ASSESSING RECURRENT NEURAL NETWORKS AS A PREDICTION TOOL FOR QUOTED STOCK PRICES ON THE NAIROBI SECURITIES EXCHANGE

AHMED ABDILLAHI MUCHUKU

D63/11486/2018

A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE REQUIREMENTS TO THE AWARD OF DEGREE OF MASTER OF SCIENCE IN FINANCE, FACULTY OF BUSINESS AND MANAGEMENT SCIENCES, UNIVERSITY OF NAIROBI

2022

DECLARATION

This research project is my original work and has not been presented by a student for a degree in any other university.

Signed: ____________________ Date: 10/11/2022

Muchuku, Abdillahi Ahmed

D63/11486/2018

This research project has been submitted for presentation with our approval as University Supervisors

Signed: ____________________ Date: 21 May 2022

Dan Kibet Chirchir,

Lecturer, Department of Finance and Accounting

University of Nairobi

Signed: ____________________ Date: May 21, 2022

Dr Winnie Nyamute

Senior Lecturer,

Department of Finance and Accounting

University of Nairobi

ACKNOWLEDGEMENTS

A message of deep gratitude goes out to my immediate supervisor, Mr. Dan Chirchir (CPA, CFA) for his passionate and clear guidance throughout this journey. My thanks also extend to the co-supervisor, Dr Winnie Nyamute for her fruitful albeit brief supervisory engagement. Further, I am eternally indebted to the goodwill showcased by the Chairman of my department, Prof. Cyrus Iraya Mwangi, at my weakest. To the entire Faculty and the University of Nairobi, thank you.

DEDICATION

I dedicate this paper to my father, Al-hajj Abdillahi alias MYM, for his unwavering support, and to my late mother, Margaret Wambui, my siblings, and grandmother.

TABLE OF CONTENTS

List of Figures and Tables

List of Abbreviations

1. ANN - Artificial Neural Network
2. ARIMA – Autoregressive Integrated Moving Average
3. GRU – Gated Recurrent Unit
4. FNN – Feed-forward Neural Network
5. LSTM – Long-Short-term-Memory
6. MAE- Mean Absolute Error
7. NSE- Nairobi Securities Exchange
8. RNN – Recurrent Neural Network
9. RMSE – Root Mean-squared Error
10. R2- R-Squared Measure

Abstract

The relevance and success of financial time-series modelling is predicated upon the inefficiencies of the securities markets. Each forecasting attempt first denies that prices of securities mimic a random-walk process and are hence unpredictable. As traders and investors exploit predictive models successfully, the market efficiency increases, which renders previous strategies ineffective. This prompts investors and researchers to deploy newer and better approaches. While Machine Learning algorithms have already received sparse attention in the Kenyan financial literature, their use has been limited to the rudimentary, vanilla Feed-forward Networks (FNN). In this study, we explored the predictive ability of Recurrent Neural Networks (RNNs) using a sliding window approach on the Kenyan bourse. The population consisted of all the stocks listed on the Nairobi Securities Exchange and the sample was the twenty stocks listed as part of the NSE-20 index through the non-probabilistic purposive sampling technique. We collected ten years' price and volume data from the NSE. Common technical indicators were used to transform the raw features into nine independent inputs. We used the Python coding language and its libraries to perform the predictive modelling in this study. 70% of the data was used for training while 30% of the data tested the models. We then developed two Recurrent Networks, the Gated Recurrent Unit (GRU) and the Long-Short-term-Memory (LSTM) on each stock and benchmarked them against two traditional approaches; The Box- Jenkins ARIMA and the vanilla Artificial Neural Networks (FNN). The R-Squared and Root Mean-Squared Error metrics formed a basis of comparison between the Recurrent models and the traditional models. Through a paired T-test for the difference between means of the R-Squared (95% level of confidence), we found that one of the Recurrent Network models, the GRU, performed significantly better than all other models in the study. The other Recurrent Network (LSTM) was the second-best model by the means of both the R-Squared and the RMSE, followed by the FNN and last, the ARIMA. Further, we explored different parameters and found that a shorter predictive window of {20:10}, and a smaller batch size (40) resulted in better fits for the neural network models on the stocks sample selected. The study has broadened the variety of market research tools available to the Kenyan market participants and has added Recurrent Neural Networks to the Kenyan empirical literature.

CHAPTER ONE: INTRODUCTION

1.1 Background

The rapid development of computing in the last decade (the 2010s) has led to utilising technologies that were not feasible before. Computers have become smaller, cheaper, faster, and more powerful (Rashid et al., 2016). The simultaneous increase in connectivity, data capture, and storage technologies has led to the phenomenon of Big Data, which involves the collection, storage, and manipulation of massive amounts of data to acquire valuable insights (Khan et al., 2014). The increasing power of computers, the diminishing cost of adoption, and data availability have led to the vitalisation of information-related industries, disciplines, and careers.

Artificial Intelligence (AI) is a growing body of knowledge that imputes human cognitive abilities to artificial systems. Artificial Intelligence's base is the rigorousapplication of expertise in diverse fields such as probability theory, calculus, Number Theory, statistics, and the human element, psychology. Intelligence calls for the preference to react to the environment as opposed to following rigid incomprehensive rules. The flexibility brings a level of pragmaticism that traditional rule-based systems cannot achieve. Numerous companies use AI for various tasks, such as translation, image recognition, and natural language processing. Wu et al. (2016) explore the state-of-the-art Encoder-Decoder approach Google uses for its translation service.

The availability of development tools in coding languages such as Python has lessened the prototyping, creation and deployment times of models. The ease of access to such open-source resources has also encouraged the participation of modellers, who are not well-versed in the technical requirements. It is necessary instead, to have a deep understanding of the task and goals of the project and the tools are readily available for almost all conceivable use cases.

Despite the worldwide success of Machine Learning algorithms, developing nations are still lagging in research, training, and adoption. For example, in a review of chat-bot technologies in the financial sector, Bhatti (2019) found that only seven percent of the 42 banks and 41 insurance companies used AI technology. The high approval by customers, as shown by the positive responses, notwithstanding. Similarly,Paul (2020) observed the poor uptake of AI in

Africa and remarked that the continent is decades behind the world. Nevertheless, AI will undoubtedly transform African economies, such as health, education, manufacturing, finance, communication, and agriculture.

Retail investment in Africa is still in its infancy. So are the tools available to investors to navigate developing markets like the NSE, which will inevitably lead to the stagnation of the domestic exchange. Low activity causes capital flight and non-participation of investors in the domestic capital markets.

1.1.1 Recurrent Neural Networks

The extensive attention generated by neural networks has naturally caused their evolution, with each modification improving their performance or suiting an entirely original use case. The elementary unit of a neural network is the Perceptron, which performs a simple mapping of inputs to outputs. We can combine these units to form more sophisticated mathematical structures, such as the vanilla Multi-layer Perceptron (MLP). The MLP is also known as the Feed-forward Neural Network, which hints that information only flows in one direction: forward. While still widely used, the MLPs have been replaced by their variants with special modifications to fit particular tasks suitably. For example, Convolutional Neural Networks are often used to process images, while Recurrent Networks model ordered sequences such as time-series data.

Recurrent Networks are a class of neural networks specifically altered to process ordered or sequential data efficiently. Sequences are data items whose order is critical (Raschka & Mirjalili, 2017). The data exists in various forms, such as speech, language translation pairs, music and time-series. The sub-field of time-series forecasting has benefited from being part of the much broader sequence modelling efforts. Recent developments in the domain have emanated from language translation endeavours used by technology companies, such as Google. For example, the Gated Recurrent Unit was developed to translate English and French (Cho et al., 2014). Because of the flexibility of neural networks, we can repurpose networks developed for other tasks to time-series modelling with ease. Time-series forecasting is a pertinent domain in finance and economics. Correct predictions of stock prices reduce risk and increase returns on investments. Although the argument for Recurrent Neural Networks is strong, researchers have not them applied to model the Kenyan markets.

1.1.2 Stock Prices

Stocks represent units of ownership of securities in an exchange. Investors hold stocks for a variety of reasons, including cashflows, capital gain and speculation. The performance of capital markets can gauge the health of an economy or a sector. Sometimes, however, the stock market performance can be uncorrelated with the broader economy's health when monetary policies are undertaken to prop up the market artificially or due to market inefficiencies. Investors hold stocks for years or even decades, while some traders and speculators buy and sell securities for short periods.

To navigate the stock market, market participants use various methods to select and trade stocks. For instance, some investors use fundamental analysis, stock news and technical analysis, while others use statistical models and advanced quantitative techniques. To achieve higher Jensen's alpha, some participants resort to sophisticated algorithms and infrastructure. On developed exchanges, high-frequency trading is prevalent, where powerful computers undertake millions of deals per second (Aldridge, 2010). Compared to advanced bourses of Europe and North America, the Kenyan stock market is limited in the variety and sophistication of securities and tools available for market players.

1.1.3 Financial Models

Financial Models are mathematical formulations that capture economic phenomena to make actionable generalisations. Given that financial markets are non-deterministic, models do not substitute reality, but guide decision-makers in making informed approximations. A viable financial model encapsulates the underlying patterns of a dataset while tolerating its inability to predict random patterns. A model that learns the random errors from past data is over-fitted and has no predictive power. Modellers test the viability of a model by exposing it to a simulated data environment. Historical data are separated into the training and testing sets where the former mimics the in-sample data, and the latter represents the out-of-sample data. The model learns the training set's patterns to make forecasts. The modeller compares the predicted to actual historical values through accuracy and error metrics to determine fitness. This study takes it one step further by applying statistical inference on the performance metrics after running the models through all stocks in our sample.

1.1.4 Recurrent Neural Networks for Stock Price Prediction

Since stock price data is a quintessential example of sequential data, the case for using Recurrent networks is glaring. Stock data is recorded as a time-series with temporally ordered sequences of periodic prices. Advances in Recurrent architectures have resulted in many powerful models such as the LSTM (Hochreiter & Schmidhuber, 1997) and the GRU, custom-designed for sequence prediction. The LSTM networks have been modelling time-series data for a long time since their inception. However, the relatively new GRU has rivalled the performance of the LSTM while being computationally cheaper (Chung et al., 2014).

A time-series modeller must determine a variety of adjustable parameters to arrive at the best formulation. Examples are the number of past periods to consider, the number of future periods to forecast, and the number of features to use as inputs. To improve the performance of the models, some researchers have combined two or more architectures for stock prediction. For example, Lu et al. (2020) prove that an ensemble of two neural networks, the CNN-LSTM, yield better forecasting than each of them in isolation.

1.1.5 The Nairobi Securities Exchange

The Nairobi Securities Exchange (NSE), formerly known as Nairobi Stock Market, is the sole Kenyan securities market. The exchange was established in 1954, making it the earliest bourse in East Africa. The domestic securities market is under the jurisdiction, overview, and supervision of the Kenyan Capital Market Authority (CMA). The NSE became a full member of the World Federation of Exchanges in 2018, a founding member of the African Federation of Exchanges (AFE), and the East African Securities Exchanges Association (EASEA)(Nairobi Securities Exchange, 2021). The NSE has listed seven indices, 62 company shares, debt and government securities, 13 sectors, and 22 trading participants. It is itself a listed company. As of April 6, 2021, NSE had the following statistics:

Table 1:Nairobi Securities Exchange market statistics

Metric	**Quantity**
Market capitalisation	KES 2,468.64 billion
No. of shares traded	16,407,400 shares
Derivatives turnover	1,320,500
Equity turnover	613,622,803.
Number of indices	7
Listed companies	62
Number of sectors	13
Trading participants	22

1.2 Research Problem

To test the applicability of cutting-edge neural networks to finance, researchers have compared deep learning models with traditional models in statistics. Song (2018) contrasted four AI models to predict the New York Stock Exchange and NASDAQ stock prices. The demonstrated that neural network models outperformed statistical models by up to 5%. Similarly, Keskitalo (2020) studied the difference in predictive performance between econometric and neural network models and found that a neural network architecture (LSTM) performed significantly better than moving average models.

Assuming that a model's performance does not vary on different stock exchanges, it would be of great convenience to appropriate the models fashioned for overseas stocks in the domestic markets. However, this presumption would be disastrous since stocks move differently on each bourse. Consider the study by Hansson (2017), who studied the inter-market performance of different models. The study chose exchanges of three countries: Sweden, Brazil and the United

States of America. The paper compared the performance of an LSTM model in each market. The LSTM performed phenomenally on the Swedish stocks but performed dismally on the developed USA and Brazilian bourses. Hansson's (2017) investigation illustrates the need for a focused and tailored approach to meet the challenges of the domestic market.

Kenyan researchers have also used neural networks to forecast prices on the domestic markets. For instance, Mwikamba (2019) and Wamalwa (2019) compared the Feed-forward neural network to Box-Jenkins models in estimating future inflation and maize prices, respectively. All Kenyan studies have used the rudimentary Feed-forward Networks, which, according to McGonagle et al. (2021), are not as effective as Recurrent Networks in modelling time-series data. Recurrent neural networks are improvements to Feed-forward networks since they have a memory mechanism that results in better predictive performance. The Kenyan forecasting efforts also suffer from uncomprehensive methodologies. For example, in modelling loans performance to determine the creditworthiness of borrowers, Juma (2016) used the entire dataset available as training data and did not validate the model. Without testing, the model is sub-optimal. The lack of feature creation by any domestic studies presents an opportunity to enhance the models.

This study differs from the models by using the more suitable RNNs and comparing them to the popular MLPs and ARIMA. We compared the models developed using R-Squared and statistical inference. Feature creation is also a significant focus to enhance the input variables. We investigated a sliding window approach to forecasting to provide a more realistic simulation. The model only learns the patterns of fixed windows of past data to make multistep predictions.

Owing to the low uptake of neural network algorithms in the Kenyan financial sector (Bhatti, 2019), a necessity arises to harness the power of neural networks in the domestic securities market. The reviewed Kenyan papers on AI models showed the popularity of a relatively outdated neural-based time-series forecasting approach. Using direct price data alone without feature creation and one-step forecasts instead of the more realistic multistep projections further invalidates the formulations by past NSE forecasting endeavours. We also needed thorough feature creation to fortify the inputs. For the mentioned reasons, we have set a firm context for conducting this study.

1.3 Research Objectives and Questions

The main aim of this study is to compare the performance of Recurrent models with traditional models, the Box-Jenkins models and Feed-forward Neural Networks. We transform price and volume data through feature engineering to generate panel data for the models' training and testing inputs. The following questions and hypotheses arise as motivation for this project:

I. What is the best overall model to predict the Kenyan securities by simple averages of the performance metric: the goodness-of-fit measure (R-Squared)?

II. What is the best overall model to predict the Kenyan securities by statistical inference of the R-squared?

III. Is there a significant difference in predictive capability between Recurrent Neural Networks and the two benchmark models, the Box-Jenkins models and Feed-forward Neural Networks?

IV. What are the best parameters and sliding window structures for the models that maximise forecasting accuracy in the Kenyan stock market?

1.4 Value of Study

This research provides a state-of-the-art market research tool bespoke to the need of Kenyan retail investors and brokers. Emerging financial markets requires a healthy quantity of retail investors to attract institutional investors who value the liquidity of exchanges. The reviewed studies have primarily applied the incongruous Feed-forward Network architecture to model time-series data. Recurrent networks have not been featured in Kenyan academic literature.

Academic literature abounds on deep learning models developed for forecasting security exchanges around the world. However, studies on the African markets are few. Understandably, the scant literature on the mapping of the African financial markets may be because of the relative disinterest of the African public in participation in domestic exchanges or the underdeveloped capital markets. Researchers should conduct more studies to give investors and other market participants confidence through information sufficiency. This study adds to the meagre scholarly materials on the application of Recurrent Networks for time-series forecasting

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter explores the theoretical and empirical foundation of the research. In each level, we expound on the financial theories, the concept of the Perceptron, back-propagation, Feed-forward Neural Networks, Recurrent Neural networks, and finally, the empirical review of Kenyan and worldwide studies.

2.2 Theoretical Review

To forecast share prices, we must first understand what drives the stock markets. Price movements result from decisions of individual investors acting in concert. It is natural logic, we assume, that each participant seeks to maximise their utility while minimising risk. The success of the stated assumption depends on whether the investors' behaviour matches the reality of the markets. Such is what the competing schools of thought, Efficient Market Hypothesis (EMH), and behavioural finance seek to understand.

2.2.1Efficient Market Hypothesis

The EMH operates in a perfect world of perfect information, perfect rationality, and costless transactions. Behavioural finance advocates for a more agnostic view. Markets are as inefficient as the imperfection they harbour in terms of investors' behaviour, rationality, and information asymmetry.

Under the assumption of perfect information to all market participants, zero trading costs, and agreement by market participants of the implications of the data to the current and future market, Fama (1970) proposed the Efficient Market Hypothesis. He offered market efficiency at three levels: weak-form, semi-strong form efficiency, and strong-form efficiency, each progressively claiming a more comprehensive capture of information in the prices. The EMH has been one of the most debated topics in finance and economics and has resulted in market studies aiming to disprove this hypothesis. Key to the EMH is the random-walk assumption that prices follow unpredictable patterns. Mutua and Mutothya (2014) studied 18 stocks on the Kenyan Nairobi Securities Exchange. The researchers used serial correlation and Runs test to conclude that the random walk process is not a good descriptor of how the Kenyan stock market moves. Subsequently, Kamau (2018) rejected the random-walk null hypothesis, concluding that the NSE market is non-random.

2.2.2Fundamental Analysis

Fundamental analysis entails using macroeconomic and microeconomic factors to estimate the actual value of an investment. Fundamental analysts act as enhancers of market efficiency by exploiting news events and keeping tabs on crucial ratios and performance metrics changes. Warren Buffett, the famous American billionaire investor, is renowned for applying fundamental analysis to inform his buy-and-hold strategy of companies. Buffet (1984) provided a critical evaluation of the EMH and concluded that the markets are inefficient. He observed that investment funds adhering to value investing had a superior performance that could not solely be attributed to luck. The article emphasised the role of sound investment strategies in successful fund management.

Griffioen (2003) criticises fundamental analysts that they need to track many variables and correctly deduce the effect of these variables on the future cash flows of the security, an incredibly daunting task. This complexity makes wholesome fundamental analysis almost impossible. To reduce risk, investors should consider relevant factors besides the current market structure.

2.2.3Technical Analysis

Fang (2014) defines technical analysis as studying historical market series to predict future market movements. Benjamin (1942) notes that technical analysts simplify historical data by charting, improving clarity, and lessening the information lag. The technical analysts view the market as weak-form efficient. They subscribe to the behavioural finance view that price patterns represent the collective emotions of fear and greed, which are predictable.

Park and Irwin (2004) surveyed past studies addressing the profitability of technical analysis. They divided the analysed studies between early studies and modern studies. The study revealed that technical strategies showed better performance in modern studies than in earlier studies. Technical analysis was suitable for foreign exchange markets in both early and contemporary studies. To formalise and simplify the technical analysis, traders and investors alike use technical indicators, mathematical computations on market variables such as price and volume to indicate potential reversals or continuity of trends (Colby, 2002).

2.2.4Chart Pattern Hypotheses of Securities Markets.

Understanding chart patterns is relevant, as it represents the efforts of visual and subjective modelling unaided by computers. Statistical and neural network models perform this task more systematically and consistently, guided by quantitative methods. The Dow and Elliot wave hypotheses are related approaches for subjective modelling using charting patterns. The overview of these hypotheses provides an excellent contrast to the quantitative models explored in this study.

Charles Dow posited the Dow hypotheses to explain the mechanism of changes in the markets. Dow's work was further enhanced and articulated by Hamilton and Rhea (Benjamin, 1942). The Basic tenets of Dow theory are:

i. The markets are highly efficient.
ii. Second, the bear and bull markets move in distinct phases.
iii. The importance of averages is underscored, and inter-index tracking is advised.
iv. Investors should keep a close eye on volume data to know the stage of the trend.
v. Trend continuity is assumed unless a reasonable assessment provides a contrary view.

The other influential hypothesis is the Elliot wave hypothesis of chart patterns. This technical analysis approach comprises chart patterns that give technical analysts a guide to visualise the price patterns to predict future trends. Traders observe markets seem to move in repeating waves, and its practitioners often swear on the ability of wave analysis to pinpoint turning points in the market. Veneziani (2011) narrates how the legendary Wallstreet trader Paul Tudor Jones made a fortune by predicting the market crash of 1987 using Elliot wave analysis. The theory describes the market as manifesting in nine cycles according to size and duration, with the most extensive lasting decades.

Empirical results of these hypotheses show inconsistency in outperforming other trading strategies. For example, Kim (2019) suggests that an investor is better off picking an equally weighted portfolio in numerous stocks than relying on the Dow hypotheses.

2.2.5Box-Jenkins Models.

Box-Jenkins models are also known as Auto-regressive Integrated Moving Average (ARIMA) models. The creation of these models involves a particular approach known as the Box-Jenkins methodology. The autoregressive model aims to forecast a sequence only based on past values of the same series (Paolella, 2018). For example, an AR (1) model is:

$$y = \beta_0 + \beta_1 y_{t-1} + \varepsilon_t \qquad 1$$

The above equation is an autoregressive model, AR(p), of the first order where p=1, similar to a linear regression equation showing the autocorrelation of value y_t with one of its previous values. The order refers to the number of lagged values used. A generalised form of an autoregressive model of order p with parameters β_i is (Zhang, 2021) :

$$y_t = \sum_{i=1}^{p} \beta_i \, y_{t-i} + \varepsilon_t \qquad 2$$

An AR process of order 0 is equal to the error ε_t and is therefore known as a white noise process. On the other side of the ARIMA model is the moving average processes, MA(q). MA models solve the weakness of the AR process of not accounting for the correlation structure of previous error terms and their contribution to predicting the current value (Zhang, 2021). A moving average model of order q with parameters θ_q has a generalised form of:

$$y_t = \sum_{j=1}^{q} \theta_q \, \varepsilon_{t-j} + u_t \qquad 3$$

The combination of AR and MA processes (ARMA) ensures that both the long-term Memory and the short-term dependencies are captured by the former and the latter. When selecting the number of lags (p, q) that are useful in AR or MA models, autocorrelation (ACF) and partial autocorrelation function (PACF) functions become useful. Autocorrelation refers to the influence of past value with lag k on the current variable y_t as summarized by the following equation:

$$\mathrm{ACF} = \mathrm{corr}(y_t, y_{t-k}) \quad (4)$$

The partial autocorrelation determines the AR(q) component. An ARMA model has the form:

$$y_t = \sum_{i=1}^{p} \beta_i y_{t-i} + \sum_{j=1}^{q} \theta_q \varepsilon_{t-j} + u_t \quad (5)$$

The ARMA model, however, assumes stationarity. A time-series is stationary if it has zero mean and has constant variance. Conversion to stationarity can be done either by detrending or, most commonly, by differencing with d as the order of differencing. The systematic search for parameters $p, d,$ and q is the essence of the Box-Jenkin methodology (1983).

2.3 Neural Network models

Human beings have studied nature and used it as inspiration for discoveries and technological progress. Intelligence was a challenging area of nature that we could not tackle. This difficulty changed when Rosenblatt (1957) proposed a new approach to processing information heavily inspired by biological neurons. Rosenblatt (1957) explains the neuron be a processing intermediary between the inputs and outputs.

2.3.1Single Perceptron as a Linear Regressor.

According to Aggarwal (2018), a perceptron or a neuron is the building block of a neural network. A neuron is a mathematical construct that iteratively connects inputs to outputs through weights, biases, and activation functions. The difference between a neuron and a simple linear regressor is how they arrive at a solution. The linear regressor provides an analytical solution while the Perceptron systematically approximates the weights and biases during the learning process. A perceptron with a unit activation function performs the same purpose as a linear regressor. A single perceptron is, however, a linear classifier. With the poor classification performance of the Perceptron in the 1950s, researchers abandoned their study. The application of activation functions that solved non-linearities in the datasets revived the research. The mathematical definition of a perceptron with a step function is;

$$f(x) = \begin{cases} 1, & W.x + b > 0 \\ 0, & otherwise \end{cases} \quad 6$$

$f(\boldsymbol{x})$ is the output of the Perceptron, which is either 1 or 0 activated by a step function. $\boldsymbol{x}$ is the input, and $\boldsymbol{b}$ is the bias. The above model is simplistic and a poor classifier because it contains only one neuron and uses a step activation function. Adding more neurons and using better activation functions is the power behind neural network architectures.

Figure 1 A single perceptron with three inputs and one output

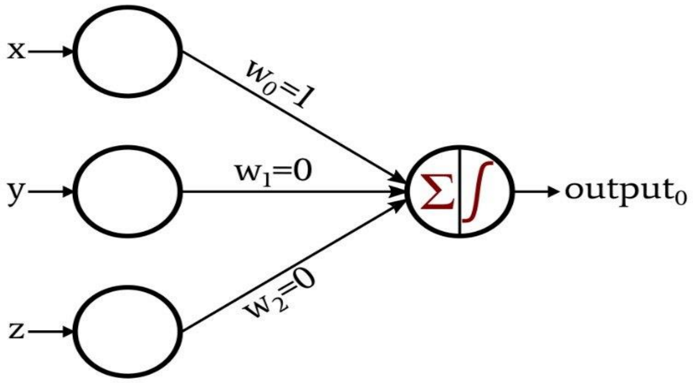


Figure 1 above shows a perceptron as the summation of the dot product of weights and inputs passing through the activation function to get the output. The difference between the linear regressor and the Perceptron is the lack of an activation function.

2.3.2Learning Mechanism of a Perceptron.

A linear regression equation is of the form:

$$f(x) = y = w.x + b \quad 7$$

w is the weight, x is the input, and b is the bias.

The forward pass stage: The Perceptron approximates the output y by first initialising the weight w and the bias b to small random values to calculate output y_1, then, Calculates the loss as the difference between the model output and the desired output that is:

$$loss\ or\ cost = w.x + b - y_1 \quad 8$$

The *back-pass phase.* The Perceptron calculates the slope or the differential of the loss function to minimise the loss then updates the weights as below:

$$w_{new} = w_{old} + \gamma(desired - output) * input \qquad 9$$

Where γ is the learning rate, and w is the weight vector (Maynard, 2020). In machine learning, the learning rate is the pace at which the updating of parameters occurs. The completion of one iteration is known as an epoch.

2.3.3The Feed-forward Neural Network

A perceptron in isolation is relatively weak and cannot give the excellent performances that machine learning practitioners have come to expect of neural networks. Like biological neurons, artificial neurons need to be connected to form a network. A Feed-forward Network or a Multi-Layer Perceptron is the simplest of the neural network architectures. In this form, information flows only in one direction. FNN defines a mapping of desired output to model output, that is $y = f(x, \theta)$ while learning parameter θ (Goodfellow et al., 2016). Aggarwal (2018) notes that the *Tanh* and the *Relu* functions have become more commonly used activation functions.

According to Gurney (1997), in an MLP, the training process of a neuron is scaled to the whole network using the following logic:

i. Feed the input layer with data patterns and let the hidden units evaluate the data to produce outputs.
ii. Compare the outputs to the desired outcome and calculate the cost function as their difference.
iii. Calculate the slope of the cost function of the output node and the hidden node in a process known as gradient descent.

During training, a critical issue of overfitting arises. An overfitted model learns from a dataset so well that it fails to generalise any out-of-sample data (Gurney, 1997). Overfitting defeats the purpose of modelling: learning the specific patterns of a dataset without losing the bigger picture. Gurney (1997) suggests using a validation dataset to track the model to solve this problem. Overfitting occurs if the performance on the training set continues to improve while the performance on the validation dataset starts to deteriorate. Training should stop, and the model parameters that resulted in the best performance on the validation adopted should stop.

Feed-forward neural networks are unsuitable for time-dependent or sequential datasets (McGonagle, et al., 2021). That is why exploring the Recurrent variants of neural networks for time-series analysis is necessary.

2.4 Recurrent Neural Networks

2.4.1RNN with Tanh Activation.

Recurrent neural networks (RNNs) are derived from and are an improvement of FNNs. RNNs have a memory mechanism that enables them to process variable-length sequences in diverse datasets. For this reason, they are applicable in handwriting recognition, time-series forecasting, speech recognition, translation, and many other tasks. Aggarwal (2018) describes an RNN architecture as being an FNN that is unfolded through time. The functioning of learning systems in biology closely resembles that of RNN because the past information is retained and appropriately recalled in response to the environment. The following equation describes the memory mechanism. The hidden state at time t, (h_t) is the output or the hidden state and the current input.

$$h_t = f(h_{t-1}, x_t) \qquad 10$$

Applying an activation function, for example, a hyperbolic tangent (Tanh) which converts the hidden state h_t to values between 1 to -1, we get;

$$h_t = Tanh(w_{hh} * h_{t-1} + \ w_{hx} * x_t \ ) \qquad 11$$

That is, the current hidden state (h_t) is the sum of the product of the weight between the hidden states (w_{hh}) and the previous hidden state(h_{t-1}), and the product of the weight between the input and the current hidden state (w_{hx}) and the current input (x_t) activated using the hyperbolic tangent function (Aggarwal, 2018). Therefore, the current output (y_t) is equal to the product of the weight at the output state (w_{hy}) and the current hidden state described above, thus;

$$(y_t = w_{hy} * \ h_t) \qquad 12$$

Sequential data necessitates respect for the order of items in the dataset (Raschka & Mirjalili, 2017). An example of sequential data is time-series, and natural language, where the order in which the items occur is essential, and any disorder thereof changes the meaning of the data. Raschka and Mirjalili (2017) classify sequential data models as described by the structure and timing of inputs and outputs into five variations:

i. One-to-one models take one input, and the output is a sequence.
ii. The Many-to-one models take a sequence as input with only one output.
iii. Many-to-many models that have both inputs and outputs as sequences.

The vanilla kind of RNN suffers from three difficulties: the vanishing and exploding gradient problem, difficulty in training, and failure to process long sequences (Aggarwal, 2018).

2.4.2Long-Short-term-Memory Network.

Hochreiter and Schmidhuber (1997) conceptualised the earliest form of a Long-Short-term-Memory (LSTM), an architecture that marked the advent of new recurrent networks able to process long-term dependencies. Long-term dependencies are relevant patterns occurring far back in either time or sequence. The vanilla RNN is unable to learn these patterns and therefore is a poor sequential modelling choice. This weakness is due to the multiplicative effect of weights between the hidden layers during back-propagation, resulting in either exploding or vanishing gradients (Raschka & Vahid, 2015). Hochreiter and Schmidhuber (1997) proposed a novel architecture to solve the gradient instability during training. LSTM has a distinct feature: the cell state, which is the long-term memory of the network. Mathematical structures known as gates modify the cell state by controlling what information enters or exits.

Figure 2 Diagrammatic representation of an LSTM layer.

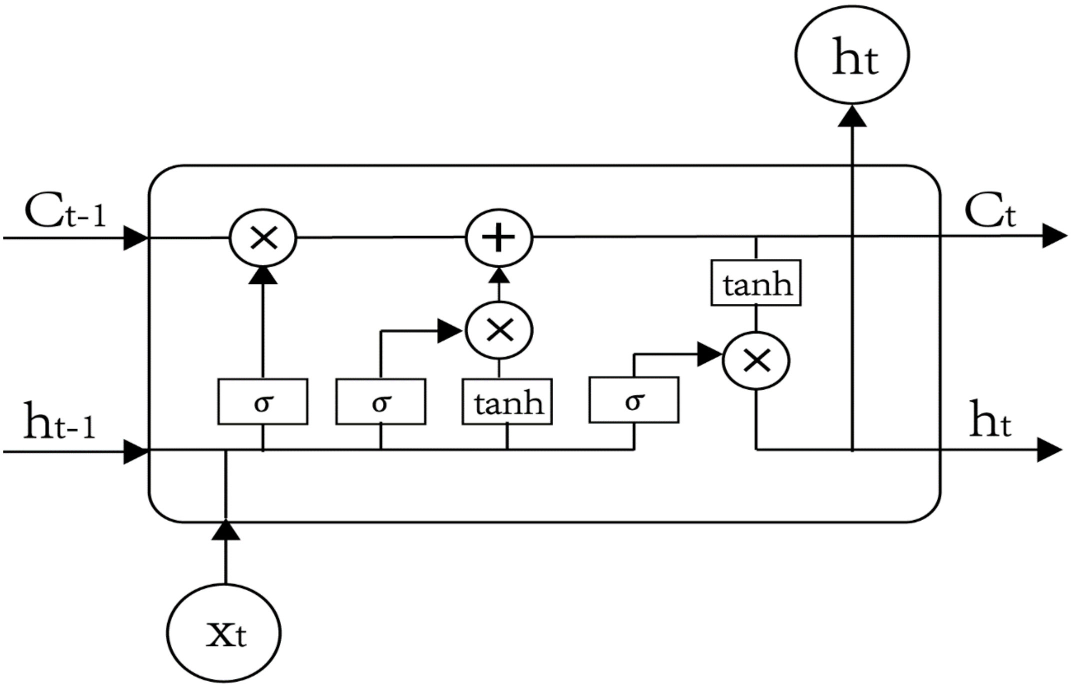


Figure 3 (Raschka & Vahid, 2015) represents the previous cell-state C^{t-1} being converted to the current cell state C^t following modification by the following gates: The forget gate (f) defines what enters and what exits the memory of the cell state. The forget gate is:

$$f^t = \sigma(W_f * [x^t, h^{t-1}] + b_f) \qquad 13$$

Where "f^t" is the current forget gate, and *'w'* are the weights. b_f represents the bias, and σ is the sigmoid function that converts the result of the equation into values ranging from 0 to 1. The second gate is the update gate which is a product of input gate i and node g. This gate performs the actual updating of the values to the cell state. The update gate is:

$$i = \sigma(W_i * [x^t, h^{t-1}] + b_i), \qquad 14$$

and node g:

$$g = \tanh\,(W_g * [x^t, h^{t-1}] \ + \ b_g) \qquad 15$$

The new cell state C^t is, therefore, the old cell state C^{t-1} after pointwise multiplication with the forget gate f^t.

$$C^t = C^{t-1} * f^t + i * g \qquad 16$$

the third and final gate, the output gate (o), decides how to update the value of the hidden state (h^t) to the next timestep as follow:

$$o = \sigma(W_o * [x^t, h^{t-1}] + b_o) \qquad 17$$

and finally, the output, which is the updated hidden state.

$$h^t = o * \tanh(C^t) \qquad 18$$

This ability to remember and discard information based on prediction relevance makes LSTM perform phenomenally in sequential data modelling.

2.4.3 Gated Recurrent Unit.

Cho et al.(2014) introduced the Gated Recurrent Unit (GRU), another recurrent network variant. The researchers trained the proposed neural network to translate between French and English. This version features gating mechanisms akin to those of LSTM. However, While the LSTM contains four gating units, the GRU has only two, toning down the former's complexity. The two gates are known as the update and the reset gates. The simplicity of the GRU architecture as compared to LSTM is attractive to resource-constrained modellers. For instance, Chung et al.(2014) empirically proved that the GRU performed better than traditional RNN in modelling a music dataset, and its performance was comparable to the LSTM. Cho et al. (2014) describe the reset gate as:

$$r_j = \sigma([W_r x]_j + [U_r h_{(t-1)}]_j), \qquad 19$$

and the update gate as:

$$z_j = \sigma([W_z x]_j + [U_z h_{(t-1)}]_j) \qquad 20$$

Candidate activation vector at time t:

$$\hat{\mathrm{h}}_j^t = \tanh([\mathrm{Wx}]_j + [U(r * h_{(t-1)})]_j) \qquad 21$$

finally, the actual activation vector:

$$h_j = z_j h_j^{(t-1)} + (1 - z_j)\hat{h}_j^t \qquad 22$$

The variables are:

r_j is the reset gate of element j.

z_j is the update of element j.

σ is the sigmoid activation function that ranges the values from 0 to 1.

W_r and U_r are the weight matrices.

$h_{(t-1)}$: the previous hidden state.

$\hat{h}_j^t$ is the candidate activation vector.

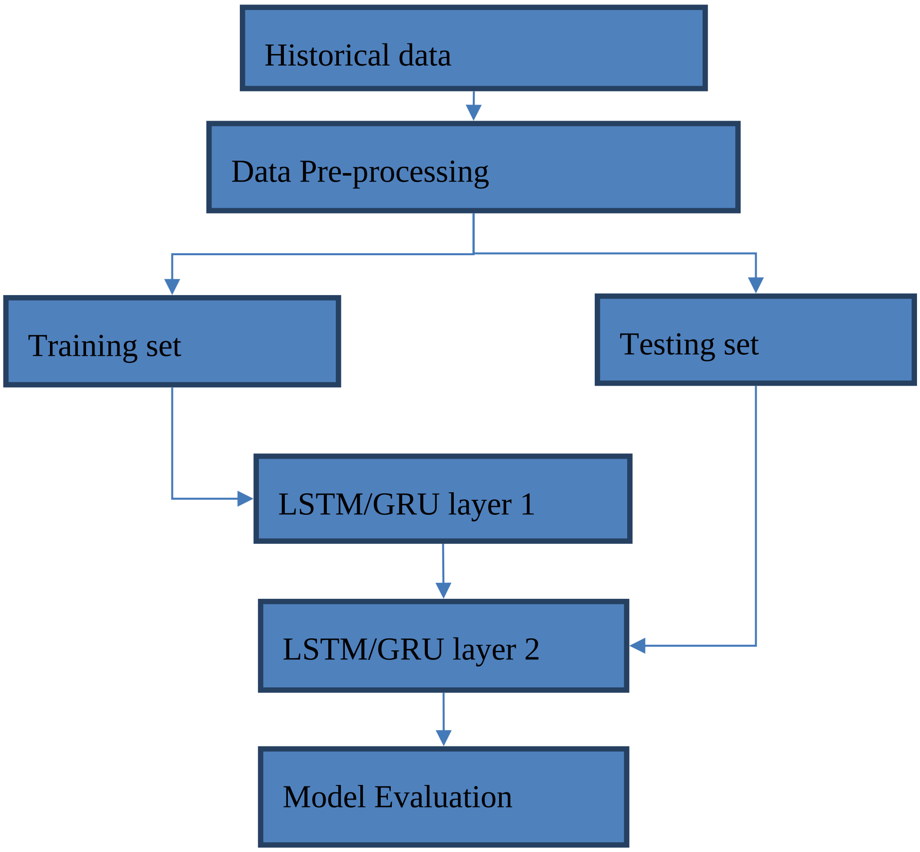
h_j is the actual activation vector.

3.4.4 Implementing the Proposed Models

The Long-Short Term Memory and the Gated Recurrent Unit

Implementing this architecture uses the same libraries as the ANN benchmark and differs only with extra LSTM layers through *Keras*. There are two methods for preparing the proposed LSTM and the GRU network: static and dynamic. The former uses the predicted values as inputs to the next step, while the latter uses only the historical values, as illustrated on the following flowchart. This project adopts the static method.

Figure 3: Static LSTM/GRU Creation Method



2.5 Empirical Review

Hansson (2017) conducted an inter-market investigation into the market-specific implication of prediction using LSTM. The research aimed to find if the prediction capabilities of LSTM varied by the type of market. The study deployed an ARIMA variant as the benchmark. The researcher evaluated the model using the sum of squared residuals. The three markets investigated were the small Swedish market, the emerging market of Brazil, and the developed market of the United States of America. Hansson revealed the LSTM model outperformed the statistical model in the small Swedish exchange. Still, the benefits were less remarkable on emerging and matured markets of Brazil and the USA, respectively. This study underscores the need to conduct studies on each market independently to find the unique characteristics affecting the applicability of the models under consideration. Whichever model(s) chosen for a particular market should not be assumed to work universally for all exchanges.

Song (2018) made a comparative investigation into four Machine Learning models, namely LSTM, GRU, Support Vector Machine(SVM), and XGBoost. The research used twenty public companies from the New York Stock Exchange (NYSE) and NASDAQ. The study also created features using technical indicators, such as the Relative Strength Index, the Average Directional Movement Index, and the Parabolic Stop and Reverse. Song discovered that the Recurrent network models, especially GRU, outperformed the SVM and XGBoost models by up to 5%.

Juma (2016) proposed a Neural Network back-propagation model for appraising loan applications at KCB bank. She used data from the bank's financial information systems to extract financial ratios of corporate clients, then utilised the ratios as inputs to her model. The model aimed to classify the loan as either a performing or a non-performing loan. The resultant model was a binary classifier activated using the sigmoid function. Juma (2016) avoided model evaluation altogether and only trained the model. The researcher was oblivious to the concept of over-fitting. With only 16 respondents, the study could have utilised less sophisticated but more effective models for the problem at hand, such as support vector machines and random forest, instead of Neural Networks, which are more data-intensive.

Mwikamba (2019) used the Gross Domestic Product and oil prices data as inputs to a vanilla ANN with inflation data as output. The research held the ratio of 70:20:10 of the data for training, testing, and validation, respectively. A comparative base model, ARIMA, was compared and contrasted against the ANN. Using the RMSE as a performance measure, the research confirmed that the ANN outperformed the ARIMA model by 100.6% of the base model. There are two weaknesses of the model developed. First, Mwikamba (2019) uses a trial-and-error method to determine the number of neurons and the number of layers in the model. There are better parameter optimisation tools in languages like Python, whose results would have converged to create a better model. Second, the research does not utilise modern and more effective Neural Network architectures like Recurrent Neural Networks, limiting the model's applicability.

Wamalwa (2019) used the same Feed-forward ANN architecture with univariate data input to predict maize prices in Kenya. With Root Mean Squared Error (RMSE) as the performance metric, the study compared univariate and multivariate inputs, linear models versus Neural

Networks. The research found that univariate input produced better results than multivariate inputs and that the FNN model outperformed linear models, such as ARIMA. The study, however, conducted no feature engineering, which could have otherwise improved the model.

Wanjawa (2014) modelled an FNN back-propagation algorithm with Mean Absolute Percentage Error (MAPE) as the performance metric. Wanjawa selected three stocks in the NSE, and the data was fed as input. Wanjawa split the data between training and testing at a ratio of 80%:20%. The research used a manual hyperparameter search for the number of neurons per layer. The study also tested their model for overseas markets and used SVM as a comparative model. State-of-the-art RNN architectures have consistently outperformed FNN architectures for sequential data because they have built-in sequence handling mechanisms (Raschka & Mirjalili, 2017). RNNs are therefore ideal for time-series forecasting and Natural Language Processing (NLP). Using unsuitable architecture in time-series forecasting results in a sub-optimal model. offered forecasting for one time step only, which does not cater to practical situations where multistep forecasting, say, a week's daily forecast prices, is desired. The validation data is just as key as the testing data to check the model during training to reduce the risk of over-fitting. The lack of feature engineering and only using price as the input limits the model's predictive capability.

2.6 Summary and Implications

The current literature on the Kenyan markets delineates the scarcity of quality state-of-the-art forecasting models, essential toolkits for market analysts and individual retail investors alike. This chapter investigates related studies from other researchers and concludes that there is value to be mined from these studies, but the resultant models face the following shortcomings:

i. The reviewed models used Feed-forward networks instead of Recurrent Network architectures.
ii. The models developed for the Kenyan markets utilised manual parameter optimisation instead of the appropriate grid searching techniques.
iii. All the models for the Kenyan markets only performed a single time step forecast which could inflate the performance but not applicable for users desiring more realistic look-ahead projections.

iv. None of the reviewed models explored the effect of the sliding window on the performance measure.

This research solves the listed problem by:

i. Using Recurrent Network architecture, which is much more suitable for handling temporal data.
ii. Perform automated parameter and hyper-parameter optimisation to choose the best settings for the model.
iii. By Exploring two RNN architectures, the research provides a more comprehensive assessment of the state-of-the-art forecasting techniques than previous studies in Kenya. The two variants are the Long-Short-term-Memory (LSTM) and the Gated Recurrent Unit (GRU).
iv. Performing feature engineering then uses the features generated as inputs to the models. Such features include technical indicators such as RSI, OBV, and MACD.
v. Performing a multistep forecast using a sliding window approach to enable users to determine the desired number of time steps to be predicted.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

This project seeks to establish the viability of recurrent neural networks in forecasting the price of indices in the Nairobi Securities Exchange. The project compares two Recurrent models and two traditional models. The Recurrent networks are the LSTM and the GRU, while the benchmark are the Box-Jenkins models and the vanilla Feed-forward Neural networks. This chapter details the course for effective comparison of these models, the research design, data specifications, the evaluation criteria, software, programming language, tools specifications, the creation of the baseline models, and finally, the procedure for implementing the proposed models.

3.2 Research Design

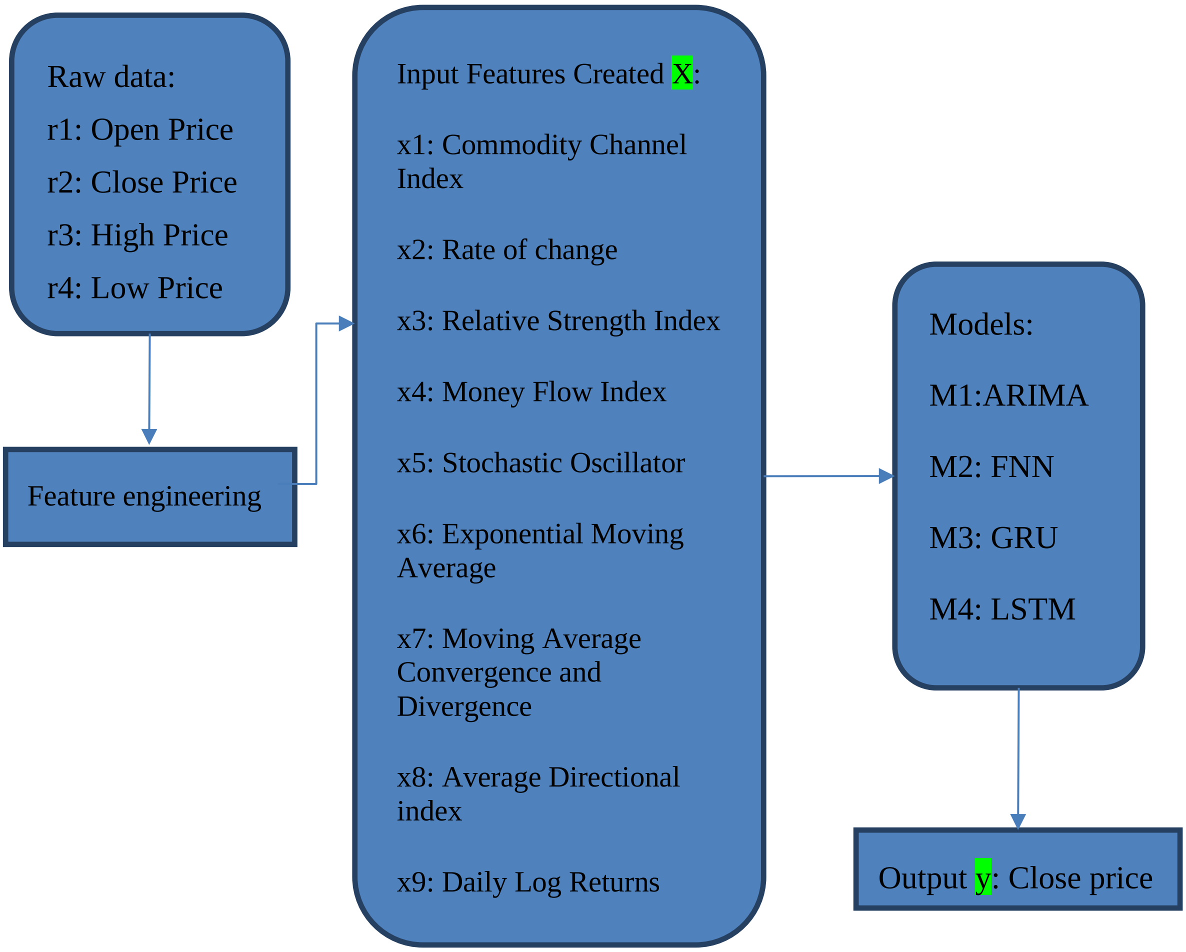
This research aims to provide solutions to problems or answers to existential questions in the business and society. Kothari (1985) categorises research into four levels. The first categorisation is whether the study is descriptive or analytic. The former explains the state-of-affair without the need to provide a solution. However, analytical research provides a rigorous critique of the current state, often with the aspiration to move to a better state. The second categorisation distinguishes between applied and fundamental studies. Applied research solves an immediate issue faced by society or business, while fundamental research generalises establishing a theory or strengthening an existing one. Kothari (1985) provides a third grouping of research projects: quantitative versus numerical measurement is imperative or qualitative. The fourth category differentiates between conceptual research design and empirical design—the former deals with abstract ideas, for example, pure mathematics and theoretical physics disciplines. The latter is based on issues drawn from real-life experiences.

Research design rarely conforms to only one of these categories, but a combination thereof that ensures congruity to the problem at hand. Therefore, this project is of applied, analytic, quantitative, and empirical design. Chapter 2 highlighted the deficiencies of the models developed in past studies by researchers to solve financial forecasting problems in the Kenyan financial markets. This project uses an empirical approach to determine the propriety of the RNN over the more commonly studied ARIMA models and the vanilla Feed-forward Neural

Networks. We compare performance measures to justify using RNNs over other financial forecasting techniques.

3.2.1 Conceptual Framework

Figure 4: Conceptual Diagram



The diagram above summarises the entire study. We collected the raw data from the Nairobi Securities Exchange. Through feature creation, input features comprising commonly used technical indicators was formulated from the raw features as independent variables. The dependent variable is the 'Close Price'. The features used by the four models for training and validation in the ratio of 7:3.

3.2.2 Software and Tools

The choice of a programming language is paramount to the success of a Machine Learning or a Data Science project. The primary coding language used by the reviewed researchers on the Kenyan financial markets is C#. However, the Python programming language has achieved mainstream support in the Data Science and Artificial Intelligence domains because of the ease of use without losing its general-purpose edge (Müller & Guido, 2016). The extensive usage of the language has led to numerous libraries for specific tasks, such as data analysis, Machine Learning, and statistics.

This study uses Python coding language with the following libraries:

i. pandas 1.3.0 for data analysis and manipulation, Scikit-learn Machine Learning.
ii. Darts library for time-series forecasting and analysis
iii. SciPy 1.7.0 for statistical analysis.
iv. NumPy for array manipulation.
v. Keras built on top of TensorFlow for neural network building and training.
vi. Technical analysis library for feature engineering.
vii. Matplotlib for data visualization.

The Jupyter Notebook is the primary development environment. These tools and libraries form a crucial arsenal for designing, testing, and evaluating the models to answer the research questions.

3.2.3 Implementing the Base models

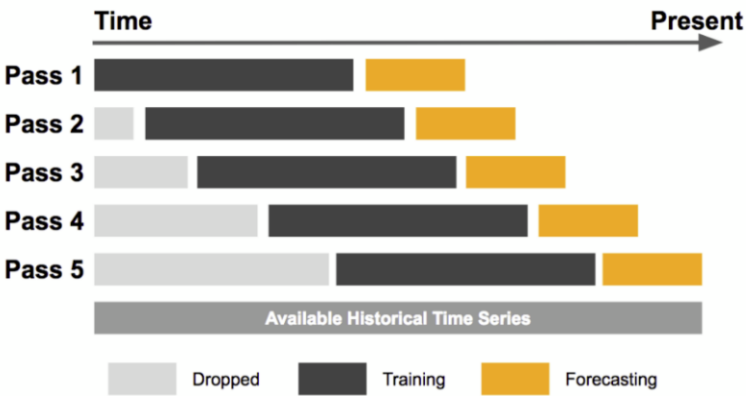
Auto-regressive Integrated Moving Average

Box-Jenkins's methodology guides the ARIMA modelling through model identification, parameter tuning, and validation. We create an autoregressive moving average model using the Python libraries with p, d, q as function parameters. The input variable is the "close price". The grid-searching technique is used to find hyper-parameters. We fit the ARIMA model to the training set. We generate the predictions and compare with the test data using the R-squared. The Box-

Implementing the Vanilla Artificial Neural Network

The second benchmark model is the FNN, the crudest neural network architecture. A pre-processed multivariate dataset with ten features is required for implementing this architecture. The price information for the input and output is the typical price, the mean value of the close, high and low prices. We train the model on the dataset in a rolling or sliding window approach. The following demonstrates the concept of a rolling window of inputs and forecasts (Bell, 2018).

Figure 5: Training-forecast data setup



The study experiments with the following three sliding window structures: 20:10; and 30:15, which are compared through a one-tailed T-test. The Keras library provides a simple way to define and build neural networks. The *sklearn* library provides the tools for the evaluation of the resultant model.

3.3 Population and Sample

The population in this study is all the stocks listed on the Nairobi Securities Exchange. There are 62 publicly traded shares on the Nairobi Securities exchange. This study applies a non-probability purposive sampling method. When sampling a dataset, the researcher can choose crucial variables to be considered. The companies whose stocks we considered must satisfy two criteria:

a. The stock should be part of a stock index, preferably NSE-20, based on a market value other than the all-share index.
b. The stock should be over five years of available historical data.

The duration varies in numerous studies dictated by the availability of data; for instance, Song (2018) and Hanson (2017) use eight years of daily stock prices, while Wanjawa (2014) uses only five years of daily information. The granularity of input data in hours or even minutes can increase the utility of models for intraday traders. This research therefore uses price data spanning 10 years. We split each stock's daily price data by a commonly used ratio of 7:3 for training and testing. We fit the four models with the training data and then validate using the testing data of each stock.

3.3.1 Data Collection

The Nairobi Securities exchange allows access to historical datasets after subscription according to the number of days desired. However, Wamalwa (2019) notes other data sources, such as the NSE licensed data vendors listed on the exchange website. This project uses secondary data sourced from the NSE, with five raw features. These are; the Open price, the highest price of the day, the lowest price of the day, the close price, and the volume data. These facets of the data derive the modified values using technical indicators as the ultimate inputs to the model.

3.4 Data Analysis

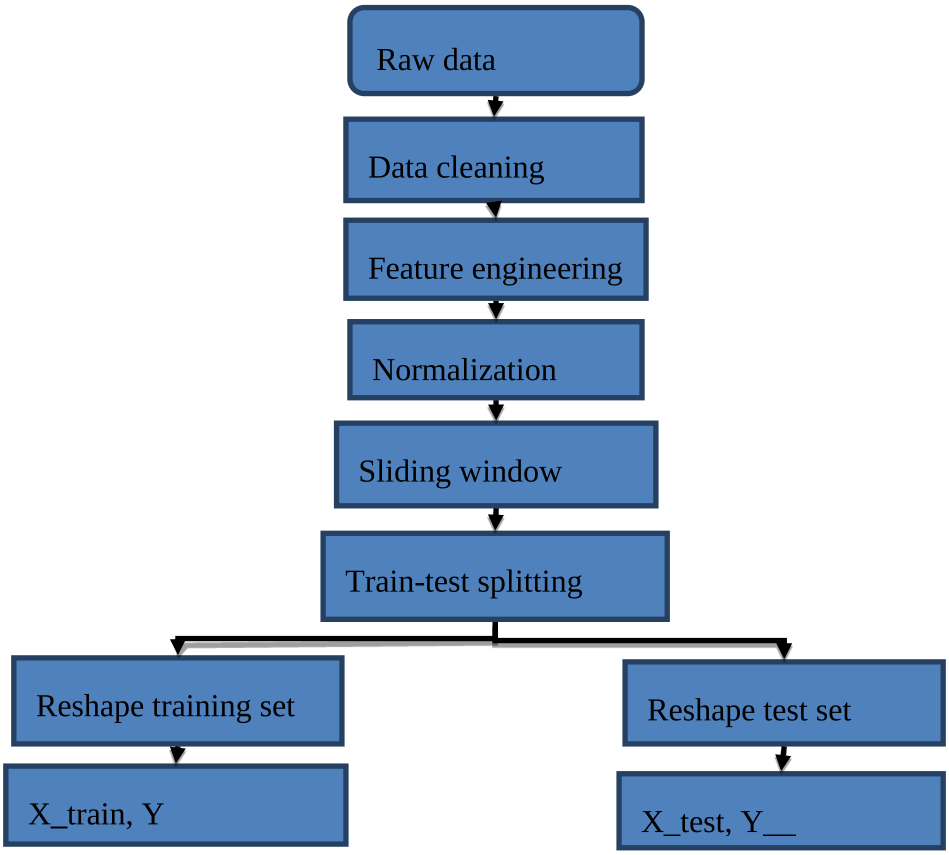
3.4.1 Data Preparation and Feature Creation

Data preparation refers to the modification of datasets to suit the need of a modelling task. This process also helps acquaint the researchers with the dataset. Changing the dataset also enables the refinement of inputs into the desired model for training. Pyle (1999) sees data preparation as part of a data mining process rather than a separate task. Data preparation also involves data cleaning, removing errors and outliers, type modification, normalisation, and train-test splitting. The following figures summarise the data preparation process. The first step is to identify and calculate the input variables.

Table 2: Input Variables

Variable	Description
Open price	the price traded at the first tick of the day
High price	the highest value of stock price for the day
Low price	the lowest value of stock price for the day
Close price	the price traded at the last tick of the day
Volume	the number of stocks traded per day

Figure 6: Data Pre-processing Summary



The Technical Indicator values are modifications and combinations of features in table 2. These indicators are:

i. Commodity Channel Index
ii. Rate of change
iii. Relative Strength Index
iv. Money Flow Index
v. Stochastic Oscillator
vi. Exponential Moving Average
vii. Moving Average Convergence and Divergence
viii. Average Directional index
ix. Daily Log Returns

3.4.2 Performance measures

A performance measure is a yardstick against which a model is evaluated. It is crucial to determine whether a formulation achieves the purpose set out by modellers. An appropriate measure also provides a basis for comparing competing models. The choice of a success metric is highly dependent on the purpose of the model. For example, logarithmic loss and confusion matrix are natural metrics of choice in classification algorithms, while error measures gain prominence in regression. Some performance metrics can be used in classification and regression domains, while others, such as the Sharpe ratio, can only be used in financial models.

Mean error variants and the R-squared are widely used by researchers in models that have continuous variables as inputs. All the studies reviewed in the second chapter have used one or a combination of these metrics. For instance, they are the mean absolute error (MAE), the mean-squared error (MSE), and the root Mean Absolute Error (RMSE). The MAE is formulated as follows (Bajaj, 2021):

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{23}$$

Where:

y_i : ground-truth/ desired value

$\hat{y}_i$: predicted value from the regression model

N: number of datums

The mean squared error is the average squared difference between the desired values and the predicted values. It is a popular metric because it is differentiable, but Bajaj (2021) remarks that it is more prone to outliers than other metrics due to the squaring effect. The root-mean-squared-error (RMSE) retains differentiability and solves the error exaggeration problem by using the square root of MSE as below.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (24)$$

The R-squared measure is an alternative to the error measures by measuring the model's usefulness in explaining the patterns in a dataset. The R-squared indicates what portion of the total variation is explained by the model. Effectively, it shows the goodness of fit of a model. The formula for this measure is:

$$R^2 = 1 - \frac{Unexplained\ variation}{Total\ variation} \quad (25)$$

3.4.3 Hypothesis Testing

To determine the superiority of one model over the other, we use a one-tailed T-test for the difference between two means of the performance measure. The R-squared is the primary metric of comparison among the four models. Each model is juxtaposed against each other to find which model is statistically more predictive than the others, given a 95% level of confidence. We rank the models according to their performance and conclude whether Recurrent networks are better than the two traditional models in this study.

3.5 Procedure

1. Installation: Install all software packages and libraries on the personal computer, including python 3.6, TensorFlow, Sci-kit Learn, Pandas, and NumPy.
2. Import dependencies: Open the web-based Jupyter Notebook coding environment and import installed libraries and dependencies. These are tools necessary for data handling, visualisation, model building, and feature engineering.
3. Data cleaning:
    3.1. Import data from Excel using the *Pandas.read_csv ()* function.
    3.2. Impute missing values with a week's moving average.
4. Feature generation and data preparation:
    4.1. Engineer 10 features from the imported dataset by applying the technical analysis library. The features generated must be stationary.
    4.2. Split the data into the training, testing sets in the ratio 70:30, respectively.
    4.3. Scale the training features and the testing sets separately to values between 0 and 1 using the *MinMaxScaler* to improve the training speed.
    4.4. Define the sliding window, for example: use ten past values to predict five future values. The input data has two dimensions of shape (10,10). The first represents the number of features, while the second is the number of past timesteps considered. However, the output data is one-dimensional, with the shape (5) consisting only of the future forecasting values.
    4.5. Convert both the training and testing sets to NumPy arrays to be fed into the models
5. Model creation:
    5.1. Construct the LSTM and GRU models by stacking layers using Sequential, LSTM, Dropout, and Dense methods, then specify the parameters.
    5.2. Compile the model specifying parameters to complete the creation.
6. Model training:
    6.1. Train the model using the features generated.
    6.2. Fit the model to the training dataset.
7. Forecasting:
    7.1. Use the model to forecast the values of the testing dataset.
8. Evaluation and comparison:

8.1. Evaluate the model using performance measures, such as the root-mean-squared error and the R-Squared. Compare the LSTM and GRU models to the benchmark through statistical tests.

CHAPTER FOUR: ANALYSIS, RESULTS, AND DISCUSSIONS

4.1 The Box-Jenkins Benchmark

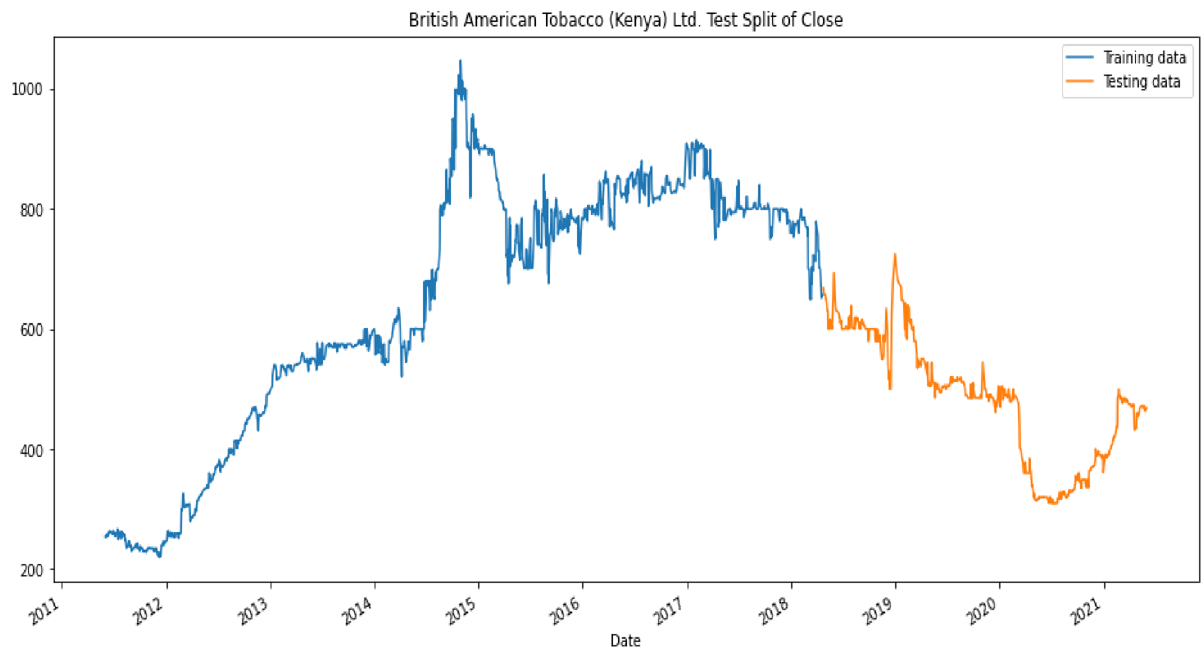
The ARIMA model without seasonality or exogenous variables is a linear benchmark to an otherwise stochastic line-up of neural network models. This model also acts as a methodological comparative to justify the sliding window approach to data pre-processing adopted by the neural network variants. Ten years' daily historical data was sourced through an NSE licensed data vendor Synergy Solutions ltd. The environment used for coding is Jupyter Notebook environment. The data preparation for the ARIMA is minimised only to satisfice ARIMA- specific requirements. We clean the data using the Python programming language and handy statistical packages and only use the 'Close' price to fit and test the model.

We split the data into training and testing sets in the customary ratio of 70:30. Although the two sets are historical data, the training simulates in-sample data while testing set proxies out-of-sample future data not available at the split date. For more robust modelling, the two sets should not contaminate each other to avoid data snooping. Test set integrity is paramount, for it is the only way a model's performance evaluated for the decision to improve, deploy, or discard. Adopting a model carries a risk of direct monetary loss in financial applications, which should caution modellers against cheating with the test set. Data snooping incidents can sometimes happen due to ignorance. This issue has been considered by most pre-processing methods, such as Scikit-Learn's "*fit_transform*" and "*transform*" calls, which should only be applied to the train and test sets, respectively.

We subject the data to tests to determine statistical properties. The most relevant property for a Box-Jenkins model is stationarity, ascertained through the Augmented Dickey-Fuller test. The null hypothesis is that the series of historical data is stationary. To determine the number of ARIMA parameters (p, d, q), we use autocorrelation functions, PACF, and ACF plots. The "*auto_arima*" function from the statistics package '*pmdarima'* is a grid-searching tool for the Box-Jenkins parameters. We use the following statistical packages to fit and test the model.

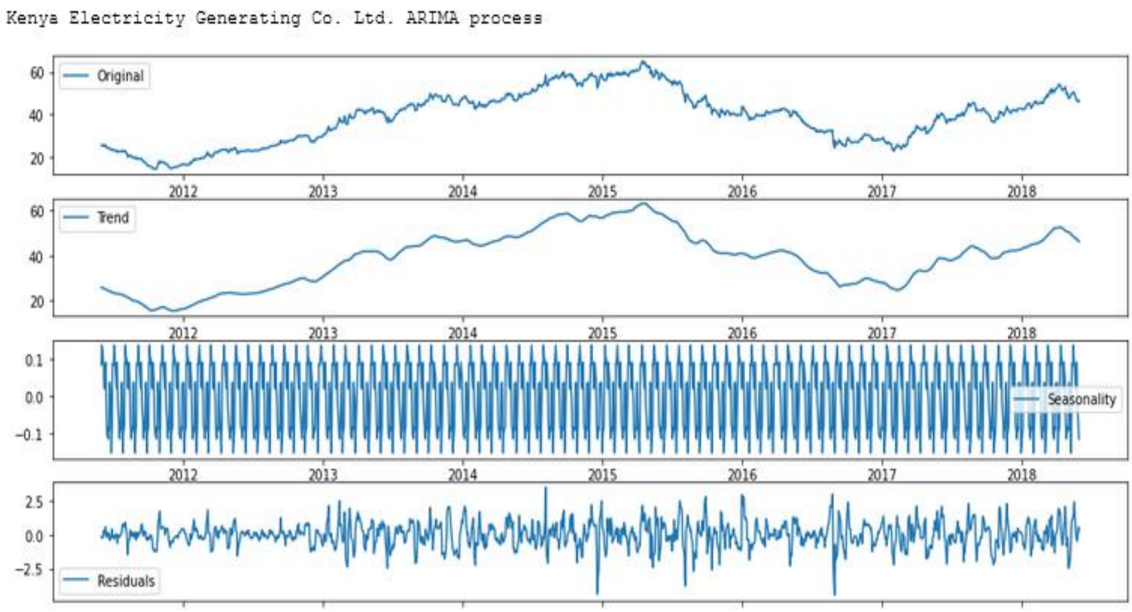
Package	Task
NumPy	Array manipulations
Matplotlib	Visualizations
Statsmodels	Model fitting and statistical tools
Pandas	Data frame manipulations, importing
Pmdarima	Model selection

Figure 7: Example of Train-Test split on Close Price



We can visualise trend and seasonality of stock prices by decomposing the time-series into its parts. For example, the plot below shows the decomposition of 'Close Price' of Kenya Electricity Generating Company to seasonal, trend, and residual components.

Figure 8: Time-series Decomposition

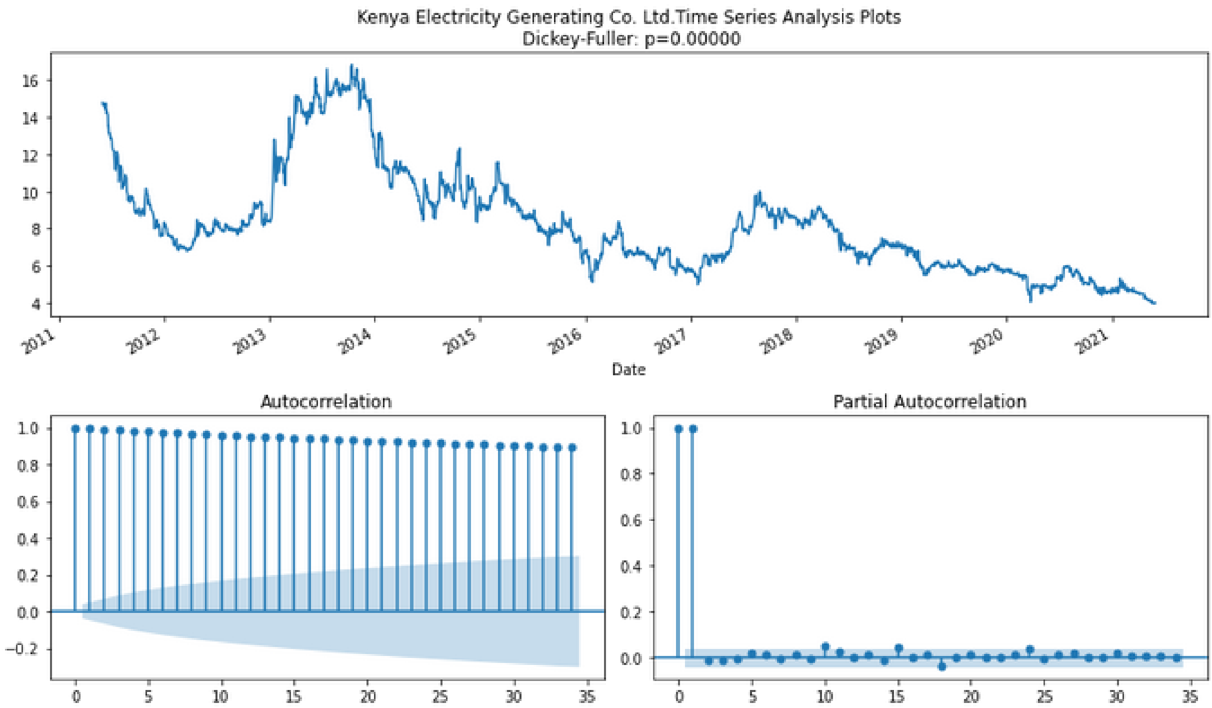


The '*auto_arima'* function performs a grid search for the Box-Jenkins parameters on the *train* set and outputs the following iterations for Kenya Electricity Generating Company Ltd.

```
Kenya Electricity Generating Co. Ltd.
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=-249.875, Time=0.45 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=-252.377, Time=0.20 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=-254.542, Time=0.20 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=-254.395, Time=0.25 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=-253.761, Time=0.13 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=-253.071, Time=0.25 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=-252.837, Time=0.35 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=-251.035, Time=0.35 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=-255.981, Time=0.17 sec
 ARIMA(2,1,0)(0,0,0)[0]             : AIC=-254.528, Time=0.54 sec
 ARIMA(1,1,1)(0,0,0)[0]             : AIC=-254.288, Time=0.14 sec
 ARIMA(0,1,1)(0,0,0)[0]             : AIC=-255.830, Time=0.12 sec
 ARIMA(2,1,1)(0,0,0)[0]             : AIC=-252.484, Time=0.18 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0]
Total fit time: 3.361 seconds
Kenya Electricity Generating Co. Ltd. (1, 1, 0)
```

The ACF and the PACF plots useful in the manual selection of parameters are shown below.



4.2 ARIMA Results and Discussion

After we fit the model to the training set, we set it to predict the entire duration of the test set. We compare the model with the actual values of the out-of-sample data.We scaled both the forecasted values and the test set using the "*MinMaxScaler*". We calculate the Root Mean Squared Error and the R-Squared metrics for each stock as follows.

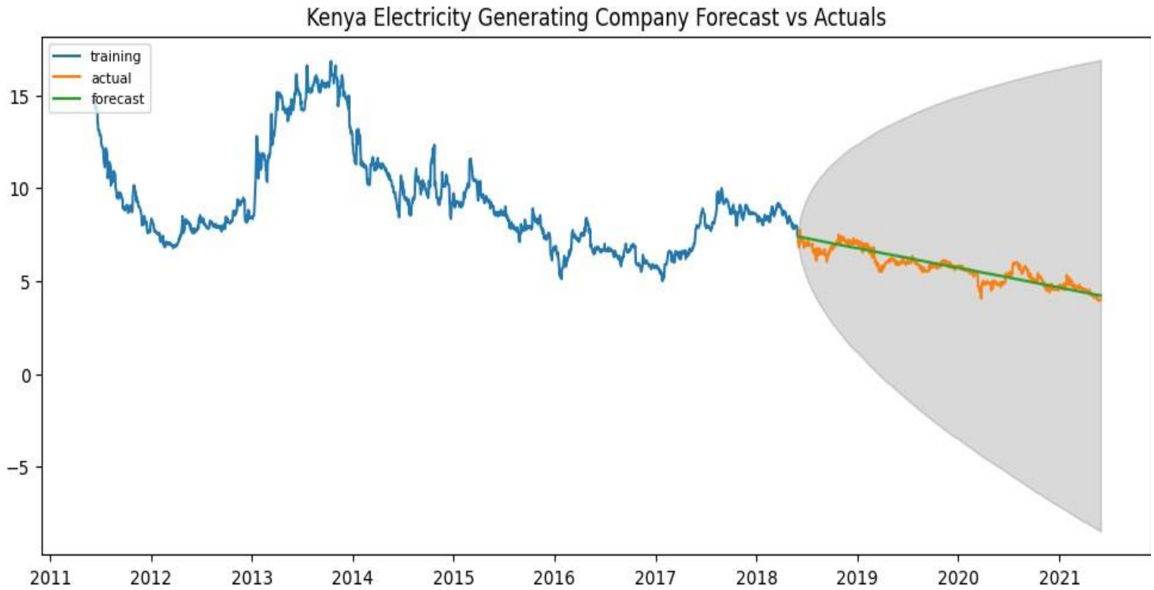
Figure 9: ARIMA Results

ARIMA Results			
sc	**RMSE**	**R-squared Scores**	**Order(p,d,q)**
Absa Bank Kenya PLC	0.010265536	0.295481241	(0, 1, 1)
Bamburi Cement Ltd	1.078411259	-3.368950717	(0, 1, 1)
Britam Holdings Ltd	0.084593813	-23.11300067	(2, 1, 2)
British American Tobacco (Kenya) Ltd	3.127352812	-8.158690466	(2, 1, 4)
Centum Investment Co. PLC	0.174071561	-6.318519063	(2, 1, 2)
Co-operative Bank of Kenya Ltd	0.061043649	-9.881562116	(1, 1, 0)
Diamond Trust Bank Kenya Ltd	1.286655849	-8.459199431	(3, 1, 3)
East African Breweries Ltd	0.611152218	-6.382944108	(1, 1, 1)
Equity Group Holdings Ltd	0.134466494	-4.725069307	(0, 1, 1)
KCB Group Plc	0.11430917	-3.878136607	(0, 1, 4)
Kenya Electricity Generating Co. Ltd	0.003924963	0.791606014	(1, 1, 0)
Kenya Power Lighting Co. Ltd	0.01307377	0.225968053	(1, 1, 0)
Kenya Reinsurance Corp. Ltd	0.021443042	-9.541317544	(1, 1, 0)
Nairobi Securities Exchange Ltd	0.040336707	-4.499412598	(0, 1, 0)
Nation Media Group	0.358926663	-1.662229637	(2, 2, 2)
NCBA Group PLC	0.068912653	-2.162361615	(0, 1, 1)
Safaricom PLC	0.049045418	-0.36669269	(1, 1, 1)
Stanbic Holdings PLC	0.119501023	-0.679412536	(2, 1, 2)
Standard Chartered Bank Kenya Ltd	0.264668883	-1.480213993	(1, 1, 0)
WPP Scangroup Ltd	0.049640246	-0.131268005	(0, 1, 3)
Average	**0.383589786**	**-4.67479629**	

The ARIMA model without seasonality is a trend summarising model. The model determines the direction of the training series and interpolates the results over the testing period. Thus, the more similar the in-sample values are to the out-of-sample values, the higher the accuracy. An example of a high accuracy situation is that of Kenya Electricity Generating Co. Ltd as plotted next.

The p-value of the first lag is the only statistically significant lag at a 95% levelof confidence. The fitted model is an auto-regressive model with one order of differencing.

Figure 10: The Best ARIMA Prediction (Kenya Electricity Generating Co. Ltd)



The figure above shows the training, testing, and forecasted values of Kenya Electricity Generating Co. Ltd superimposed on the same chart. The Arima model forecast is most accurate when the future direction does not deviate much from the past trend. We can therefore interpret this model as a trend-sensitive naïve model. With Kenya Electricity Generating Co. Ltd predictions, all the predicted values lie within the confidence interval. In various instances, however, the complexity of the time-series requires equally complex methods.

4.3 The Feed-Forward Neural Network Results and Analysis

The methodology adopted by this study was adhered to for the deep learning models starting with the Feed-forward Networks. We pre-processed the data according to the model at hand. We use the same train-test split to separate the dataset. An essential yet tricky task was getting the data in the right shape and format. For instance, data normalisation enables faster training and reliability of the models on the test set and final deployment. Feature generation was also a significant focus of this study. We generated nine features using commonly used technical price and volume indicators.

The base window used in this study was {20:10}, interpreted as; considering the patterns present in the generated dataset of the last month to forecast the followingfortnight's daily price on a rolling basis. The input data must be in the shape of (number ofsamples, number of look-back periods, number of features). Having separated the training set from the testing set, we further structure the datasets to reflect the rolling window concept of our problem through a method shared by Brownlee (Brownlee, 2017b). An example of this scaled data layout for Safaricom Ltd. is depicted as follows. (Note that this is only one of the 1736 sequences in the training set of Safaricom plc).

Figure 11: Training Inputs and output sample

TRAINING INPUT SEQUENCE[1]

	Close	MACD	CCI	RSI	Differenced_EMA	STOCH%d	ROC	ATR	Log Return
0	0.041112	0.526632	0.805556	0.618303	0.527165	0.949534	0.412773	0.008356	0.468635
1	0.044740	0.526632	0.805556	0.618303	0.527165	0.949534	0.412773	0.008356	0.468635
2	0.041112	0.526632	0.805556	0.618303	0.527165	0.949534	0.412773	0.008356	0.373641
3	0.043531	0.526632	0.805556	0.618303	0.527165	0.949534	0.412773	0.008356	0.452865
4	0.045949	0.526632	0.805556	0.618303	0.527165	0.949534	0.412773	0.008356	0.452618
5	0.044740	0.526632	0.607143	0.618303	0.527165	0.949534	0.412773	0.008356	0.405429
6	0.044740	0.526632	0.607143	0.618303	0.527165	0.949534	0.384117	0.008356	0.421138
7	0.049577	0.526632	0.970588	0.805918	0.527165	0.948835	0.450981	0.008356	0.483612
8	0.050786	0.526632	0.800000	0.832156	0.527165	0.947870	0.440981	0.008356	0.436607
9	0.049577	0.526632	0.634615	0.713400	0.527165	0.946742	0.412328	0.008356	0.405669
10	0.053204	0.526632	0.875000	0.814487	0.527165	0.946629	0.450200	0.008356	0.467368
11	0.049577	0.518472	0.333333	0.565804	0.506475	0.944100	0.421879	0.011154	0.374907
12	0.049577	0.511885	0.333333	0.565804	0.504520	0.941735	0.384117	0.005808	0.421138
13	0.049577	0.506538	0.375000	0.565804	0.502920	0.938888	0.374856	0.001381	0.421138
14	0.048368	0.499652	0.230769	0.471373	0.496620	0.939316	0.374821	0.000041	0.405609
15	0.045949	0.489769	0.041667	0.331537	0.486474	0.940600	0.328976	0.001207	0.389901
16	0.049577	0.491392	0.666667	0.569517	0.503129	0.942376	0.384117	0.004441	0.467903
17	0.049577	0.491745	0.666667	0.569517	0.501782	0.943724	0.384117	0.000642	0.421138
18	0.053204	0.498944	0.944444	0.724509	0.515653	0.944696	0.412004	0.003809	0.467368
19	0.048368	0.491877	0.388889	0.451615	0.492065	0.944529	0.384117	0.008665	0.359379

TRAINING OUTPUT SEQUENCE [1]

	Close Price
0	0.048368
1	0.048368
2	0.047158
3	0.049577
4	0.042322
5	0.048368
6	0.047158
7	0.045949
8	0.048368
9	0.053204

The Augmented Dickey-Fuller test shows that most of the features used are stationary. We made no effort to induce stationarity for those features where the property is absent because neural networks can handle chaotic sequences. Using the window structure ensures trend and seasonality components of a time-series are irrelevant because of the rolling of training and testing samples. We define the model using *TensorFlow* and *Keras* comprising *"Sequential"*, two hidden layers of type *Dense* and an output layer whose shape corresponds to the output size. The general structure of the model is as follows.

Figure 12 FNN Design

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_47 (Dense)             (None, 20)                2220
_________________________________________________________________
dense_48 (Dense)             (None, 20)                420
_________________________________________________________________
dense_49 (Dense)             (None, 5)                 105
=================================================================
Total params: 2,745
Trainable params: 2,745
Non-trainable params: 0
```

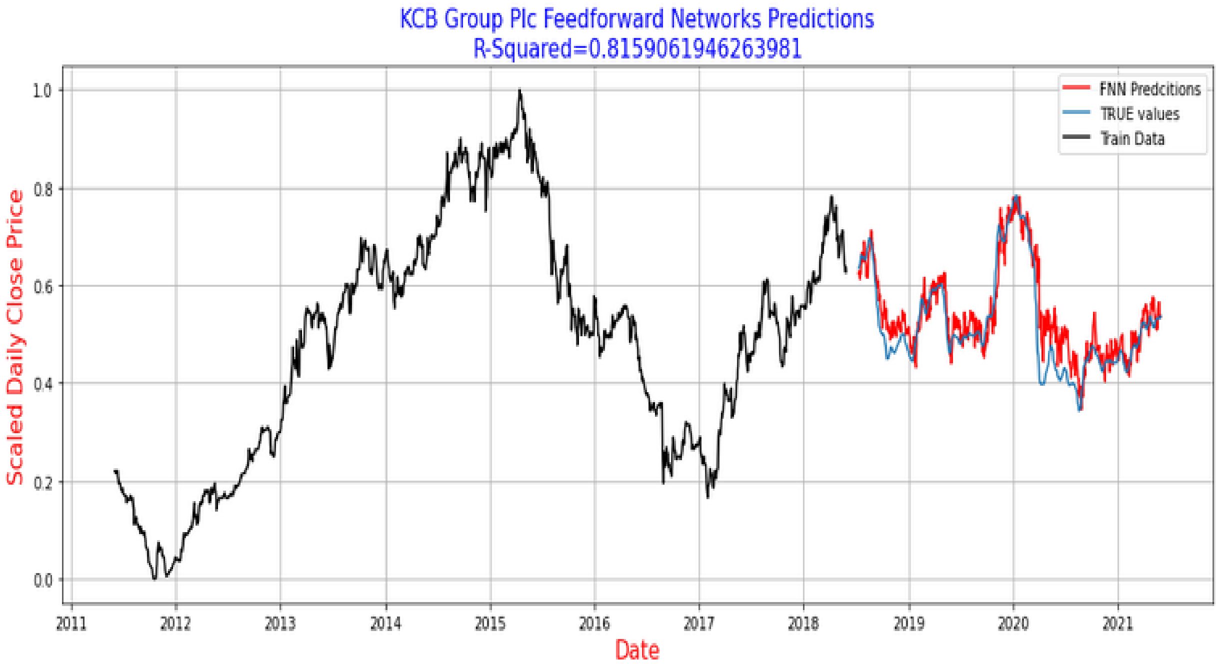
This model acted as a Neural Network benchmark for the two upcoming Recurrent networks. As with the other neural networks in this study, we explored the effect of changing the width of the networks (which is the number of neurons per layer). We experimented with the following number of neurons per layer; (10,20,40). We recorded the Root Mean Squared Error and R-Squared metrics. Consider the following table.

Table 3: FNN performance per Neuron Configuration

Company	R2 10 Neurons	R2 20 Neurons	R2 40 Neurons	Mean per Company
Absa Bank Kenya PLC	0.7702609	0.749888159	0.638655727	0.719601596
Bamburi Cement Ltd	0.746018393	0.840509012	0.87858471	0.821704038
Britam Holdings Ltd	0.59075219	0.679645178	0.217483209	0.495960192
British American Tobacco (Kenya) Ltd	0.852063507	0.865365847	0.933997203	0.883808852
Centum Investment Co. PLC	0.689229758	0.829780933	-0.436996717	0.360671325
Co-operative Bank of Kenya Ltd	0.687983413	0.731236844	0.832550477	0.750590245
Diamond Trust Bank Kenya Ltd	0.938554598	0.827999181	0.903239192	0.88993099
East African Breweries Ltd	-0.408712261	0.630903073	0.805016018	0.342402277
Equity Group Holdings Ltd	0.837404788	0.850752915	0.819475442	0.835877715
KCB Group Plc	0.748124172	0.817186069	0.815906195	0.793738812
Kenya Electricity Generating Co. Ltd	-1.021329336	-0.824915516	-1.556413393	-1.134219415
Kenya Power Lighting Co. Ltd	-3.723792595	-3.274101321	0.757678149	-2.080071922
Kenya Reinsurance Corp. Ltd	0.618281112	0.434933153	0.882997915	0.64540406
Nairobi Securities Exchange Ltd	0.364464677	0.673365788	0.873646909	0.637159125
Nation Media Group	0.629966592	0.760490172	0.659087722	0.683181495
NCBA Group PLC	0.498370219	0.688177102	0.865686399	0.684077907
Safaricom PLC	0.056150495	0.849632899	0.841627434	0.582470276
Stanbic Holdings PLC	-0.00868159	0.345533144	-4.822303478	-1.495150641
Standard Chartered Bank Kenya Ltd	0.273924499	0.700493681	0.598044678	0.524154286
WPP Scangroup Ltd	0.692590389	0.808952503	0.821769005	0.774437299
FNN Average R2 score	**0.241581196**	**0.449291441**	**0.31648664**	**0.335786426**

The width of 20 neurons per layer achieved the highest R-Squared of 0.449, suggesting that the middle range of the number of neurons per layer is ideal. The R-squared is low but positive, indicating some predictive capability. A one-tailed paired T-test comparison with ARIMA's R-Squared produces a p-value of 2.10868E-05, well below the statistical threshold of 0.05 level of significance, to reject the null hypothesis that the two models have the same forecasting capability. One of the best predictive performances by the FNN architecture was achieved by the British American Tobacco (Kenya) Ltd.

Figure 13: An Excellent FNN prediction



The figure above shows a sample plot of the training data 'Close Price', test data 'Close Price' and FNN prediction values of the KCB Group Plc. This prediction has an R-squared of 0.816.

4.4 Proposed Recurrent Neural Networks Results

We then implemented Recurrent Neural Networks namely the GRU and the LSTM using TensorFlow's "*Keras*", For consistency and comparability, the depth (number of layers) was maintained at two. We varied the number of neurons per layer as the Feed-forward model. The following is the shape of the LSTM and GRU model used.

Figure 14: GRU/LSTM Design

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional_2 (Bidirectio  (None, 40)               4800
 nal)

 dense_6 (Dense)             (None, 10)                410

=================================================================
Total params: 5,210
Trainable params: 5,210
Non-trainable params: 0
```

These Recurrent models have specific layers for handling temporal sequences. We trained, fitted, saved, and used the models to predict the forecast period.

4.4.1 Long-Short Term Memory Predictive Performance

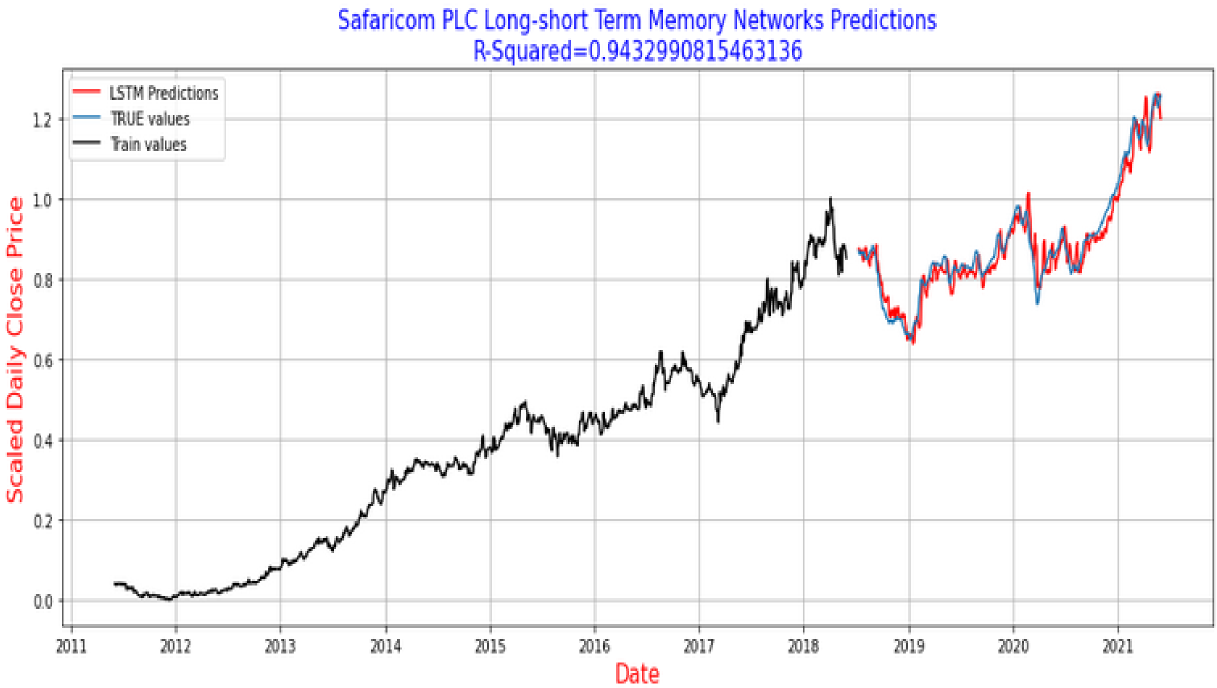
The results in the following table represent the R-squared for each neuron configuration when we trained the models on the stocks that are part of the NSE-20 Index. The best width, as indicated by the number highest mean of 0.658962 was 20 neurons.

Table 4: Prediction results for LSTM per Neuron Configuration

Company	R2 10 Neurons	R2 20 Neurons	R2 40 Neurons	Mean per Company
Absa Bank Kenya PLC	0.853088971	0.860711721	0.868791806	0.860864166
Bamburi Cement Ltd	-0.581524323	0.628792612	0.002968622	0.016745637
Britam Holdings Ltd	0.866377586	0.895012281	0.093151608	0.618180492
British American Tobacco (Kenya) Ltd	0.88831606	0.831701179	0.917418601	0.87914528
Centum Investment Co. PLC	0.953775524	0.965212211	0.94812691	0.955704881
Co-operative Bank of Kenya Ltd	0.927251619	0.924716605	0.894081462	0.915349895
Diamond Trust Bank Kenya Ltd	0.969741586	0.96418032	0.982306363	0.97207609
East African Breweries Ltd	0.840796032	0.868054381	0.868754168	0.859201527
Equity Group Holdings Ltd	0.899714012	0.858982371	0.884357634	0.881018006
KCB Group Plc	0.924810584	0.786980916	0.859745256	0.857178919
Kenya Electricity Generating Co. Ltd	0.49913104	0.774796733	0.740626258	0.67151801
Kenya Power Lighting Co. Ltd	0.115885062	0.834841647	-0.024591983	0.308711575
Kenya Reinsurance Corp. Ltd	0.905598572	0.902842705	0.918556457	0.908999245
Nairobi Securities Exchange Ltd	-0.148733786	0.020564215	-0.839105627	-0.322425066
Nation Media Group	0.055254944	-1.237449142	0.951568336	-0.076875288
NCBA Group PLC	0.806760968	0.910049378	0.831854291	0.849554879
Safaricom PLC	0.923061956	0.925470845	0.943299082	0.930610628
Stanbic Holdings PLC	0.878897633	0.835399765	0.80729675	0.840531383
Standard Chartered Bank Kenya Ltd	0.929889839	0.927790976	0.919389167	0.925689994
WPP Scangroup Ltd	0.100311367	-0.299401919	0.266612553	0.022507334
LSTM Average R2 score	**0.630420262**	**0.65896249**	**0.641760386**	**0.643714379**

The LSTM achieved its best fit on Diamond Trust Bank Ltd using a width of 40 neurons per layer with an R-Squared of 0.9823.

Figure 17: An Excellent Prediction by an LSTM model



The preceding figure indicates a plot of one of the best LSTM predictions against the true "Close Price" of Safaricom Ltd. The LSTM model used had 20 Neurons and a batch size of 40 and achieved an R-Squared of 0.9432.

4.4.2 Gated Recurrent Unit Predictive Performance

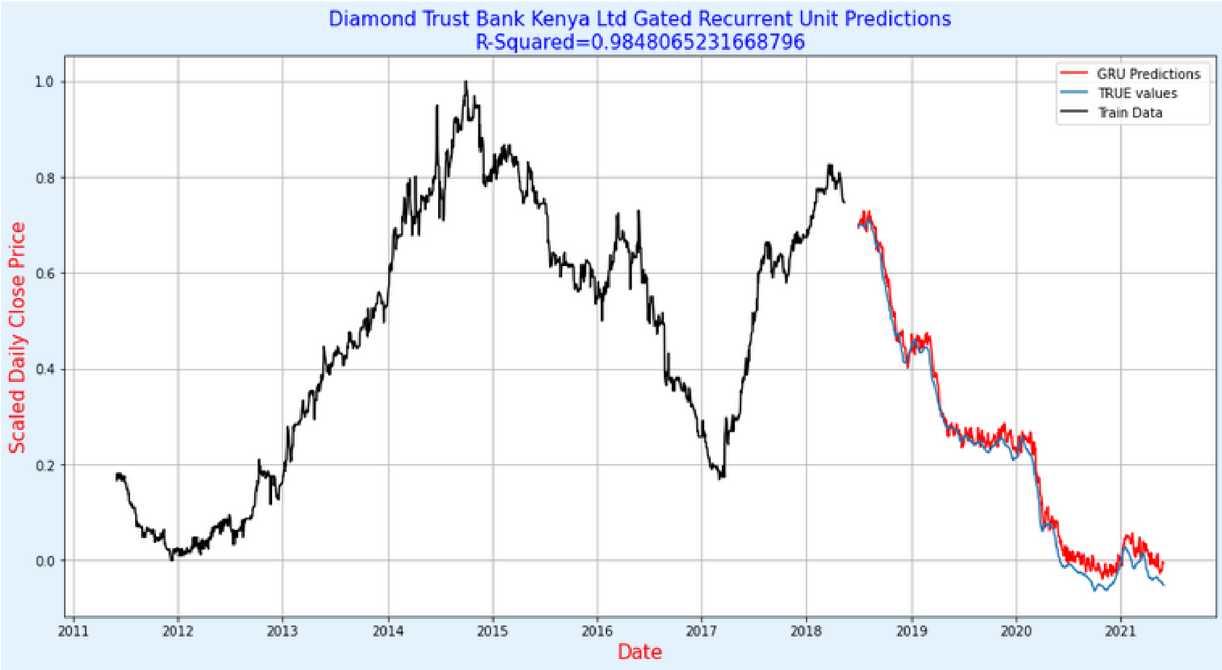
The results in the next table represent the R-squared for each neuron configuration when we trained the GRU models on the stocks that are part of the NSE-20 Index. The best width, as indicated by the highest mean of R-Squared measure of 0.693425, was 40 neurons.

Table 5: Prediction results for GRU per Neuron Configuration

Company	R2 10 Neurons	R2 20 Neurons	R2 40 Neurons	Mean per Company
Absa Bank Kenya PLC	0.848457376	0.832406728	0.852905601	0.844589902
Bamburi Cement Ltd	0.298090705	-0.134967666	0.155655203	0.106259414
Britam Holdings Ltd	0.897710594	0.859835107	0.82529926	0.86094832
British American Tobacco (Kenya) Ltd	0.927161214	0.942391588	0.956353416	0.941968739
Centum Investment Co. PLC	0.94395153	0.959718319	0.954221878	0.952630576
Co-operative Bank of Kenya Ltd	0.742860868	0.871357831	0.85131564	0.82184478
Diamond Trust Bank Kenya Ltd	0.96492826	0.989638686	0.984806523	0.979791156
East African Breweries Ltd	0.773948526	0.846450511	0.875701384	0.832033473
Equity Group Holdings Ltd	0.865639675	0.865624426	0.874267084	0.868510395
KCB Group Plc	0.878118508	0.878700397	0.916417707	0.891078871
Kenya Electricity Generating Co. Ltd	0.721234648	0.715342849	0.71674894	0.717775479
Kenya Power Lighting Co. Ltd	0.027504639	-0.13536393	-0.576369895	-0.228076396
Kenya Reinsurance Corp. Ltd	0.866515621	0.792611	0.744973524	0.801366715
Nairobi Securities Exchange Ltd	0.00855318	0.244615441	-0.142731832	0.036812263
Nation Media Group	0.038556454	-0.986735439	0.514100078	-0.144692969
NCBA Group PLC	0.864445467	0.864836534	0.876591712	0.868624571
Safaricom PLC	0.915382744	0.911939703	0.934919052	0.920747166
Stanbic Holdings PLC	0.808063256	0.900547535	0.871390814	0.860000535
Standard Chartered Bank Kenya Ltd	0.937947958	0.925503481	0.938356992	0.933936143
WPP Scangroup Ltd	0.323649639	0.012369564	0.743583601	0.359867602
GRU Average R2 score	**0.682636043**	**0.607841133**	**0.693425334**	**0.661300837**

The GRU achieved its best fit on Diamond Trust Bank with a width of 20 neurons and R-squared measure of 0.959.

Figure 18: Best GRU and Overall Performance



The preceding figure shows a plot of the best GRU predictions against the true "Close Price" of Diamond Trust Bank (K) Ltd. The GRU model used had 40 Neurons and a batch size of 40.

The above prediction represents the best forecasting by a model in the {20:10} window. The R-squared value of .985 shows a near perfect fit on the Diamond Trust Bank Ltd.'s 'close' prices.

Other trivial observations are:

i. Banking and technology companies on the NSE such as Safaricom Ltd and Diamond Trust Bank (K) and are easier to predict than companies such as Bamburi Ltd.
ii. The most predictable company is the Diamond Trust Bank (K) Ltd.
iii. The batch size is the number of samples passed during each training epoch.

After experimenting with two sizes, we found batch size is one of the most critical parameters. Statistical inference was used to make conclusions. We used 20 stock data fitted with several parameters. We tabled the results of each run, then subjected to paired T-tests for comparison of models, batch sizes, and sliding windows.

4.5 Comparing the Models

For the Neural Network models, the GRU consistently outperforms other models as indicated by R-Squared on all neuron configurations. Consider the following table.

Figure 19: Comparison of Models by their R-Squared

Company	FNN	GRU	LSTM
Absa Bank Kenya PLC	0.719601596	0.844589902	0.860864166
Bamburi Cement Ltd	0.821704038	0.106259414	0.016745637
Britam Holdings Ltd	0.495960192	0.86094832	0.618180492
British American Tobacco (Kenya) Ltd	0.883808852	0.941968739	0.87914528
Centum Investment Co. PLC	0.360671325	0.952630576	0.955704881
Co-operative Bank of Kenya Ltd	0.750590245	0.82184478	0.915349895
Diamond Trust Bank Kenya Ltd	0.88993099	0.979791156	0.97207609
East African Breweries Ltd	0.342402277	0.832033473	0.859201527
Equity Group Holdings Ltd	0.835877715	0.868510395	0.881018006
KCB Group Plc	0.793738812	0.891078871	0.857178919
Kenya Electricity Generating Co. Ltd	-1.134219415	0.717775479	0.67151801
Kenya Power Lighting Co. Ltd	-2.080071922	-0.228076396	0.308711575
Kenya Reinsurance Corp. Ltd	0.64540406	0.801366715	0.908999245
Nairobi Securities Exchange Ltd	0.637159125	0.036812263	-0.322425066
Nation Media Group	0.683181495	-0.144692969	-0.076875288
NCBA Group PLC	0.684077907	0.868624571	0.849554879
Safaricom PLC	0.582470276	0.920747166	0.930610628
Stanbic Holdings PLC	-1.495150641	0.860000535	0.840531383
Standard Chartered Bank Kenya Ltd	0.524154286	0.933936143	0.925689994
WPP Scangroup Ltd	0.774437299	0.359867602	0.022507334
	0.335786426	**0.661300837**	**0.643714379**

The one-tailed T-test for the difference between each pair of average R-Squared yields the following results.

Models Comparison Using the One-tailed Paired T-test Inferential statistics for the difference between two means					
ARIMA:FNN	ARIMA:GRU	ARIMA: LSTM	FNN:GRU	FNN:LSTM	LSTM:GRU
0.000590385	0.000240865	0.000232062	0.047984556	0.076818276	0.334848412

The above test compares pairs of the model using their R-Squared scores. The following conclusions are being drawn from the Tests of significance above;

I. We reject the null hypothesis with a 95% level of confidence that the Feed-forward network (FNN) model for a window structure of {20:10} offers no better predictions than the non-seasonal ARIMA.

II. We reject the null hypothesis with a 95% level of confidence that the LSTM model for the window structure of {20:10} offers no better predictions than the non-seasonal ARIMA.

III. We reject the null hypothesis with a 95% level of confidence that the GRU model for a window structure of {20:10} offers no better predictions than the non-seasonal ARIMA.

IV. We reject the null hypothesis with a 95% level of confidence that the GRU model for a window structure of {20:10} offers no better predictions than the FNN model of the same configuration.

V. We do not reject the null hypothesis with a 95% level of confidence that the GRU model for a window structure of {20:10} offers no better predictions than the LSTM model of the same configuration.

VI. We do not reject the null hypothesis with a 95% level of confidence that the LSTM model for a window structure of {20:10} offers no better predictions than the FNN model of the same configuration.

The difference in the predictive capability of GRU vis-à-vis LSTM models and LSTM vis-à-vis FNN of this configuration were not statistically significant. The GRU model, however, offers more stability in forecasting in most scenarios tested.

4.6 Comparing Window Sizes

Window size refers to the combination of past periods to consider and the future periods to forecast. This study compared two arbitrary window structures and tested for the statistical significance of the choice of size. The two structures are {20:10}, 20 past days to predict 10, and {30:15}, 30 past, 15 predictions.

Figure 20: Hypothesis Test Results for Window structure

Company	Mean R2 score {20:10}	Mean R2 score {30:15}
Absa Bank Kenya PLC	0.808351888	0.787955385
Bamburi Cement Ltd	0.31490303	-0.124030046
Britam Holdings Ltd	0.658363001	0.633961101
British American Tobacco (Kenya) Ltd	0.901640957	0.87928669
Centum Investment Co. PLC	0.756335594	0.892226025
Co-operative Bank of Kenya Ltd	0.82926164	0.781617183
Diamond Trust Bank Kenya Ltd	0.947266079	0.927608728
East African Breweries Ltd	0.677879092	0.605603828
Equity Group Holdings Ltd	0.861802039	0.798231947
KCB Group Plc	0.8473322	0.773776584
Kenya Electricity Generating Co. Ltd	0.085024691	0.016831737
Kenya Power Lighting Co. Ltd	-0.666478914	-1.069257611
Kenya Reinsurance Corp. Ltd	0.785256673	0.627564048
Nairobi Securities Exchange Ltd	0.117182107	-0.412321836
Nation Media Group	0.15387108	-0.883222493
NCBA Group PLC	0.800752452	0.640312372
Safaricom PLC	0.811276023	0.809291093
Stanbic Holdings PLC	0.068460425	0.425275635
Standard Chartered Bank Kenya Ltd	0.794593474	0.826063204
WPP Scangroup Ltd	0.385604078	0.460331224
Average	**0.546933881**	**0.472062019**
T-test for {20:10} vs {30:15} windows	**0.0330626**	

The one-tailed, paired T-test returns a p-value of 0.033 in favour of structure {20:10} indicating that we should reject the null hypothesis that the choice of window structure does not matter. The GRU consistently maintained its lead against all the other models, with a mean RMSE score of 0.077043037.

4.7 Comparing Batch Sizes

The choice of batch size is a hyperparameter, meaning that it is determined outside the model by the architect. In this study, we choose two arbitrary batch sizes, 40 and 80. Using a two-tailed T-test, we compare the average RMSE of the models with each size. The result of the investigation is as follows.

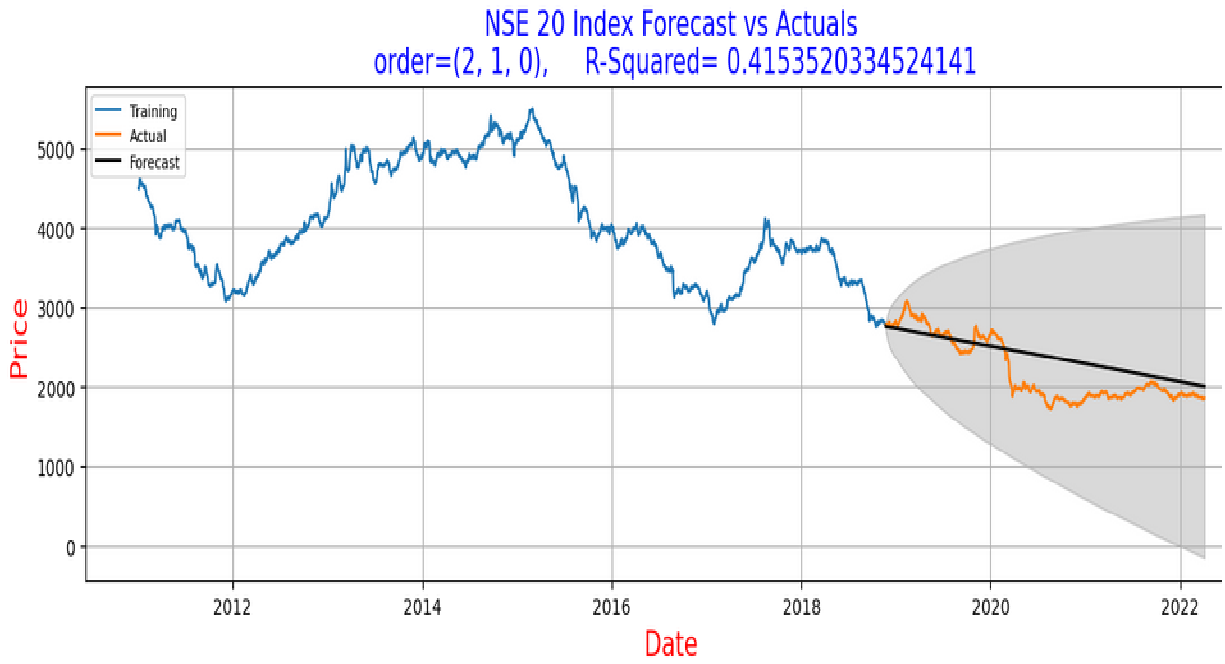
Company	Batch 40	Batch 80
Absa Bank Kenya PLC	0.808351888	0.799980664
Bamburi Cement Ltd	0.31490303	-1.077711137
Britam Holdings Ltd	0.658363001	0.639673318
British American Tobacco (Kenya) Ltd	0.901640957	0.012781309
Centum Investment Co. PLC	0.756335594	0.939285875
Co-operative Bank of Kenya Ltd	0.82926164	0.805066807
Diamond Trust Bank Kenya Ltd	0.947266079	0.946440752
East African Breweries Ltd	0.677879092	0.603321893
Equity Group Holdings Ltd	0.861802039	0.78894248
KCB Group Plc	0.8473322	0.750999804
Kenya Electricity Generating Co. Ltd	0.085024691	0.01255294
Kenya Power Lighting Co. Ltd	-0.666478914	-0.961702303
Kenya Reinsurance Corp. Ltd	0.785256673	0.786456581
Nairobi Securities Exchange Ltd	0.117182107	-0.132404796
Nation Media Group	0.15387108	-0.412434299
NCBA Group PLC	0.800752452	0.692877701
Safaricom PLC	0.811276023	0.823863976
Stanbic Holdings PLC	0.068460425	-1.270169289
Standard Chartered Bank Kenya Ltd	0.794593474	0.884052238
WPP Scangroup Ltd	0.385604078	0.329462882
	0.546933881	**0.29806687**

Task	Batch 40 vs 80
T-test for Batchsize	**0.011407622**

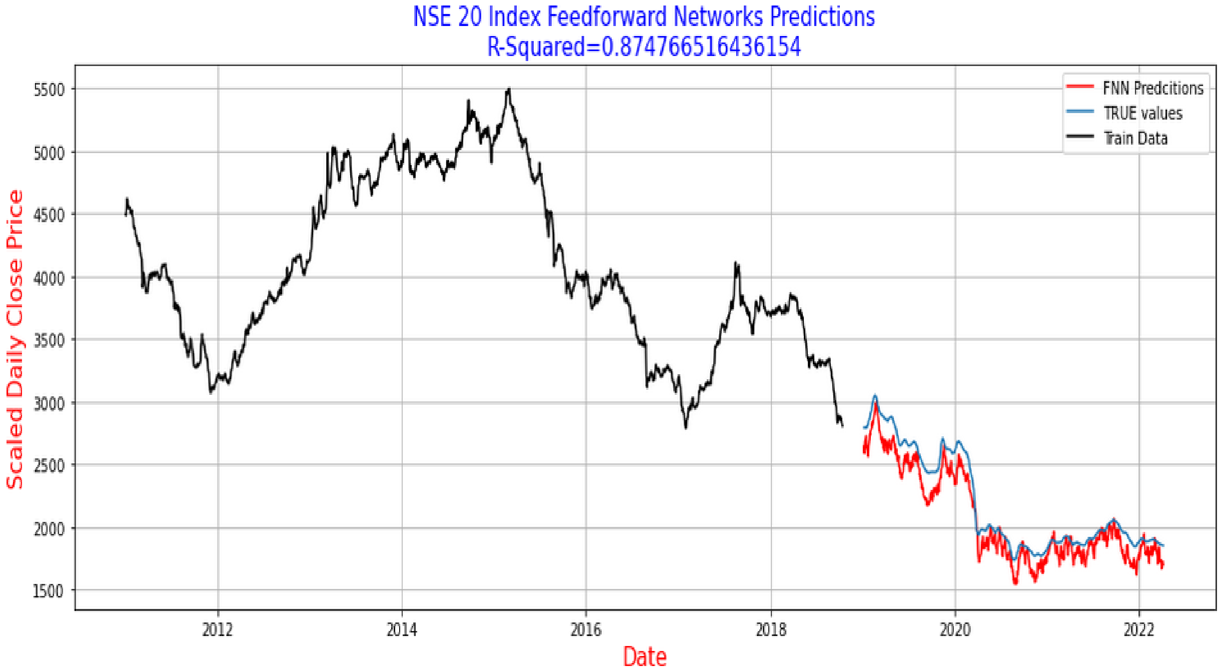
The resulting p-value favours the smaller batch size of 40 at a 95% confidence level. We reject the null hypothesis of the irrelevance of batch size. Doubling the batch size within the range worsened the performance of the models.

4.8 Performance on the Actual NSE-20 Index.

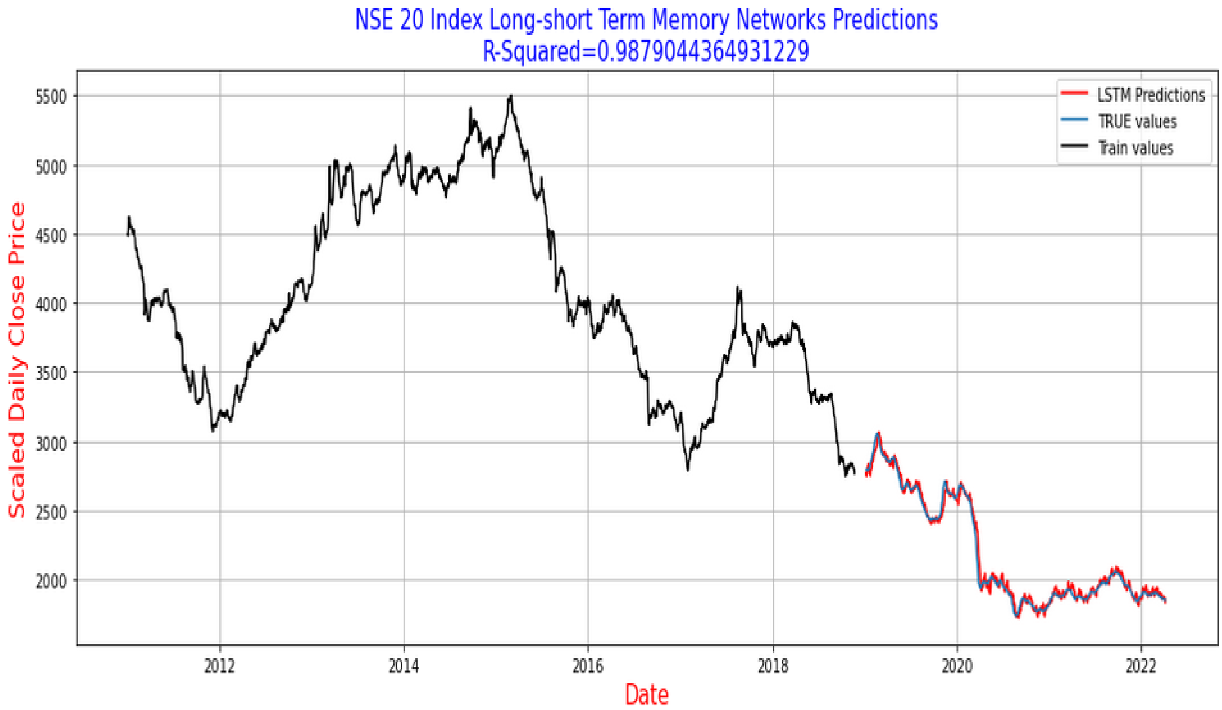
The models' performance on the actual NSE-20 index differs only slightly from the performance on the individual stocks that constitute the index. Both recurrent networks (LSTM, GRU) outperform the FNN and the ARIMA but the LSTM slightly edged out GRU on window {20:10} using 40 neurons and 40 batches. The result is as shown on the following plots.



The figure above shows the result of a forecast using a non-seasonal ARIMA model with p, d, q parameters of (2, 1, 0), on the NSE-20 index. The model correctly predicted the direction of the 30% out-of-sample test data (Close Price) when trained with 70% of in-sample data (Close Price). All the predicted values were within the 95% level of confidence. The goodness of fit of predicted values, as indicated by the R-squared of 0.415 indicates a positive fit but still lags the Neural Network models in forecasting the NSE-20 index.



The plot above shows the forecasting attempt of an FNN model on the NSE-20 index. 70% of the data was used for training the model while 30% validated the model. The number of batches used when training the model was 40 and the number of neurons per layer was 40 neurons. With an R-Squared of 0.875, the FNN is consistently the third-best model in forecasting the NSE-20 and the individual stocks that constitute the index.



The plot above shows the forecasting attempt of an LSTM model on the NSE-20 index. 70% of the data was used for training the model while 30% validated the model. The model had a batch size of 40 and a width of 40 neurons per layer. With an R-squared of 0.988, the LSTM was the best model in forecasting the NSE-20 and the second-best in forecasting the individual stocks that constitute the index.

The plot above shows the forecasting attempt of GRU model on the NSE-20 index. 70% of the data was used for training the model while 30% validated the model. The model had a batch size of 40 and a width of 40 neurons per layer. With an R-Squared of 0.94, the GRU model was the second-best model in forecasting the NSE-20 and the best in forecasting the individual stocks that constitute the index.

CHAPTER FIVE: SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Summary

This project aimed to determine whether Recurrent models offer better forecasting capability than the linear ARIMA and the Feed-forward Neural Networks. The study was prompted by the unavailability of literature testing the viability of the Recurrent Network models to forecast the Kenyan financial markets. As benchmark models, the ARIMA and Feed-forward models offered comparatives to determine the suitability of the espoused models. We tested two Recurrent models for feasibility to forecast the NSE stock prices. We sourced data from the NSE licensed data vendor, Synergy Solutions Ltd. The historical data comprised twenty companies' daily stock data for ten years of the NSE-20 index.

Five features were present in the original dataset, four of which were price-related, and the other was the trading volume. Using the Python programming language and its utilities for Statistics and Machine Learning, we fitted and evaluated univariate non-seasonal ARIMA, the Feed-forward Neural Networks, the LSTM and the GRU models for each company's stock. We recorded the RMSE and R-Squared scores. The primary metric used for statistical inference was the R-squared, which quantifies the goodness of fit between the actual values and model predictions. Feature engineering was a crucial focus of the pre-processing data stage. Nine more features were generated using the technical analysis library. We cleaned the new dataset for missing values, and the dataset was scaled and further processed to fit the problem description.

This study aimed to explore multiple period forecasting on a rolling window basis. This approach considers recent price action to determine the future movement of a time-series. We created the in-sample train data and the out-of-sample test dataset. The test and the train sets were further shaped into the sliding window structure necessary to answer our research questions. The models were fitted, saved, and retrieved to make predictions. The forecasts were compared with the out-of-sample test set, then evaluated using the RMSE and R-Squared. We applied statistical inference for the following purposes:

i. To compare the four models against each other.

ii. To determine the best window size.
iii. To ascertain the optimal batch size.

5.2 Conclusion

We made the following conclusions arrived at either using simple averages or statistical significance:

I. Without statistical inference, the best model deduced from the error metrics and a goodness-of-fit measure for every scenario was the Gated Recurrent Unit (GRU), followed by the Long- Short Term Memory (LSTM), the Feed-forward network, and last, the ARIMA.
II. With statistical inference through the paired, one-tailed T-test for the difference between two means, the GRU has a significant predictive power over the ARIMA and FNN and an insignificant predictive power over the LSTM when evaluated with a batch size of 40 on a {20:10} sliding window structure. Both Recurrent networks outperform the FNN and the ARIMA on this configuration.
III. With statistical inference, the choice of the sliding window structure matters. All models perform significantly worse when the sliding window is increased.
IV. With statistical inference, the batch size significantly matters. All neural network models perform better with a batch size of 40 than 80.
V. Without statistical inference, with a batch size of 40, a width of 20, and a sliding window structure of {20:10}, the GRU is the best model.
VI. The Best predictive performance was achieved by The GRU on Diamond Trust Bank using 40 neurons and 40 batches.

5.3 Congruity with Other Studies

The GRU outperformed all other models, but had its closest competitor in LSTM. These conclusions are in keeping with Song (2018) on the superiority of GRU for stock prediction, but contrast the findings of Sethia & Raut (2019), who found that the LSTM outperformed the GRU. Both Recurrent Networks outperformed the ARIMA and the Feed-forward networks in all the scenarios examined.

5.4 Utility for the Study

This research has provided state-of-the-art market research tools bespoke to the need of Kenyan investors and brokers. Previous Kenyan studies have primarily applied the incongruous Feed-forward Network architecture to model time-series data. The more appropriate, Recurrent networks have not been featured in Kenyan academic literature. Feature creation, which has been lacking in previous Kenyan studies, has been applied to the inputs of the models. The study has introduced Recurrent Networks to the Kenyan empirical literature and started further empirical consideration for cutting-edge sequence models in academia and industry.

The window structure methodology utilised is also valuable to short-term traders who only require rolling forecasts a few days at a time. The GRU should be considered before the LSTM and the FNN for window-style forecasting, since it has the lowest RMSE and the highest R-Squared. This study recommends a shorter forecasting period, even if the look-back period remains the same. For example, a GRU model using a window structure of {30:5} should be preferred to that using {30:15}, predicting 5 days by considering the past 30 days is more accurate than predicting 15 days with the same look-back.

5.5 Limitations of the Study

The proposed window-style forecasting works better for tighter look-back-to-forecast windows. Expanding the windows reduces the accuracy and introduces risk when such forecasts are used to inform trade executions. Therefore, this study is more useful to short-term stock traders than long-term traders. We argue that the long-term prediction of stocks using technical tools is a fool's errand. Fundamental analysis presents a better alternative for such long-term exploits.

Another limitation is that the study uses easily available daily stock data. This means that short-term oriented intraday traders will not find the utility for this study unless they reimplement the methodology using finer timeframes. Neural network models require more data during training, so with exposure to more granular data, say hourly, minutely or second, the utility for intraday trading will be assured. However, the reliance on highly granular forecasting assumes high volatility of the market and low execution costs. The Kenyan stock market is not highly volatile and therefore, intraday trading is not advisable.

This study only uses 20 stocks that are part of the NSE-20 index under the assumption that tracking the index gives a good representation of the whole Kenyan market. A valid concern should be: Can a sample constituted solely through a value-based index give a good representation of the whole Kenyan stock market?

Despite these limitations, the methodology and models adopted can guide an intelligent investor into profitability in the Kenyan securities market.

5.6 Recommendations for Further Study.

Time-series Modelling is a continuous and pervasive function of organisations worldwide. A perfect model is impossible because the future holds innumerable risks. The best we can hope for is to notice repeating patterns in the historical information that might prove relevant to the future outlook. This project has demonstrated the power of Recurrent Networks, especially the GRU over traditional models, but there are many more parameters to tune. Future studies should probe other architectures, such as the Convolutional Neural Networks (CNN), famous in the image processing domain, or ensemble models, such as Conv-LSTMs. We can also investigate different data input types: Stock chart images can be fed as training inputs to a visual learner instead of stock prices. Sequence forecasting will continue to be a pertinent task for data miners in the foreseeable future. Employing an assemblage of tools can only aid in bringing innovative solutions to complex business problems.

REFERENCES

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer. https://doi.org/10.1007/978-3-319-94463-0

Aldridge, I. (2010, July 8). *What is High-Frequency Trading, Afterall?* [Article]. HuffPost. https://www.huffpost.com/entry/what-is-high-frequency-tr_b_639203

Bajaj, A. (2021, May 2). *Performance Metrics in Machine Learning [Complete Guide]*. Neptune.Ai. https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide

Bell, F. (2018, September 6). *Forecasting at Uber: An Introduction*. Uber Engineering Blog. https://eng.uber.com/forecasting-introduction/

Benjamin, H. S. (1942). The Dow Theory of Stock Prices. *Social Research*, *9*(2), 204–224.

Bhatti, A. athar. (2019). Exploring the adoption of Artificial Intelligence in the Finance Industry: The case of Chatbots in the Kenyan Finance Industry. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3493340

Buffet, W. (1984). The Superinvestors of Graham-and-Doddsville. *Columbia Business School, 1984*. https://www8.gsb.columbia.edu/sites/valueinvesting/files/files/Buffett1984.pdf

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv:1406.1078 [Cs, Stat]*. http://arxiv.org/abs/1406.1078

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv:1412.3555 [Cs]*. http://arxiv.org/abs/1412.3555

Colby, R. W. (2002). *The Encyclopedia of Technical Market Indicators* (Second). McGraw-Hill. https://b-ok.africa/book/458975/3937c3?dsource=recommend

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.2307/2325486

Fang, J. (2014). Essays on technical analysis in stock markets: A thesis presented in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Finance at Massey University, Albany, New Zealand [Thesis, Massey University]. https://mro.massey.ac.nz/handle/10179/6783

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT press. http://www.deeplearningbook.org

Griffioen, G. A. W. (2003). *Technical analysis in financial markets*. Thela Thesis.

Gurney, K. (1997). *An introduction to neural networks* (Second print 1999). UCL Press Limited.

Hansson, M. (2017). *On stock return prediction with LSTM networks*. https://core.ac.uk/download/pdf/289960401.pdf

Hochreiter, S., & Schmidhuber, J. (1997). Long-Short-term-Memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Juma, J. (2016). *An Artificial Neural Network-Based Expert System for Loan Application Evaluation at Kenya Commercial Bank* [Moi University]. http://ir.mu.ac.ke:8080/xmlui/bitstream/handle/123456789/783/JANE%20THESIS%20MOST%20CURRENT%20FOR%20BINDING.pdf?sequence=1&isAllowed=y

Kamau, D. (2018). *Testing the Weak-form Efficiency of the Nairobi Securities Exchange Market.* [University of Nairobi]. http://erepository.uonbi.ac.ke/bitstream/handle/11295/152844/DANIEL%20MBURU%20KAMAU.pdf?sequence=1

Keskitalo, johan. (2020). *A Comparison of Recurrent Neural Networks Models and Econometric Models for Stock Market Predictions* [Umea University]. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjwuM7LlsfzAhXm_rsIHdL8DrsQFnoECBAQAQ&url=http%3A%2F%2Fwww.diva-portal.org%2Fsmash%2Fget%2Fdiva2%3A1466163%2FFULLTEXT01.pdf&usg=AOvVaw1TP6OGd_6dY2XXNqKK83S7

Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., Shiraz, M., & Gani, A. (2014). Big Data: Survey, Technologies, Opportunities, and Challenges. *The Scientific World Journal*, *2014*, e712826. https://doi.org/10.1155/2014/712826

Kim, D.-K. (2019). The Dogs of the Dow Theory – Is It Valid? *International Journal of Economics and Finance*, *11*, 43. https://doi.org/10.5539/ijef.v11n5p43

Kothari, C. R. (1985). Research Methodology Methods and Techniques by C.R. Kothari (z-lib.org).pdf (2nd ed.). New Age Publisher (p) ltd.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity*, *2020*, e6622927. https://doi.org/10.1155/2020/6622927

Maynard, M. (2020). Neural Networks Introduction to Artificial Neurons, Backpropagation and Multilayer Feed-forward Neural Networks with Real-World Applications (Advanced Data Analytics Book 2) (Book 2).

McGonagle, J., García, J. A., & Mollick, S. (2021). *Feed-forward Neural Networks | Brilliant Math & Science Wiki*. https://brilliant.org/wiki/Feed-forward-neural-networks/

Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. *Oreily Media*, *First Edition*, 392.

Mutua, N., & Mutothya, N. (2014). An Empirical Investigation of The Random Walk Hypothesis of Stock Prices on The Nairobi Stock Exchange. 1, 33–59.

Mwikamba, G. (2019). *An Artificial Neural Network Model For Forecasting Inflation In Kenya* [University of Nairobi]. http://erepository.uonbi.ac.ke/bitstream/handle/11295/107392/Mwikamba_An%20Artificial%20Neural%20Network%20Model%20For%20Forecasting%20Inflation%20In%20Kenya.pdf?sequence=1&isAllowed=y

Nairobi Securities Exchange. (2021). *Nairobi Securities Exchange (NSE)* [Securities Exchange]. https://www.nse.co.ke/nse/about-nse.html

Pankratz, A. (1983). *Forecasting with Univariate Box—Jenkins Models: Concepts and Cases* (1st ed.). https://b-ok.africa/book/927792/4e9f9c

Paolella, M. S. (2018). *Linear Models and Time-Series Analysis: Regression, ANOVA, ARMA and GARCH*. Wiley. https://b-ok.africa/book/3631438/4d11c9?dsource=recommend

Park, C.-H., & Irwin, S. H. (2004). *The Profitability of Technical Analysis: A Review* (SSRN Scholarly Paper ID 603481). Social Science Research Network. https://doi.org/10.2139/ssrn.603481

Paul, E. (2020, March 19). Alliance4ai wants to increase AI adoption levels in Africa to meet global standards. *Techpoint Africa*. https://techpoint.africa/2020/03/19/alliance4ai-ai-africa-global/

Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann.

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning* (Second Edition). Packt Publishing Ltd.

Raschka, S., & Vahid, M. (2015). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow (Second Edition). Packt Publishing Ltd.

Rashid, S., Shakeel, R., Bashir, H., Malik, K., & Wajib, K. (2016). Moore's Law Effect on Transistors Evolution. *International Journal of Computer Applications Technology and Research*, *5*(7), 5.

Rosenblatt, F. (1957). *Rosenblatt1958.pdf*. https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf

Sethia, A., & Raut, P. (2019). Application of LSTM, GRU and ICA for Stock Price Prediction: Proceedings of ICTIS 2018, Volume 2 (pp. 479–487). https://doi.org/10.1007/978-981-13-1747-7_46

Song, Y. (2018). *Stock Trend Prediction: Based on Machine Learning Methods*. https://escholarship.org/content/qt0cp1x8th/qt0cp1x8th.pdf?t=p63d23

Veneziani, V. (2011). The Greatest Trades of All Time: Top Traders Making Big Profits from the Crash of 1929 to Today.

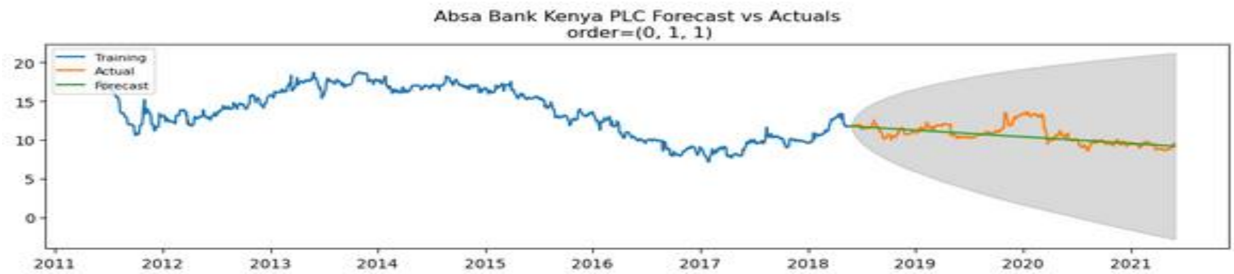
Wamalwa, T. (2019). An Artificial Neural Network Model for Predicting Retail Maize Prices in Kenya. 69.

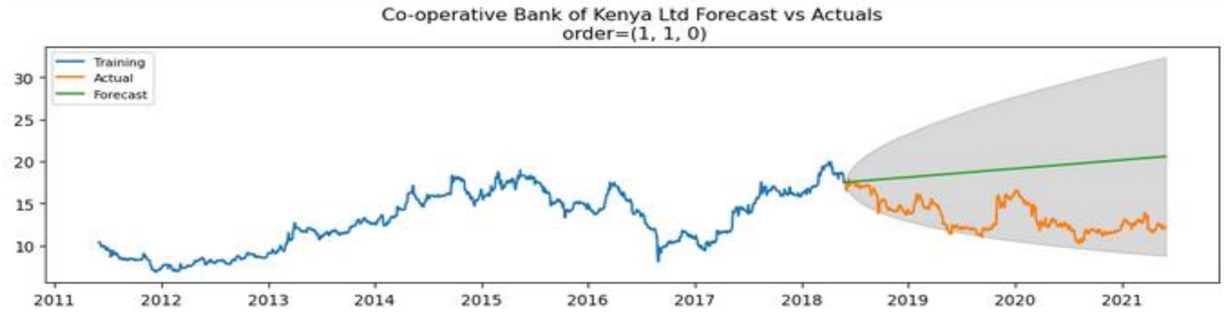
Wanjawa, B. (2014). *A-Neural-Network-Model-for-Predicting-Stock-Market-Prices-at-the-Nairobi-Securities-Exchange.pdf* [Research Repository]. https://www.researchgate.net/profile/Barack-Wanjawa/publication/269087026_A_Neural_Network_Model_for_Predicting_Stock_Market_Prices_at_the_Nairobi_Securities_Exchange/links/547f00360cf2de80e7cc70dc/A-Neural-Network-Model-for-Predicting-Stock-Market-Prices-at-the-Nairobi-Securities-Exchange.pdf

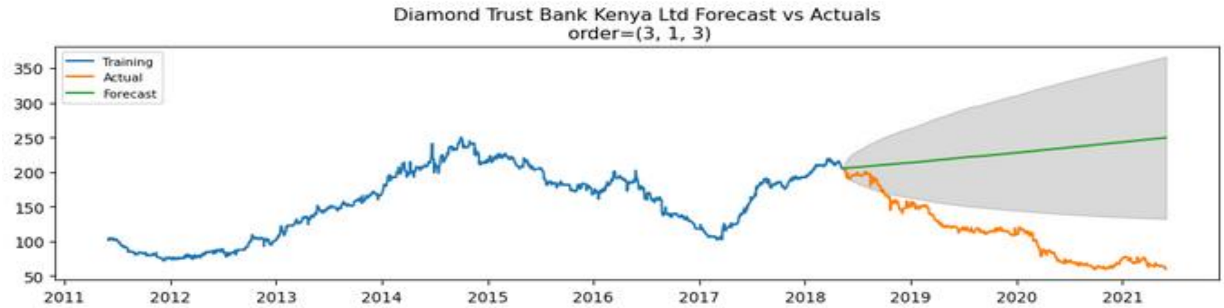
Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., … Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv:1609.08144 [Cs]*. http://arxiv.org/abs/1609.08144

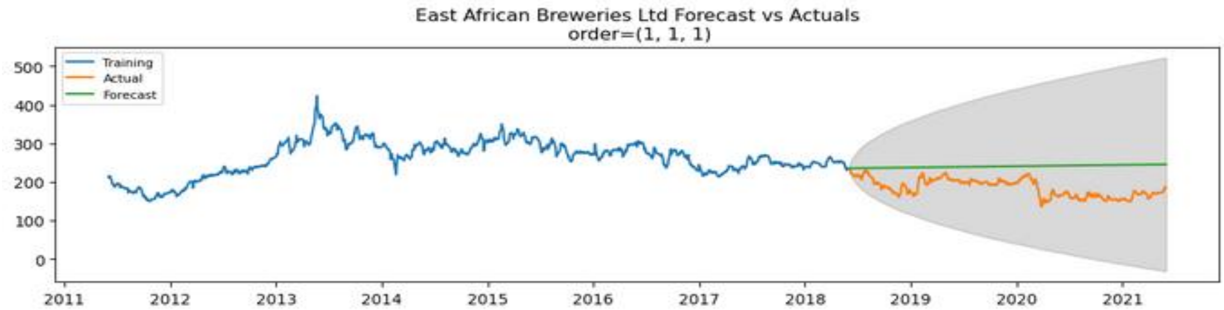
Zhang, M. (2021). Time-series: Autoregressive models AR, MA, ARMA, ARIMA. 77.

APPENDIX A: ARIMA PLOTS



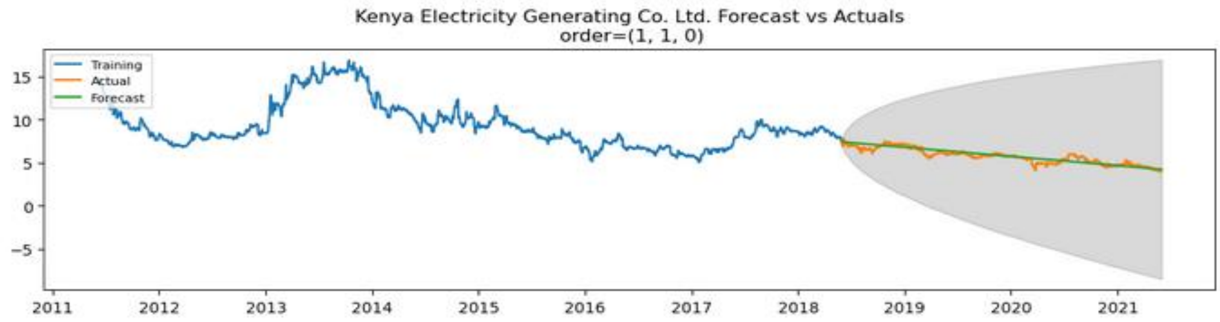

Co-operative Bank of Kenya Ltd Forecast vs Actuals
order=(1, 1, 0)
Training
Actual
Forecast
30
25
20
15
10
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


Diamond Trust Bank Kenya Ltd Forecast vs Actuals
order=(3, 1, 3)
Training
Actual
Forecast
350
300
250
200
150
100
50
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


East African Breweries Ltd Forecast vs Actuals
order=(1, 1, 1)
Training
Actual
Forecast
500
400
300
200
100
0
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

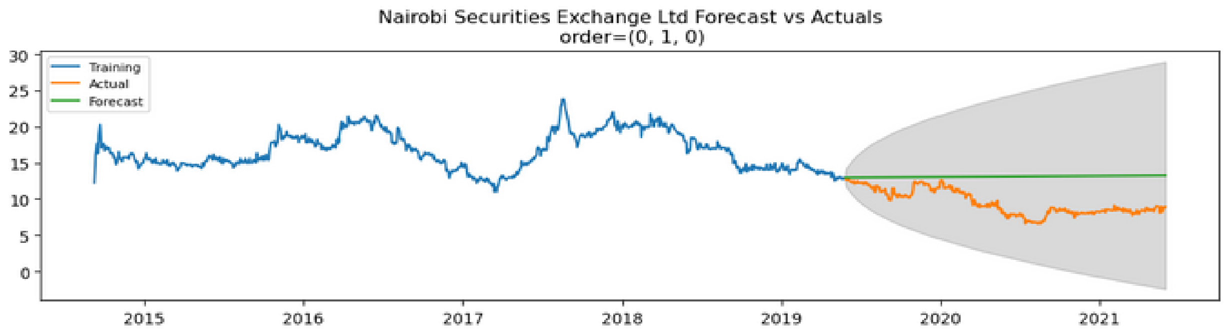

Equity Group Holdings Ltd Forecast vs Actuals
order=(0, 1, 1)
Training
Actual
Forecast
80
60
40
20
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

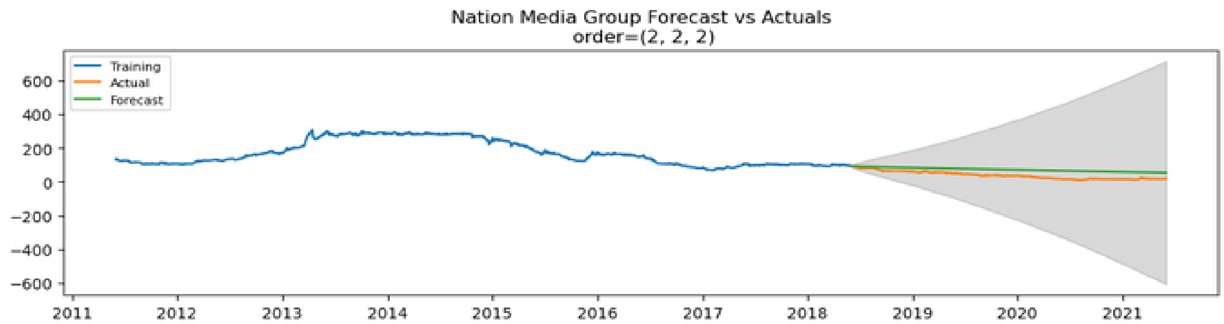

KCB Group Plc Forecast vs Actuals
order=(0, 1, 4)
Training
Actual
Forecast
80
60
40
20
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


Kenya Electricity Generating Co. Ltd. Forecast vs Actuals
order=(1, 1, 0)
Training
Actual
Forecast
15
10
5
0
−5
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

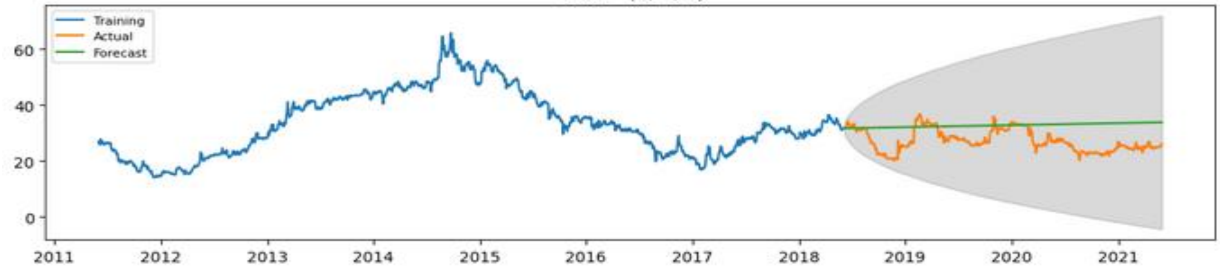

Kenya Power _ Lighting Co. Ltd Forecast vs Actuals
order=(1, 1, 0)
Training
Actual
Forecast
20
15
10
5
0
−5
−10
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021


Kenya Reinsurance Corp. Ltd. Forecast vs Actuals
order=(1, 1, 0)
Training
Actual
Forecast
8
6
4
2
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

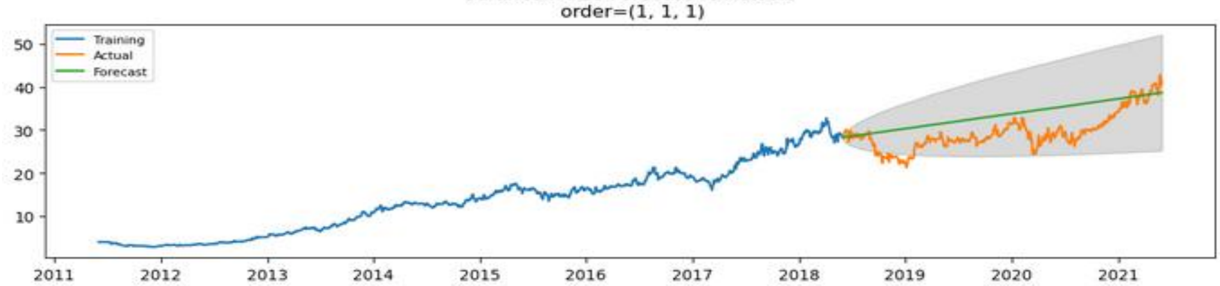

Nairobi Securities Exchange Ltd Forecast vs Actuals
order=(0, 1, 0)
Training
Actual
Forecast
30
25
20
15
10
5
0
2015
2016
2017
2018
2019
2020
2021


Nation Media Group Forecast vs Actuals
order=(2, 2, 2)
Training
Actual
Forecast
600
400
200
0
−200
−400
−600
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

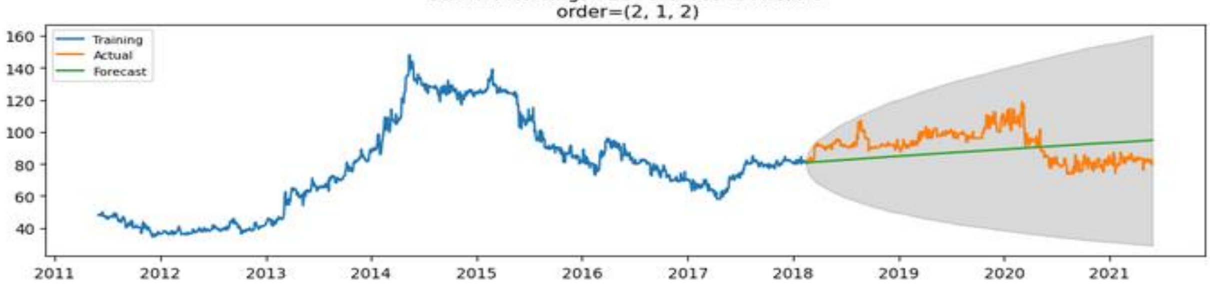
NCBA Group PLC Forecast vs Actuals
order=(0, 1, 1)



Safaricom PLC Forecast vs Actuals
order=(1, 1, 1)



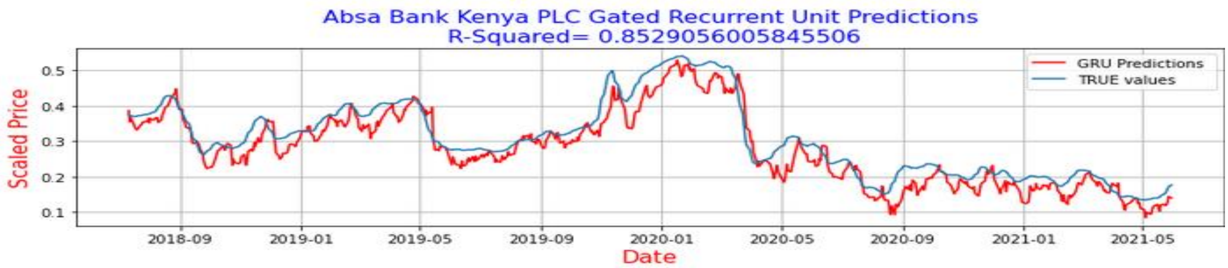
Stanbic Holdings PLC Forecast vs Actuals
order=(2, 1, 2)

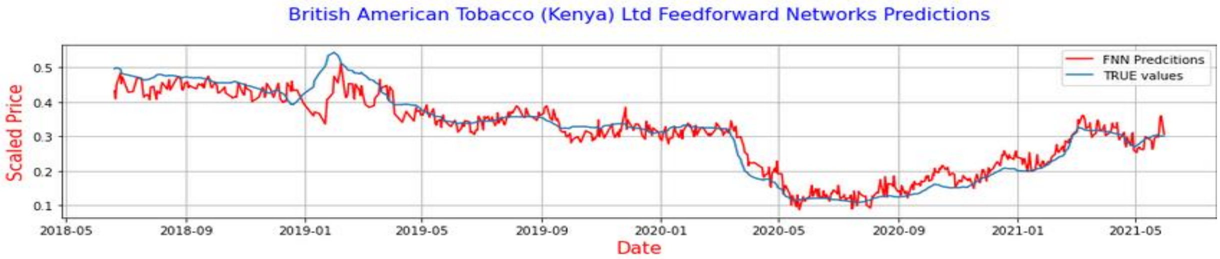


Standard Chartered Bank Kenya Ltd Forecast vs Actuals
order=(1, 1, 0)

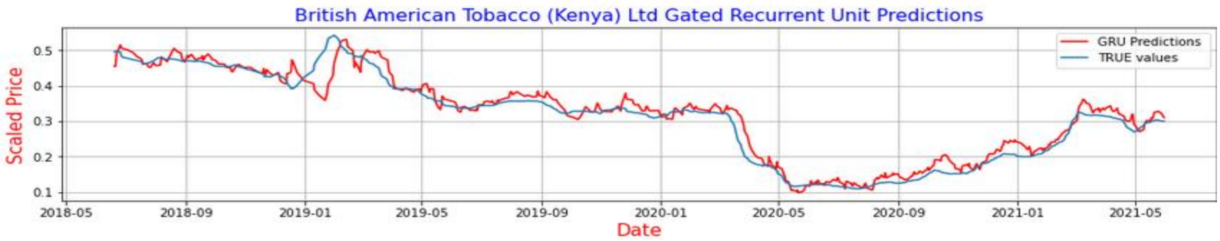
WPP Scangroup Ltd Forecast vs Actuals
order=(0, 1, 3)

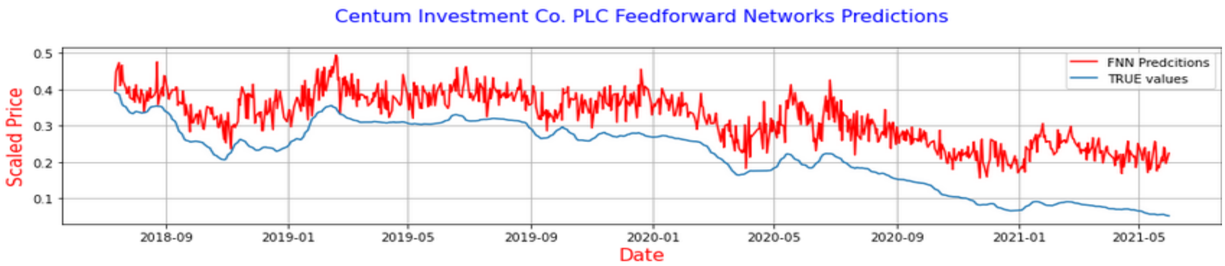
APPENDIX B: NEURAL NETWORK PLOTS

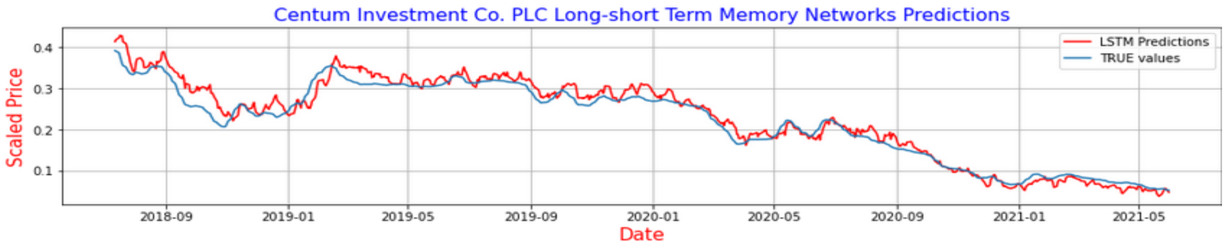



British American Tobacco (Kenya) Ltd Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4 0.5
2018-05 2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date


British American Tobacco (Kenya) Ltd Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4 0.5 0.6
2018-05 2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date

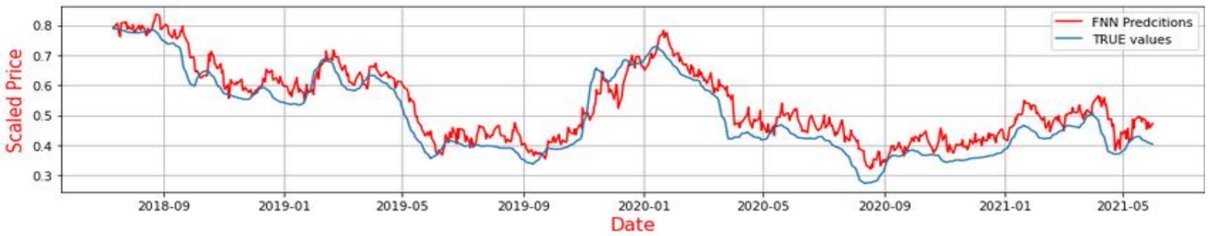

British American Tobacco (Kenya) Ltd Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4 0.5
2018-05 2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date


Centum Investment Co. PLC Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4 0.5
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date


Centum Investment Co. PLC Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date

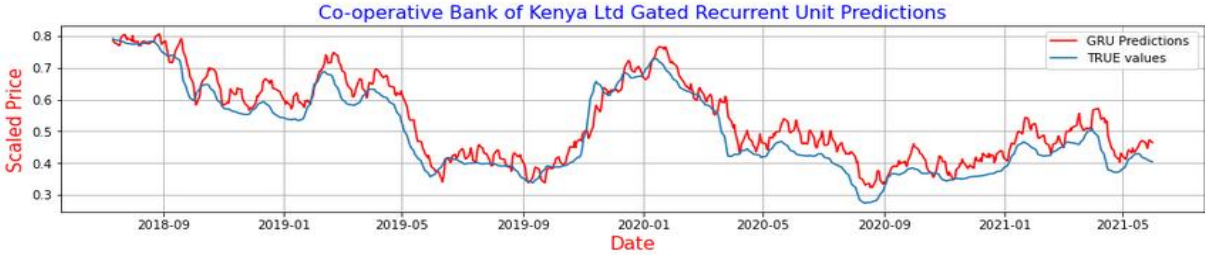

Centum Investment Co. PLC Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
0.1 0.2 0.3 0.4
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05
Date

Co-operative Bank of Kenya Ltd Feedforward Networks Predictions

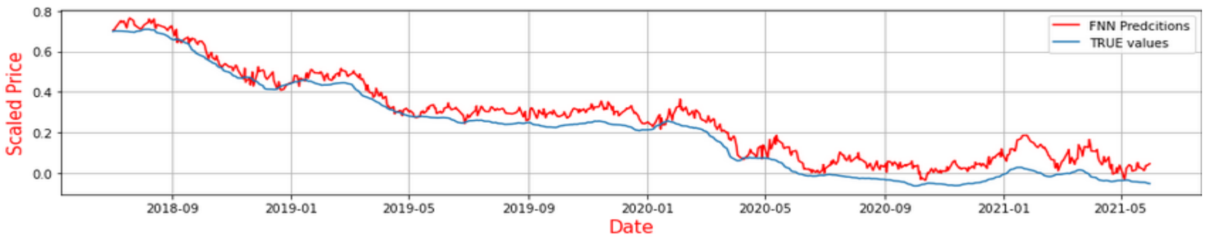


Co-operative Bank of Kenya Ltd Long-short Term Memory Networks Predictions

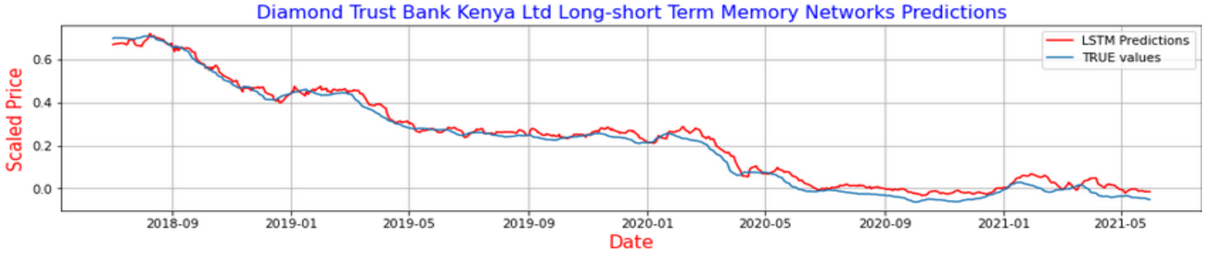
Co-operative Bank of Kenya Ltd Gated Recurrent Unit Predictions



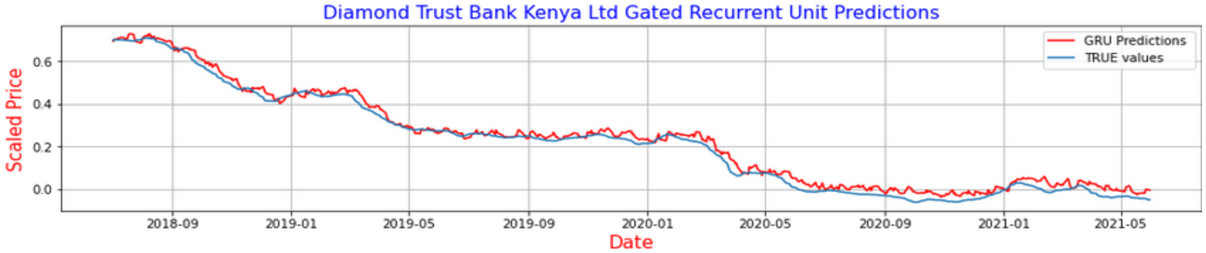
Diamond Trust Bank Kenya Ltd Feedforward Networks Predictions

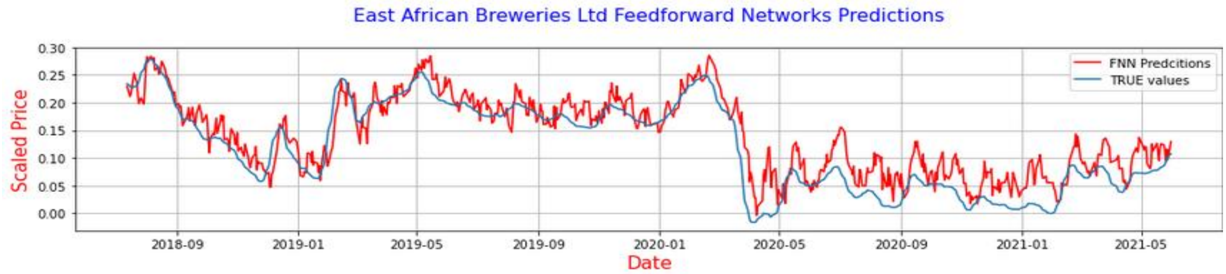


Diamond Trust Bank Kenya Ltd Long-short Term Memory Networks Predictions



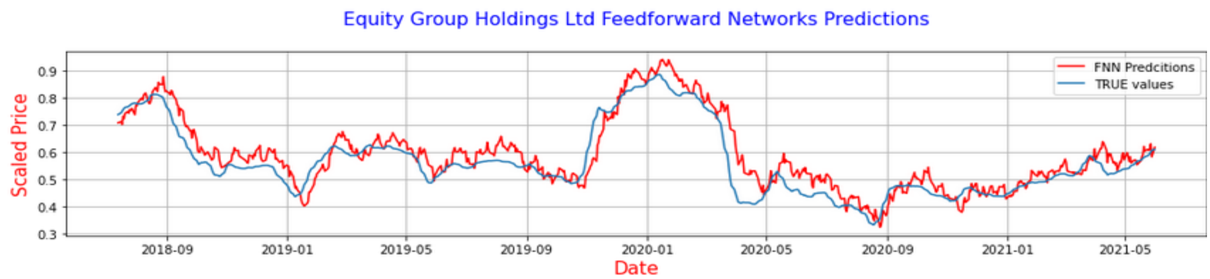
Diamond Trust Bank Kenya Ltd Gated Recurrent Unit Predictions

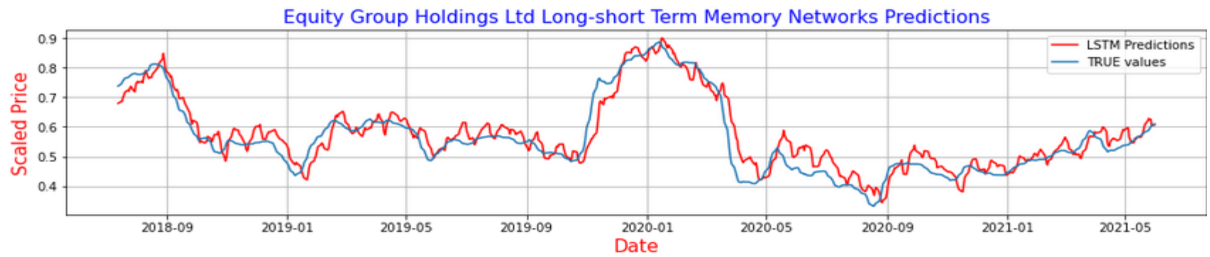


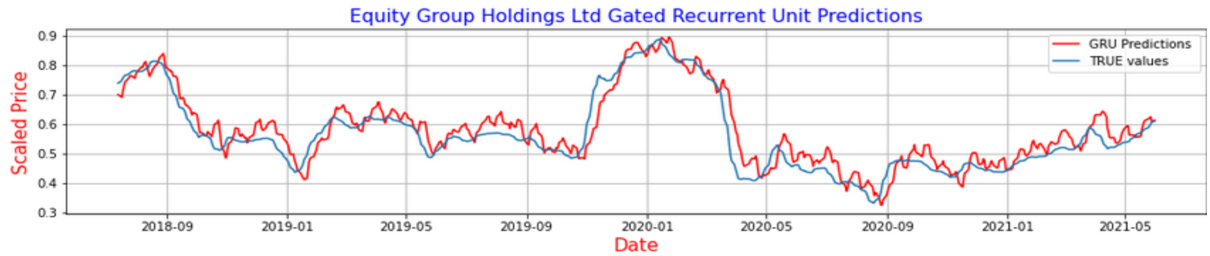

East African Breweries Ltd Feedforward Networks Predictions
Scaled Price
Date
FNN Predcitions
TRUE values
0.00 0.05 0.10 0.15 0.20 0.25 0.30
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05


East African Breweries Ltd Long-short Term Memory Networks Predictions
Scaled Price
Date
LSTM Predictions
TRUE values
0.00 0.05 0.10 0.15 0.20 0.25 0.30
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05

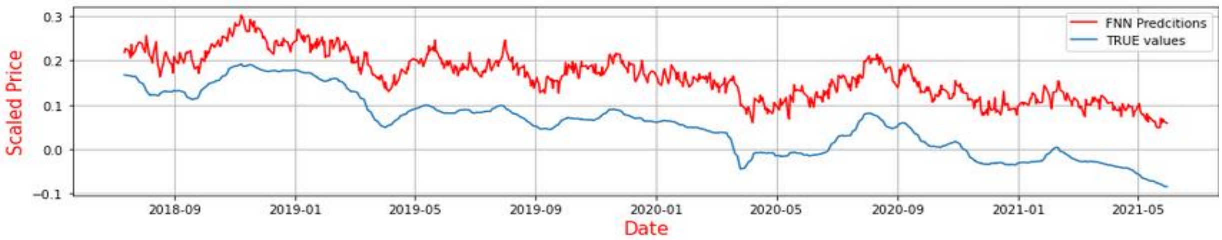

East African Breweries Ltd Gated Recurrent Unit Predictions
Scaled Price
Date
GRU Predictions
TRUE values
0.00 0.05 0.10 0.15 0.20 0.25 0.30
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05


Equity Group Holdings Ltd Feedforward Networks Predictions
Scaled Price
Date
FNN Predcitions
TRUE values
0.3 0.4 0.5 0.6 0.7 0.8 0.9
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05


Equity Group Holdings Ltd Long-short Term Memory Networks Predictions
Scaled Price
Date
LSTM Predictions
TRUE values
0.4 0.5 0.6 0.7 0.8 0.9
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05


Equity Group Holdings Ltd Gated Recurrent Unit Predictions
Scaled Price
Date
GRU Predictions
TRUE values
0.3 0.4 0.5 0.6 0.7 0.8 0.9
2018-09 2019-01 2019-05 2019-09 2020-01 2020-05 2020-09 2021-01 2021-05

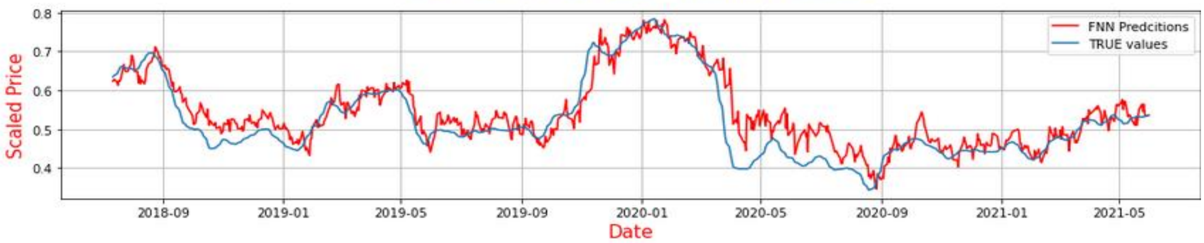
Kenya Electricity Generating Co. Ltd. Feedforward Networks Predictions



Kenya Electricity Generating Co. Ltd. Long-short Term Memory Networks Predictions

Kenya Electricity Generating Co. Ltd. Gated Recurrent Unit Predictions

KCB Group Plc Feedforward Networks Predictions



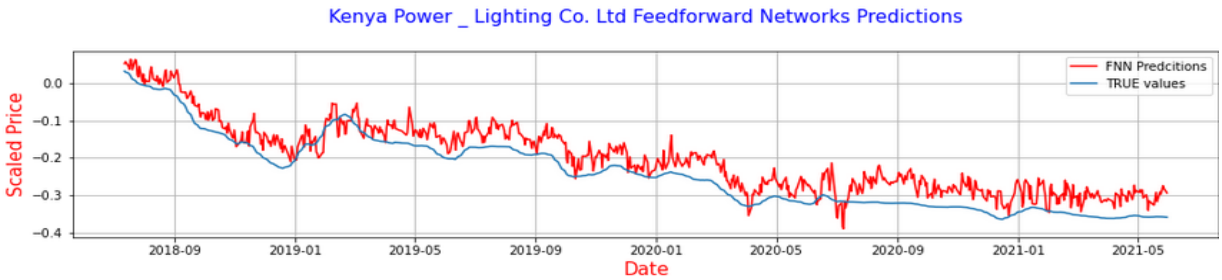
KCB Group Plc Long-short Term Memory Networks Predictions

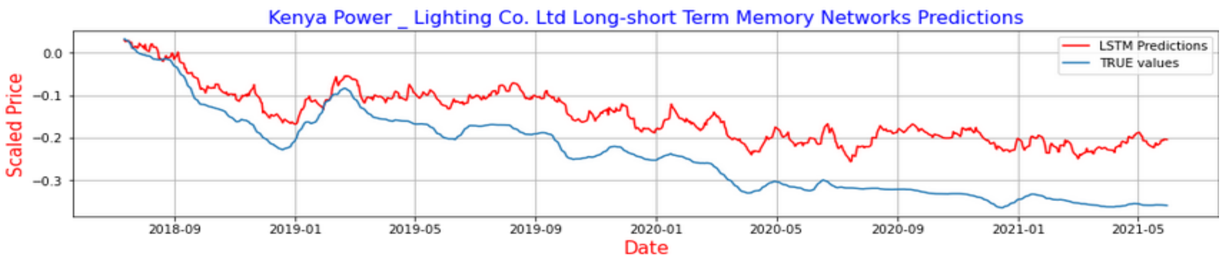
KCB Group Plc Gated Recurrent Unit Predictions

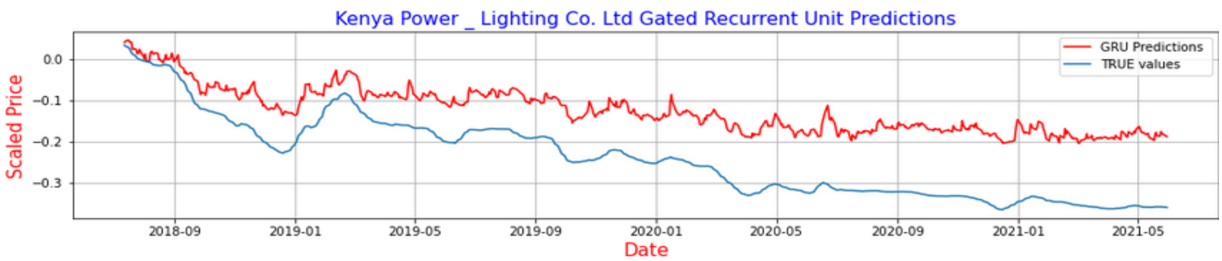

Kenya Electricity Generating Co. Ltd. Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
Date

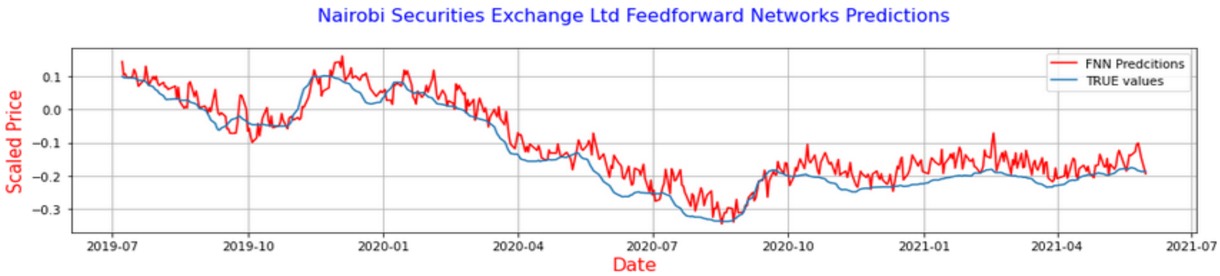

Kenya Electricity Generating Co. Ltd. Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
Date

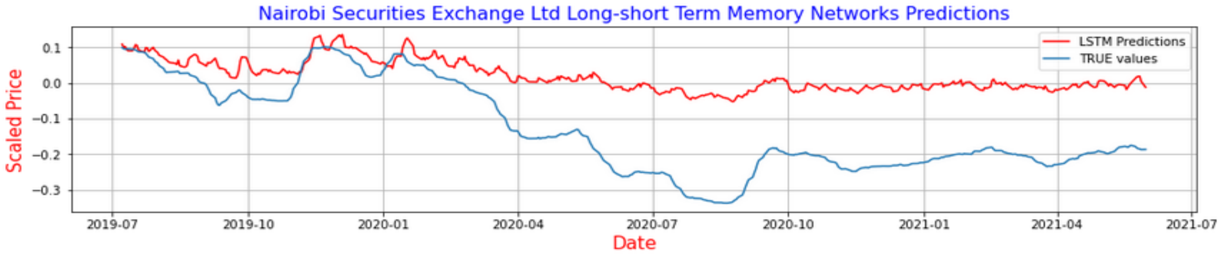

Kenya Electricity Generating Co. Ltd. Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
Date

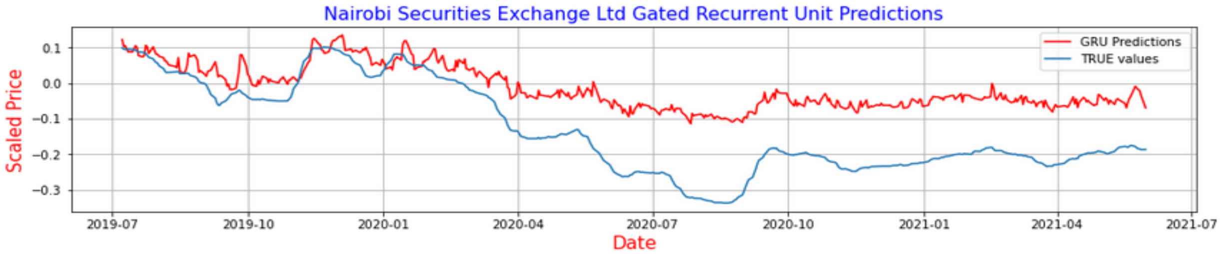

Kenya Power _ Lighting Co. Ltd Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
Date

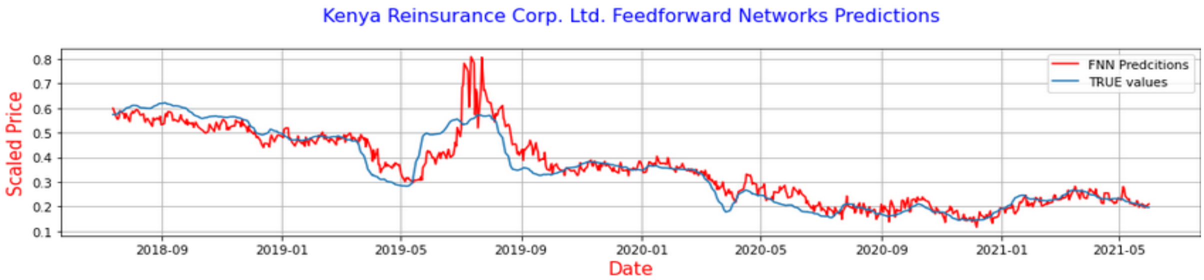

Kenya Power _ Lighting Co. Ltd Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
Date

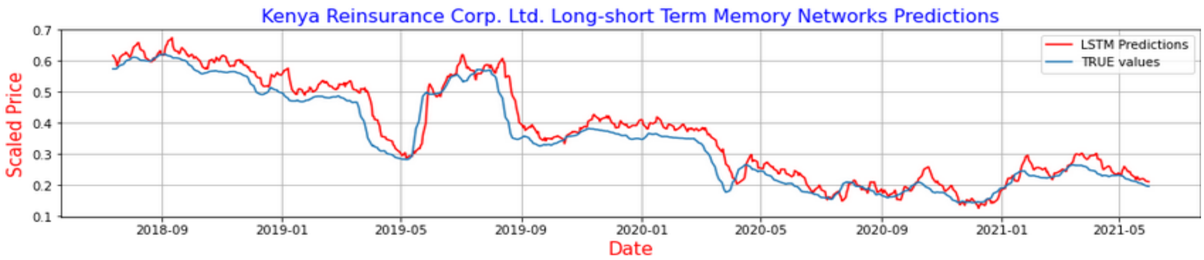

Kenya Power _ Lighting Co. Ltd Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
Date

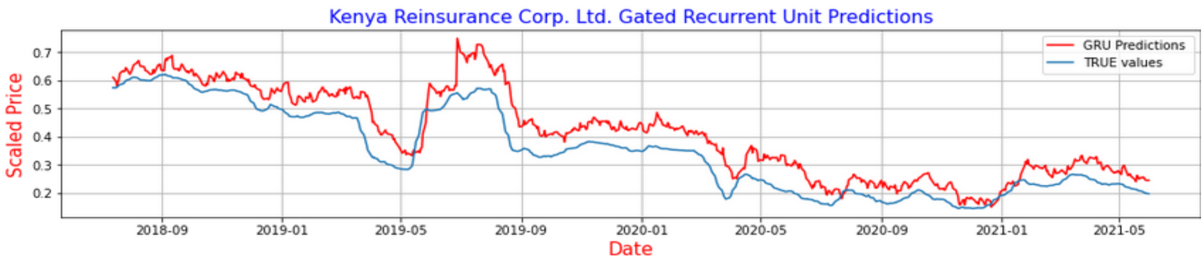

Nairobi Securities Exchange Ltd Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
0.1
0.0
−0.1
−0.2
−0.3
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
Date

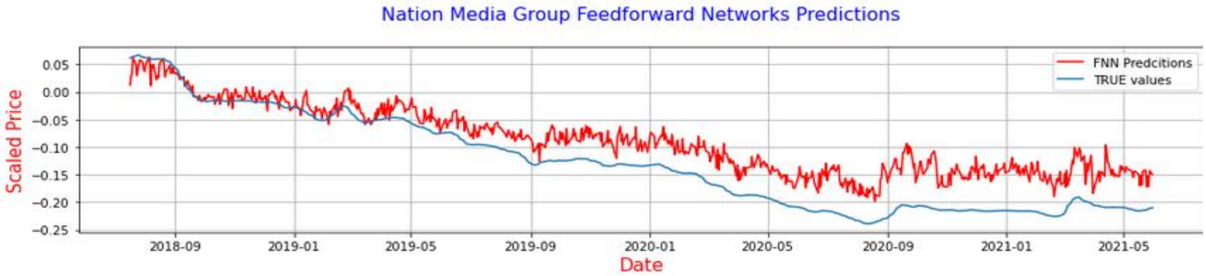

Nairobi Securities Exchange Ltd Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
0.1
0.0
−0.1
−0.2
−0.3
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
Date


Nairobi Securities Exchange Ltd Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
0.1
0.0
−0.1
−0.2
−0.3
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
Date


Kenya Reinsurance Corp. Ltd. Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2018-09
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
Date


Kenya Reinsurance Corp. Ltd. Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2018-09
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
Date


Kenya Reinsurance Corp. Ltd. Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
0.7
0.6
0.5
0.4
0.3
0.2
2018-09
2019-01
2019-05
2019-09
2020-01
2020-05
2020-09
2021-01
2021-05
Date


Nation Media Group Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
Date


Nation Media Group Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
Date

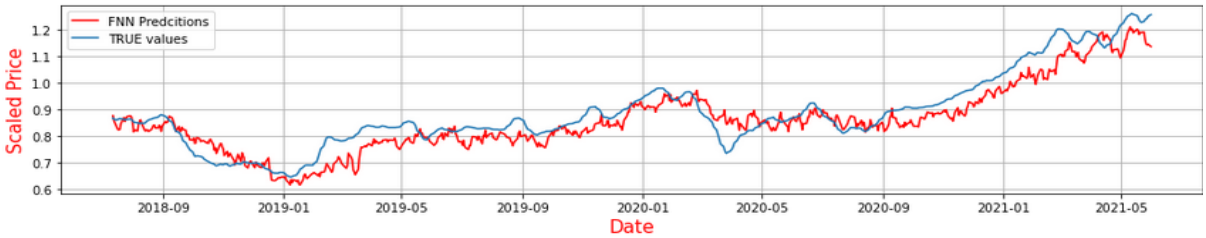

Nation Media Group Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
Date


NCBA Group PLC Feedforward Networks Predictions
FNN Predcitions
TRUE values
Scaled Price
Date

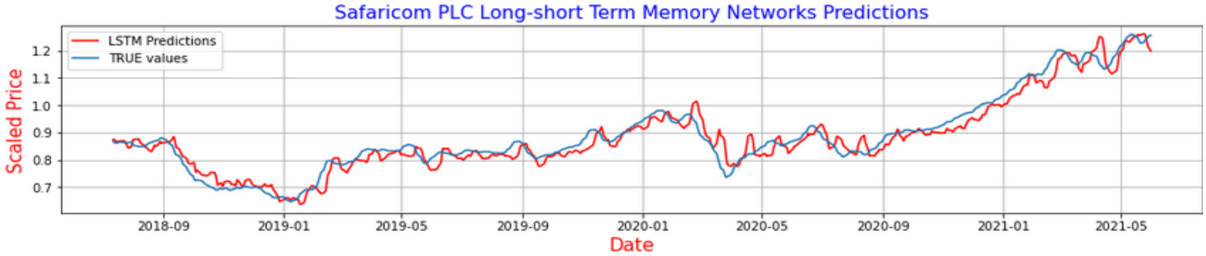

NCBA Group PLC Long-short Term Memory Networks Predictions
LSTM Predictions
TRUE values
Scaled Price
Date


NCBA Group PLC Gated Recurrent Unit Predictions
GRU Predictions
TRUE values
Scaled Price
Date

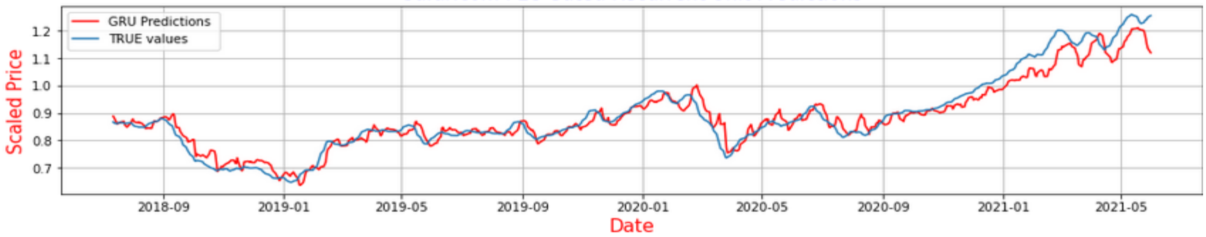
Safaricom PLC Feedforward Networks Predictions



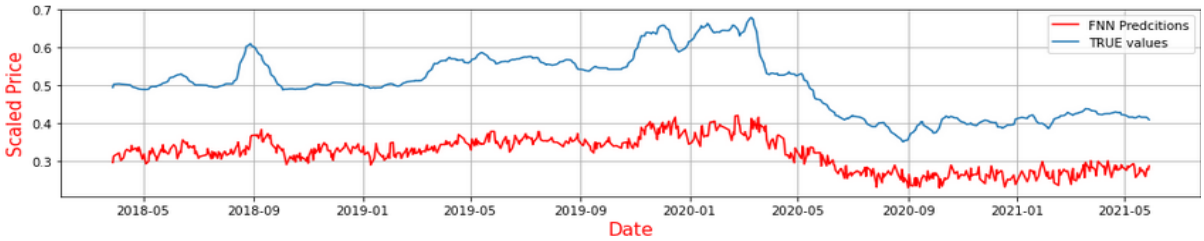
Safaricom PLC Long-short Term Memory Networks Predictions



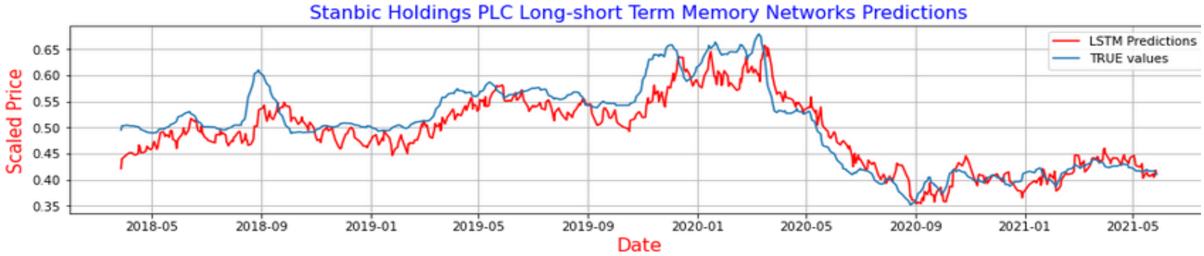
Safaricom PLC Gated Recurrent Unit Predictions



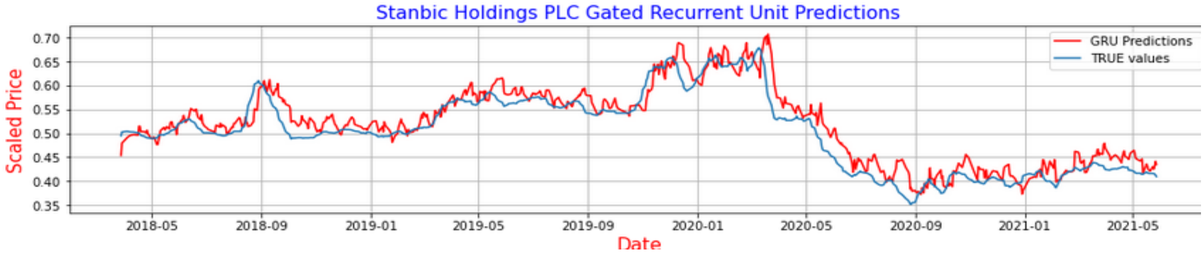
Stanbic Holdings PLC Feedforward Networks Predictions



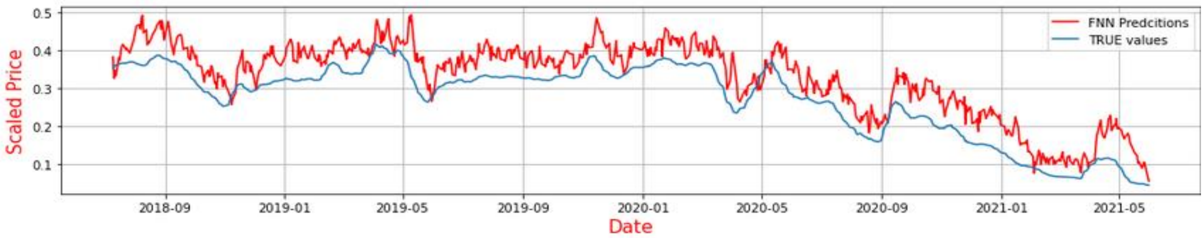
Stanbic Holdings PLC Long-short Term Memory Networks Predictions



Stanbic Holdings PLC Gated Recurrent Unit Predictions

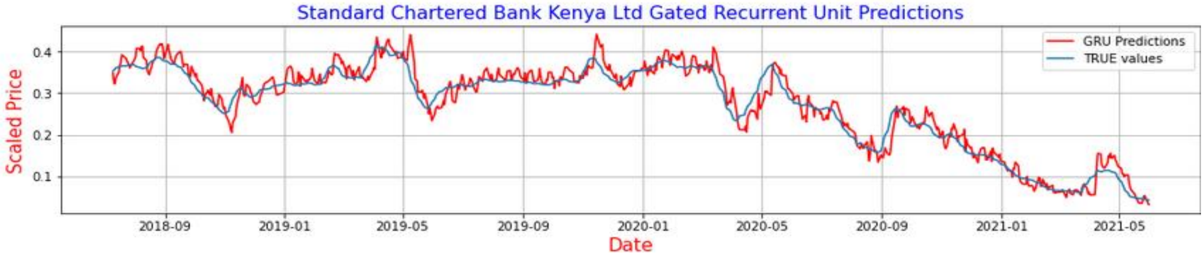


Standard Chartered Bank Kenya Ltd Feedforward Networks Predictions

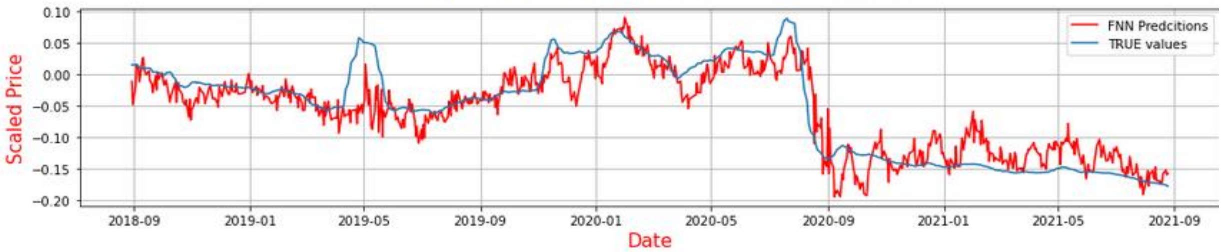


Standard Chartered Bank Kenya Ltd Long-short Term Memory Networks Predictions

Standard Chartered Bank Kenya Ltd Gated Recurrent Unit Predictions



WPP Scangroup Ltd Feedforward Networks Predictions



WPP Scangroup Ltd Long-short Term Memory Networks Predictions

WPP Scangroup Ltd Gated Recurrent Unit Predictions