. Trend and outliers should also be removed from the data before it is fed into the ANN.

Outliers are defined as data points which fall outside the range of other data points (Zhang et al., 1998). The presence of a trend and outliers complicates the data abstraction and decreases the performance of the ANN (Zhang et al., 1998). In this study, the data is transformed as follows:

- Linear interpolation is done to replace null or missing values.
- Prices and index values are converted into the first difference of natural logarithms.
- Relative changes in the daily levels of the volume, exchange rates and interest rates are computed.
- Data is normalised to fall within the (0, 1) range.

3.7.1 Linear interpolation

As expected of financial market data and because of different holiday and calendar events in various countries, the data has some missing values. Various methods are suggested in literature to remedy missing data points (Cogo & Corazza, 2015). The easiest and probably most common approach is to discard the missing data points. In this study, this would not be wise given the fact that missing values may contain some essential information which may influence the performance of the ANN. A more pragmatic approach is to perform linear interpolation on the missing data (Kaastra & Boyd, 1996). Linear interpolation is an imputation method that fills in missing or corrupt data points by expressing the missing data point in terms of a regression over non-missing, adjacent data points on the data set. If there are two known
values A and C with co-ordinates \((Cx; Cy)\) and co-ordinates \((Ax; Ay)\). A missing value at point B can be found by the linear interpolation formula (Zhang et al., 1998):

\[
B_y = C_y + \frac{(Ay - Cy) \times (Bx - Cx)}{Ax - Cx}
\]  

(2)

### 3.7.2 De-trending

De-trending removes the growth of a time series and enables easy interpretability of the data by the machine learning algorithm (Kaastra & Boyd, 1996). De-trending in this research is done in two steps: (i) prices and index values are converted into the first difference of natural logarithms and (ii) daily relative changes in levels are calculated for volume, exchange rates and interest rates.

#### 3.7.2.1 Logarithmic returns

A method of de-trending data is converting it into the first difference of natural logarithms returns. Logarithmic transformation is useful in simplifying the data which impacts training speed (Kaastra & Boyd, 1996). Taking the first difference of the data is useful in removing trends from the data. The formula for converting data into logarithmic returns is given as (Brooks, 2008):

\[
r_t = \frac{d\log\left(\frac{P_t}{P_{t-1}}\right)}{P_{t-1}}
\]  

(3)

Where,

- \(r_t\) is the daily return at time \(t\)
- \(P_t\) is the closing price of the index at time \(t\)
- \(P_{t-1}\) is the closing price of the index at time \(t-1\)

#### 3.7.2.2 Relative change in level of variable

Another method of de-trending data is to calculate the relative change in the daily values of an indicator (Kaastra & Boyd, 1996). Trend and seasonality is removed by replacing the original time series with differences between subsequent values. It is more appropriate to use relative change in the value of an indicator as opposed to the real values as this ensures
uniformity in the data (Brooks, 2008). Relative change in the value of an indicator is defined by the formula (Brooks, 2008):

\[
\text{Relative change} = \frac{\text{Value of indicator at time } t - \text{value of indicator at time } t - 1}{\text{value of indicator at time } t - 1}
\]

(4)

### 3.7.3 Normalisation

The chaotic nature of financial data may cause problems for forecasting models as the huge variations in values within a short period may make it difficult for the model to optimally learn the pattern in the data (Cogo & Corazza, 2015). In the case of ANNs, the problem is compounded by the fact that the activation function which governs the firing of any neuron is bounded and takes in data within a range of values (Kaastra & Boyd, 1996). Normalisation is done on the data to bring it within the range of the ANN, usually (0, 1) or (-1, 1) depending on the activation function in use (Kaastra & Boyd, 1996). There are several types of normalisation; these include across channel normalisation, long channel normalisation and external normalisation (Zhang et al., 1998). For the purposes of this research, external normalisation is used which normalises the data within the (0, 1) range of the sigmoid activation function. External normalisation is given by the formula (Zhang et al., 1998):

\[
\text{Normalisation} = \frac{x - x^\prime}{\sigma}
\]

(5)

Where,

\(\sigma\) is the standard deviation

\(x\) is the raw data

\(x^\prime\) is the mean of data set \(x\).

### 3.8 Data partitioning

The partitioning of the data into training and testing sets follows the data pre-processing stage. The training set is the largest and is used to train the ANN whilst the testing set is used to test the ability of the ANN to generalise unseen data (Srinivasan & Liew, 1994). The size of the training and testing sets has been the subject of considerable debate in literature. The training and testing sets should be representative of the population sample as inappropriate
partitioning of the data may affect ANN parameters during training (Srinivasan & Liew, 1994). Kaastra and Boyd (1996) advocate a problem-specific approach to splitting the data as they claim no one size fits all. They insist that the best training set to testing set ratio should be either 90% vs 10%, 80% vs 20% or 70% vs 30%. Hamid (2004) empirically investigates the impact of various training and testing set ratios on the performance of ANNs trained to do various forecasting tasks. The author concludes that there is no direct relationship between the ratio of training and testing sets and ANN performance, however, he suggests the use of 30% for testing and 70% for training.

The testing set may be randomly selected from the data or follow immediately after the train set (Kaastra & Boyd, 1996). The advantage of randomly selecting the testing set is that it rids the ANN of bias associated with one market dynamics. The advantage of selecting a testing set that comes immediately after the training set is that the data is usually recent and the ANN is tested on more recent data that defines more recent market dynamics (Kaastra & Boyd, 1996).

In this study, the first 70% of the data is allocated to the training set and the remaining 30% is allocated to the testing set. The testing set immediately follows the training set for the results to be reflective of more recent market phenomena. The training set is from 8 January 2010 to 1 April 2015 and the testing set is from 2 April 2015 to 10 July 2017.

3.9 Input variable selection

Kaastra and Boyd (1996) state that not all input variables are relevant for a particular forecasting problem, hence it is important that having identified the input variables, one must select the relevant input variables. In machine learning, input variable selection is the process of selecting a subset of relevant input (predictor) variables that are most pertinent to the predictive modelling task. Kursa (2010) states the advantages of input variable selection as improvement in forecast accuracy, reduced computational time, reduced measurement and storage requirements and better model understanding.

Cherkassky and Zhong (2001) state that even though ANNs are data-driven models, presenting the ANN with a lot of irrelevant input variables and expecting it to filter the redundant variables will delay learning and training time. Churchland and Sejnowski (2005) argue that using more variables as input does not necessarily translate into better performance of the ANN; instead, the more sensitive the target response is to the input variables, the better the performance of the ANN.
Guyon and Andre (2003) define relevance of a variable in the following manner: let $F$ be a full set of variables, $F_i$ a variable and $S_i = F - F_i$. The following categories of relevance can be applied to the variable:

**Strong relevance** (A variable $F_i$ is strongly relevant if)

$$P(C | F_i, S_i) \neq P(C | S_i) \quad (6)$$

**Weak relevance** (A variable $F_i$ is weakly relevant if)

$$P(C | F_i, S_i) = P(C | S_i) \quad (7)$$

**Irrelevance** (A variable $F_i$ is irrelevant if)

$$P(C | F_i, S_i^\sim) = P(C | S_i^\sim) \quad (8)$$

Strong relevance of a variable means that if that input variable alone is removed, the performance of the model is affected negatively. Weak relevance of a variable means that if that input variable is removed alongside other input variables, the performance of the model is affected negatively (Yu & Liu, 2004). An irrelevant variable means that if that input variable is removed, the performance of the model is not affected negatively. An optimal subset of input variables should include all strongly relevant variables, no irrelevant variables and a subset of weakly relevant variables (Yu & Liu, 2004).

There are two approaches for input variable selection: filter methods and wrapper methods (Ocran & Biekpe, 2007). Filter methods select input variables based on their correlation strength to the target response variable. Filter methods are algorithm-independent and are strictly data-dependent (Altinbas & Tayfun, 2015). One such method is the Pearson correlation coefficient. Wrapper methods on the other hand, evaluate possible interactions between subsets of variables. Unlike filter approaches, wrapper methods have the advantage of detecting interactions between input variables in explaining the target response variable (Guyon & Andre, 2003). They accurately rank variables by their level of importance in explaining the target response variable. One common wrapper method is the Boruta algorithm which is used in this study to select the relevant predictor variables.
3.9.1 The Boruta algorithm

The statistical software R provides an implementation of a wrapper method called the Boruta algorithm. The Boruta algorithm has been used extensively for input variable selection in machine learning tasks (De Silva & Leong, 2015). The algorithm selects relevant input variables in explaining a given target response variable by creating a mirror image of attributes and comparing them to their permuted copies (Kursa, 2010).

The Boruta algorithm iteratively removes variables which are less correlated to the target response variable than some randomly generated attribute (Kursa, 2010). Yu and Liu (2004) conducted experiments in selecting input variables for over 1,000 datasets for various machine learning models using the Boruta algorithm and report a sensitivity close to 100%.

The Boruta algorithm functions in the following manner (Kursa, 2010):

i. Make at least 5 shadow copies of all the variables.

ii. Iterate through the shadow copies and remove their correlation with the target variable.

iii. Compute the Z score of the shadow copies by running a Random Forest algorithm classifier.

iv. Get the results of the highest Z score and assign a rank to each variable that scored a Z score better than its shadow variable.

v. Remove all variables with Z scores lesser than its shadow copy as these variables are irrelevant.

vi. Remove all shadow copies and repeat the procedure until all variables are ranked.

In this study, prior to training the ANN, input variable selection is done using the Boruta algorithm which allocates an importance score to each input variable in explaining the target response variable. A decision is made whether to drop or retain the variable based on the importance score. The variables selected by the Boruta algorithm are used as ANN input.

3.10 Artificial Neural Network construction

Constructing an ANN involves defining the architecture of the ANN. Architecture relates to the manner in which the neurons in an ANN are structured (Moyo & Sibanda, 2014). Selecting the number of input neurons is quite easy as each input neuron corresponds to an input variable, however, selecting the number of hidden layers and the number of neurons in the hidden layers is not an easy task (Kaastra & Boyd, 1996).
3.10.1 Number of hidden layers

One of the most difficult tasks in the design of ANNs is determining the optimal number of hidden layers (Zhang et al., 1998). Kaastra and Boyd (1996) state that the hidden layers are responsible for detecting the underlying data-generating pattern. They suggest that having too many hidden layers weighs heavily on the ANN computation time and affects the accuracy of the model. They conclude that at most, one hidden layer with enough hidden neurons is sufficient to adequately capture the representation of the problem and ensure accuracy. They advise that ANN researchers in financial forecasting should begin with one or two hidden layers and play around with the number of neurons and if that does not yield satisfactory results, then they can increase the number of hidden layers to three or four. Kaastra and Boyd (1996), however, caution that more than two hidden layers will seldom yield positive results. Chaoba (2009) states that having too few or too many hidden layers leads to the model overfitting. Overfitting occurs when the model has reduced degrees of freedom and results in the ANN memorising individual data points instead of learning patterns on the entire dataset.

Several researchers have experimented with the number of hidden layers and report varying results. Chaudhuri and Ghosh (2015) begin training their ANN on two hidden layers, then they decrease the number of hidden layers to one. They conclude that as the number of hidden layers is decreased from two to one, the accuracy of the ANN on the testing set increases. Cortez et al. (2010) argue that the first hidden layer is useful in extracting the local variables of the inputs whilst the second layer is useful in identifying the global variables of the inputs. Moyo and Sibanda (2014) claim that in general, ANNs with fewer hidden neurons provide better performance and less overfitting than ANNs with a lot of hidden neurons. They caution that ANNs with fewer hidden neurons may be powerless to learn the data. Hamid (2004) suggests using a number of hidden neurons which are at least three quarters of the total number of input variables. Sahoo, Schladow and

3.10.2 Number of hidden neurons

Another critical ANN design decision is determining the number of hidden neurons. Piotrowski and Napiórkowski (2013) state that the hidden neurons define the complexity and power of the ANN. In the absence of an exact method to determine the number of hidden neurons, several heuristics have been suggested in literature. Moyo and Sibanda (2014) claim that in general, ANNs with fewer hidden neurons provide better performance and less overfitting than ANNs with a lot of hidden neurons. They caution that ANNs with fewer hidden neurons may be powerless to learn the data. Hamid (2004) suggests using a number of hidden neurons which are at least three quarters of the total number of input variables. Sahoo, Schladow and
Reuter (2009) suggest that the number of hidden neurons be set to three times the number of input variables.

Kaastra and Boyd (1996) state that the best way to determine the number of hidden neurons is through experimentation. In this study, the number of hidden neurons in the research is varied from 5 to 50 following their suggestion.

3.11 Artificial Neural Network training

Training is the process of presenting the ANN with known examples of input to target response variable data mappings to find a set of weights which will reduce the error and ensure that the output from the ANN is as close as possible to the desired target response (Kaastra & Boyd, 1996). Several critical decisions must be made when training the ANN. The decisions include the selection of the training algorithm, learning rate, momentum and number of training iterations. All these parameters have a profound influence on the performance of the ANN (Kaastra & Boyd, 1996).

3.11.1 Training algorithm

The selection of a training algorithm is an important decision in the ANN design process. The backpropagation algorithm is the most popular ANN algorithm (Kaastra & Boyd, 1996). This algorithm uses a gradient descent mechanism which gradually changes the weights towards the steepest error coordinates (Kaastra & Boyd, 1996). There are several variations of the backpropagation algorithm. Several studies have been conducted to compare the performance of backpropagation algorithms for forecasting tasks. Lahmiri (2011) performs a comparative study of various backpropagation algorithms in forecasting tasks. The results indicate that the conjugate and the Levenberg-Marquardt (LM) algorithms perform better than the rest of the algorithms in terms of both accuracy and speed. Harrilall and Seetharam (2015) use the Levenberg-Marquardt algorithm in forecasting SAVI returns. They report fast convergence and training speed. Convergence is attained when the ANN shows no further improvement in error despite continued training for a given set of parameters. Wilamowski and Yu (2010) compare various algorithms in a forecasting competition and report the Levenberg-Marquardt algorithm as being both faster and more stable than the rest of the algorithms. Other authors who have reported on the speed and stability of the Levenberg-Marquardt algorithm include Sakamoto, Matsumoto, Kuwahara and Hayami (2005).
In this research, the ANN is trained on the Levenberg-Marquardt algorithm. The algorithm provides an efficient method of minimising least-squares problems in various disciplines. The Levenberg-Marquardt algorithm combines the speed of the Gauss-Newton method and the convergence ability of the gradient steepest descent algorithms (Zhang et al., 1998). When the current solution is far from the desired target, it descends along the steepest gradient by becoming a gradient descent method and when the current solution is close to the target, it becomes a Gauss-Newton method (Wilamowski & Yu, 2010).

### 3.11.2 Learning rate and momentum

The learning rate and momentum constitute one of the most important ANN parameters. The learning rate determines the rate of adjustments of ANN weights in response to an error in the training set (Harrilall & Seetharam, 2015). One of the biggest drawbacks of the backpropagation algorithm is that it can get stuck in a local minimum (Kaastra & Boyd, 1996). The momentum ensures that the ANN does not converge to a local minimum but a global minimum error surface during the training process. The learning rate and momentum are both chosen to be between 0 and 1 (Zhang et al., 1998). It is still uncertain as to which values of the learning rate and momentum are optimal. A general consensus is that a momentum and learning rate that are too high will result in the ANN overshooting the global minimum increasing training time, whilst a learning rate and momentum that are too low will result in the ANN taking too long to get to the global minima, resulting in it being trapped in a local minimum (Kaastra & Boyd, 1996).

Several heuristics have been developed to determine the optimal learning rate and momentum. Gonzalez (2000) suggests that for a one hidden layer ANN, a learning rate of 0.7 and a momentum of 0.3 is sufficient for good performance. Majumder and Hussian (2010) suggest a learning rate of 0.1 and a momentum of 0.5. Kaastra and Boyd (1996) state that the best approach to determining a learning rate is to start with a higher learning rate of about 0.9 and gradually decrease the learning rate as training progresses. The learning rate and momentum can be left at the default values provided by the vendor software, in this study, the learning rate and the momentum is set at the default value provided by R software of 0.7.

### 3.11.3 Number of training iterations

The number of iterations influences the adjustment of weights in response to errors during training (Zhang et al., 1998). If the number of iterations is set too low, the ANN may under-learn and if it is set too high, the ANN may over-learn (Devadoss & Ligori, 2013). Although the number of iterations is usually determined through experimentation, there is some guidance
as to selecting this parameter. Tsai and Lee (2011) state that the number of iterations should be set anywhere between 1,000 to 3,500. Kaastra and Boyd (1996) state that ANNs should be allowed to train until a state of convergence is reached which they suggest is anywhere between 50 to 5,000 iterations. In this research, the number of iterations is set to 1,000. This is done to maximise the probability of the ANN finding a global minimum error surface.

3.11.4 Transfer function

Another important design decision is the selection of the most appropriate transfer function. The transfer function is responsible for governing the firing of neurons in the ANN (Haykin, 2010). There are three types of transfer functions: threshold, linear and sigmoid activation (Kaastra & Boyd, 1996). Most financial market forecasting studies use the sigmoid activation function (Haykin, 2010). This is due to the non-linear characteristic of financial market data and the differentiability of the sigmoid function (Hamid, 2004). The sigmoid activation function has a lower and upper bound of between 0 and 1. It is monotonically increasing meaning that the output is constantly increasing and never decreases (Haykin, 2010). This study makes use of the sigmoid activation function.

3.12 Evaluation of the Artificial Neural Network

Piotrowski and Napiórkowski (2013) state that the performance of ANNs is best evaluated on the testing set rather than the training set because that way, the generalisation ability of the unseen data can be best assessed. It is important to evaluate how well the model performs on unseen data instead of data it has already been exposed to during the estimation and training period. Brooks (2008) states that models are expected to perform well on the training set hence, to assess how well the model performs, evaluation must be done on the testing set.

In this research experiments are conducted where the ANN's performance is evaluated in forecasting (i) the next trading day’s SAVI return and (ii) the movement direction in the next trading day’s SAVI return. In the first scenario, the ANN forecasts a continuous target variable, hence the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE) are used as the measures of performance. In the second scenario the ANN forecasts a discrete target variable hence the accuracy, sensitivity, specificity and kappa static are used as the measures of performance.
3.12.1 Root Mean Square Error

The Root Mean Square Error (RMSE) has been extensively used as a measure of model performance in various fields (Zhang et al., 1998). The RMSE aggregates the model residuals into a single value which shows the difference between the actual and forecasted values. Tong et al. (2005) state that the major advantage of the RMSE over other performance measures is the direct interpretability of the measure. A model which perfectly maps the model inputs to the target will have an RMSE of 0. The RMSE is defined by the formula (Kaastra & Boyd, 1996):

\[
RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \left( t_{k,p} - o_{k,p} \right)^2}
\]

Where,

- \( P \) is total training set,
- \( k \) is the size of training set
- \( t_{k,p} \) is the target output,
- \( o_{k,p} \) and is the actual output

3.12.2 Mean Absolute Percentage Error

A measure of performance commonly used in forecasting tasks due to its simplicity and ease of interpretability is the Mean Absolute Percentage Error (MAPE). MAPE is defined as the average of absolute percentage errors of a forecasting model (Zhang et al., 1998). A model which perfectly maps the inputs to the target has a MAPE of 100%. Brooks (2008) states that a MAPE of below 100% indicates that the model performs better than a Random Walk model. The formula for computing the MAPE is (Zhang et al., 1998):

\[
MAPE = \frac{1}{N} \sum_{t=1}^{N} \frac{A_t - F_t}{A_t}
\]

(10)
Where,

\( N \) is the size of dataset

\( A_t \) is the actual forecast

\( F_t \) is the future forecast.

### 3.12.3 Accuracy

Accuracy is defined as the total number of correct forecasts made over the total number of forecasts made (Brooks, 2008). One major advantage of accuracy as a measure of performance in classification problems is its simplicity and ease of interpretability, however, a key disadvantage of the measure is its inability to properly represent imbalanced problems where the target response variable has unequal classes i.e. one class outnumbers the other class by a huge margin (Majumder & Hussain, 2010). Accuracy of a model is defined by the formula (Bassis, 2015):

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} 
\]

Where,

TP (true positive) is the fraction of positives correctly identified,

TN (true negative) is the fraction of negatives correctly identified,

FP (false positive) is the fraction of negatives incorrectly identified as positive,

FN (false negative) is the fraction of positives incorrectly identified as negative
3.12.4 Sensitivity

A performance measure used in evaluating the performance of models with imbalanced datasets is sensitivity (Engelbrecht, 2002). Sensitivity is defined as the fraction of positives correctly classified as positive. Sensitivity of a model is defined by the formula (Bassis, 2015):

\[
\text{Sensitivity} = \frac{TP}{TP + FN}
\]  
(12)

3.12.5 Specificity

Another performance measure used in evaluating the performance of models with imbalanced datasets is specificity (Engelbrecht et al., 1995). Specificity is defined as the fraction of negatives correctly classified as negative. Specificity is defined by the formula (Bassis, 2015):

\[
\text{Specificity} = \frac{TN}{TN + FP}
\]  
(13)

3.12.6 Kappa

The Cohen’s kappa statistic handles both balanced and imbalanced classification problems (Kaastra & Boyd, 1996). This metric measures the performance of a classifier relative to the performance of a random classifier (Gregory & Dunne, 2012). A perfect classifier has a Cohen’s kappa of 1. Values of the Cohen’s kappa less than 0 indicate a useless classifier (Kuhn, 2008). The Cohen’s kappa statistic is defined by the formula (Bassis, 2015):

\[
\text{Kappa} = \frac{P(B) - P(F)}{1 - P(F)}
\]  
(14)

Where,

\(P(F)\) is the accuracy of the classifier

\(P(B)\) is the accuracy from a random classifier
3.13 Artificial Neural Network implementation

The software that ANNs are implemented on should be scalable and deployable (Kaastra & Boyd, 1996). An example of such software is the statistical software R.

3.13.1 Software

The software used to implement the ANN models in this research is the statistical software R version 3.3.3. The computer used to run the models is a Dell Dual Core processor Windows 2010 operating system. The training and testing of the ANNs is conducted within the Classification and Regression training (Caret) library in R. The caret package is the most popular machine learning library in R (Pradhan, 2010). The library, which was introduced by the renowned mathematician Max Kuhn in 2008, is fully implemented within the statistical software R. It contains over 200 different methods whose sole purpose is to facilitate machine learning through model building. The caret package was developed with the following objectives (Kuhn, 2008):

➢ To make the process of machine learning, model building and training seamless.
➢ To create semi-automated approaches for model tuning and optimisation of model parameters.
➢ To create a generic library that can be extended to other parallel processing systems.

3.14 Reliability of the study

Reliability refers to the repeatability and consistency of the results (Thornhill et al., 2015). To ensure that the ANN model is reliable, various experiments are conducted where the ANN is trained to forecast (i) the next trading day’s SAVI return and (ii) the movement direction in the next trading day’s SAVI return.

The experiments are conducted under similar but different conditions and the consistency of the results is examined.
3.15 Validity of the study

Validity is defined as the extent to which the methods used accurately measure that which is required to be measured (Thornhill et al., 2015). To ensure validity, the performance of the ANN is evaluated by comparing the predicted values from the ANN against the actual values of the SAVI in the testing set.

3.16 Ethical considerations of the study

Research ethics serve the purpose of providing guidelines for the responsible conduct of research (Creswell, 2009). Thornhill et al. (2015) name three objectives in ethical research:

i. Protection of human participants
ii. Protection of society and environment
iii. Protection of confidentiality and honesty.

This research uses secondary data collected from the INET BFA database. As this research does not involve any human or environmental participants, this research is concerned with the ethical consideration of protection of confidentiality and honesty. Thornhill et al. (2015) state that the data should be collected accurately, comprehensively and legally. In this research, access to the INET BFA database was legally granted by the University of Johannesburg Library. Privacy and confidentiality of the data was maintained by ensuring that the data was kept safe, free from interference and protected from unwanted use. Furthermore, plagiarism in the research is avoided by ensuring that all sources of information obtained from other authors and researchers are duly referenced using the Mendeley software.

3.17 Limitations of the study

The limitations of the research are outlined below.

➢ The first limitation of the research is related to the data. In order to avoid the structural breaks in time series associated with the 2007-2009 financial crisis and to incorporate new changes in the SAVI calculation, this research considers data from January 2010 onwards, which limits the data sample size.

➢ The second limitation pertains to the design of the ANN. Literature does not provide a method of determining the sufficient number of hidden neurons to be used in building an ANN model, hence this parameter is determined through trial and error which is largely time consuming.
3.18 Significance of the study

The key contribution of this study is providing evidence on whether a Multilayer Perceptron ANN can forecast the SAVI. This research adds to the existing literature on the performance of ANNs in forecasting stock market volatility in South Africa. The study results may assist researchers in their quest to find a more suitable model to optimally forecast stock market volatility in South Africa. The findings of this research may serve as a starting point for future research into the effectiveness of computational techniques in forecasting stock market volatility in South Africa.

3.19 Conclusion

This chapter has presented the data collected and method used for analysis in the research. The data is collected from the McGregor BFA database. The dataset comprises seven years of historical data (January 2010 to July 2017) of 29 technical, fundamental and intermarket variables considered to be predictors of the SAVI. The method used for the analysis is the Multilayer Perceptron ANN model. Chapter 4 presents the empirical results of the research.
Chapter 4: Empirical results

This chapter presents the empirical results of the research. The aim of the research is to forecast the SAVI using a Multilayer Perceptron ANN. The objectives of the research are to (i) evaluate the performance of a Multilayer Perceptron ANN in forecasting the next trading day’s SAVI return and (ii) evaluate the performance of a Multilayer Perceptron ANN in forecasting the movement direction in the next trading day’s SAVI return.

To achieve these objectives, two experiment scenarios are conducted using the software package R. In the first, the ANN is trained to forecast a continuous target response variable in the form of the next trading day’s SAVI return. In the second, the ANN is trained to forecast a binary target response variable in the form of the movement direction of the next trading day’s SAVI return. After training, the performance of the ANN is evaluated on the testing set. The results on the testing set enable various inferences to be made on the performance of a Multilayer Perceptron ANN in forecasting the SAVI.

4.1 Experiments and results

4.1.1 Scenario A experiments and results

The first objective of the research is to evaluate the performance of a Multilayer Perceptron ANN in forecasting the next trading day’s SAVI return. The experiments conducted under this objective are classified as ‘Scenario A’. The measures of performance used to assess this objective are the RMSE and MAPE.

Table 4.1: Boruta results

<table>
<thead>
<tr>
<th>Variable</th>
<th>Importance</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>JT</td>
<td>9,832</td>
<td>Confirmed</td>
</tr>
<tr>
<td>JTV</td>
<td>1,438</td>
<td>Rejected</td>
</tr>
<tr>
<td>VL</td>
<td>5,657</td>
<td>Confirmed</td>
</tr>
<tr>
<td>VIX</td>
<td>13,42</td>
<td>Confirmed</td>
</tr>
<tr>
<td>PLAT</td>
<td>2,734</td>
<td>Rejected</td>
</tr>
<tr>
<td>GLD</td>
<td>1,105</td>
<td>Rejected</td>
</tr>
<tr>
<td>OIL</td>
<td>2,918</td>
<td>Rejected</td>
</tr>
<tr>
<td>USD</td>
<td>2,43</td>
<td>Rejected</td>
</tr>
<tr>
<td>SP</td>
<td>12,494</td>
<td>Confirmed</td>
</tr>
<tr>
<td>JB9</td>
<td>3,577</td>
<td>Rejected</td>
</tr>
</tbody>
</table>
Table 4.1 above shows the Boruta results which indicate that 11 of the 29 input variables are confirmed as having explanatory power over the next trading day’s SAVI return. Two input variables are classified as tentative and 16 input variables are rejected. The results show that the variable VIX (VIX return on day t-1) has the highest importance of 13.45%, whilst SAV2 (SAVI return on day t-2) has the lowest importance. The results indicate that of the variables considered, 13 variables in total are relevant in explaining the next trading day’s volatility and as such, are selected as ANN input variables.
Table 4.2: Training set results

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Number of ANN hidden neurons</th>
<th>RMSE</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>5</td>
<td>0.0299771</td>
<td>97.4</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>10</td>
<td>0.0299740</td>
<td>96.3</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>15</td>
<td>0.0299743</td>
<td>97.2</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>20</td>
<td>0.0299954</td>
<td>97.4</td>
</tr>
<tr>
<td>Experiment 5</td>
<td>25</td>
<td>0.0299935</td>
<td>96.8</td>
</tr>
<tr>
<td>Experiment 6</td>
<td>30</td>
<td>0.0299845</td>
<td>96.4</td>
</tr>
<tr>
<td>Experiment 7</td>
<td>35</td>
<td>0.0299932</td>
<td>96.2</td>
</tr>
<tr>
<td>Experiment 8</td>
<td>40</td>
<td>0.0299987</td>
<td>97.3</td>
</tr>
<tr>
<td>Experiment 9</td>
<td>45</td>
<td>0.0299955</td>
<td>96.2</td>
</tr>
<tr>
<td>Experiment 10</td>
<td>50</td>
<td>0.0299985</td>
<td>96.5</td>
</tr>
</tbody>
</table>

*Source: R Software*

Table 4.2 above shows the training set results for various experiments conducted. The RMSE values vary within a similar range throughout all the experiments illustrating the reliability and consistency of the results. The closeness of the RMSE values indicates that the ANN did not overfit (overlearn) the data on the training set. The best RMSE performance of 0.0299740 is attained in experiment 2, when the number of hidden neurons is set to 10. The MAPE which measures the variability between the actual and predicted values varies between 96% to 97%. A MAPE less than 100% indicates that the ANN outperformed the Random Walk model on the training set. This indicates that the SAVI is not a random process and as such non-linear models are justified to forecast and model its dynamics.
Table 4.3: Testing set results

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Number of hidden neurons</th>
<th>RMSE</th>
<th>MAPE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>5</td>
<td>0.02800719</td>
<td>97.9</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>10</td>
<td>0.02800795</td>
<td>96.2</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>15</td>
<td>0.02799758</td>
<td>97.5</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>20</td>
<td>0.02799221</td>
<td>98.2</td>
</tr>
<tr>
<td>Experiment 5</td>
<td>25</td>
<td>0.02799351</td>
<td>97.6</td>
</tr>
<tr>
<td>Experiment 6</td>
<td>30</td>
<td>0.02799427</td>
<td>97.1</td>
</tr>
<tr>
<td>Experiment 7</td>
<td>35</td>
<td>0.02799551</td>
<td>96.3</td>
</tr>
<tr>
<td>Experiment 8</td>
<td>40</td>
<td>0.02799621</td>
<td>97.5</td>
</tr>
<tr>
<td>Experiment 9</td>
<td>45</td>
<td>0.02799725</td>
<td>92.2</td>
</tr>
<tr>
<td>Experiment 10</td>
<td>50</td>
<td>0.02799655</td>
<td>96.6</td>
</tr>
</tbody>
</table>

Source: R Software

Table 4.3 above shows the testing set results for the various experiments conducted. The RMSE values vary within a similar range indicating the reliability of the results. The closeness of the RMSE values indicates that the ANN did not overfit the data on the testing set. The ANN seems to perform significantly better on the testing set than on the training set, judging by the RMSE values, indicating a good generalisation ability by the ANN. A similar sentiment is shared by Harrilall and Seetharam (2015) and Ladokhin (2009). The best performance in terms of RMSE on the testing set is 0.02799221, achieved in experiment 4 when the number of hidden neurons is set to 20. The MAPE varies between 96% to 97% indicating that the ANN outperformed the Random Walk model on the testing set.

Although the testing set results are better than the training set, overall the results are disappointingly worse than those reported by Harrilall and Seetharam (2015) at Witswatersrand University. An RMSE of 0.02679 is reported by the authors on their Time Delay Neural Network. Notwithstanding the fact that they use a much longer investigation period (2007-2012), the implications of this result suggest two possibilities: (i) the proposal that Harrilall and Seetharam (2015) make that perhaps using fundamental and technical input
variables beyond the historical values of the SAVI as a means of improving ANN performance is potentially invalid; in fact the results seem to suggest that perhaps investor sentiment (fear and greed) could be the only major driver of the SAVI and that fundamental and economic indicators are of less importance when it comes to predicting the SAVI. This leads to the question, how does one effectively quantify investor sentiment? The use of proxies such as trading volume as suggested by Lei (2012) may come into play, however these proxies may not fully represent the feeling and sentiment of investors at a particular time. (ii) The results indicate that a Time Delay Neural Network may capture the dynamics of stock market volatility better than a Multilayer Perceptron ANN due to its use of a recurrent short-term memory structure.

The results seem to give credence to the assertion made by Harrilall and Seetharam (2015, p. 66) that, the SAVI is not forecastable and that “any guess at next period’s volatility is just as good as a guess”. The results also differ from the conclusions made by Ladokhin (2009) and Ghosh (2015) who state that ANNs are capable of forecasting stock market volatility, after reporting RMSE of 0.097 and 0.0923 respectively. Our results seem to fundamentally contradict the assertion made by Poon and Granger (2003), that stock market volatility is perfectly forecastable. We argue that stock market volatility could be forecastable, however, the extent to which it is forecastable appears to be completely negligible, at least in the case of the SAVI.

![Figure 4.1: Actual versus predicted](Source: R Software)
Figure 4.1 above shows the actual versus the predicted values on the testing set. The predicted values seem to be trailing the SAVI quite well in terms of the overall trend and not so much the magnitude.

Figure 4.2: Autocorrelation plot

Source: R Software

Figure 4.2 above shows a plot of the error autocorrelation function. The plot shows that the correlation of the prediction errors on the testing set falls within the 95% confidence limit indicating that the errors are uncorrelated and are in fact white noise. This indicates that the specified ANN is quite adequate and stable.

4.1.2 Scenario B experiments and results

The second objective of the research is to evaluate the performance of a Multilayer Perceptron ANN in forecasting the movement direction in the next trading day’s SAVI return. The experiments conducted under this objective are classified as ‘Scenario B’. The measures of performance used to assess this objective are accuracy, sensitivity, specificity and kappa statistic.
Table 4.4: Boruta results

<table>
<thead>
<tr>
<th>Variable</th>
<th>Importance</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>JT</td>
<td>20,725</td>
<td>Confirmed</td>
</tr>
<tr>
<td>JTV</td>
<td>4,909</td>
<td>Tentative</td>
</tr>
<tr>
<td>VL</td>
<td>6,623</td>
<td>Confirmed</td>
</tr>
<tr>
<td>VIX</td>
<td>7,862</td>
<td>Confirmed</td>
</tr>
<tr>
<td>PLAT</td>
<td>2,873</td>
<td>Rejected</td>
</tr>
<tr>
<td>GLD</td>
<td>1,121</td>
<td>Rejected</td>
</tr>
<tr>
<td>OIL</td>
<td>3,621</td>
<td>Rejected</td>
</tr>
<tr>
<td>USD</td>
<td>3,185</td>
<td>Rejected</td>
</tr>
<tr>
<td>SP</td>
<td>8,166</td>
<td>Confirmed</td>
</tr>
<tr>
<td>JB9</td>
<td>5,714</td>
<td>Tentative</td>
</tr>
<tr>
<td>JB3</td>
<td>1,569</td>
<td>Rejected</td>
</tr>
<tr>
<td>RUS</td>
<td>2,177</td>
<td>Rejected</td>
</tr>
<tr>
<td>IND</td>
<td>6,132</td>
<td>Tentative</td>
</tr>
<tr>
<td>CHN</td>
<td>5,99</td>
<td>Confirmed</td>
</tr>
<tr>
<td>BRZ</td>
<td>2,392</td>
<td>Rejected</td>
</tr>
<tr>
<td>VIXI</td>
<td>5,507</td>
<td>Tentative</td>
</tr>
<tr>
<td>VIXR</td>
<td>2,063</td>
<td>Rejected</td>
</tr>
<tr>
<td>BOV</td>
<td>6,219</td>
<td>Confirmed</td>
</tr>
<tr>
<td>SEN</td>
<td>5,381</td>
<td>Tentative</td>
</tr>
<tr>
<td>MIC</td>
<td>12,816</td>
<td>Confirmed</td>
</tr>
<tr>
<td>SCI</td>
<td>3,672</td>
<td>Rejected</td>
</tr>
<tr>
<td>DAY1</td>
<td>1,549</td>
<td>Rejected</td>
</tr>
<tr>
<td>DAY2</td>
<td>2,484</td>
<td>Rejected</td>
</tr>
<tr>
<td>DAY3</td>
<td>1,316</td>
<td>Rejected</td>
</tr>
<tr>
<td>DAY4</td>
<td>1,467</td>
<td>Rejected</td>
</tr>
<tr>
<td>DAY5</td>
<td>1,244</td>
<td>Rejected</td>
</tr>
<tr>
<td>SAV1</td>
<td>76,121</td>
<td>Confirmed</td>
</tr>
<tr>
<td>SAV2</td>
<td>-0,189</td>
<td>Rejected</td>
</tr>
<tr>
<td>SAV3</td>
<td>1,603</td>
<td>Rejected</td>
</tr>
</tbody>
</table>

*Source: R Software*
Table 4.4 above shows the Boruta results which indicate that 8 of the 29 input variables are confirmed as having explanatory power over the movement direction of the next trading day’s SAVI return. Five input variables are classified as tentative and 16 input variables are rejected. The results show that the variable SAV1 (SAVI return on day t-1) has the highest importance of 76.1%, whilst SAV2 (SAVI return on day t-2) has the lowest importance. The results indicate that of the variables considered, 13 variables in total are relevant in explaining the movement direction in the next trading day’s volatility and as such are selected as ANN input variables.

Table 4.5: Training set results

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Number of hidden Neurons</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>5</td>
<td>0.6904828</td>
<td>0.7725054</td>
<td>0.7141077</td>
<td>0.475254</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>10</td>
<td>0.6731732</td>
<td>0.7758124</td>
<td>0.7120145</td>
<td>0.469521</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>15</td>
<td>0.6942462</td>
<td>0.7824412</td>
<td>0.7257951</td>
<td>0.457965</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>20</td>
<td>0.6902472</td>
<td>0.7725752</td>
<td>0.7164028</td>
<td>0.459879</td>
</tr>
<tr>
<td>Experiment 5</td>
<td>25</td>
<td>0.7013871</td>
<td>0.7799232</td>
<td>0.7173897</td>
<td>0.445897</td>
</tr>
<tr>
<td>Experiment 6</td>
<td>30</td>
<td>0.6812621</td>
<td>0.7733862</td>
<td>0.7139666</td>
<td>0.443469</td>
</tr>
<tr>
<td>Experiment 7</td>
<td>35</td>
<td>0.6927691</td>
<td>0.7738639</td>
<td>0.7138833</td>
<td>0.443953</td>
</tr>
<tr>
<td>Experiment 8</td>
<td>40</td>
<td>0.6932872</td>
<td>0.7739957</td>
<td>0.7139963</td>
<td>0.442586</td>
</tr>
<tr>
<td>Experiment 9</td>
<td>45</td>
<td>0.6932891</td>
<td>0.7736339</td>
<td>0.7139692</td>
<td>0.445282</td>
</tr>
<tr>
<td>Experiment 10</td>
<td>50</td>
<td>0.6935997</td>
<td>0.7731199</td>
<td>0.7133386</td>
<td>0.442596</td>
</tr>
</tbody>
</table>

Source: R Software

Table 4.5 above shows the training set results for the various experiments conducted. The results show an accuracy of between 71% to 72% across the experiments, indicating the reliability and consistency of the results. The closeness of the accuracies indicate that the ANN did not overfit the data on the training set. The highest accuracy (72.5%) is achieved in experiment 3 when the number of hidden neurons is set to 15. The Kappa statistic which compares the predicted accuracy to that of chance varies between 44% to 47% indicating that the ANN is at least 44% better than a random guess at predicting the next trading day’s movement direction in volatility. The sensitivity which measures the proportion of True Positives (i.e. correctly predicted upward movement) varies around 70% whilst specificity, which measures the proportion of True Negatives (i.e. correctly predicted downward
movement), varies between 77% to 78%, indicating that the ANN is more efficient in forecasting the downward movement than the upward movement of volatility.

**Table 4.6: Testing set results**

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Number of hidden Neurons</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>5</td>
<td>0.6908397</td>
<td>0.7525084</td>
<td>0.7237077</td>
<td>0.487525</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>10</td>
<td>0.6855725</td>
<td>0.7558194</td>
<td>0.7130125</td>
<td>0.489862</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>15</td>
<td>0.6946565</td>
<td>0.7524415</td>
<td>0.7347951</td>
<td>0.477965</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>20</td>
<td>0.6908397</td>
<td>0.7425753</td>
<td>0.7144029</td>
<td>0.485878</td>
</tr>
<tr>
<td>Experiment 5</td>
<td>25</td>
<td>0.6924749</td>
<td>0.7499237</td>
<td>0.7272275</td>
<td>0.475892</td>
</tr>
<tr>
<td>Experiment 6</td>
<td>30</td>
<td>0.6936392</td>
<td>0.7439622</td>
<td>0.7272275</td>
<td>0.476993</td>
</tr>
<tr>
<td>Experiment 7</td>
<td>35</td>
<td>0.6955553</td>
<td>0.7498667</td>
<td>0.7289662</td>
<td>0.483399</td>
</tr>
<tr>
<td>Experiment 8</td>
<td>40</td>
<td>0.6923696</td>
<td>0.7435996</td>
<td>0.7277896</td>
<td>0.478963</td>
</tr>
<tr>
<td>Experiment 9</td>
<td>45</td>
<td>0.6945399</td>
<td>0.7489899</td>
<td>0.7289666</td>
<td>0.486352</td>
</tr>
<tr>
<td>Experiment 10</td>
<td>50</td>
<td>0.6921596</td>
<td>0.7425528</td>
<td>0.7273336</td>
<td>0.476232</td>
</tr>
</tbody>
</table>

*Source: R Software*

Table 4.6 above shows the testing set results for the various experiments conducted. The results show that the values of accuracy range between 71% to 73%, indicating that the results are consistent and reliable. The closeness of the accuracies indicate that the ANN did not overfit the data on the testing set. The results show that the ANN performed better on the testing set than on the training set indicating a good generalisation ability. The highest accuracy of 73.4% is achieved in experiment 3 when the number of hidden neurons is set to 15. The Kappa statistic varies between 47% to 49% indicating that the ANN performs at least 47% better than a random guess, as to the next trading day’s movement direction of volatility. The sensitivity varies between 68% to 69% whilst specificity varies between is 74% to 75% indicating that the ANN is more accurate in forecasting the downward movement than the upward movement of volatility. This confirms the SAVI as being more of an investor fear barometer than a future-looking, implied volatility index. Our results are quite similar to those of Dixit and Roy (2013) who report an overall accuracy of 70% and much better accuracy in forecasting the downward movement of the VIX Index.
These results are quite encouraging insofar as they show that whilst the next period’s value of the SAVI may not be entirely forecastable, its movement direction does indicate signs of being predictable. Although the SAVI is not yet a tradeable product, it can however be utilised as an asset class that investors can get exposure to through trading in a variance future. Given our results, perhaps, a trading strategy can be used to generate abnormal profits.

![Autocorrelation of Error 1](image)

**Figure 4.3: Autocorrelation plot**

*Source: R Software*

Figure 4.3 above shows a plot of the error autocorrelation function. The plot shows that the correlation of the prediction errors on the testing set falls within the 95% confidence limit indicating that the errors are uncorrelated and are in fact white noise. This indicates that the specified ANN is quite adequate and stable.

### 4.2 Conclusion

This chapter presented the empirical results of the research. A Multilayer Perceptron ANN is applied in forecasting the SAVI. Several experiments are conducted in forecasting (i) the next trading day’s SAVI return and (ii) the movement direction in the next trading day’s SAVI return. The results of the experiments indicate that the Multilayer Perceptron ANN performs significantly better in forecasting the next period’s movement direction of volatility as opposed to next period’s expected volatility. Furthermore, the ANN was more accurate in forecasting the downward movement of volatility than the upward. Chapter 5 presents the research conclusions and future research work.
Chapter 5: Conclusion and future work

In this study, a Multilayer Perceptron ANN was used to forecast the SAVI. Several experiments were conducted where the ANN was trained to forecast the next trading day’s SAVI return and the next trading day’s movement direction in the SAVI return. The performance of the ANN was evaluated on the testing set. In the previous chapter, the results and findings of the study were discussed. This chapter presents the research conclusions and future work recommendations.

5.1 Reason for undertaking the study

The motivation for conducting this study is the limited and inconclusive nature of research regarding the application of ANNs in forecasting stock market volatility in Africa. Most of the studies conducted so far have been in more developed markets, whilst in the context of Africa and the broader third world, only one study by Harrilall and Seetharam (2015) has been made public.

5.2 Summary of findings

The aim of this study was to forecast the SAVI using a Multilayer Perceptron ANN. The study was conducted over the period 8 January 2010 to 10 July 2017. The two objectives of the study were to (i) evaluate the performance of a Multilayer Perceptron ANN in forecasting the next trading day’s SAVI return and (ii) to evaluate the performance of a Multilayer Perceptron ANN in forecasting the movement direction in the next trading day’s SAVI return.

To achieve these objectives several experiments were conducted. Firstly the Boruta algorithm was used to select the relevant predictor variables and secondly ANN training and testing was conducted. The results of the experiments show that whilst the ANN performed dismally in forecasting the next period’s volatility, it fared considerably better in forecasting the movement in the next period’s volatility. Similar to the findings reported by Malliaris (1996) in the United States of America (USA) and Dixit and Roy (2013) in India, the results indicate that the ANN is more accurate in forecasting the downward movement in volatility as opposed to the upward movement, confirming the SAVI as being an investor fear gauge. The empirical results do indicate some potential of the Multilayer Perceptron ANN as a possible tool to forecast the movement direction of volatility.
5.3 Limitations

The limitations of the research are outlined below.

➢ The first limitation of the research is related to the data. In order to avoid the structural breaks in time series associated with the 2007-2009 financial crisis and to incorporate new changes in the SAVI calculation, this research considers data from January 2010 onwards which limits the data sample size.

➢ The second limitation pertains to the design of the ANN. Literature does not provide a method of determining the sufficient number of hidden neurons to be used in building an ANN model, hence this parameter is determined through trial and error which is largely time consuming.

5.4 Future research

A recommendation for future research may be to attempt using a combination of various volatility models in forecasting the SAVI. An ensemble of the GARCH model, which is known for its effectiveness in modelling conditional variance in volatility, and the ANN model which is a non-linear model, may capture the dynamics of stock market volatility more efficiently.
List of References


Appendix

Appendix A: ANN in R programming language

library(xlsx)
library(caret)
library(Boruta)
rmse <- function(error)
{
  sqrt(mean(error^2))
}
prevrows1 <- function(data,n) {sapply(1:n,function(x) c(rep(NA,x),head(data,-x)))}
prevrows2 <- function(data,n) {sapply(2:n,function(x) c(rep(NA,x),head(data,-x)))}
prevrows3 <- function(data,n) {sapply(3:n,function(x) c(rep(NA,x),head(data,-x)))}
dataset <- read.xlsx("TMCOM.xlsx", sheetName = "Sheet1")
dataset <- na.locf(dataset)
dataset$Date <- as.Date(dataset$Date)
str(dataset)
dataset$BOV <- gsub("\\", ",", dataset$BOV)
dataset$SEN <- gsub("\\", ",", dataset$SEN)
dataset$MIC <- gsub("\\", ",", dataset$MIC)
dataset$SCI <- gsub("\\", ",", dataset$SCI)
dataset$SAV <- as.numeric(as.character(dataset$SAV))
dataset$JT <- as.numeric(as.character(dataset$JT))
dataset$JTV <- as.numeric(as.character(dataset$JTV))
dataset$VL <- as.numeric(as.character(dataset$VL))
dataset$VIX <- as.numeric(as.character(dataset$VIX))
dataset$PLAT <- as.numeric(as.character(dataset$PLAT))
dataset$GLD <- as.numeric(as.character(dataset$GLD))
dataset$OIL <- as.numeric(as.character(dataset$OIL))
dataset$USD <- as.numeric(as.character(dataset$USD))
dataset$SP <- as.numeric(as.character(dataset$SP))
dataset$JB9 <- as.numeric(as.character(dataset$JB9))
dataset$JB3 <- as.numeric(as.character(dataset$JB3))
dataset$RUS <- as.numeric(as.character(dataset$RUS))
dataset$IND <- as.numeric(as.character(dataset$IND))
dataset$CHN <- as.numeric(as.character(dataset$CHN))
dataset$BRZ <- as.numeric(as.character(dataset$BRZ))
dataset$VIXI <- as.numeric(as.character(dataset$VIXI))
dataset$VIXR <- as.numeric(as.character(dataset$VIXR))
dataset$BOV <- as.numeric(as.character(dataset$BOV))
dataset$SEN <- as.numeric(as.character(dataset$SEN))
dataset$MIC <- as.numeric(as.character(dataset$MIC))
dataset$SCI <- as.numeric(as.character(dataset$SCI))
dataset$DAY1 <- as.numeric(as.character(dataset$DAY1))
dataset$DAY2 <- as.numeric(as.character(dataset$DAY2))
dataset$DAY3 <- as.numeric(as.character(dataset$DAY3))
dataset$DAY4 <- as.numeric(as.character(dataset$DAY4))
dataset$DAY5 <- as.numeric(as.character(dataset$DAY5))
str(dataset)
dataset$SAV <- Delt(dataset$SAV)
dataset$JT <- Delt(dataset$JT)
dataset$VIX <- Delt(dataset$VIX)
dataset$SP <- Delt(dataset$SP)
dataset$VIXI <- Delt(dataset$VIXI)
dataset$VIXR <- Delt(dataset$VIXR)
dataset$VL <- Delt(dataset$VL)
dataset$BOV <- Delt(dataset$BOV)
dataset$SEN <- Delt(dataset$SEN)
dataset$MIC <- Delt(dataset$MIC)
dataset$SCI <- Delt(dataset$SCI)

# calculate relative changes

dataset$JTV <- (dataset$JTV - prevrows1(dataset$JTV,1))/prevrows1(dataset$JTV,1)
dataset$PLAT <- (dataset$PLAT - prevrows1(dataset$PLAT,1))/prevrows1(dataset$PLAT,1)
dataset$GLD <- (dataset$GLD - prevrows1(dataset$GLD,1))/prevrows1(dataset$GLD,1)
dataset$OIL <- (dataset$OIL - prevrows1(dataset$OIL,1))/prevrows1(dataset$OIL,1)
dataset$JB9 <- (dataset$JB9 - prevrows1(dataset$JB9,1))/prevrows1(dataset$JB9,1)
dataset$JB3 <- (dataset$JB3 - prevrows1(dataset$JB3,1))/prevrows1(dataset$JB3,1)
dataset$USD <- (dataset$USD - prevrows1(dataset$USD,1))/prevrows1(dataset$USD,1)
dataset$RUS <- (dataset$RUS - prevrows1(dataset$RUS,1))/prevrows1(dataset$RUS,1)
dataset$IND <- (dataset$IND - prevrows1(dataset$IND,1))/prevrows1(dataset$IND,1)
dataset$CHN <- (dataset$CHN - prevrows1(dataset$CHN,1))/prevrows1(dataset$CHN,1)
dataset$BRZ <- (dataset$BRZ - prevrows1(dataset$BRZ,1))/prevrows1(dataset$BRZ,1)

dataset <- na.omit(dataset)
dataset$SAV1 <- prevrows1(dataset$SAV,1)
dataset$SAV2 <- prevrows2(dataset$SAV,2)
dataset$SAV3 <- prevrows3(dataset$SAV,3)
dataset$JTV <- prevrows1(dataset$JTV,1)
dataset$PLAT <- prevrows1(dataset$PLAT,1)
dataset$GLD <- prevrows1(dataset$GLD,1)
dataset$OIL <- prevrows1(dataset$OIL,1)
dataset$JB9 <- prevrows1(dataset$JB9,1)
dataset$JB3 <- prevrows1(dataset$JB3,1)
dataset$USD <- prevrows1(dataset$USD,1)
dataset$RUS <- prevrows1(dataset$RUS,1)
dataset$IND <- prevrows1(dataset$IND,1)
dataset$CHN <- prevrows1(dataset$CHN,1)
dataset$BRZ <- prevrows1(dataset$BRZ,1)
dataset$JT <- prevrows1(dataset$JT,1)
dataset$VIX <- prevrows1(dataset$VIX,1)
dataset$SP <- prevrows1(dataset$SP,1)
dataset$VIXI <- prevrows1(dataset$VIXI,1)
dataset$VIXR <- prevrows1(dataset$VIXR,1)
dataset$VL <- prevrows1(dataset$VL,1)
dataset$BOV <- prevrows1(dataset$BOV,1)
dataset$SEN <- prevrows1(dataset$SEN,1)
dataset$MIC <- prevrows1(dataset$MIC,1)
dataset$SCI <- prevrows1(dataset$SCI,1)
dataset$DAY1 <- prevrows1(dataset$DAY1,1)
dataset$DAY2 <- prevrows1(dataset$DAY2,1)
dataset$DAY3 <- prevrows1(dataset$DAY3,1)
dataset$DAY4 <- prevrows1(dataset$DAY4,1)
dataset$DAY5 <- prevrows1(dataset$DAY5,1)
dataset$SAVM <- ifelse(dataset$SAV - prevrows1(dataset$SAV,1) > 0, 1, -1)
dataset <- na.omit(dataset)
dataset$SAV <- as.numeric(as.character(dataset$SAV))
dataset$JT <- as.numeric(as.character(dataset$JT))
dataset$JTV <- as.numeric(as.character(dataset$JTV))
dataset$VL <- as.numeric(as.character(dataset$VL))
dataset$VIX <- as.numeric(as.character(dataset$VIX))
dataset$PLAT <- as.numeric(as.character(dataset$PLAT))
dataset$GLD <- as.numeric(as.character(dataset$GLD))
dataset$OIL <- as.numeric(as.character(dataset$OIL))
dataset$USD <- as.numeric(as.character(dataset$USD))
dataset$SP <- as.numeric(as.character(dataset$SP))
dataset$JB9 <- as.numeric(as.character(dataset$JB9))
dataset$JB3 <- as.numeric(as.character(dataset$JB3))
dataset$RUS <- as.numeric(as.character(dataset$RUS))
dataset$IND <- as.numeric(as.character(dataset$IND))
dataset$CHN <- as.numeric(as.character(dataset$CHN))
dataset$BRZ <- as.numeric(as.character(dataset$BRZ))
dataset$VIXI <- as.numeric(as.character(dataset$VIXI))
dataset$VIXR <- as.numeric(as.character(dataset$VIXR))
dataset$BOV <- as.numeric(as.character(dataset$BOV))
dataset$SEN <- as.numeric(as.character(dataset$SEN))
dataset$MIC <- as.numeric(as.character(dataset$MIC))
dataset$SCI <- as.numeric(as.character(dataset$SCI))
dataset$DAY1 <- as.numeric(as.character(dataset$DAY1))
dataset$DAY2 <- as.numeric(as.character(dataset$DAY2))
dataset$DAY3 <- as.numeric(as.character(dataset$DAY3))
dataset$DAY4 <- as.numeric(as.character(dataset$DAY4))
dataset$DAY5 <- as.numeric(as.character(dataset$DAY5))
dataset$SAV1 <- as.numeric(as.character(dataset$SAV1))
dataset$SAV2 <- as.numeric(as.character(dataset$SAV2))
dataset$SAV3 <- as.numeric(as.character(dataset$SAV3))
dataset$SAVM <- as.factor(as.character(dataset$SAVM))
str(dataset)
corrdataset <- dataset
### calculate correlation matrix
correlationMatrix <- cor(corrdataset[,2:32])
### summarize the correlation matrix
print(correlationMatrix)
# Splitting data between training and testing 70% vs 30%
training <- dataset[1:1309,]
testing <- dataset[1310:1870,]
training$Date <- NULL
training$SAVM <- NULL
testing$SAVM <- NULL
training$SAV <- NULL
testing$SAV <- NULL
##### Boruta to select important variables
set.seed(13)
bor.model <- Boruta(SAV ~ ., data = training, maxRuns=101, doTrace=0)
#summary(bor.model)
boruta.cor.matrix <- attStats(bor.model)
important.variables <- names(training)[bor.model$finalDecision!="Rejected"]
boruta.cor.matrix
#save boruta matrix

boruta <- as.data.frame(boruta.cor.matrix)

boruta2 <- round(boruta.cor.matrix[,1:5],3)

boruta_final <- cbind(boruta2, boruta)

boruta <- subset(boruta_final, select=c(1,2,3,4,5,11))

write.xlsx(boruta, file = "boruta.xlsx")

#neural net during training Regression

library (neuralnet)

training2<- training

grid <- expand.grid(.decay=c(0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009, 0.01), .size=c(15))

model_nnet<- train(training[,2:14],training[,1],method='nnet',trControl=fitControl)

print(model_nnet)

plot(model_nnet)

#get the RMSE on the test set

##forecasting the test set now

###Forecastation

forecastions<- forecast.train(object=model_nnet,testing[,3:14])

close_test <- as.data.frame(forecastions)

close_test$actual_close <- testing$SAVIClose

names(close_test)[1]<-paste("forecasted")

names(close_test)[2]<-paste("actual")

## compute the RMSE

error <- close_test$actual - close_test$forecasted

rmse(error)

##plot a graph of time versus actual versus forecasted
Finalstaff <- cbind(testing[,1],close_test)

colnames(Finalstaff) <- c("Date","Forecasted","Actual")

Finalstaff$Date <- as.Date(Finalstaff$Date)

par(mfrow=2:1)

plot(Finalstaff$Date,Finalstaff$Actual,type = "l",xlab = "DATE",ylab="Actual",col ="blue")

plot(Finalstaff$Date,Finalstaff$Forecasted,type= "l",xlab = "DATE",ylab="Forecasted", col = "red")

#neural net during training Classification

#check table

table(training$SAVM)

#check classes distribution

prop.table(table(training$SAVM))

##fit tune for cross-validation

##for performing cross-validation and avoid overfitting of the model

grid <- expand.grid(.decay=c(0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008,0.009, 0.01), .size=c(20))

model_nnet<-train(data_balanced_both[,1:13], data_balanced_both[,14],method='nnet',trControl=fitControl )

print(model_nnet)

plot(model_nnet)

forecast_raw <-forecast(model_nnet,testing[,2:14], type="raw")

confusionMatrix(forecast_raw,testing[,15]