Application of Deep Neural Networks in Forecasting Foreign Currency Exchange rates

A dissertation submitted to the Department of Business Information Systems, Faculty of Commerce, University of Venda, in partial fulfillment of the requirements for the award of a

Master of Commerce Degree

(e-Science)

of the

University of Venda

by

Nemavhola Andisani

Supervisor Dr Chibaya Colin (Sol Plaatje University)
Co-Supervisor Prof NM Ochara (University of Venda)

April 2021

Declaration

I, Nemavhola Andisani, Student number: 15005397, declare that this research report is my own work being submitted in partial fulfillment for the award of a Master of Commerce degree in Business Information Systems at the University of Venda. This work has not been submitted, in full or in part, for any other degree or examination at any other university or higher learning institution. It is original in design and execution, and all reference material contained therein has been duly acknowledged.

04th of April 2021

Main author | Date

19th of April 2021

Main supervisor | Date

19th of April 2021

Co - supervisor | Date

Abstract

The global foreign currency exchange (Forex) market is regarded as one of the most important financial markets in the world, with daily transactions exceeding $4 trillion. In financial market research, forecasting currency rates is a crucial problem. Forex is notorious for being very volatile and difficult to forecast.

In this study, we investigated the use of deep learning approaches in forex forecasting and compared the success of the Long Short-Term Memory (LSTM) model to the performance of AutoRegressive Integrated Moving Average (ARIMA) and Support vector regression (SVR) when predicting forex rates of US Dollar (USD) pair with South African Rand (ZAR) using daily timeframe data obtained from the Metatrader trading platform.

The LSTM outperformed the SVR and ARIMA models according to MSE data. The LSTM is typically good for predicting USDZAR speeds, although being surpassed by the ARIMA model when the Mean Absolute Error (MAE) was assessed.

Keywords: AutoRegressive Integrated Moving Average, Long Short-Term Memory, Mean Absolute Error, Mean Squared Error, Support Vector Regression.

Acknowledgements

I wish to express my special sincere gratitude to my supervisors, Dr. Colin Chibaya and Professor Ochara for their guidance and patience. This research study would not have been a success without their professional guidance. Thank you both for your trust in me. Thank you for how you empowered me in doing my research.

I would also want to thank my colleagues who tirelessly listened to my ideas and kept on encouraging me to continue working on the project. Thank you for your support.

A special thank you goes to my funders, the National e-Science Postgraduate Teaching and Training Platform, the University of Venda, the University of Witwatersrand, and the Department of Science and Innovation, and the CSIR, for all the financial support. Without you, pursuing these studies would have been close to impossible.

Lastly, I bow down to the All might God for life and energy. My family cannot go unmentioned for being there for me throughout this whole journey. Thank you, Lord. Thank you, family.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

ARIMA	AutoRegressive Integrated Moving Average
Forex	Foreign currency exchange
LSTM	Long Short-Term Memory
MAE	Mean absolute error
MSE	Mean squared error
RNN	Recurrent Neural Network
RMSE	Root mean squared error
SVR	Support vector regression

List of Appendices

Chapter 1 : Introduction

1.1 Background of the study

The global foreign currency exchange (Forex) market is one of the world's best financial markets (Galeshchuk & Mukherjee, 2017; Huang, Lai, Nakamori & Wang, 2004). Every day, more than $4 trillion is exchanged (Talebi, Hoang & Gavrilova, 2014). Forex rates are expressed in terms of base currency pairings, which reflect the number of currency units that may be exchanged for each base currency unit (Galeshchuk & Mukherjee, 2017). Because of economic, political, and trader psychological aspects, currency rates are unpredictable and difficult to anticipate (Kamruzzaman & Sarker, 2004). (Huang et al., 2004). The study is motivated by the question, how can forex rates be anticipated given dynamic economic and political conditions?.

In time-series forecasting, the AutoRegressive Integrated Moving Average (ARIMA) approach has been widely employed (Kamruzzaman & Sarker, 2004). The ARIMA, on the other hand, is a generic model for univariate datasets that was constructed assuming linear time series (Kamruzzaman & Sarker, 2004). A univariate dataset in this context has only one variable of relevance in the investigation (Adhikari & Agrawal, 2013; Siegel, 2016).

In recent years, neural networks have been used to anticipate currency rates (Chandrasekara & Tilakaratne, 2009). Many researchers have demonstrated that artificial neural networks (ANN) outperform ARIMA models in this aspect (Hill, O'Connor, & Remus, 1996; Kamruzzaman & Sarker, 2004; Kim & Kang, 2019; Wang & Leu, 1996).

It is mostly due to the fact that ANNs are non-linear (Di Persio & Honchar, 2016; Hagan, Demuth, Beale, Jesu's, 2014; Kondratenko & Kuperin, 2003; McNelis, 2005), data-driven (Culkin & Das, 2017), and allow little assumptions about the model of the problem (Worasucheep, 2015). However, more research has revealed that ANN has its own limits, such as the overtraining problem that occurs from the adoption and local solution of empirical risk reduction principles (Cao & Tay, 2001). ANN models also require many extremely difficult-to-create control parameters, all of which are extremely limited in their application (Alamili, 2011). As a result, Support Vector Machines (SVM) were developed to address the ANN's problems (Kamruzzaman & Sarker, 2004).

SVMs have been shown to be more successful than ANNs on several occasions (Kim, 2003; Thissen et al., 2003). They are considered as a crucial technology for producing forecasting results for the currency market via classification and regression (Pujari et al., 2018). When regression is used, SVM is referred to as Support Vector Regression (SVR) (Pujari et al., 2018).

Recently, Deep Learning has been introduced and implemented for forecasting forex rates as well (Chatzis, Siakoulis, Petropoulos, Stavroulakis & Vlachogiannakis, 2018; Handa, Shrivas & Hota, 2019; Ni, Wang, Zhang,Yu & Qi, 2019). Deep Learning is part of a larger collection of machine learning algorithms that are centered on learning knowledge representation rather than specialized task computing (Ramadhani & Rismala, 2016). The following are the most recommended models for deep learning; (a) Convolutional Neural Network (CNN), (b) Recurrent Neural Network (RNN), specifically the Long Short-Term Memory architecture (LSTM), and (c) Recurrent Convolutional Neural Network (RCNN) (Vargas, De Lima & Evsukoff, 2017).

The Recurrent Neural Network (RNN) is a neural sequence paradigm that produces cutting-edge output on large tasks such as language modeling

(Kombrink, Mikolov, Karafiat & Burget, 2010). Regularization is required for good neural network implementations, as we all know (Huang, Lai, Nakamori & Wang, 2004). Unfortunately, the most effective approach of regularization for neural Feedforward networks does not perform well with RNNs (Huang, Lai, Nakamori & Wang, 2004). The fundamental disadvantage of RNNs in simulation is that they require significantly more connections and resources than standard context networks (Huang, Lai, Nakamori & Wang, 2004). RNNs offer good results because they are based on the rough recurrence of similar patterns seen in exchange rate time series (Huang, Lai, Nakamori & Wang, 2004). Such frequent yet subtle sequences will offer a good prediction (Huang et al., 2004).

LSTMs are used to solve the problem of long-term dependent cells in generic RNNs (Pujari, Sayyed, Shahani & Rupani, 2018). The basic LSTM architecture is made up of a collection of regularly connected subnetworks called memory blocks. For reading, writing, and resetting memory cells, blocks made up of one or more auto-connected memory cells and three multiplier modules, as well as input, output, and lost gates. The memory cells of LSTM's structure were used to solve the RNN problem (Pujari, Sayyed, Shahani & Rupani, 2018).

The purpose of this research is to look into the use and performance of a deep learning model called the Long Short-Term Memory (LSTM) for forecasting currency rates. Forecasting is the study of patterns and the creation of future forecasts based on historical and present facts (French, 2017). This particular deep learning model is chosen for its ability to forecast time series data (Ni et al., 2019). The study will also compare the performance of the LSTM against the performances of other time series and machine learning algorithms such as ARIMA and SVR when predicting forex rates of the South African Rand against the US dollar (as a case study and proof of concept).

1.2 Statement of the problem

Forex rate forecasting is an important topic in financial market research (Ni, Wang, Zhang, Yu & Qi, 2019; Jozef, Evzen & Lukas, 2017). Often, investors hope to maximize returns on investments (Kim & Kang, 2019). However, such desires of investors to maximize returns usually backfire, resulting in losses instead (Cheng, 2007). An estimated 90% of traders lose money in trading (Cheng, 2007) because of poor forecasting (Yong et al., 2018; Hoang, 2013), other psychological issues (Huang et al., 2004), overconfidence, or a lack of discipline (Levich & Packer, 2017).

The challenge that this study attempts to analyze is the recurring losses suffered by traders, as well as the performance of the LSTM versus the ARIMA and SVR models for forecasting forex rates to prevent additional losses on the part of traders. It appears that traders require assistance in making more informed judgments (Qiu, Wang & Zhou, 2020; Yong et al., 2018). The precise questions we ask in this study are:

a) How do we design an LSTM deep neural network model for forecasting the South African Rand exchange rate against the US dollar?

(b) How does the proposed LSTM model compare in performances to forecasting forex rates using the ARIMA and SVR models?

The answer sought from the first question will present a forex rate forecasting model, while the answer to the second question evaluates the proposed model against the performances of competing forex forecasting models.

The objectives of this study are the following:

a) To design an LSTM deep neural network model for forecasting the South African Rand exchange rate against the US dollar.
b) To investigate and compare the LSTM model's performance against the ARIMA and SVR models when forecasting USDZAR.

The objectives of this study will help drive the study in the right direction to answer the research questions.

1.3 Justification of the Study

A decline in the economy has pushed people into trading (Shahram & Komeil, 2018; DraKoln, 2008). New traders usually lose their capital within the first six months of trading (Shahram & Komeil, 2018). These traders are not well informed about the harsh reality of trading (Shahram & Komeil, 2018; DraKoln, 2008). Investors or traders suffer from depression and anxiety because of the financial losses experienced (Action, 2016; Skapinakis et al., 2006). Such financial difficulties affect people's social life and relationships (Christopher & Janet, 2004). Most people who experience financial problems end up in debts, into drugs, or into possibilities of committing suicide (Richard et al., 2017; Christopher & Janet, 2004).

In the forex markets, even professional traders lose money (Hayley & Marsh, 2016). Excess losses put traders at risk (Shahram & Komeil, 2018; Carlson & Osler, 2003), potentially suffering from stress and other mental health problems (Richard, Elliott, Roberts & Jansen, 2017; Christopher & Janet, 2004).

Some of the traders end up losing their jobs due to the gambler ruin problem (Carlson & Osler, 2003). The proposed model considers all these stakeholders as direct beneficiaries.

The significance and value of this study are emphasized by coming up with a deep neural network model (LSTM) that can help traders in the corporate world when they make informed trading decisions to minimize losses (Yasir et al., 2019).

Exchange rates are key economic indicators in each country (Yasir et al., 2019). Government investment decisions are based on forex rates (Yasir, et al., 2019). The forecast of forex rates is, thus, also important to nations (Yasir, et al., 2019).

Successful completion of this study has implications for life, where stakeholders are better informed. Besides allowing the researcher to obtain a Master of Science degree, the study also creates a name and goodwill for the university at which these studies are completed. The immediate community may use the proposed model for personal benefits. Similarly, forex markets may acquire better insights into the trends thereto.

1.4 Delimitations

This study fills a gap in the discipline of computational sciences, financial mathematics, and partially in economics. The study investigates transactions between currency pairs using computational neural networks to forecast exchange rates.

The researcher mainly anticipates the problem of lack of domain knowledge. How do we develop the LSTM model? The researcher will attempt to interact with people of such expertise. Another problem is lack of mobility because of the covid-19 pandemic. Simulated data will, rather, be collected and used. The potential effects of the proposed intervention in the foreign exchange market are studied and mapped to real-life contexts.

1.5 Operational definitions

A few keywords of this study are defined in this section as follows:

- AutoRegressive Integrated Moving Average (ARIMA) is a hybrid of autoregressive (AR) with the moving average (MA) method (Adhikari & Agrawal, 2013; Hipel & McLeod, 1994).
- Foreign Exchange (Forex) is a form of currency exchange where a party receives certain units from one currency to buy the sum of a proportion in another currency (Galeshchuk & Mukherjee, 2017).
- Long short-term memory (LSTM) is a feedback-connected artificial recurrent neural network (RNN) architecture (Hochreiter, 1997).
- Support vector regression (SVR) is a regression technique of Support vector machine (Pujari et al., 2018).
- A time series is a set of data points that are generally measured over a period oftime (Adhikari & Agrawal, 2013; Cochrane, 2005; Hipel & McLeod, 1994; Raicharoen, Lursinsap & Sanguanbhokai, 2003).

1.6 Cost Estimate

Completion of this study necessitates the use of the following budget:

Item	**Cost estimate**
Printing	R 5000.00
Internet data	R 6000.00
Amazon AWS cluster	R10 000.00
English proofreader	R 4000.00
Others	R 5000.00
Total	**R 30 000.00**

Table 1.1 Estimated costs

1.7 Work plan

The planning of this work that indicates the important target dates as agreed with the supervisors is shown in figure 1.1.

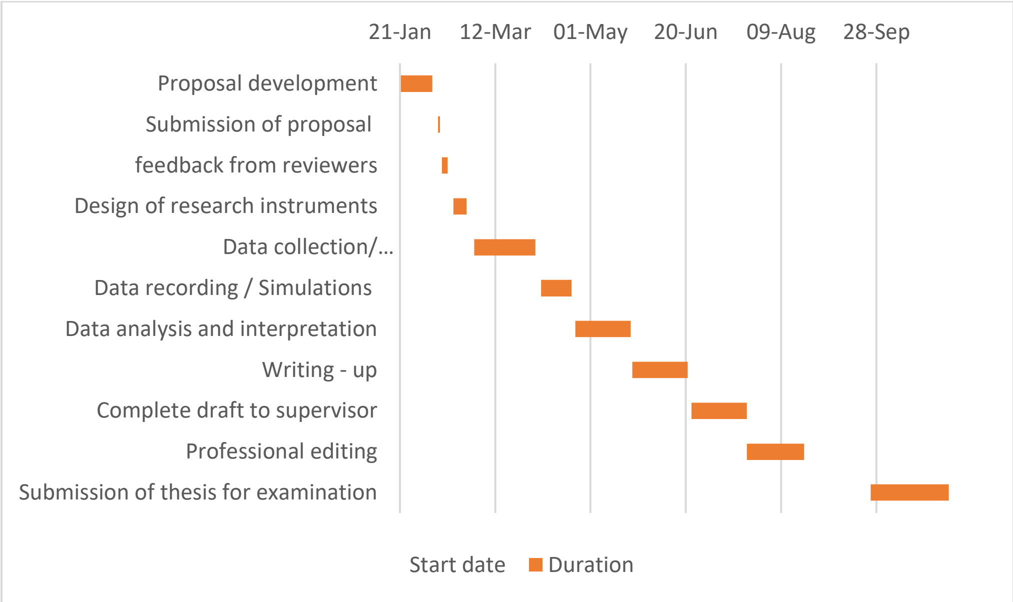

Figure 1.1 Work plan

1.8 Structure of thesis

The dissertation will proceed as follows: Chapter 2 will present a concise literature review, along with the gap we try to fill in the body of knowledge. In chapter 3, we will present the methodology we follow along with the theoretical framework thereto. Data collection strategies, reporting, and the experiments administered will be presented in chapter 4, along with the analysis of the data collected. We will conclude the dissertation in chapter 5, highlighting our key observation, the contributions of the work, and the recommendations emanating. Other important aspects of the study will be added in the appendices of this write-up.

1.9 Summary

This commenced by giving some background to this study in section 1.1. The statement of the problem was pinpointed as consisting of two questions (a) How do we design an LSTM deep neural network model for forecasting the South African Rand exchange rate against the US dollar? (b) How does the proposed LSTM model compare in performance to forecasting forex rates using the ARIMA and SVR models? The answer sought from the first question will prescribe a forex rate forecasting model, while the answer to the second question evaluates the LSTM model against the performances of the ARIMA and SVR models.

The justification of the study was given, emphasizing the negative effects of uninformed trading by newcomers in the trading industry. This justification pointed to the decline in the economy which has pushed people into trading. However, traders lose and fall into depression. Losses affect people's social life and relationships. Losing traders end up in debts, into drugs, or into possibilities of committing suicide.

The proposed model helps traders with informed trading decisions to minimize losses. Government investment decisions are also informed.

The chapter also gave the delimitations, cost estimate, proposed work plan, and the structure of the dissertation. The next chapter presents a Literature review and the gap this study attempts to fill in the body of knowledge.

Chapter 2 : Literature Review

2.1 Introduction

This chapter reviews similar previous studies, focusing, mainly, on the background of forex, computational interventions, as well as the methodologies or techniques that have been proposed in the past for predicting foreign exchange. It focuses on the use of time series, machine learning, and deep learning in forex trading. It also investigates the elements that influence or affect traders in the forex market. After describing the gap in the corpus of knowledge that we hope to address, the chapter ends.

2.1.1 Overview of the Chapter

Figure 2.1 shows the breakdown of this chapter in pictures, showing the six sub-sections of interest, as well as the sequence in which these sub-sections are tackled.

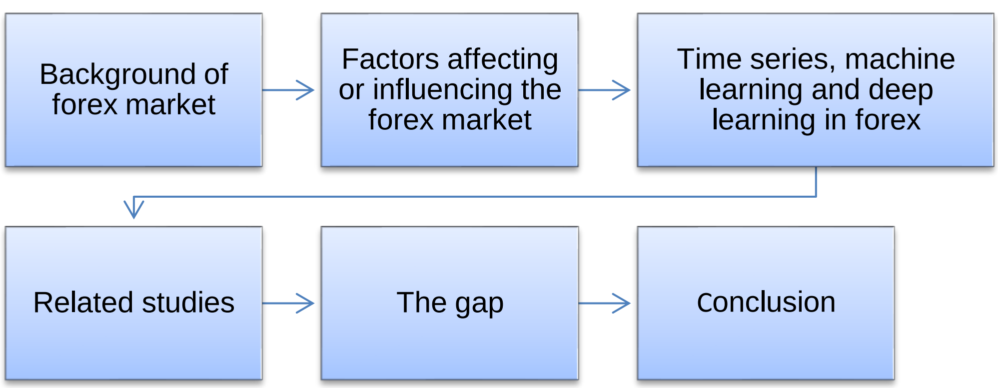

Figure 2.1 Overview of the chapter

2.2 Background of forex market

Following the breakdown of a Bretton Woods agreement that maintained the currency values of gold controlled by central banks, international currency exchange, often known as Forex, began in 1973 (Amiri, Zandieh, Vahdani, Soltani & Roshanaei, 2010). Forex is a form of exchange where a party receives certain units of currency from one denomination to buy the sum of a proportion currency in another denomination (Galeshchuk & Mukherjee, 2017). In this forex market, the principal strategy is to buy and sell low (Amiri, 2010). One trader, for example, sees the Euro rising in value versus the US dollar and decides to purchase EUR / USD at a lower price and sell the currency pair at a higher price when the price rises (Talebi et al., 2014). The most traded currency pairs, also known as major pairs in the forex market, are Euro against United States dollar (EUR/USD), United States dollar against Japanese Yen (USD/JPY), British pound against United States dollar (GBP/USD) and United States dollar against Swiss franc USD/CHF (Violeta, 2010). The base currency is the first currency in a currency pair, while the counter currency is the second currency in the pair (Violeta, 2010).

Different trading sessions on the European, Asian, and American Forex markets are held 24 hours a day, 5 days a week in the worldwide market (Galeshchuk & Mukherjee, 2017). The other advantages of forex are that any currency that is not restricted by the central bank can be traded (Cerrato, Sarantis & Saunders, 2011) and there's no fixed location (Ding & Hiltrop, 2010). Over recent years, the rate of trade in the global foreign exchange grew rapidly (Levich & Packer, 2017). The Bank of International Settlements (BIS) data showed that daily sales on conventional forex products and derivatives rose to $5.3 trillion in 2013 from a reported $590 billion in 1989 (Levich & Packer, 2017). This market has a turnover of nearly 160 times that of the New York

Stock Exchange, the world's largest stock exchange (Lal, 2012; Levich & Packer, 2017).

To predict the market direction, traders usually rely on historical information and news (Kamruzzaman & Sarker, 2004). In Forex markets terms, there are two types of analyses, namely, fundamental and technical analyses. The fundamental term refers to the market movement in conjunction with news or factors which can affect the economics of a country. The technical assessment mainly shows the demand trend of supply through market movement by reading diagrams and indicators of current market prices (Cao et al., 2005). Both factors determine the forex market trends. However, much news on forex trading excludes our African currencies.

2.3 Factors affecting traders or influencing the forex market

Due to economic, political, and psychological variables, currency rates are unpredictable and difficult to anticipate (Kamruzzaman & Sarker, 2004). (Huang et al., 2004). Efficient Market Hypothesis critics claim that investors are inherently irrational, with predictable and financially ruinous preconceptions such as over-confidence and being emotionally involved (Levich & Packer, 2017). The human mind cannot keep up with the market, performing manual trading 24 hours a day (Hoang, 2013). Also, traders can expect unrealistic returns at a limited risk. Fear makes traders decide badly on trade and lack of discipline can result in traders violating trading standards they have pledged to follow. Nevertheless, by using experienced advisors, traders will help avoid these risks. Expert consultants are computer programs based on logic and rationality, which execute automated transactions without human emotions. Expert consultant systems can track the market 24 hours a day and conduct business according to their algorithms (Hoang, 2013). Desires to explore these advantages in the African arena partly motivates the undertaking of this study.

2.4 Time series, machine and deep learning in forex

For most investors, foreign exchange has become another significant investing field apart from indexes, options, funds, and bonds (Baruník, Kočenda & Vácha, 2017). As a result, time series exchange rate data forecasting has become a popular issue in financial mathematics and financial market research (Abu-Mostafa & AF, 1996; Barunk, Koenda & Vácha, 2017; Deboeck, 1994).

For more than two decades, the AutoRegressive Integrated Moving Average (ARIMA) method has been utilized to forecast time series (Box, et al., 2015). In this context, a Time series is a sequence of data points, typically generally monitored over time (Adhikari & Agrawal, 2013; Cochrane, 2005; Hipel & McLeod, 1994; Raicharoen, Lursinsap & Sanguanbhokai, 2003). The ARIMA model was used for the assessment of new modeling methods owing to its success (Baasher & Fakhr, 2011). ARIMA, on the other hand, is a uniform model developed on the assumption that it is linear (Baasher & Fakhr, 2011; Zhang, 2003). The downside of this model is that because of the equivocal and unpredictable nature of data, they are unable to grasp the latent dynamics that occur in time series data (Handa et al., 2019. Therefore, these models are not often used nowadays to classify trends and patterns in historical data, and because of their unpredictable nature not appropriate for reliable analysis of time series results (Handa et al., 2019). Now, methods such as Autoregressive Conditional Heteroskedasticity (ARCH), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Machine Learning, and Deep Learning are used to resolve the challenge when processing and predicting time series results (Handa et al., 2019).

Over the years, machine learning and statistical methods have been introduced for Financial forecasting of foreign exchange. Machine learning is defined as a technique by which the system interacts with its environment in a way that changes the structure of the system and changes the interaction process itself as a result of structural changes behaviour (Alamili, 2011; Ryll & Seidens, 2019). Machine learning is made up of three approaches: Supervised, Unsupervised, and Reinforcement Learning (Bishop, 2006). Supervised learning involves learning from inputs with labels to predict the target variable. Unsupervised learning is learning with no labels to find patterns. Reinforcement is learning through the use of punishment and rewards (Bishop, 2006). However, even though machine learning technology is well adapted for a variety of approximation tasks, it is considered a "black box" model that cannot fully describe their performance behaviour (Alamili, 2011; Ryll & Seidens, 2019).

Some of the machine-based methods that have been used for financial forecasting are Regression Analysis, Discriminate Analysis, Logistic Models, Factor Analysis, Decision Trees, Artificially Neural Networks (ANN), Fuzzy Logic, Genetic Algorithms, and Support Vector Machines (SVM) (Kecman, 2001). In recent years, neural networks have found valuable applications in the analysis and modeling of financial time series (Tang & Fishwick, 1993; Wang & Leu, 1996; Yao & Tan, 2000; Zimmermann et al., 2001). For more irregular series and multi-period forecasting, Wang and Leu (Wang & Leu, 1996) and many other researchers have demonstrated that ANN outperforms ARIMA models. Zhang and Hu (Zhang & Hu, 1998) investigated the capacity of backpropagation neural networks to estimate an exchange rate in 1998. Many features of ANNs make their forecasting useful and attractive. This is due to the fact that, unlike traditional model-based methods, ANNs are data-driven and self-adaptive, with just a few assumptions about the model for research issues (Adhikari & Agrawal, 2013; Kamruzzaman, Begg & Sarker, 2006; Zhang, 2003; Zhang, Patuwo & Hu, 1998). Second, ANNs are common (general). Third, ANNs are universal approximators of functions. ANNs are not linear, ultimately (Huang et al., 2004). There are diverse ANN forecasting models (Adhikari & Agrawal, 2013). Multilayer perceptrons (MLPs), which have only one hidden layer of Feed-Forward Network (FNN), are the most common and prominent among them (Zhang, 2003; Zhang, et al., 1998). There is another version of FNN known as Time Lagged Neural Network (Faraway & Chatfield, 1998; Kihoro, et al., 2006). Hamzacebi in 2008 proposed a Seasonal Artificial Neural Network (SANN) model which is surprisingly simple and has also been proved experimentally accurate and efficient in the estimation of seasonal time series (Hamzacebi, 2008).

Support Vector Machines (SVM) were, at some point, also introduced and implemented to forecast an exchange rate. The key advantage of SVM over ANN is that it can mitigate systemic risks rather than minimize computational risk, as used by neural networks (Kamruzzaman & Sarker, 2004). These (SVM) have often proved more effective than ANNs in today's literature (Kim, 2003; Thissen et al., 2003). Ryll & Seidens (2019), indicated that in all surveyed studies of ANN, only 34% achieve better performance than SVMs. The SVM is regarded as an important technology for obtaining forecast results for the forex market. It can be used both for classification and regression (Pujari et al., 2018). Classification is described as a strategy to categorize data instances (Baradwaj & Pal, 2012). Regression is the values of numeric or continuous attributes (Uysal & Güvenir, 1999). In a case where regression is involved, SVM is called Support vector regression (SVR). The linear regression feature allows a necessary prediction method to minimize error. SVM uses a hyperplane to distinguish data sets, i.e. this algorithm provides two-dimensional plane output as a line separating the classes in question.

A new technique has recently been used to implement financial markets in the study of computational intelligence, called Deep learning. Deep learning is a form of neural network with many layers in which the complexity of the network is profoundly expressed. This Neural Network is popular in Image Processing, Speech Recognition, and Video Processing and is also achieving substantial success in Time Series Forecasting (Chatzis et al., 2018; Handa et al., 2019; Ni et al., 2019). According to Handa et al., (2019), Due to the data's changing behaviour, deep learning algorithms may uncover underlying trends and patterns. Intelligent techniques have become more common in time-series forecasts in recent decades because of their exactness (Handa, et al., 2019). Handa et al. (2019) found a lot of literature on the traditional methods of the neural network. However, very few pieces of literature are available for the application of the Deep Neural Network (Handa et al., 2019). Some of the well-known deep learning techniques are convolution neural networks (CNN), Recurrent Neural Network (RNN), and long short-term memory (LSTMs).

The Recurrent Neural Network (RNN) is a paradigm for neural sequences that generate cutting-edge output on large problems such as language modelling (Mikolov et al., 2010). Regularization is required for good neural network implementations, as we all know. Unfortunately, the most effective approach of regularization for neural Feedforward networks does not perform well with RNNs. The fundamental disadvantage of RNNs in simulation is that they require significantly more connections and resources than standard context networks. RNNs will offer good results due to the rough recurrence of similar patterns present in exchange rate time series. Such regular, although subtle, sequences will provide an accurate forecast (Huang et al., 2004).

LSTMs are used to solve the problem of long-term dependent cells in a generic RNN (Pujari, Sayyed, Shahani & Rupani, 2018). A basic LSTM design is made up of a collection of regularly connected subnetworks known as memory blocks. It is made up of blocks that contain one or more auto-connected memory cells as well as three multiplier modules, input, output, and lost gates for reading, writing, and resetting memory cell operations. The memory cells of the LSTM structure were used to solve the RNN issues (Pujari, Sayyed, Shahani & Rupani, 2018).

2.5 Research Gap

Fischer & Krauss (2018) examined constituent stocks of the S&P 500 for financial market predictions using long short-term memory (LSTM) networks and memory- free classification approaches, i.e., a random forest (RAF), a deep neural net (DNN), and a logistic regression classifier (LOG). The LSTM networks surpassed the memory-free classification methods when predicting out-of-sample directional movements of the S&P 500 (Fischer & Krauss, 2018). For financial time series prediction, Handa, Shrivas, and Hota employed Error Back Propagation Network (EBPN), Deep Neural Network: Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) in 2019. It was discovered that the LSTM outperformed RNN and EBPN (Handa, Shrivas & Hota, 2019).

This study aims to go deeper into the usage of a deep learning model known as the Long Short-Term Memory (LSTM) for forecasting currency rates. This deep learning model was chosen because of its ability to forecast time series data (Ni et al., 2019). When forecasting currency rates of the South African Rand versus the US dollar, the proposed model (LSTM) will be compared to the performance of various time series and machine learning algorithms such as the Support Vector Regression (SVR) and AutoRegressive Integrated Moving Average (ARIMA).

By using the LSTM deep neural network to estimate foreign exchange rates for the USD/ZAR currency pair, this study aims to close the knowledge gap. To the best of our knowledge, there is little or no research that employs the LSTM deep neural network to analyse the USD/ZAR currency exchange.

2.6 Summary

This chapter started by giving some background to forex markets before presenting the factors that affect traders or influence the forex markets. Mainly, economic, political, and traders' psychological factors are dominant. Also, fear makes traders decide badly, along with a lack of discipline.

The chapter then went on to discuss time series, machine learning, and deep learning in forex markets. Prevalent in this discussion was that ARIMA has been tried. However, it is a uniform model built on the basis that it is linear. Machine learning and statistical methods have also been introduced. However, machine learning is considered a "black box" model. Support Vector Machines, in particular, have demonstrated advantages over other models in that they mitigate systemic risks rather than minimize computational risk. Deep learning is the new technology used in financial systems. LSTMs are used to eliminate the long-term dependence problem.

Thereafter, the chapter dwelt on research gap we explore. Precisely, by using the LSTM deep neural network to estimate foreign exchange rates for the USD/ZAR currency pair, this study aims to close the knowledge gap. To the best of our knowledge, there is little or no research that employs the LSTM deep neural network to analyse the USD/ZAR currency exchange. The next chapter shows how the research will be conducted and the methods that are going to be used.

Chapter 3: Methodology

3.1 Introduction

This chapter gives an insight into the research methodology we embrace. It starts by presenting a detailed Research Paradigm that informs our reasoning and argumentation, as well as the research design. The research design of the proposed models forms the bulky part of the chapter, emphasizing the instruments we use for data collection, the samples and the sampling techniques, data collection, recording and reporting, and the data analysis techniques we use.

3.1.1 Overview of the chapter

Figure 3.1 summarizes the overview of this chapter in picture, showing the main sections of emphasis.

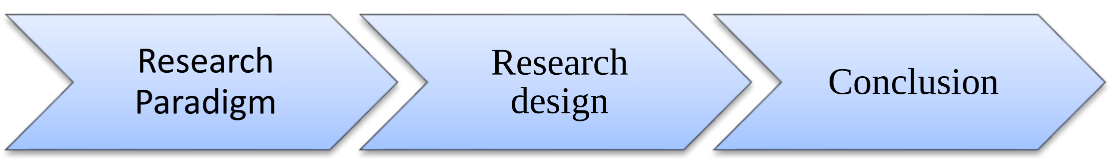

Figure 3.1 Overview of chapter

3.2 Research Paradigm

There are several theoretical frameworks supporting scientific research like the one undertaken in this study. Some authors talk of action research approaches (Ulf & Karin, 2016; Livari & Venable, 2009), while others point to design science research (Bisandu, 2016). In this context, action research aims to address both practical problems of individuals in a difficult situation (Ulf & Karin, 2016) and social scientific goals through collaborative efforts within a mutually accepted ethical framework. (Livari & Venable, 2009). An action researcher is part of the action research during the diagnosis of the problem and during the development of the solution to the problem (Ulf & Karin, 2016). However, action research is more focused on social sciences. Our research project does not fit into the action research context as we are more focused on scientific artifacts.

On the other hand, design science research builds new innovative artifacts to solve some identified organizational problems (Bisandu, 2016). It supports research activities aimed at improving existing artifacts to create new innovative artifacts. Our work seeks to improve existing forex rates forecasting systems by proposing a deep learning model based on the LSTM. The improved artifacts will likely inspire new products in the field. This connotes and purports design science research as the suitable grounding theoretical framework.

In this work, we do not follow the interpretivism paradigm where researchers understand the world as comprising human experiences, where reality is discovered through participant's views and backgrounds (Thanh & Thanh, 2015). This is because we do not make our conclusions based on how people feel or what they say. We solely rely on quantifiable observations extracted from structured scientific experiments, whose data can lead to statistical analyses. The positivism paradigm, therefore, best describes the work presented in this project.

Positivism connotes scientific investigations where predictions follow law-like patterns in which theories and hypotheses are tested (Creswell, 2008). It emphasizes the objectivity of the research process (Creswell, 2008). This is in line with what we seek to achieve in this study, to develop a forex rates prediction model. Experimental tests will be carried out to validate the proposed model. The quality standards which this paradigm produces are validity, reliability, and objectivity, which can be adjusted with the use of triangulation of data if not triangulation of methods and theories (Taylor & Medina, 2011).

An inductive approach purports inductive reasoning (Widodo, 2006), where progression proceeds from observations (Hmedan & Nafi, 2016). We do not subjectively move from observation to ideas. We do not move from observations to generalizations. Rather, scientific methods of arriving at conclusions are followed, pointing to deductive reasoning. Precisely, deductive approaches are concerned with developing hypotheses based on existing theories, and then run experiments to test those hypotheses (Harvey, 2012). Principles, rules, theories, or concepts are presented first, and then their application is tested (Hmedan & Nafi, 2016). Often, deductive reasoning begins with an expected pattern, then a theory leads to a new hypothesis. This new hypothesis is put to test for acceptance or rejection. Our proposed research work focuses on improving existing artifacts, improvements that are then tested for validity in given circumstances. This research work, therefore, falls under the design science theoretical framework, positivist paradigm, with deductive reasoning.

Qualitative research is an all-encompassing method that includes discovery (Williams, 2007). It is an unfolding model that occurs in a natural context, allowing the researcher to acquire a degree of detail via active participation in the actual events (Williams, 2007). The phenomenon being investigated often portrays the participant’s viewpoint (Zubin & Jane, 2014). We indicated that we follow the scientific route based on scientific experiments, where personal viewpoints are excluded. To be more specific, we seek quantitative proofs that may be statistically treated to support or reject our arguments (Williams, 2007).

We seek for causes and effective relationships between variables by using mathematical formulae, graphs, and tables. Data will be obtained from computational simulations related to the performances of the different forex prediction models investigated in this study. In this case, simulation is referred to as running the algorithm.

The following are the steps taken when the design science research theoretical framework is applied:

a) Identify Problem - Identification of an applicable procedure problem requires identifying a flaw in an established system and justifying the importance of seeking a solution to it (Hevner, 2007).
b) Define Objective(s) of a Solution - Objectives from the problem description should be inferred, and what is conceivable, and probable should also be specified. After the measurement point, this target will serve as the criterion until it is judged that the object has been reached to solve the defined problem. When defining the goals, they can be either quantitative or qualitative (Peffers, Tuunanen, Rothenberger & Chatterjee, 2007).
c) Design and Build - It requires going from the study goal to prove that constructing the described objects is feasible. The definition consists of an interpretation of the area examined and the application of applicable scientific and technological expertise. The build refers to the development of the artifact based on the information that illustrates this artifact (Doyle, Sammon & Neville, 2015).
d) Evaluation - The pursuit of efficient artifacts relies on the usage of current forms to obtain the desired outcomes when the regulation is already satisfied in the problem environment (Bisandu, 2016).

e) Communication - Presentation of design science studies must be efficient consultation, both technology-driven and management-driven (Bisandu, 2016).

3.3 Research Design

Bai Kaimin et al. (2016) demonstrated that a single hidden neural network layer and a deep learning model are predictable for the stock market. They demonstrated that a deeper learning model is more predictable than a single hidden neural network layer (Qu & Zhao, 2019). However, one disadvantage of neural networks is their susceptibility to overfitting (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). However, the dropout regularization is used to prevent overfitting and co-dependence between neurons (Hinton, Srivastava, Krizhevsky, Sutskever & Salakhutdinov, 2012).

The rectified linear unit (relu) activation function: relu(x) = max(0, x) is the most used function in training deep learning models. Relu activations improve gradient flow, reduce the training time (Glorot et al. 2011), and have become the state of the art in deep learning (LeCun, Bengio & Hinton, 2015; Clevert, Unterthiner & Hochreiter, 2016; Ramachandran, Zoph & Le, 2017). The tuning of hyperparameters is usually carried out by analytical experiments, which require high measurement costs due to the broad variety of hyperparameter settings (Dautel, Härdle, Lessmann & Seow, 2020). We embrace all these features when we administer our experiments as well.

A random search for hyperparameter tuning is the most used. The following hyperparameters are the ones to commonly consider when tuning:

a) Number of hidden layers: 1, 2, 3, 4.
b) Number of neurons per hidden layer: 25, 50, 100, 200, 400, 800, 1600.

c) Dropout: 0 to 60%, in steps of 10%.

d) Optimizer and learning rate: Adam and RMSprop with various learning rates.

e) Batch size: 16, 32, 64, 128, 256

f) Epochs

In 2020, Dautel, Härdle, Lessmann & Seow experimented with the following currency pairs: EUR/USD, GBUP/USD, USD/JPY, and USD/CHF for financial time series forecasting. The experiment involved the following deep learning models: Gated recurrent units (GRU), Long-short term memory (LSTM), Feedforward neural networks (FNN), and Structural Recurrent Neural Network (SRNN). The models were trained with 32-sample minibatch sizes and the Adam optimizer with default parameters for a total of 100 epochs, with early halting after 10 periods due to no improvement in validation loss. Validation training was held for around 20% of the training set. Three hidden layers, each with 50 neurons, and dropout layers, each with a 25% dropout rate after each hidden layer (Dautel, Härdle, Lessmann & Seow, 2020) were used. Their results and conclusion indicated that the LSTM model had higher predictive accuracy than GRU, FNN, and SRNN when forecasting financial time series. We are inspired by these results which further motivated us to verify these outcomes under reproduced conditions.

3.3.1 Instruments

The study will make use of the following techniques: ARIMA, SVM, and LSTM. It seeks to evaluate the performance of the techniques to find the most efficient technique for forecasting the USD/ZAR currency pair. We discuss the implementation procedures to follow in each case.

a) Autoregressive integrated moving average (ARIMA)

ARIMA is a hybrid of autoregressive (AR) with moving average (MA) methods. The mathematical formulation of the ARIMA ($\boldsymbol{p, d}$ **and** $\boldsymbol{q}$) model using lag polynomials is given below (Adhikari & Agrawal, 2013; Hipel & McLeod, 1994).

$$\left(1 - \sum_{i=1}^{P} \varphi_i L^i\right)(1 - L)^d y_t = \left(1 + \sum_{j=1}^{q} \theta_j L^j\right)\varepsilon_t$$

Where $\boldsymbol{L}$ is the lag operator, $\boldsymbol{\theta_i}$ are the parameters of the moving average part and $\boldsymbol{\varepsilon_t}$ are the error terms. $\boldsymbol{p, d}$ **and** $\boldsymbol{q}$ are integers greater than or equal to zero (Adhikari & Agrawal, 2013).

ARIMA is a regression analysis approach that assesses the intensity of one dependent variable in relation to other transition variables. It is, however, a generic univariate model that assumes that time series are linear (Kamruzzaman & Sarker, 2004; Adebiyi, Adewumi & Ayo, 2014. The goal of the model is to anticipate possible securities or financial market developments by evaluating differences in series values rather than real values. (Kamruzzaman & Sarker, 2004; Adebiyi, Adewumi & Ayo, 2014).

To implement ARIMA, we will use the following steps:

a) Load the dataset that will be used.
b) After loading, the data should be pre-processed. Pre-processing involves the conversion of data types, creating new or deleting features, checking for missing numbers, and making the data univariate.
c) The data is then checked for stationarity. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) and Augmented Dickey Fuller Test (ADF) are used

to confirm stationarity. If both p-values are > 0.05, the series is not stationary.

d) Apply differencing to make the data stationary. Determine the d value that will make the data stationary.

e) Check for seasonality, trend, observed and residual by applying decomposition. If there is seasonality, Seasonal ARIMA or SARIMA is implemented in step 7. If there is no seasonality, we use ARIMA.

f) Create ACF and PACF plots to determine MA order (q) and the AR order (p).

g) The values d, p, and q are used as input parameters when fitting the ARIMA model. Before fitting the model, the dataset should be split into training and testing.

h) After fitting and training the model, the validation or testing dataset is used to predict future values.

i) The predicted and actual values are compared, and their error is measured using RMSE, MSE, and MAE.

b) Support Vector Regression (SVR)

Smola (1996) proposed a variant of SVM for regression. The proposed model is called Support Vector Regression (SVR) (Drucker, Burges, Kaufman, Smola & Vapnik,1997). In Support vector regression, the objective is to find the loss function that does ignores the errors that are positioned at a certain distance from the actual data points values (Berwick, 2003). The following are important component units of SVM:

i. Kernel

The Radial basis function (RBF) kernel will be used in this research. The data will be transformed from low to high dimensions using the RBF. The RBF kernel is defined as:

$$K(x, \acute{x}) = \exp\left(-\frac{\|x - \acute{x}\|^2}{2\sigma^2}\right)$$

Where $\|x - \acute{x}\|^2$ is the Euclidean distance between two input feature vectors and σ^2 is a free parameter.

ii. Hyperplane

A hyperplane is a line that divides two classes into different sections. A hyperplane can be written as $w^T x + b = 0$, where w is a weight vector, x is an input vector and *b* is the bias (Berwick, 2003). Finding the most optimal hyperplane requires an optimization background.

iii. Boundary line

The boundary lines are established as two parallel lines with error threshold values drawn on both sides of the Support Vector, epsilon (ε). These lines define the boundaries between data points.

iv. Support Vectors

Support vectors guide which points are nearest to the limit. Points difference is medium or less. In Support vector regression we use these two constraints : $y_i - (wx_i) + b \leq \varepsilon$ and $(wx_i) + b - y_i \leq \varepsilon$. The two constraints are used to allow some deviation ε between the target (the value we trying to predict) y_i, and the function $f(x) = wx + b$ which models the data. The main objective is to minimize over-complexity $\|w\|^2$.

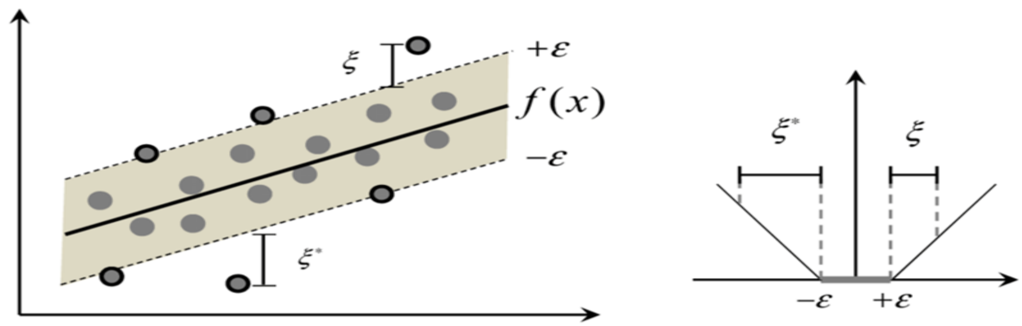

Figure 3.2 Support vector regression (SVR) and the intensive loss function, source: Researchgate.net

The underlying principle behind SVR is to find the right fit. In SVR, the hyperplane with the largest number of points is the best suited (Berwick, 2003).

To implement the SVR model, we will use the following steps:

a) Load the dataset that will be used.
b) After loading, the dataset should be pre-processed and explored. Pre-processing involves the conversion of data types and creating new or deleting features.
c) Feature scaling is the 3rd step, it is done to normalize the dataset.
d) The dataset should be split into training and testing.
e) When fitting the SVR model it is important to specify the kernel. There are different kernels to choose from such as linear, RBF, and Gaussian, their application varies with the problem being solved.
f) After fitting and training the model, the validation or testing dataset is used to predict future values.
g) The predicted and actual values are compared and their error is measured using RMSE, MSE, and MAE.

c) Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture that is used in the field of deep learning (Hochreiter & Schmidhuber, 1997). A typical LSTM unit consists of a cell, an input gate, an output gate, and an oblivion gate. The cell remembers values over arbitrary periods, and the three gates control information flow into and out of the cell (Nagpure, 2019). Figure 3.3 illustrates this scenario in pictures.

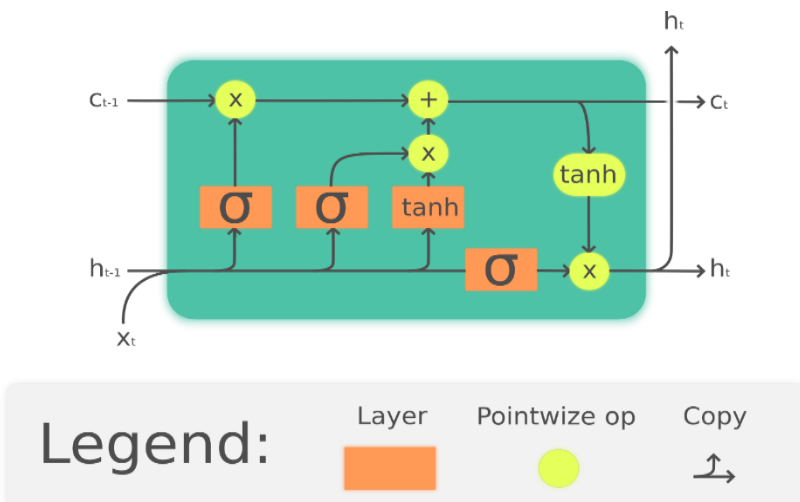

Figure 3.3 Figure of Long short-term memory, source: Wikipedia.com

The equations for a forward pass of an LSTM are as follows:

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right)$$

$$O_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = O_t \circ \sigma_h(C_t)$$

Where $\sigma_g$is the sigmoid function, σ_h is the hyperbolic tangent function, $\circ$ denotes the Hadamard product, W, U, b are weight matrices and bias vector parameters, and x_t is the input vector. Additionally, f_t is the forget gate's vector, i_t would be the input gate's vector, O_t would be the output gate's vector, h_t would be the hidden state vector and c_t is the cell state vector.

The LSTM model in this investigation will use the following hyperparameters:

Activation function	Relu
Optimizer	Adam optimizer. The Adam optimizer uses backpropagation to train weights of the neural network.
Number of hidden layers	1, 2, 3, 4
Number of neurons per hidden layer	25, 50, 100,150, 200, 400, 800, 1600.But we will also experiment with other values.
Dropout:	0 to 60%, in steps of 10%.
Batch size	16, 32, 64, 128, 256
Sliding window or lookback	30,60 and 90 days
Dense layers	1,2,3,4
Dense layer neurons	1,2,3,4,5,6
Epochs	50, 100, 150, 200

Table 3.1 Model hyperparameters

To implement the LSTM, we will use the following steps:

a) Load the dataset that will be used.

b) After loading, the dataset should be pre-processed and explored. Pre-processing involves the conversion of data types and creating new or deleting features.
c) Feature scaling is the 3rd step, it is done to normalize the dataset.
d) The dataset should be split into training and testing.
e) Create a look back function and select the size of the sliding window.
f) Convert features into NumPy array and reshape the array into a shape accepted by the LSTM model.
g) Compile the LSTM model and fit it.
h) After fitting and training the model, the validation or testing dataset is used to predict future values.
i) The predicted and actual values are compared, and their error is measured using RMSE, MSE, and MAE.

3.3.2 Data, Samples, and sampling techniques

The data used in this study is structured as follows: date, time, open, low, high, close, and volume. These features except date and time are of float data type whereas date and time are of data time. In this study, we are interested in predicting the "close" which is the closing price of the exchange rate. A detailed explanation of all the data features will be given in 3.3.3

No sampling techniques will be used when we make use of this data. This study will make use of the sequential flow of data since we are dealing with time-series data.

3.3.3 Data collection

The data came from the MetaTrader API, which is a trading platform. Traders utilize MetaTrader as their trading platform. Historical data on currencies, commodities, and indexes may be seen on the site. The data for this investigation was gathered throughout a 24-hour period. Date, time, open, low, high, close, and volume are all included in this feature set.

In this context, the date refers to a particular day in the market, on the other hand, time refers to the time of the day. Open is the initial exchange rate that a currency pair experienced at the beginning of the day, On the contrary, low refers to the lowest exchange rate a currency pair experienced, while high points to the highest exchange rate a currency pair experienced. In this case, close indicates the closing exchange rate the currency pair closed at while volume connotes the number of trades in a given period. Table 3.2 summarizes the dataset used in this study.

Currency pair	Number of observations (Rows) in dataset	Date
USDZAR	3440	2006.01.16-2020.10.23

Table 3.2 Data used in the study

3.3.4 Recording and reporting

A dell XPS laptop with 16GB and i7-10510U (8MB cache, up to 4.9GHZ,4 cores) processor will be used. Python programming will be the IDE of choice. The following libraries will be key in this study: Numpy, pandas, seaborn, Keras, Matplotlib, and sklearn.

a) Data pre-processing

The dataset will be cleaned to remove any unwanted data and if there is missing data, a compensation method will be used to make up for lost data.

It will also be checked for redundancy. The data will be scaled using the minmax scaling method defined in the following formulae after pre-processing.

$$X_{\_std} = \frac{(X - X.min(Axis = 0))}{X.max(Axis = 0) - X.max(Axis = 0)}$$

$$X\_scaled = X\_std * (max - min) + min$$

Where, min, max = range of features. The data collected will be divided into training and testing datasets with a 70: 30 percent ratio for training and testing purposes, respectively. The following procedure summarizes data pre-processing: The USD/ZAR data will be loaded using pandas package, it will then be checked for missing values and whether it has the desired datatypes or not. The data will also be scaled so that the values are within the desired range, this will yield faster computation and better results.

b) Performance measure or benchmark criteria

These are the approaches that will be used to evaluate the prediction models under investigation. Performance measures or indicators are metrics used to evaluate a system's quantitative performance (Cuenin,1987). Two metrics are of interest as follows:

i. The mean squared error (MSE)

The average of squared errors for least squares regression using the loss function is called the mean squared error (Lehmann, 2006). MSE is calculated using the formula below.

$$MSE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}(x_i - \overline{x}_i)^2$$

Where *n* is the data points of all variables, *x* is the vector of observed values of the variable being predicted, with $\bar{x}$ being the predicted values.

ii. Mean absolute error (MAE)

The MAE is a method for calculating errors based on several measurements of the same phenomenon (Willmott & Matsuura, 2005). The formula below shows how to calculate it.

$$MAE = \left(\frac{\sum_{i=1}^{n} |y_i - x_i|}{n}\right)$$

Where y_i is the predicted value and x_i is the true value. The value of n is the same as in the MSE formula.

To measure MSE from each model, we subtract the squared predicted values from the squared actual values and then divide by the total number of data points. Similarly, MAE is measured by subtracting the actual value from the predicted value and then divide by the total number of data points.

3.3.5 Analysis techniques

This is a comprehensive analysis of the results, focused on the extraction of central tendencies, variability, correlation, and explanations to the inferential observations thereto. It is a comprehensive analysis of the forecasts and systemic compilation of data (Johnson, 2004).

The common view is that quantitative scholars follow a methodological model of 'positivism' (Guba & Lincoln, 1994). The investigator and truth (study) are distinct in positivism (Onwuegbuzie, 2002).

The testing methods of hypothesis shall include deductive reasoning and shall start with specified hypotheses and determine if theories apply (De Waal, 2001; Hyde, 2000; Woiceshyn & Daellenbach, 2018).

The null hypothesis of this study is that the LSTM model's performances are not significantly different from that of ARIMA and SVR when forecasting the USD/ZAR exchange rate. This is statistically denoted as Ho: $\mu 1 = \mu 2$, where μ is the average performance of each model. Two alternate hypotheses emanate as follows: H1: $\mu 1 > \mu 2$ and H2: $\mu 1 < \mu 2$.

Precisely, positivists assert that science entails confirmation or falsification of hypotheses, and that hypothesis testing processes should be conducted objectively (Onwuegbuzie, 2002). Before certain assumptions are disproved or accepted, the performance of the LSTM model will be compared to that of the ARIMA and SVR models.

3.4 Ethics

Ethics simply refers to the study of morality. Deciding how to act from a legal or ethical point of view in a particular situation is not a straightforward matter (Oliver, 2010).

According to Neuman (2014), there are guiding principles that point out that ethical study considerations provide for legitimate and moral methods of analysis. The following ethical principles advise the research carried out:

3.4.1 Permission to conduct the research study

The researcher obtained permission from relevant authorities to collect data from the participant (Metatrader) before conducting the study.

3.4.2 Informed consent

Marshall, Adebamowo, Adeyemo, Ogundiran, Strenski, Zhou, & Rotimi (2014), define informed consent as a voluntary agreement to participate in the research. The researcher gave a permission form that stated the study's purpose and aims, as well as the fact that participation was optional and that participants might withdraw from the study at any moment if they so desired.

3.4.3 Voluntary participation

Khawula (2016), notes that participation in research should be optional and that participants should fail to share such details and should have the right to withdraw from the research at any time. The researcher notified Metatrader that participation was solely voluntary. Metatrader was free to choose, refuse or withdraw from the study at any point of the process.

3.4.4 Confidentiality

Lyons & Coyle (2015), state that confidentiality is closely tied up with informed consent. Metatrader was made aware that their material will be kept confidential. The researcher treated the information provided by Metatrader as confidential as possible. He continued to take care not to breach or compromise confidentiality. The study complied with this particular ethical requirement by ensuring that the information supplied by Metatrader was securely preserved.

3.4.5 No harm to participants

Bryman (2016), states that the research process should not harm or stress the research participants in any way. The researcher made sure that the participants in this study were not in any danger.

Participation was essential to achieving scientifically and socially important aims. The benefit of incorporating ethical practice is that it helps to avoid malpractices in the study.

3.5 Summary

This chapter presented the methodology and the theoretical framework underpinning this study. It started by characterizing the paradigm of the study, precisely pointing to a design science research, emphasizing the objectivity of the research process. The quality standards produced in this paradigm are validity, reliability, and objectivity, all adjusted through triangulation. The study follows a deductive reasoning approach, leading to quantitative proofs that can be subjected to statistical analyses.

The chapter also presented the research design in which the rectified linear unit (relu) activation function: relu(x) = max(0, x) was assumed, where a random search for hyperparameter tuning was used. The study used ARIMA, SVM, and LSTM and sought to compare the performance of these techniques. Details on how the ARIMA model works were given. Implementation of the support vector regression model was described, along with the steps for implementing the LSTM model. Mainly, the hyperparameters used by the LSTM model were described in detail, together with the algorithm thereto.

The data used in this study was characterized. No sampling techniques are used. The dataset was explained, and related data reporting methods were stated. In addition, pre-processing procedures were explained, along with the performance measures we extract. Precisely, the MSE and the MAE are of interest.

These are the quantities that are further analyzed in the next chapter. The main analysis will include the central tendencies, variability, correlation, and explanations to the inferential observations thereto. Such analyses would provide evidence for accepting or refuting the null hypothesis of this study that the LSTM model’s performances are not significantly different from that of ARIMA and SVR when forecasting the USD/ZAR exchange rate. Ethical issues were discussed last.

Chapter 4: Analysis of Results

4.1 Introduction

This chapter administers the experiments from which we extract results that form the basis for accepting or rejecting our null hypothesis which states that the LSTM model's performances are not significantly different from those of the ARIMA and SVR model when forecasting the USD/ZAR exchange rate (Ho: µ1 = µ2). The same data would be useful in the selection of an alternate hypothesis between H1: µ1 > µ2 and H2: µ1 < µ2, should that be necessary. The chapter will give a detailed explanation of how the experiment was conducted and how the results were achieved. The chapter presents descriptive statistics, tests for normality, correlations, and inferential statistics from the outcomes of the experiments. These statistics form the basis for the discussions and conclusions drawn in chapter 5.

The hypotheses that this study seeks to test, accept or reject are presented here. The main data collected in the study are the MSE and MAE error rates of the ARIMA, SVR, and LSTM models. The model with the lowest MSE and MAE error rates is desirably deemed as better. If μ_1 denotes the average performance of the LSTM model, while μ_2 and μ_3 indicate the average performances of the ARIMA and SVR models respectively, then the following three hypotheses are tested in this study.

- $H_0: \mu_1 = \mu_2 \ or \ \ \mu_3 \rightarrow$ There is no significant difference between the models.
- $H_1: \mu_3 \ or \ \mu_2 > \ \mu_1 \ \rightarrow$ Other models are better than the LSTM model
- $H_2 : \mu_3 \ or \ \mu_2 \ < \ \mu_1 \rightarrow$ Other models are weaker than the LSTM model

One and only one of the three hypotheses will be accepted at a time.

4.1.1 Overview of the chapter

Figure 4.1 summarizes the overview of the rest of this chapter, pointing to the administration of three experiments from which MSE and MAE are reported from each model. The findings thereto are discussed last, before we conclude the chapter in section 4.6.

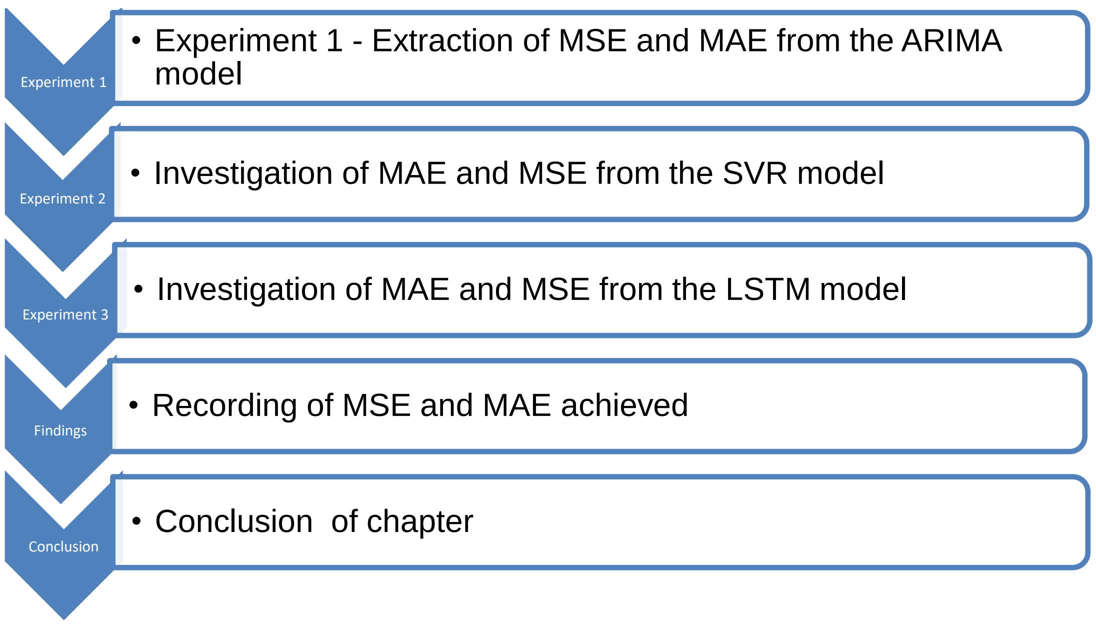

Figure 4.1 Diagram showing overview of chapter

4.2 Experiment 1 – Extraction of MSE and MAE - ARIMA

This experiment measures the MSE and MAE achieved when the ARIMA model was used. MSE is a critical metric for evaluating the predictor or estimator's quality. Equally, MAE also assesses the model's quality of the predictor or estimator.

The null hypothesis for this particular experiment is Ho: the ARIMA model's MSE and MAE are not significantly different from other models' MSE and MAE.

Statistically, the null hypothesis can be stated as H_0: $\mu_1 = \mu_2$, where μ indicates the average MSE or MAE of a model. In this case, μ_1 represents the average MSE / MAE of the ARIMA model, while μ_2 indicates the average MSE / MAE of either the SVR or LSTM models depending on which model the ARIMA is being compared to at the time.

Two alternate hypotheses arise. The first alternate hypothesis would be directionally stated as H_1: the ARIMA model outperforms other models. This is statistically denoted as H_1: $\mu_1 > \mu_2$, where μ_1 is the MSE / MAE of the ARIMA model, while μ_2 is the MSE / MAE of other models, depending on which model is under scrutiny at the time. The second alternate hypothesis would also be directionally stated as H_2: the ARIMA model underperforms than other models, statistically denoted as H_2: $\mu_1 < \mu_2$, where μ_1 is the MSE / MAE of the ARIMA model, while μ_2 is the MSE / MAE of other models, depending on which model is under scrutiny at the time.

Two dependent variables are apparently of interest (MSE and MAE). A dependent variable is that variable which we investigate and measure in an experiment. There are two key independent variables worth mentioning: namely time and base currency. An independent variable is that variable which we alter or manipulate while we keep track of the dependent variable. All the other variables involved in this experiment are controlled, including the forex currencies under investigation, machine specifications, and any other code parameters. Below is the setup of this experiment.

Title: investigation of MSE and MAE from the ARIMA model.

Null-Hypothesis: $H_0 : \mu_1 = \mu_2$ – *the ARIMA model's performances are not significantly different from those of the other models (SVR or LSTM)*

Alternate Hypothesis: *$H_1 : \mu_1 > \mu_2$ –the ARIMA model outperforms the other models*

Alternate Hypothesis: *$H_2 : \mu_1 < \mu_2$ – the ARIMA model underperforms than the other models*

Dependent variables: MSE and MAE

Independent variables: Time, base currency.

Controlled variables: Environment, machine specifications, parameters.

Procedure: The data was loaded, cleaned, and explored. ACF and PACF graphs were plotted to see if the data is stationary or not. All possible model parameters were estimated, and their AIC values were calculated. The lowest AIC value parameters were then fitted into the Arima model.

Algorithm 4.1:

```
1.Plot(dataset)
        while(graph is non-stationary = True)
                {smoothen graph to make it stationary}

2. ACF/PACF(Stationary Graph)

3. Estimate all possible model parameters

4. Calculate AIC values of all model parameters

5. plot (residual of model parameters)
        if(residual graph with no lag)
                {Forecast Dataset with these parameters}
        else choose other estimated model parameters and repeat step 3
```

Figure 4.2 ARIMA algorithm Source: SlideShare Link: https://www.slideshare.net/amrinderarora/arima-forecasting-presentation-by-sera-cresta-nora-alosaimi-and-puneet-mahana

Data collection: Table 4.1 summarizes the data collected.

Replication	MSE	MAE
1	0.022	0.11532114935607636

	2	0.022	0.11532114935607636
	3	0.022	0.11532114935607636
	4	0.022	0.11532114935607636
	5	0.022	0.11532114935607636
	6	0.022	0.11532114935607636
	7	0.022	0.11532114935607636
	8	0.022	0.11532114935607636
	9	0.022	0.11532114935607636
	10	0.022	0.11532114935607636

In [1]:
```
import pandas as pd
from pandas import DataFrame
from statsmodels.tsa.arima_model import ARIMA
from matplotlib import pyplot as plt
```

1. Loading of data

In [2]:
```
df = pd.read_csv("./Forex_data/USDZARNov.csv",delimiter=',', parse_dates=[0], header=0,skiprows=0)
```

In [3]:
```
df.columns=['date','time','open','high','low','close','volume']
df.tail(2)
```

Out[3]:

	date	time	open	high	low	close	volume
3438	2020-10-22	00:00	16.3172	16.3919	16.1917	16.2020	76669
3439	2020-10-23	00:00	16.1993	16.2399	16.1968	16.2183	3767

In [4]:
```
df=df.drop(['time'],axis=1)
```

In [5]:
```
df['date'] = pd.to_datetime(df['date'], infer_datetime_format=True)
df.set_index('date', inplace=True)
df = df.astype(float)
df.tail()
```

Out[5]:

	open	high	low	close	volume
date					
2020-10-19	16.5341	16.5453	16.4167	16.5068	43308.0
2020-10-20	16.5078	16.5689	16.4140	16.4684	63185.0
2020-10-21	16.4739	16.4784	16.2498	16.3178	70204.0
2020-10-22	16.3172	16.3919	16.1917	16.2020	76669.0
2020-10-23	16.1993	16.2399	16.1968	16.2183	3767.0

Figure 4.3 Loading of USDZAR data

Figure 4.3 shows the packages used and how the USD/ZAR data was loaded into the system using the pandas package. The data had missing column names. The missing column names had to be added. The “time” column had to be dropped because it was static or always 00;00. The date column was indexed because we wanted to forecast the closing price using the daily timeframe.

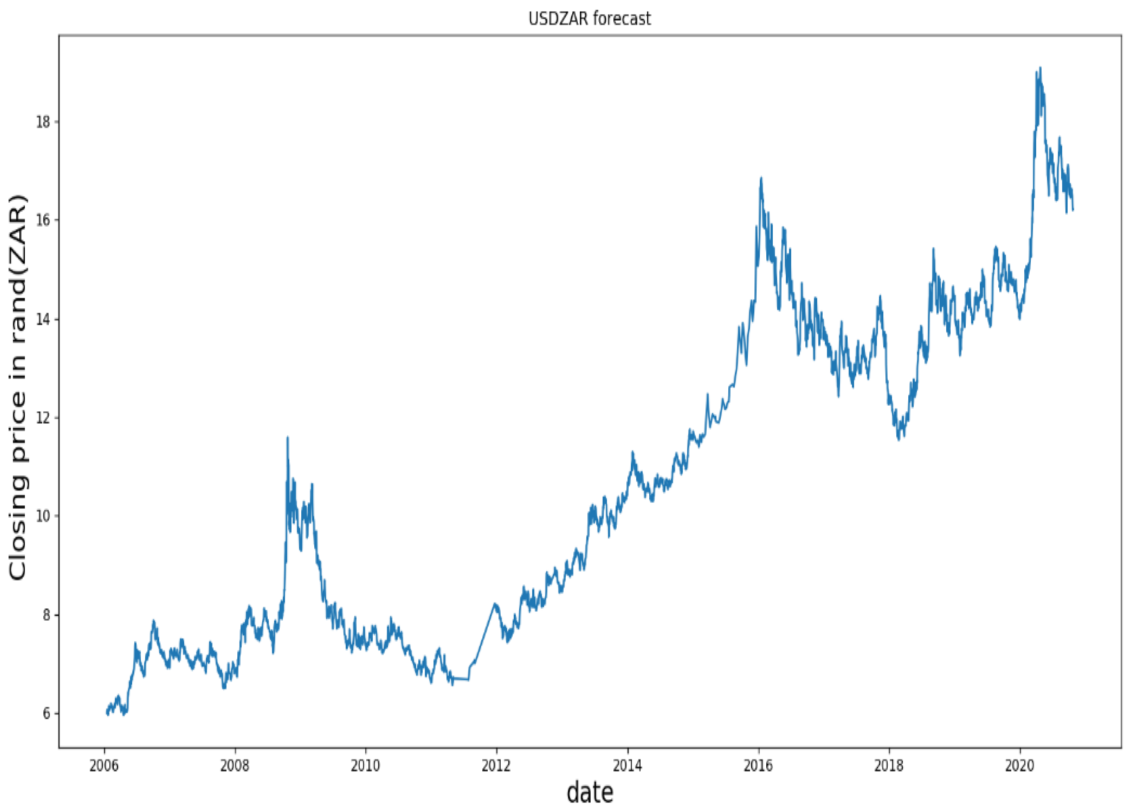

Figure 4.4 Visualization of USDZAR currency pair

The closing price of the data was visualized as shown in figure 4.4. Visualization allows us to understand the data better, possibly elucidating trends and other patterns. In this case, the USDZAR currency pair experienced an increase in the exchange rate.

The ARIMA model is univariate, as we mentioned. As a result, we primarily focused on the USDZAR currency pair's closing price.

The closing price of the USDZAR currency pair was examined for stationarity. It's crucial to check for stationarity in order to get reliable findings. The ARIMA model requires data to be stationarized. When parameters like mean, variance, and covariance do not fluctuate over time, the data is said to be stationary. For statistical testing, we employed the ADF (Augmented Dickey-Fuller) and KPSS (Kwiatkowski-Phillips-Schmidt-Shin) techniques. The stationarity of the closing price was determined using these approaches. Except for the technology underlying them, the goal of these two statistical procedures is the same. KPSS removes the pattern to make the data stable, whereas ADF employs differencing to transform the dataset. Both tests should be used to ensure the intended outcomes. It is crucial to note, however, that both statistical tests contradict each other. The ADF specifies that the dataset is NOT stationary, but the KPSS specifies that it is stationary. We can reject the null hypothesis in favor of the alternative that the series is stationary if the 'test statistic' is smaller than the 'critical value'. Table 4.1 summarizes the findings in this regard.

	ADF test	KPSS test
t-statistics	-0.960	7.865
p-value	0.768	0.010

Table 3.1 stationary statistics test

Both p-values are greater than 0.05 (the critical value). This is an indication that the data is not stationary. Since the results achieved by both the ADF and KPSS tests indicated non-stationarity, we had to decompose the data and re-visualize the closing price. The implementation code for decomposing the results is shown in figure 4.5 and the results of the decomposition are shown in figure 4.6.

How to Choose Between Additive and Multiplicative

The additive model is useful when the seasonal variation is relatively constant over time.

The multiplicative model is useful when the seasonal variation increases over time.

```
result = seasonal_decompose(df['close'], model='multiplicative',period=365)
plt.figure(figsize=(16,8))
result.plot()
plt.figure(figsize=(16,8))
#plt.savefig('usdzardailytrend.png')
plt.show()
```

```
<Figure size 1152x576 with 0 Axes>
```

Figure 4.5 snapshot code used for seasonal decomposition

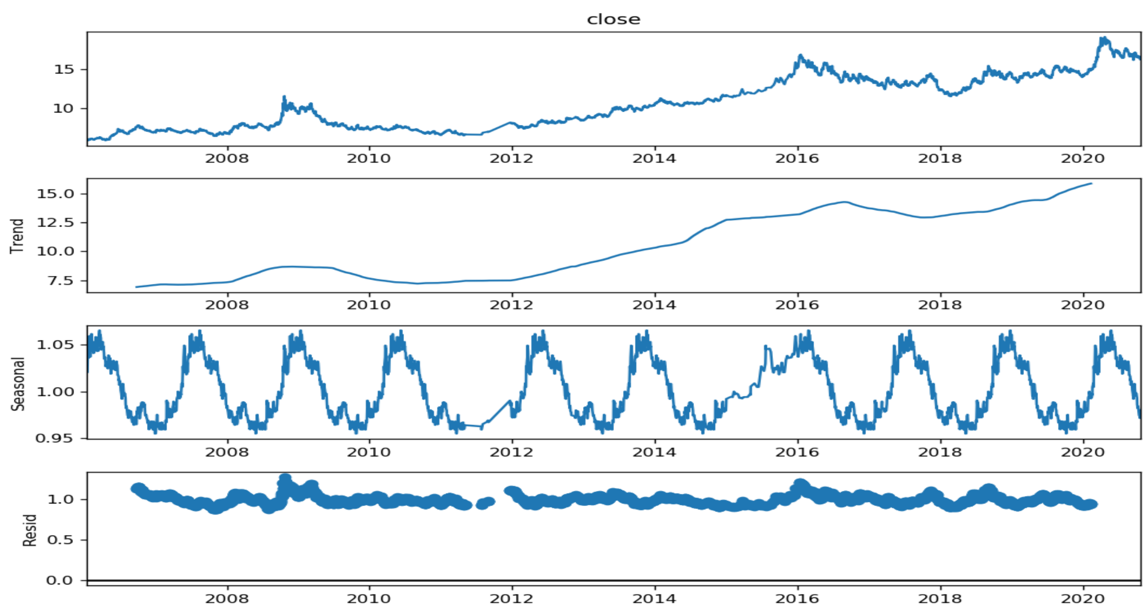

Figure 4.6 Seasonal decomposition results

The multiplicative model was chosen for the closing price because closing prices were increasing with time. The 365 periods was chosen because we are using a daily timeframe over a year of 365 days. The closing price data was then made stationery using a differencing (d) shift of 1 and 2.

```
In [15]:
import itertools
p = range(1, 4)
d = range(1, 2)
q = range(1, 3)
pdq = list(itertools.product(p, d, q))
aics = []
params = []
for param in pdq:
    model = ARIMA(df['close'], order=param)
    model_fit = model.fit()
    aic = model_fit.aic
    aics.append(aic)
    params.append(param)
combo = list(zip(aics, params))
combo.sort()
combo_array = np.array(combo)
print(combo_array)
```

```
[[-4129.923979480849 (1, 1, 2)]
 [-4129.652627375843 (2, 1, 1)]
 [-4127.957247127208 (2, 1, 2)]
 [-4125.892447784176 (3, 1, 2)]
 [-4116.067315728427 (3, 1, 1)]
 [-4111.475454623573 (1, 1, 1)]]
```

Figure 4.7 snapshot code of grid search iteration

The three primary parameters are important: p, q, and d. The order of the AR term in this experiment is p, the order of the MA term is q, and the order of differencing necessary to keep the time series stable is d. To determine the values of p, q and d we used a grid search as shown in figure 4.7. The Akaike Information Criterion (AIC) estimator of the relative quality of statistical models was utilized as a performance criterion.

According to the outcomes reported, the lowest AIC is -4129.923979481297, whose parameters are (1,1,2). These are the parameters that were then utilized in the ARIMA model. A MSE = 0.022 and MAE = 0.11532114690886489 were achieved. These two results will be compared to the results obtained using the other two models (SVR and LSTM).

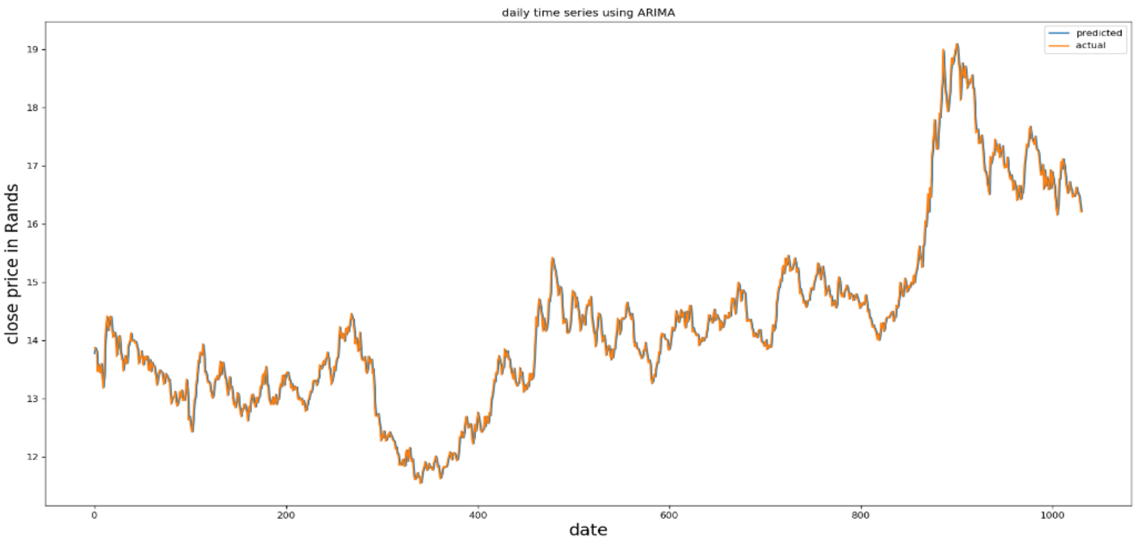

Figure 4.8 ARIMA USDZAR forecast

Figure 4.8 shows the forecast of the predicted and actual closing prices.

4.3 Experiment 2 – Extraction of MSE and MAE - SVR model

This experiment similarly measures the MSE and MAE, but is now achieved when the SVR model was used. The null hypothesis, in this case, is Ho: the SVR model's MSE and MAE are not significantly different from that of the other models. Statistically, H_0: $\mu_1 = \mu_2$, where μ_1 represents the average MSE / MAE of the SVR model, while μ_2 indicates the average MSE / MAE of either the ARIMA or LSTM models. Similarly, two alternate hypotheses arise, where H1: the SVR model outperforms the other models (H_1: $\mu_1 > \mu_2$) and H_2: the SVR model underperforms (H_2: $\mu_1 < \mu_2$).

Equally, the same dependent variables are of interest (MSE and MAE). Two independent variables are also manipulated (and base currency). All controlled variables are the same (forex, system specifications, and parameters). The experiment setup is summarized below.

Title: investigation of MSE and MAE from the SVR model.

Null-Hypothesis: $H_0 : \mu_1 = \mu_2$ – *the SVR model's performances are not significantly different from those of the other models (ARIMA/LSTM)*

Alternate Hypothesis:$H_1 : \mu_1 > \mu_2$–*the SVR model outperforms other models*

Alternate Hypothesis: $H_2 : \mu_1 < \mu_2$ – *the SVR model underperforms*

Dependent variables: MSE and MAE

Independent variables: Time, base currency.

Controlled variables: Environment, machine specifications, parameters.

Procedure: Load the dataset that will be used. After loading, the dataset should be pre-processed and explored. Pre-processing involves the conversion of data types and creating new or deleting of features. Feature scaling is the 3rd step, it is done to normalize the dataset. The dataset should be split into training and testing. When fitting the SVR model it is important to specify the kernel. There are different kernels to choose from such as linear, RBF, and Gaussian, their application varies with the problem being solved. After fitting and training the model, the validation or testing dataset is used to predict future values. The predicted and actual values are compared, and their error is measured using RMSE, MSE, and MAE.

Algorithm 4.2:

1. Load the dataset
2. Scale the dataset
3. Split into train and test set
4. Fit the model

5. Measure the accuracy
6. Tune the parameters
7. Repeat the experiment

Data collection: Table 4.2 summarizes the data collected.

Replication	MSE	MAE
1	0.02007424794650156	0.12298385946739537
2	0.02007424794650156	0.12298385946739537
3	0.02007424794650156	0.12298385946739537
4	0.02007424794650156	0.12298385946739537
5	0.02007424794650156	0.12298385946739537
6	0.02007424794650156	0.12298385946739537
7	0.02007424794650156	0.12298385946739537
8	0.02007424794650156	0.12298385946739537
9	0.02007424794650156	0.12298385946739537
10	0.02007424794650156	0.12298385946739537

Below are the packages that were used when developing the SVR model.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
from subprocess import check_output
```

Figure 4.9 Packages used when creating the SVR model

Run
Code

Reshaping of the closing price

In [26]:
```
Y=df['close'].values.reshape(-1,1)
```

Scaling and transforming of data

In [10]:
```
sc_X = StandardScaler()
sc_y = StandardScaler()
X = sc_X.fit_transform(X)
Y = sc_y.fit_transform(Y)
```

Split of data into training and testing

In [11]:
```
train_size = int(df.shape[0] * 0.70)
trainX = X[:train_size]
trainY = Y[:train_size]
testX = X[train_size:]
testY = Y[train_size:]
```

Figure 4.10 Snapshot showing reshaping and splitting of data

The data was explored the same way as in Figures 4.4 and 4.4. Unlike ARIMA model, SVR is multivariate. All features (independent data set (X)) are utilized when implementing the SVR model to predict the target (dependent variable(Y)) which is the closing price.

The target feature in this research was “close” which is the closing price of the USD/ZAR currency pair. Values in all columns in the dataset were converted to float values.

The reason for this was to avoid having a string or integer values because we are dealing with financial data which is usually in float data type. The data were then scaled and transformed as shown in figure 4.10 using “scaler.fit_transform()” function. The reason for this was to normalize, transform the data and to help the algorithm perform better and fast.

The dataset was split into 70% training and 30% testing. The SVR model was built using the following parameters:

Building SVR model

```
SupportVectorRegModel= SVR (C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1,
    gamma='auto', kernel='linear', max_iter=-1, shrinking=True,
    tol=0.001, verbose=False)
SupportVectorRegModel.fit(trainX,trainY)
```

```
C:\Users\Andisani\AppData\Roaming\Python\Python36\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(**kwargs)
```

```
12]: SVR(gamma='auto', kernel='linear')
```

Figure 4.11 parameters used when building SVR model

The kernel used is linear. The reason for using such a kernel is because this is a linear regression problem. Moreover, other kernels (polynomial and radial basic function) did not perform well, the linear regression achieved better MSE and MAE compared to them. The study explored various values for all the parameters in figure 4.11. However, the optimal values for the parameters are the ones shown in figure 4.11.

The model was fed two train datasets, one with independent variables (open, high, low, and volume) and one with a dependent variable (close), which is the closing price.

Following training, the model was given a test dataset containing independent variables (open, high, low, and volume). The predict function was given the test dataset with the independent variable. Based on what it learnt during training the model, the function forecasts the target variable (closing price).

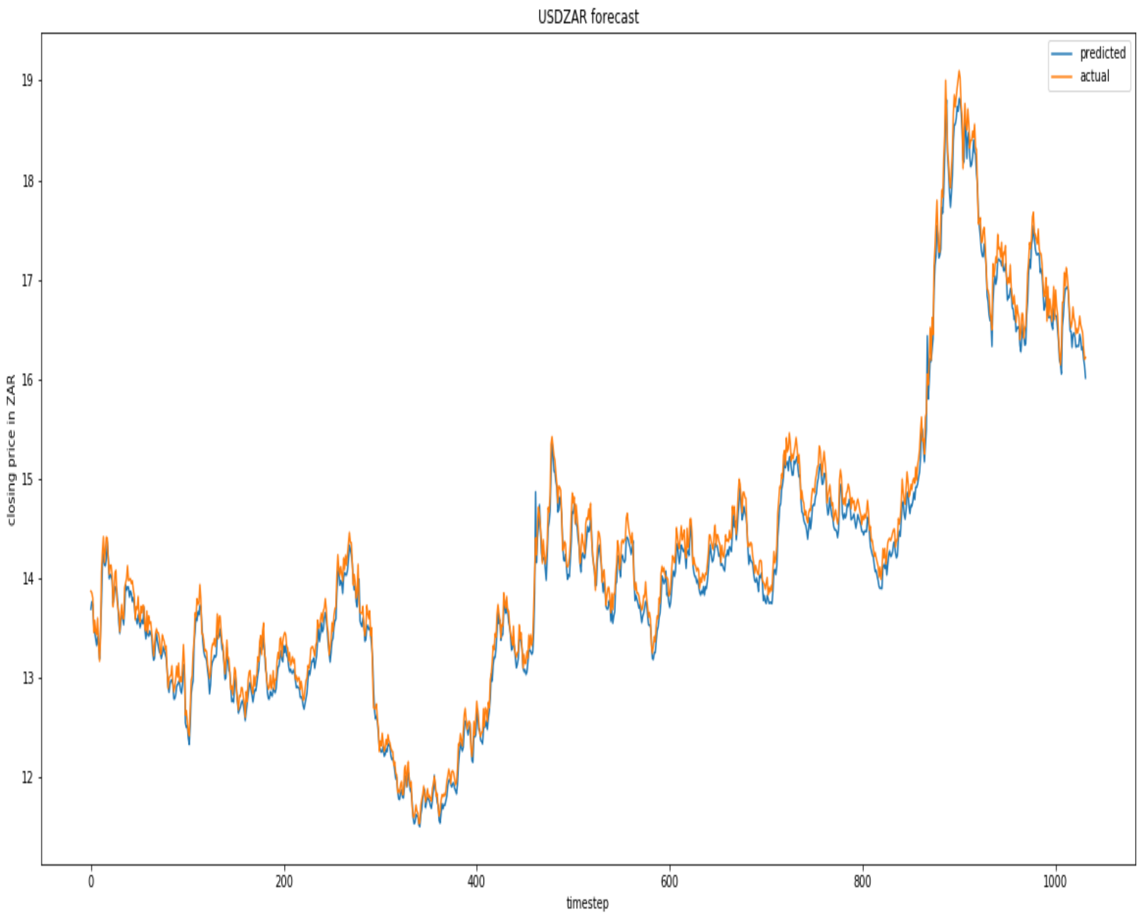

Figure 4.12 USDZAR forecast using SVR

After obtaining the expected closing price, the "inverse transform()" method was used to undo the previous transformation. The expected and actual closing prices were plotted against each other.

Figure 4.12 shows their plot. The predicted closing price and the actual closing price error had to be measured using the MSE and MAE. The achieved MSE was 0.02007424794650156 and the MAE was 0.12298385946739537.

4.4 Experiment 3: Extraction of MSE and MAE - LSTM model

Equally, the third experiment measures the MSE and MAE achieved when the LSTM model was used. The null hypothesis, in this case, is H_0: the LSTM model's MSE and MAE are not significantly different from those of the other models. Statistically, H_0: $\mu_1 = \mu_2$, where μ_1 represents the average MSE / MAE of the LSTM model, while μ_2 indicates the average MSE / MAE of either the ARIMA or SVR models. Similarly, two alternate hypotheses arise, where H1: the LSTM model outperforms the other models (H_1: $\mu_1 > \mu_2$) and H_2: the LSTM model underperforms (H_2: $\mu_1 < \mu_2$).

The same dependent, independent and controlled variables are of interest. The experiment setup is summarized below.

Title: Investigation of MSE and MAE from the LSTM model.

Null-Hypothesis: *H_0 : $\mu_1 = \mu_2$ – the LSTM model's performances are not significantly different from those of the other models (ARIMA/SVR)*

Alternate Hypothesis:*H_1:μ_1>μ_2 the LSTM model outperforms other models*

Alternate Hypothesis: *H_2 : $\mu_1 < \mu_2$ – the LSTM model underperforms*

Dependent variables: MSE and MAE

Independent variables: Time, base currency.

Controlled variables: Environment, machine specifications, parameters.

Procedure: The data was loaded, cleaned, and explored. It was then split into training and testing. A window(lookback) function was created to look back 60 days to improve the model's performance. The model was defined, complied, and fitted with scaled data. The performance was measured and the whole process was repeated with different hyperparameters.

Algorithm 4.3:

1. Define the network of the model
2. Compile Network
3. Fit Network
4. Evaluate Network's performance
5. Make predictions
6. Measure MSE and MAE
7. Repeat with different hyperparameters

Data collection: Table 4.3 summarizes the data collected.

Replication	MSE	MAE
1	0.001048298957599274	0.13500656115066173
2	0.0007628087709397718	0.12938841726170028
3	0.00012097195577863402	0.12476220941913224
4	0.001177669250442446	0.14845379924922023
5	0.0029228691415538197	0.14763348283693772
6	6.093722966678897e-05	0.12450194743696108
7	0.007197983799635698	0.1504619081852048
8	0.002435017999624422	0.1311052974493929
9	0.00325889041242768	0.14832248081384702
10	3.5597871073237614e-05	0.12060257638147637

The LSTM's MSE and MAE values changed with each replication because of the weights and the activation function. Random numbers are generated when the weights of the model are being adjusted in the layers, this is done to improve learning and achieve better results.

Figure 4.13 shows the packages used when creating the LSTM model. After loading the packages, the data was loaded and explored as shown in Figures 4.3 and 4.4.

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Input, LSTM, Dense
from keras.optimizers import Adam
import math
```

Figure 4.13 LSTM packages

```python
import math
#get the data to train the model on
training_data_len = math.ceil(len(dataset)*.70)
```

```python
#scale the data,it is important to scale and transform the data before transferring it i
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(dataset)
```

```python
scaled_data
```

```
array([[0.00342393],
       [0.00494567],
       [0.00791308],
       ...,
       [0.78847734],
       [0.77966643],
       [0.78090666]])
```

```python
#creating trained dataset
train_data=scaled_data[0:training_data_len, :]
```

```python
train_data
```

```
array([[0.00342393],
       [0.00494567],
       [0.00791308],
       ...,
       [0.60927656],
       [0.60404176],
       [0.59348084]])
```

Figure 4.14 Data scaling and transformation

The data was then split into 70% training and 30% testing. After splitting, the data was scaled and transformed using MinMaxScaler and transform function as shown in figure 4.14. The LSTM model works by taking the data sequentially. In this study, a 60 days lookback was used to look back 60 days into the past when training and testing. The output data of the lookback is as shown in figure 4.15. This data was collected as an array of 59 X variables/features and 1 Y variable or target feature.

```
x_train=[]
y_train=[]
```

```
#60 days lookback
for i in range(60,len(train_data)):
    x_train.append(train_data[i-60:i,0])
    y_train.append(train_data[i,0])
    if i<=60:
        print(x_train)
        print(y_train)
        print()
```

```
[array([0.00342393, 0.00494567, 0.00791308, 0.00167392, 0.0033098 ,
       0.        , 0.00448915, 0.00661959, 0.01118483, 0.01406093,
       0.01179353, 0.00760873, 0.01065222, 0.0089783 , 0.0101957 ,
       0.01141309, 0.01788051, 0.0167392 , 0.01141309, 0.01350549,
       0.01445658, 0.0125544 , 0.00875004, 0.00738047, 0.00513589,
       0.00464132, 0.00814134, 0.00867395, 0.01171744, 0.01116201,
       0.0167392 , 0.01559789, 0.01559789, 0.01171744, 0.01653377,
       0.02039139, 0.0221414 , 0.02586968, 0.02343488, 0.02267401,
       0.02183705, 0.01985878, 0.01719573, 0.0167392 , 0.01934139,
       0.02556533, 0.02967404, 0.02488054, 0.02914143, 0.02291749,
       0.0233588 , 0.02548924, 0.02434793, 0.01635877, 0.01563594,
       0.01103266, 0.00639133, 0.00509785, 0.00814134, 0.01342941])]
[0.014456584593846022]
```

```
#convert the x_train and y_train dataset to numpy arrays
x_train,y_train =np.array(x_train),np.array(y_train)
```

```
x_train.shape
```

```
7]: (2348, 60)
```

Figure 4.15 Look back output array

Figure 4.15 shows the Look back output arrays and reshaping of the training data. This was done since the LSTM model required 3-dimensional data. The code shown in figure 4.16 was used to build the LSTM model. The LSTM model contained 150 epochs, two layers of 100 neurons each, an activation function relu, an Adam optimizer, a dropout layer of 20%, and two dense layers of 2 and 1 neuron(s), respectively. The research did experiment with modifying the parameters of the LSTM, but the values listed above produced the best results.

Building LSTM Model

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Input, LSTM, Dense
from keras.optimizers import Adam
model = Sequential()
model.add(LSTM(100,activation='relu',return_sequences=True,input_shape=(x_train.shape[1],1) ))
model.add(LSTM(100,activation='relu'))
model.add(Dropout(0.20))
model.add(Dense(2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['mae'])
model.fit(x_train, y_train,epochs = 150)
```

```
Epoch 105/150
2348/2348 [==============================] - 12s 5ms/step - loss: 4.5456e-04 - mae: 0.0139
Epoch 106/150
2348/2348 [==============================] - 12s 5ms/step - loss: 3.7101e-04 - mae: 0.0129
Epoch 107/150
2348/2348 [==============================] - 11s 5ms/step - loss: 4.9121e-04 - mae: 0.0147
Epoch 108/150
2348/2348 [==============================] - 12s 5ms/step - loss: 4.3722e-04 - mae: 0.0137
Epoch 109/150
2348/2348 [==============================] - 12s 5ms/step - loss: 4.3902e-04 - mae: 0.0138
Epoch 110/150
2348/2348 [==============================] - 12s 5ms/step - loss: 4.3242e-04 - mae: 0.0139
Epoch 111/150
2348/2348 [==============================] - 12s 5ms/step - loss: 4.3772e-04 - mae: 0.0135
```

Figure 4.16 Building LSTM model

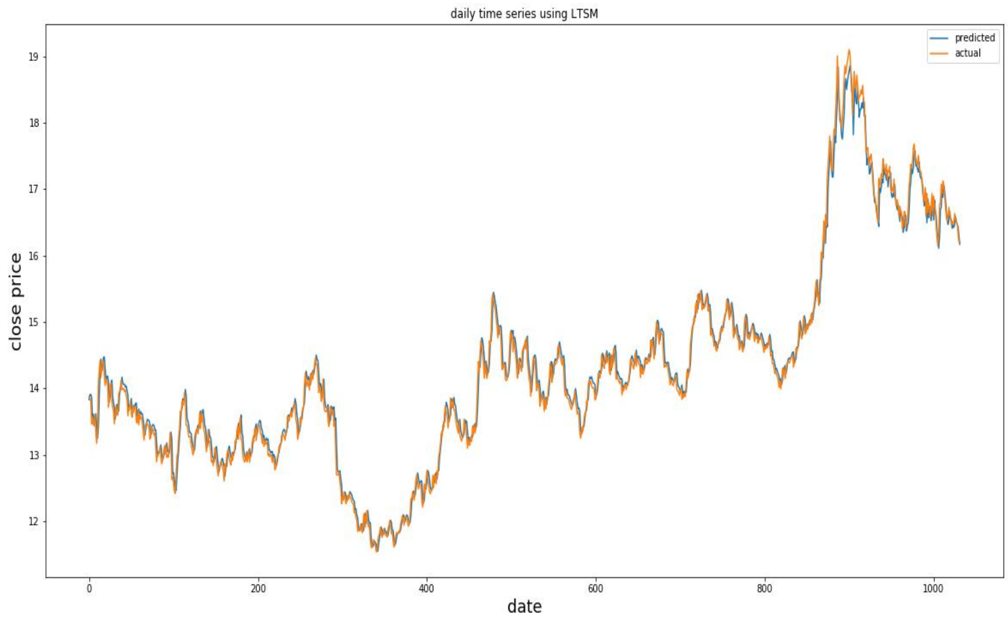

Figure 4.17 LSTM model results

After training the model was fed X variables of the test data to predict the target or Y values. Figure 4.17 shows the results of the predicted values plotted with the actual values. The LSTM achieved the lowest MSE of 0. 3.5597871073237614e-05 and MAE 0.12060257638147637

4.5 Findings

The average results achieved from the three experiments are summarized in the table below. These results will be discussed further in the next chapter.

	ARIMA	SVR	LSTM
MSE	0.022	0.02007424794650156	0.001902104539
MAE	0.11532114690886489	0.12298385946739537	0.136023868

Table 4.2 Models MSE and MAE average results

4.6 Summary

Key in this chapter is the administration of the three experiments that sought to determine the MSE and MAE achieved when the ARIMA, SVR, and the LSTM models were used. The null hypothesis of the first experiment was: $H_0 : \mu_1 = \mu_2$ concerning to the ARIMA model's performances. The alternate hypotheses were both directional, and were stated as H_1: $\mu_1 > \mu_2$ and $H_2 : \mu_1 < \mu_2$. The null hypothesis of the second experiment was: $H_0 : \mu_1 = \mu_2$ with respect to the SVR model's performances. The alternate hypotheses in this case were directional as well, and were stated as $H_1 : \mu_1 > \mu_2$ and $H_2 : \mu_1 < \mu_2$. The third experiment had similar hypotheses with respect to the performances of the LSTM model.

There were no sufficient evidence to support the acceptance of the null hypotheses in the first and second experiment. Rather, the second alternate hypothesis in both cases, $H_2 : \mu_1 < \mu_2$ holds. Precisely, the ARIMA and the SVR models were both outperformed by the LSTM model in all the categories measured. Precisely, table 4.2 summarized the performances of the three models. The next chapter concludes the study.

Chapter 5: Conclusions & Recommendations

5.1 Introduction

The purpose of this chapter is to reflect on the results yielded, discuss viewpoints, make recommendations, and the conclusions thereto. We look closely at the potential factors that influence USDZAR exchange rates. We also look at the summary of what each chapter covered. Most importantly, we provide the answers to the research questions posed in chapter 1, before we draw our conclusions, highlighting the main contributions of the work, as well as the future research directions.

5.1.1 Overview of the Chapter

Figure 5.1 depicts the breakdown of this chapter in a diagram, showing the seven sub-sections of interest, as well as the sequence in which these sections are tackled.

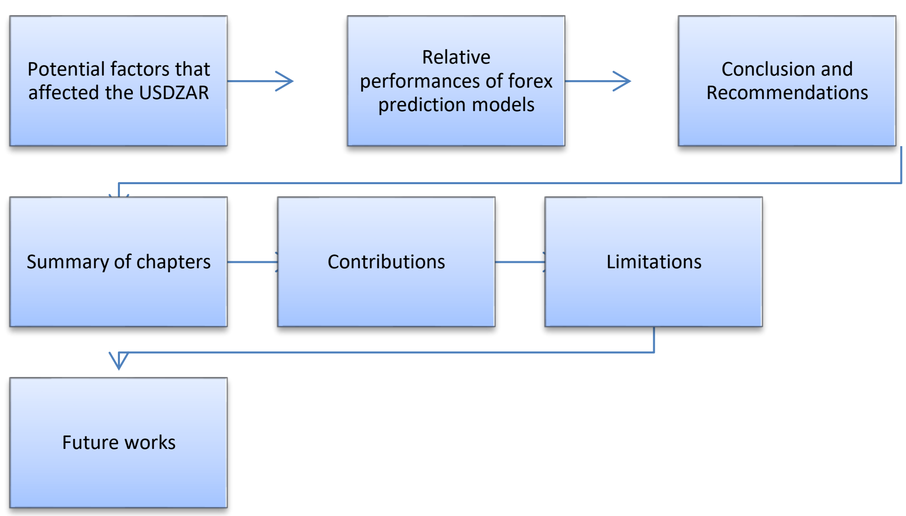

Figure 5.1 overview of the chapter

5. 2 Potential factors that affected the USDZAR

The USDZAR exchange rate appears to increase at the beginning and end of every year. The cause of this trend could be holidays and holiday trips by traders. These are the times that most people take holidays and travel.

The USDZAR exchange rate also appears to have increased from the year 2006 until the year 2020. The reasons for such an increase can be attributed to fewer tourists and investors coming into South Africa because of high crime rate, corruption cases, lack of jobs, retrenchments, strikes, closing down of companies, and the Covid 19 pandemic. All these factors played a role in the weakening of the rand against the USD.

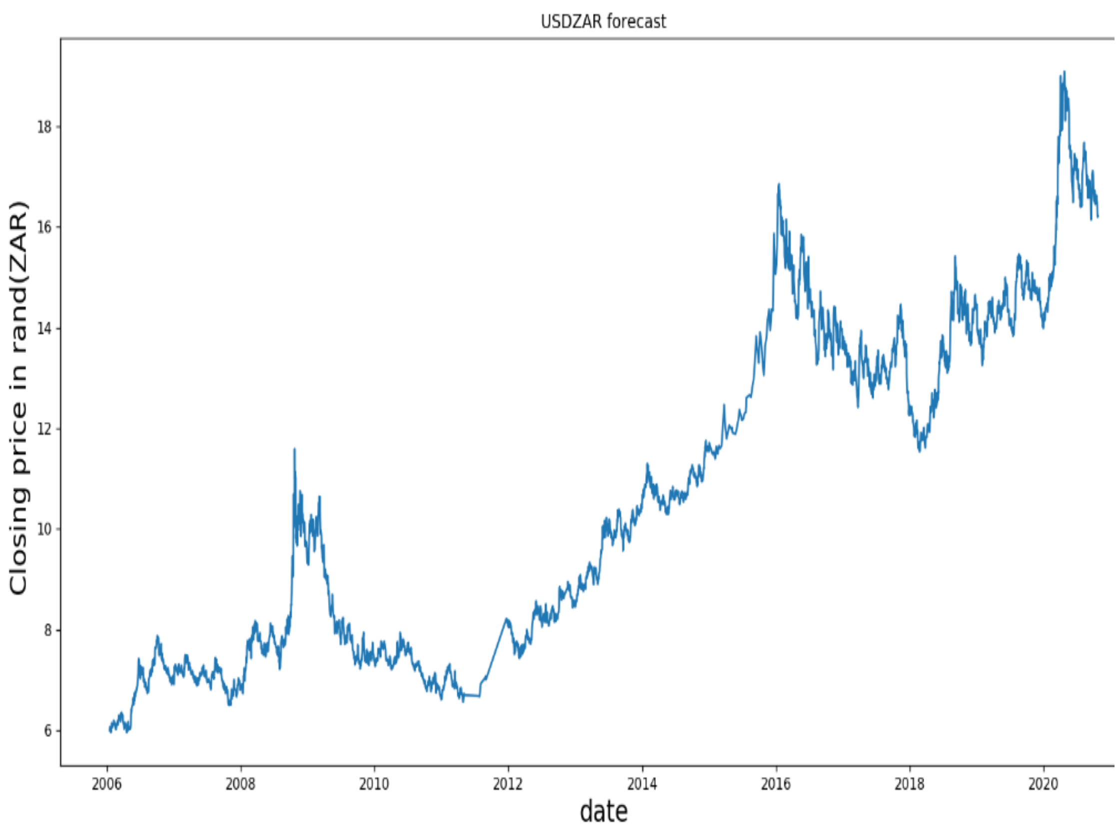

Figure 5.2 USDZAR forecast

South Africa reported its first Covid 19 case in March 2020. Lockdown was announced on the 27th of March in the same year. This lockdown had stringent rules against normal trading, such as no selling of alcohol and banning of international travels. This has had a big impact on the economy and the exchange rates. Figure 5.2 depicts the USDZAR forex rates over years. Prediction of such exchange rates using different prediction models was the theme of this study. The next section reports the relative performances of these different models.

5.3 Relative performances of forex prediction models

Table 5.1 reports the MSE and MAE achieved when each of the three forex prediction models under study was used. In the MSE category, the LSTM outperformed the others. The ARIMA model has the smallest MAE. These findings give adequate evidence to support the alternative hypothesis of experiment 3 that: $H_2 : \mu_1 > \mu_2$. To be more specific, we have enough evidence to dismiss the possibility of no significant differences between the three models and instead believe that the LSTM model beats the others.

	ARIMA	SVR	LSTM
MSE	0.022	0.02007424794650156	0.001902104539
MAE	0.11532114690886489	0.12298385946739537	0.136023868

Table 5.1 Average MSE and MAE results

5.4 Answers to the research questions

The study sought to answer the following questions in succession:

a) How do we design an LSTM deep neural network model for forecasting the South African Rand exchange rate against the US dollar? We determined that the most successful and efficient LSTM model should include 150 epochs, two layers, and 100 neurons each after creating and testing numerous LSTM models. It should have the activation function relu, the Adam optimizer, a 25% dropout layer, and two dense layers with two and one neuron (s) each.

b) How does the proposed LSTM model compare in performance to forecasting forex rates using the ARIMA and SVR models? In terms of MSE, the LSTM beat the SVR and ARIMA models. The LSTM, however, is generally effective in forecasting USDZAR rates, despite being outperformed by the ARIMA model when the MAE was assessed.

5.5 Summary of chapters

Chapter 1 introduced the study by giving the broader statement of the problem, along with the research questions, aims, objectives, significance of the study, and motivation of the study.

Chapter 2 gave the background of the forex market, factors affecting the foreign market, information regarding time series machine learning, and deep learning in forex. The gap we explored was elucidated.

Chapter 3 presented the methodology followed in this study, along with the theoretical framework that informed the reasoning and argumentations undertaken. The research design was presented.

Chapter 4 mainly administered three experiments, all aimed at deciding on which forecasting model achieved the best MSE measure. The LSTM model outclassed the ARIMA and the SVR model in this regard.

Chapter 5 discusses the factors that influence exchange rates, as well as the results achieved. It summarizes the dissertation and provides conclusions.

5.6 Contributions

This study fills a gap in the discipline of computational sciences, financial mathematics, and partially in economics. It, thus, makes contributions both from an academic and from a commercial angle. From the commercial perspective, the developed model may help stakeholders make better and informed decisions when trading.

The model makes a theoretical and methodological contribution, it adds value and gives hyperparameters that can help build a more improved LSTM model for forex forecasting.

5.7 Limitations

The limitations of this study were time and Covid 19. Covid 19 affected the trade and tourism sector. As a result, the forex market was affected too. The effects of Covid 19 in the forex market can later impact the LSTM model's prediction accuracy given that it was trained during the pandemic.

5.8 Future works

More research is needed to identify effective ways of improving sentiment analysis using natural language processing (NLP). Hopefully, sentiment analysis would bring about better predictions and understanding of market behaviour. Also, more research is needed to investigate the building of a trading robot using the LSTM model, hopefully, along with the desired NLP.

References

Abu-Mostafa, Y.S. and Atiya, A.F., 1996. Introduction to financial forecasting. *Applied intelligence*, *6*(3), pp.205-213.

Acton, R., 2016. The missing link. *How tackling financial difficulty can boost recovery rates in IAPT. London: Money and Mental Health Institute*.

Adebiyi, A.A., Adewumi, A.O. and Ayo, C.K., 2014. Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, *2014*.

Adhikari, R. and Agrawal, R.K., 2013. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*.

Alamili, M., 2011. Exchange rate prediction using support vector machines: a comparison with artificial neural networks.

Amiri, M., Zandieh, M., Vahdani, B., Soltani, R. and Roshanaei, V., 2010. An integrated eigenvector–DEA–TOPSIS methodology for portfolio risk evaluation in the FOREX spot market. *Expert Systems with Applications*, *37*(1), pp.509-516.

Baasher, A.A. and Fakhr, M.W., 2011, October. Forex trend classification using machine learning techniques. In *Proceedings of the 11th WSEAS international conference on Applied computer science* (Vol. 1, No. 1, pp. 41-47). World Scientific and Engineering Academy and Society (WSEAS).

Bandara, K., Bergmeir, C. and Smyl, S., 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert systems with applications*, *140*, p.112896.

Baradwaj, B.K. and Pal, S., 2012. Mining educational data to analyze students' performance. *arXiv preprint arXiv:1201.3417*.

Baruník, J., Kočenda, E. and Vácha, L., 2017. Asymmetric volatility connectedness on the forex market. *Journal of International Money and Finance*, *77*, pp.39-56.

Berwick, R., 2003. An Idiot's guide to Support vector machines (SVMs). *Retrieved on October*, *21*, p.2011.

Bisandu, D.B., 2016. Design science research methodology in Computer Science and Information Systems. *International Journal of Information Technology*, *5*(4), pp.55-60.

Bishop, C.M., 2006. *Pattern recognition and machine learning*. springer.

Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M., 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Bryman, A., 2016. *Social research methods*. Oxford university press.

Cao, L. and Tay, F.E., 2001. Financial forecasting using support vector machines. *Neural Computing & Applications*, *10*(2), pp.184-192.

Carlson, J.A. and Osler, C.L., 2003. Currency risk premiums: Theory and evidence. *Staff Reports*, *70*.

Cerrato, M., Sarantis, N. & Saunders, A.(2011). An investigation of customer order flow in the foreign exchange market. *Journal of Banking & Finance,* 35(8), pp. 1892--1906.

Chandrasekara, N.V. and Tilakaratne, C.D., 2009. Forecasting exchange rates using artificial neural networks. *Sri Lankan Journal of Applied Statistics*, *10*(2009), p.187.

Chatzis, S.P., Siakoulis, V., Petropoulos, A., Stavroulakis, E. and Vlachogiannakis, N., 2018. Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert systems with applications*, *112*, pp.353-371.

Cheng, G., 2007. *7 Winning Strategies For Trading Forex: Real and actionable techniques for profiting from the currency markets*. Harriman House Limited.

Chen, Y.Y., Lin, Y.H., Kung, C.C., Chung, M.H. and Yen, I., 2019. Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, *19*(9), p.2047.

Christopher, G. D. & Janet, M., 2004. Centre for Research on Stress, Coping and Well-being. Carleton University, Ottawa: Department of Psychology. *The Consequences of Financial Stress for Individuals, Families, and Society.*

Chu, H. & Ke, Q., 2017. Research methods: What's in the name?. *Library & Information Science Research,* 39(4), pp. 284--294.

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cochrane, J.H., 2005. Time series for macroeconomics and finance. *Manuscript, University of Chicago*, pp.1-136.

Cuenin, S., 1987. The Use of Performance Indicators in Universities: An International Survey. *International Journal of Institutional Management in Higher Education*, *11*(2), pp.117-39.

Culkin, R. and Das, S.R., 2017. Machine learning in finance: the case of deep learning for option pricing. *Journal of Investment Management*, *15*(4), pp.92-100.

Dautel, A.J., Härdle, W.K., Lessmann, S. and Seow, H.V., 2020. Forex exchange rate forecasting using deep recurrent neural networks. *Digital Finance*, *2*(1), pp.69-96.

De Waal, C., 2001. On Peirce. Belmont, CA: Wadsworth.

Dean, J., Corrado, G.S., Monga, R., Chen, K., Devin, M., Le, Q.V., Mao, M.Z., Ranzato, M.A., Senior, A., Tucker, P. and Yang, K., 2012. Large scale distributed deep networks.

Deboeck, G.J. ed., 1994. *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets* (Vol. 39).

John Wiley & Sons.Devore, J. L., 2011. Probability and Statistics for Engineering and the Sciences*. s.l.:Cengage learning.*

Di Persio, L. & and Honchar, O., 2016. Artificial neural networks approach to the forecast of stock market price movements. *International Journal of Economics and Management Systems,* Volume 1.

Ding, L. and Hiltrop, J., 2010. The electronic trading systems and bid-ask spreads in the foreign exchange market. *Journal of International Financial Markets, Institutions and Money*, *20*(4), pp.323-345.

Doyle, C., Sammon, D. and Neville, K., 2015. Building an evaluation framework for social media-enabled collaborative learning environments (SMECLEs). *Journal of Decision Systems*, *24*(3), pp.298-317.

DraKoln, N., 2008. *Winning the trading game: why 95% of traders lose and what you must do to win* (Vol. 322). John Wiley & Sons.

Draper, N.R. and Smith, H., 1998. *Applied regression analysis* (Vol. 326). John Wiley & Sons.

Dreos, S., Papaemmanouil, O. & Chaudhuri, S., 2015. Overview of data exploration techniques. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data.*, pp. 277--281.

Drucker, H. et al., 1997. Advances in neural information processing systems.

Faraway, J. & Chatfield, C., 1998. Time series forecasting with neural networks: a comparative study using the air line data. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 47*(2), pp. 231--250.

Fischer, T. & Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research,* 270(Elsevier), pp. 654--669.

French, J., 2017. The time traveller's CAPM. *Investment Analysts Journal, 46*(2), pp. 81--96.

Galeshchuk, S. & Mukherjee, S., 2017. Deep networks for predicting direction of change in foreign exchange rates. *Intelligent Systems in Accounting, Finance and Management, 24*(4), pp. 00--110.

Glorot, X., Bordes, A., & Bengio, Y., 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323)

Guba, E. G. & Lincoln, Y. S., 1994. Competing paradigms in qualitative research. *Handbook of qualitative research, 2*(163-194), p. 105.

Hagan, M., Demuth, H., Beale, M. & Jesu´s, O., 2014. Neural Network Design *(2nd Edition). 2nd ed. United States: Martin Hagan.*

Hamzaçebi, C., 2008. Improving artificial neural networks' performance in seasonal time series forecasting. *Information Sciences*, *178*(23), 4550-4559.

Handa, R., Shrivas, A.K. and Hota, H.S., 2019. Financial Time Series Forecasting using Back Propagation Neural Network and Deep Learning Architecture. *Int. J. Recent Technol. Eng*, *8*, pp.3487-3492.

Handa, R. & Shrivas, A., 2017. PREDICTION OF FOREX DATA USING NEURAL NETWORK TECHNIQUES WITH FEATURE EXTRACTION. *Review of Business and Technology Research,14*(2),pp.16--38.

Hayley, S. and Marsh, I.W., 2016. What do retail FX traders learn?. *Journal of International Money and Finance*, *64*, pp.16-38.

Hevner, A.R., 2007. A three cycle view of design science research. *Scandinavian journal of information systems*, *19*(2), p.4.

Hill, T., O'Connor, M. and Remus, W., 1996. Neural network models for time series forecasts. *Management science*, *42*(7), pp.1082-1092.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hipel, K.W. and McLeod, A.I., 1994. *Time series modelling of water resources and environmental systems*. Elsevier.

Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation, 9*(8), pp. 1735-1780.

Huang, W., Lai, K. K., Nakamori, Y. & Wang, S., 2004. Forecasting foreign exchange rates with artificial neural networks: A review. *International Journal of Information Technology \& Decision Making, 3*(1), pp. 145--165.

Hyde, K. F., 2000. Recognising deductive processes in qualitative research. *Qualitative market research: An international journal.*

Hyndman, R.J. and Koehler, A.B., 2006. Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), pp.679-688.

Johnson, R.B. and Onwuegbuzie, A.J., 2004. Mixed methods research: A research paradigm whose time has come. *Educational researcher*, *33*(7), pp.14-26.

Jozef, B., Evzen, K. & Lukas, V., 2017. Asymmetric volatility connectedness on the forex market. *Journal of International Money and Finance,* 77, pp. 39--56.

Kamruzzaman, J., Begg, R., & Sarker, R., 2006. *Artificial neural networks in finance and manufacturing*. IGI Global.

Kamruzzaman, J. & Sarker, R., 2004. Application of support vector machine to forex monitoring. *IEEJ Transactions on Electronics, Information and Systems, 124*(10), pp. 1944--1951.

Kecman, V., 2001. *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT press.

Khawula, B. M. S., 2016. *An evaluation of community participation in the integrated development planning (IDP) process: A case study of Umzumbe municipality in the province of Kwazulu-Natal in South Africa* (Doctoral dissertation).

Kihoro, J., Otieno, R., & Wafula, C., 2006. SEASONAL TIME SERIES FORECASTING: A COMPARATIVE STUDY OF ARIMA AND ANN MODELS. *African Journal of Science and Technology, 5*.

Kim, S. & Kang, M., 2019. Financial series prediction using Attention LSTM. *arXiv preprint arXiv:1902.10877.*

Kombrink, S., Mikolov, T., Karafiat, M. & Burget, L., 2011. Recurrent neural network based language modeling in meeting recognition. In: *Twelfth annual conference of the international speech communication association.*

Kondratenko, V. & Kuperin, Y. A., 2003. Using recurrent neural networks to forecasting of forex. *arXiv preprint cond-mat/0304469.*

Koronidis, G., 2013. Pairs trading in forex markets: A test of the cointegration approach with the EUR/USD and GBP/USD currency pairs. LeCun, Y., Bengio, Y., & Hinton, G., 2015. Deep learning. *Nature*, *521*(7553), 436–444.Leitch, G., & Tanner, J. E., 1991. Economic forecast evaluation: Profits versus the conventional error measures. *The American Economic Review*, *81*(3), pp.580–590.

Lehmann, E.L. and Casella, G., 2006. *Theory of point estimation*. Springer Science & Business Media.

Levich, R. M. & Packer, F., 2017. Development and Functioning of FX Markets in Asia and the Pacific. *Financial Markets, Institutions & Instruments, 26*(1), pp. 3--58.

Marshall, P. A., Adebamowo, C. A., Adeyemo, A. A., Ogundiran, T. O., Strenski, T., Zhou, J., & Rotimi, C. N., 2014. Voluntary participation and comprehension of informed consent in a genetic epidemiological study of breast cancer in Nigeria. *BMC medical ethics*, *15*(1), pp. 1-11.

McNelis, P.D., 2005. *Neural networks in finance: gaining predictive edge in the market*. Academic Press.

Nagpure, A., 2019. Prediction of multi-currency exchange rates using deep learning. *Int. J. Innov. Technol. Explor. Eng,* 8(6), pp. 316--322.

Neuman., L., W., 2014. Social Research Methods: Qualitative and Quantitative Approaches. 7th ed. Pearson Education Limited

Ni, L. et al., 2019. Forecasting of forex time series data based on deep learning. *Procedia computer science,*147, pp. 647--652.

Oliver, P., 2010. *The student's guide to research ethics*. McGraw-Hill Education (UK).

Onweugbuzie, A.J., 2002. WHY CAN'T WE ALL GET ALONG? TOWARDS A FRAMEWORK FOR UNIFYING RESEARCHPARADIGMS. *Education*, *122*(3).

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S., 2007. A design science research methodology for information systemsresearch. *Journal of management information systems*, *24*(3), pp.45-77.

Pujari, M.V., Sayyed, A.H., Shahani, H., Rupani, D. and Student, B.E., 2018. Forex Trading System. *International Journal of Engineering Science*, *17116*.

Qiu, J., Wang, B. & Zhou, C., 2020. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one,* 15(1).

Qu, Y. and Zhao, X., 2019, June. Application of LSTM neural network in forecasting foreign exchange price. In *Journal of Physics: Conference Series* (Vol. 1237, No. 4, p. 042036). IOP Publishing.

Raicharoen, T., Lursinsap, C. and Sanguanbhokai, P., 2003, May. Application of critical support vector machine to time series prediction. In *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03.* (Vol. 5, pp. V-V). IEEE.

Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Ramadhani, I. and Rismala, R., 2016, July. Prediction of multi currency exchange rates using correlation analysis and backpropagation. In *2016 International Conference on ICT For Smart Society (ICISS)* (pp. 63-68). IEEE.

Rehman, M., Khan, G.M. and Mahmud, S.A., 2014. Foreign currency exchange rates prediction using cgp and recurrent neural network. *IERI Procedia*, *10*, pp.239-244.

Richardson, T., Elliott, P., Roberts, R. and Jansen, M., 2017. A longitudinal study of financial difficulties and mental health in a national sample of British undergraduate students. *Community mental health journal*, *53*(3), pp.344-352.

Ryll, L. and Seidens, S., 2019. Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey. *arXiv preprint arXiv:1906.07786.*

Shahram, M. & Komeil, T., 2018. Is It Necessary to Restrict Forex Financial Trading?A Modified Model. *Journal of Money and Economy, 13*(1), pp. 63-80.

Siegel, A., 2016. *Practical business statistics*. Academic Press.

Siripurapu, A., 2014. Convolutional networks for stock trading. *Stanford Univ Dep Comput Science,* pp. 1-6.

Skapinakis, P. et al., 2006. Socio-economic position and common mental disorders: Longitudinal study in the general population in the UK. *The British Journal of Psychiatry, 189*(2), pp. 109--117.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), pp.1929–1958.

Steel, R., George, D. & Torrie, J. H., 1960. Principles and procedures of statistics. *Principles and procedures of statistics.*

Talebi, H., Hoang, W. and Gavrilova, M.L., 2014. Multi-scale foreign exchange rates ensemble for classification of trends in forex market. *Procedia Computer Science*, *29*, pp.2065-2075.

Tang, Z. & Fishwick, P. A., 1993. Feedforward neural nets as models for time series forecasting. *ORSA Journal on Computing, 5*(4), pp. 374--385.

Uysal, İ. & Güvenir, H. A., 1999. An overview of regression techniques for knowledge discovery. *Knowledge Engineering Review,* 14, pp. 319-340.

Vargas, M.R., De Lima, B.S. and Evsukoff, A.G., 2017, June. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (pp. 60-65). IEEE.

Violeta, G., 2010. Introduction to the foreign exchange market.

Wang, J.H. and Leu, J.Y., 1996, June. Stock market trend prediction using ARIMA-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)* (Vol. 4, pp. 2160-2165). IEEE.

Willmott, C. J. & Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research, 30*(1), pp. 79--82.

Woiceshyn, J. & Daellenbach, U., 2018. Evaluating inductive vs deductive research in management studies. *Qualitative Research in Organizations and Management: An International Journal.*

Worasucheep, C., 2015, May. Forecasting currency exchange rates with an Artificial Bee Colony-optimized neural network. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3319-3326). IEEE.

Yao, J. & Tan, C. L., 2000. A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing, 34*(1-4), pp. 79--98.

Yasir, M., Durrani, M.Y., Afzal, S., Maqsood, M., Aadil, F., Mehmood, I. and Rho, S., 2019. An intelligent event-sentiment-based daily foreign exchange rate forecasting system. *Applied Sciences*, *9*(15), p.2980.

Yong, Y.L., Lee, Y., Gu, X., Angelov, P.P., Ngo, D.C.L. and Shafipour, E., 2018. Foreign currency exchange rate prediction using neuro-fuzzy systems. *Procedia computer science*, *144*, pp.232-238.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, pp.159-175.

Zhang, G., Patuwo, B. E. & Hu, M. Y., 1998. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting, 14*(1), pp. 35--62.

Zheng, Y., Liu, Q., Chen, E., Ge, Y. and Zhao, J.L., 2014, June. Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management* (pp. 298-310). Springer, Cham.

Zimmermann, H.-G., Neuneier, R. & Grothmann, R., 2001. Multi-agent modeling of multiple FX-markets by neural networks. *IEEE Transactions on Neural Networks, 12*(4), pp. 735--743.

Appendix A- Screenshots of Important code

1 ARIMA CODE

Importing packages

```
import pandas as pd
from pandas import DataFrame
from statsmodels.tsa.arima_model import ARIMA
from matplotlib import pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
import numpy as np
```

Loading data

```
df = pd.read_csv("./Forex_data/USDZARNov.csv",delimiter=',',
parse_dates=[0], header=0,skiprows=0)

df.columns=['date','time','open','high','low','close','volume']
```

Visualization

```
plt.figure(figsize=(16,8))
plt.title('USDZAR forecast')
plt.plot(df['close'])
plt.xlabel('date',fontsize=18)
plt.ylabel(' Closing price in rand(ZAR)',fontsize=18)
plt.savefig('usdzardaily.png')
plt.show()
```

Seasonal decomposition

```
result = seasonal_decompose(df['close'],
model='multiplicative',period=365)
plt.figure(figsize=(16,8))
result.plot()
plt.figure(figsize=(16,8))
#plt.savefig('usdzardailytrend.png')
plt.show()
```

Checking stationarity

```
# Adfuller test
from statsmodels.tsa.stattools import adfuller
adf_test = adfuller(df['close'])
```

```python
print('stat=%.3f, p=%.3f' % adf_test[0:2])
if adf_test[1] > 0.05:
 print('Probably not Stationary')
else:
 print('Probably Stationary')

# kpss test

from statsmodels.tsa.stattools import kpss
kpss_test = kpss(df['close'], nlags='auto')
print('stat=%.3f, p=%.3f' % kpss_test[0:2])
if kpss_test[1] > 0.05:
    print('Probably Stationary')
else:
    print('Probably not Stationary')
```

Finding lowest AIC and model values

```python
import itertools
p = range(1, 4)
d = range(1, 2)
q = range(1, 3)
pdq = list(itertools.product(p, d, q))
aics = []
params = []
for param in pdq:
    model = ARIMA(df['close'], order=param)
    model_fit = model.fit()
    aic = model_fit.aic
    aics.append(aic)
    params.append(param)
combo = list(zip(aics, params))
combo.sort()
combo_array = np.array(combo)
print(combo_array)
```

Training and testing

```python
from sklearn.metrics import mean_squared_error
#Building model
X = df['close'].values
size = int(len(X) * 0.70)
train, test = X[0:size], X[size:len(X)]
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(1, 1, 1))
    model_fit = model.fit(disp=0)
```

```
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]
    history.append(obs)
    print('predicted=%f, expected=%f' % (yhat, obs))
# plot
plt.plot(test)
plt.plot(predictions, color='red')
plt.show()
```

MSE and MAE calculation

```
from sklearn.metrics import mean_squared_error
error = mean_squared_error(test, predictions)
from sklearn.metrics import mean_absolute_error
mean_absolute_error(test, predictions)
```

Results visualization

```
plt.figure(figsize=(20,10))
plt.plot(predictions,label='predicted')
plt.plot(test,label='actual')
plt.legend()
plt.xlabel('date',fontsize=18)
plt.ylabel('close price in Rands',fontsize=18)
plt.title("daily time series using ARIMA")
plt.savefig('usdzardaARIMA.png')
```

2 Support vector regression

Importing packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import
r2_score,mean_squared_error,mean_absolute_error
from subprocess import check_output
```

Loading data

```
df = pd.read_csv("./Forex_data/USDZARNov.csv",delimiter=',',
parse_dates=[0], header=0,skiprows=0)

df.columns=['date','time','open','high','low','close','volume']
df=df.drop(['time'],axis=1)


df['date'] = pd.to_datetime(df['date'], infer_datetime_format=True)
df.set_index('date', inplace=True)
df = df.astype(float)
df.head()

Y=df['close'].values.reshape(-1,1)
```

Data scaling

```
sc_X = StandardScaler()
sc_y = StandardScaler()
X = sc_X.fit_transform(X)
Y = sc_y.fit_transform(Y)
```

Creating a scaled dataset for rain and test data

```
train_size = int(df.shape[0] * 0.70)
trainX = X[:train_size]
trainY = Y[:train_size]
testX = X[train_size:]
testY = Y[train_size:]
```

Build SVR

```
SupportVectorRegModel= SVR (C=1.0, cache_size=200, coef0=0.0,
degree=3, epsilon=0.1,
    gamma='auto', kernel='linear', max_iter=-1, shrinking=True,
    tol=0.001, verbose=False)
SupportVectorRegModel.fit(trainX,trainY)
Y_pred=SupportVectorRegModel.predict(testX)
y_pred = sc_y.inverse_transform(Y_pred)
close = sc_y.inverse_transform(np.reshape(testY, (testY.shape[0],
1)))
```

MSE and MAE

```
mse=mean_squared_error(close,y_pred)
mae=mean_absolute_error(close,y_pred)
```

Visualization

```
plt.figure(figsize=(20,10))
```

```
plt.plot(y_pred,label='predicted')
plt.plot(close,label='actual')
plt.xlabel('timestep')
plt.ylabel('closing price in ZAR' )
plt.legend()
plt.title("USDZAR forecast")
plt.savefig('usdzar1.png')
```

3 LSTM

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Input, LSTM,
Dense
from keras.optimizers import Adam
import math
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Input, LSTM,
Dense
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
import math
```

Loading data

```
df = pd.read_csv("./Forex_data/USDZARNov.csv",delimiter=',',
parse_dates=[0], header=0,skiprows=0)
df.columns=['date','time','open','high','low','close','volume']
df=df.drop(['time'],axis=1)
df['date'] = pd.to_datetime(df['date'], infer_datetime_format=True)
df.set_index('date', inplace=True)
df = df.astype(float)
Y=df['close'].values.reshape(-1,1)
#create dataframe containing the closing price only
data=df.filter(['close'])
#convet the dataframe to numpy array
dataset=data.values


#get the data to train the model on
```

```
training_data_len = math.ceil(len(dataset)*.70)
#scale the data,it is important to scale and transform the data
before transferring it into an LSTM/nueral netork
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(dataset)
#creating trained dataset
train_data=scaled_data[0:training_data_len, :]

x_train=[]
y_train=[]


#60 days lookback
for i in range(60,len(train_data)):
    x_train.append(train_data[i-60:i,0])
    y_train.append(train_data[i,0])
    if i<=60:
        print(x_train)
        print(y_train)
        print()
#convert the x_train and y_train dataset to numpy arrays
x_train,y_train =np.array(x_train),np.array(y_train)
#reshape the data
x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))
y_train.shape
```

Building a LSTM model

```
model = Sequential()
model.add(LSTM(100,activation='relu',return_sequences=True,input_sha
pe=(x_train.shape[1],1) ))
model.add(LSTM(100,activation='relu'))
model.add(Dropout(0.20))
model.add(Dense(2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam',
metrics=['mae'])
model.fit(x_train, y_train,epochs = 150, batch_size = 20)
```

Create a scaled dataset for test data

```
test_data=scaled_data[training_data_len-60: , : ]
#create x and y test
x_test=[]
y_test=dataset[training_data_len: , :]
for i in range(60,len(test_data)):
    x_test.append(test_data[i-60:i,0])

#convert data to numpy array
```

```
x_test=np.array(x_test)
x_test=np.reshape(x_test,(x_test.shape[0],x_test.shape[1] ,1))
```

Get the models predicted values

```
predictions=model.predict(x_test)
predictions=scaler.inverse_transform(predictions)
```

Visualization

```
plt.figure(figsize=(20,10))
plt.plot(predictions,label='predicted')
plt.plot(y_test,label='actual')
plt.legend()
plt.xlabel('date',fontsize=18)
plt.ylabel('close price',fontsize=18)
plt.title("daily time series using LTSM")
```

MAE and MSE error

```
MSE=np.mean(predictions-y_test)**2
mean_absolute_error(y_test, predictions)
```

Appendix B- Editing Certificate

SOL PLAATJE
UNIVERSITY

Dr. J. Sibanda (Senior Lecturer: English)
School of Education
Private Bag X 5008, Kimberley, 8300
North Campus, Chapel Street, Kimberley

Website: www.spu.ac.za

20 April 2021

Certificate of language editing

To whom it may concern

I hereby confirm that I have proofread and edited the following M.Com. dissertation in Business Information Systems using the Windows Tracking System to reflect my comments and suggested corrections for the author (s) to action.

Title: Application of Deep Neural Networks in Forecasting Foreign Currency Exchange rates

Author(s) Andisani Nemavhola supervised by Dr Colin Chibaya (SPU) and Professor Ochara (University of Venda)

Although the greatest care was taken in the editing of this document, the final responsibility for the product rests with the author (s).

Signature

20 April 2021
Date

Appendix C – Similarity Report

ORIGINALITY REPORT

16%	2%	14%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Andisani Nemavhola, Colin Chibaya, Nixon Muganda Ochara. "Application of the LSTM - Deep Neural Networks - in Forecasting Foreign Currency Exchange rates", 2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), 2021 Publication	10%
2	Submitted to Polytechnic of Namibia Student Paper	3%
3	link.springer.com Internet Source	1%
4	towardsdatascience.com Internet Source	1%
5	Submitted to University of Leicester Student Paper	1%

Appendix D – Ethics certificate

ETHICS APPROVAL CERTIFICATE

RESEARCH AND INNOVATION
OFFICE OF THE DIRECTOR

NAME OF RESEARCHER/INVESTIGATOR:
Mr A Nemavhola

STUDENT NO:
15005397

PROJECT TITLE: Application of deep neural networks in the forecasting of foreign currency exchange rates.

ETHICAL CLEARENCE NO: SMS/21/BIS/02/2404

SUPERVISORS/ CO-RESEARCHERS/ CO-INVESTIGATORS

NAME	INSTITUTION & DEPARTMENT	ROLE
Dr C Chibaya	University of Venda	Supervisor
Prof NM Ochara	University of Venda	Co - Supervisor
Mr A Nemavhola	University of Venda	Investigator – Student

Type: **Masters Research**
Risk: **Minimal risk to humans, animals or environment (Category 2)**
Approval Period: **April 2021 – April 2023**

The Research Ethics Social Sciences Committee (RESSC) hereby approves your project as indicated above.

General Conditions
While this ethics approval is subject to all declarations, undertakings and agreements incorporated and signed in the application form, please note the following.

- The project leader (principal investigator) must report in the prescribed format to the REC:
  - Annually (or as otherwise requested) on the progress of the project, and upon completion of the project
  - Within 48hrs in case of any adverse event (or any matter that interrupts sound ethical principles) during the course of the project.
  - Annually a number of projects may be randomly selected for an external audit.
- The approval applies strictly to the protocol as stipulated in the application form. Would any changes to the protocol be deemed necessary during the course of the project, the project leader must apply for approval of these changes at the REC. Would there be deviated from the project protocol without the necessary approval of such changes, the ethics approval is immediately and automatically forfeited.
- The date of approval indicates the first date that the project may be started. Would the project have to continue after the expiry date; a new application must be made to the REC and new approval received before or on the expiry date.
- In the interest of ethical responsibility, the REC retains the right to:
  - Request access to any information or data at any time during the course or after completion of the project,
  - To ask further questions; Seek additional information; Require further modification or monitor the conduct of your research or the informed consent process.
  - withdraw or postpone approval if:
  - Any unethical principles or practices of the project are revealed or suspected.
  - It becomes apparent that any relevant information was withheld from the REC or that information has been false or misrepresented.
  - The required annual report and reporting of adverse events was not done timely and accurately,
  - New institutional rules, national legislation or international conventions deem it necessary

ISSUED BY:
UNIVERSITY OF VENDA, RESEARCH ETHICS COMMITTEE
Date Considered: April 2021

Name of the RESSC Chairperson of the Committee: Prof Takalani Mashau

Signature: PMashau

UNIVERSITY OF VENDA
OFFICE OF THE DIRECTOR
RESEARCH AND INNOVATION
2021 -04- 18
Private Bag X5050
Thohoyandou 0950

University of Venda
PRIVATE BAG X5050, THOHOYANDOU, 0950, LIMPOPO PROVINCE, SOUTH AFRICA
TELEPHONE (015) 962 8504/8313 FAX (015) 962 9060
"A quality driven financially sustainable, rural-based Comprehensive University"

Appendix E – UHDC certificate

UNIVERSITY OF VENDA

OFFICE OF THE DVC: RESEARCH AND POSTGRADUATE STUDIES

TO : MR/MS A. NEMAVHOLA

SCHOOL OF MANANGEMENT SCIENCES

FROM: PROF. N.N FEZA

DVC: RESEARCH AND POSTGRADUATE STUDIES

DATE : 19 JULY 2021

DECISIONS TAKEN BY UHDC OF 19th JUY 2021

Application for approval of Masters Proposal Report in Management Sciences: **A. Nemavhola (15005397)**

Topic: "Application of Deep Neural Networks in Forecasting Foreign Currency Exchange Rates."

Supervisor	SPU	Dr. C. Chibaya
Co-supervisor	UNIVEN	Prof. N.M Ochara

UHDC approved Masters proposal

PROF. N.N FEZA
DVC: RESEARCH AND POSTGRADUATE STUDIES

======= IN GOD WE TRUST =======