Strathmore
UNIVERSITY

MODELLING OF STOCK MARKET VOLATILITY USING HYBRID WAVELET TRANSFORM DATA PREPROCESSING AND ARTIFICIAL NEURAL NETWORK IN THE KENYAN SECURITIES MARKET.

Esther Okoti

Thesis presented in fulfillment of the academic requirement for the degree of Msc Mathematical Finance & Risk Analytics of Strathmore University

March 2024

0

Declaration and recommendation

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Signature .... Date 3/11/2024

Esther Okoti

Recommendation

This proposal has been submitted for assessment with our approval as supervisors according to Strathmore University regulations.

Signature .... Date 3/11/2024

Dr Samuel Chege
Strathmore University

Abstract

Financial time-series modelling is a complex task, and methods such as artificial neural networks have been used for over the past two decades. Deep neural networks have shown to be more efficient in many domains, including physics, engineering, biomedicine, signal processing, mathematics, and statistics. This study proposes a model that uses deep learning technique combined with discrete wavelet data preprocessing to improve stock volatility forecasting in the Kenyan frontier market. The study discusses the role of wavelet transforms in time series analysis and demonstrates their advantages in general and in time series denoising, using a sample of 4 stocks of time series data from the Nairobi Securities Exchange (NSE). The forecasting model proposed for financial time series compares the performance of a Discrete Wavelet Transform Long Short Term Memory and and standard Long Short Term Memory. The financial time series data is decomposed using the Discrete Wavelet Transform, and the resulting approximation and detail coefficients are used as input variables in a Long Short-Term Memory neural network to predict future stock returns. This study is built on the Python scripting environment and use TensorFlow as a system for deep learning for training, prediction, and comparison.

In order to establish the efficacy and superiority of the recommended hybrid model, the forecasting performance of the analyzed models is evaluated using four metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Bayesian Information Criterion (BIC), and Akaike Information Criterion (AIC). According to the empirical findings the DWT-LSTM models for all data sets used were better at predicting stock volatility than the standard LSTM models, with lower performance metrics values.

Keywords: *Wavelet Transform, Long Short Term Memory, Optimizers, Stock volatility.*

Contents

List of Figures

List of Tables

Acknowledgment

First, I thank God for His presence and providence, my family for being my truest support system, my classmates and friends for the camaraderie, and all the academic staff. A special debt of immeasurable profound gratitude is owed to DAAD having funded my graduate studies and continuously raised the bar by providing resources needed for the successful completion of this study.

Dedication

This thesis is lovingly dedicated to my family for all their support, love, care and protection. Above and beyond, Thank you!

Abbreviations

ABSA	Amalgamated Banks of South Africa
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutionary Neural Networks
CWT	Continuous Wavelet Transform
DSS	Decision Support System
DWT	Discrete Wavelet Transform
EMH	Efficient Market Hypothesis
GRUs	Gated Recurrent Units
HMM	Hidden Markov Model
KCB	Kenya Commercial Bank
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
NCBA	National Commercial Bank of Africa
NSE	Nairobi Securities Exchange
RMSE	Root Mean Square Error
RMSProp	Root mean squared propagation
RNN	Recurrent Neural Network
SSA	Singular Spectrum Analysis
SVM	Support Vector Machine
SVR	Support Vector Regression
XLNET	eXtreme Language understanding NETwork

Chapter 1

Introduction

1.1 Background to the study

The field of stock volatility modeling has been a subject of great interest among researchers who have dedicated significant efforts to explore various methods for predicting stock volatility. As a result, there has been a continuous emergence of new approaches and techniques directed at enhancing accuracy and effectiveness of stock volatility forecasting. The advancement of cloud technology and large scale data stock volatility forecasting has emerged as a pressing research issue especially with development of artificial intelligence (AI) and computational intelligence, studies on stock market prediction have given these methods more consideration (Mintarya et al., 2023). Unlike standard time series methods, these approaches can effectively deal with the chaotic, noisy, and unpredictable nature of stock market data, resulting in more accurate predictions (Chen and Hao, 2017). As a result, these techniques offer novel and advantageous alternatives, which makes it appealing for researchers to use them for financial market forecasting.

The complexities of everyday stock trading reveal diverse patterns across different timescales, providing clues into future price movements. Short-term stock projections are based on frequent trade data, but long-term forecasts prioritize less frequent data. This makes stock markets to be chaotic systems that are naturally noisy, non-parametric, non-linear, and deterministic (Ashrafijoo et al., 2022, Serrano, 2022) and therefore the need to use sophisticated modelling to capture these characteristics for accurate forecasting.

Market players, whether individual or institutional investors, could continuously outperform the market if they could precisely predict market behavior. This incentivizes the development of advanced stock market prediction models that use innovative machine learning and computational intelligence approaches. As a result encouraging the development of precise stock market prediction models using the novel models of machine learning and computational intelligence techniques.

Empirical research in stock market prediction typically employs two main analytical methods: fundamental analysis and technical analysis. Fundamental analysis uses company-specific and market-related data to forecast stock values, whereas technical analysis relies on previous price and volume data. Fundamental analysis assumes that securities, including financial markets, have underlying values that investors can measure. Technical analysis, on the other hand, predicts chart behavior based on historical data, such as price movements and trading volume. Although both strategy has merits and disadvantages, empirical investigations show that neither regularly outperforms the other in accurately predicting stock prices. Some researchers argue that a combination of the two approaches, known as "quantamental" analysis, may provide the best results by combining the strengths of both approaches (Maltêz, 2022, Saini and Sharma, 2022, Sloan, 2019). Quantamental analysis attempts to combine the strengths of both approaches by using fundamental analysis to identify stocks with good long-term prospects and technical analysis to identify stocks that are undervalued or overvalued in the short term. This methodology aids investors in pinpointing stocks that have the potential to exceed market performance over extended periods.

In 1970, Fama introduced the EMH, suggesting that stock prices fluctuate randomly, resembling the characteristics of a random walk. As per the principle of random walks, future movements in a security's price level are indistinguishable from a series of completely random numbers. The hypothesis suggests that subsequent changes in price are independent random variables with an identical distribution. Therefore, the price changes have no memory, indicating that there is no reliable way to predict the future from the past. Furthermore, EMH implies that it is not possible to predict future price changes using current data. Fama's theory categorizes market efficiency into three types: weak-form, semi-strong-form, and strong-form efficiency (Fama, 1970).

Fama's weak-form EMH asserts that current stock prices already incorporate all past price movements, suggesting that technical analysis, which relies on historical price data, is unable to consistently outperform a simple buy-and-hold strategy. This idea was introduced by (Fama, 1965) and further discussed by (Leigh et al., 2002).

In contrast, the semi-strong form of the EMH, also proposed by Fama in 1965, argues that stock prices accurately reflect all publicly available information, including not only past prices but also other factors such as political events, economic indicators, interest rates, and company-specific information. (Wang et al., 2011) emphasized the significance of this publicly available information in influencing stock prices. Under the semi-strong form, active management strategies that rely on utilizing all available information may not consistently outperform passive strategies like holding a market index. This suggests that attempting to outperform the market through extensive research and analysis may not always be fruitful.

The strong form of the EMH, proposed by Fama in 1970, goes even further, claiming that all information, including insider knowledge, is already reflected in stock prices. This implies that even investors with access to insider information cannot consistently earn higher returns than the overall market (Leigh et al., 2002).

The EMH implies uncertainty in predicting stock market returns, particularly in its strongest form, as discussed by (Timmermann and Granger, 2004). Despite the EMH's assertion of rational pricing, various market anomalies challenge this hypothesis. These anomalies include phenomena such as herding behavior, recency bias, loss aversion, overreaction and underreaction, short-term momentum, long-term reversal, and fluctuations in asset price volatility (Fama, 1965). While the EMH provides a framework for understanding market efficiency, it does not fully explain all observed market behaviors. For example, (Borovkova and Tsiamas, 2019) noted that the EMH fails to account for certain market anomalies, suggesting limitations to its applicability.

The EMH serves as a foundational theory in finance, but its limitations and challenges highlight the ongoing debate surrounding market efficiency and the complexities of financial markets.

These anomalies indicate that financial markets are not always perfectly efficient, and investors may potentially profit by identifying and exploiting these market inefficiencies. However, it is crucial to acknowledge that these anomalies may be challenging to consistently exploit, as they could be short-lived and subject to sudden reversals. Behavioral finance has emerged as a means to reconcile the EMH and market anomalies by incorporating human psychology and behavioral biases (Chaqmakchi, 2022, Noreen et al., 2022). Understanding how human psychology and behavioral biases influence the market empowers investors to make more informed financial decisions.

1.2 Problem Statement

Time series prediction has always piqued the attention of the academic community, in part because of the practical applications but also because of the inherent intricacy of time-dependent data. There are several well-established methods for evaluating time-series data, but they have always been limited by the pre-processing and comprehension required to use them.

In the empirical studies of (Hegazy et al., 2014, Mehtab and Sen, 2020) artificial neural networks are used for stock forecasting, in contrast, due to the high level of noise in financial data (Dastgerdi and Mercorelli, 2022, Liu et al., 2022b, Passalis et al., 2021), the accuracy of these models is limited. Appropriate algorithms that remove noise from financial time series without affecting the real values is required to improve the performance of these models. Noise reduction algorithms such as the Wavelet Transform or Kalman Filter significantly improve the accuracy of LSTM models in predicting stock volatility (Dastgerdi and Mercorelli, 2022) by reducing the interference of noise in the data, these algorithms aid the neural network identify meaningful patterns in the market volatility, resulting in more precise predictions. The DWTs method offers a more optimal combination of wavelet transforms and thresholding methods, which further improves the accuracy of predictions.

1.3 Objectives

1.3.1 General Objective

To forecast financial time series based on a comparative analysis of the discrete wavelet transform (DWT) denoising Artificial Neural Network (ANN) and standard Artificial Neural Network (ANN) analysis.

1.3.2 Specific Objectives

- To explore the potential of a discrete wavelet Transform denoising technique in improving the accuracy of LSTM stock market volatility prediction in a frontier market.
- To evaluate the performance of DWT-LSTM and standard LSTM across a sample of three optimizers Adam, ADAgrad and RMSprop optimizers in predicting stock market volatility in a frontier market.

1.4 Significance of the Study

The research demonstrates the benefits of employing the hybrid wavelet transform to denoise time series data and artificial neural networks for modeling and forecasting in the Kenyan frontier market. Accurate forecasting of stock market volatility is essential for the efficient functioning of financial markets. It enables investors to make informed decisions about buying, selling, or holding their assets, and enables financial institutions to manage their risks more effectively. The study combines two powerful techniques, wavelet transform and ANN, to preprocess the data and create a forecasting model that takes into account the complex, non-linear relationships that affect stock market volatility. This hybrid approach enhances the accuracy of volatility forecasts, thereby enabling investors and financial institutions to make more informed decisions regarding their investments and risk management strategies.

The potential to promote the adoption of innovative techniques in financial forecasting, which leads to the development of more sophisticated and efficient models. The study has broader implications in the scientific discourse of artificial intelligence and machine learning in frontier markets.

Chapter 2

Literature review

2.1 Introduction

Accurate modeling of market volatility requires taking into consideration events that include recessions, expansions, and phases with high or low volatility. The observed volatility in stock market returns and prices is attributable to the fact that necessary rates of return, which are influenced by cyclical and other short-term changes in aggregate demand, are similarly very volatile. There are various approaches for modeling time series data. To explicitly depict time effects, one technique is to utilize a simple function of the delayed values of the time series, which is usually a linear function.

With popular models either AR or ARMA (Gunawan and Astika, 2022, Putri et al., 2021), this technique to time series data analysis is widely used in statistics literature. The alternative technique involves indirectly expressing temporal effects via latent variables, which are designed to maintain the memory of the data's dynamics. These latent variables, also known as hidden states, are updated on a regular basis utilizing data from the current time step as well as information carried over from prior time steps. The second category comprises recurrent neural networks (RNN), which were originally developed in cognitive research but are now frequently used in computer science. State space models, which are widely used in econometrics and statistics, are another form of model that implicitly describes time.

Classical mathematical methods, such as the Kalman filter (Dastgerdi and Mercorelli, 2022, Li et al., 2022) and autoregressive models (Gunawan and Astika, 2022, Rubi et al., 2022), rely on multiple measurements taken over time to estimate unknown variables while accounting for statistical noise and inaccuracies. The Autoregressive Integrated Moving Average (ARIMA) model (Gunawan and Astika, 2022, Putri et al., 2021) has been widely used to model financial time series, including stock prices and commodity prices. Conversely, it requires the time series data to be stationary, which may result in a loss of interpretability and structure. With a few exceptions, almost all classical models assume that data has a linear relationship. Because real-world time series data are frequently nonlinear, this assumption raises serious concerns about the robustness of classical time series models. Deep learning autoregressive models, such as the LSTM network, (Dastgerdi and Mercorelli, 2022, Hu et al., 2020), Multi task transformer model (Mirjebreili et al., 2022), BERT (Costola et al., 2023) and XL-NET (Cui et al., 2023) have become increasingly popular in finance owing to their capacity to capture extensive nonlinear correlations between variables, particularly sentiment analysis in stock forecasting.

There are differing views on the efficacy of linear and nonlinear models, which are used to analyze past market changes and make market predictions. While non-linear models are considered superior in some studies, linear models can work equally well or better in others. Researchers have attempted to analyze nonlinear time series and develop techniques for separating noise from time series data, such as Padding-Based Fourier Transform Denoising (Song et al., 2021) was used to several deep learning models based on time series in order to forecast the closing prices of the S&P500, SSE, and KOSPI. Results indicate that the combination of the proposed denoising method and deep learning models approach not only performs better than the fundamental models in terms of prediction accuracy but also reduces the time lag issue.

Singular Spectrum Analysis (SSA) (Broomhead et al., 2020, Fathi et al., 2022), which decomposes the original time series into a series of singular values that contain independent information. The empirical study of (Hill and Motegi, 2019) performed blockwise white noise tests to investigate the noise that exists in financial markets. They discovered that while the white noise hypothesis cannot be accepted for financial markets in the United States and the United Kingdom, it may be accepted for financial markets in China and Japan, demonstrating weak-form efficiency. In order to distinguish between the roles of various types of information and noise in stock price movements, (Brogaard et al., 2022) proposed a return variance decomposition model and discovered that 31% of return variation is related to noise. (Zargar et al., 2019) developed the "opening noise trading model" to assess the share of noise in the opening price of the stock market, while (Liang et al., 2019) proposed a new multioptimal combination wavelet transform approach to reduce distortion in signal reconstruction. This is based on the assumption that the noise in the stock's opening price has no bearing on the real price change brought on by the disclosure of fresh information. Additionally, they provide evidence that the opening stock price of all Nifty stocks frequently includes noise.

(Chandar et al., 2016) proposes a forecasting model for financial time series based on Discrete Wavelet Transform (DWT) and Artificial Neural Network (ANN). Five datasets were used to apply the suggested model. Accuracy tests for each dataset revealed that the proposed model performs better than a conventional approach. It also demonstrated that the hybrid forecasting method outperformed methods that did not employ the wavelet transform in terms of performance. Other studies have proposed various models and methods for predicting stock prices and separating noise from time series data, including the SVR-ARMA model for stock price prediction based on wavelet denoising (Zhang, 2021), and the self-identification ResNet-ARIMA order model (Khanarsa et al., 2020). These empirical studies emphasize the importance of denoising time series data before using in neural networks to improve the performance accuracy in frontier market volatilty analysis.

2.2 Wavelet Analysis

The stock market is a market where investors with different characteristics make decisions over various time horizons ranging from minutes to years. These decisions are made on different time scales, including both speculative and investment activities (Gong et al., 2022). In such a complex environment, wavelet analysis is a valuable analytical tool to handle the differences in time horizons and time scales involved in economic decisions. The analysis of economic and financial time series in the frequency domain is an important area of research, particularly in frontier markets where time series' statistical features are time-variant and evolutionary (Torrence and Compo, 1998). The frequency domain analysis of time series involves transforming the data from the time domain to the frequency domain using methods such as the Fourier

Transform or Wavelet Transform. This allows for the identification of periodic patterns or cycles in the data that may not be apparent in the time domain.

In the context of economic and financial time series, frequency domain analysis identifies cyclical patterns in macroeconomic indicators such as GDP, inflation, and interest rates. It also identifies periodicities in financial time series such as stock prices, exchange rates, and commodity prices. The wavelet transform analyzes a signal using a succession of wavelets, each with a different scale, as opposed to the Fourier Transform, which employs a series of sine waves with various frequencies. The benefit of utilizing a wavelet is that, unlike their counterparts in the Fourier Transform, wavelets are localized in time (Chao et al., 2014, Torrence and Compo, 1998). Overall, using wavelet threshold-denoising techniques in combination with deep neural networks offers a powerful tool for predicting stock volatility and maximizing profits.

Wavelets were first introduced by French mathematician and physicist Jean Morlet and his colleagues in the mid-1980s (Grossmann and Morlet, 1984). Morlet, along with mathematicians Yves Meyer, Ingrid Daubechies, and others, played a key role in the development of wavelet theory (Daubechies, 1992, Meyer, 1990), which has since become an important mathematical tool in a wide range of fields, including signal processing, image compression, and data analysis with applications in time-frequency analysis, feature extraction, and denoising.

Wavelet transform includes Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT) (Zhang, 2021). The wavelet transforms decompose a signal at different timescales. It is defined as a set of basic functions $\psi_{a,b}(t)$ that can be generated by translating and scaling the mother wavelet as follows (Daubechies, 1992),

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right); \quad a > 0, \, -\infty < b < \infty, \tag{2.1}$$

where a is the scale parameter and b determines the location of the wavelet.

The CWT is used for mapping the changing properties of non-stationary signals. However, CWT is a time-frequency representation of a signal $f(t)$ that can be defined by the following equation,

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}^*(t)dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi_{a,b}^* \left(\frac{t-b}{a}\right) f(t)d, \tag{2.2}$$

where $\psi_{a,b}^*$ is the mother wavelet conjugate (Percival et al., 2004, Poularikas, 1999). The Continuous Wavelet Transform (CWT) requires the use of a complex conjugate pair of mother wavelets. This is because the CWT is based on convolving the signal with a scaled and translated version of the mother wavelet and its complex conjugate, which allows for both positive and negative frequency components to be analyzed. The complex conjugate pair of wavelets ensures that the CWT is able to capture both the amplitude and phase information of the signal.

The wavelet coefficients $W_f(a,b)$ are obtained by continuously varying the scale parameter and the position parameter in order to select the different portions of the signal and analyze the different scale variations (Mallat and Mallat, 1999). The constituent wavelets of the original signal are obtained by multiplying each coefficient by the appropriate scaled and shifted wavelet. The mostly used mother wavelet for CWT is the "Morlet" function which extracts features with equal variance in time and frequency.

The DWT is a discrete set of the wavelet scales and translations. It is specially adapted for the sampled value (Daubechies, 1992). However, this transform decomposes the signal into a mutually orthogonal set of wavelets. This specificity represents the main difference between DWT and CWT. The DWT employs a dyadic grid, where the mother wavelet is scaled by power two ($a = 2^j$) and translated by an integer ($b = k2^j$), where k is a location index running from 1 to $2^{-j}N$ (N is the number of observations) and j runs from 0 to J (J is the total number of scales),

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k). \tag{2.3}$$

The DWT coefficients are obtained from the following expression:

$$W_{j,k} = W(2^j, k2^j) = 2^{-j/2} \int_{-\infty}^{\infty} f(t)\psi(2^{-j}t - k)dt. \tag{2.4}$$

The means of the Inverse Discrete Wavelet Transform (IDWT) is calculated in order to reconstruct the original signal (or its parts) from the wavelet coefficients $W_{j,k}$ such that:

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} W_{j,k} \psi_{j,k}(t), \tag{2.5}$$

where $\psi_{j,k}(t)$ are the wavelet basis functions and $W_{j,k}$ are the corresponding wavelet coefficients. The wavelet basis functions are obtained by dilating and translating a mother wavelet function $\psi(t)$, according to the parameters j and k. The parameter j controls the scale of the wavelet, with j corresponding to finer scales, while k controls the location of the wavelet in time.

The DWT technique is well suited for noise filtering, data reduction, and singularity detection, making it an excellent option for time series data processing. Medical time series data, audio and video data, and image data have all been subject to extensive investigations on the decomposition of an original time series, with excellent results (Sarkar et al., 1998, Strang and Nguyen, 1996).

2.3 Artificial Neural Networks

Advancements in artificial intelligence and machine learning, as well as the availability of vast amounts of data and increased computing power, deep learning methods have emerged as a promising approach to stock volatility prediction creating new opportunities for developing advanced techniques for predicting stock volatility (Hegazy et al., 2014, Liu et al., 2022a, Mehtab and Sen, 2020, Milosevic, 2016, Song, 2018). These methods rely on neural networks to automatically learn complex patterns and relationships from the historical data, and use them to predict the future trends of the stock market. Deep learning methods such as recurrent neural networks, Convolutional Neural Networks, and long short-term memory networks have been applied in the field of stock volatility prediction, and have shown promising results in terms of accuracy and performance (Kumar and Thenmozhi, 2014). As a result, there is a growing demand for an accurate and dependable predictive model that can comprehensively capture the unpredictable and nonlinear nature of the market.

The recent surge in artificial neural network (ANN) research activities has demonstrated that neural networks have powerful pattern classification and prediction capabilities. ANNs have been successfully used for a wide range of tasks in business, industry, and science (Bhagya Raj and Dash, 2022, Bhatt and Shrivastava, 2022, Mumali, 2022, Okoro et al., 2022). There exists several variants of machine learning techniques that have been developed for application to the stock forecasting problem in literature. These variants may differ in their underlying algorithms, feature selection methods, training procedures, and hyper parameters, among other factors. The development and evaluation of these variants are typically based on empirical studies that use historical market data to assess their predictive performance, often measured in terms of accuracy, precision, recall, F1-score, or other metrics.

Despite the advancements in stock volatility prediction methods, it is important to note that no single method can provide a perfect solution to this problem. The stock market is a highly complex and dynamic system that is affected by numerous internal and external factors, making it difficult to accurately predict its future trends. Therefore, a combination of different approaches and expert analysis is necessary to make informed decisions and maximize returns in the stock market. (Kumar and Thenmozhi, 2014) made a study on the development of several machine learning techniques, Artificial Neural Networks (ANNs), Convolutionary Neural Networks (CNN), Support Vector Machines (SVMs), Decision Support System (DSS), Support Vector Regression (SVR), Naive Bayes networks and Hidden Markov Model (HMM) in stock market prediction. However, due to the limitations and strengths of each technique, researchers have sought to enhance forecasting accuracy by combining methods, such as KNN + SVM (Puspitasari and Rustam, 2018) or ANN + SVM (Kurani et al., 2023). Feature selection, extraction, and optimization techniques, such as principal component analysis, genetic algorithms, wavelet transforms, and particle swarm optimizations, have been utilized to improve forecast accuracy. Additionally, clustering as an unsupervised learning method has been examined for predicting stock prices. Recently, deep learning methods have gained renewed interest in stock market prediction. Deep neural networks, Convolutional Neural Networks, and long short-term memory networks have been applied to extract relevant features from complex and noisy data and identify hidden nonlinear relationships.

2.3.1 Long Short Term Memory

LSTM is a powerful and popular type of neural network that is commonly used in natural language processing, speech recognition, and other sequence prediction tasks (Ghimire et al., 2022, Khalil and Pipa, 2022, Shashidhar et al., 2022). In this study the focus is in using LSTM for stock market volatility modelling. It is a type of recurrent neural network (RNN) architecture that is designed to handle the vanishing gradient problem, which occurs when training deep neural networks for stock forecasting. LSTM was introduced in 1997 by Sepp Hochreiter and Jürgen Schmidhuber (Hochreiter and Schmidhuber, 1997).

At the time, traditional RNNs were limited in their ability to learn long-term dependencies, due to the vanishing gradient problem the tendency for gradients to shrink as they are propagated backwards through time (Olah, 2015). Hochreiter and Schmidhuber's innovation was to introduce a new type of memory cell that could selectively forget or remember information over long periods of time.

LSTM networks use a memory cell that is responsible for maintaining information over a sequence of inputs. The memory cell has three gates: the input gate, forget gate, and output gate. These gates control the flow of information into and out of the memory cell, allowing the network to selectively retain or discard information from previous inputs. The arichtecture of the LSTM Network is shown on Appendix A.1.

The architecture of LSTM networks makes them particularly effective in handling long-term dependencies in sequential data. The memory cell allows the network to remember information from earlier inputs, even if it has been many time steps since that information was last seen. There have been many extensions and variations of the original LSTM architecture, such as gated recurrent units (GRUs) (Gupta et al., 2022) introduced by (Cho et al., 2014). It combines the input and forget gates into a single "update gate." Additionally, it combines the hidden state and cell state and performs some other adjustments. The resulting model, which has been gaining popularity, is less complex than traditional LSTM models. Peephole LSTM (Adwait Dathan and Shanmuga Priya, 2022) introduced by (Gers et al., 2000) allows the gate layers to observe the state of the cell. Coupled LSTM (Chen et al., 2022) which instead of separately deciding what to forget and what we should add new information to, we make those decisions together. We only forget when we're going to input something in its place. We only input new values to the state when we forget something older. Bidirectional LSTM (Althelaya et al., 2018) which has the ability to process input data in both forward and backward directions. All these variants Staudemeyer and Morris (2019) have improved the performance of recurrent neural networks for various tasks.

2.3.2 Optimizer

The type of optimizer employed considerably influences how quickly the algorithm converges to the minimal value. Also, some idea of randomization is required to prevent staying stuck in a relative minimum and failing to achieve the absolute minimum. Adam is one of the more effective stochastic optimization techniques that only needs first-order gradients and requires little memory (Kingma and Ba, 2014). It combines the benefits of two well-known techniques, AdaGrad (Duchi et al., 2011), which performs well with sparse gradients, and RMSProp (Zou et al., 2019), which performs well in non-linear and non-stationary environments. Adam is an effective and efficient method that has been utilized successfully in earlier studies (Gudla and Bhoi, 2023, Liu et al., 2023). The benefits are summarized into its simple implementation, modest memory needs, and suitability for non-stationary cases. By identifying a number of parameters to minimize the objective function, it optimizes the deep learning model. As a result, it is thought that the Adam optimized LSTM neural network is an effective tool for predicting stock volatility.

2.4 Research Gaps and Discussion of Literature Review

The limitations of Artificial Neural Networks (ANNs) in handling nonlinear and non-stationary data in time series modeling have been acknowledged in the literature. As a result of the high autocorrelation of time series data, ANNs may produce forecasts that are similar to the last observed data. This may result in the prediction being a continuation of historical trends rather than accurately reflecting high-frequency and irregular changes for predictions (De Vos and Rientjes, 2005). Additionally, most time series data contain noise, making it essential to eliminate this noise to manage non-stationary data better.

Despite these limitations, ANNs have still received a lot of attention in time series modeling because they provide accurate predictions when the data is properly pre-processed, and the model is well-designed. Through the literature review researchers have developed various approaches to improve ANN's prediction accuracy in modeling, such as hybrid models that combine ANNs with other techniques such as wavelets or fourier transform algorithms, and data pre-processing techniques such as detrending, deseasonalization, and normalization.

In summary, while ANNs have some limitations in handling time series data, they remain a popular and effective technique for time series modeling when the data is pre-processed and the model is appropriately designed. The LSTM to be specific is always further designed as it permits many different variants and topologies for specific problems Staudemeyer and Morris (2019).

Chapter 3

Methodology

Introduction

The data sources, screening techniques, and variable definitions utilized are described in this section.

3.1 Research Design

The study adopts a quantitative research as statistical inference will be used to draw conclusions on the study. The goal of this thesis's empirical investigation is to forecast stock volatility using a comparative analysis of a wavelet transform data preprocessing LSTM and a standard LSTM in the Kenyan frontier securities market.

3.2 Data

The Nairobi Securities Exchange has 63 listed companies (active and inactive) as of 2022. A sample of 4 data sets from the banking sector will be used in the construction of the experiments. The banking sector plays a crucial role in the economy by facilitating financial intermediation i.e managing risks, and providing financial services. Researching this sector provides insights into the broader economic landscape. Banking is highly regulated due to its systemic importance and potential impact on financial stability. Studying the regulatory framework and its implications is a significant area of research interest. The data used are the daily closing price of over a span of 10 years, recorded from January 2013 to December 2022 because it is long enough to capture variations in the data while still being relatively recent and relevant to current conditions of diverse macroeconomic and political events. Assessed to other frequencies, daily data is preferred for this experiment because intraday (tick-level) data is often much more noisy and has lower predictive value, while data from longer intervals is less useful realistically. Features such as momentum indicators which are commonly used in stock chart analysis are engineered from daily data. In this study, the focus is on a sample 4 largest and most liquid single banking sector stocks available in the dataset. For each stock, the goal is to engineer the same feature and target variable, training our neural network on all stocks in the prescribed sample so as to generalize the model, rather than just focusing on learning one stock.

Acquiring the stock price data and extracting the relevant signal to be denoised, in this case, the daily returns.

Computed the daily returns for each of the selected stocks using equation below,

$$r_{i,t+1} = \ln\left(\frac{P_{i,t+1}}{P_{i,t}}\right) \quad \text{for } i = 1, \ldots, 4 \tag{3.1}$$

In this equation, $r_{i,t+1}$ represents the log return for asset i at time $t+1$. $P_{i,t+1}$ denotes the price of asset i at time $t+1$, and $P_{i,t}$ represents the price of asset i at time t. The $\ln(\cdot)$ function represents the natural logarithm. The notation "$i = 1, \ldots, 4$" indicates that the equation is valid for assets i ranging from 1 to 4.

3.2.1 Wavelet Denoising

Selection of Wavelet

The wavelet transform is applied to the signal using a chosen db4 wavelet function, decomposition level of 2, and threshold processing of 2 parameters. It has properties that meet the requirements of discrete wavelet transform. Its characteristics are as follows;

(1) Compact support: Which means it has a limited range where it is non-zero and is zero outside of that range. This characteristic allows it to effectively capture localized features in the data with a sharp drop-off in performance.

[0]

The data is sourced from `https://www.investing.com`.

(2) Small support length: The db4 wavelet has a relatively small support length. It conveniently shortens computation time compared to some other wavelets, reducing data processing time and training time.
(3) Compression: The db4 provides a good trade-off between compression efficiency and reconstruction quality.
(4) Numerical Stability: db4 wavelet is known for their numerical stability and desirable mathematical properties, which makes it widely used in scientific and engineering applications.

Selection of Threshold Parameter

The universal thresholding method is used, this aims to automatically adapt the threshold value based on the characteristics of the data. For each specified level within the range $[m, n]$. The formula for the threshhold value is as follows:

$$\lambda = \sqrt{(}2 * \log 2(cD)), \tag{3.2}$$

here, cD represents the detail coefficients at the current decomposition level. The threshold value λ is calculated for each level within the specified range $[m, n]$. The method adapts to the length of the detail coefficients, making it a data-driven approach that aims to reduce noise while preserving important signal features. This helps determine which coefficients should be retained and which should be set to zero during the thresholding step of wavelet denoising.
This thesis's is based on the Python language environment and uses TensorFlow as the deep learning framework for training, prediction, and comparison. The wavelet transform is performed using the pywt.wavedec() function, which decomposes the input sequence into approximation coefficients (cA) and detail coefficients (cD) for each level.

Selection of Threshold Functions.

The wavelet coefficients are denoised using the soft thresholding method, which involves shrinking the coefficients towards zero based on a computed threshold value, effectively removing the noise from the signal. The soft thresholding method is applied to detail coefficients for the levels. If a coefficient at a given level has an absolute value greater than or equal to this threshold, it is retained (with soft thresholding applied), and if it is smaller, it is set to zero. The expression is as follows;

$$S(x,\lambda) = \begin{cases} \text{sign}(x)\cdot(|x|-\lambda) & \text{if } |x| > \lambda \\ 0 & \text{if } |x| \leq \lambda \end{cases}, \tag{3.3}$$

where sign(x) is the signum function, which returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$. The soft thresholding function shrinks the magnitude of the input signal x towards zero by an amount proportional to the difference between the magnitude of x and the threshold value λ. If the magnitude of x is less than or equal to λ, then the output is set to zero.

The wavelet transform decomposes the signal into different frequency sub-bands, with the high-frequency sub-bands containing noise and low-frequency sub-bands containing relevant information. The resulting denoised signal is then reconstructed by performing the inverse wavelet transform as follows;

$$x(t) = \sum_{i=0}^{n}\sum_{j=0}^{N-1} h_j^{(i)} \cdot \phi_j^{(i)}(t) + \sum_{i=1}^{n}\sum_{j=0}^{N-1} d_j^{(i)} \cdot \psi_j^{(i)}(t), \tag{3.4}$$

where:

n is the level of decomposition, N is the length of the signal, $h_j^{(i)}$ is the approximation coefficient at level i and position j, $\phi_j^{(i)}(t)$ is the scaling function at level i and position j, $d_j^{(i)}$ is the detail coefficient at level i and position j, $\psi_j^{(i)}(t)$ is the wavelet function at level i and position j, t is the time variable. The wavelet reconstruction formula computes the original signal $x(t)$ by adding up the approximation and detail coefficients obtained during the wavelet decomposition. The approximation coefficients are obtained by applying the low-pass filter to the original signal, while the detail coefficients are obtained by applying the high-pass filter to the original signal.

The denoised data is used as the train-test split of ratio 80:20. The data is then reshaped to fit the input format of the LSTM network. The LSTM model is defined with two LSTM layers with 100 units each, and a dense output layer.

3.2.2 LSTM Network

The first step in the LSTM is to decide what information is going to be thrown away from the cell state. The forget gate layer is responsible for deciding what information to discard from the previous cell state C_{t-1}, based on the input at time step $t-1$ (h_{t-1}) and the current input (x_t). This is represented by the equation;

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f),$$

where f_t is the forget gate vector, σ is the sigmoid activation function, W_f and b_f are the weight matrix and bias vector of the forget gate layer, and $[h_{t-1}, x_t]$ is the concatenation of the previous hidden state h_{t-1} and the current input x_t. The forget gate vector f_t has the same dimensions as the cell state C_{t-1}, and is used to modulate which elements of the previous cell state to forget and which to remember.

The next step is to choose new information that will be kept in the cell state. Two sections make up this, the sigmoid layer (input gate layer) and the tanh layer. These two are combined in the subsequent phase to produce an update to the state. The input gate layer is responsible for determining which information to update in the cell state C_{t-1}, based on the current input x_t and the previous hidden state h_{t-1}. This is represented mathematically as;

$$i_t = \sigma(W_f.[h_{t-1}, x_t] + b_i),$$

where i_t is the input gate vector, σ is the sigmoid activation function, W_i and b_i are the weight matrix and bias vector of the input gate layer, and $[h_{t-1}, x_t]$ is the concatenation of the previous hidden state h_{t-1} and the current input x_t. The input gate vector i_t has the same dimensions as the cell state C_{t-1}, and is used to modulate which elements of the previous cell state to update and which to leave unchanged.

The next step is to generate a new candidate cell state $\tilde{C}t$ based on the current input x_t and the previous hidden state $ht - 1$. This is done using a hyperbolic tangent (tanh) activation function to squish the values to the range [-1, 1]. This represented mathematically as;

$$\tilde{C}t = \tanh(W_C.[h_{t-1}, x_t] + b_C),$$

where $\tilde{C}t$ is the candidate cell state, tanh is the hyperbolic tangent activation function, W_C and b_C are the weight matrix and bias vector of the tanh layer, and $[ht - 1, x_t]$ is the concatenation of the previous hidden state h_{t-1} and the current input x_t. The candidate cell state $\tilde{C}t$ has the same dimensions as the cell state $Ct - 1$, and represents the new information that could be added to the cell state.

The cell state is regulated through a combination of additive and multiplicative interactions. Here's the equation for updating the cell state ;

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,$$

where, $\odot$ represents element-wise multiplication, f_t forget gate, i_t input gate and $\tilde{C}t$ is the candidate cell state. This equation shows how the cell state is updated at each time step in an LSTM network.

Finally,the output gate is responsible for deciding which information from the cell state C_t to output as the hidden state h_t, based on the current input x_t and the previous hidden state h_{t-1}. This is represented by the equation as;

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o),$$

where O_t is the output gate vector, σ is the sigmoid activation function, W_o and b_o are the weight matrix and bias vector of the output gate layer, and $[h_{t-1}, x_t]$ is the concatenation of the previous hidden state h_{t-1} and the current input x_t. The output gate vector O_t has the same dimensions as the cell state C_t, and is used to modulate which elements of the cell state to output and which to ignore.

The next step is to generate the actual output by combining the filtered cell state and the output gate vector. This is done by first passing the cell state C_t through a hyperbolic tangent (tanh) activation function to squish the values to the range [-1, 1], and then multiplying it element-wise by the output gate vector O_t. This is represented mathematically as;

$$h_t = O_t. \tanh(C_t),$$

where h_t is the output (hidden) state, tanh is the hyperbolic tangent activation function, and C_t is the cell state after the input and forget gate layers have updated it. The final output h_t has the same dimensions as the hidden state h_{t-1} and the current input x_t, and represents the filtered version of the cell state that we have decided to output. The sigmoid activation functions are used to output a number between 0 and 1 for each element of the cell state. A value of 1 means that the corresponding information should be output completely, while a value of 0 means that the information should be ignored completely. The arichtecture of the LSTM Network is shown on Appendix A.1.

3.2.3 Optimizer

The ADAgrad optimizer effectively employs a distinct learning rate for each parameter and time step. The logic behind ADAgrad is that rare parameters must have higher learning rates, whilst frequent parameters must have lower learning rates. In other terms, the stochastic gradient descent update for ADAgrad is;

$$\eta_t = \frac{\alpha}{\sqrt{\sum_{i=1}^{t} g_i^2 + \epsilon}},$$

g_i represents the cost function's gradient with respect to the parameter at the i-th iteration, and ϵ is a tiny constant for numerical stability. The learning rate is derived based on the previous gradients obtained for each parameter. Hence,

$$\eta_t = \frac{\alpha}{\sqrt{G_t}},$$

where G_t is the matrix of sums of squares of previous gradients. The problem with this optimization is that as the number of iterations increases, the learning rates begin to decline rapidly.

RMSprop considers addressing a dropping learning rate by only using a limited number of past gradients. Updates become

$$\eta_t = \frac{\alpha}{\sqrt{v_t} + \epsilon},$$

where

$$v_t = \gamma v_{t-1} + (1-\gamma) g_t^2.$$

v_t is an exponentially decaying average of the squares of past gradients. The learning rate η_t is scaled by $\frac{\alpha}{\sqrt{v_t+\epsilon}}$, where the square root of v_t acts as a kind of normalization factor for the learning rate adjustment. This approach helps to mitigate the vanishing and exploding gradient problem by adapting the learning rate based on the magnitude of the gradients. It allows for more stable and efficient training, especially when dealing with deep neural networks and non-convex optimization problems.

Adam, or Adaptive Moment Estimation, calculates adaptive learning rates for each parameter by taking into account the continuously decaying average of past squared gradients and past gradients. This can be expressed as

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t.$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2.$$

The v_t and m_t can be thought of as estimates of the first and second moments of the gradients, respectively, hence the term Adaptive Moment Estimation. When this was first utilized, researchers discovered that there was an inherent bias towards 0, which they counteracted by using the following estimates:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}.$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}.$$

This brings us to the final gradient update rule.

$$\eta_t = \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}.$$

$$\theta_{t+1} = \theta_t - \eta_t \hat{m}_t.$$

The gradient update uses the moments of the distribution of weights, allowing for more statistically sound descent.

3.2.4 Dropouts

Another critical part of model training is keeping the weights from growing too large, which can lead to overfitting, in which the model focuses too much on specific data points. One technique to addressing this is to impose a penalty for large weights, with the definition of "large" depending on the sort of regularization applied.

To reduce overfitting, a technique explores the possibility of temporarily disabling some neurons, forcing the model to rely on a broader variety of neurons rather than becoming unduly reliant on specific groups. Dropouts are frequently used for this reason, increasing the robustness of neurons and encouraging them to forecast trends without focusing on a single neuron. With dropout, the mistake rate continues to fall; without dropout, the error rate flattens.

The Early Stopping callback is also used to prevent overfitting during training. It monitors the validation loss during training and stops training when the validation loss stops decreasing, thus preventing the model from overfitting. So, using early stopping allow us to train the model for more epochs, as long as it stops before it starts overfitting. The training history is plotted to visualize the loss over epochs. The Early Stopping callback takes several arguments:

Monitor: The quantity to be monitored in this case the val loss.

Patience: The number of epochs with no improvement in the monitored quantity after which training will be stopped. In this case the patience=10, training will stop if the monitored quantity does not improve for 10 consecutive epochs.

Verbose: Whether to print information about early stopping to the console during training.

Mode: Whether to minimize or maximize the monitored quantity. In this case monitor='val loss' and mode='min', training will stop when the validation loss stops decreasing.

By monitoring the loss over epochs, one can observe whether the neural network is improving or not. Ideally, we want the loss to decrease over time, which indicates that the network is learning and improving its performance on the training data. However, if the loss stops decreasing or starts increasing, it is an indication that the network has reached its capacity or is overfitting to the training data.

3.3 Performance Metrics

Once the model is trained, it is used to make predictions on the testing set. The performance of the model is evaluated using mean absolute error, root mean squared error, Bayesian Information Criterion and Akaike Information Criterion.

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(|A_k - P_k|)^2},$$

where: N = the number of observations, A_k = the actual value of the kth observation, P_k = the predicted value of the kth observation

Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|A_k - P_k|,$$

where: N = the number of observations, A_k = the actual value of the kth observation, P_k = the predicted value of the kth observation.

Bayesian Information Criterion (BIC):

$$BIC = -2ln(L) + Kln(n),$$

where L is the likelihood of the model, k is the number of parameters in the model, and n is the number of observations in the data set.

Akaike Information Criterion (AIC):

$$AIC = -2ln(L) + 2K,$$

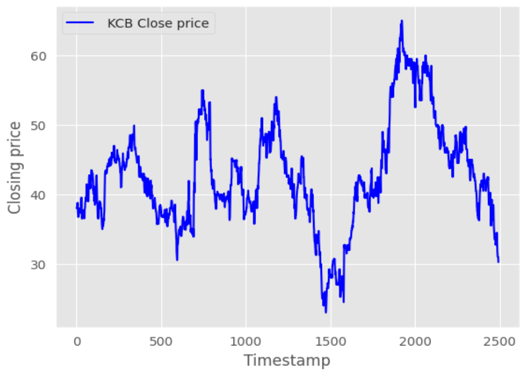
where L is the likelihood of the model and k is the number of parameters in the model.

In all cases, lower values of the indicate a better fit of the model to the data.

Chapter 4

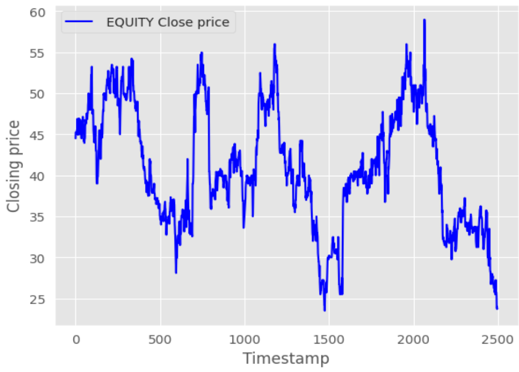
Data Analysis, Results and Discussions

4.1 Exploratory Data Analysis

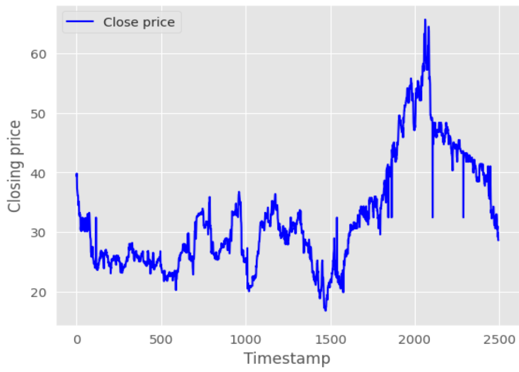


(a) KCB Price

(b) ABSA Price



(c) EQUITY Price

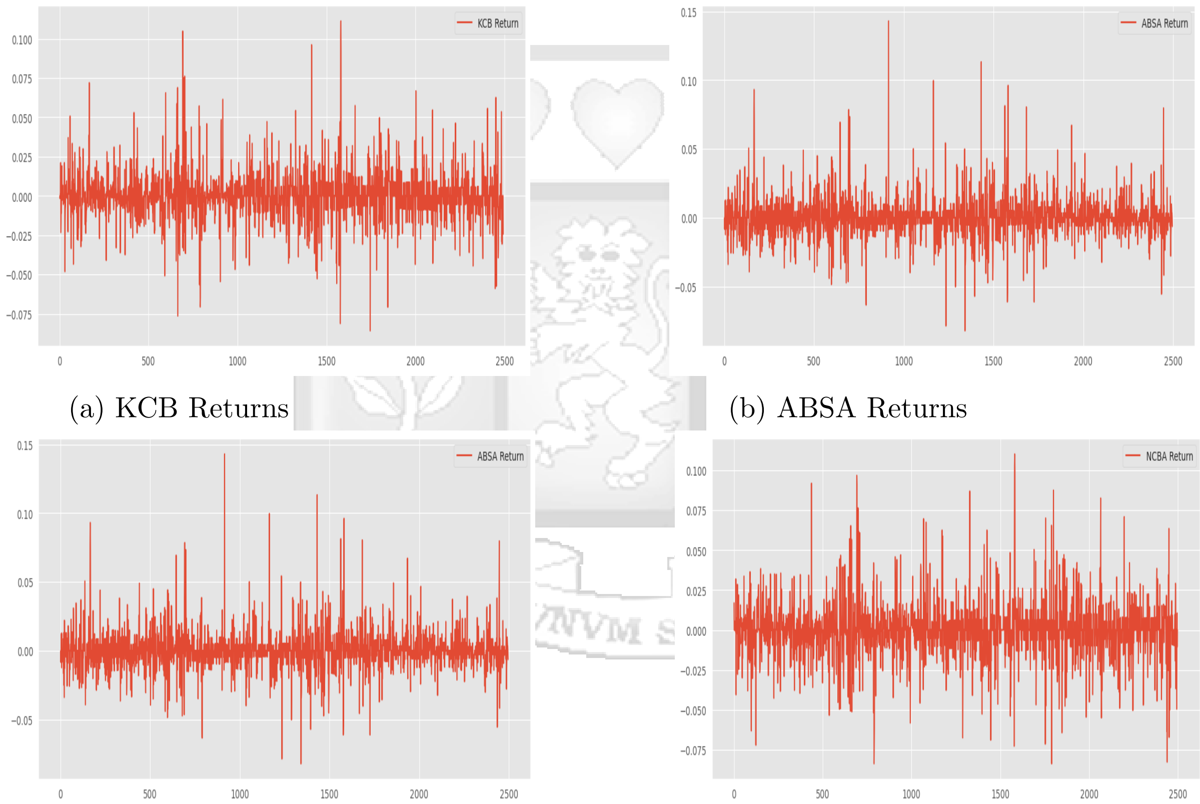


(d) NCBA Price

Figure 4.1: Closing Price Trends for the period January 2013 to December 2022

Figure 4.1 above, displays the evolution of the closing prices over the sample period, the data ranges from January 2013 to December 2022. There are sharp dips for all the stocks towards the end of 2019 and beginning of 2020 which may have been brought about by the effects of COVID- 19 as investors withdrew to safer assets such as bonds, a steady increase towards end of 2020 beginning of 2021 the economy was in recovery and finally a decreases as at end of 2022, which may have been attributed by the macroeconomic economic factors at that time. The price movements of the assets are unpredictable and follow a random pattern. The behavior of price trends in a random walk are explained using the efficient market hypothesis (EMH). This means that a price increase or decrease in the past has no bearing on the likelihood of a similar price movement in the future.

As a result, the price trends in a random walk appears to be unpredictable and erratic, with no discernible pattern or trend.

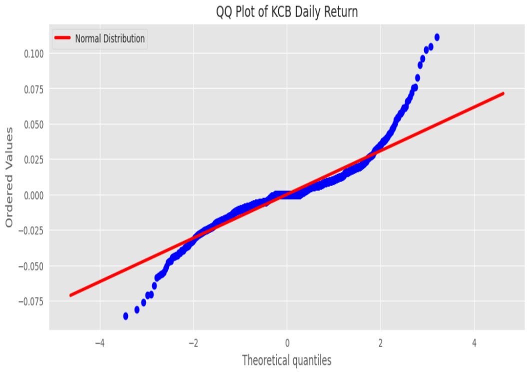


(a) KCB Returns (b) ABSA Returns

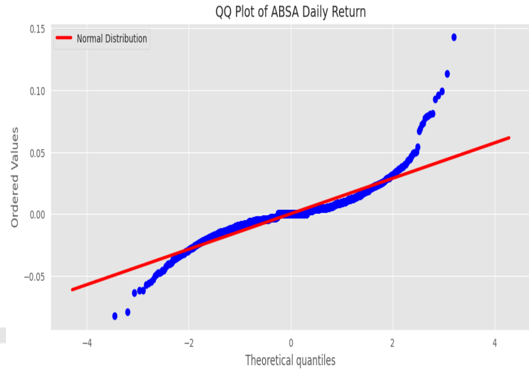
(c) EQUITY Returns (d) NCBA Returns

Figure 4.2: Daily Returns of the data sets the period January 2013 to December 2022

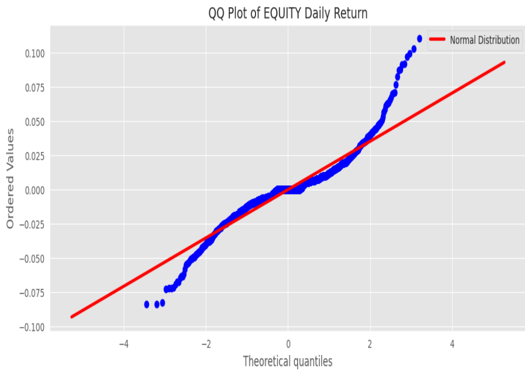
Figure 4.2 displays the evolution of the daily returns of the data sets continuously compounded from the closing prices. An assessment of the trend characteristics of the time series reveals a stylised fact about financial time series. There is volatility clustering effect on the compounded returns on all the data sets attributed to the fact that financial markets are not perfectly efficient, and that information is not always immediately incorporated into prices. The volatility clustering phenomenon observed asserts that where periods of high volatility tend to be followed by periods of high volatility, and periods of low volatility tend to be followed by periods of low volatility. This indicates that volatility tends to cluster over time, rather than being spread randomly.



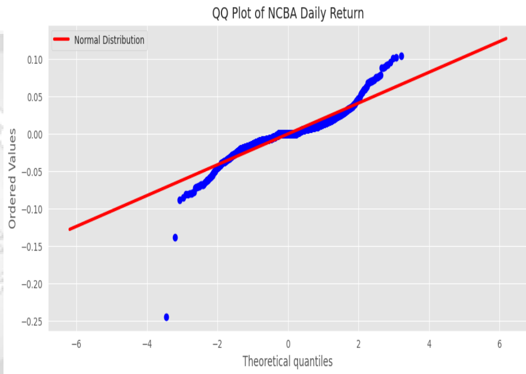
(a) KCB QQ Plot



(b) ABSA QQ plot



(c) EQUITY QQ plot



(d) NCBA QQ Plot

Figure 4.3: QQ plots of daily Returns

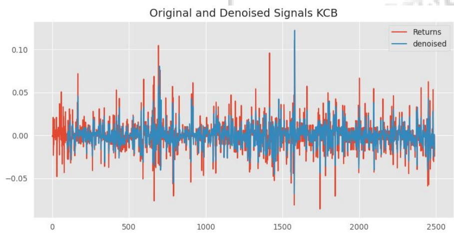
Figure 4.3 displays the normality probability of the data sets. The figure illustrates the QQ plot associated with the returns of the data sample. It has an S-shaped curve which means that over the sample period there seem to be large fluctuations from the normal distribution over the tails.

Table 4.1: The table shows the descriptive statistics of returns data for the period (January 2013 to December 2022).

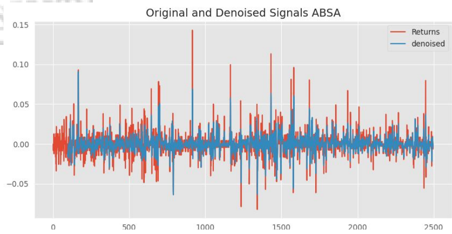
	KCB	ABSA	EQUITY	NCBA
Mean	0.000025	0.000199	−0.000097	0.000087
Maximum	0.111111	0.142857	0.110169	0.104167
Minimum	−0.085714	−0.082251	−0.083799	−0.245000
Std. Dev.	0.015413	0.014302	0.017601	0.020641
Skewness	0.662837	1.224411	0.365943	−0.519796
Observations	2492	2496	2497	2495
Kurtosis	7.407761	12.305327	5.386072	12.305405

KCB, ABSA and EQUITY data sets have a positive skewness value indicating that the distribution has a long tail to the right, while NCBA has a negative skewness value indicating that the distribution has a long tail to the left. All the datasets have high kurtosis values above 3 indicating that the distributions have sharp peaks and heavy tails.

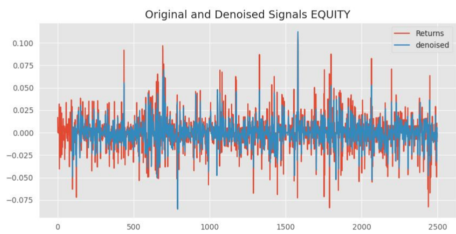
4.2 Sensitivity to the DWT denoising



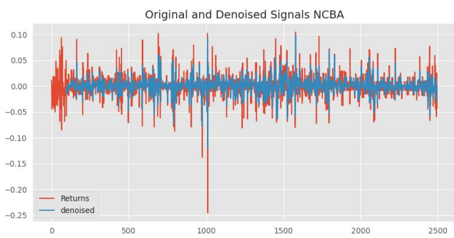
(a) KCB Volatility denoising



(b) ABSA Volatility denoising



(c) Equity Volatility denoising



(d) NCBA Volatility denoising

Figure 4.4: Wavelet Denoising

The visualization resulting from wavelet denoising of returns data consists of two plots. The first plot shows the original returns data in red and the denoised returns data in blue on the same axes. This allows us to observe how much noise has been removed from the signal and how well the denoised signal captures the important features of the original signal and whether there are any artifacts or distortions introduced by the denoising process. As observed the denoising process has removed noise without affecting the real values of the original data this is an important need in the analysis process.

4.3 Forecasting, Optimizer and Model Comparisons.

Table 4.2: Summary of Simulation Results Adam Optimizer

BANK	MODEL	RMSE	MAE	AIC	BIC
KCB	LSTM	0.14285461	0.09413755	241792	724613
	DWT-LSTM	0.00486723	0.00379738	239160	721037
ABSA	LSTM	0.09808798	0.05944971	241491	724013
	DWT-LSTM	0.00361827	0.00258440	239316	706972
EQUITY	LSTM	0.17059743	0.11498182	241937	724754
	DWT-LSTM	0.00626951	0.00446096	239594	712606
NCBA	LSTM	0.08838256	0.05533667	241408	723931
	DWT-LSTM	0.03129270	0.01626006	241021	703646

Table 4.3: Summary of Simulation Results RMSprop Optimizer

BANK	MODEL	RMSE	MAE	AIC	BIC
KCB	LSTM	0.14492746	0.09643944	241804	724625
	DWT-LSTM	0.01176501	0.00881133	239840	721735
ABSA	LSTM	0.10316837	0.07065492	241532	724053
	DWT-LSTM	0.00859835	0.00611359	239917	707580
EQUITY	LSTM	0.17205951	0.11643077	241944	724761
	DWT-LSTM	0.01442161	0.010652068	240112	713139
NCBA	LSTM	0.10355925	0.07672291	241535	724056
	DWT-LSTM	0.01428342	0.01102133	240380	703118

Table 4.4: Summary of Simulation Results Adagrad Optimizer

BANK	MODEL	RMSE	MAE	AIC	BIC
KCB	LSTM	0.14601971	0.09528068	241810	724681
	DWT-LSTM	0.01273805	0.00962460	239906	721797
ABSA	LSTM	0.09821318	0.05947165	241492	724014
	DWT-LSTM	0.00862699	0.00598847	239919	707582
EQUITY	LSTM	0.17297235	0.11554726	241948	724765
	DWT-LSTM	0.01382641	0.01023687	240159	713183
NCBA	LSTM	0.08874557	0.05595954	241411	723934
	DWT-LSTM	0.01234328	0.00865592	240276	703019

Table 4.2, 4.3 & 4.4 displays the forecasting outcomes of the suggested technique (DWT-LSTM) and the standard LSTM across the 3 optimizers ADAM, ADAgrad and RMSprop. The Long Short Term Memory (LSTM) performance measures obtained with the wavelet transform are less than those obtained with the conventional Long Short Term Memory technique, which is based on original data.

Using the KCB bank dataset as an example, the RMSE, MAE, AIC, and BIC calculated using the conventional methodology and based on the original data set as per the Adam optimizer are 0.14285461, 0.09413755, 241792, and 724613, respectively. The values produced using the suggested method (DWT-LSTM) as per the Adam Optimizer in comparison are, 0.00486723, 0.00379738, 239160, and 721037, respectively. As a result, the suggested technique predicts stock volatility with lower errors. Similar conclusions are obtained with the other sampled stocks in the sector and the sampled optimizers used.

For the KCB, ABSA and EQUITY bank datasets, the DWT-LSTM model with the Adam Optimizer performs better than all the other models, for NCBA DWT-LSTM model with the Adagrad Optimizer performs better than all the other models. This finding emphasizes the importance of using multiple optimizers and tune their hyper-parameters to find the one that yields the best performance for a given task.

All four of the analyzed stocks in the banking sector have a strong impact from this discrete wavelet denoising factor and even with the sample of the 3 optimizers.

4.4 Discussion

In order to estimate the stock volatility of securities in the sampled Kenyan banking securities frontier market, this research sought to introduce a hybrid forecasting technique that combines the discrete wavelet transform with Long Short Term Memory neural network. The historical data are first divided up using the described technique's discrete wavelet transform. The generated approximation and detail coefficients are then utilized as an input variable to predict the future stock volatility once the original data has been decomposed. The wavelet coefficients are plotted using separate subplots for each level as shown on Appendix B.1. According to the simulation findings, the approximation coefficients coupled with detail coefficients resulted in higher accuracy compared to conventional model that uses original data series.

The training history is plotted to visualize the loss over epochs as shown on Appendix D.1. The loss function is plotted on the y-axis and the number of epochs on the x-axis. Since it is important to monitor the loss over epochs to ensure that the neural network is learning effectively and not over fitting to the training data. The Early Stopping callback is used to prevent over fitting during training. Observing the trend in the plots Appendix D.1, the model data sets with a DWT-LSTM Adam optimizer is performing better than the data sets with any other model tuning as it is training for more epochs and hence better convergence which leads to better performance as the model has more time to learn and improve its weights.

Overall, the actual vs predicted volatility visualization allows us to evaluate the performance of the LSTM network in predicting volatility over time. It provides a clear way to observe how well the model is performing and to identify areas where it may need further refinement. As observed on Appendix C.1, the DWT-LSTM Adam optimizer performs better for all data sets as the line representing predicted volatility in blue is close to the line representing actual volatility in red as compared to the standard LSTM which has less convergence. The standard LSTM is consistently underestimating volatility across all data sets as it does not capture periods of high frequency accurately.

By combining these two techniques, the DWT-LSTM Adam optimizer hybrid approach, captures both the short-term and long-term patterns in the data, improving the accuracy of volatility forecasts. This is especially useful in financial applications, where accurate volatility forecasts is critical for risk management, investment decisions and monitoring. The empirical findings can be widely used in the selection of methods for processing time series data as wavelet decomposition and reconstruction improves the generalization ability of the LSTM prediction model as it catches the peaks and the prediction accuracy as visualized on Appendix C.1.

Chapter 5

Conclusion and Recommendations

Making good returns is the major motivation for stock market investments, which necessitates correct knowledge of the market, understanding price fluctuations and forecasting its future direction. Therefore, in order to predict future stock values, investors need strong and trustworthy instruments.

The main objective of this study was to forecast financial time series based on a comparative analysis of the DWT-LSTM and standard LSTM statistical analysis using three different optimizers Adam, Adagrad and RMSprop. The comparative analysis of DWT-LSTM and a standard LSTM has shown that the DWT-LSTM models improves the performance of LSTM. According to the empirical findings the DWT-LSTM models are better at predicting stock volatility than the standard LSTM model, with lower Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), AIC and BIC values. In real-world trading, this original model may produce false or incorrect signals due to market changes. As a result, investors may experience considerable losses. The conclusion of the study therefore is that incorporating denoising technique in modeling stock volatility is critical in ensuring more precise prediction of volatility are captured, and thus more accurate forecasts of future trends can be obtained.

A great deal of further research can be done in this relatively nascent area of financial time series prediction. There has been a growing interest in the use of wavelet denoising and deep neural networks for signal processing and analysis in various fields. There are multiple axes on which this research could vary in the experimental setup.

The first axis to consider is the challenges of using wavelets on the decomposition level. The choice of the decomposition level affects the accuracy of the analysis, as well as the computational complexity of the algorithm. If the decomposition level is too low, important signal details may be lost, leading to a less accurate analysis. On the other hand, if the decomposition level is too high, the resulting coefficients may contain noise or artifacts that can lead to inaccurate results.

Selecting the appropriate wavelet function for the analysis is important as different wavelet functions have different properties, such as the number of vanishing moments, which can affect the accuracy of the analysis. Choosing the right wavelet function requires domain-specific knowledge and experimentation. Understanding the different families of wavelets, one is able to select the appropriate wavelet for a given signal processing problem.

While deep neural networks have shown great potential in various applications, there are still challenges in their design, training, and interpretation. In particular, the choice of network architecture, optimization algorithm, and hyperparameters greatly impact the performance of deep neural networks. Therefore, more research is needed to develop effective and efficient deep neural network architectures and training algorithms for signal processing and analysis.

In addition, there is a need for more research on the integration of wavelet denoising and deep neural networks for signal processing and analysis. While these techniques have been used separately, their combination has the potential to improve the performance of signal processing and analysis. However, there are still challenges in the integration of these techniques, such as determining the optimal order of operations and addressing the potential loss of information in the denoising process.

The literature on the gaps in the use of wavelet denoising and deep neural networks highlights the need for more research to address these gaps and to further develop these techniques for signal processing and analysis in this field. By addressing these gaps, we potentially improve the performance of signal processing and analysis and enable new applications in areas of finance.

In conclusion, incorporating Volume data into the predictive model would be an additional intriguing modification that might be done; doing so should improve prediction validity by giving the model more data to draw on.

Bibliography

Adwait Dathan, R. and Shanmuga Priya, S. (2022). Yield forecast of soyabean crop using peephole lstm framework. In *Proceedings of Third Doctoral Symposium on Computational Intelligence: DoSCI 2022*, pages 261–270. Springer.

Althelaya, K. A., El-Alfy, E.-S. M., and Mohammed, S. (2018). Stock market forecast using multivariate analysis with bidirectional and stacked (lstm, gru). In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–7. IEEE.

Ashrafijoo, B., Farahmand, F.-h., Alavi Matin, Y., Rahmani, K., et al. (2022). Designing an optimal model using artificial neural networks to predict non-linear time series (case study: Tehran stock exchange index). *Journal of System Management*, 8(4):65–80.

Bhagya Raj, G. and Dash, K. K. (2022). Comprehensive study on applications of artificial neural network in food process modeling. *Critical reviews in food science and nutrition*, 62(10):2756–2783.

Bhatt, A. N. and Shrivastava, N. (2022). Application of artificial neural network for internal combustion engines: a state of the art review. *Archives of Computational Methods in Engineering*, 29(2):897–919.

Borovkova, S. and Tsiamas, I. (2019). An ensemble of lstm neural networks for high-frequency stock market classification. *Journal of Forecasting*, 38(6):600–619.

Brogaard, J., Nguyen, T. H., Putnins, T. J., and Wu, E. (2022). What moves stock prices? the roles of news, noise, and information. *The Review of Financial Studies*, 35(9):4341–4386.

Broomhead, D., Jones, R., King, G., and Pike, E. (2020). Singular system analysis with application to dynamical systems. In *Chaos, noise and fractals*, pages 15–27. CRC Press.

Chandar, S. K., Sumathi, M., and Sivanandam, S. (2016). Prediction of stock market price using hybrid of wavelet transform and artificial neural network. *Indian journal of Science and Technology*, 9(8):1–5.

Chao, B. F., Chung, W., Shih, Z., and Hsieh, Y. (2014). Earth's rotation variations: a wavelet analysis. *Terra Nova*, 26(4):260–264.

Chaqmakchi, D. W. (2022). Stock market and financial market efficiency.

Chen, Y. and Hao, Y. (2017). A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications*, 80:340–355.

Chen, Z., Leung, K. K., Wang, S., Tassiulas, L., Chan, K., and Towsley, D. (2022). Use coupled lstm networks to solve constrained optimization problems. *IEEE Transactions on Cognitive Communications and Networking*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Costola, M., Hinz, O., Nofer, M., and Pelizzon, L. (2023). Machine learning sentiment analysis, covid-19 news and stock market reactions. *Research in International Business and Finance*, page 101881.

Cui, Y., Jiang, Y., and Gu, H. (2023). Novel sentiment analysis from twitter for stock change prediction. In *Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II*, pages 160–172. Springer.

Dastgerdi, A. K. and Mercorelli, P. (2022). Investigating the effect of noise elimination on lstm models for financial markets prediction using kalman filter and wavelet transform. *WSEAS Trans. Bus. Econ*, 19:432–441.

Daubechies, I. (1992). *Ten lectures on wavelets*. SIAM.

De Vos, N. and Rientjes, T. (2005). Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences*, 9(1/2):111–126.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business*, 38(1):34–105.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417.

Fathi, A. Y., El-Khodary, I. A., and Saafan, M. (2022). Integrating singular spectrum analysis and nonlinear autoregressive neural network for stock price forecasting. *IAES International Journal of Artificial Intelligence*, 11(3):851.

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

Ghimire, S., Deo, R. C., Wang, H., Al-Musaylh, M. S., Casillas-Pérez, D., and Salcedo-Sanz, S. (2022). Stacked lstm sequence-to-sequence autoencoder with feature selection for daily solar radiation prediction: a review and new modeling results. *Energies*, 15(3):1061.

Gong, X., Zhang, W., Wang, J., and Wang, C. (2022). Investor sentiment and stock volatility: New evidence. *International Review of Financial Analysis*, 80:102028.

Grossmann, A. and Morlet, J. (1984). Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736.

Gudla, S. P. K. and Bhoi, S. K. (2023). A study on effect of learning rates using adam optimizer in lstm deep intelligent model for detection of ddos attack to support fog based iot systems. In *Computing, Communication and Learning: First International Conference, CoCoLe 2022, Warangal, India, October 27–29, 2022, Proceedings*, pages 27–38. Springer.

Gunawan, D. and Astika, W. (2022). The autoregressive integrated moving average (arima) model for predicting jakarta composite index. *Jurnal Informatika Ekonomi Bisnis*, pages 1–6.

Gupta, U., Bhattacharjee, V., and Bishnu, P. S. (2022). Stocknet—gru based stock index prediction. *Expert Systems with Applications*, 207:117986.

Hegazy, O., Soliman, O. S., and Salam, M. A. (2014). A machine learning model for stock market prediction. *arXiv preprint arXiv:1402.7351*.

Hill, J. B. and Motegi, K. (2019). Testing the white noise hypothesis of stock returns. *Economic Modelling*, 76:231–242.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hu, Y., Ni, J., and Wen, L. (2020). A hybrid deep learning approach by integrating lstm-ann networks with garch model for copper price volatility prediction. *Physica A: Statistical Mechanics and its Applications*, 557:124907.

Khalil, F. and Pipa, G. (2022). Is deep-learning and natural language processing transcending the financial forecasting? investigation through lens of news analytic process. *Computational Economics*, 60(1):147–171.

Khanarsa, P., Luangsodsa, A., and Sinapiromsaran, K. (2020). Self-identification resnet-arima forecasting model. *WSEAS transactions on systems and control*, 15(21):196–211.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, M. and Thenmozhi, M. (2014). Forecasting stock index returns using arima-svm, arima-ann, and arima-random forest hybrid models. *International Journal of Banking, Accounting and Finance*, 5(3):284–308.

Kurani, A., Doshi, P., Vakharia, A., and Shah, M. (2023). A comprehensive comparative study of artificial neural network (ann) and support vector machines (svm) on stock forecasting. *Annals of Data Science*, 10(1):183–208.

Leigh, W., Purvis, R., and Ragusa, J. M. (2002). Forecasting the nyse composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decision support systems*, 32(4):361–377.

Li, X., Feng, S., Hou, N., Li, H., Zhang, S., Jian, Z., and Zi, Q. (2022). Applications of kalman filtering in time series prediction. In *Intelligent Robotics and Applications: 15th International Conference, ICIRA 2022, Harbin, China, August 1–3, 2022, Proceedings, Part III*, pages 520–531. Springer.

Liang, X., Ge, Z., Sun, L., He, M., and Chen, H. (2019). Lstm with wavelet transform based data preprocessing for stock price prediction. *Mathematical Problems in Engineering*, 2019.

Liu, B., Yu, Z., Wang, Q., Du, P., and Zhang, X. (2022a). Prediction of sse shanghai enterprises index based on bidirectional lstm model of air pollutants. *Expert Systems with Applications*, 204:117600.

Liu, M., Yao, D., Liu, Z., Guo, J., Chen, J., et al. (2023). An improved adam optimization algorithm combining adaptive coefficients and composite gradients based on randomized block coordinate descent. *Computational Intelligence and Neuroscience*, 2023.

Liu, Q., Liu, Z., and Zhang, C. (2022b). Heteroscedasticity test of high-frequency data with jumps and market microstructure noise. *Applied Stochastic Models in Business and Industry*, 38(3):441–457.

Mallat, S. and Mallat, C. (1999). 7.2 classes of wavelet bases. *A Wavelet Tour of Signal Processing; Elsevier Science & Technology: Amsterdam, The Netherlands*, pages 241–254.

Maltêz, F. A. A. (2022). Deep reinforcement learning for investing: A quantamental approach for portfolio management. Master's thesis.

Mehtab, S. and Sen, J. (2020). A time series analysis-based stock price prediction using machine learning and deep learning models. *International Journal of Business Forecasting and Marketing Intelligence*, 6(4):272–335.

Meyer, Y. (1990). Orthonormal wavelets. In *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference, Marseille, France, December 14–18, 1987*, pages 21–37. Springer.

Milosevic, N. (2016). Equity forecast: Predicting long term stock price movement using machine learning. *arXiv preprint arXiv:1603.00751*.

Mintarya, L. N., Halim, J. N., Angie, C., Achmad, S., and Kurniawan, A. (2023). Machine learning approaches in stock market prediction: A systematic literature review. *Procedia Computer Science*, 216:96–102.

Mirjebreili, S. M., Solouki, A., Soltanalizadeh, H., and Sabokrou, M. (2022). Multi-task transformer for stock market trend prediction. In *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 101–105. IEEE.

Mumali, F. (2022). Artificial neural network-based decision support systems in manufacturing processes: A systematic literature review. *Computers & Industrial Engineering*, page 107964.

Noreen, U., Shafique, A., Ayub, U., and Saeed, S. K. (2022). Does the adaptive market hypothesis reconcile the behavioral finance and the efficient market hypothesis? *Risks*, 10(9):168.

Okoro, E. E., Obomanu, T., Sanni, S. E., Olatunji, D. I., and Igbinedion, P. (2022). Application of artificial intelligence in predicting the dynamics of bottom hole pressure for under-balanced drilling: Extra tree compared with feed forward neural network model. *Petroleum*, 8(2):227–236.

Olah, C. (2015). Understanding lstm networks.

Passalis, N., Kirtas, M., Mourgias-Alexandris, G., Dabos, G., Pleros, N., and Tefas, A. (2021). Training noise-resilient recurrent photonic networks for financial time series analysis. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1556–1560. IEEE.

Percival, D. B., Wang, M., and Overland, J. E. (2004). An introduction to wavelet analysis with applications to vegetation time series. *Community Ecology*, 5:19–30.

Poularikas, A. (1999). The handbook of formulas and tables for signal processing. ed. alexander d. poularikas boca raton: Crc press llc, 1999.

Puspitasari, D. and Rustam, Z. (2018). Application of svm-knn using svr as feature selection on stock analysis for indonesia stock exchange. In *AIP Conference Proceedings*, volume 2023, page 020207. AIP Publishing LLC.

Putri, R., Usman, M., Virginia, E., et al. (2021). Modeling autoregressive integrated moving average (arima) and forecasting of pt unilever indonesia tbk share prices during the covid-19 pandemic period. In *Journal of Physics: Conference Series*, volume 1751, page 012027. IOP Publishing.

Rubi, M. A., Chowdhury, S., Rahman, A. A. A., Meero, A., Zayed, N. M., and Islam, K. A. (2022). Fitting multi-layer feed forward neural network and autoregressive integrated moving average for dhaka stock exchange price predicting. *Emerging Science Journal*, 6(5):1046–1061.

Saini, A. and Sharma, A. (2022). Predicting the unpredictable: an application of machine learning algorithms in indian stock market. *Annals of Data Science*, 9(4):791–799.

Sarkar, T. K., Su, C., Adve, R., Salazar-Palma, M., Garcia-Castillo, L., and Boix, R. R. (1998). A tutorial on wavelets from an electrical engineering perspective. i. discrete wavelet techniques. *IEEE Antennas and Propagation Magazine*, 40(5):49–68.

Serrano, W. (2022). The random neural network in price predictions. *Neural Computing and Applications*, pages 1–19.

Shashidhar, R., Patilkulkarni, S., and Puneeth, S. (2022). Combining audio and visual speech recognition using lstm and deep convolutional neural network. *International Journal of Information Technology*, 14(7):3425–3436.

Sloan, R. G. (2019). Fundamental analysis redux. *The Accounting Review*, 94(2):363–377.

Song, D., Baek, A. M. C., and Kim, N. (2021). Forecasting stock market indices using padding-based fourier transform denoising and time series deep learning models. *IEEE Access*, 9:83786–83796.

Song, Y. (2018). *Stock trend prediction: Based on machine learning methods*. PhD thesis, UCLA.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Strang, G. and Nguyen, T. (1996). *Wavelets and filter banks*. SIAM.

Timmermann, A. and Granger, C. W. (2004). Efficient market hypothesis and forecasting. *International Journal of forecasting*, 20(1):15–27.

Torrence, C. and Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78.

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

Zargar, F. N., Kumar, D., et al. (2019). Opening noise in the indian stock market: Analysis at individual stock level. *Theoretical Economics Letters*, 9(01):21.

Zhang, X. (2021). Svr-arma stock price prediction model based on wavelet noise reduction. In *The Sixth International Conference on Information Management and Technology*, pages 1–4.

Zou, F., Shen, L., Jie, Z., Zhang, W., and Liu, W. (2019). A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11127–11135.

Appendix A

A.1 LSTM architecture

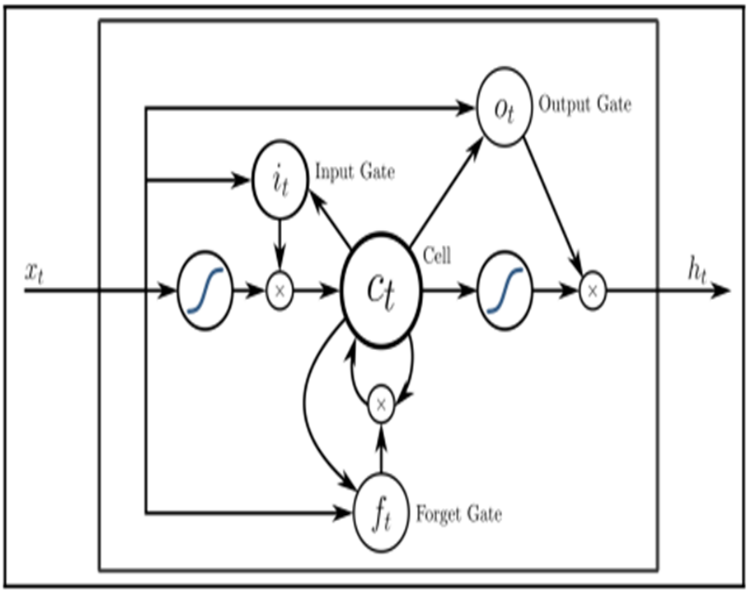


Figure A.1: Graph of LSTM unit

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f). \tag{A.1}$$

$$i_t = \sigma(W_f.[h_{t-1}, x_t] + b_i). \tag{A.2}$$

$$\tilde{C}t = \tanh(W_C.[h_{t-1}, x_t] + b_C). \tag{A.3}$$

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o). \tag{A.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t. \tag{A.5}$$

$$h_t = O_t. \tanh(C_t), \tag{A.6}$$

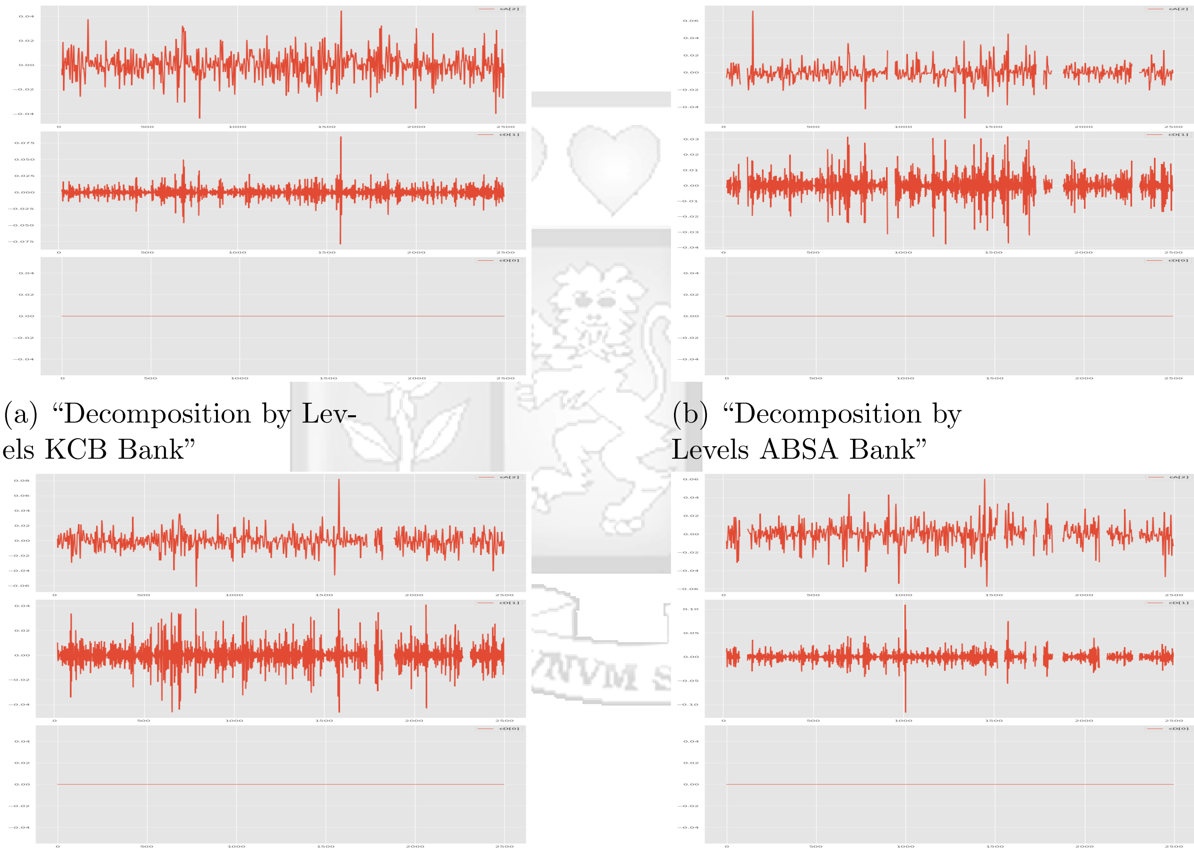
0

Graph of LSTM unit Source by `https://www.nomidl.com/deep-learning/what-is-lstm-and-how-does-it-work/`

Appendix B

B.1 Signal Approximations

The wavelet coefficients are plotted using separate subplots for each level as shown below;



(a) "Decomposition by Levels KCB Bank"

(b) "Decomposition by Levels ABSA Bank"

(c) "Decomposition by Levels Equity Bank"

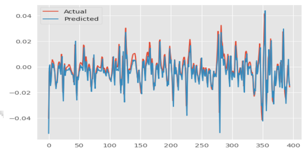
(d) "Decomposition by Levels NCBA Bank"

Figure B.1: Wavelet Decomposition at Level 2

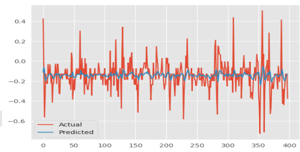
Appendix C

C.1 Actual vs Predicted Visualization

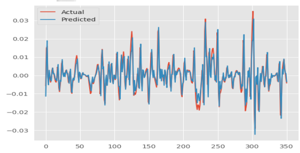
C.1.1 Adam Optimzer



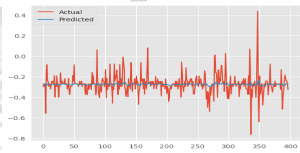
(a) KCB DWT-LSTM



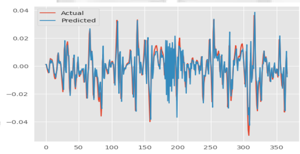
(b) KCB Standard LSTM



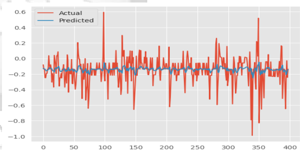
(c) ABSA DWT-LSTM



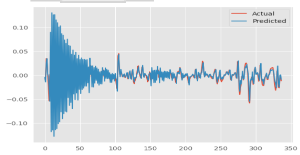
(d) ABSA Standard LSTM



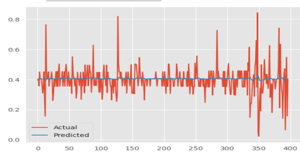
(e) EQUITY DWT-LSTM



(f) EQUITY Standard LSTM



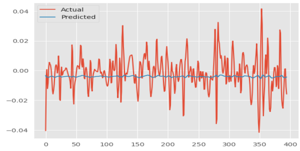
(g) NCBA DWT-LSTM



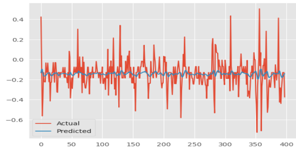
(h) NCBA Standard LSTM

Figure C.1: Actual VS Predicted of DWT-LSTM and Starndad LSTM ADAM Optimizer

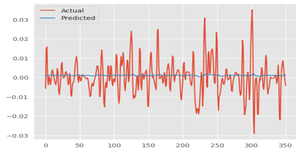
C.1.2 RMSprop Optimzer



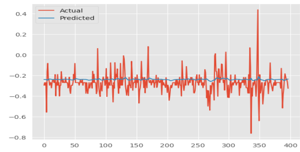
(a) KCB DWT-LSTM



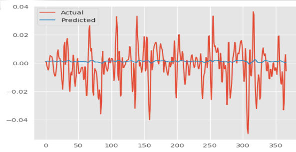
(b) KCB Standard LSTM



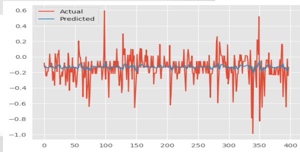
(c) ABSA DWT-LSTM



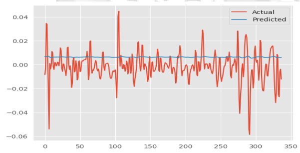
(d) ABSA Standard LSTM



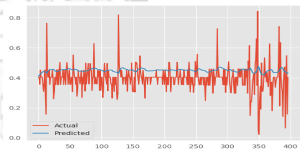
(e) EQUITY DWT-LSTM



(f) EQUITY Standard LSTM



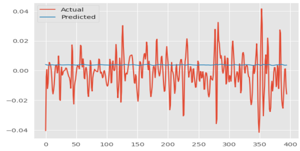
(g) NCBA DWT-LSTM



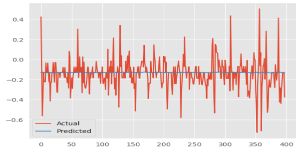
(h) NCBA Standard LSTM

Figure C.2: Actual VS Predicted of DWT-LSTM and Starndad LSTM RMSprop Optimizer

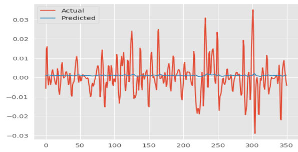
C.1.3 ADAgrad Optimzer



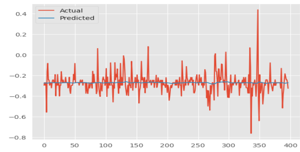
(a) KCB DWT-LSTM



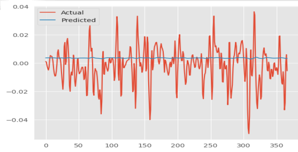
(b) KCB Standard LSTM



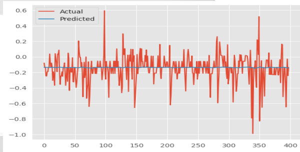
(c) ABSA DWT-LSTM



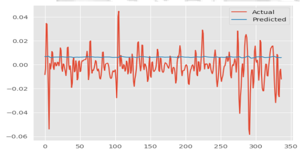
(d) ABSA Standard LSTM



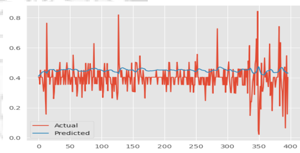
(e) EQUITY DWT-LSTM



(f) EQUITY Standard LSTM



(g) NCBA DWT-LSTM



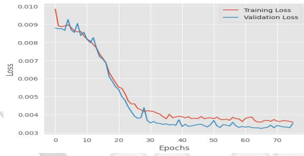
(h) NCBA Standard LSTM

Figure C.3: Actual VS Predicted of DWT-LSTM and Starndad LSTM ADAgrad Optimizer

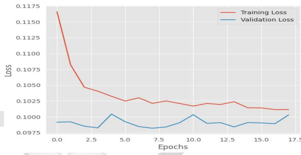
Appendix D

D.1 Sensitivity to Loss Over Epochs

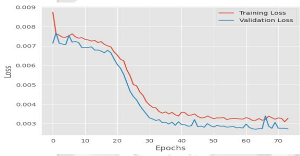
D.1.1 Adam Optimzer



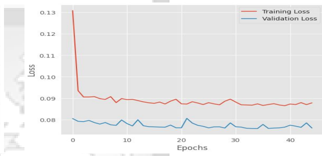
(a) KCB DWT-LSTM



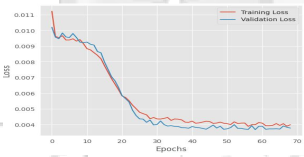
(b) KCB Standard LSTM



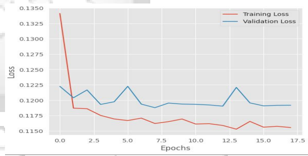
(c) ABSA DWT-LSTM



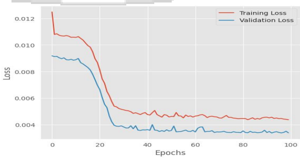
(d) ABSA Standard LSTM



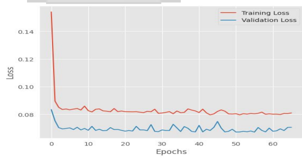
(e) EQUITY DWT-LSTM



(f) EQUITY Standard LSTM



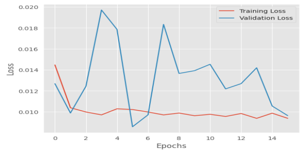
(g) NCBA DWT-LSTM



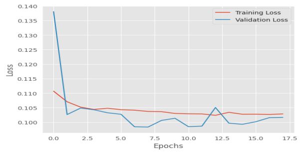
(h) NCBA Standard LSTM

Figure D.1: Loss Over Epochs of DWT-LSTM vs Standard LSTM ADAM Optimizer

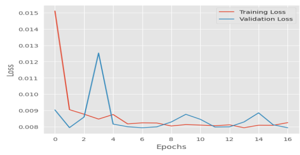
D.1.2 RMSprop Optimzer



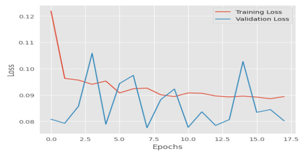
(a) KCB DWT-LSTM



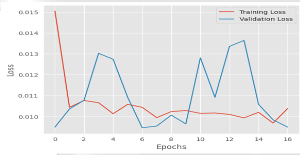
(b) KCB Standard LSTM



(c) ABSA DWT-LSTM

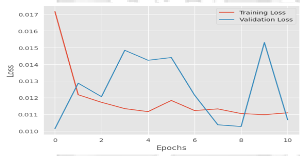


(d) ABSA Standard LSTM

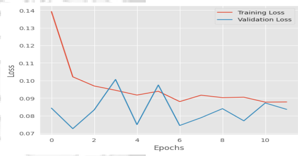


(e) EQUITY DWT-LSTM

(f) EQUITY Standard LSTM



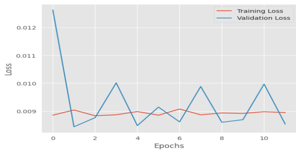
(g) NCBA DWT-LSTM



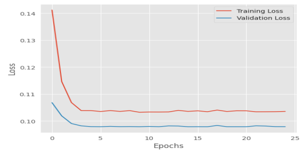
(h) NCBA Standard LSTM

Figure D.2: Loss Over Epochs of DWT-LSTM vs Standard LSTM RMSprop Optimizer

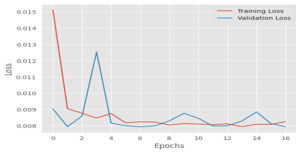
D.1.3 ADAgrad Optimzer



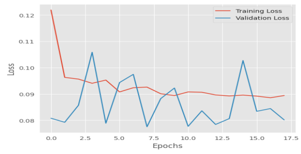
(a) KCB DWT-LSTM



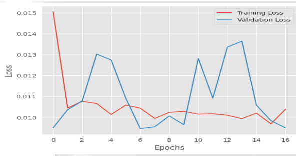
(b) KCB Standard LSTM



(c) ABSA DWT-LSTM

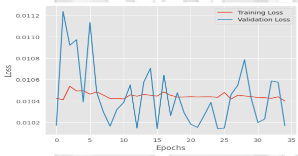


(d) ABSA Standard LSTM

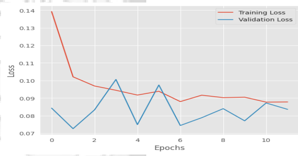


(e) EQUITY DWT-LSTM

(f) EQUITY Standard LSTM



(g) NCBA DWT-LSTM



(h) NCBA Standard LSTM

Figure D.3: Loss Over Epochs of DWT-LSTM vs Standard LSTM ADAgrad Optimizer

MODELLING OF STOCK MARKET VOLATILITY USING HYBRID WAVELET TRANSFORM DATA PREPROCESSING AND ARTIFICIAL NEURAL NETWORK IN THE KENYAN SECURITIES MARKET..pdf

ORIGINALITY REPORT

4%	7%	8%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	mdpi-res.com Internet Source	1%
2	blog.usejournal.com Internet Source	1%
3	www.researchgate.net Internet Source	1%
4	www.mdpi.com Internet Source	1%
5	idr.l2.nitk.ac.in Internet Source	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		

Strathmore UNIVERSITY

11th July 2023

Ms Okoti Esther,
esther.okoti@strathmore.edu

Dear Ms Okoti,

RE: Forecasting of Stock Market Volatility using Hybrid Wavelet Transform Data Preprocessing and Artificial Neural Network in the Kenyan Securities Market

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** research proposal. Your application reference number is **SU-ISERC1794/23.** The approval period is from **11th July 2023 to 10th July 2024.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Mr Ambrose Rachier,**
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
11-Jul-2023
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA