2023-06

Deep learning model for predicting stock prices in Tanzania

Joseph, Samuel

NM-AIST

https://doi.org/10.58694/20.500.12479/2204

DEEP LEARNING MODEL FOR PREDICTING STOCK PRICES IN TANZANIA

Samuel Thomas Joseph

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master's in Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

June, 2023

ABSTRACT

Stock price prediction models help to provide investors with tools for making better data-driven decisions. Machine learning and deep learning techniques have been successively utilized in various countries to develop these models. However, there is a shortage of literature on the efforts to exploit these techniques to forecast stock prices in Tanzania. Hence, this study was conducted to address this gap. The study selected active companies from the Dar es Salaam Stock Exchange (DSE) and developed Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU) models to forecast the next day closing prices of the companies. Long Short-Term Memory was the overall best model with a Root Mean Square Error (RMSE) of 4.1818 and Mean Absolute Error (MAE) of 2.1695. Findings revealed that it was significant to account for the number of outstanding shares of each company when developing a joint model for forecasting the stock prices of multiple companies. Specifically, LSTM attained an RMSE of 10.4734 before accounting for outstanding shares and 4.7424 after accounting for outstanding shares, showing an improvement of 54.72%. Furthermore, findings showed that investors' participation attributes helped to improve prediction accuracy. Specifically, LSTM realized an RMSE of 4.1818 when these attributes were appended from that of 4.7424 without them, showing an improvement of 11.8%. The resulting model was deployed in a web-based prototype, whereby, end-user validation results indicated that 76% of respondents rated the system as High in terms of its forecasting ability. In future, the study recommends exploration of more features.

DECLARATION

I, Samuel Thomas Joseph, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this dissertation is my original work and that it has neither been submitted nor is concurrently submitted for degree award in any other institution.

Samuel Thomas Joseph		26/7/2023
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Neema Mduma		26-07-2023
Name of Supervisor 1	**Signature**	**Date**

Dr. Devotha Nyambo		26/7/2023
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and, with this recommendation for acceptance by the Nelson Mandela African Institution of Science and Technology, a dissertation titled ***"Deep Learning Model for Predicting Stock Prices in Tanzania"*** in partial fulfilment of the requirements for the degree of the Master's in Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology.

Dr. Neema Mduma	Mduma	26-07-2023
Name of Supervisor 1	**Signature**	**Date**

Dr. Devotha Nyambo	Nyambo	26/7/2023
Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGEMENTS

First and foremost, I would like to thank the Almighty God for granting me life, sound health, and Grace throughout my study period, to Him alone be all the Glory.

Subsequently, special thanks go to my supervisors; Dr. Neema Mduma and Dr. Devotha Nyambo of the School of Computational and Communication Sciences and Engineering for guiding me throughout this study. Successively, I would like to thank Dr. Dina Machuve and Dr. Elizabeth Mkoba who also dedicated their time to offer intellectual support in the conduction of this work. I also extend my appreciation to Mr. Brian Chuwa for the expert advice received during data analysis being an investment advisor from Beacon Financials Limited, Tanzania.

Furthermore, I extend my sincere gratitude to the management of the Nelson Mandela African Institution of Science and Technology for giving me admission to pursue my studies and for the continued support for the whole period of my studies. Moreover, I sincerely appreciate the rest of the staff members of CoCSE for the constructive reviews and comments during the seminar presentations.

I am highly indebted to my parents; my father, Mr. Thomas Joseph, and mother, Mrs. Patricia Mwesiga, for the love and care throughout my life. I deeply appreciate my siblings Grace, Dorcas and Gideon for their invaluable prayers and inspiration throughout the two years of my Masters journey. I sincerely cherish my colleagues, to mention a few, Filbert Mlawa, Christian Elinisa, and Charles Paschal, for the togetherness and encouragement.

I thank the leadership of the companies visited during focus group discussions for their cooperation in setting up the meetings and necessary logistics. I also thank all respondents for allocating their valuable time to participate in this study.

DEDICATION

To stock intermediary companies, in Tanzania, seeking to improve investment advisory through state-of-the-art deep learning methods. To deep learning researchers, in Tanzania and beyond, striving to provide innovative solutions for economic development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

API	Application Programming Interface
ATS	Automated Trading System
Bi-LSTM	Bidirectional Long Short-Term Memory
CDS	Central Depository System
CMSA	Capital Markets and Securities Authority
CNN	Convolutional Neural Network
CoCSE	School of Computational and Communication Sciences and Engineering
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DEAP	Dar es Salaam Stock Exchange Enterprise Acceleration Program
DSE	Dar es Salaam Stock Exchange
EGM	Enterprise Growth Market
FGD	Focus Group Discussion
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HTML	Hypertext Markup Language
IPO	Initial Public Offering
KYC	Know Your Customer
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
NM-AIST	The Nelson Mandela African Institution of Science and Technology
NSE	National Stock Exchange
RMSE	Root Mean Squared Error

RNN	Recurrent Neural Network
RSI	Relative Strength Index
SMA	Simple Moving Average
SME	Small and Medium Enterprise
SRNN	Simple Recurrent Neural Network
SVR	Support Vector Regression
TOL	Tanzania Oxygen Limited
TSI	Tanzania Share Index
WAN	Wide Area Network
WFE	World Federation of Exchanges
WMA	Weighted Moving Average

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

A stock market is a place were buying and selling of shares takes place. A share is a financial security that represents part ownership of a company. That is to say, through buying of shares one becomes part owner of said company (Dingli & Fournier, 2017b). The stock market provides an alternative means for companies to secure capital through issuance of shares to the public. The acquired capital enables respective companies to increase their scales of operations, reaching more cities and villages; start new ventures or buy potential subsidiaries (Chhajer *et al.*, 2022). As such, a healthy stock market helps to facilitate economic development. On the other hand, shareholders can benefit from the stock market through dividend payouts when subscribed companies make profit, or through selling their shares when they increase in value (Ansah *et al.*, 2022).

The law of demand and supply holds that when the demand for a product increases or its supply decreases, the price of the product increases. Similarly, if the demand decreases or the supply increases, the product price decreases. The same applies for the stock market (Dingli & Fournier, 2017a; Massele *et al.*, 2015). Shares in stock markets are characterized by frequent rise and fall in prices (Bustos & Pomares-Quimbaya, 2020). An investor needs to buy when there is a good chance that the stock price will rise, in order to make profit from the difference between the selling price and the buying price. As a result; over the years, studies have been conducted to predict stock prices with the goal of making profit from the investments. Statistical and machine learning methods have been frequently utilized (Dingli & Fournier, 2017b).

Machine learning has proven to be an effective method, with many studies successfully employing a variety of machine learning techniques to predict stock prices (Dingli & Fournier, 2017b). Some of the techniques used are: Genetic algorithms (Huang & Li, 2017), Fuzzy logic (Ghanavati *et al.*, 2016), Ensembles (Huang *et al.*, 2018), Support vector machines (Ren *et al.*, 2019) and Rough sets (Kim & Enke, 2016). Recent advances in computational power and machine learning have led to the deep learning methods (Jiang, 2021). These methods can compute hidden non-linear relationships among features and thereby improve prediction (Hu *et al.*, 2021). They have been used successfully for text and image recognition. Recent studies have positively applied them in the stock market as well (Jiang, 2021).

The Dar es Salaam Stock Exchange (DSE) is the only stock exchange in Tanzania. It reserves the legal mandate to list and trade shares, and/or to delist them. It was formed as a result of the formation of the Capital Markets and Securities Act in 1994 and the creation of the Capital Markets and Securities Authority (CMSA) in 1995. It was incorporated in 1996 and began trading operations in 1998 (Munisi, 2017).

One of the major problems facing DSE is low level of financial literacy among the majority of the people (Massele *et al.*, 2015). Unlike developed countries, people in developing countries lack sufficient skills, knowledge and experience about stock markets (Munisi, 2017). Consequently, most of them tend to make poor investment decisions, fail to realize profit and eventually decline from investing. Hence, this study identified the need to develop a deep learning model for predicting stock prices in Tanzania. The model is expected to provide informed decision support through leveraging past historical stock data.

Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU) are deep learning techniques that were designed for handling time-series data such as stock data (Dixit *et al.*, 2018). Unlike traditional Recurrent Neural Networks, LSTM, Bi-LSTM and GRU can memorize previous information for a long time, as they do not suffer from vanishing and exploding gradients (Yanhui, 2021). This makes them suitable for stock market predictions because stock market trends tend to repeat over time (Obthong *et al.*, 2020). As a result, this study chose to employ these techniques to predict stock prices in Tanzania. However, from the review of related works, it was discovered that most of the studies that developed a joint model to predict the stock prices of multiple companies, did not take into account the number of outstanding shares of each company. Nonetheless, as shown in this study, the number of outstanding shares is an important factor to consider when building a joint model to predict the stock prices of multiple companies. Furthermore, it was discovered that most of the studies did not include investors' participation among the features. Investors' participation is a potential factor to consider because stocks are traded in the form of an auction and therefore investors' behaviors and ultimately the quality of their participation plays a major role in triggering price volatility (FinSights Lab, 2020). Therefore, this study took into account the number of outstanding shares of each company and incorporated the participation of both foreign and local investors among the features for the model, in order to ascertain whether doing so would help to improve prediction accuracy.

1.2 Statement of the Problem

There is low level of financial literacy among Tanzanians, whereby, the majority do not have sufficient skills, knowledge and experience about the stock market (Massele *et al.*, 2015; Munisi, 2017). This prohibits them from making appropriate returns from investing in the market thereby causing many to abstain from it. This poses a threat to future issuances and the sustainability of the market.

Furthermore, the size of the Dar es Salaam Stock Exchange (DSE) in the country is too small for efficiency (Abbas *et al.*, 2016). For a stock market to be efficient there needs to be a good number of listed companies in order to provide broad diversification to investors (Marcopolis, 2020). Provision of diversification is paramount as it promotes a healthy market by triggering creative initiatives from listed companies to retain and/or grow investor trust. However, currently the DSE consists of 28 listed companies which is a smaller number compared to those that perform better such as the Nairobi Stock Exchange of Kenya which consists of 63 listed companies or the National Stock Exchange of India which consists of 1641 listed companies to mention a few (Nairobi Securities Exchange, 2022; Raveendran, 2019).

Moreover, DSE lacks a competitive intermediary landscape in the country (FinSights Lab, 2020). The study acknowledged that there are few market intermediaries, whereby according to the DSE website, there are only 16 licensed market intermediaries. Worse than that, most of those intermediaries are concentrated in Dar es Salaam region despite the existence of 30 other regions in the country where 90% of the population resides (FinSights Lab, 2020). This culminates into low level of market penetration resulting in few number of investors and few number of transactions (FinSights Lab, 2020; Massele *et al.*, 2015).

Additionally, the market intermediaries lack efficient tools to supplement the investment advisory process (Massele *et al.*, 2015). Human experts are limited in that they cannot memorize long sequences of past data so as to couple it with present information in order to provide more accurate forecasts of future prices (Bustos & Pomares-Quimbaya, 2020). This necessitates the need to explore efficient technological solutions such as deep learning. Deep learning models have been developed in various countries to predict stock prices (Chen *et al.*, 2019; Labiad *et al.*, 2018; Nguyen *et al.*, 2019), however, it was observed that most of the developed models neither took into account the number of outstanding shares of the companies involved nor accommodated investors' participation among the predictive features. Besides, literature is not readily available related to exploitation of either machine learning or deep learning for predicting stock prices in Tanzania. Therefore, this study was conducted to address this gap.

1.3 Rationale of the Study

There have been positive initiatives by the DSE aimed at increasing its size in order to provide more portfolio diversification to investors. In particular, in 2013, the DSE launched its new segment known as the Enterprise Growth Market (EGM) whose goal is to provide flexible exchange-entry requirements to qualifying small and medium enterprises (SMEs) in order to enable them to raise capital from the public to meet their growth requirements (Capital Markets and Securities Authority, 2013). This initiative has already yielded fruits with six (6) SMEs getting listed between 2013 and 2020 inclusive. As a follow up to this initiative, in January 2020, DSE launched its Enterprise Acceleration Program (DEAP) which is dedicated to training and broadening the minds of entrepreneurs, business owners and managers, who are running SMEs, on different ways of procuring capital (Marcopolis, 2020). Furthermore, in 2021, DSE launched the "Endeleza Project", as an extension to its enterprise acceleration program. The project is a non-trading segment of the exchange that lists qualifying SMEs in order to profile and enhance their visibility to possible sources of private capital such as those in the form of private equity funds or angel investors; later, the enrolled private equities or angel investors, when they want to exit, could exit through the exchange (Endeleza Project, 2021). With these initiatives, the size of the DSE is expected to keep growing.

However, low financial literacy among the majority of the people, which causes them to make poor investment decisions and eventually refrain from investing, remains a major challenge to DSE (Munisi, 2017). Total dependence on expert advice may not always work due to human limitations and expedience or experience biases (Clifford, 2022). Therefore, this study adds to current DSE initiatives by developing a deep learning model for predicting stock prices in Tanzania. The model will complement the investment advisory process by learning from thousands of past historical stock data to provide informed price forecasts. As a result, the model will help to increase profit-making chances and thereby attract more investors into the market.

1.4 Research Objectives

1.4.1 Main Objective

The main objective of this study was to develop a deep learning model for predicting stock prices in Tanzania.

1.4.2 Specific Objectives

The study aimed to achieve the following specific objectives:

(i) To determine the relevant features for predicting stock prices in Tanzania.

(ii) To develop a deep learning model for predicting stock prices in Tanzania.

(iii) To deploy and validate the developed model.

1.5 Research Questions

The study intended to answer the following questions:

(i) What are the relevant features for predicting stock prices in Tanzania?

(ii) How a deep learning model for predicting stock prices in Tanzania can be developed?

(iii) How to deploy and validate the model?

1.6 Significance of the Study

The developed model will serve as an aid to investment advisors when recommending stocks to investors. Currently the exchange consists of 28 listed companies, with more companies expected to join due to ongoing initiatives. As the number of companies is growing, it becomes difficult for the investment advisor to consistently track each company as it can result into expedience and experience biases (Clifford, 2022). However, the model will provide a broadened view of the exchange to the advisors enabling them to devise suitable portfolios to match different clients' needs.

The developed model will increase investors' chances of making profit from subsequent investments. Based on the model, investment advisors will be able to devise suitable portfolios to suit various clients' financial positions. This will enable different types of investors to benefit from the market. Moreover, the portfolios will enable the investors to diversify their equity investments thereby increasing their chances of getting profit as opposed to relying on only one company.

The developed model will enable market intermediaries to push more content into the media in the form of analysis. This is because the model will serve as a convenient utility to financial experts, to enable them to carry out investigations of different market trends and patterns as well as to verify their intuitions and assumptions. Consequently, this will encourage them to create informative and educative content into the media, such as social media, which will eventually promote market awareness and activity.

The study contributes to the body of knowledge on whether accounting for the number of outstanding shares of each company when building a joint model for predicting the stock prices of multiple companies can help to improve prediction accuracy. Additionally, the study investigated whether inclusion of investors' participation attributes among the features can help to improve prediction accuracy. The study laid a foundation for future researches that will employ deep learning to predict stock prices in Tanzania.

1.7 Delineation of the Study

This study focused on predicting the next day's closing price for companies in DSE. The study does not predict the closing price for other time spans such as for the next week, next month or next quarter. Furthermore, the study focused on a class of deep learning techniques known as Recurrent Neural Networks. Specifically, the study developed three models: Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU).

This is a supervised learning study, since the target variable, next day's closing price, is known. It is a regression-based study because it focuses on predicting a continuous numerical value.

CHAPTER TWO

LITERATURE REVIEW

2.1 Financial System Overview

A financial system is an economic arrangement by which financial institutions facilitate the transfer of funds and assets between borrowers, lenders and investors (Raubenheimer, 2019). The aim of a financial system is to lubricate the economy by ensuring efficient distribution of funds from the savers, surplus spending units, to the users, deficit spending units (Carney, 2019). A financial system comprises of financial markets, financial intermediaries, laws, regulations and techniques for smooth exchange of funds or financial resources (Dar es Salaam Stock Exchange PLC (DSE), 2016).

Financial markets refer to any marketplace where the trading of financial securities takes place. Financial markets are usually classified into two categories: money markets and capital markets (Moradi *et al.*, 2016). Money markets offer a short-term lending and borrowing system. A good example of the money market are treasury bills. Through buying a treasury bill, one is lending to the government the specified amount of money for a short period, usually less than one year (Lotto, 2018). On the other hand, capital markets were designed to offer companies and organizations an opportunity to diversify their means of acquiring long-term capital through issuance of bonds or shares (Atenga & Mougoué, 2021). Bonds are debt securities that represent a loan made by an investor to a borrower, which may be a corporate or government. They are long-term fixed-income investments that can have maturity periods of up to fifteen years or more (Raubenheimer, 2019). On the other hand, shares (or stocks) are financial securities that represent ownership of part of a company. Unlike bonds, shares do not have a maturity date as they are not debt instruments. However, investors in shares benefit through dividend payouts or from profits made from selling their shares when they increase in value (Schiereck *et al.*, 2018). Capital markets are composed of two interdependent segments: primary markets and secondary markets. Primary markets are where new issuances of shares or bonds are sold to the public for the first time. Secondary markets are where purchased shares or bonds are traded. An example of a secondary market is a stock exchange such as the Dar es Salaam Stock Exchange (Dar es Salaam Stock Exchange PLC (DSE), 2016). Trading of securities in a secondary market is done through market intermediaries known as brokers which are usually companies (Schiereck *et al.*, 2018).

2.2 The Status of the Dar es Salaam Stock Exchange

The Dar es Salaam Stock Exchange (DSE) is the only stock exchange in Tanzania. It was incorporated in 1996 and began trading operations in 1998 with Tanzania Oxygen Limited (TOL) listing as the first company. Since its establishment, the DSE has been growing gradually, both in terms of size and infrastructure.

In terms of size, in 1998 there were only two (2) listed companies, however, the number grew to 18 companies by 2013, of which 12 were domestic companies and six (6) were foreign companies. Currently, there are 28 listed companies in total, of which 22 are domestic companies. The investor base has also been growing; as at 2021, there were over 600 000 investment accounts at the exchange as opposed to 300 000 investment accounts in 2013 (Marcopolis, 2020).

In terms of infrastructure, in 2013, the exchange migrated to a more efficient Automated Trading System (ATS) and Central Depository System (CDS) in order to ensure smooth conduction of transactions with increasing number of market participants. In 2014, the ATS was deployed on the Wide Area Network (WAN) to allow remote trading by brokers. Also in the same year (2014), the government uplifted the regulation that limited foreign investors to owning only up to 60% of shares of a listed company (Historical Background | Dar Es Salaam Stock Exchange PLC, 2020). This opened the door for more foreign investments, which culminated into rise in share prices due to increase in demand (FinSights Lab, 2020). This can be seen in Fig. 1 whereby there was a surge in the Tanzania Share Index (TSI) in 2014 and onwards compared to the preceding years.

It is also worth noting that, on the 21 January, 2019, DSE was accepted as a full member in the World Federation of Exchanges (WFE), and on the 26 September, 2019, it was reclassified to Frontier Market status from previously being unclassified (*Dar Es Salaam Stock Exchange PLC Achieves Full Membership to the World Federation of Exchanges | Dar Es Salaam Stock Exchange PLC*, 2019; *DSE Achieves Frontier Market Status, Country Classification By FTSE Russell | Dar Es Salaam Stock Exchange PLC*, 2019). This has substantial impact in enhancing investors' confidence in the exchange, as well as profiling and increasing visibility of the exchange to large global fund managers who have interest in frontier markets.

Furthermore, in 2015, the regulatory framework was improved further, to permit the use of mobile phone technology in Initial Public Offerings (IPOs) and secondary trading. To begin with, a USSD application was launched on the 20 August, 2015 followed by an Android application on the 20 November, 2020 (Marcopolis, 2020). Table 1 and Table 2 show steps to register through the USSD application, whereby Table 1 shows the steps to obtain customer identification code or Know Your Customer code (KYC) while Table 2 shows the steps to complete the registration process. On the other hand, the Android application is available on Google Play Store through searching “DSE Mobile Trading Platform”. Both applications are intended to make it more convenient for investors to participate in the market and also to engage more people upcountry where brokers may not be present. This implies deepening of the investor base which can lead to more activity in the market.

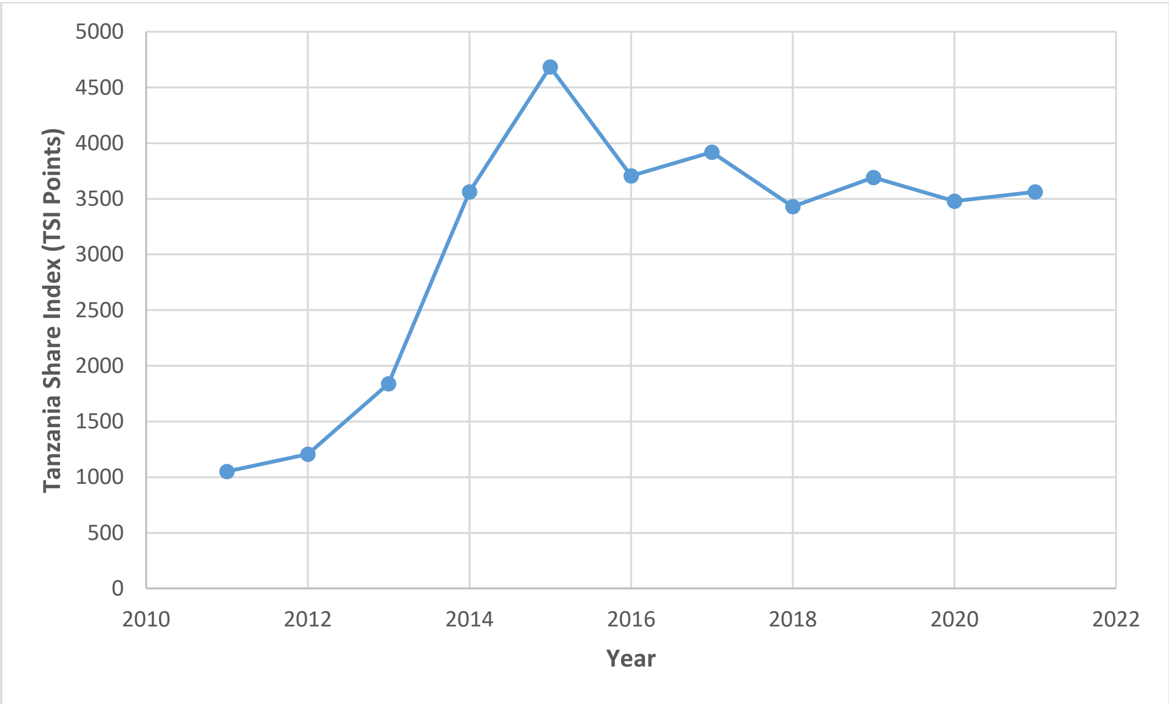

Figure 1: Tanzania share index averages (2011-2021)

Table 1: Initial registration steps on the DSE USSD application

Step	Description
1	Dial *152*00#
2	Choose "Government Payments"
3	Choose "DSE"
4	Choose "Register"
5	Choose "Investor Identification"
6	Choose "NIDA (National Identification)"
7	Enter your National Identification Number
8	Take note of the code received

Table 2: Final registration steps on the DSE USSD application

Step	Description
1	Dial *152*00#
2	Choose "Government Payments"
3	Choose "DSE"
4	Choose "Register"
5	Choose "Open CSD (Central Securities Depository) account"
6	Enter the code noted in the earlier steps
7	Choose a broker
8	Enter bank account number
9	Enter branch where the bank account was opened
10	Confirmation that the request has been received

2.3 Time Series Data and Shortcoming of Traditional Neural Networks

Time series data refers to data that is sequential in nature, changing over time. Examples of time series data include: weather measurements, audio signals, video signals and stock data (Parray *et al.*, 2020). Time series data possess two major characteristics: a trend and a cycle. Trend means a general direction in which something is changing, it could be upward or downward, whereas cycle means the tendency to resume a pattern or trend over time (Obthong *et al.*, 2020). These characteristics of time series data provide an intuition that, in order to best model such data, a model needs to possess the ability to memorize previous information in order to use it in determining present and future outcomes. For example, in order to build a model to classify the type of event happening at every point in a movie, then the model architecture needs to provide for memorizing previous video frames, that the model has seen during training, in order to use that information in understanding the current frame. Traditional neural networks did not provide this support and were thus insufficient at modelling time series data (Olah, 2015; Yu *et al.*, 2019).

Furthermore, traditional neural networks, were limited in that, they could neither consume a sequence as input nor produce a sequence as output. However, this was desirable when it came to modelling sequential data because, with sequential data, either the input or output or both can be sequential (Fig. 2). For example, in machine translation both the input, human voice, and output, another voice or text, are sequential. In sentiment analysis, the input can be a sequence while the output is a category. In music generation, a single tone (input) generates a piece of music (sequential output) (Yanhui, 2021; Yu *et al.*, 2019).

Moreover, it is noteworthy that in contemporary stock market predictions, the input is provided in the form of sequences, while the output, such as the closing price, is a non-sequence (Brownlee, 2018; Nayak, 2020). This is because stock data are time series in nature and therefore providing the input in the form of sequences helps the model to grasp the context (trend) and thereby predict more accurately (Nayak, 2020).

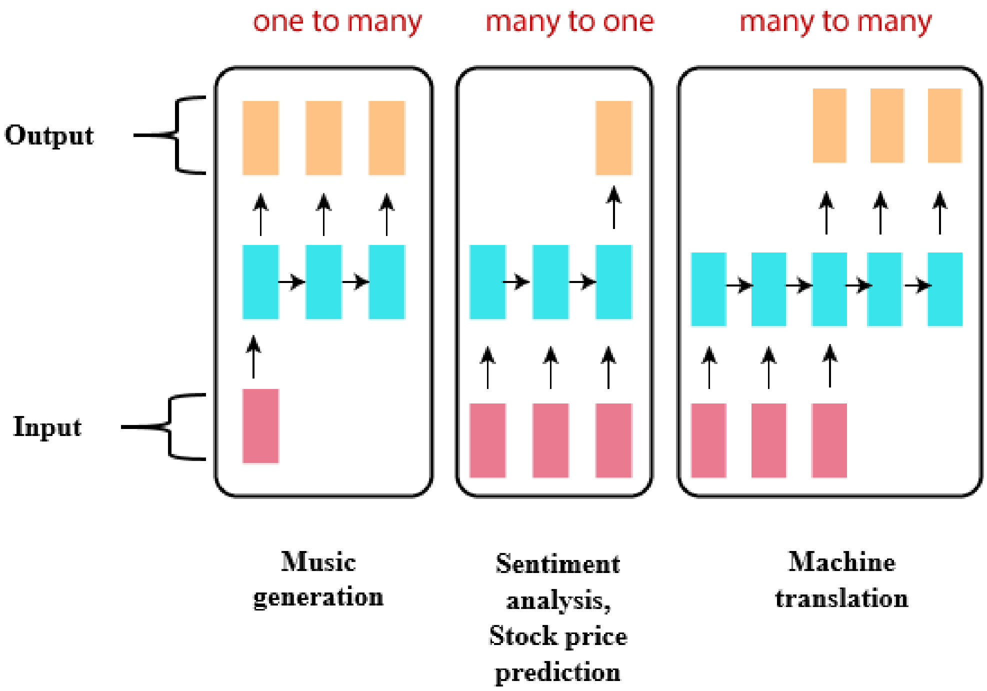

Figure 2: Illustration of variable-size input and output (Yanhui, 2021)

2.4 Overview of Recurrent Neural Networks

Recurrent Neural Networks (RNNs), as the name infers, are neural networks with loops to persist information obtained from previous observations, in order to use it in determining present or future outcomes. That is to say, the output from an RNN unit is informed by both the current input and previous inputs (Yu *et al.*, 2019). The structure of RNNs is shown in Fig. 3.

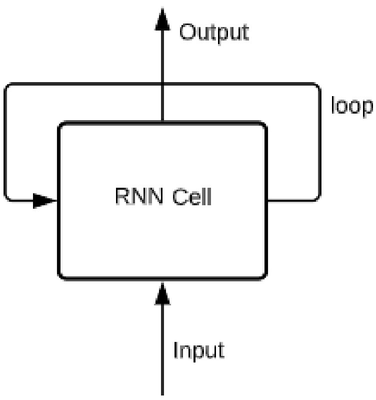

Figure 3: Recurrent Neural Network architecture (Olah, 2015)

Unlike traditional neural networks, RNNs can support variable-size input and output, making them suitable for handling sequential data (Lipton *et al.*, 2015). However, it was discovered that RNNs are not reliable at memorizing information for a long time as they suffer from vanishing and exploding gradients as the size of the network grows (Pascanu *et al.*, 2013).

Vanishing and exploding gradients are challenges that occur during the training of deep neural networks and they hinder effective learning (Pykes, 2020). These challenges arise during back propagation, whereby, the gradients tend to undergo continuous matrix multiplication because of the chain rule (Alese, 2018). As a result, if the gradients are small, they decrease exponentially until they eventually vanish, this is known as the vanishing gradient problem. On the other hand, if the gradients are large, they increase exponentially until they blow up (overflow), this is known as the exploding gradient problem (Li, 2022). Exploding gradients cause large changes in model weights which make the network unstable whereas vanishing gradients result in very small changes in model weights which slow down the learning process (Equation 1).

$$W_{new} = W - \alpha \frac{\partial Loss}{\partial W} \tag{1}$$

Where:

W_{new} is the new weight matrix

W is the old weight matrix

α is the learning rate

$\frac{\partial Loss}{\partial W}$ is the derivative of the loss function with respect to the weight. It is obtained as a product of derivatives through back propagation using chain rule. It could be vanishing or exploding.

One of the solutions to these challenges is to use gated recurrent neural network architectures such as Long Short-Term Memory (Chung, 2016).

2.5 Overview of LSTM, Bi-LSTM and GRU

Long Short-Term Memory networks (LSTMs) are special recurrent neural networks that can memorize previous information for a long time (Yu *et al.*, 2019). They store information based on the current input (short-term), and also based on previous inputs (long-term) (Staudemeyer & Morris, 2019). They are composed of three gates: the forget, input and output gates that decide which information to forget, which new information to input and which information to output respectively (Hu *et al.*, 2021). Figure 4 illustrates the overview of an LSTM.

Gates provide a way to optionally let information through. They utilize a sigmoid function which outputs values between zero and one, representing how much of the previous information should be let through. A value of zero means "totally discard this information" whereas a value of one means "totally keep this information" (Olah, 2015).

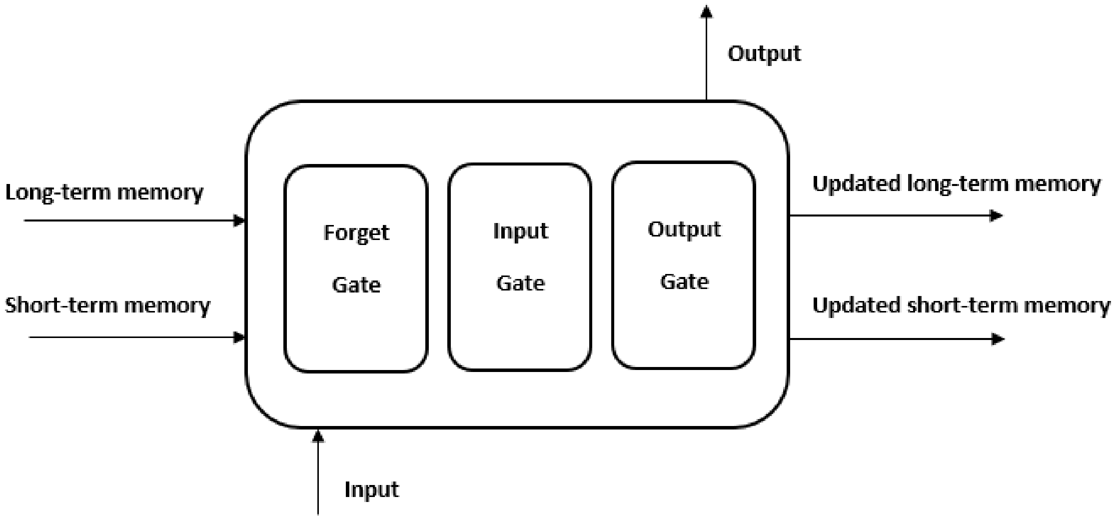

Figure 4: Overview of LSTM (Saxena, 2021)

Long Short-Term Memory networks do not suffer from vanishing and exploding gradients, this is because both its long-term and short-term states are maintained by a gating function (sigmoid function) which can allow the previous states to pass through unchanged without increasing or decreasing exponentially at each layer (Nandakumar, 2018).

Several variants to the original LSTM were proposed over the years, such as the Bidirectional LSTM in 2005 (Graves & Schmidhuber, 2005) and Gated Recurrent Unit (GRU) in 2014 (Cho *et al.*, 2014). The normal LSTM model is unidirectional and therefore it learns the entire input sequence, as provided to it, in only one direction. However, the bidirectional LSTM learns the entire input sequence in both directions, forward and backward, and merges the results from both directions using either summation, multiplication, averaging or concatenation (Zvornicanin, 2022). Figure 5 illustrates the working of a bidirectional LSTM.

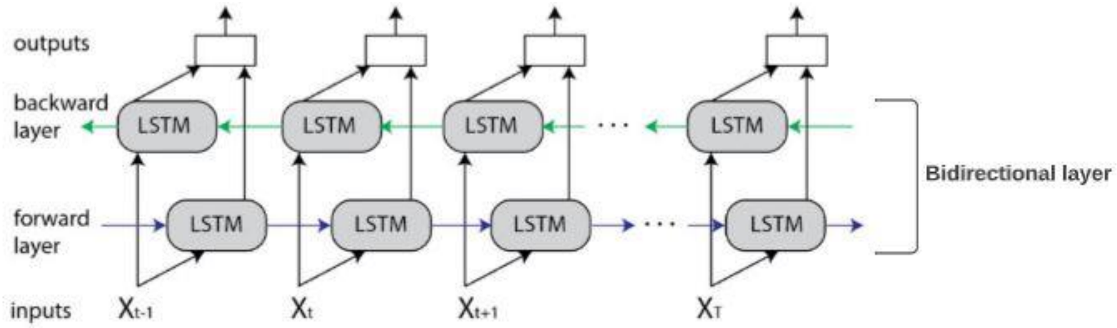

Figure 5: The working of a bidirectional LSTM (Zvornicanin, 2022)

On the other hand, the GRU is an LSTM variant with two gates: update and reset gates. The update gate controls the information that flows into memory while the reset gate controls the information that flows out of memory. Instead of having long-term and short-term state, the GRU merges both states to form the hidden state. The resulting architecture is simpler and faster than the original LSTM, nevertheless, the GRU has been proven effective at learning long-term dependencies in time series data (Chung *et al.*, 2014). Figure 6 illustrates the overview of a GRU.

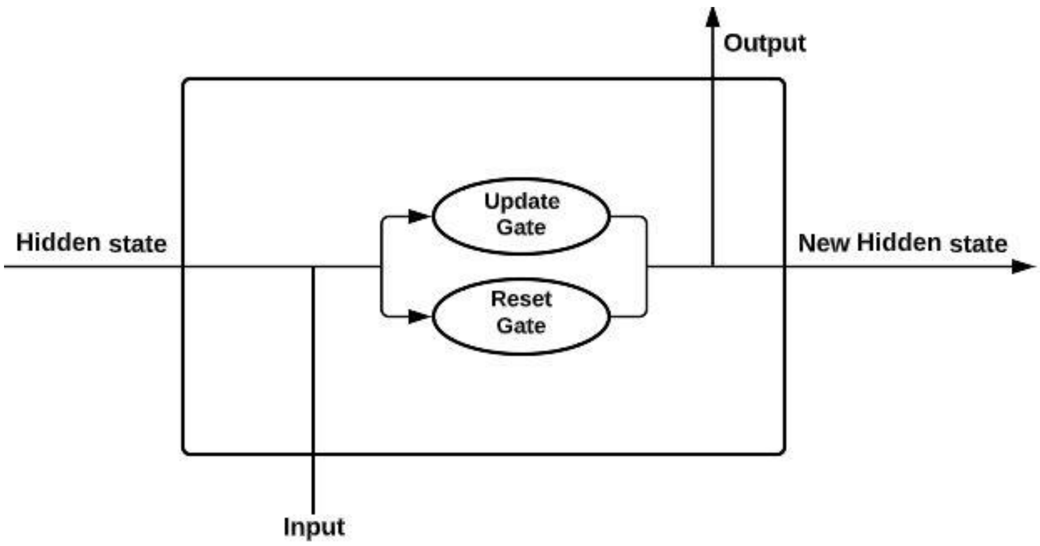

Figure 6: Overview of a GRU (Loye, 2019)

2.6 Related Works

Manishimwe *et al.* (2021) proposed an integrated mobile application based on machine learning model for predicting stock prices in East African markets. Long Short-Term Memory deep learning architecture was used. The data used for the study were collected through an API from Nasdaq stock exchange, which is a stock exchange from USA. The data included the volume, high, low, opening and closing prices of Microsoft, Apple, Facebook, Amazon and Alibaba. The study did not report whether the resulting model was eventually tested on actual data from East African markets. Moreover, the study did not consider investors' participation among the predictive attributes.

Alkhatib *et al.* (2022) developed GRU, LSTM, Bi-LSTM, Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN) and a hybrid of CNN and LSTM models to predict next day's adjusted closing price of Apple, ExxonMobil, Tesla and Snapchat. Separate models were developed for each company. The predictive features used were: high price, low price, opening price, volume, difference of high and low price and difference of opening and closing price. The results showed that all models had comparative results, that is, no model showed better results, than other models, or continuously outperformed other models. However, the study did not explore using investors' participation attributes.

Lakshminarayanan and McCrae (2019) developed LSTM and Support Vector Regression (SVR) models to predict the daily closing price of companies listed in the Dow Jones index. The Dow Jones index was considered because it covered a wide range of companies from a variety of sectors. The predictive features used were the previous closing price, oil price and gold price. The LSTM model outperformed the SVR model, however, the study did not include investors' participation among the features.

Samarawickrama and Fernando (2017) developed a model to predict the daily closing price of selected listed companies of Colombo Stock Exchange. Feedforward, Simple Recurrent Neural Network (SRNN), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks were employed for building the models. Previous closing, high and low prices were used as predictive inputs. The results showed that SRNN and LSTM networks produced lower errors compared to feedforward networks. However, the study focused only on historical prices and did not include other features such as investors' participation.

Salimath *et al.* (2021) performed a comparative study of LSTM, GRU and a hybrid of LSTM and GRU for predicting the daily closing price of 25 companies (from different sectors) listed on the National Stock Exchange of India. The predictive features used were: previous closing price, average of (opening, low and high price), trading volume, 10-day simple moving average (SMA), 10-day weighted moving average (WMA), relative strength index (RSI), stochastic K% and stochastic D%. The results showed that the GRU model outperformed LSTM and the hybrid of LSTM and GRU. The study did not account for the differences in number of outstanding shares of the companies.

Sen *et al.* (2021) chose eight prominent sectors from the National Stock Exchange of India (NSE), and subsequently selected five active companies from each sector based on the NSE market report. The study developed LSTM models for each sector using the selected companies as datasets. Previous closing prices were used as predictive inputs, and the goal was to predict next day's closing price. The actual returns yielded by the optimum risk portfolios were computed and were compared with the LSTM predicted returns. The results showed that the actual returns yielded by the optimum risk portfolios closely matched those predicted by the LSTM model. However, the study used only one predictive feature, previous closing prices.

Sirignano and Cont (2019) employed LSTM on a high frequency dataset containing billions of transactions of US equities. The study found that prediction accuracy could neither be improved through building company-specific models nor through building sector-specific models, rather, including price and order flow history over many past observations improved prediction accuracy. This indicated that the patterns learned by the model are universal and not company or sector-specific. However, the study did not consider the differences in number of outstanding shares among the companies.

2.7 Research Gap

From the review of related works, it was discovered that most of the studies have not explored the aspect of investors' participation on stock price prediction. In frontier markets such as the DSE, foreign investors play a huge role in triggering price volatility (Marcopolis, 2020). However, the participation of foreign investors is influenced by the behaviour of domestic institutional investors (Jain & Singh, 2022). This study therefore explored whether inclusion of investors' participation variables, namely, turnover from shares bought by foreign investors and turnover from shares bought by domestic investors, among the features, can help to improve stock price prediction.

Furthermore, most studies that built a single model to predict the stock prices of multiple companies, did not consider the differences in number of outstanding shares of the companies. However, in a multi-company perspective, a better way to compare and understand the variations in demand and supply for a company is to weigh the companies' demand and supply variables with respect to corresponding number of outstanding shares. For instance; a company

A has 500 outstanding shares (total number of issued shares), where as a company B has 1000 outstanding shares. If on a particular trading day, both have a value of 10 as volume (number of shares traded), then a better way to understand the significance of that uptake as it relates to each company is to divide the value with the corresponding company's number of outstanding shares. Therefore, this study examined whether accounting for the number of outstanding shares of each company when building a single model to predict stock prices of multiple companies can help to improve stock price prediction.

Additionally, it was discovered that there is a shortage of literature related to exploitation of machine learning and deep learning to predict stock prices in Tanzania. Therefore, this study was conducted to fill this gap.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Study Area

This study was conducted in Dar es Salaam region in Tanzania. The selection of this area was based on the fact that the Dar es Salaam Stock Exchange (DSE) is located in Dar es Salaam region. Furthermore, most of the stock brokerage companies are located in Dar es Salaam.

3.2 Programming Language and Libraries

The study employed Python version 3.7 as the programming language. It was chosen because it offers a vast set of open-source libraries to support deep learning. Table 3 shows the libraries used in this study and the reasons for using them.

Table 3: Libraries used in this study and the reasons for using them

Library	Reason
Keras	To access classes and functions for developing deep learning models
Tensorflow	To provide low-level binaries for executing keras models
Keras-tuner	To tune keras models
Pandas	To access functions for data analysis and manipulation
Numpy	To access functions for manipulating multidimensional arrays
Scikit-learn	To access functions for data preprocessing and model evaluation
Matplotlib	To access functions for creating graphs

3.3 Model Development Environment

All models were developed on Google Colab using Keras with a Tensorflow backend. Google colab is a hosted jupyter notebook service that enables writing and execution of arbitrary python code through the browser (Singh, 2022). It was chosen as the development environment because it ensures fast training of deep learning models by leveraging faster hardware provided by Google such as Graphics Processing Units (GPUs), through the cloud. The connection to Google Colab was made from a computer preinstalled with Ubuntu 20.04 and equipped with a 3.6GHz Intel Core i7 CPU, 512GB SSD storage, and 16GB RAM.

3.4 Prototype Development Tools

The web-based prototype consisted of both front-end and backend. The front-end was developed using Hypertext Markup Language (HTML) and styled using custom Cascading Style Sheets (CSS) and those from Bootstrap. Additionally, Javascript and Datatable plugin were employed to add interactivity to the webpages and tables, respectively. On the other hand, the backend was developed using python, whereby, Django python framework was employed in order to ensure fast development while aligning with security standards. The database engine used was MySQL.

3.5 Research Framework

Figure 7 shows the research framework followed in this study. A dataset consisting of historical stock transactions was acquired and preprocessed. Thereafter, active companies were chosen, and the resulting dataset was transformed into a form suitable for supervised learning. Subsequently, LSTM, Bidirectional LSTM and GRU were trained on the dataset. The best model was embedded in a web-based prototype for predicting stock prices.

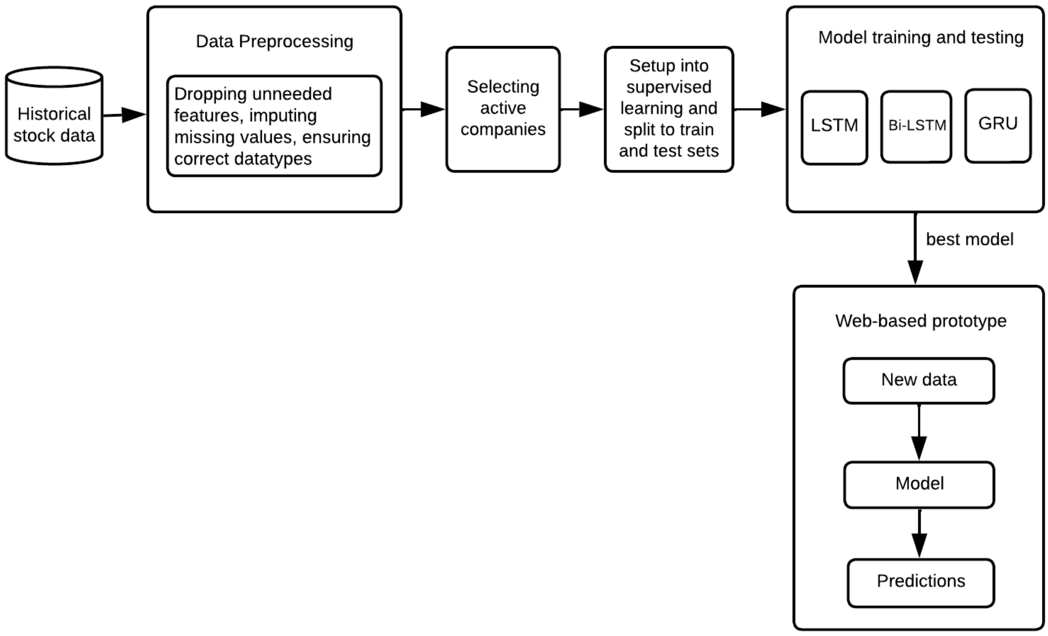

Figure 7: Research framework

3.6 Study Design

Cross-sectional research design was adopted to identify relevant features for predicting stock prices in Tanzania, through Focus Group Discussions (FGDs) whereby each FGD was conducted once and for all. The design allows collection of information at a single point in time, while allowing one to estimate the prevalence of outcome of interest (relevant features) as samples are always taken from the whole population (Kothari, 2004). Additionally, cross-sectional design is cost-effective, and takes little time while assuring appropriate quality of information.

3.7 Sampling Procedure

Purposive sampling was used to select participants of focus group discussions. The study identified stockbrokers and investment advisors from stock brokerage companies to be suitable participants for the study, due to their knowledge and experience of investments and investors. In total, the study conducted four FGDs at four different brokerage companies. The criteria considered for selecting the companies were: presence of a research department, having up-to-date website, and being recently awarded by DSE was considered to be the added advantage. In the case of similar ranking, random selection was employed in order to finally pick all the four companies. Table 4 shows the number and composition of participants from each company whereby company names have been anonymized as A, B, C and D.

Table 4: Participants composition in conducted FGDs

Company Name	Participants	Stockbrokers	Investment advisors
Company A	4	2	2
Company B	4	3	1
Company C	5	3	2
Company D	4	2	2
TOTAL	17	10	7

3.8 Data Collection

Primary data were collected through FGDs with the aim of identifying relevant features for predicting stock prices in Tanzania. On the other hand, secondary data consisting of the relevant features were acquired from DSE.

3.8.1 Primary Data

The FGD guide involving topics of discussion is attached as Appendix 1. To ensure equal involvement, each participant was given a chance to address a topic without interruption, followed by a discussion and a short debate. Similar responses were grouped, coded and analyzed using SPSS statistical software.

3.8.2 Secondary Data

The secondary data consisting of the identified features were obtained from DSE covering the period from 02 January, 2018 to 08 April, 2022. The original dataset before preprocessing consisted of 29 391 samples and 16 features (Fig. 8). Table 5 shows the features and their descriptions.

	Company Name	Date	Opening Price	Closing Price	High Price	Low Price	Deals	Outstanding Bids	Outstanding Offers	Volume	Outstanding Bids/Outstanding Shares	Outstanding Offers/Outstanding Shares	Volume/Outs
0	DSE	2016-07-12	500.0	935.0	1000.0	705.0	201.0	622986.0	4000.0	849121.0	0.026149	0.000168	(
1	DSE	2016-07-13	935.0	1000.0	1010.0	1000.0	289.0	777186.0	70027.0	1129302.0	0.032622	0.002939	(
2	DSE	2016-07-14	1000.0	1000.0	1010.0	1000.0	139.0	901396.0	159348.0	491423.0	0.037836	0.006689	(
3	DSE	2016-07-15	1000.0	1020.0	1030.0	1000.0	116.0	176140.0	26928.0	862768.0	0.007393	0.001130	(
4	DSE	2016-07-18	1020.0	1040.0	1060.0	1000.0	70.0	1170096.0	23110.0	264928.0	0.049114	0.000970	
...	...	...	...	...	...	...	...	...	...	...	...	...	
29386	DCB	2022-04-08	195.0	195.0	195.0	195.0	2.0	585.0	100723.0	607.0	0.000009	0.001485	(
29387	CRDB	2022-04-08	380.0	380.0	375.0	370.0	29.0	3640.0	400003.0	51121.0	0.000001	0.000153	(
29388	VODA	2022-04-08	770.0	770.0	740.0	740.0	1.0	0.0	1307390.0	200.0	0.000000	0.000584	(
29389	NICO	2022-04-08	580.0	610.0	620.0	560.0	9.0	10.0	18214.0	46397.0	0.000000	0.000263	(
29390	YETU	2022-04-08	510.0	510.0	0.0	0.0	0.0	0.0	4790.0	0.0	0.000000	0.000395	(

29391 rows × 16 columns

Figure 8: Dataset before preprocessing

Table 5: Description of the features in the dataset

Feature	Description
Opening price	The price of a stock at market open, on a given trading day
Low price	The lowest price at which the stock was sold on a given trading day
High price	The highest price at which the stock was sold on a given trading day
Closing price	The price of a stock at market close, on a given trading day
Deals	Number of transactions on a particular stock, on a given trading day
Volume	Number of shares, of a particular stock, traded on a given trading day
Outstanding bids	Number of shares, of a particular stock, ordered but yet unbought at the end of a particular trading day
Outstanding offers	Number of shares, of a particular stock, offered for sale but yet unsold at the end of a particular trading day
Volume/Outstanding shares	The ratio of volume to total issued shares of a company, on a given trading day
Outstanding bids/Outstanding shares	The ratio of outstanding bids to total issued shares of a company, on a given trading day
Outstanding offers/Outstanding shares	The ratio of outstanding offers to total issued shares of a company, on a given trading day
Price change	The difference between closing price and opening price of a particular stock on a given trading day
Turnover from shares bought by foreign investors	Amount made due to shares purchased by foreign investors on a given trading day
Turnover from shares bought by local investors	Amount made due to shares purchased by domestic investors on a given trading day

The other features were *date* and *company name* which represented the date of the given trading day and the name of a particular listed company respectively. However, these features were dropped as they were not required for developing the model.

3.9 Data Preprocessing

The study dropped all the features that were not required to build the model. Specifically, the *Company Name* and *Date* features were dropped in order to maintain anonymity. Missing values in each feature were imputed by taking the mean of all samples that had the same closing price as the one where the missing value occurs. Successively, all features were converted to the correct datatype. Since all the features were numeric, they were converted into float datatype. Float datatype was chosen because it was the datatype that could accommodate all numeric datatypes in the dataset. Furthermore, the study scaled all the features to put them on a similar scale, between zero and one, by utilizing *MinMaxScaler ( ).*

3.10 Selecting Companies with Active Shares

According to DSE regulations, on a given trading day, a company's shares are considered to have traded significantly if the number of shares traded is 0.005% of its total issued shares (*Dar Es Salaam Stock Exchange PLC Rules*, 2022). Therefore, in order to identify companies with active shares, the study took the average of *Volume/Outstanding shares* of each company over the last one year. The last one year was considered in order to ensure that only companies that have been active as at recent are selected, as opposed to averaging over all four years which could have led to selecting a company that has not been recently active. According to the dataset used in the study, the last one year, represented 8 April, 2021 to 8 April, 2022. Companies whose average of *Volume/Outstanding shares* was greater or equal to 0.005% were selected.

Additionally, data visualizations were performed on the closing price attribute of each of the selected companies. Companies whose closing price was static for over one year were dropped. This is because a company with active shares is expected to have intervals of positive and negative price movements over time (Active Stocks, 2022).

3.11 Feature Relevance

The study employed the *f_regression ( )* function from scikit-learn's feature selection library to ascertain the relevance of each feature with respect to the target variable. The function returns a positive correlation statistic whereby the higher the value, the more relevant the feature (Brownlee, 2020). Since there was more than one company, feature scores were obtained for each company in turn, and then the average score of each feature across all the selected companies was calculated.

3.12 Supervised Learning Setup

The resulting dataset consisting of companies with active shares was transformed into input and output components for supervised learning. In contemporary stock market predictions, the input is provided in form of sequences in order to capture the trend (context-specific information) in the data and hence improve prediction accuracy (Nandakumar, 2018; Nayak, 2020). This is possible through leveraging recurrent neural network architectures such as LSTM, Bi-LSTM and GRU that support handling of sequential input and memorizing of context-specific information for a long time (Lipton *et al.*, 2015).

The data was transformed into three dimensions, *(samples, timesteps, features),* this is because the models used in this work required data to conform to that format (Brownlee, 2018). Table 6 provides the description of each dimension.

Table 6: Description of the dimensions

Dimension	Description
Samples	Total number of input sequences
Timesteps	Length of each input sequence. In stock market predictions it represents the number of preceding days the model should refer when predicting the next day closing price
Features	Number of attributes per input sequence

Specifically, the data was split into finite length input sequences and each input sequence was mapped to a corresponding output (closing price). Figure 9 provides an illustration of the supervised learning setup. For simplicity, a few features have been shown and the value of timesteps has been taken as three, meaning that each input sequence contained three rows. Only a few input sequences have been shown.

	opening_price	high_price	low_price	volume	closing_price	
0	500	1000	705	849121	935	
1	935	1010	1000	1129302	1000	1st input sequence
2	1000	1010	1000	491423	1000	
3	1000	1030	1000	862768	1020	Output
4	1020	1060	1000	264928	1040	
5	1040	1070	1020	264646	1060	
6	1060	1070	1060	89791	1060	

	opening_price	high_price	low_price	volume	closing_price	
0	500	1000	705	849121	935	
1	935	1010	1000	1129302	1000	
2	1000	1010	1000	491423	1000	2nd input sequence
3	1000	1030	1000	862768	1020	
4	1020	1060	1000	264928	1040	Output
5	1040	1070	1020	264646	1060	
6	1060	1070	1060	89791	1060	

	opening_price	high_price	low_price	volume	closing_price	
0	500	1000	705	849121	935	
1	935	1010	1000	1129302	1000	
2	1000	1010	1000	491423	1000	
3	1000	1030	1000	862768	1020	3rd input sequence
4	1020	1060	1000	264928	1040	
5	1040	1070	1020	264646	1060	Output
6	1060	1070	1060	89791	1060	

Figure 9: Illustration of supervised learning setup

In order to achieve this kind of supervised learning setup. Firstly, the data were sorted in ascending order, which is a conventional way of sorting time-series data before training (Dingli & Fournier, 2017b). This was achieved as follows:

data = data.sort_values(by=['Company Name', 'Date'], ascending = True);

Thereafter, a python script was written to actualize the setup (Fig. 10). In the end, the timesteps variable was considered as one of the hyperparameters that was tuned, in order to find the value that yielded best results. All the data were converted to numpy array in order to leverage numpy's parallelism features for faster execution during training. Eighty percent (80%) of the samples were used to train the model whereas the remaining 20% were used for testing.

```python
# Supervised learning setup
import math
timesteps = 30
n_features = data.drop('Company Name', axis='columns').shape[1]
X_train = np.empty(shape=(0, timesteps, n_features))
X_test = np.empty(shape=(0, timesteps, n_features))
y_train, y_test = [], []
target_index = data.drop('Company Name', axis='columns').columns.get_loc('Closing Price')
for company_name in selected_companies:
  company_data = data.loc[data['Company Name'] == company_name]
  company_data = company_data.drop(['Company Name'], axis='columns')
  # total number of time-series samples would be length minus timesteps
  dim_0 = company_data.shape[0] - timesteps
  x = np.zeros((dim_0, timesteps, n_features))
  y = np.zeros((dim_0, 1))
  company_data = scaler.transform(company_data.to_numpy())
  for i in range(dim_0):
      x[i] = company_data[i : timesteps + i]
      y[i] = company_data[timesteps + i, target_index]
  train_len = math.ceil(0.8 * len(x))
  train_x, train_y = x[0:train_len, :], y[0:train_len, 0]
  # trim to size divisible by batch-size
  train_x, train_y = trim_dataset(train_x), trim_dataset(train_y)
  y_train = np.append(y_train, train_y)
  X_train = np.append(X_train, train_x, axis=0)
  test_x, test_y = x[train_len: , :], y[train_len:, 0]
  test_x, test_y = trim_dataset(test_x), trim_dataset(test_y)
  y_test = np.append(y_test, test_y)
  X_test = np.append(X_test, test_x, axis=0)
```

Figure 10: Python script used to achieve supervised learning setup

3.13 Construction of the Models

Three different models were developed, namely: Long Short-Term Memory (LSTM), Bidirectional LSTM and Gated Recurrent Unit (GRU). Table 7 shows the hyperparameters that were tuned in the construction of each model.

Table 7: Hyperparameters tuned

Hyperparameter	Description
Hidden layer size	The number of hidden layers
Number of hidden units	The number of neurons per hidden layer
Batch size	Number of samples to be processed by a model before updating the weights
Number of epochs	Total number of iterations through the entire training set
Timesteps	The number of preceding days a model should refer before predicting the next day closing price

In each model, the output layer was a dense layer with a linear activation function and it consisted of one neuron, as it produced only a single output (closing price). On the other hand, hyperbolic tangent function (tanh) was used as the activation function in the hidden layers, while the hidden layer size and number of hidden units were tuned as hyperparameters. Adam was used as the optimization algorithm and mean squared error was the loss function. Averaging was used to combine the results of forward and backward pass in Bidirectional LSTM.

Figure 11 shows how hyperparameter tuning was approached, whereby the bayesian optimization function of keras tuner was utilized. The function samples a few possible combinations of hyperparameters randomly and then based on their performance it chooses the next best possible combination of hyperparameters. The process is continued until the optimal values are found or until the maximum number of allowed trials (*max_trials*) have been exhausted (Wadekar, 2021). Early stopping was also used in order to stop training when the validation error reached a minimum (Chen, 2020).

```python
import keras_tuner
class MyOptimalModel(keras_tuner.HyperModel):
  def build(self, hp):
    model = Sequential()
    # Now checking with two hidden LSTM layers
    hp_units1 = hp.Int('units1', min_value=20, max_value=300, step=30)
    hp_units2 = hp.Int('units2', min_value=20, max_value=300, step=30)
    model.add(LSTM(units=hp_units1, input_shape=(X_train.shape[1], X_train.shape[2]),
              activation='tanh', return_sequences=True))
    model.add(LSTM(units=hp_units2, activation='tanh', return_sequences=False))
    model.add(Dense(1))
    model.compile(optimizer="adam", loss="mse")
    return model
  def fit(self, hp, model, *args, **kwargs):
    return model.fit(*args, batch_size=hp.Choice("batch_size", [16, 32, 64, 128]), **kwargs)
tuner = keras_tuner.BayesianOptimization(MyOptimalModel(), objective="val_loss", max_trials=100)
tuner.search(X_train, y_train, epochs=500, validation_data=(X_test, y_test),
             callbacks=[EarlyStopping(monitor='val_loss', patience=5)],
             verbose=0, shuffle=False)
# Print best hyperparameters
for h_param in [f"units{i}" for i in range(1,3)] + ["batch_size"]:
  print(h_param, tuner.get_best_hyperparameters()[0].get(h_param))
```

Figure 11: Hyperparameter tuning

At the end, the timesteps were tuned by substituting each of the following values in turn: 10, 20, 30, 40 and 50. Appendix 3 shows part of the code used in GRU model development.

3.14 Experimental Setup

Five (5) experiments were conducted with each of the models, as follows:

(i) Experiment 1

In this experiment, the models were developed without taking into account the number of outstanding shares of each company nor investors' participation. A total of nine (9) features were used, which were: opening price, low price, high price, previous closing price, price change, deals, volume, outstanding bids, and outstanding offers. Also, in this experiment, the hyperparameters were tuned for each model, and the obtained optimal values were sustained in experiment 2 and in experiment 3.

(ii) Experiment 2

In this experiment, the models were developed while taking into account the number of outstanding shares of each company, however, investors' participation was not considered. In order to account for the number of outstanding shares, the volume, outstanding bids and outstanding offers of each company were divided with the corresponding number of outstanding shares. Precisely, similar features were used as in the first experiment with the exception that volume, outstanding bids and outstanding offers were replaced by *volume/outstanding shares, outstanding bids/outstanding shares* and *outstanding offers/outstanding shares* respectively. Therefore, a total of nine (9) features were used, which were: opening price, low price, high price, previous closing price, price change, deals, volume/outstanding shares, outstanding bids/outstanding shares, and outstanding offers/outstanding shares.

(iii) Experiment 3

In this experiment, the models were developed while taking into account both the number of outstanding shares of each company and investors' participation. In order to account for investors' participation, two additional features were included: turnover from shares bought by foreign investors and turnover from shares bought by local investors. Hence, a total of 11 features were used, which were: opening price, low price, high price, previous closing price, price change, deals, volume/outstanding shares, outstanding bids/outstanding shares, outstanding offers/outstanding shares, turnover from shares bought by foreign investors and turnover from shares bought by local investors.

(iv) Experiment 4

In this experiment, only historical stock prices were used as the features. Hence, a total of four (4) features were used, which were: previous closing price, opening price, low price, and high price. Hyperparameter tuning was re-conducted in order to find optimal values for the current feature set.

(v) Experiment 5

In this experiment, only the previous closing price was used as the feature. Hyperparameter tuning was re-performed in order to find the optimal values with the current feature. This experiment is known as "univariate time-series forecasting" since it used only one input feature (Brownlee, 2018).

3.15 Evaluation of the Models

The following regression evaluation metrics were employed in order to evaluate the performance of the models:

(i) Mean Squared Error (MSE)

This calculates the average squared difference between the predicted values and the observed values (Equation 2). It represents the variance, how widely spread the predicted values are from the observed values.

(ii) Root Mean Squared Error (RMSE)

This is the square root of the mean squared error (Equation 3). It represents the standard deviation of the predicted values from the corresponding actual values.

(iii) Mean Absolute Error (MAE)

This measures the average magnitude of error in a set of predictions without considering their direction. It is calculated as the mean of the absolute values of the differences between observed values and corresponding actual values (Equation 4).

(iv) Mean Absolute Percentage Error (MAPE)

This is the average relative error expressed as a percentage (Equation 5). Since it measures the relative error, it is not changed by a global scaling of the target variable.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{4}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \times 100\% \tag{5}$$

Where: Y_i is the ith observed value, $\hat{Y}_i$ is the corresponding predicted value and n is the number of observations.

3.16 Prototype Development

Extreme programming method was utilized. This agile technique was chosen because it ensures fast delivery of a software, with minimal cost, while emphasizing stakeholder involvement (Valacich, 2017). Figure 12 shows the lifecycle of the extreme programming methodology.

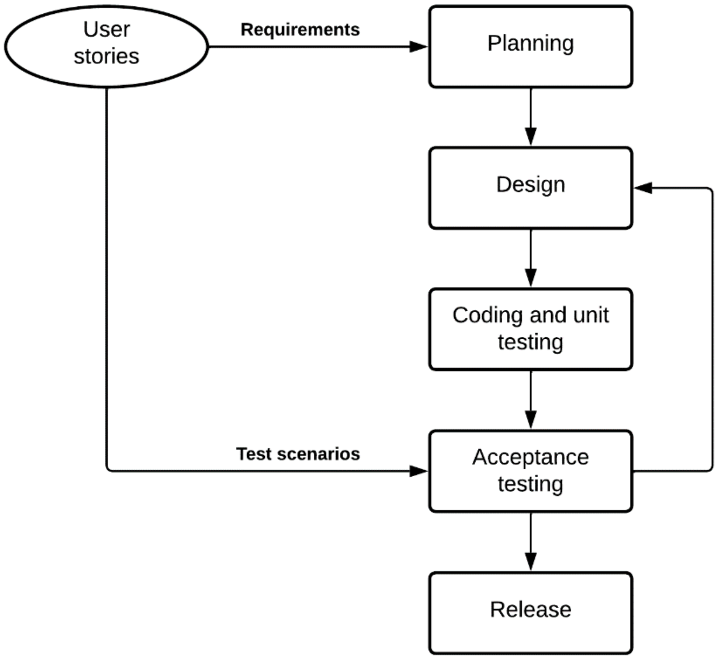

Figure 12: Extreme programming lifecycle (Valacich, 2017)

The five (5) phases of the extreme programming approach included:

(i) Planning

This is the first stage where stakeholders explained their requirements in the form of user stories to describe the desired product. A use case diagram was drawn in order to facilitate communication with stakeholders and thereby effectively capture the requirements of different actors of the system. Figure 13 shows the use case diagram of the system, whereby two actors were identified: a stock personnel (a broker or an investment advisor) and an administrator who would have higher privileges.

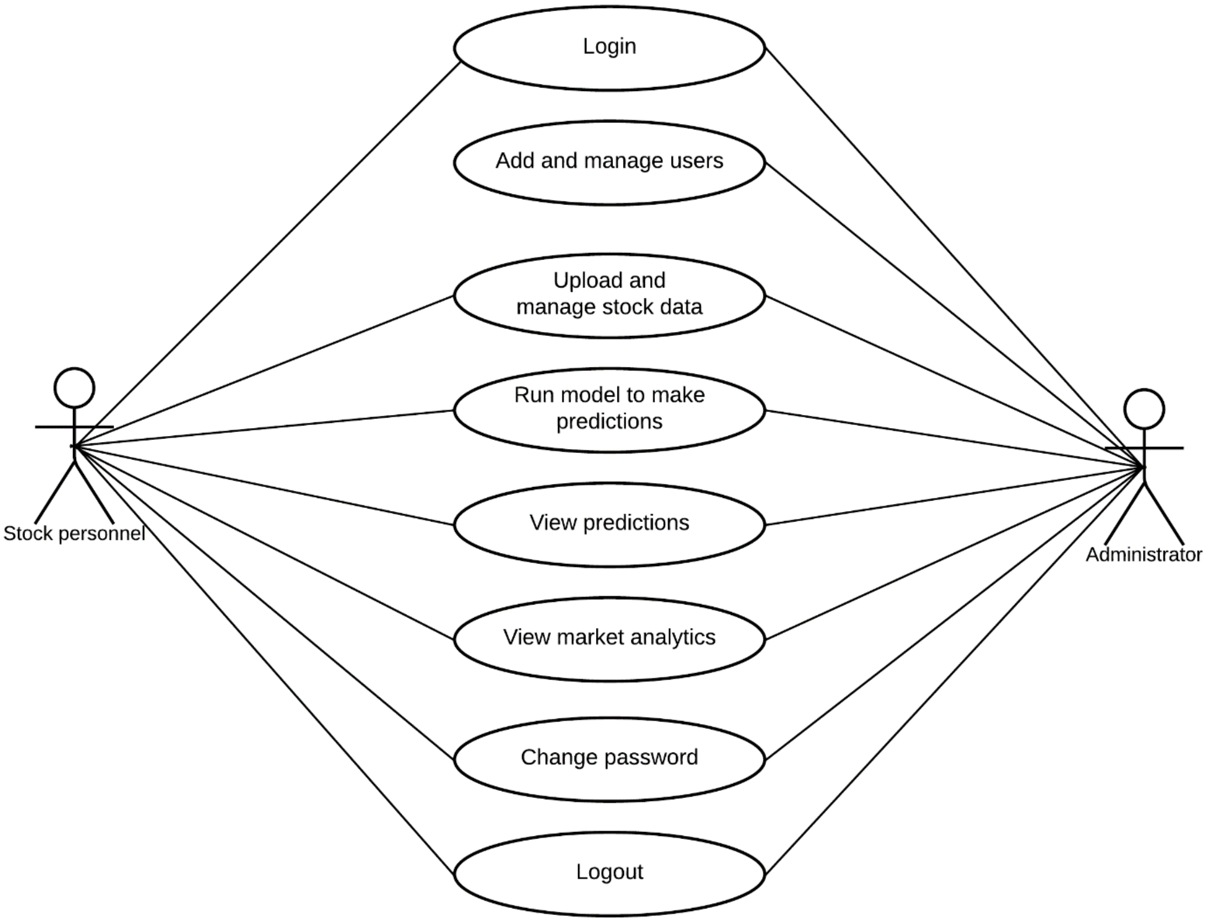

Figure 13: Use case diagram of the system

Thereafter, the requirements (functionalities) were ranked into very important, important and medium important based on stakeholder perspectives and workflow order (Table 8).

Table 8: Functionalities ranked in collaboration with stakeholders

Functionality	Rank	Reason
Upload and manage stock data	Very important	Data is required in order to make predictions
Run model to make predictions	Very important	The system is focused on stock price prediction; therefore, this is the core function
View predictions	Very important	Predictions have to be seen in order to make investment decisions
View market analytics (this was concerned with visualizations of stock price and volume over time)	Important	This function can only be implemented after completing the first three functions
Login	Medium important	This function can be deferred until all essential functions have been implemented
Change password	Medium important	The function can be deferred until all essential functions have been implemented
Logout	Medium important	The function can be deferred until all essential functions have been implemented
Add and manage users	Medium important	The function can be deferred until all essential functions have been implemented

Subsequently, the functionalities were grouped into iterations whereby those identified as very important formed the first iteration, those identified as important formed the second iteration while those identified as medium important formed the third iteration (Table 9). The iterations were implemented successively beginning with the first iteration then the second and then the third.

Table 9: Functionalities grouped into iterations

Iteration	Functionalities
Iteration 1 (First iteration)	❖ Upload and manage stock data ❖ Run model to make predictions ❖ View predictions
Iteration 2 (Second iteration)	❖ View market analytics
Iteration 3 (Third iteration)	❖ Login ❖ Logout ❖ Change password ❖ Add and manage users

(ii) Designing

This is where the system was designed in terms of physical design (user interfaces) and logical design (flowcharts). Figure 14 shows a flowchart of the make predictions use case.

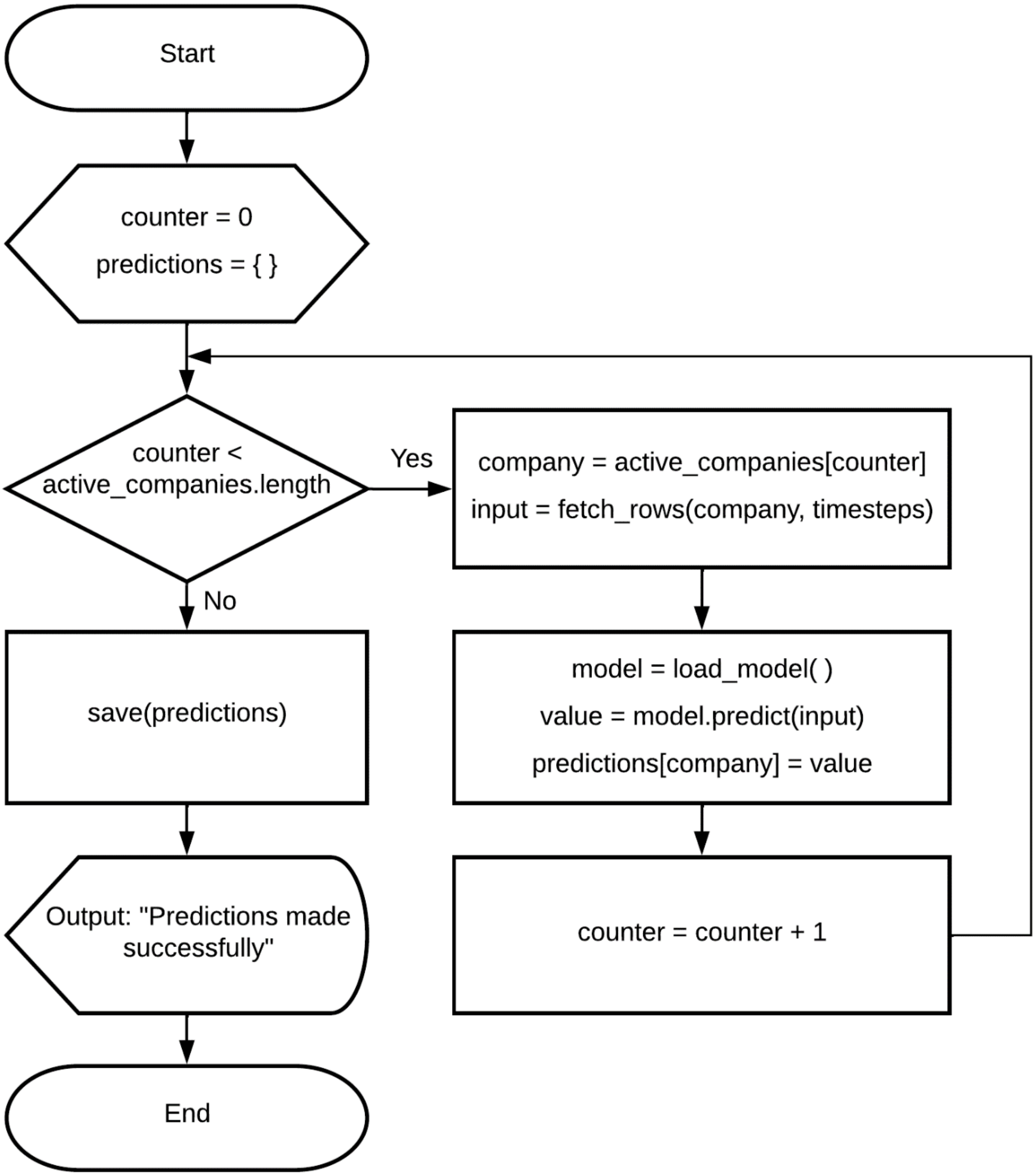

Figure 14: Flowchart of the make predictions use case

(iii) Coding and unit testing

Actual coding was done in order to implement all the designs and thereby accomplish every functionality in each iteration. Each functionality was tested to ensure it worked correctly before proceeding with the next functionality.

(iv) Acceptance testing

The system was presented to stakeholders to verify that it met the proposed requirements. Areas of improvement were identified for further refinements. Specifically, it was suggested to show predicted price change and predicted percentage change in addition to showing the predicted closing price. Furthermore, it was commented to add a visual impression by showing red colour for negative price changes and green colour for positive price changes. Also, to show visualization of predictions over time.

(v) Release

The final product, after refinements, was presented to stakeholders. New requirements were noted down and provided as recommendations (Section 5.2), for further improvements in the future.

3.17 Prototype Validation

A total of 17 stakeholders were engaged to validate the system. Specifically, they validated the stock price prediction system in terms of: usability of the system, consistency of the system, ability of the system to provide feedback, ability to predict the closing price, ability to make visualizations and usefulness of the system. The validators were required to rate each aspect into a scale of either Very High, High, Average, Low or Very Low (Appendix 2).

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

4.1.1 Identified Features

The results from the FGDs conducted with stockbrokers and investment advisors from the study area showed that historical stock prices (previous closing price, opening price, low price and high price) were mentioned by the majority of respondents (94%) to be suitable features, followed by investors' participation (71%), investors' sentiment (65%) and performance of a company (58%). On the other hand, all respondents (100%) mentioned demand (represented by outstanding bids and volume) and supply (represented by outstanding offers) to be a strong factor for stock price fluctuation, however 88% of respondents suggested that there is a need to account for the number of outstanding shares in the demand and supply variables since each company has different number of outstanding shares. Figure 15 provides an illustration of the results from FGDs. Table 10 elaborates further the results from FGDs.

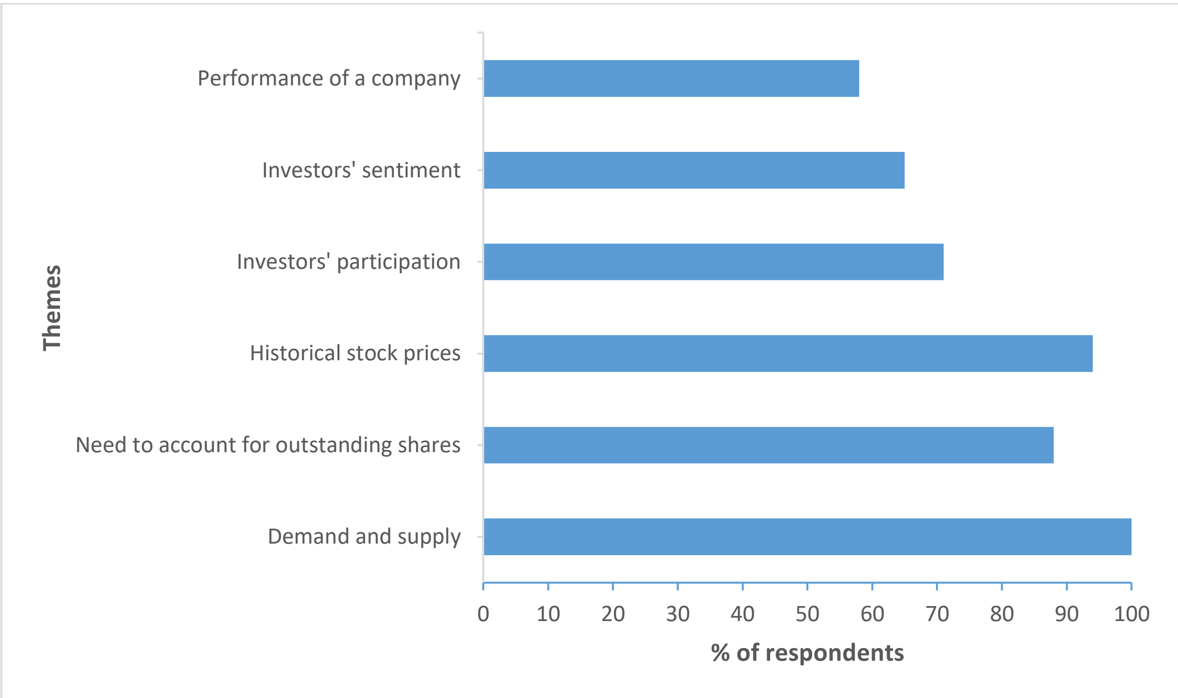

Figure 15: Themes identified during FGDs

Table 10: Information obtained from FGDs

Theme	Reasons given
Historical stock prices	The closing price correlates with the previous closing price. Furthermore, the closing price of a stock at the end of a trading day is calculated as the volume-weighted average of all the prices on that day.
Investors' participation	Trading is conducted in the form of an auction, hence the quality of investors' participation on a given trading day, determines the direction of the price. Foreign investors are more risk tolerant due to their good experience of stock markets, however, their investment decisions are influenced by the investment decisions of domestic institutional investors.
Investors' sentiment	Release of positive news about a particular company such as a tender offer, can cause increase in demand for the company and subsequent price increase.
Performance of a company	Positive company performance usually results in higher dividend payouts. This results in increase in demand for the company and vice-versa.
Demand and supply	When the demand for particular company shares increases their price increases and vice-versa.
Need to account for the number of outstanding shares	Each company has different number of outstanding shares. In order to build a joint model for multiple companies, it is required to divide the demand and supply variables of each company by the corresponding number of outstanding shares.

On the other hand, Table 11 shows the features represented by each theme.

Table 11: Features represented by each theme

Theme	Features
Historical closing prices	Opening price, low price, high price, previous closing price
Investors' participation	Turnover from shares bought by foreign investors, turnover from shares bought by local investors
Investors' sentiment	Sentiment (obtainable through analyzing news articles and comments related to a certain listed company)
Performance of a company	Accounting ratios such as profitability ratios, obtainable from companies' periodic financial statements
Demand and supply	Demand variables: outstanding bids, volume
	Supply variables: outstanding offers
Need to account for outstanding shares	Outstanding bids/outstanding shares, volume/outstanding shares and outstanding offers/outstanding shares

However, the features related to the performance of companies were not incorporated into the dataset because financial statements are released by companies on a quarterly, semi-annually and annual basis, which makes it difficult to incorporate such data into a daily dataset (Bustos & Pomares-Quimbaya, 2020; Munisi, 2017). Furthermore, investors' sentiment was not incorporated into the dataset due to the low availability of such data (in Tanzania) for analysis, this is because the market intermediaries have not been active enough to push information about the market into the media in order to create awareness and speculation (Abbas *et al.*, 2016; FinSights Lab, 2020; Massele *et al.*, 2015). Table 12 shows the final list of features available in the dataset, indicating the method by which each feature was obtained. Table 13 shows the importance score of each feature according to F score values returned by *f_regression ( ).* Figure 16 shows a graphic illustration of feature importance scores. All features had scores greater than zero, however, some features had very high scores compared to others. Hence, experiments were conducted with different combinations of the features in order to determine the combination that yields best results.

Table 12: Features in the dataset as obtained from empirical literature, FGDs or both

Feature	Empirical literature	FGDs	Both
Opening price			✓
High price			✓
Low price			✓
Closing price			✓
Deals	✓		
Price change	✓		
Volume	✓		
Outstanding bids	✓		
Outstanding offers	✓		
Volume/outstanding shares		✓	
Outstanding bids/outstanding shares		✓	
Outstanding offers/outstanding shares		✓	
Turnover from shares bought by foreign investors		✓	
Turnover from shares bought by local investors		✓	

Table 13: Feature Importance Scores

Feature	Importance Score
Opening price	1619.4257
High price	931.3051
Low price	903.4921
Previous closing price	1670.2642
Deals	211.6698
Price change	83.1323
Volume	302.7133
Outstanding bids	425.9978
Outstanding offers	447.5113
Volume/outstanding shares	302.7309
Outstanding bids/outstanding shares	426.1642
Outstanding offers/outstanding shares	447.5719
Turnover from shares bought by foreign investors	152.6547
Turnover from shares bought by local investors	151.5805

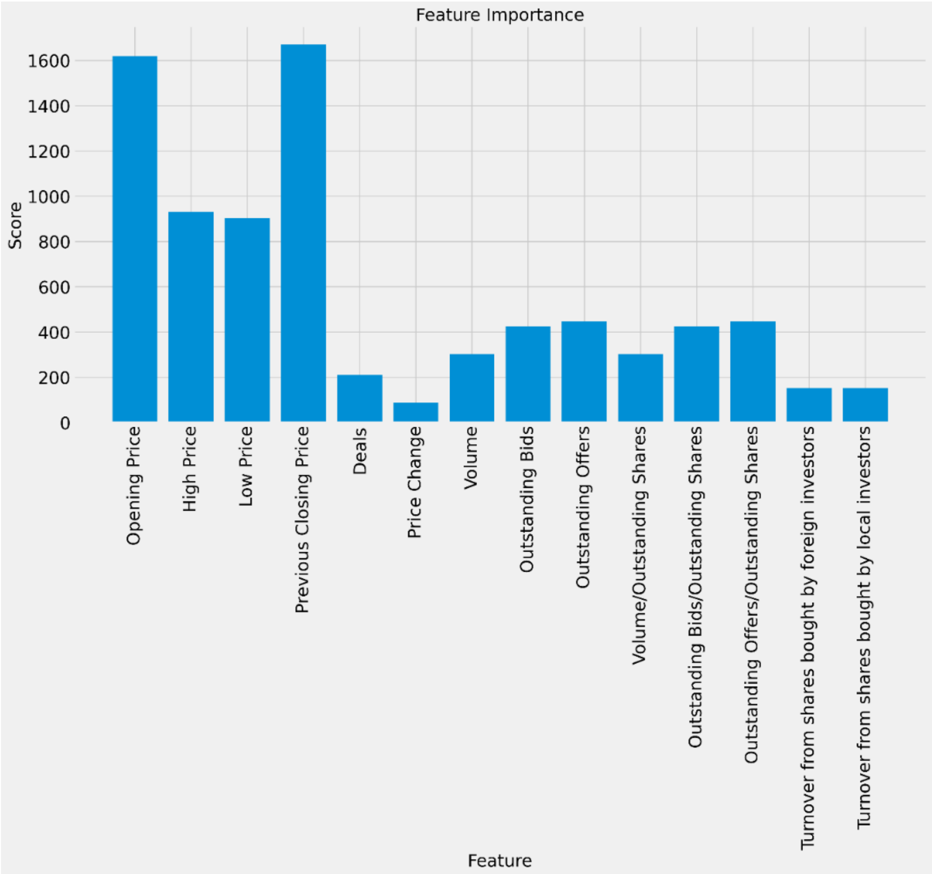

Figure 16: Graphic illustration of feature importance

4.1.2 Selected Companies

Table 14 shows companies whose average of *volume/outstanding shares* was greater or equal to 0.005%.

After performing data visualizations on the closing price attribute of each of the identified companies in Table 14, TBL and TCC were eventually dropped because it was observed that their closing prices have been static for over two (2) years. Figure 17 shows the closing price history of TBL whereas Fig. 18 shows the closing price history of TCC.

Table 14: Companies Whose Average of Volume/Outstanding Shares ≥ 0.005%

Company	Volume/Outstanding Shares (%)
CRDB	0.0137
DSE	0.0520
NICO	0.0151
NMB	0.0099
SWIS	0.0310
TBL	0.0089
TCC	0.0091
TCCL	0.0230
TOL	0.0214

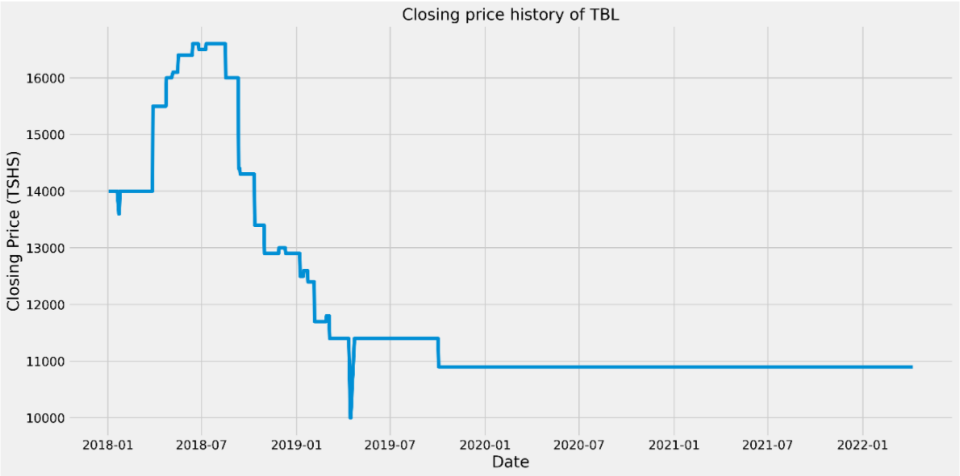

Figure 17: Graph showing closing price history of TBL

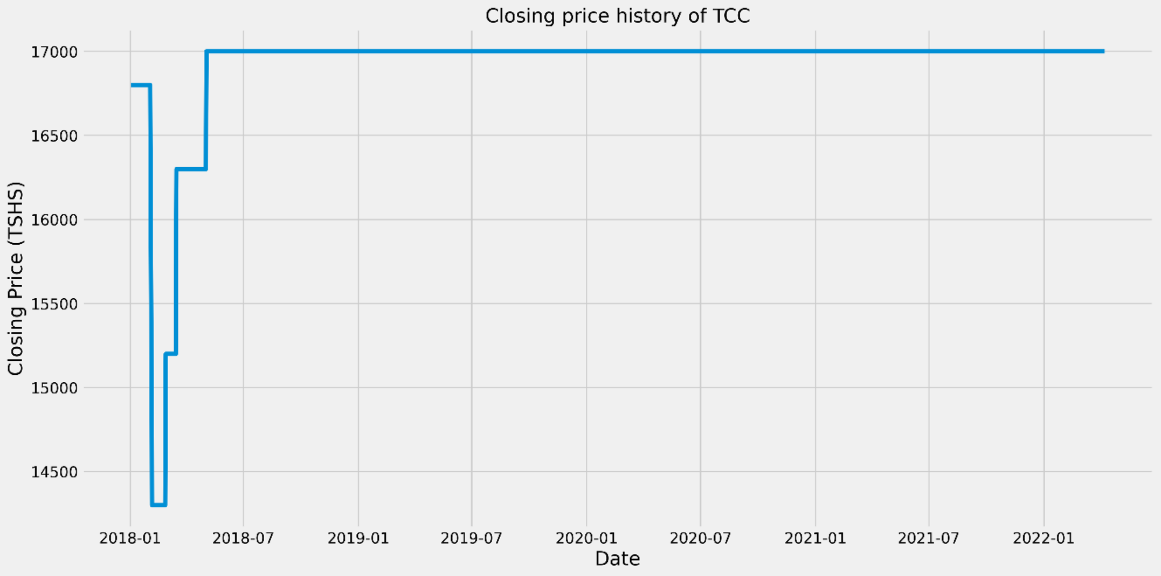

Figure 18: Graph showing closing price history of TCC

On the other hand, TPCC was appended even though its average of *volume/outstanding shares* was 0.0034%, which was below the threshold of 0.005%, because its price has been increasing gradually over the years, and as such it has been attracting investors' attention. Figure 19 shows the closing price history of TPCC.

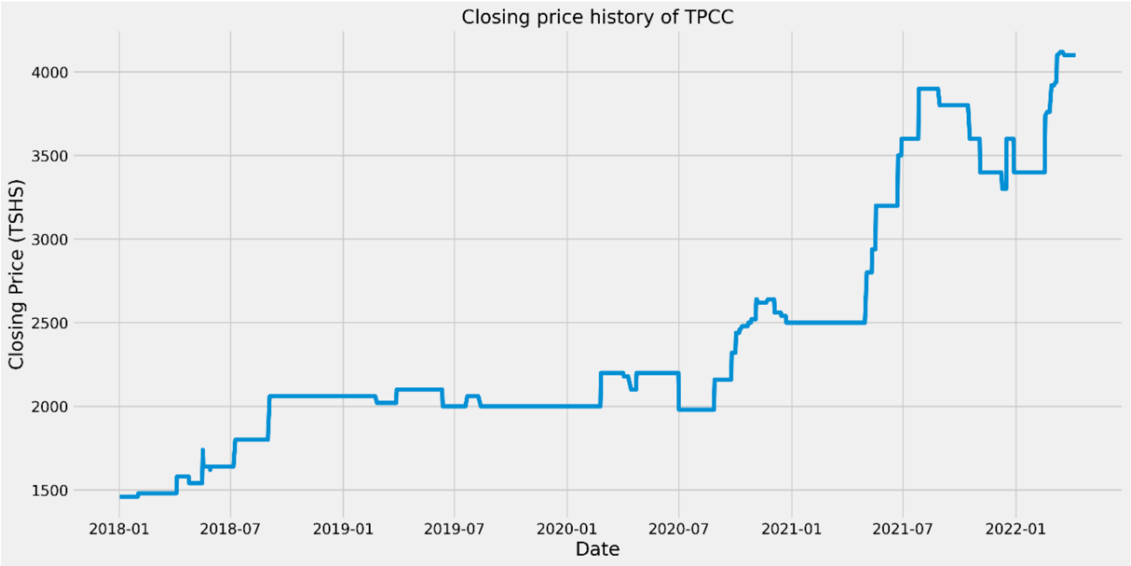

Figure 19: Graph showing closing price history of TPCC

Therefore, the final dataset consisted of eight (8) companies which were: CRDB, DSE, NICO, NMB, SWIS, TCCL, TPCC and TOL. In total the dataset had 8668 rows and 14 features. Figure 20 shows the dataset consisting of selected companies, before dropping the *company name* and *date* features.

	Company Name	Date	Opening Price	Closing Price	High Price	Low Price	Deals	Outstanding Bids	Outstanding Offers	Volume	Outstanding Bids/Outstanding Shares	Outstanding Offers/Outstanding Shares	Volume/Outs
0	DSE	2016-07-12	500.0	935.0	1000.0	705.0	201.0	622986.0	4000.0	849121.0	0.026149	0.000168	[illegible]
1	DSE	2016-07-13	935.0	1000.0	1010.0	1000.0	289.0	777186.0	70027.0	1129302.0	0.032622	0.002939	[illegible]
2	DSE	2016-07-14	1000.0	1000.0	1010.0	1000.0	139.0	901396.0	159348.0	491423.0	0.037836	0.006689	[illegible]
3	DSE	2016-07-15	1000.0	1020.0	1030.0	1000.0	116.0	176140.0	26928.0	862768.0	0.007393	0.001130	[illegible]
4	DSE	2016-07-18	1020.0	1040.0	1060.0	1000.0	70.0	1170096.0	23110.0	264928.0	0.049114	0.000970	[illegible]
...	...	...	...	...	...	...	...	...	...	...	...	...	
8663	TPCC	2022-04-08	4100.0	4100.0	4260.0	4260.0	2.0	500.0	0.0	1000.0	0.000003	0.000000	[illegible]
8664	NICO	2022-04-08	580.0	610.0	620.0	560.0	9.0	10.0	18214.0	46397.0	0.000000	0.000263	[illegible]
8665	DSE	2022-04-08	1420.0	1500.0	1540.0	1500.0	6.0	9279.0	5000.0	15377.0	0.000389	0.000210	[illegible]
8666	CRDB	2022-04-08	380.0	380.0	375.0	370.0	29.0	3640.0	400003.0	51121.0	0.000001	0.000153	[illegible]
8667	NMB	2022-04-08	2700.0	2700.0	2580.0	2500.0	7.0	4090.0	3842.0	6306.0	0.000008	0.000008	[illegible]

8668 rows × 16 columns

Figure 20: Dataset consisting of selected companies

4.1.3 Experimental Results

(i) Experiment 1 Results

Table 15 shows the optimal hyperparameter values which were also sustained in experiment 2 and in experiment 3.

Table 15: Optimal Hyperparameter Values for Each Model

Hyperparameter	LSTM	Bi-LSTM	GRU
Hidden layer size	2	2	2
Hidden units	80:20	50:110	290:140
Batch size	64	64	64
Number of epochs	198	116	178
Timesteps	10	10	10

The results obtained are as shown in Table 16.

Table 16: Results of Experiment 1

MODEL	MSE	RMSE	MAE	MAPE
LSTM	109.6950	10.4734	6.9094	0.6359
Bi-LSTM	146.6017	12.1079	7.6524	0.7909
GRU	155.2097	12.4583	8.8794	0.8663

The results show that the LSTM model performed better than the other models, followed by the Bi-LSTM model.

(ii) Experiment 2 Results

The results are as shown in Table 17.

Table 17: Results of Experiment 2

MODEL	MSE	RMSE	MAE	MAPE
LSTM	22.5851	4.7424	2.4377	0.2147
Bi-LSTM	70.9222	8.4215	5.4102	0.5113
GRU	75.9728	8.7162	5.8629	0.5802

The results show that taking into account the number of outstanding shares helped the models to learn better. This can be evidenced by the drop in RMSE in each model as compared to the results of experiment 1. In particular, the RMSE of the LSTM model dropped by 54.72%, while that of the Bi-LSTM model dropped by 30.45% whereas that of the GRU model dropped by 30.04%.

(iii) Experiment 3 Results

The results are as shown in Table 18.

Table 18: Results of Experiment 3

MODEL	MSE	RMSE	MAE	MAPE
LSTM	17.4876	4.1818	2.1695	0.2106
Bi-LSTM	66.0114	8.1247	5.1218	0.5024
GRU	67.8337	8.2361	5.3437	0.5233

In experiment 3, in addition to accounting for outstanding shares, investors' participation attributes were also appended. The results show that inclusion of investors' participation attributes helped to improve the prediction accuracy even further. As compared to experiment 2 results, the RMSE of the LSTM model dropped by 11.8%, while that of the Bi-LSTM model dropped by 3.5% whereas that of the GRU model dropped by 5.5%.

(iv) Experiment 4 Results

Table 19 shows the optimal hyperparameter values for this experiment.

Table 19: Optimal Hyperparameter Values for Experiment 4

Hyperparameter	LSTM	Bi-LSTM	GRU
Hidden layer size	2	2	2
Hidden units	200:20	140:110	80:140
Batch size	16	128	16
Number of epochs	98	114	130
Timesteps	10	10	10

The results were showed in Table 20.

Table 20: Results of Experiment 4

MODEL	MSE	RMSE	MAE	MAPE
LSTM	30.5578	5.5279	3.3733	0.3589
Bi-LSTM	39.8279	6.3109	4.4128	0.6043
GRU	61.8147	7.8622	4.8340	0.4943

The results show that the RMSE of the LSTM model increased by 32.19% compared to experiment 3. On the other hand, the RMSE of Bi-LSTM dropped by 22.32% while that of GRU dropped by 4.54% compared to experiment 3.

(v) Experiment 5 Results

Table 21 shows the optimal hyperparameter values for this experiment.

Table 21: Optimal Hyperparameter Values for Experiment 5

Hyperparameter	LSTM	Bi-LSTM	GRU
Hidden layer size	2	2	2
Hidden units	140:170	170:170	200:260
Batch size	16	128	64
Number of epochs	64	50	102
Timesteps	10	10	10

The results were showed in Table 22.

Table 22: Results of Experiment 5

MODEL	MSE	RMSE	MAE	MAPE
LSTM	38.8456	6.2326	3.7810	0.3496
Bi-LSTM	60.4729	7.7764	4.6515	0.4061
GRU	67.6766	8.2266	5.1715	0.5127

The results show that the RMSE of the LSTM model increased by 49.04% compared to that of experiment 3. On the other hand, the RMSE of Bi-LSTM model dropped by 4.29% while that of GRU model dropped by 0.12% compared to experiment 3.

4.1.4 Stakeholder Requirements

The results from the FGDs conducted with the stockbrokers and investment advisors showed that 94% suggested to put data entry through form and through excel, 88% requested for the ability to view and manage entered data, and to view predictions, 82% requested for visualization of market statistics and ability to login and logout of the system. On the other hand, 47% requested for an interface to change the password while 29% of respondents suggested to put user authorization to differentiate between the administrator (who will have permission to add or delete users) and other staff. The FGDs results are illustrated in Fig. 21.

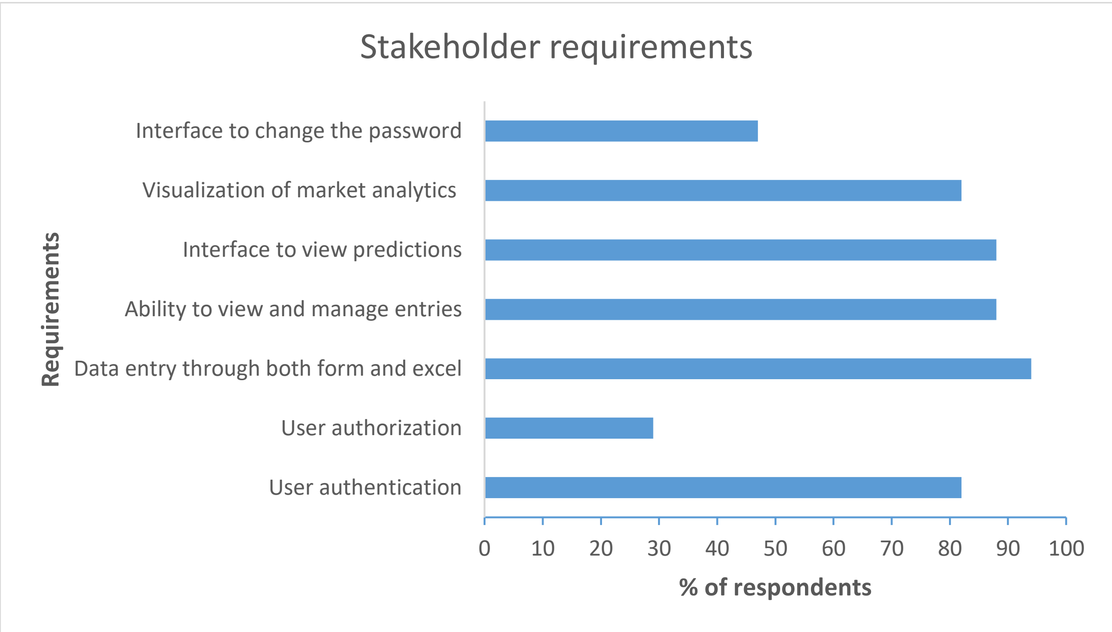

Figure 21: Stakeholder requirements for prototype development

4.1.5 The Developed Prototype

The developed prototype consists of login interface (Fig. 22), data entry form (Fig. 23), excel upload interface (Fig. 24), view and manage stock data interface (Fig. 25), market analytics page (Fig. 26), an interface to run the model (Fig. 27), a page for viewing predictions (Fig. 28), change password interface (Fig. 29) and an interface for the admin to add or delete users (Fig. 30).

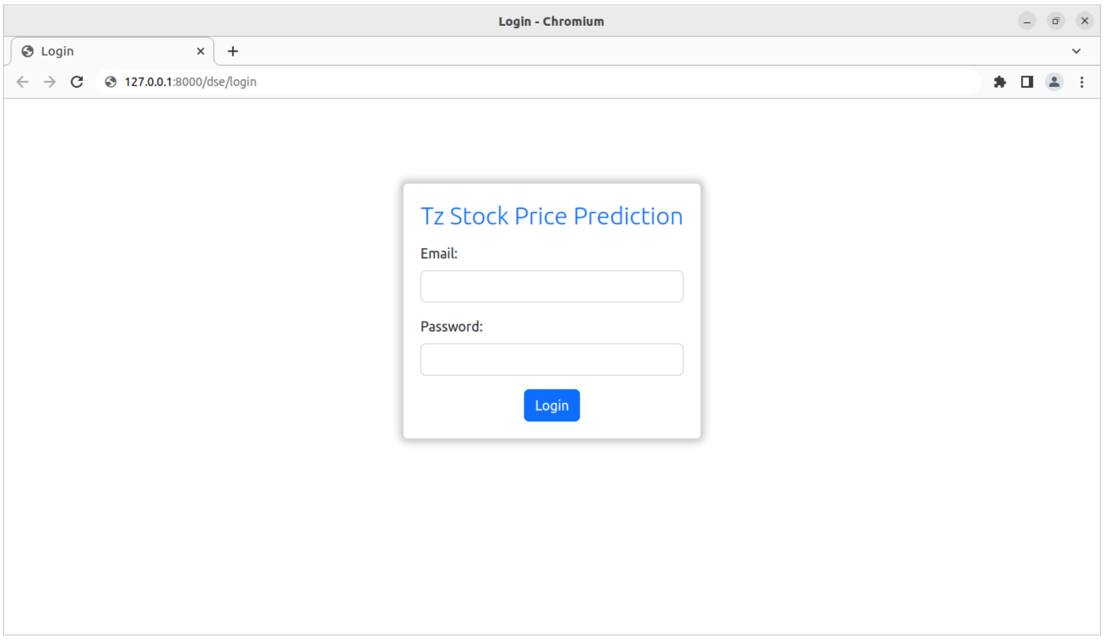

Figure 22: Login interface

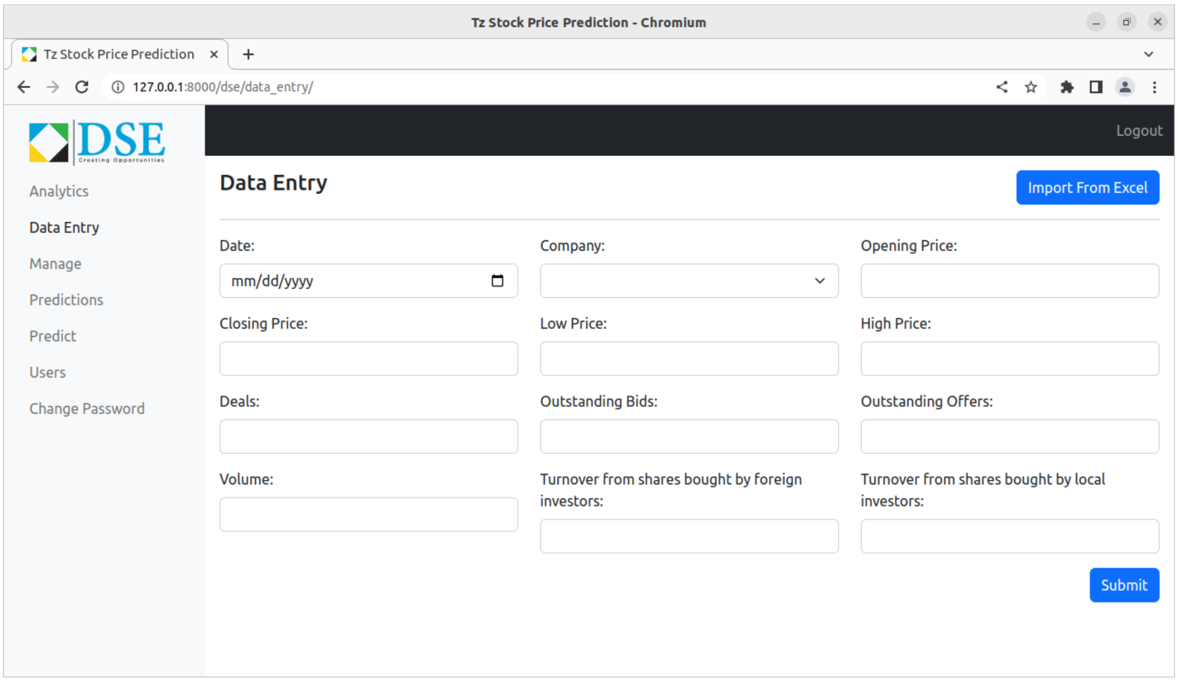

Figure 23: Data entry form

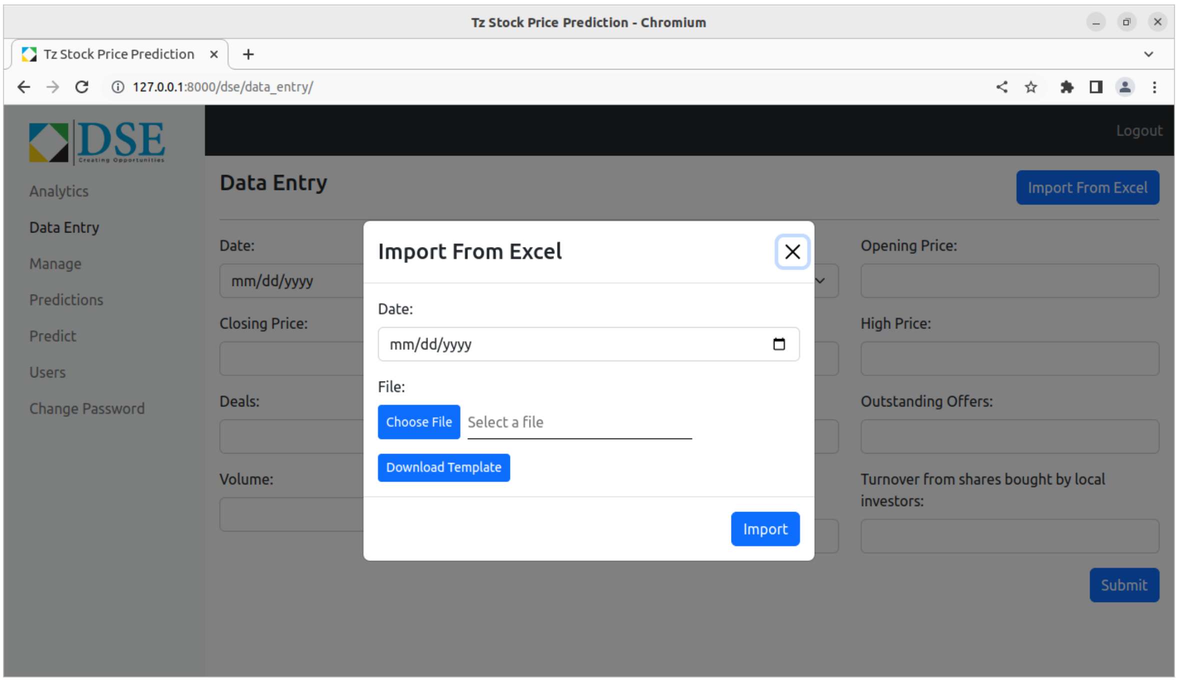

Figure 24: Excel upload interface

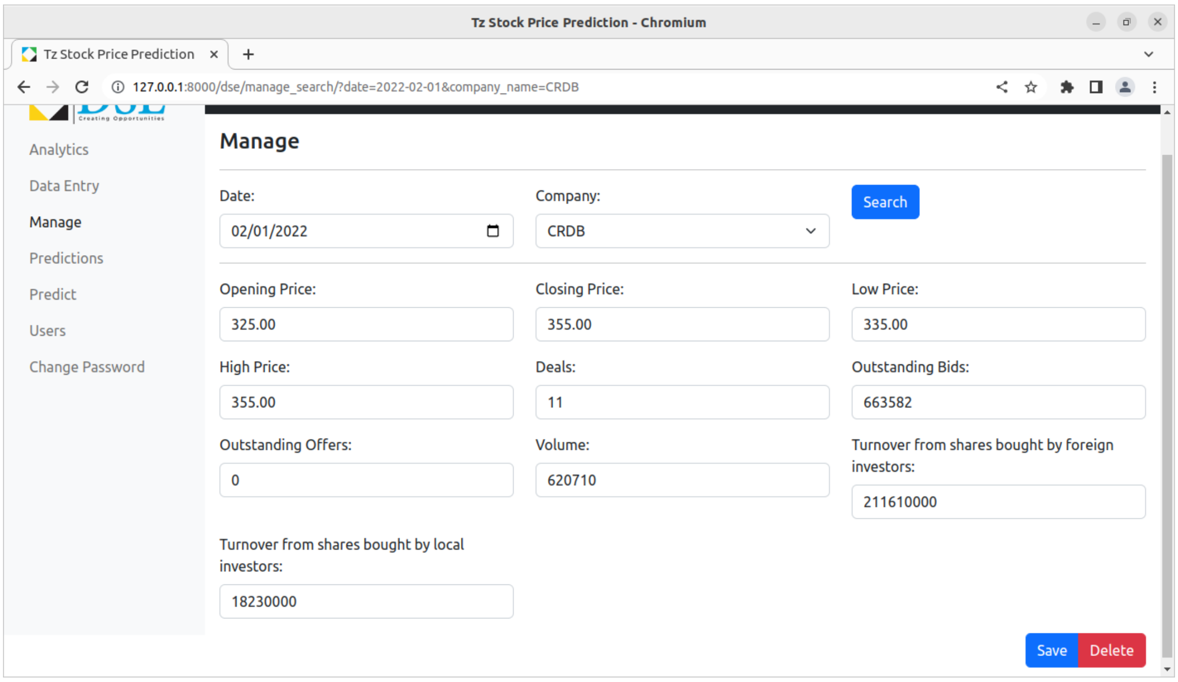

Figure 25: Manage entered data

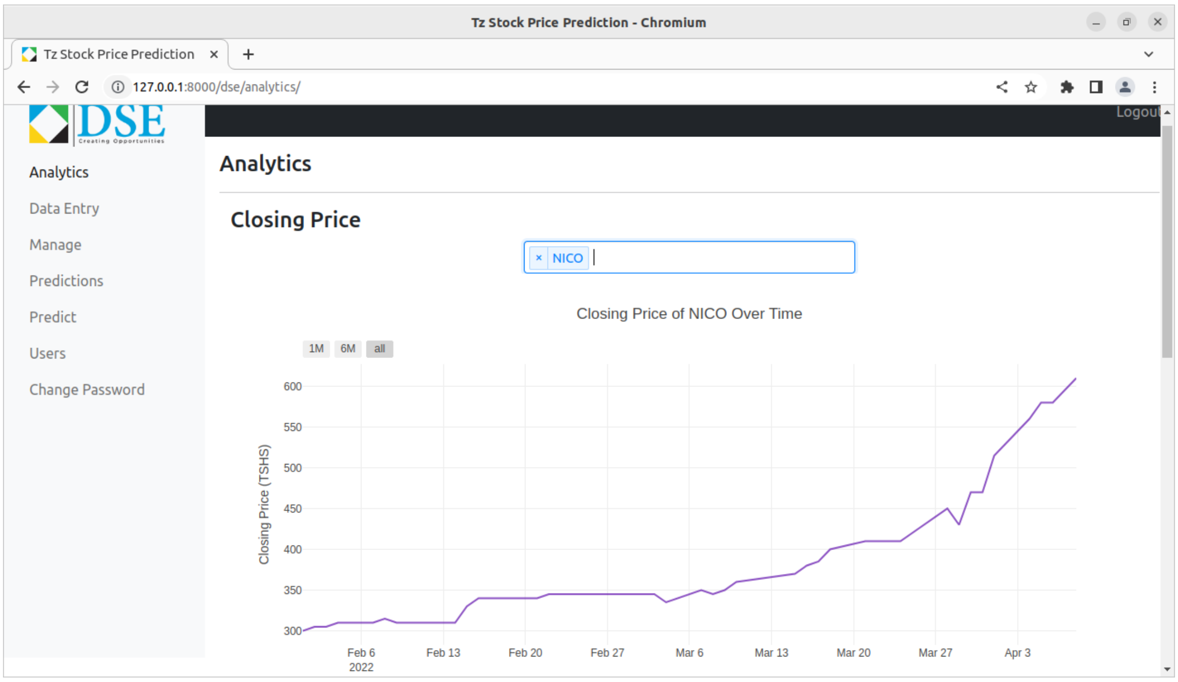

Figure 26: Market analytics page

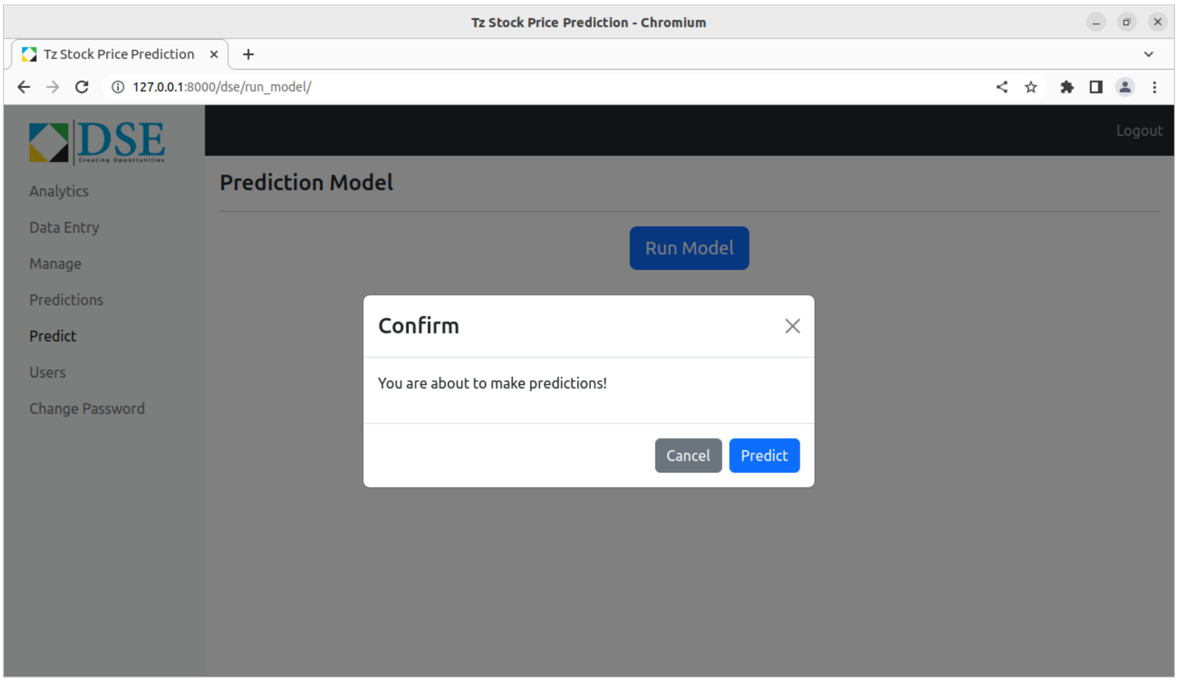

Figure 27: Interface to run the model

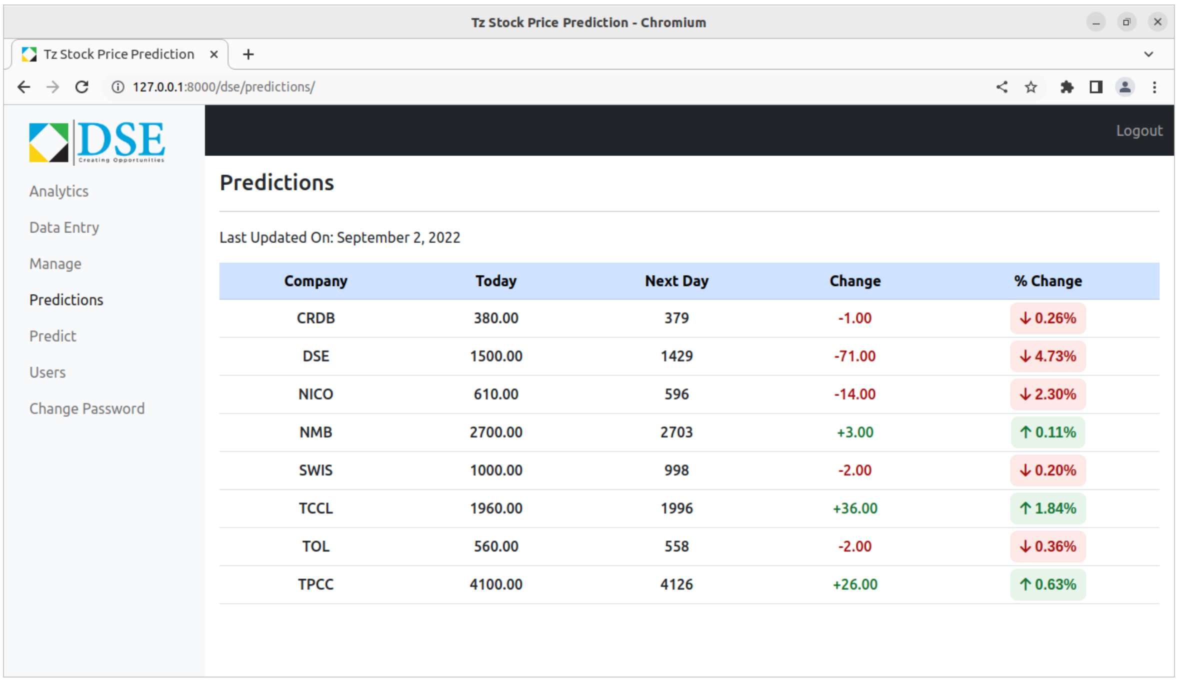

Figure 28: Page for viewing predictions

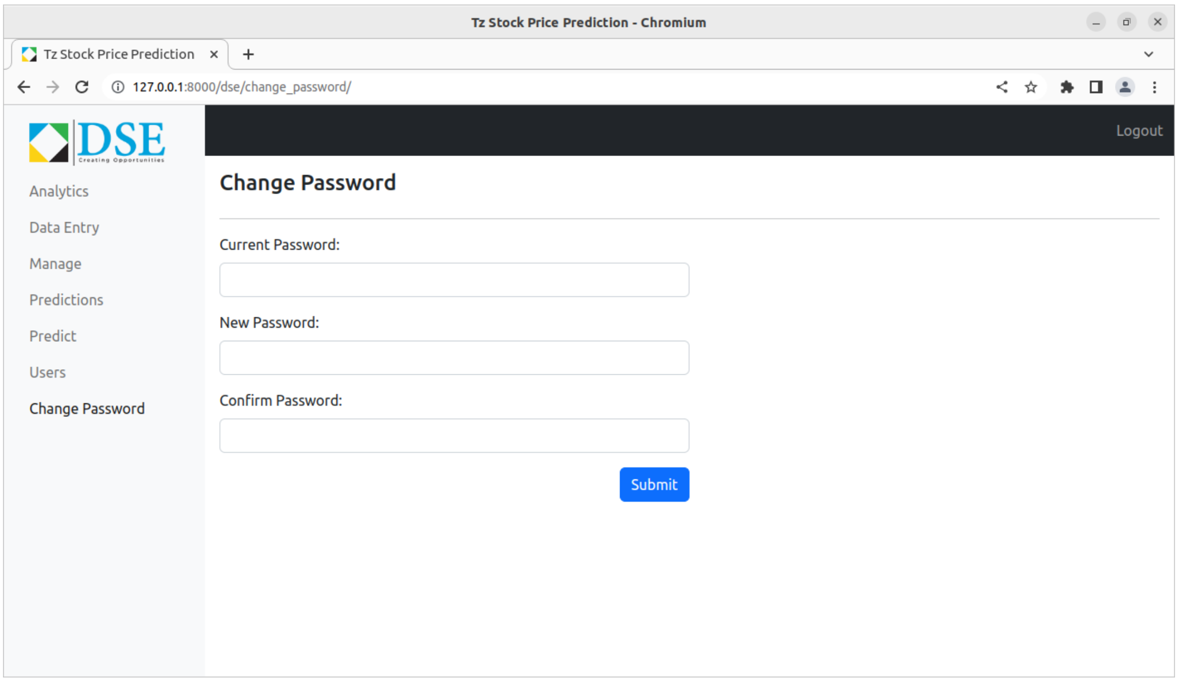

Figure 29: Change password interface

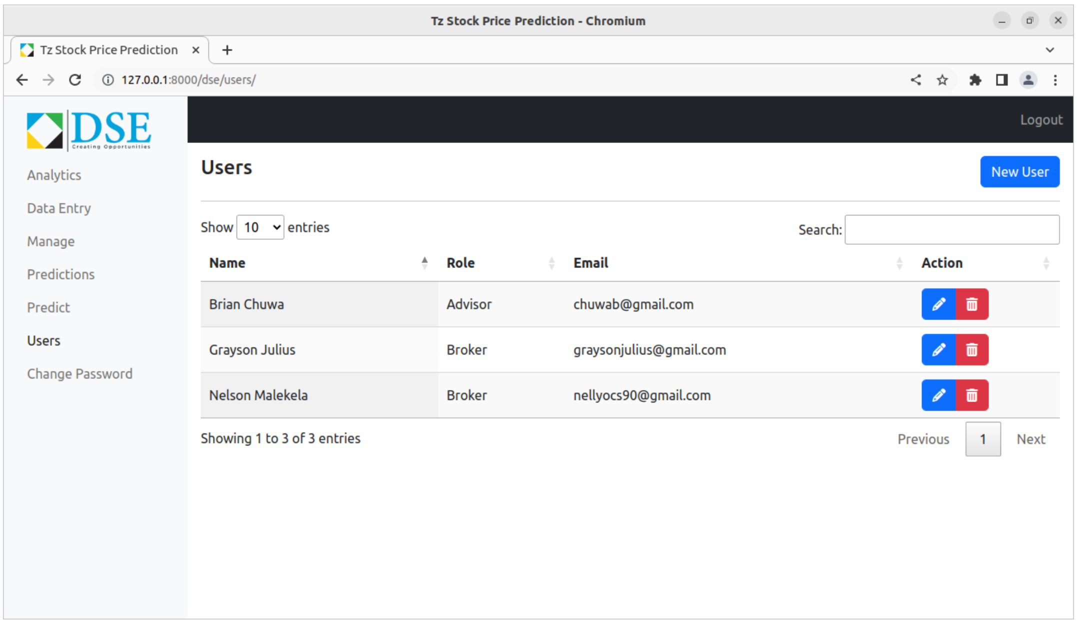

Figure 30: User management interface (only visible to the administrator)

4.1.6 Prototype Validation Results

The prototype was hosted on heroku platform, whereby, stockbrokers and investment advisor s from selected intermediary companies, were given time to assess the tool for two weeks. The participants responded to the provided questionnaire whereby each presented aspect was rated into a scale of either Very High, High, Average, Low or Very Low. The results showed that

94% of respondents rated the system as Very High in terms of its ability to give feedback, 76% rated the system as High in terms of its ability to predict the closing price while 65% rated the system as High in terms of its ability to visualize market statistics. Figure 31 shows the stakeholder evaluation results.

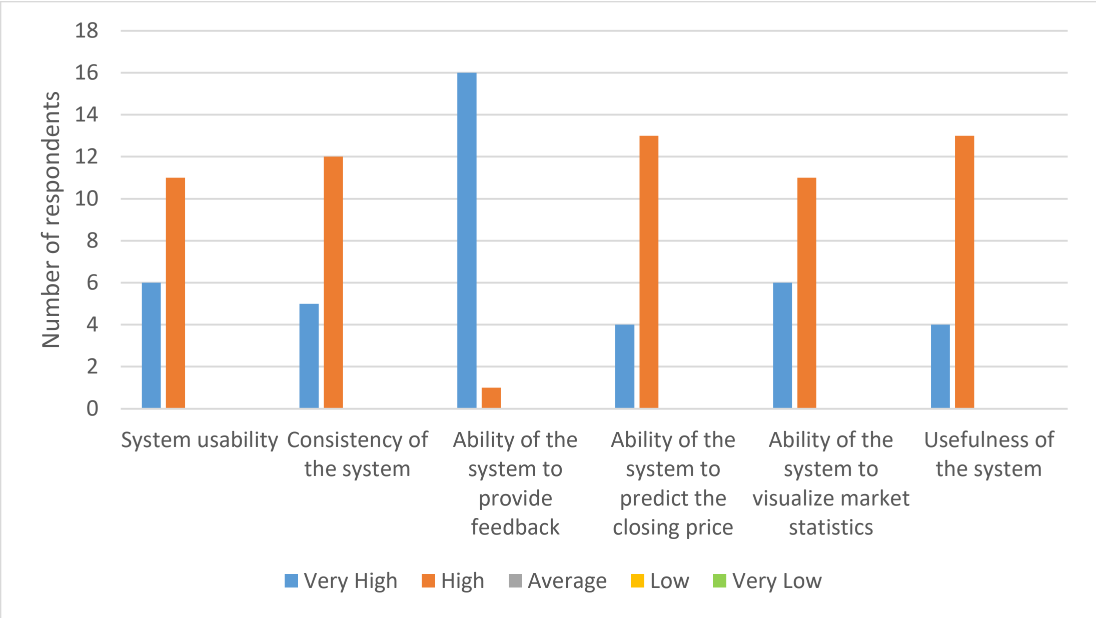

Figure 31: Stakeholder validation results

4.2 Discussion

The results showed that about eight companies were found to be active out of 22 domestic listed companies. This shows that the number of companies with active shares is very small to provide enough diversification to investors, rendering the market unattractive. When a company fails to realize improvement in performance after listing, it loses investors' interest and gradually drops in value resulting into inactive shares. Three reasons have been identified as potential factors for company deterioration in performance upon listing. First, companies have a tendency to go public at the peak of their long-run performance, which they know cannot be sustained in the future (Munisi, 2017). Second, the dilution of shares when a company goes public can give rise to agency problem which in turn, may cause poor financial performance in the long run (Kim *et al.*, 2004). Third, managers may attempt to window-dress accounting information before listing in order to portray artificial good performance to attract investors to invest in shares (Munisi, 2017). Due to these reasons a company may fail to reach investors' expectations upon listing and thereby gradually lose value.

Tanzania Breweries Limited (TBL) and TCC prices were found to have been static for over two years in spite of their high trading volumes. According to the information obtained from stockbrokers and investment advisors, this is likely due to block trades that are heavily conducted on their shares. Block trades are large, privately negotiated securities' transactions conducted outside the open market to lessen the effect on a security's price (*Block Trade,* 2022). Block trades are reflected in a stock's volume statistics but are not used to update the closing price (*Dar Es Salaam Stock Exchange PLC Rules*, 2022). Therefore, the study dropped these companies, as including them would have distracted the model from properly learning the relationship between volume and the closing price.

The GRU model had higher RMSE than the LSTM model. This is probably due to its simpler structure which could have made it not to efficiently memorize information. The LSTM model consists of three gates that control its memory cells, namely; forget gate, input gate and the output gate; on the other hand, the GRU model consists of only two gates: update gate and reset gate. Furthermore; the LSTM model stores information based on the current input (short-term memory) and based on previous inputs (long-term memory) whereas the GRU model keeps track of both short-term and long-term information through only one memory structure known as the hidden state (Olah, 2015; Staudemeyer & Morris, 2019; Yu *et al.*, 2019).

The Bi-LSTM model had higher RMSE than the LSTM model. This is likely due to its complex structure which traverses the input data both forward and backward (Zvornicanin, 2022). This shows that the stock dataset used in this study did not require this complex mechanism.

The performance of all the models improved when the number of outstanding shares of each company was taken into account. In particular, the RMSE of the LSTM model dropped by 54.72%, whereas that of the Bi-LSTM model dropped by 30.45% while that of the GRU model dropped by 30.04%. This indicates that the models were able to correctly learn the relationship between volume, outstanding bids, outstanding offers and the closing price when the number of outstanding shares was taken into account compared to when it was not.

The performance of all the models improved further, when turnover from shares bought by foreign investors and turnover from shares bought by local investors were appended among the features. Specifically, the RMSE of the LSTM model dropped a further 11.8%, whereas that of the Bi-LSTM dropped a further 3.5% while that of the Bi-LSTM model dropped a further 5.5%. This implies that inclusion of investors' participation among predictive attributes can help to improve prediction accuracy.

The RMSE of the LSTM model increased by 32.19% in experiment 4 when only opening price, high price, low price and previous closing price were used as features. It also increased by 49.04% in experiment 5 when only the previous closing price was used as the feature. Nevertheless, the LSTM model performed best in experiment 3 when it was given eleven input features. This suggests that LSTM can compute more information when given more features, and thereby improve prediction accuracy. Nelson *et al.* (2017) employed LSTM with 179 features in order to predict the direction of the stock price, whether it will be up or down, in the next 15 minutes. The results showed that LSTM was able to learn from the huge input dimension without the need to use any dimension reduction technique. Interestingly, it outperformed the Multi-layer Perceptron (MLP) and Random Forest models.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study was conducted in order to develop a deep learning model for predicting stock prices in Tanzania. Recurrent neural network architectures, particularly, Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit were considered for model development due to their ability to efficiently learn long-term dependencies in time-series data. Specifically, the study was conducted to investigate whether accounting for the number of outstanding shares of each company can help to improve the prediction accuracy of the model when building a single model to predict the stock prices of multiple companies. Furthermore, the study also investigated whether inclusion of investors' participation among predictive attributes can help to improve prediction accuracy.

The results showed that the LSTM model performed better than the other models at predicting the next day's closing price. Accounting for the number of outstanding shares helped to improve prediction accuracy of all the models. The prediction accuracy improved further when investors' participation attributes were appended among the predictive features. Specifically, the RMSE of the LSTM model dropped by 54.72% when the number of outstanding shares of each company was taken into account. It also dropped a further 11.8% when investors' participation attributes were appended.

5.2 Recommendations

Stockbrokers and investment advisors should use the developed prototype in order to ensure more accurate stock recommendations. The developed prototype will help to reduce human error and bias during advisory. This will eventually help to reduce investment risk and thereby attract more investors into the market.

The developed prototype focuses on predicting next day's closing price based on historical stock data. The prototype can be improved to provide investment suggestions. This can be achieved through embedding a machine learning model for classifying someone's financial risk tolerance level. After identification of a client's risk tolerance level, the prototype will be able to provide applicable investment recommendations.

Researchers could improve on this work by exploring algorithmic trading. Algorithmic trading merges both the prediction and the trading decision to take based on the prediction. Deep reinforcement learning is a possible direction in this case. This is because it has advantages of simulating many possible cases and making faster and better trading choices than human traders (Jiang, 2021).

Furthermore, there is a shortage of literature to exploit machine learning and deep learning to detect fraud in financial statements in Tanzania. Therefore, research needs to be conducted to fill the available gap. The resulting models will help managers of companies and stock market regulators in Tanzania, to accurately identify misleading reports. Ultimately, this will help to ensure the sustainability of the companies and the stock market.

Moreover, this study focused on LSTM, Bi-LSTM and GRU to predict stock prices in Tanzania. Hence future studies can investigate other deep learning approaches such as generative adversarial networks or use hybrid model combinations that have been least studied in literature. On the other hand, other types of attributes can also be explored.

The results showed that there are only about eight active companies out of 22 domestic listed companies. Therefore, the study recommends that the Dar es Salaam Stock Exchange (DSE) and the Capital Markets and Securities Authority (CMSA) in collaboration with the investment advisors need to conduct careful assessment of a company before allowing it to list on the exchange. This will help to ensure that companies listed are only those with potential for growth, thereby reassuring investor satisfaction and a healthy stock market in the long run.

REFERENCES

Abbas, A. O., Pei, Y. X., & Rui, Z. (2016). Impact of stock market on economic growth evidence: Dar-es Salaam stock exchange–Tanzania. *Journal of Finance and Accounting*, *4*(6), 321–327.

Active Stocks. (2022). Investopedia. https://www.investopedia.com/terms/a/active-stocks.asp.

Alese. (2018). *The curious case of the vanishing & exploding gradient | Explaining | Medium*. https://medium.com/learn-love-ai/the-curious-case-of-the-vanishing-exploding-gradient-bf58ec6822eb.

Alkhatib, K., Khazaleh, H., Alkhazaleh, H. A., Alsoud, A. R., & Abualigah, L. (2022). A New Stock Price Forecasting Method Using Active Deep Learning Approach. *Journal of Open Innovation: Technology, Market, and Complexity*, *8*(2), 96.

Ansah, K., Denwar, I. W., & Appati, J. K. (2022). Intelligent Models for Stock Price Prediction: A Comprehensive Review. *Journal of Information Technology Research*, *15*(1), 1–17. https://doi.org/10.4018/JITR.298616.

Atenga, E. M. E., & Mougoué, M. (2021). Return and volatility spillovers to African equity markets and their determinants. *Empirical Economics*, *61*(2), 883–918.

Brownlee, J. (2018). How to develop LSTM models for time series forecasting. *Machine Learning Mastery, 14*. https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting.

Brownlee, J. (2020). How to perform feature selection for regression data. *URI= https://machinelearningmastery. com/feature-selection-for-regression-data*.

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A Systematic Review. *Expert Systems with Applications*, *156*, 113464.

Capital Markets and Securities Authority. (2013). *A Guide to Enterprise Growth Market Segment at the Dar es Salaam Stock Exchange*.

Carney, M. (2019). The growing challenges for monetary policy in the current international monetary and financial system. In *Remarks at the Jackson Hole Symposium* 23, 377-

411.

Chen. (2020). *A Practical Introduction to Early Stopping in Machine Learning*. Medium. https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd.

Chen, M. Y., Liao, C. H., & Hsieh, R. P. (2019). Modeling public mood and emotion: Stock market trend prediction with anticipatory computing approach. *Computers in Human Behavior*, *101*, 402–408.

Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal*, *2*, 100015. https://doi.org/10.1016/j.dajour.2021.100015.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1724–1734.

Chung. (2016). *How does LSTM help prevent the vanishing (and exploding) gradient problem in a recurrent neural network?* https://www.quora.com/How-does-LSTM-help-prevent-the-vanishing-and-exploding-gradient-problem-in-a-recurrent-neural-network/answer/Wonchang-Chung-1.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. https://doi.org/10.48550/arXiv.1412.3555.

Clifford, D. (2022). *The 5 Biggest Biases That Affect Decision-Making*. NeuroLeadership Institute. https://neuroleadership.com/your-brain-at-work/seeds-model-biases-affect-decision-making.

Dar es Salaam Stock Exchange PLC achieves Full Membership to the World Federation of Exchanges | Dar es Salaam Stock Exchange PLC. (2019). https://dse.co.tz/content/dar-es-salaam-stock-exchange-plc-achieves-full-membership-world-federation-exchanges.

Dar Es Salaam Stock Exchange PLC Rules. (2022). Dar es Salaam Stock Exchange PLC.

Dar es Salaam Stock Exchange PLC (DSE). (2016). *Handbook*. Dar es Salaam Stock Exchange PLC (DSE).

Dingli, A., & Fournier, K. S. (2017a). Financial time series forecasting-a deep learning approach. *International Journal of Machine Learning and Computing*, *7*(5), 118–122.

Dingli, A., & Fournier, K. S. (2017). Financial time series forecasting-a deep learning approach. *International Journal of Machine Learning and Computing*, *7*(5), 118-122.

Dixit, M., Tiwari, A., Pathak, H., & Astya, R. (2018). An overview of deep learning architectures, libraries and its applications areas. In *2018 International Conference on Advances in Computing, Communication Control and Networking*, 293-297.

DSE Achieves Frontier Market Status, Country Classification by FTSE Russell | Dar es Salaam Stock Exchange PLC. (2019). https://dse.co.tz/content/dse-achieves-frontier-market-status-country-classification-ftse-russell.

Endeleza Project. (2021). https://endeleza.dse.co.tz.

FinSights Lab, F. (2020). *Final Report: Understanding the current state of participation in Tanzania's capital markets*.

Ghanavati, M., Wong, R. K., Chen, F., Wang, Y., & Fong, S. (2016). A Generic Service Framework for Stock Market Prediction. *2016 IEEE International Conference on Services Computing*, 283–290. https://doi.org/10.1109/SCC.2016.44.

Graves, A., & Schmidhuber, J. (2005, July). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 4*, 2047-2052.

Historical Background | Dar es Salaam Stock Exchange PLC. (2020). https://dse.co.tz/content/historical-background.

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, *4*(1), 9.

Huang, C. F., & Li, H. C. (2017). An Evolutionary Method for Financial Forecasting in Microscopic High-Speed Trading Environment. *Computational Intelligence and*

Neuroscience, *2017*, 1–18. https://doi.org/10.1155/2017/9580815.

Huang, Q., Kong, Z., Li, Y., Yang, J., & Li, X. (2018). Discovery of trading points based on Bayesian modeling of trading rules. *World Wide Web*, *21*(6), 1473–1490.

Jain, T., & Singh, S. P. (2022). Impact of Domestic Institutional Investors on Foreign Institutional Investors in India: An Analysis. *Management Dynamics*, *19*(2), 15–33. https://doi.org/10.57198/2583-4932.1020.

Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, *184*, 115537.

Kim, K. A., Kitsabunnarat, P., & Nofsinger, J. R. (2004). Ownership and operating performance in an emerging market: Evidence from Thai IPO firms. *Journal of Corporate Finance*, *10*(3), 355–381.

Kim, Y., & Enke, D. (2016). Developing a rule change trading system for the futures market using rough set analysis. *Expert Systems with Applications*, *59*, 165–173. https://doi.org/10.1016/j.eswa.2016.04.031.

Kothari, C. R. (2004). *Research methodology*. https://ccsuniversity.ac.in/bridge-library/pdf/Research-Methodology-CR-Kothari.pdf.

Labiad, B., Berrado, A., & Benabbou, L. (2018). Short term prediction framework for moroccan stock market using artificial neural networks. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Application*, 1-6.

Lakshminarayanan, S. K., & McCrae, J. P. (2019). A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction. *Artificial Intelligence and Cognitive Science*, 446-457.

Li, K. (Yi). (2022). *Vanishing and Exploding Gradients in Neural Network Models: Debugging, Monitoring, and Fixing*. Neptune.Ai. https://neptune.ai/blog/vanishing-and-exploding-gradients-debugging-monitoring-fixing.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. https://doi.org/10.48550/arXiv.1506.00019.

Lotto, J., & Mmari, C. T. (2018). Domestic debt and economic growth in Tanzania. *Journal of Economics and Management Sciences*, *1*(1), 207-207.

Loye. (2019). *Gated Recurrent Unit (GRU) With PyTorch*. FloydHub Blog. https://blog.floydhub.com/gru-with-pytorch/

Manishimwe, A., Alexander, H., Kaluuma, H., & Dida, M. (2021). Integrated mobile application based on machine learning for East Africa stock market. J*ournal of Information Systems Engineering and Management, 6*(3), 1-7.

Marcopolis. (2020). *Dar es Salaam Stock Exchange: Discussing Capital Markets in Tanzania with Moremi Marwa*. Marcopolis. https://marcopolis.net/dar-es-salaam-stock-exchange-discussing-capital-markets-in-tanzania-with-moremi-marwa.htm.

Massele, J., Darroux, C., Jonathani, H., & Fengju, X. (2015). Challenges faced by Dar-es-Salaam stock exchange market in Tanzania. *Research Journal of Finance and Accounting, 4*(15), 36-42.

Moradi, Z. S., Mirzaeenejad, M., & Geraeenejad, G. (2016). Effect of bank-based or market-based financial systems on income distribution in selected countries. *Procedia Economics and Finance*, *36*, 510–521.

Munisi, G. H. (2019). Financial performance of initial public offerings: Companies listed on Dares Salaam Stock Exchange. *Banking, Economics and Business Research*, 20-27.

Nairobi Securities Exchange. (2022). *Nairobi Securities Exchange PLC*. https://www.nse.co.ke.

Nandakumar, R. (2018). *Stock Price Prediction Using Long Short Term Memory*. *05*(03), 7.

Nayak, A. (2020). *Predicting Stock Price with LSTM*. Medium. https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944.

Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In *2017 International Joint Conference on Neural Networks,*1419-1426.

Nguyen, D. H. D., Tran, L. P., & Nguyen, V. (2019). Predicting stock prices using dynamic

LSTM models. In *Applied Informatics: Second International Conference, International Conference on Artificial Intelligence 2019, Madrid, Spain,* 199-212.

Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques. *2nd International Conference on Finance, Economics, Management and IT Business, Vienna House Diplomat Prague, Prague, Czech Republic*, 63-71.

Olah, C. (2015). Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs.

Parray, I. R., Khurana, S. S., Kumar, M., & Altalbe, A. A. (2020). Time series data analysis of stock price movement using machine learning techniques. *Soft Computing*, *24*(21), 16509–16517.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 1310-1318.

Pykes, K. (2020, May 17). *The Vanishing/Exploding Gradient Problem in Deep Neural Networks*. Medium. https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11.

Raubenheimer, H. (2019). *African capital markets: Challenges and opportunities*.

Raveendran, R. (2019). Listed Companies in NSE | List of Stocks 2022. *IndianCompanies.In*. https://indiancompanies.in/listed-companies-in-nse-with-symbol.

Ren, R., Wu, D. D., & Liu, T. (2019). Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE Systems Journal*, *13*(1), 760–770. https://doi.org/10.1109/JSYST.2018.2794462.

Salimath, S., Chatterjee, T., Mathai, T., Kamble, P., & Kolhekar, M. (2021). Prediction of Stock Price for Indian Stock Market: A Comparative Study Using LSTM and GRU. In *Advances in Computing and Data Sciences: 5th International Conference on Advances in Computing and Data Sciences 2021, Nashik, India, April 23–24, 2021, Revised Selected Papers, Part II 5,* 292-302.

Samarawickrama, A. J. P., & Fernando, T. G. I. (2017). A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. *2017 IEEE International Conference on Industrial and Information Systems*, 1–6. https://doi.org/10.1109/ICIINFS.2017.8300345.

Saxena, S. (2021). LSTM | Introduction to LSTM | Long Short Term Memor. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm.

Schiereck, D., Freytag, A., Grimm, M., & Bretschneider, W. H. (2018). *Public corporations in Africa: A continental survey on stock exchanges and capital markets performance* (No. 2018-013). Jena Economic Research Papers.

Sen, J., Mehtab, S., Dutta, A., & Mondal, S. (2021). *Precise Stock Price Prediction for Robust and Optimized Portfolio Design Using an LSTM Model*. https://arxiv.org/abs/2203.01326.

Singh, P. (2022). *How to use Google Colab for Machine Learning Projects*. Naukri Learning. https://www.naukri.com/learning/articles/how-to-use-google-colab-for-machine-learning-projects.

Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, *19*(9), 1449–1459.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. https://doi.org/10.48550/arXiv.1909.09586.

Valacich, J. (2017). *Information Systems Today: Managing the Digital World*. Pearson. https://lib.hpu.edu.vn/handle/123456789/34512.

Wadekar, S. (2021). Hyperparameter Tuning in Keras: TensorFlow 2: With Keras Tuner: RandomSearch, Hyperband, Bayesian Optimization. *The Startup*.

Yanhui, C. (2021). *A Battle Against Amnesia: A Brief History and Introduction of Recurrent Neural Networks*. Medium. https://towardsdatascience.com/a-battle-against-amnesia-a-brief-history-and-introduction-of-recurrent-neural-networks-50496aae6740.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Zvornicanin, E. (2022). *Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science*. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm.

APPENDICES

Appendix 1: Focus Group Discussion Guiding Questions

1. What are the factors contributing to stock price fluctuations?
2. Do you use historical stock data to anticipate stock price changes?
3. Based on DSE daily transaction data, what attributes can be used to infer next day closing price?
4. What other information (attributes) can be considered in forecasting the closing price?
5. What is the status of DSE in terms of general activeness?
6. How about the ability of listed companies to grow (increase in value) and meet shareholders' expectations?
7. How do you determine whether a company's shares are active or not?
8. Do you have a model or system to predict stock prices?
9. What functionalities do you expect in a system for predicting stock prices?
10. Do you think the system will be beneficial in your organization? In what ways?

Appendix 2: Prototype Validation Questionnaire

Kindly rate each criteria by ticking (✓) where appropriate.

Evaluation Criteria	**Very High**	**High**	**Average**	**Low**	**Very Low**
Usability of the system					
Consistency of the system					
Ability of the system to provide feedback					
Ability of the system to predict the closing price					
Ability of the system to visualize market statistics					
Usefulness of the system					

Appendix 3: Code for development of GRU model

```
# Importing libraries
import numpy as np
import random as python_random
import os
import tensorflow as tf

# To get reproducible results.
seed_value= 0

# Setting the `PYTHONHASHSEED` environment variable at a fixed value
os.environ['PYTHONHASHSEED']=str(seed_value)

# The below is necessary for starting Numpy generated random numbers
# in a well-defined initial state.
np.random.seed(seed_value)

# The below is necessary for starting core Python generated random numbers
# in a well-defined state.
python_random.seed(seed_value)

# The below set_seed() will make random number generation
# in the TensorFlow backend have a well-defined initial state.
tf.random.set_seed(seed_value)

# Configuring a new global tensorflow session
session_conf = tf.compat.v1.ConfigProto(intra_op_parallelism_threads=1, inter_op_parallelism_threads=1)
sess = tf.compat.v1.Session(graph=tf.compat.v1.get_default_graph(), config=session_conf)
tf.compat.v1.keras.backend.set_session(sess)
# More libraries
import pandas as pd
import matplotlib.pyplot as plt

# Get Google-Drive access so as to load data from Google-Sheet
from google.colab import auth
from google.auth import default
import gspread
auth.authenticate_user()
creds, _ = default()
gc = gspread.authorize(creds)

# Load the data
worksheet = gc.open('daily_dse').sheet1
df = pd.DataFrame(worksheet.get_all_records())
df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%Y')
df.head()

# We replace empty string(s) with NaN, then we use dropna to drop them
df = df.replace(',','', regex=True)
df = df.replace(r'^\s*$', np.NaN, regex=True)
df = df.replace(r'#VALUE!', np.NaN, regex=True)
df = df.replace(r'#N/A', np.NaN, regex=True)
df = df.replace(r'#DIV/0!', np.NaN, regex=True)
# then we use dropna to drop rows with NaN
df = df.dropna(axis='index', how='any')

df['Price Change'] = df['Closing Price'] - df['Opening Price']
print('Shape: ', df.shape)

# Datatypes: All columns are supposed to be float except the 'Company Name' column
for column in df.columns.tolist():
    if column ==  'Company Name':
        df[column] = df[column].astype(str)
    elif column == 'Date':
        df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
    else:
        df[column] = df[column].astype('float64')
```

```python
print(df.dtypes)

df = df.sort_values(by=['Date'], ascending=True)
df = df.drop(['Date'], axis='columns')

# Select active companies
selected_companies = ['CRDB', 'NMB', 'TPCC', 'SWIS', 'DSE', 'TOL', 'TCCL', 'NICO']]
data = df[df['Company Name'].isin(selected_companies)]
print(data.shape)

# Scale the feature data
from sklearn.preprocessing import MinMaxScaler
data_to_scale = data.drop(['Company Name'], axis='columns')
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data_to_scale.values)
data.head()

# Creating a function to trim the dataset to size that is divisible by batch_size
batch_size = 64
def trim_dataset(arr):
    no_of_rows_drop = arr.shape[0] % batch_size
    if no_of_rows_drop > 0:
        return arr[:-no_of_rows_drop]
    else:
        return arr

# Supervised learning setup
import math
timesteps = 30
n_features = data.drop('Company Name', axis='columns').shape[1]
X_train = np.empty(shape=(0, timesteps, n_features))
X_test = np.empty(shape=(0, timesteps, n_features))
y_train, y_test = [], []
target_index = data.drop('Company Name', axis='columns').columns.get_loc('Closing Price')
for company_name in selected_companies:
  company_data = data.loc[data['Company Name'] == company_name]
  company_data = company_data.drop(['Company Name'], axis='columns')
  # total number of time-series samples would be length minus timesteps
  dim_0 = company_data.shape[0] - timesteps
  x = np.zeros((dim_0, timesteps, n_features))
  y = np.zeros((dim_0, 1))
  company_data = scaler.transform(company_data.to_numpy())
  for i in range(dim_0):
    x[i] = company_data[i : timesteps + i]
    y[i] = company_data[timesteps + i, target_index]
  train_len = math.ceil(0.8 * len(x))
  train_x, train_y = x[0:train_len, :], y[0:train_len, 0]
  # trim to size divisible by batch-size
  train_x, train_y = trim_dataset(train_x), trim_dataset(train_y)
  y_train = np.append(y_train, train_y)
  X_train = np.append(X_train, train_x, axis=0)
  test_x, test_y = x[train_len: , :], y[train_len:, 0]
  test_x, test_y = trim_dataset(test_x), trim_dataset(test_y)
  y_test = np.append(y_test, test_y)
  X_test = np.append(X_test, test_x, axis=0)
# Reshape the target variable
y_train = np.reshape(y_train, (-1, 1))
y_test = np.reshape(y_test, (-1, 1))
print('y_train shape:', y_train.shape)
print('y_test shape:', y_test.shape)
print('X_train shape:', X_train.shape)
print('X_test shape:', X_test.shape)

# Model construction: importing libraries
from keras.models import Sequential
from keras.layers import GRU
from keras.layers import Dense
from keras.callbacks import EarlyStopping
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_percentage_error
from math import sqrt
import time

# Function to calculate adjusted_r2
def adjusted_r2(y_true, y_pred):
  n = len(y_test) + len(y_train)
  return 1 - (1-r2_score(y_true, y_pred)) * (n-1)/(n-X_train.shape[2]-1)

# Tuning
!pip install keras-tuner --upgrade

import keras_tuner
class MyOptimalModel(keras_tuner.HyperModel):
  def build(self, hp):
    model = Sequential()
    # Now checking with two hidden layers
    hp_units1 = hp.Int('units1', min_value=20, max_value=300, step=30)
    hp_units2 = hp.Int('units2', min_value=20, max_value=300, step=30)
    model.add(LSTM(units=hp_units1, input_shape=(X_train.shape[1], X_train.shape[2]),
              activation='tanh', return_sequences=True))
    model.add(LSTM(units=hp_units2, activation='tanh', return_sequences=False))
    model.add(Dense(1))
    model.compile(optimizer="adam", loss="mse")
    return model
  def fit(self, hp, model, *args, **kwargs):
    return model.fit(*args, batch_size=hp.Choice("batch_size", [16, 32, 64, 128]), **kwargs)
tuner = keras_tuner.BayesianOptimization(MyOptimalModel(), objective="val_loss", max_trials=100)
tuner.search(X_train, y_train, epochs=500, validation_data=(X_test, y_test),
             callbacks=[EarlyStopping(monitor='val_loss', patience=5)],
             verbose=0, shuffle=False)
# Print best hyperparameters
for h_param in [f"units{i}" for i in range(1,3)] + ["batch_size"]:
  print(h_param, tuner.get_best_hyperparameters()[0].get(h_param))

# Print best epoch (train for 500 epochs)
model = Sequential()
model.add(GRU(290, input_shape=(X_train.shape[1], X_train.shape[2]), activation='tanh',
              return_sequences=True))
model.add(GRU(140, activation='tanh', return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train, epochs=500, batch_size=batch_size,
                    validation_data=(X_test, y_test), verbose=0, shuffle=False)
val_loss_per_epoch = history.history['val_loss']
best_epoch = val_loss_per_epoch.index(min(val_loss_per_epoch)) + 1
print('Best epoch: %d\n' % (best_epoch,))

# Print best epoch (train for 1000 epochs)
model = Sequential()
model.add(GRU(290, input_shape=(X_train.shape[1], X_train.shape[2]), activation='tanh',
              return_sequences=True))
model.add(GRU(140, activation='tanh', return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train, epochs=1000, batch_size=batch_size,
                    validation_data=(X_test, y_test), verbose=0, shuffle=False)
val_loss_per_epoch = history.history['val_loss']
best_epoch = val_loss_per_epoch.index(min(val_loss_per_epoch)) + 1
print('Best epoch: %d\n' % (best_epoch,))

# Fit the best model
model = Sequential()
model.add(GRU(290, input_shape=(X_train.shape[1], X_train.shape[2]), activation='tanh',
             return_sequences=True))
```

```
model.add(GRU(140, activation='tanh', return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train, epochs=best_epoch, batch_size=batch_size,
                    validation_data=(X_test, y_test), verbose=0, shuffle=False)
# plot history
plt.style.use('fivethirtyeight')
plt.figure(dpi=300, figsize=(16,8))
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
# plt.savefig('experiment1.png', bbox_inches="tight", dpi=300)
plt.show()
# predictions on train
preds = model.predict(X_train)
preds = (preds * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
y =  (y_train * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
# train RMSE
mse = mean_squared_error(y, preds)
print("Train RMSE: %.4f" % sqrt(mse))
print("\n")
# predictions on test
start = time.time()
predictions = model.predict(X_test)
end = time.time()
predictions = (predictions * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
y_actual =  (y_test * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
#RMSE
test_mse = mean_squared_error(y_actual, predictions)
rmse = sqrt(test_mse)
print("Test MSE: %.4f" % test_mse)
print("Test RMSE: %.4f" % rmse)
print("Test MAE: %.4f" % mean_absolute_error(y_actual, predictions))
print("Test R-Squared: %.4f" % r2_score(y_actual, predictions))
print("Test Adj R-Squared: %.4f" % adjusted_r2(y_actual, predictions))
print("Test MAPE: %.4f" % (mean_absolute_percentage_error(y_actual, predictions) * 100))
print("Test NRMSE: %.4f" % (rmse/(np.amax(y_actual)-np.amin(y_actual))))
print("Test Ratio of RMSE to the Mean of actual values: %.4f" % (rmse/np.mean(y_actual)))
print("Prediction time (seconds): %.4f" % (end - start))

result1 = pd.DataFrame({'actual': y_actual.flatten(), 'predictions': predictions.flatten()})

# Build the GRU model
model = Sequential()
model.add(GRU(80, input_shape=(X_train.shape[1], X_train.shape[2]), activation='tanh',
              return_sequences=True))
model.add(GRU(110, activation='tanh', return_sequences=True))
model.add(GRU(50, activation='tanh', return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train, epochs=200, batch_size=batch_size,
                    validation_data=(X_test, y_test), verbose=0, shuffle=False)
# plot history
plt.style.use('fivethirtyeight')
plt.figure(dpi=300, figsize=(16,8))
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
# plt.savefig('experiment1.png', bbox_inches="tight", dpi=300)
plt.show()
# predictions on train
preds = model.predict(X_train)
preds = (preds * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
y =  (y_train * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
```

```
# train RMSE
mse = mean_squared_error(y, preds)
print("Train RMSE: %.4f" % sqrt(mse))
print("\n")
# predictions on test
start = time.time()
predictions = model.predict(X_test)
end = time.time()
predictions = (predictions * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
y_actual =  (y_test * scaler.data_range_[target_index]) + scaler.data_min_[target_index]
#RMSE
test_mse = mean_squared_error(y_actual, predictions)
rmse = sqrt(test_mse)
print("Test MSE: %.4f" % test_mse)
print("Test RMSE: %.4f" % rmse)
print("Test MAE: %.4f" % mean_absolute_error(y_actual, predictions))
print("Test R-Squared: %.4f" % r2_score(y_actual, predictions))
print("Test Adj R-Squared: %.4f" % adjusted_r2(y_actual, predictions))
print("Test MAPE: %.4f" % (mean_absolute_percentage_error(y_actual, predictions) * 100))
print("Test NRMSE: %.4f" % (rmse/(np.amax(y_actual)-np.amin(y_actual))))
print("Test Ratio of RMSE to the Mean of actual values: %.4f" % (rmse/np.mean(y_actual)))
print("Prediction time (seconds): %.4f" % (end - start))

result2 = pd.DataFrame({'actual': y_actual.flatten(), 'predictions': predictions.flatten()})
```

RESEARCH OUTPUTS

(i) Research Paper

Joseph, S., Mduma, N., & Nyambo, N. (2023). A Deep Learning Model for Predicting Stock Prices in Tanzania. *Engineering. Technology & Applied Science Research, 13*(2), 10517–10522.

(ii) Poster Presentation