

2021-03

Ethiopian Agricultural Market Price Data Analysis and Prediction using Deep Learning Models

Ahmed, Kebede

<http://ir.bdu.edu.et/handle/123456789/12397>

Downloaded from DSpace Repository, DSpace Institution's institutional repository



BAHIR DAR UNIVERSITY

**BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND POSTGRADUATE
STUDIES**

FACULTY OF COMPUTING

Master of Science in Software Engineering

MSc Thesis on

**Ethiopian Agricultural Market Price Data Analysis and Prediction
using Deep Learning Models**

By

Ahmed Kebede Kassaw

Bahir Dar, Ethiopia

March, 2021



Ethiopian Agricultural Market Price Data Analysis and Prediction Using Deep Learning Models

By

Ahmed Kebede Kassaw

A Thesis Submitted to the School of Research and Graduate Studies of Bahir Dar

Institute of Technology, BDU in Partial Fulfillment for the Degree of Master of
Science in Software Engineering in the Faculty of Computing

Advisor: Dr. Gebeyehu Belay

Bahir Dar, Ethiopia

March, 2021

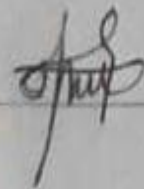
Copywrite ©2021 Ahmed kebede Kassaw

DECLARATION

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and refereed all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation / fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student Ahmed Kebede

Signature



Date of submission: March 2, 2021

Place: Bahir Dar

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF RESEARCH AND GRADUATE STUDIES
COMPUTING FACULTY

Approval of Thesis for Defense Result

I hereby confirm that the change required by the examiners have been carried out and incorporated in the final thesis.

Name of the Student Ahmed Kebede Kassaw Signature [Signature] Date March 2, 2021

As a member of the board examiners, we examined this thesis entitled "Ethiopian Agricultural Market Price Data Analysis and Prediction Using Deep Learning Models" by Ahmed Kebede Kassaw. We hereby certify that the thesis is accepted for fulfilling the requirement for the award of the degree of Master of Science in Software Engineering.

Board of Examiners

Advisor: Gebeyehu Belay (Dr) [Signature] March 2, 2021
Name Signature Date

External Examiner: Fekade Getahun [Signature] March 2, 2021
Name Signature Date

Internal Examiner: Selhi G. (Ass. Prof) [Signature] March 2, 2021
Name Signature Date

Chairperson: Dagnachew Melesse [Signature] March 2, 2021
Name Signature Date

Chair Holder: Gebeyehu B. (Dr) [Signature] March 2, 2021
Name Signature Date

Faculty Dean: Belete B. [Signature] March 2, 2021
Name Signature Date



Acknowledgements

This thesis would not be possible without the help of God (Allah) followed by my family especially my mother and my father. Next, my deepest gratitude goes to my advisor Dr. Gebeyehu Belay, who has supported me by giving immense advice and suggestions starting from title selection and formulation throughout my thesis preparation.

Abstract

Ethiopia Commodity Exchange (ECX) is a commodity exchange established April 2008 and it is a national multi-commodity exchange that brings together buyers and sellers of agricultural commodities. It has been collecting a handful of time series data in each year. An accurate and efficient prediction of such time series data is conducive for evading market risks and supports the agricultural system that realize the balance of supply and demand of agricultural commodities. However, the agricultural commodity market is traditional and the market price data is not used for dynamic market price data analysis and prediction. Many countries proposed and utilizing different prediction mechanism for the future price of their market. However, the research approaches are different and the computational algorithms also based on the real market situation. Moreover, the market features that have impact on one country may not have similar impact in another country. Hence, the intention of this thesis work is analyzing and predicting the target market using deep learning models based on both fundamental and technical market analysis approaches. Four most commonly used deep learning models were explored for their predictive ability of target market price based on both the fundamental and technical indicator features of coffee, sesame and haricot bean and wavelet transform was used for denoising the price data. The technical indicators calculated from price data were given to the Boruta feature selection algorithm and 39 features for Coffee, 24 for Sesame and 27 features for Haricot bean were selected. The sentiment scores were calculated from the financial news related to each commodity. Lastly, the denoised price data together with selected technical indicator and sentiment scores were used as an input to the models. Three experiments were conducted for each commodity; the first was model building and prediction with only the price data. The second was using the price data with commodity technical indicators (CTI) and the third one was using price data, CTI and sentiment scores. From models, (Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), convolutional neural network (CNN) and CNN-LSTM network), the performance of LSTM is shown to be the best fit model with minimum Mean Squared Error (MSE) of 9.6×10^{-5} , which is small enough, and it indicates that the predictive model performs well.

Keywords: Commodity technical indicators; Deep learning; Price prediction.

Table of Contents

DECLARATION	III
Abstract	iv
Acknowledgements.....	iii
List of figures	v
List of Tables	ix
List of Abbreviations	x
CHAPTER ONE	1
INTRODUCTION	1
1. Introduction	1
1.2 Statement of the Problem	2
1.3 Objective	3
1.3.1 General Objective	3
1.3.2 Specific Objectives	3
1.4 Scope and Limitation	3
1.5 Significance of the Study	4
1.6 Thesis Organization.....	4
CHAPTER TWO	5
LITRETURE REVIEW	5
2.1 Market analysis approaches	5
2.1.1 Fundamental Analysis	5
2.1.2 Technical analysis.....	7
2.2 Wavelet-based denoising.....	9
2.3 Feature selection algorithms	10
2.3.1 Boruta Feature Selection Algorithm	10
2.4 Deep learning algorithms	12

2.4.1 Recurrent Neural Network.....	12
2.4.2 Long-short term memory	14
2.4.3 Convolutional Neural Network	18
2.5. Related Works	20
CHAPTER TREE	28
RESEARCH METHODOLOGY	28
3.1 Data collection and preprocessing.....	29
3.1.1 Data Collection	29
3.1.2 Market Price Data preprocessing.....	30
3.1.3 News Headlines Preprocessing.....	36
3.2 Experimental Data.....	40
3.3 Model Selection.....	41
3.4 Metrics.....	42
3.5 Tools and Techniques.....	43
3.6 Evaluation.....	43
CHAPTER FOUR.....	44
EXPERIMENT AND RESULT DISCUSSION	44
4.1. Model Comparison and Selection (RQ1)	45
4.1.1 Recurrent Neural Network Building and Prediction	46
4.1.2 Long-Short Term Memory Model Building and Prediction.....	54
4.1.3 Convolutional Neural Network Model Building and Prediction.....	61
4.1.4 CNN-LSTM Model Building and Prediction	69
4.2 Comparison of Technical and Fundamental Indicators.....	80
CHAPTER FIVE	83
CONCLUSION AND RECOMMENDATION.....	83
5 Conclusion and Recommendation.....	83
Reference	84

List of figures

Figure 2.1. Feature selection steps of Boruta algorithm	11
Figure 2.2. The Basic architecture of Vanilla RNN.....	13
Figure 2.3. Unfolded RNN architecture.....	13
Figure 2.4. Typical LSTM network	15
Figure 2.4a. Forget gate of LSTM cell	15
Figure 2.4b. Input gate of LSTM network	16
Figure 2.4c. Output gate of LSTM network	17
Figure 2.5. Convolution of input vector Y using filter Z.....	18
Figure 2.6: The convolution operation.....	19
Figure 3.1 Research Methodology	28
figure 3.2. Filter Bank Scheme for Wavelet Transform	31
Figure 3.3 Wavelet decomposition (a) Approximation (b) Detailed using symlet 16.....	32
Figure 3.4 a & b Shows close and denoised signal of the closing price	33
Figure 3.4(c) Shows denoised and original closing price data	33
Figure 3.5 OHCL plus 83 technical indicators	34
Figure 3.6 Show Boruta feature selection on technical indicator	36
Figure 3.7 shows news headlines preprocessing.....	38
Figure 3.7 shows preprocessed news dataset.....	39
Figure 4.1 Shows the RNN model structure we build during our experiment	46
Figure 4.2a shows the validation and training loss of RNN for coffee.....	47
Figure 4.3 Shows the predictive and real price for coffee using RNN	49
Figure 4.4 Show the training and validation loss of RNN for sesame.....	50
Figure 4.5 Shows the prediction result of RNN for Sesame price.....	51
Figure 4.6 shows the train and validation loss of RNN model on haricot	52
Figure 4.7 Shows the prediction of haricot bean price using RNN	53
Figure 4.8 shows the build LSTM model	54
Figure 4.9 shows the training and validation loss of LSTM.....	55
Figure 4.10 Shows the prediction result of LSTM	56
Figure 4.11 Shows training and validation loss of LSTM model on Sesame.....	57

Figure 4.12 shows the prediction of Sesame price using LSTM model	59
Figure 4.13 Shows the training and validation loss LSTM on Haricot Beans.....	60
Figure 4.14 shows the prediction result of LSTM trained on Haricot beans.....	61
Figure 4.15 Shows the basic CNN model created	62
Figure 4.16 shows the training and validation loss of CNN for coffee	63
Figure 4.17 shows the predicted and real values of coffee using CNN.....	64
Figure 4.18 Shows the training and validation loss of Sesame using CNN	65
Figure 4.19 show the predicted and real values of sesame using CNN.....	66
Figure 4.20 show the training and validation losses of CNN for Haricot Bean	68
Figure 4.21 Shows the predicted and real value of haricot bean using CNN	69
Figure 4.22 Show basic CNN-LSTM model	70
Figure 4.23 shows the training and validation loss of CNN-LSTM on coffee price	71
Figure 4.24 Show the predicted and real of coffee dataset using CNN-LSTM.....	72
Figure 2.25 train and validation loss of CNN-LSTM on sesame	73
Figure 4.26 shows the predicted and real dataset of sesame using CNN-LSTM	74
Figure 4.27 Train and Validation loss of CNN-LSTM on haricot bean dataset	75
Figure 4.28 predicted and real dataset of Haricot bean using CNN-LSTM	76
Figure 4.29 show the prediction of coffee with CTI and News data using LSTM.....	80
Figure 4.30 shows prediction of sesame with CTI and News.....	81
Figure 4.31 Shows prediction of Haricot beans price using with news and with CTI separately	81

List of Tables

Table 2.1. Article review summary.....	25
Table 3.1 show the raw market price data recorded for coffee price.....	29
Table 3.2 coffee price after feature selection.....	35
Table 3.3 show some examples of news headline that we collected.	37
Table 4.1 Shows performance measures of four models on the three commodities.....	77
Table 4.2 shows the mean squared(MSE) error incurred during the experiments.....	79
Table 4.3 shows MSE incurred for each input features	82

List of Abbreviations

ANN	Artificial Neural network
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
CCI	Commodity Channel Index
CNN	Convolutional Neural Network
CTIs	Commodity Technical Indicators
DA	Directional Accuracy
DOHCL	Denoised Open High Low Close
DWT	Discrete Wavelet Transform
ECX	Ethiopian Commodity Exchange
EMA	Exponential Moving Average
FCF	Free Cash Flow
GDP	Gross Domestic Product
KNN	K-nearest Neighbors
LSTM	Long Short-Term Memory
MAE	Mean Average Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NLP	Natural Language Processing
NSS	News Sentiment Score
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROE	Return on equity
RSI	Relative Strength Index
SVM	Support Vector Machine
SMA	Simple Moving Average
TIs	Technical Indicators
TF- IDF	Term Frequency Inverse Document Frequency
VADER	Valence Aware Dictionary and Sentiment Reasoned

CHAPTER ONE

INTRODUCTION

1. Introduction

Future Market data analysis and prediction using different computational algorithm is regularly practices in modern marketing systems (Ghosh, Bose, Maji, & Debnath, 2019; Mehtab, 2020; Yu & Yan, 2020). Accurate future prediction method is conducive for evading market risk and uncertainty. It is also pertinent to support agricultural commodities, balance the supply and demand of agricultural commodities, and provide decision making for government to adjust the economic structure (Y. Zhang, 2018). However, in Ethiopian agricultural commodities market price analysis and future price prediction is a challenge, and the marketing transaction is traditional.

There are three main approaches related to the information required to make a market prediction. The first approach is technical analysis which evaluates the commodity market based on charts and trends and predicts the future price of the stock. The second is fundamental analysis which evaluates the commodity market based on external information as political and economic factors taken from unstructured data as news articles, financial reports, or others affecting the business to analyze the market value of its share. The third approach considers all relevant information coming from both, financial time series and textual data. Technical analysis and Fundamental analysis are the two main schools of thought that finance practitioners and analysts subscribe to when making trading decisions and future commodity market price prediction (Beyaz, Tekiner, Zeng, & Keane, 2020).

Many developed and developing countries proposed and utilizing different prediction mechanism for the future price of their agricultural market (Nhita, et al., 2019; Ouyang, Wei, & Wu, 2019; Rajeswari & Suthendran, 2019; Rikumahu, 2019; Varun et al., 2019; Y. Zhang, 2018). However, there is a gape in a direct implementation of their findings. The research approaches also different and computational algorithms are based on the real market situation. Moreover, the market features that have impact on one country may not

have similar impact in another country (Bulti & Ray, 2019). The same is true for Ethiopia that lacked published and automated agricultural future price prediction method, and yet the agricultural market process is in a traditional manner. As a result, we proposed deep learning technologies to the target market to amend the agricultural market strategies.

This thesis explores the predictability of Ethiopian agricultural commodity future price based on data that is derived from both technical and fundamental market analysis. The study aims to explore different deep learning algorithms that are best fit in the prediction of target market price and price trend. Besides, it identifies market determinant features that impact Ethiopian market price. Wavelet transform can be used to de-noise the data and, finally based on model's prediction performance, selection and recommendation of the best fit model can be done in the prediction of Ethiopian agricultural market price and price trend.

1.2 Statement of the Problem

Currently, agriculture is the backbone of Ethiopia's economy, however, its agricultural market is conducted in a customary way. This is due to a lack of technological development and historical market data are still not used for the prediction of commodity market price. Although, a handful of market price data have been collected by both governmental and non-governmental organizations in each year, yet little have been done to analyze the data and not used for future market price prediction. Investors, farmers and traders start their business without appropriate current and future market information, which leads to high risk. In addition, agricultural market data analyses performed so far are still manual, which is time taking and error prone.

Berhane et al., (2018) and Muanenda & Yohannes, (2018) touched this area when they applied Kalman filtering for coffee price and machine learning frameworks for Ethiopian market price prediction respectively. However, the proposed models are statistical and traditional machine learning which needs domain experts for feature extraction (Xu B. et al., 2018). It means that statistical and traditional predictive models cannot extract important feature by themselves rather the determinant market features for the given predictive model is determined by the feature extraction human experts. Hence, the study

aims to explore the predictability of the Ethiopian agricultural commodity market and efficient data analysis as well as prediction models using deep learning algorithms. Therefore, the research issues are formulated as follows:

- ❖ Which deep learning model is the best fit model in the prediction of Ethiopian agricultural Commodity market price?
- ❖ What market analysis approach is determinant in the target market prediction?
- ❖ Which is the best way to use the two main market analysis approaches in the market price prediction?

1.3 Objective

1.3.1 General Objective

The objective of this thesis is analyzing the existing Ethiopian agricultural commodity market price data and forecasting its market price using Deep learning models based on both fundamental and technical market analysis approaches.

1.3.2 Specific Objectives

- ❖ Analyze the existing problem by reviewing different research papers.
- ❖ Select and prioritize the determinant market analysis features in forecasting the Ethiopian agricultural market price.
- ❖ Develop market price predictive model using Deep Learning Techniques
- ❖ Identifying the best fit model, from the selected deep learning algorithm, in analyzing and forecasting the Ethiopian agricultural commodity market price.
- ❖ Using selected models, evaluate and prioritize the individual and combined contribution of the two market analysis approaches for market price prediction.

1.4 Scope and Limitation

Nowadays, the Ethiopian commodity exchange is hosting a large number of market commodities and started to gather a large amount of commodity market price data on the daily basis. Such kinds of valuable and organized data support to analyze the past well and forecast the future price of the Ethiopian agricultural commodity market with greater

accuracy, minimum market risk and uncertainty. However, due to its broadness, it is not possible for us to cover all of the market sectors and we focused only on agricultural commodities goods that ECX collects namely Coffee, Sesame and haricot bean. In addition, we focused only on prediction of the closing price of the market based on historical price data and fundamental analysis of the market, we did not include the Amharic news article due to the shortage of time. Governmental policy change, natural disaster and un instability factors are not incorporated in this thesis work.

1.5 Significance of the Study

It is well known that above 80% of Ethiopians are agrarians and most of the industries are directly or indirectly dependent on agricultural outcomes. In addition, in the Ethiopia almost all export items are agricultural commodities. Predicting the price of agricultural commodities such as vegetables, fruits, cereals etc. is very important for economists, farmers, agriculturist's, and industrialists. Market price prediction helps the entire user of agriculture to take an effective decision regarding the selling price of their crop, and also helps policy makers for agricultural decisions.

1.6 Thesis Organization

The following sections include a description of the literature review in Chapter 2; a proposed methodology followed for the study in Chapter 3; experiment and result discussion in Chapter 4; conclusion and recommendation of the study in Chapter 5.

CHAPTER TWO

LITRETURE REVIEW

This section will explain the two market analysis approaches; technical indicators and fundamental indicators and also what deep learning is and popular algorithms used by previous researchers to predict commodity market prices. This will provide a background of the technologies we use as part of this research. In addition, we will see a brief overview of the papers that served as a guide to our work.

2.1 Market analysis approaches

Commodity markets analysis and prediction is one of the very interesting and most fascinating inventions of our time due to its impact on business, education, jobs, technology and thus on the economy (Hiransha et al. 2018). Analyzing stock market movements and price behaviors is extremely challenging because of the markets dynamic, nonlinear, non-stationary, nonparametric, noisy, and chaotic nature of the market. Agricultural commodity markets can be affected by many highly interrelated factors that such as economic and political factors (Zhong and Enke, 2017). Technical and fundamental analysis are the two main market analysis approaches to analyze the financial markets (R & Christie, 2018).

2.1.1 Fundamental Analysis

Fundamental analysis involves carefully analyzing the market at its most basic or fundamental financial level. Its main goal is to assess the business model, financial statements, profit margin, and other key indicators to determine the financial situation of a market and the intrinsic value of its stock. Matt Krantz, (2016) discussed some of fundamental indicators calculated during company analysis which include:

Earnings per Share (EPS): This is the portion of a company's profit that is assigned to each share of its stock. It's essentially the bottom line of net income, just on a per-share basis.

$$\text{EPS is calculated as } \frac{\text{profit of the company}}{\text{market share}} \quad (1)$$

Price to Earnings Ratio (P/E): The price-to-earnings ratio measures the relationship between the stock price of a company and its per-share earnings. It helps traders determine if a commodity market price is oversold or overbought relative to others in the same sector.

Projected Earnings Growth (PEG): To calculate Projected Earnings Growth, divide the P/E ratio by the company's 12-month growth rate.

$$\text{PEG} = \text{PE Ratio} / \text{Earnings Growth} \quad (2)$$

Free Cash Flow: it is the result of net operating profit after tax of a company added with its depreciation minus the working capital of the company. Cash is essential to the sustenance and amelioration of a market. FCF is calculated as:

$$\text{FCF} = \text{Depreciation} - \text{Working capital}. \quad (3)$$

Price to Book Ratio (P/B): the price-to-book ratio is a fundamental analysis indicator that compares the book value of a stock to its market value. P/B can be calculated by dividing the most recent closing price of a commodity by the book value per share as listed in the company's annual report. Book value is the difference of all assets and liabilities. It is the theoretical value of a company.

Return on Equity (ROE): It measures how well a company generates positive returns to its shareholders' investment and calculated as:

$$\text{ROE} = \text{Profit Margin} \times \text{Turnover Ratio} \times \text{Leverage} \quad (4)$$

Dividend Payout Ratio (DPR): It shows a portion of net income a company returns to its shareholders and how much it sets aside for growth, cash reserve, and debt repayments. It can be obtained by dividing the total dividend amount by the company's net income in the same trading period.

Price to Sales Ratio (P/S): The price-to-sales ratio is a fundamental analysis indicator that can help determine the fair value of a stock by utilizing a company's market capitalization and revenue.

Dividend Yield Ratio: it tells how much they are getting back from every dollar they've invested in the company's stock market.

2.1.2 Technical analysis

Technical analysis uses the market historical prices data of a company and trading volume information in both deciding what the stock price will be and making a trading decision (Beyaz et al., 2020). Technical indicators can be derived from the historical data based on mathematical equations that produce a value that is then plotted on your chart. R & Christie, (2018) discussed some of the important technical indicators employed during technical analysis which include:

Simple moving average: is the average price of a given period, with equal weighting given to the price of each period. It can be calculated as:

$$SMA = (\text{Sum (Price, } n))/n \quad (5)$$

Where n is time period.

Exponential moving average: a moving average that gives more weight to the latest data. The model uses 8-day, 20-day and 200-day EMAs. It is calculated by taking the average of the past price as described in equation 5.

$$EMA = k(Pc - EMA(t - 1)) + EMA(t - 1) \quad (6)$$

$$k = \frac{2}{1+n} \quad (7)$$

Where n is a period of EMA and Pc is closing price of current day.

Moving Average Convergence/Divergence: a trend-following indicator. MACD can be obtained by subtracting the 26-day of EMA from the 12-day of EMA. The 9-day EMA of the MACD is used as a signal line. The first component of MACD signal is MACD which can be obtained from the 9-days and 12-days EMA values as shown in equation 8, the second is the signal which is 9-day EMA of MACD and the difference of MACD and signals gives MACD histogram as shown in equation 9 and 10 respectively.

$$MACD = EMA(12) - EMA(26) \quad (8)$$

$$signal = EMA(MACD, 9) \quad (9)$$

$$Histogram = MACD - signal \quad (10)$$

Commodity channel index: used to determine whether a market is overbought or oversold. It assesses the relationship between asset price, its moving average and deviations from that average. CCI can be calculated as shown in equation 11.

$$CCI = \frac{Pt - SMA(Pt)}{0.015 \sigma_{mad}(Pt)} \quad (11)$$

$$Pt = \frac{Ph + Pl + pc}{3} \quad (12)$$

Where σ_{mad} is the mean absolute deviation of the price t. Ph., Pl and Pc are high, low and close price respectively.

Rate of change: measures the percentage change in price between the current price and the price a certain number of periods ago and it can be calculated as follows.

$$ROC = [(C_{pt} - C_{pn}) / C_{pn}] \quad (13)$$

Where C_{pt} and C_{pn} are closing price today and closing price n days ago respectively.

Relative strength index: it compares the size of recent profits and recent losses. it is intended to reveal the strength or weakness of a market price trend from a range of closing price for a given period of time and is can be calculated as shown in equation 14.

$$RSI = 100 - \frac{100}{1 + U/D} \quad (14)$$

Where U and D, are the average value of positive and negative price change over n days respectively.

Momentum (MO): it compares where the current price is in relation to where the price was in the past can be calculated as shown in equation (15).

$$MO = Pc - Pp \quad (15)$$

Where P_c is the current price and P_p is the past price of the stock.

2.2 Wavelet-based denoising

Due to the existence of the high noise in market price data, which is inevitable for the deep neural networks trained using the original data to accurately predict the market price, and to address this issue wavelet transform is the best option (Liang, Ge, Sun, He, & Chen, 2019). Wavelet transformation is an upgraded version of Fourier transformation which can serve as a very good filter to decrease the noise and smooth out the commodity market data and helps us to focus more on the main trend of the market. It is a time-frequency decomposition that decomposes the time-series data in both time and frequency domains even if the data is non-stationary. It has achieved many successful applications in fields of feature engineering, such as signal processing, scientific calculation, image processing (Z. Li & Tam, 2017; M. Zhang et al., 2020).

The simplest idea of wavelet denoising model is presented as (16):

$$f(t) = x(t) + \varepsilon(t) \quad (16)$$

Where $f(t)$ is the observed signal, $x(t)$ is the real signal and $\varepsilon(t)$ is the centered Gaussian white noise.

The purpose of wavelet denoising is to filter out $\varepsilon(t)$ as far as possible. Similarly, we assume that the Market price data consists of a major trend like the real signal and fluctuations like noise, as shown in equation 16. Moreover, we suppose that the major trend is predictable and the fluctuations are unpredictable. Therefore, the basic idea is to capture the major trend and filter out the noise. We focus on the prediction of the major trend only.

$$f(n) = x(n) + \varepsilon(n) \quad (18)$$

Where $f(n)$ is the observed market data, $x(n)$ is the predictable major smooth trend and $\varepsilon(n)$ is the unpredictable fluctuations.

2.3 Feature selection algorithms

Due to the complicated nature of deep neural networks (DNNs) and other deep learning methods, they have been mostly treated as black box tools (Lu & Noble, 2018.). In many scientific areas, interpretability of scientific results is becoming increasingly important, as researchers aim to understand why and how deep learning models make a certain decision (Pilnenskiy & Smetannikov, 2020). For instance, if a financial transaction is flagged to be fraudulent, then the security teams want to know which activities or behaviors led to the flagging. Therefore, we should have a predictive mode which is explainable and interpretable to reason out why certain decisions are made. It helps us convince domain experts. Therefore, to come up with an interpretable and explainable predictive model, feature selection plays a vital role.

Feature selection is a dimensionality reduction process that focuses on reducing the number of features when developing a predictive model. It is desirable to reduce the number of input variables to those that are believed to be most useful to a model for reducing the computational cost of modeling, improve the model performance and to predict the target variable (Long, Lu, & Cui, 2019). It is one of the apparent tasks in the data analytic research where the datasets have a large number of variables to reduce the dimensionality and to increase the explain ability and interpretability of the predictive model.

There are four fundamental reasons why feature selection is needed. The first reason is to simplify the model by reducing the number of parameters. The second reason is to minimize the training time of the predictive model. third, to reduce overfitting by enhancing generalization, and the last is to avoid the curse of dimensionality (Z. Chen et al., 2020). To this end, we planned to use feature selection technics to make our predictive model interpretable, explainable and improve the prediction accuracy.

2.3.1 Boruta Feature Selection Algorithm

The Boruta feature selection algorithm is a wrapper method algorithm built around the random forest classification algorithm. It follows all-relevant selection and tries to capture all the important, interesting features you might have in your dataset (Maya Gopal P.S,

Bhargavi R, 2018). What makes a Boruta feature selection so special are: it works for both classification and regression problems, it takes into account multi-variable relationships, it is an improvement on random forest variable importance measure which is a very popular method for feature selection, it considers all variable which is relevant to the outcome variable whereas most of the other feature selection algorithms follows a minimal optimal method where they rely on a small subset of features, it can handle the interaction between variables and fluctuating nature of a random forest importance measure.

The Boruta algorithm follows the following core steps during feature selection.

- Firstly, the dataset is expanded by adding copies of all variables
- The added attributes are shuffled randomly to remove any correlation with the response variable
- Random forest is run on the whole dataset and Z-score is computed for all attributes
- From all the shadow attributes, find the one with the maximum Z score and assign a hit to every attribute that scored higher than the one with the maximum Z-score.
- For each attribute with undetermined importance perform a two-sided test of equality with the one obtained for shadow attribute with maximum Z-score.
- Mark attributes which have importance significantly lower than the shadow with maximum Z-score as ‘unimportant’ and permanently remove them.
- Remove all shadow artificially added attributes.
- Repeat until the feature importance score is assigned for all the attributes. All the above Boruta feature selection steps are illustrated in figure 2.1.

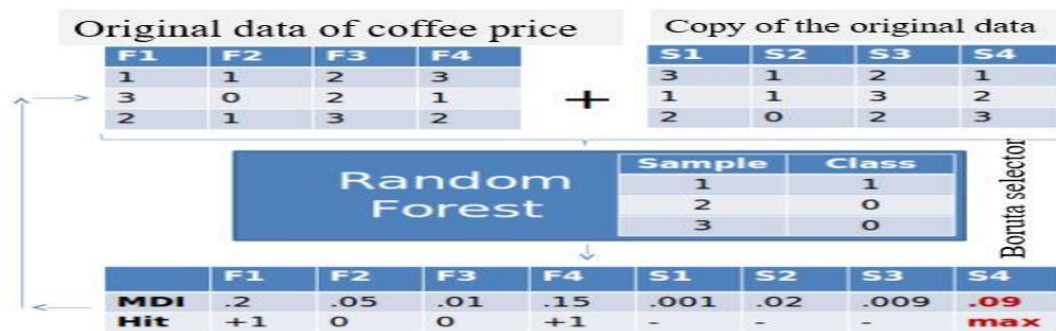


Figure 2.1. Feature selection steps of Boruta algorithm

2.4 Deep learning algorithms

Deep learning is a broader family of Machine Learning algorithm based on learning data representations, as opposed to other task specific algorithms. It uses a cascade of multi layered non-linear processing units called as neurons, which can perform feature extraction and transformation automatically (Shyu, Chen, & Iyengar, 2018). Recently, deep learning models have exhibited powerful capability in automatically generating effective features, and been successfully applied in different natural language processing tasks. Existing studies on market trend prediction have also focused on automatically generating effective features based on deep neural network models (Krishna, 2018; Long et al., 2019; R. Singh, 2017; Vargas, 2018).

This work looks to those algorithms to see if any are particularly useful in prediction of Ethiopian commodity market price data and price trend given a set of inputs generated from both technical and fundamental analysis. Algorithms investigated are: Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), Convolutional Neural Network (CNN) and hybrid of CNN-LSTM (see **Chapter 5**). These are most commonly used deep learning algorithms in market predictions and below we have brief description.

2.4.1 Recurrent Neural Network

Recurrent Neural Network (RNN) is a class of Artificial Neural Network which is designed to recognize sequences and patterns such as speech, handwriting, text, and such applications (Qiu, Shan, & Wei, 2020). RNN benefits cyclic connections in the structure which employ recurrent computations to sequentially process the input data and this helps RNNs to utilize the previous state of the hidden neurons to learn the current state. Along with the current input, RNNs take the information they have learnt previously in time and use the internal memory state to learn sequential information.

In a traditional neural network, all input vector units are assumed to be independent and sequential information cannot be made use of in it. However, whereas in the RNN model, sequential data of the time series generates a hidden state and is added with the output of network dependent on hidden state as shown in fig. 2. A typical RNN is looks like:

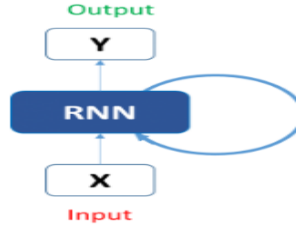


Figure 2.2. The Basic architecture of Vanilla RNN

This may be intimidating at first sight, but once unfolded, it looks a lot simpler:

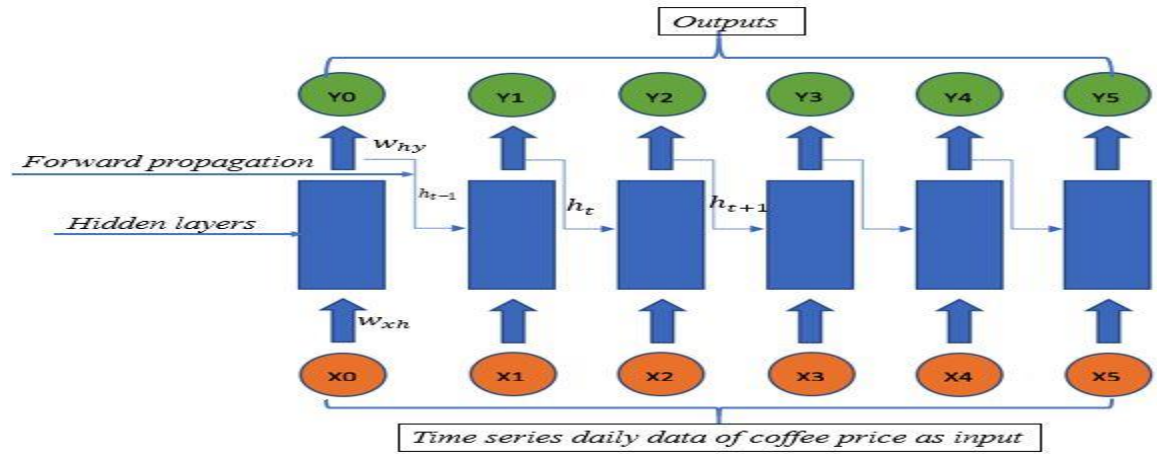


Figure 2.3. Unfolded RNN architecture

The input to hidden layer equation is given as:

$$h_t = g_n(w_{xh} x_t + w_{hy} h_{t-1} + b_h) \quad (19)$$

Where h_t : hidden layer at t^{th} instant, g_n : function, w_{xh} : input to hidden layer weight matrix, x_t : input at t^{th} instant, h_{t-1} : hidden layer at $(t-1)^{\text{th}}$ instant, b_h : bias or threshold.

Hidden to the output layer equation is given as:

$$y_t = g_n(w_{hy} h_t + b_z) \quad (20)$$

Where y_t, w_{hy}, b_z are output vector, hidden to output layer with matrix and bias respectively.

RNN (B & Mohan, 2019) takes input from two sources, one is from the present X_t and the other from the past X_{t-1} . Information from these two sources are used to decide how they react to the new set of data. This is done with the help of a feedback loop where output at each instant is an input to the next moment. Here we can say that the recurrent neural network has memory and each input sequence has plenty of information and this information is stored in the hidden state of recurrent networks memory.

As the agricultural commodity price trends are an example of time series data, this use case is the best fit for RNNs. Updates to the weights of hidden layers are back propagated based on the outcome of the output layer. However, in deep neural networks, this change will be vanishingly small to the layers in the beginning, preventing the weights to change its value and stopping the network to learn further and cases a vanishing gradient problem (Xu et al., 2018). As a result, simple RNNs fail to understand the context behind an input. Something that was said long before, cannot be recalled when making predictions in the present.

2.4.2 Long-short term memory

Long short-term memory (LSTM) is a type of recurrent neural-network architecture in which the vanishing gradient problem is solved (Ghosh et al., 2019). It can learn very long-term dependencies and works tremendously well on a large variety of problems. It is generally designed in a chain like structure, looping back to the previous layers. LSTM's are augmented by gates called "forget" gates. By controlling these gates, errors can be back propagated through any number of virtual layers. This mechanism enables the network to learn tasks that depend on events that occurred millions of time steps ago (Moghar, 2020).

LSTM networks address the problem of vanishing gradients of RNN by splitting in three inner-cell gates and build so called memory cells to store information in a long range context (Mosavi, Ardabili, & Varkonyi-koczy, 2019). As we can see from figure 2.4, a typical LSTM cell is configured mainly by four gates: input gate, input modulation gate, forget gate and output gate. The input gate takes new input information from outside and process newly coming information. The Memory cell input gate takes input from the output of the LSTM cell in the last iteration. Forget gate schedules when to forget the output data

and thus selects the optimal time lag for the input sequence. Output gate takes all results processed and generate the output.

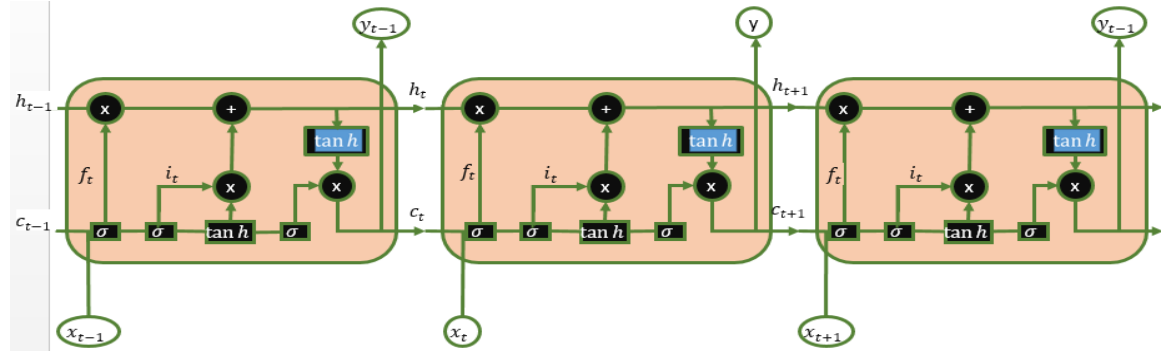


Figure 2.4. Typical LSTM network

The **Cell state** and the **hidden state** are the two states that are being transferred to the next cell. Then the LSTM memory is responsible for capturing things and manipulations to this memory. The manipulation of the captured information is done through three major mechanisms, called **gates**. Each of them is being discussed below.

A. Forget gate

A forget gate is responsible for forgetting or removing information from the cell state. The information that is not important for the model to understand things or the data that is of less importance is removed via multiplication of a filter. This helps for optimizing the prediction performance of the LSTM network.

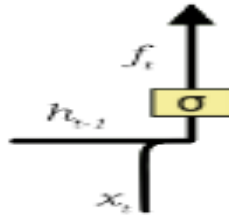


Figure 2.4a. Forget gate of LSTM cell

This forget gate takes h_{t-1} and x_t as an input variable. Where h_{t-1} is the hidden state from the previous cell states or the output of the previous cell states and x_t is the input at that particular period. Then, the given inputs are multiplied by the weight matrices and a

bias is added to the result. Following this, the sigmoid function is applied to the input value. The sigmoid function outputs a vector, with values ranging from 0 to 1, based on each number in the cell state. It is responsible for deciding which values to keep and which to discard. For instance, if the output for a particular value in the cell state is '0', it means that the forget gate wants the cell state to forget that piece of information completely from the memory. Similarly, a '1' means that the forget gate wants to remember the entire piece of information and this vector output from the sigmoid function is multiplied to the cell state.

B. Input gate

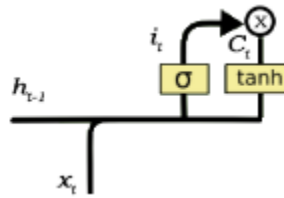


Figure 2.4b. Input gate of LSTM network

The addition of information to the cell state is the responsibility of the input gate and this addition of information is three-step process as seen from the diagram above. First step is regulating the values need to be added to the cell state using a sigmoid function. This is very similar to the forget gate and acts as a filter for all the information from h_{t-1} and x_t . The second step is creating a vector containing all possible information that can be added (as perceived from h_{t-1} and x_t) to the cell state using $\tanh$, which outputs values from -1 to +1. Multiplying the value of the regulatory to the created vector (using the $\tanh$ function) and adding this useful information to the cell state via addition operation is the last step.

C. Output gate

The selection of useful information from the current cell state and showing it out as an output is done via the output gate of the model. Information selection via output gate has again three steps: the first step is creating a vector after applying $\tanh$ function to the cell state, thereby scaling the values to the range -1 to +1. The Second is making a filter using the values of h_{t-1} and x_t , such that it can regulate the values that need to be output from

the vector created above using a sigmoid function. The last step is multiplying the values of this regulatory filter to the vector created in step 1, and finally sending it out as an output and also to the hidden state of the next cell state.

Here is its structure:

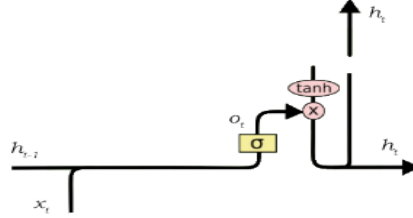


Figure 2.4c. Output gate of LSTM network

For a given time series prediction model, assume linear regression layer is applied on the output layer of the LSTM cell and let us denote the input time series as $X = (x_1, x_2, x_3, \dots, x_n)$, hidden state cells as $H = (h_1, h_2, h_3, \dots, h_n)$, and output sequence as $y = (y_1, y_2, y_3, \dots, y_n)$. LSTM do the following computations.

$$h_t = H(W_{hy}x_t + W_{hh}h_{t-1} + b_h) \quad (21)$$

$$y_t = (W_{hy}h_t + b_y) \quad (22)$$

The LSTM structure depicted in fig 2.4 is implemented through the following functions:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (23)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (24)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (25)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (26)$$

$$h_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (27)$$

Where i, f, o and c denote the input gate, forget gate, output gate, and cell activation vectors.

The cell activation (c) need to be equal to the hidden vector h and the sigmoid function can be calculated as:

$$\sigma(x) = \frac{1}{1+e^x} \quad (28)$$

2.4.3 Convolutional Neural Network

Convolutional Neural Network is a type of neural networks which have been inspired by the neurological concept of receptive fields and shows much success in the domain of computer vision and image classification (Mehtab, 2020). Like the traditional architecture of a neural network it consists input layers, hidden layers and output layers and the input layer of convolution are the output of the previous layer of convolution or pooling (S. Chen & He, 2018). Of course, they still have some unique features such as pooling layers, full connection layers, etc. The number of hidden layers in a convolutional neutral network is more than that in a traditional neural network, which, to some extent, shows that the capability of the neural network. The more the hidden layers are, the higher feature it can extract and recognize from the input (Yang, Zhai, & Tao, 2020).

The convolution operation is a 2D feature extraction technique that involves the use of a special matrix known as a kernel (Avraam et al., 2017). A convolution takes a matrix as an input and outputs another matrix proportional to the shape of the input. The elements of the output matrix are produced by convolving the kernel across the input, i.e., summing up all the elements of the Hadamard product (element-wise multiplication) of the kernel with the corresponding region of the input. The convolution operation is demonstrated with the following example in Figures 2.5a and 2.5b. Let Y be the input matrix and Z is the kernel.

$$Y = \begin{bmatrix} 3 & 3 & 2 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 3 & 1 & 2 & 2 & 3 \\ 2 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix} \quad Z = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

(a)
(b)

Figure 2.5. Convolution of input vector Y using filter Z

Figure 2.6a is a visualization of how the convolution operation starts during feature extraction. The kernel is aligned with the top-left corner of the input and it calculates the

sum of the element-wise product called Hadamard product of the overlapping region and the kernel. In the figure, this region is indicated by the shaded portion of Y (corresponding values of Z are shown in the subscript).

Let N be the output of the convolution operation. Calculating the sum of the Hadamard Product of this region, we get the first element of N: $N_{0,0} = (3 \times 0) + (3 \times 1) + (2 \times 2) + (0 \times 2) + (0 \times 2) + (1 \times 0) + (3 \times 0) + (1 \times 1) + (2 \times 2) = 12$. This is indicated by the shaded portion of the N matrix depicted in Figure 2.5a. Figures 2.6b, 2.6c and 2.6d depict the subsequent steps of the convolution operation. With each step, the kernel slides one column to the right (called a stride) and calculates the sum of the Hadamard product and so on. Once the kernel reaches the right edge, it slides a row down and starts over from the left as seen in Figure 2.6d. The convolution operation is finished once the kernel reaches the bottom edge and N is obtained. So, in essence, the convolutional operation is a weighted moving sum of the input.

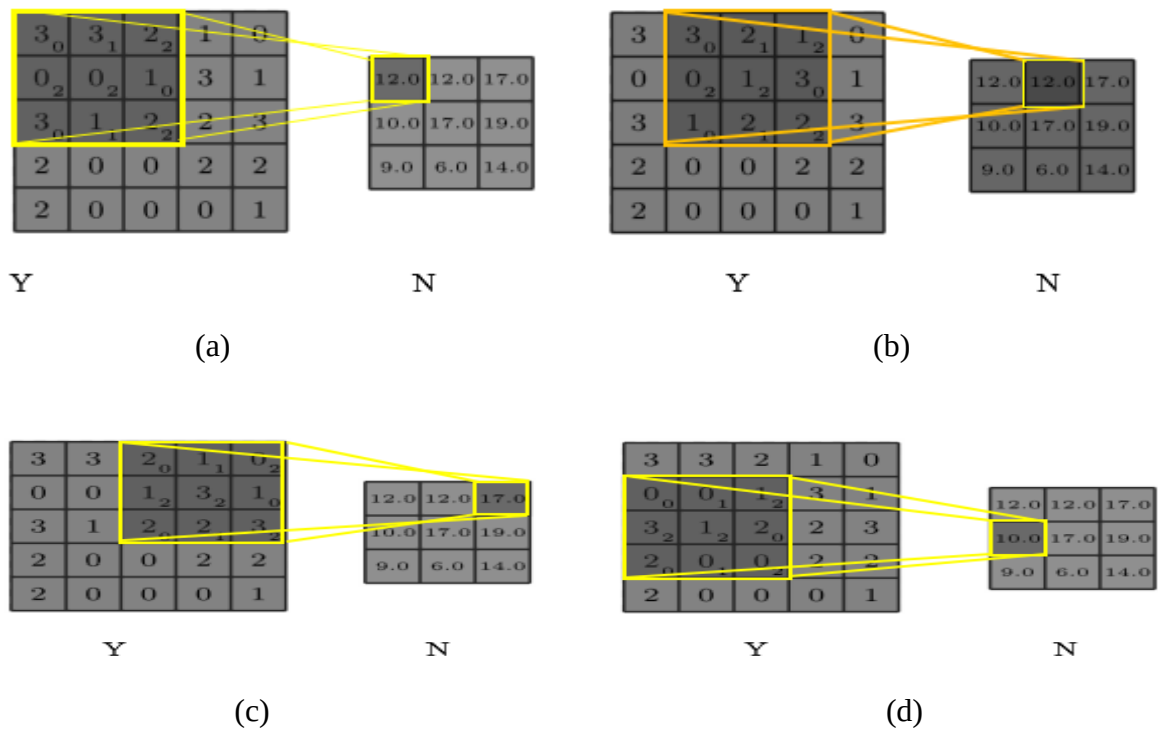


Figure 2.6: The convolution operation

The convolutional operation is usually followed by a Maxpooling operation. Max pooling is similar to convolutions in the way that they both extract information from localized

spatial regions of the input to produce a matrix. However, max pooling has no kernel associated with it. It merely slides a defined window across a 2D input and outputs the maximum number of that window.

2.5. Related Works

Mining time series data and the information from textual documents is currently hottest and an emerging topic in the data mining society. Increasingly many researchers are focusing their studies on this topic (X. Li et al., 2020; Vargas, 2017.; Vargas et al., 2017) . We have been working on Ethiopian agricultural commodity market price data analysis and prediction using deep learning models. A number of researchers published papers in the last decade on market data analysis and prediction using different approaches. The following is an overview of the papers that served as a guide to our work. It covers the four selected models (RNN, LSTM, CNN and hybrid CNN-LSTM) and in addition it looks the impact of news sentiment on market prediction.

The Networks are among the ones having the potential to model the nonlinear behavior of the market. It also applicable in speech recognition and text analysis. Lots of researchers apply recurrent neural network to solve real time problems. Based on experimental report recurrent neural network works well in both 1-day and multiday prediction. For instance (Xu et al., 2018) used recurrent neural network to predict the closing price of Pakistan commodity exchange (PXC). In this research the authors made a comparative analysis between feed forward neural network, deep neural network and recurrent neural network. The input features were open, high, close, low (OHCL) price and 12 technical indicators derived from OHCL. The result showed the recurrent neural network outperforms the other two in both 1-day ahead and multiday prediction.

Prediction of monthly price of Arecanutes in Kerala using time series and machine learning models was investigated by (Sabu & Kumar, 2020). SARIMA and Holt winters seasonal method and LSTM neural network were used as a predictive model. The daily price of arecanutes data from 2007 to 2017 was used. Based on their performance measure of RMSE LSTM model outperformed the other two. In the same manner (Jha & Sinha, 2013)

investigate the use of neural network models for agricultural price forecasting. They used ANN as a computational model and experiment was conducted on the daily price of soybean and rapeseed-mustard data from 1991 to 2010. Models trained independently and hybrid of ARIMA and Time delay neural network (TDNN). The experimental result showed that the hybrid model is more accurate than individual models.

A masters thesis done by (Faculty & Technology, 2019) utilized three machine learning models called Linear Regression, Support Vector Machine for Ethiopian Sesame market price prediction. The author conducted the experiment on price data of sesame from 2013 to 2019 to make a comparative analysis between these three machine learning algorithms. The experimental result showed that LSTM model out performs in the prediction of the given Market price. Although the work made a comparative analysis between three models it did not consider validation dataset which is very essential for parameter as well as model selection and prevent models from overfitting.

In order to analyze the look back period for stock price prediction (A. Singh & Shakya, 2020) proposed Recurrent Neural Network variants. The authors have performed a novel analysis of the parameter look-back period used with recurrent neural networks and also compared stock price prediction performance of three deep learning models: Vanilla RNN, LSTM, and Gate Recurrent Unit (GRU) for predicting stock prices of Nepal Stock Exchange (NEPSE). Eight technical indicators used along with the OHCL stock price. They used a look back period from 2 to 30 and conducted ten experiments on each look back periods. The result showed that from the three deep learning models GRU was most successful in stock price prediction and outperform the other two with the minimum MAPEs of 4.74.

A novel hybrid model based on recurrent neural networks was proposed by (Qiu et al., 2020) for stock market timing by incorporating multi-layer long short-term memory, multi-layer gated recurrent unit and one-layer Relu layer. The authors used five popular benchmark datasets from UCI Machine Learning Repository and six daily securities from Shanghai Stock Exchange and they conducted a comparative analysis between hybrid RNN and 12 state-of-the-art models to verify the superiority of the proposed hybrid RNN model

in terms prediction accuracy. The experimental demonstration shows that as opposed to 12 models, the average accuracy, MSE and AUC of hybrid RNN model (0.7406, 0.2592 and 0.7368) are significantly better than other comparison models. In addition, the proposed hybrid RNN classification procedure can be considered as a feasible and effective tool for stock market timing.

Moreover, the LSTM Recurrent Neural Network for Stock Market Prediction Using daily opening prices of two stocks in the New York Stock Exchange NYSE (GOOGL and NKE) was investigated by (Moghar, 2020) . The researchers split the dataset in to 80% for training their model and 20% for testing their model prediction performance and mean squared error (MSE) to optimize it. In addition, they used different Epochs for training data (12 epochs, 25 epochs, 50 epochs and 100 epochs) and conducted experimental analysis on each epoch. The output from their experimental demonstration shows that training with less data and more epochs can improve our testing result and at the same time allow us to have better forecasting and prediction values. For instance, losses of google stock for 12 epochs, 25 epochs, 50 epochs and 100 epochs are 0.0011, 0.001, 6.57E-04 and 4.97E-04.

An Ensemble of LSTM Neural Networks for High-Frequency Stock Market Classification Intraday stock predictions, using a large variety of Technical Analysis indicators generated from US large-cap stocks as network inputs was proposed by (Borovkova & Tsiamas, 2019). It operates in an online way, weighting the individual models proportionally to their recent performance. Lasso and ridge logistic classifiers are the two feature selection algorithms used for selection of important features from the input features. During training, the Cross-Entropy loss function was used to measure the divergence of the probability estimates output by the SoftMax layer with respect to the true labels "Sell" and "Buy". The results of the predictive framework are evaluated on the 22 tested stocks. For every stock, they made predictions during 21 weeks of the year 2014. The model trained for a total of 15 epochs and the result showed that ensemble model shows a significant upgrade of regular ensemble weighting model such as an equally weighted method or one that considers only the best individual method. In addition, it outperforms two simpler models, the Lasso and Ridge logistic regressions.

Based on some reported experimental results, we can say that the way the input data is designed to be fed and processed by CNN has a critical role on the quality of the extracted feature sets and final prediction. For example, CNN was used in (Hoseinzade & Haratizadeh, 2019) in which data of 100 companies in Borsa Istanbul were utilized to produce technical indicators and time-lagged variables. Then, variables were clustered into different groups and similar variables were put beside each other. The outcome from the experiment showed that the performance of CNN achieves F measure of 56% and outperformed baseline algorithms including a CNN with random arrangement of variables.

S. Chen & He (2018) proposed a deep learning method based on Convolutional Neural Network to predict the stock price movement of Chinese stock market. After data preprocessing, they select only five features. They conducted a classification algorithm rather than regression to show that whether the movement of the price is up or down. The result has been evaluated by different stock data, and finally indicates that CNN model is robust and can also be used to make the predictions even if the source data is 1D sequential. The final classification accuracy achieved is above 74.4%.

Convolutional Neural Network is another deep learning algorithm applied in stock market prediction after LSTM and MLP while its ability to extract efficient features has been proven in many other domains as well. In (Gao, Zhang, & Yang, 2020), CNN, LSTM, and MLP were applied to the historical data of close prices of S&P 500 index. Results showed that CNN outperformed LSTM and MLP.

Based on some reported experiments, the way how we extract features from the given data is very essential. Combining CNN with LSTM is good practice to utilize their representation abilities on different aspects of training data and then to learn a joint feature for trend prediction. For instance, TreNet is used in (Guo & Aberer, 2017), a hybrid of CNN and LSTM, to extract salient features from local raw data of time series by using CNN and LSTM to capture the sequential dependency in historical trend evolution. Power consumption dataset was used and the experimental demonstration showed that TreNet consistently outperforms baselines on the duration and slope prediction by achieving around 30% less errors at maximum.

CNN-LSTM hybrid neural network with multiple factors to predict the two real stock datasets, called JQData dataset and Pingan Bank dataset was investigated by (C. Li, 2019). At the end of the CNN-LSTM model, the authors used attention mechanism to improve the scalability and the accuracy of the CNN-LSTM model. They used 20,995 stock data for the training phase, and use 2,333 stock data for the testing phase. After training the model, they made a comparative analysis between the proposed model and other three baseline models (SVR, LSTM, and GRU) and MSE measurement was used for valuation of the model performance. The result shows that the hybrid CNN-LSTM outperforms all the other three baseline algorithms with prediction losses of MAE=0.002.

Karim et al., (2018) proposed a Hybrid Deep Learning Model for Stock Price Prediction. Hybrid model is basically a combination of Long Short-Term Memory and Gated Recurrent Unit networks. Both of the LSTM and GRU are powerful recurrent network that can perform better and faster in terms of accuracy in regression-based prediction. Firstly, they send the input to the LSTM network which will generate first level prediction. Secondly, send the output of LSTM layer to the GRU layer to get the final prediction. The authors used on S&P 500 historical time series data for training and testing of the model. The experimental result shows that 20 epochs are sufficient enough to find out the best model snapshot and the authors can have achieved MSE of 0.003% and MAPE 4.13% which is a significant improvement for stock price prediction problem.

Technical analysis and Fundamental analysis are the two main schools of thought that finance practitioners subscribe to when making trading decisions and predicting stock prices (Agustin & Batam, 2019; Beyaz et al., 2020; Vargas, 2018). Fundamental or Technical analysis is better when forecasting stock prices was investigated by (Beyaz et al., 2020). In addition, it looks whether the combined use of the two approaches is best practice. Experiment runs on 140 companies from the S&P 500 showed that predictive models using indicators based on Fundamental analysis outperform those using Technical analysis. Furthermore, in over 95% of cases, using combined indicators results in lower RMSE (0.05) compared to using Fundamental (0.141) or Technical indicators (0.1693).

The integration of fundamental and technical analysis in predicting the market price was investigated by (Agustin & Batam, 2019). The authors test the predictive ability of Fundamental Analysis and Technical Analysis model partially, then simultaneously test the integrated model of both analyses. The Fundamental Variable used in this study are Earning per Share (EPS), Dividend Payout Ratio (DPR), and Return on Equity (ROE). The technical variable used is the price during the previous six months, Positive extreme price increase, and Negative extreme price decline momentum. The experimental results of this study showed that the model based on technical analysis produces highest predictive ability compared to the other two, while integration model produces higher predictive ability compared to fundamental model.

Although many developed and developing countries proposed and utilize different prediction mechanism for future price of their agricultural market as shown in article review summary table 3.1 below, a direct implementation of their findings is not practical for Ethiopia. In addition to this, the hyper parameters used was by trial and error. Each study follows different approaches and computational algorithms based on the countries market situation. Currently, to the best of our knowledge, no one tries to use the advantages of deep learning computational algorithms for Ethiopian agricultural market price analysis and prediction. Moreover, the impact of fundamental analysis is not explored. Hence this study aims to explore predictability of agricultural commodity market price using deep learning models based on both technical and fundamental market analysis approaches.

Table 2.1. Article review summary

World wide			
Title	Author/s	Method/s	Result/Gaps
A Novel Agricultural Commodity Price Forecasting Model Based on Fuzzy Information Granulation and MEA-SVM Model	Y.Zhang &S. Na, (2018)	Fuzzy Information Granulation and MEA-SVM	* Predict with error of less than 0.6%. ** SVM is not advisable for data of multiple feature

Agricultural commodity futures prices prediction via LSTNet	(Ouyang et al., 2019)	LSTNet	* 0.0632% RMSE ** validation data not used for evaluation
Agriculture Commodity Price Forecasting using ML Techniques	(Varun, et. al, 2019)	ANN-SVM	*Since it is website prediction is based on dataset ** no guarantee for its performance.
Agricultural Product Price Prediction Model using HADT Algorithm	(Rajeswari & Suthendran, 2019)	HADT Algorithm	*90% prediction confidence **MSE is better to use for performance report.
Price Prediction for Agricultural Commodity in Bandung Regency Based on Functional Link NN and Artificial Bee Colony Algorithms	(Nhita, et.al, 2019)	NN and Artificial Bee Colony	*MAPE=7.68%&10.59% prediction for chili & onion respectively ** This indicates there is a huge error on a lot of data
Prediction of Agriculture and Mining Stock Value Listed in Kompas100 Index Using ANN Back-propagation	(Rikumahu, 2019)	ANN-Back propagation	*Prediction with 0.03848 of RMSE ** For back propagation vanishing gradient problem is challenging
Stock Market Prediction Using ML Algorithms	(Sai & Vanukuru, 2018)	SVM	*70.25% confidence **Needs expert for feature extraction
Stock Market Prediction Based on GAN	(K. Zhang,et al., 2019)	GAN	*RMSE = 4.126 ** huge errors on data
Stock price CNN-LSTM hybrid neural network with multiple factors	(C. Li, 2019)	CNN-LSTM	*MAPE = 0.0023 ** for such small feature CNN-LSTM not cost effective
In Ethiopia			
Commodity Market Price Analysis and Prediction using ML Framework	(Bulti & Ray, 2019)	SVM, K-NN, ANN&	*10.9% average MAE for ensemble ANN

		Ensemble ANN	** ANN cannot extract feature automatically
Forecasting the Ethiopian Coffee Price Using Kalman Filtering Algorithm	(Berhane,et al., 2018)	Kalman filtering	*(RMSE) is 0.000016375 ** It is statistical and cannot capture nonlinear part of data
Modeling and Forecasting the Volatility of the Export Price of Sesame in Ethiopia	(Muanenda & Yohannes, 2018)	ARIMA and GARCH	*ARMA (2, 2) model is the best-fit model. ** Domain experts needed

* represent the research findings and ** represents the gaps for a given research

CHAPTER TREE

RESEARCH METHODOLOGY

In this research, we perform both numerical analysis and textual analysis on the agricultural commodity. We used both technical and fundamental analysis approach to analyze their impact on the prediction of market price of target market. We also perform a comparative analysis between the selected five deep learning algorithms (discussed in **chapter two**) using three types of feature as an import. The first input feature consists only the denoised OHCL price of the commodity and the second is denoised OHCL plus commodity technical indicators (CTI). The third input features that we used was composed of denoised HOCL plus CTI plus News Sentiment Score (NSS) from news data. Our research methodology is presented in figure 3.1

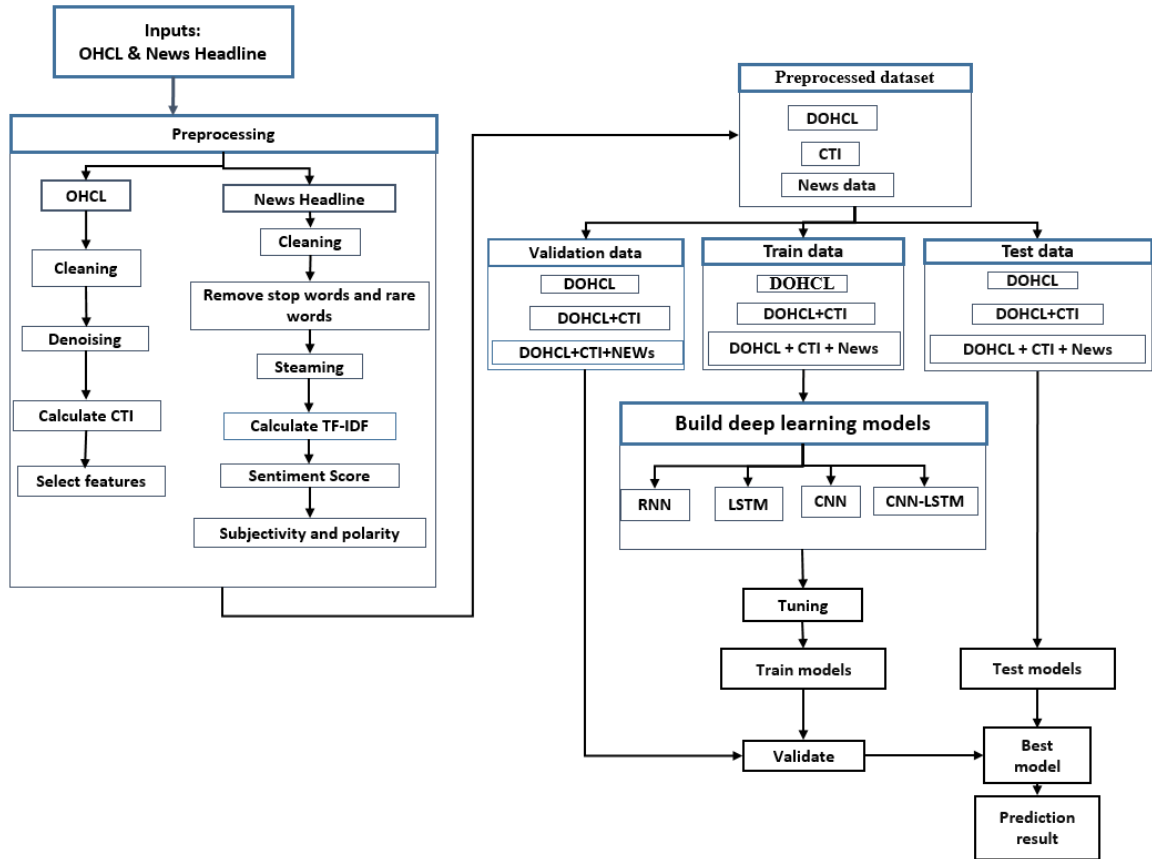


Figure 3.1 Research Methodology

Based on their prediction performance and stability the best fit model is selected. and our three main focuses are listed below:

- ❖ Empirical evaluation of the four deep neural network models using day-wise agricultural commodity data, technical indicators and fundamental analysis.
- ❖ Selecting the most important features and technical indicators and analyzing their impact on the prediction of target market.
- ❖ Analyzing combined and individual predictive ability of the two market analysis approaches.

3.1 Data collection and preprocessing

3.1.1 Data Collection

The data we used for this thesis work is collected from both commodity historical price data and financial news headlines. It was collected from Ethiopian Commodity Exchange (ECX). The collected data covered from its establishment period of January 2008 up to Dec 2020. The total records of the data are 157,956 in which the majority of the records are of coffee, which is around 110,607. Sesame and haricot have 36,637 and 10,712 rows respectively and it contained attributes namely; Trade date, symbol, warehouse, production date, opening price, closing price, lowest price, highest price and volume (ton), of haricot bean, coffee and sesame as showed in table 3.1.

Table 3.1 show the raw market price data recorded for coffee price

Trade Date	Symbol	Warehouse	Production Year	Opening Price	Closing Price	High	Low	Change	Peresentnstage Change	Volume (Ton)
05/31/2011	WSD83	HW	2003	1,200.00	0.00	1,200.00	1,200.00	0 0.0		5.41
05/31/2011	WSDA4	HW	2003	1,250.00	0.00	1,250.00	1,250.00	0 0.0		12.6
05/31/2011	WSDA3	HW	2003	1,280.00	0.00	1,280.00	1,280.00	0 0.0		5.41
05/31/2011	WLK4	GM	2003	1,160.00	0.00	1,160.00	1,160.00	0 0.0		18
05/31/2011	WBB4	BG	2003	1,100.00	0.00	1,100.00	1,100.00	0 0.0		18
05/31/2011	WBB3	BG	2003	1,140.00	0.00	1,140.00	1,140.00	0 0.0		19.81
05/31/2011	UYCB6	DL	2003	1,340.00	0.00	1,340.00	1,340.00	0 0.0		58.65
05/31/2011	UYCA5	DL	2003	1,350.00	0.00	1,350.00	1,350.00	0 0.0		7.65
05/31/2011	USDB8	HW	2003	1,265.00	0.00	1,265.00	1,265.00	0 0.0		10.2
05/31/2011	USDA7	HW	2003	1,330.00	0.00	1,335.00	1,330.00	0 0.0		33.15
05/31/2011	USDA6	HW	2003	1,345.00	0.00	1,350.00	1,340.00	0 0.0		48.45
05/31/2011	USDA5	HW	2003	1,350.00	0.00	1,350.00	1,350.00	0 0.0		12.75
05/31/2011	ULK8	GM	2003	1,345.00	0.00	1,345.00	1,345.00	0 0.0		5.1
05/31/2011	ULK7	GM	2003	1,375.00	0.00	1,375.00	1,365.00	0 0.0		40.8
05/31/2011	ULK6	GM	2003	1,385.00	0.00	1,390.00	1,385.00	0 0.0		33.15
05/31/2011	ULK5	GM	2003	1,400.00	0.00	1,400.00	1,400.00	0 0.0		30.6
05/31/2011	UJMB9	SC	2003	1,300.00	0.00	1,300.00	1,300.00	0 0.0		2.55
05/31/2011	UJMB8	BD	2003	1,280.00	0.00	1,280.00	1,280.00	0 0.0		12.75
05/31/2011	UJMB7	BD	2003	1,300.00	0.00	1,300.00	1,300.00	0 0.0		10.2
05/31/2011	UJMB6	BD	2003	1,330.00	0.00	1,330.00	1,330.00	0 0.0		5.1
05/31/2011	UHRD7	HW	2003	1,310.00	0.00	1,310.00	1,310.00	0 0.0		2.55

As we can see in table 3.1 the collected dataset contains the required attributes when compared with the other standard market price dataset. However, most of its records are zero or missing. Moreover, some data attributes are not needed for our study like; symbol, warehouse and production year. The missing values and other outliers are a very challenging problems during time series data analysis and prediction. Since we are working on a time series data, we tried to overcome those issues in our next section of data preprocessing.

3.1.2 Market Price Data preprocessing

The data what we collected from ECX is inconsistent, incomplete, noisy, chaotic, and other outliers. It is also a big issue for both data mining and data warehousing. Data preprocessing is a very important practice in the time series analysis process to make a good decision and tackle such kinds of time series problems. Major data preprocessing includes; data cleaning, data reduction and transformation, normalization and data feature selection. Each of these tasks are discussed below.

A. Data cleaning

As shown in table 3.1, the dataset has a lot of missing values. The neural network cannot handle the data that contain missing values. As a result, we used a linear interpolation mechanism to tackle this issue. It is a process of including and processing externally fetched data in to dataset and estimates the value at a point from its values of nearby points. Let you have one missing data between two values, what linear interpolation do is finding the mean of the two and assign the result for the missing records.

B. Data denoising

The other very essential practice in data preparation is noise removal. Noise for time series data mean that huge fluctuation and unnecessary information incorporated in the dataset (Zeng, 2020). Commodity price data generally has much noise and is non-stationary, which is a huge challenge to make a decision using statistical, ML (Machine Learning) as well as deep learning methods. However, a wavelet transformation (see **chapter two**), an upgraded version of Fourier transformation, can serve as a very good filter to decrease the noise in

time series data and smooth out it. As a result, in this work, we used a wavelet transform for denoising our commodity market price.

As it is shown in figure 3.2, during noise removal, discrete wavelet transformation follows four phases of decomposition, down sampling, up sampling and reconstruction.

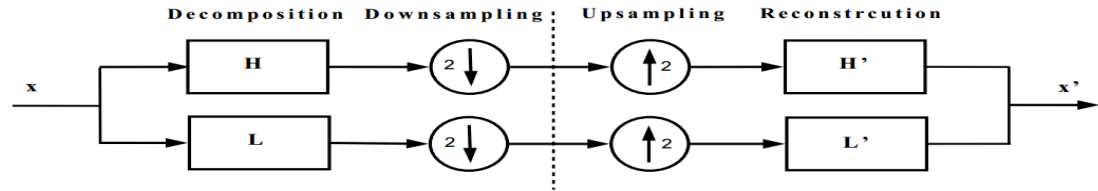


figure 3.2. Filter Bank Scheme for Wavelet Transform

In figure 3.2, H, L, are the high-pass and low-pass filters for wavelet decomposition and H', L' are and reconstruction respectively. In the decomposition phase, the low-pass filter helps us to remove the higher frequency components of the signal, and the high-pass filter picks up the remaining parts. Then, the filtered signals are down sampled by two and the results are called approximation coefficients (cA) and detail coefficients (cD). The way reconstruction process is simply the reverse process of the decomposition and for perfect reconstruction filter banks, we have $x = x'$. Where x represents the decomposed signal and x' represents reconstructed signal.

To denoise the commodity market price data, we used a soft thresholding method. The main reason for selecting this threshold is that it is not only smooths the noise from the data but also minimizes the execution time during time series denoising (Zeng, 2020). Basically, in many research papers, various thresholding methods are investigated such as hard thresholding, soft thresholding, Garrote thresholding, firm thresholding and so on. Figure 3.5 shows the wavelet decomposition for original commodity data. We can reconstruct the signal by adding details from level 1 until level 4 and approximation at level 4. Based on the denoising performance seen in the past researches (Li, Z., & Tam, V. (2017) and shape of the commodity market data, symlet 16 is appropriate for denoising the original time series.

Discrete Wavelet Transform(DWT)

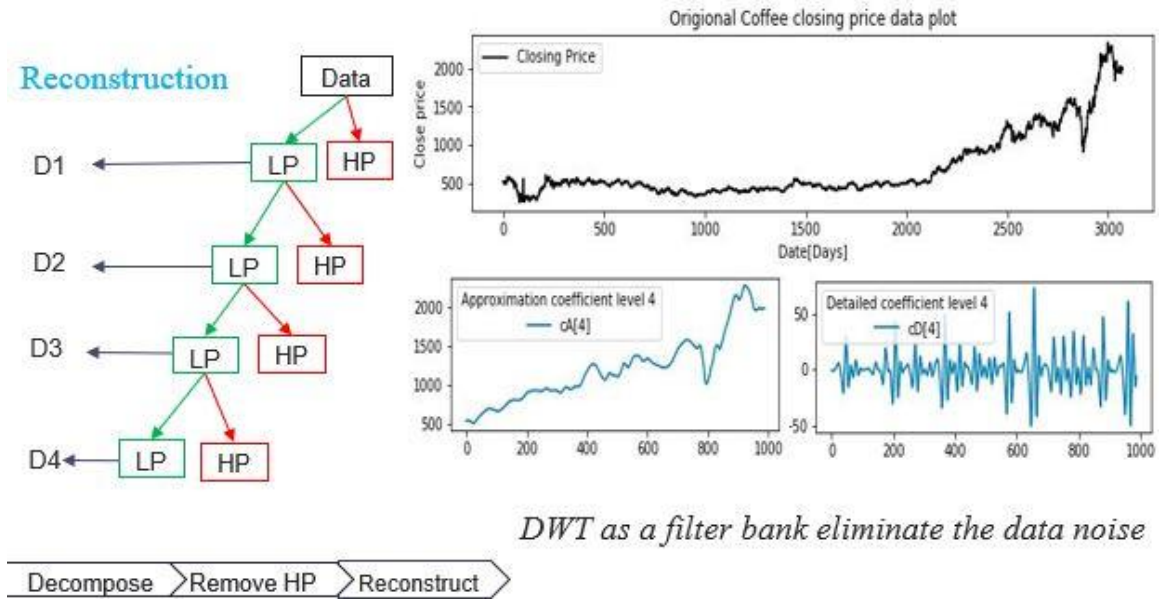


Figure 3.3 Wavelet decomposition (a) Approximation (b) Detailed using symlet 16

A signal can be further decomposed by cascade algorithm as shown in following equation:

$$\begin{aligned}
 x(t) &= A_1(t) + D_1(t) \\
 &= A_2(t) + D_2(t) + D_1(t) \\
 &= A_2(t) + D_2(t) + D_1(t) \\
 &= A_n(t) + D_n(t) + D_{n-1}(t) + \dots + D_1(t)
 \end{aligned} \tag{29}$$

Figure 3.4 shows the original signal of commodity market closing price data starting from 01/04/2016 to 21/11/2020. Meanwhile, Figure 3.6 shows the denoised signal and figure 3.7 shows the original signal and denoised signal of the closing price of commodity market data.

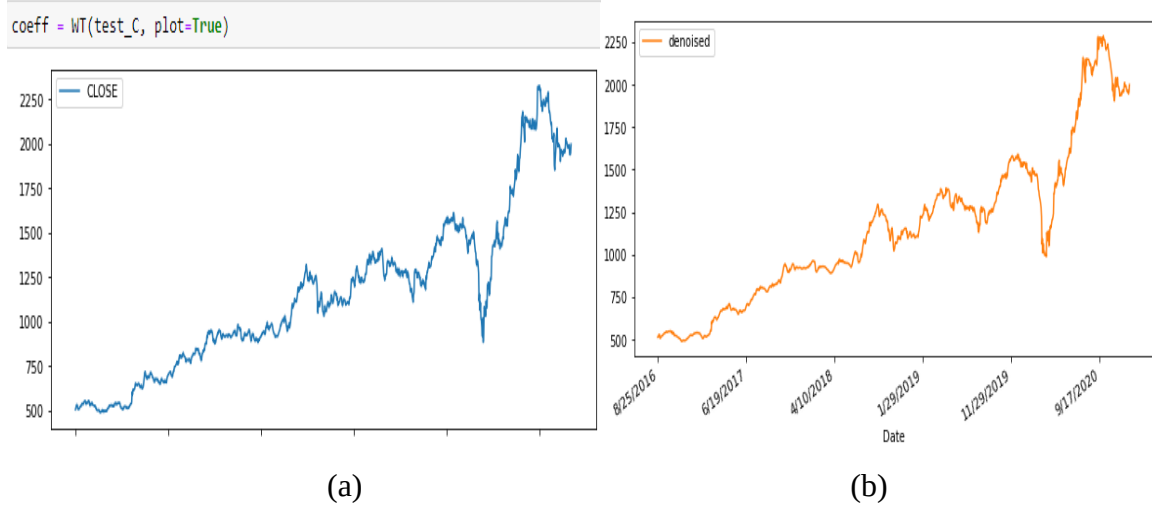


Figure 3.4 a & b Shows close and denoised signal of the closing price



Figure 3.4(c) Shows denoised and original closing price data

As we can see from the above figures, the denoised close price is smoother than the original close price of the market. Figure 3.4(a) show the closing price of coffee price starting from half of 2016 to 2020 and it was before wavelet decomposition and smoothing function applied. After the decomposition, the white noise unpredictable features removed and the reconstruction done. The result is shown in figure 3.4(b) and this helps us to make an accurate price prediction

C. Data Normalization

Time series data normalization is a rescaling or transforming of the data from the original value so that all values become 0 and 1. This can be useful and even required in some

machine learning algorithms when the data has input values with differing scales (Liu & Sullivan, 2018). Since our data is trending up and down, we used the following equation to scale or normalize it.

$$X_{norm} = \frac{X - X_{mean}}{X_{max} - X_{mean}} \quad (30)$$

D. Feature Selection

Feature selection is a very important practice in which it is a process of selecting the most relevant feature from the given data attributes to improve the performance of the predictive model. From our dataset we computed around 88 Technical Indicators (see figure 3.7) that can be taken as features and from these features, we selected the most important which represents the commodity market. For the selection of important features, we used Boruta feature selection algorithm which is wrapper method build on the top random forest classification algorithm. It follows all-relevant selection and tries to capture all the important, interesting features you might have in your dataset (see **chapter 2 & chapter 4**).

In [70]: df

Out[70]:

	Open	High	Low	Close	Adj Close	Volume	volume_adj	volume_obv	volume_cmf	volume_fi	...	momentu
Date												
7/1/2008	527.500000	537.250000	506.587006	511.037994	424.183441	23705416	-1.682334e+07	23705416	-0.709683	0.000000e+00	...	-85.4
7/2/2008	509.924988	545.750000	497.500000	536.000000	444.902924	27781780	-2.694300e+05	51487196	-0.005233	6.934890e+08	...	-20.2
7/3/2008	532.500000	532.500000	506.399994	517.525024	429.567963	19663920	-3.170005e+06	31823276	-0.044553	5.425205e+08	...	-58.4
7/4/2008	517.513000	532.799988	512.513000	524.474976	435.336700	16281456	-2.511357e+05	48104732	-0.002872	4.811826e+08	...	-44.0
7/7/2008	522.513000	530.000000	499.250000	507.049988	420.873230	14883952	-7.584216e+06	33220780	-0.074125	3.753918e+08	...	-80.2
...	...	...	...	...	...	...	...	...	...	...	...	...
12/24/2020	1947.000000	1998.800049	1945.900024	1994.150024	1994.150024	9588577	-8.045922e+08	1028075370	0.065324	-9.334220e+06	...	-23.9
12/28/2020	2006.699951	2018.000000	1995.300049	2003.300049	2003.300049	7947719	-8.069380e+08	1036023089	0.040899	2.388073e+06	...	-18.9
12/29/2020	2009.000000	2012.300049	1982.550049	1990.050049	1990.050049	8589407	-8.111966e+08	1027433682	0.076204	-1.421160e+07	...	-26.2
12/30/2020	1995.250000	2007.199951	1975.550049	1995.500000	1995.500000	10173132	-8.085448e+08	1037606814	0.072352	-4.260934e+06	...	-23.2
12/31/2020	1993.500000	2011.900024	1978.599976	1985.300049	1985.300049	8667516	-8.137245e+08	1028939298	0.014316	-1.628198e+07	...	-28.8

3072 rows x 89 columns

Figure 3.5 OHCL plus 83 technical indicators

Table 3.2 coffee price after feature selection

Technical indicator feature	meanImp	medianImp	minImp	maxImp	normHits	Decision
SMA 5 days	5.905820	5.805600	3.5773017	8.330396	1.00000000	Confirmed
SMA 10 days	4.412296	4.450179	2.5664467	6.381897	0.94949495	Confirmed
SMA 14 days	3.904888	3.865653	1.7642646	5.652071	0.86868687	Confirmed
WMA 5 days	5.108720	5.116738	1.9129550	7.403865	0.94949495	Confirmed
WMA 10 days	4.715484	4.789195	2.7125061	6.435136	0.95959596	Confirmed
WMA 14 days	4.357254	4.259337	1.9938488	6.654934	0.91919192	Confirmed
MOM 5 days	18.692034	18.659231	16.9670413	20.344669	1.00000000	Confirmed
MOM 10 days	10.406941	10.374657	9.1582195	11.926020	1.00000000	Confirmed
MOM 14 days	9.074784	9.058466	7.0379555	10.879017	1.00000000	Confirmed
STCK1 9 days	21.269620	21.339074	18.4123667	23.683564	1.00000000	Confirmed
STCK2 9 days	25.729514	25.897507	22.7613795	28.723707	1.00000000	Confirmed
RSI 14 days	15.487287	15.438833	13.8918381	16.732855	1.00000000	Confirmed
RSI 28 days	8.608997	8.602433	7.0550301	10.168651	1.00000000	Confirmed
MACD1 (26,13,9) days	16.152372	16.184757	14.2102377	18.254305	1.00000000	Confirmed
MACD SIGNAL	9.164158	9.182269	7.2546778	10.961601	1.00000000	Confirmed
MACD (15,25,15)	6.046035	5.977311	3.7503675	8.219024	1.00000000	Confirmed
R 14 days	41.157543	41.103670	38.5168811	44.780878	1.00000000	Confirmed
R 28 days	24.057435	24.017315	22.3425494	26.501515	1.00000000	Confirmed
R 50 days	19.071730	19.049960	17.2172337	21.637499	1.00000000	Confirmed
R 100 days	12.560288	12.563706	10.8720538	14.235407	1.00000000	Confirmed
CCI 14 days	17.287549	17.246896	15.4191912	19.279900	1.00000000	Confirmed
CCI 50 days	9.203503	9.272982	7.1159672	11.020916	1.00000000	Confirmed
CCI 100 days	6.259018	6.366987	3.7899522	8.369810	0.98989899	Confirmed

As it is shown in figure 3.5, after we calculate all market price technical indicators using python ta-lib function we found around 88 input features. However, all these features are not important. We used Boruta feature selection to get the best features and tested for 100 iterations. It classifies features into three categories of confirmed, tentative and rejected features. The algorithms classify 33 features as confirmed 27 features as tentative and 33 features as rejected as shown in table 3.2. the table show the first 23 confirmed feature that we used for our experiment latter in **chapter 4**.

These selected technical indicators were combined with daily price of the historical data and the feature selection method of BorutaShap with a feature importance measure of ‘gini’ and the result we got is shown in the figure 3.6 below.

```
In [89]: 1 from BorutaShap import BorutaShap
2 Feature_Selector = BorutaShap(importance_measure='gini', classification=False)
3
4 Feature_Selector.fit(X=df_X, y=df_y, n_trials=100, random_state=0, verbose=2)
```

100% ██████████ 100/100 [32:14<00:00, 19.34s/it]

```
39 attributes confirmed important: ['momentum_stoch', 'trend_ichimoku_conv', 'volatility_kcp', 'volatility_kch', 'volatility_dcm', 'momentum_uo', 'volatility_kcc', 'trend_ichimoku_base', 'momentum_wvr', 'momentum_stoch_rsi_k', 'trend_sma_slow', 'close', 'trend_adx_neg', 'low', 'trend_ichimoku_a', 'trend_psar_up', 'momentum_stoch_rsi', 'momentum_kama', 'volatility_bbh', 'trend_mas_index', 'volatility_kcl', 'trend_psar_down', 'volatility_dcl', 'trend_visual_ichimoku_b', 'volume_mfi', 'volume_vwap', 'volume_nvi', 'volume_obv', 'trend_sma_fast', 'trend_ema_fast', 'trend_visual_ichimoku_a', 'trend_ichimoku_b', 'volatility_dch', 'momentum_rsi', 'volatility_bbl', 'trend_ema_slow', 'volatility_bbp', 'volatility_bbm', 'high']
43 attributes confirmed unimportant: ['trend_kst_diff', 'volatility_bbw', 'momentum_stoch_rsi_d', 'trend_stc', 'trend_vortex_ind_neg', 'volatility_dcw', 'trend_trix', 'momentum_stoch_signal', 'volatility_kcli', 'volume_cmf', 'volatility_ui', 'trend_aroon_ind', 'momentum_tsi', 'trend_psar_down_indicator', 'momentum_ppo_signal', 'volatility_bbhi', 'trend_macd', 'volatility_kchi', 'volume', 'momentum_roc', 'trend_macd_signal', 'volatility_kcw', 'momentum_ppo_hist', 'volatility_dcp', 'volatility_bbli', 'momentum_ao', 'volume_fi', 'trend_aroon_up', 'trend_macd_diff', 'trend_kst', 'others_dr', 'trend_vortex_ind_diff', 'others_dlr', 'trend_aroon_down', 'volume_sma_em', 'momentum_ppo', 'volatility_atr', 'volume_vpt', 'trend_dpo', 'volume_em', 'trend_vortex_ind_pos', 'trend_adx', 'trend_psar_up_indicator']
4 tentative attributes remains: ['volume_adj', 'trend_cci', 'trend_adx_pos', 'trend_kst_sig']
```

```
In [90]: 1 subset = Feature_Selector.Subset()
2 subset.head()
```

Out[90]:

	momentum_stoch	trend_ichimoku_conv	volatility_kcp	volatility_kch	volatility_dcm	momentum_uo	volatility_kcc	trend_ichimoku_base
0	14.515830	521.918503	0.381719	548.954961	521.918503	0.000000	518.291667	521.918503
1	79.792746	521.625000	0.672923	561.810664	521.625000	48.787909	522.354167	521.625000
2	41.502640	521.625000	0.447904	558.176558	521.625000	45.731875	521.172224	521.625000
3	55.906868	521.625000	0.544376	553.019629	521.625000	47.816003	521.694832	521.625000
4	19.792721	521.625000	0.296125	550.985862	521.625000	43.489188	519.775885	521.625000

5 rows × 9 columns

As it is clearly seen in figure 3.6, from 89 feature given to the feature selector algorithm, 39 features or attributes are confirmed as important feature and 43 features are confirmed as not important. This helps us to reduce the dimensionality of the dataset and improve the accuracy and performance of our predictive models.

The second data, financial news headlines, which we used in our work was collected from <https://dailycoffeenews.com> for Coffee and <https://agris.fao.org/agris-search/search.do> for Sesame using BeautifulSoup which contained 5299 news headlines starting from October 5, 2012 up to November 27, 2020. The attributes of the collected data are; news headlines, URLs, date of publication and the content of the news. As we know, in fundamental market analysis approach, analyzing the market is based on different

perspective. For instance, news related to only the market price of the commodity, based on the political issues, conflicts, governmental policy and so on. In this work we focused only on the news related to the market price of the given commodity. Following snapshot shows some of the examples of news headlines which influenced the commodity prices.

Table 3.3 show some examples of news headline that we collected.

Date	Headlines	Influence
3-Nov-2020	peace-coffee-exiting-retail-in-minneapolis-wildflyer-to-take-over-original-shop	Negative
28-Oct-2020	Coffee Trading Platform Algrano Grows with \$4 Million Funding Round	Positive
28-Oct-2020	Humans Are Sending Coffee to Space, But for Real This Time	Positive
26-Oct-2020	Filtered Coffee Can Reduce Type 2 Diabetes Risk, Research Shows	Positive
6-Nov-2020	Foster Coffee Grows Faster with New East Lansing Cafe	Positive
5-Nov-2020	Dripped Coffee Brings a NY State of Mind to Lower East Side Shop	Negative

During news headlines data preprocessing, all non-alphabetical characters were removed and rest of the characters were converted to small letters. And then we remove stop words, rare words and other special characters to get rid of noise. After that, we performed steaming, which is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Next word embeddings are calculated. All these news headlines preprocessing is shown in figure 3.7 below which is taken from experimental result of Jupiter notebook.

In NLP words are represented in a distributed vector space based on their usage, and therefore words which are used similarly will have similar representation. First tokenizer takes the input words and indexes the text corpus based on word count, term frequency inverse document frequency (TF-IDF).

REMOVING STOPWORDS

```
In [20]: 1 stop = stopwords.words('english')
2 coffee['Coffee News Headlines'] = coffee['Coffee News Headlines'].apply(
3     lambda x: " ".join(x for x in x.split() if x not in stop))
4 coffee['Coffee News Headlines'].head()
```

```
Out[20]: 3441 coffee bean tea leaf plans add 100 new york st...
3442 unpacking coffee kandace ray zurichs birchbach
3443 green coffee fetches recordbreaking 105lb gesh...
3444 west virginia void coffee filled ingenuity opt...
3445 taylor maid becomes taylor lane organic coffee...
Name: Coffee News Headlines, dtype: object
```

REMOVING COMMON WORDS

```
In [318]: 1 freq = pd.Series(' '.join(coffee['Coffee News Headlines']).split()).value_counts()[:10]
2 freq
```

```
Out[318]: coffee      1092
new          214
roasters     124
de           102
extraction   73
launches     70
cup          66
la           66
roasting     62
cafe         62
dtype: int64
```

```
In [319]: 1 freq = list(freq.index)
2 coffee['Coffee News Headlines'] = coffee['Coffee News Headlines'].apply(
3     lambda x: " ".join(x for x in x.split() if x not in freq))
4 coffee['Coffee News Headlines'].head()
```

```
Out[319]: 3441 bean tea leaf plans add 100 york stores starti...
3442 unpacking kandace ray zurichs birchbach
3443 green fetches recordbreaking 105lb gesha villa...
3444 west virginia void filled ingenuity optimism
3445 taylor maid becomes taylor lane organic appoin...
Name: Coffee News Headlines, dtype: object
```

REMOVING RARE WORDS TO GET RID OF NOISE

```
In [320]: 1 freq = pd.Series(' '.join(coffee['Coffee News Headlines']).split()).value_counts()[-10:]
2 freq
```

```
Out[320]: pulley      1
parade      1
futuro      1
```

Figure 3.7 shows news headlines preprocessing

As shown in figure 3.7, the rare words in the document are considered as noise and has no any contribution in textual analysis. As a result, rare words are removed from the news data. Next, sentiment of the headlines was calculated. To calculate the sentiment, we have used Valence Aware Dictionary and Sentiment Reasoned (VADER) which is a lexicon and sentiment intensity analyzer specially tuned to sentiments expressed in social media. The headline of a financial news article can provide enough context to analyze the sentiment.

As a result, we have ignored the body of the article and focus only on the headlines for analysis. Then the sentiment polarity and intensity for each news headline was calculated using VADER algorithm. It returns the value from 0 to 1 to represent a positive polarity and the magnitude represents the intensity and value from 0 to -1 to represent negative polarity. 0 can be treated as neither positive nor negative. After we perform all text data preprocessing, we have got the dataset shown in figure 3.7

```
In [622]: tdf = pd.read_csv('C:/Users/IBRO/Documents/Ms Thesis/Thesis data/Coffee/dailycoffeenews_prepared.csv')
sia = SIA()
tdf['Compound'] = [sia.polarity_scores(text)['compound'] for text in tdf['Coffee News Headlines']]
tdf['Positivetext'] = [sia.polarity_scores(text)['pos'] for text in tdf['Coffee News Headlines']]
tdf['Neutral'] = [sia.polarity_scores(text)['neu'] for text in tdf['Coffee News Headlines']]
tdf['Negativetext'] = [sia.polarity_scores(text)['neg'] for text in tdf['Coffee News Headlines']]
tdf
```

Out[622]:

	Date	Coffee News Headlines	polarity	subjectivity	Compound	Positivetext	Neutral	Negativetext
0	2008-01-02	Ramos Brannen Carmichael and Bonchack Earn ...	0.000000	0.100000	0.0000	0.000	1.000	0.0
1	2008-01-03	Here Are Your Good Food Award Coffee Winners	0.700000	0.600000	0.8591	0.655	0.345	0.0
2	2008-01-04	The Good Food Awards How it Works How to Win...	0.750000	0.500000	0.8658	0.492	0.508	0.0
3	2008-01-06	The New Cupping Form for Farmers a Work in Pr...	0.136364	0.454545	0.4215	0.259	0.741	0.0
4	2008-01-07	What s In Your Cup Artificial Sweeteners Edition	-0.600000	1.000000	0.0000	0.000	1.000	0.0
...	...	...	...	...	...	...	...	...
2669	2020-11-11	smalltown coffees big dreams come true with nw...	0.175000	0.375000	0.6705	0.407	0.593	0.0
2670	2020-11-12	hilda leguia gonzales of cusco wins peru ...	0.300000	0.200000	0.8316	0.494	0.506	0.0
2671	2020-11-15	peace coffee exiting retail in minneapolis wil...	0.375000	0.750000	0.7003	0.367	0.633	0.0
2672	2020-11-16	coffee roasters guild names five leadership co...	0.000000	0.000000	0.0000	0.000	1.000	0.0
2673	2020-11-17	breathe in airship coffees cafe in an arkansas...	0.000000	0.000000	0.0000	0.000	1.000	0.0

2674 rows x 8 columns

Figure 3.7 shows preprocessed news dataset

Finally, we tried to integrate them with the commodity market price trend based on date to find their impact on the commodity market price. Even though the score from sentiment can reveal whether the news is positive or negative, it is hard to know how much it would affect the commodity market price. Therefore, the challenges of forecasting the accurate commodity market price was simplified to problem of predicting whether the commodity market price rises or falls, by converting the values of target column to Boolean having values 0 or 1 which denotes the price fall and price raise respectively.

3.2 Experimental Data

Our experimental data is organized from both technical indicators which is derived from the historical price data and news headlines that is scraped from different resources. It is taken from highly traded stocks of three different sectors which are Coffee, Sesame and Haricot beans sectors from ECX. We extract only the day-wise market closing price of each commodity for prediction because day wise market price is preferred since investors make decision on buying which commodity or forfeiting based on the closing price of the market. It contains the OHCL, 88 technical indicators and polarity and subjectivity attributes that are derived from around 10,000 news headlines which are related to the selected commodities. The data from historical price and technical indicators derived from historical price data using ta-lib python library combined with the sentiment scores from news headlines based on date of publication.

Training set: The training set is used to train our model and it ranges between 527.5 and 2493.5 for coffee of symbol WYCAQ1, 495.3 and 2860 for Haricot Beans RKBGA1 and 1437.5 and 5885.31 for Sesame WHGS1. We used 80% of each dataset for training the network, 10% for testing and 10% for validation. The extracted data was then subjected to normalization to unify the data range within 0 and 1 as discribed in data normalization section. Normalization of data is done to bring all price data into a common range. Since we are using market price data from different market, we need the data to be under a common range. The normalized data was given as the input to the network in a window size of 50 to predict 400 days of future market price data and the output from the predictive model was subjected to a De-normalization process for acquiring original predicted values. We used from 100 to 1000 epochs for training each network. The window size was fixed by performing hyperparameter tuning and error calculation on each window size which varies from 50 to 250. Among those windows, the window size of 50 resulted in minimum error than other window sizes.

Testing set: Used to assess the performance of our final model and fot the verification of whether the models identify the common patterns of market price data, we tried to test the model using 10% of each dataset. From dataset, we extracted only the day-wise market

closing price. Then we normalized the selected data by using the equation (30) to unify the data before giving it as input to the model.

Validation set: When we are considering multiple hypothesis spaces for a supervised learning problem, we can compare the resulting fitted models using the validation set. This means that it is used to compare different fitted models. To have a validation dataset during prediction of time series data prevents the data leakage which results in underfitting and overfitting. In order to tackle this issue we used 10% of the dataset for validation purposes.

3.3 Model Selection

In this research work, our main goal is to analyze existing agricultural commodity market data of Ethiopia and to predict market price using different computational deep learning algorithms. Keep this in mind, the study aim was to explore which deep learning technique is the best fit in the prediction of the future Ethiopian commodity market price. Throughout this work, we tried to explore different deep learning algorithms and the one that is best fit in predicting the future market price was selected.

The performance of each deep learning technique was tested using selected valuable features from the dataset during feature selection phase. Then based on their predictive ability and stability in their performance the best fit model was selected. Deep learning models selected for our thesis were RNN, LSTM, CNN and CNN-LSTM. The selection of these models is based on their prediction performance and applicability seen in reviewed researches in **chapter two**.

The rationale behind choosing Recurrent Neural Network is that it is the best and widely used model for sequence prediction and is superior at time series data processing (A. Singh & Shakya, 2020). It has been proved as one of the best methods used to solve some popular problems like speech-recognition, text pattern recognition, text generation etc. (Guo & Aberer, 2017). It has the ability to capture the temporal dynamics of time series data and enables to make robust predictions for time series with outliers and change points. One more advantage of using RNN is that unlike the conventional Artificial Neural Network

(ANN), which cannot memorize the former historical information of the sequences, RNN has a hidden state to maintain features of the whole sequence.

The motivation of choosing LSTM is that it has been proved to be effective in capturing temporal sequential information in natural language processing tasks. Since we are working on market price data and comprises abundant temporal information, we select LSTM to capture latent information in financial news, particularly to model the relationship between the commodity market movement and the news (Xu et al., 2018). It is well-suited in forecasting or predicting of time-series data and avoids the gradient vanish issue in RNN and uses historical information through the input, forget and output gate. One more advantage of using LSTM is that it can learn automatically when to forget the memory for a particular sequence.

Convolutional Neural Network (CNN) is included in this study due to the following reasons; it is an excellent technique to obtain the features of image data. CNN has shown great success at modeling image data in computer vision field and has shown fantastic success in modeling image data in computer vision and image processing (Vargas et al., 2017). For the time series data, we can transform the financial time series data to image data, and use CNN to obtain the features of stock prices. CNN can be better than RNN on catching semantic from texts and (deep learning). In addition to this CNN has superior performance in data feature extraction (C. Li, 2019).

The motivation of choosing a hybrid CNN-LSTM model is for utilizing the best advantages of both models. Convolutional Neural Networks (CNNs) is best to extract salient features from local raw data of time series while long-short term memory recurrent neural network (LSTM) is capable of capturing the sequential dependency in historical trend evolution (Guo & Aberer, 2017).

3.4 Metrics

In our work, the performance evaluation technique that we used is Mean Squared Error (MSE) which measures the average of squares of the errors. The difference between the

actual value and the predicted value is termed as an error. Equation 31 below denotes the mathematical expression for calculating the mean squared error.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (31)$$

Where y_i denotes the vector of the actual values and $\hat{y}_i$ is the vector of predicted values. It is generally used for vectors with continuous values. We used this metric for comparing the results of the Numerical Analysis models and the merged model. In addition, we used the root mean squared error and r-squared for model prediction performance measure.

3.5 Tools and Techniques

We used python libraries such as, Keras and TensorFlow, which will be used for the implementation of deep learning. Keras is very popular and powerful library running on top of TensorFlow. The sequential model is designed for simple architectures, where we just want to stack layers in a linear fashion (Hope et al., 2018.). One more tool that we used is python BeautifulSoup for web scraping to scrape the news data. For commodity market price data denoising we utilized a discrete wavelet analysis. In addition to this we used NLTK and ktrain libraries for sentiment analysis.

3.6 Evaluation

To evaluate the system in a rational method, testing datasets will feed into the selected proposed model. Subsequently, the model evaluated by comparing its output against the observed data using MSE, MAPE, training time, Directional Accuracy (DA) and MAE values for evaluating prediction performance.

CHAPTER FOUR

EXPERIMENT AND RESULT DISCUSSION

The commodity market historical price data are collected both in a formal and informal way and the financial news related to the commodities was collected from <https://dailycoffeenews.com> and <http://www.fao.org/ethiopia/news/detail-events/en/c> using python Beautiful Soup. These collected data of commodity market and a financial news article can give us valuable information to formulate and direct future market and enables us to see different market opportunities. However, commodity market price data analysis and prediction are not a common practice in Ethiopia and is often carried out using traditional tools and manual practices making the processing time taking and prone to errors.

For tackling the mentioned problems, we proposed a deep learning approach based on both fundamental and technical market data (see **Chapter 3**). To see if the proposed methodology holds, we conducted an experiment using the collected data and explored the following research questions.

Research Question (RQ) 1 [Model building and Selection], which deep learning algorithm is the best fit model in the prediction of Ethiopian agricultural market data?

To answer this research question, we investigated if every deep learning algorithm have equal performance in predicting future market price using three categorical features. The first category contained only the daily price data of the market. In the second category we used the daily price data and the selected technical indicator attribute and in the third category we used the daily price data, selected attributes from technical indicator and news sentiment score from news articles. In addition, we evaluated the prediction performance of the selected algorithms on each training and validation data.

Research Question (RQ) 2 which market analysis approach is determinant in Ethiopian market price prediction?

To answer this research question, we conducted various experiments using both fundamental and historical and also the technical market analysis data. We trained the predictive model on the separate data set of which one with only historical price data and the other is both historical data and sentiment score from news headlines. Then we evaluated the prediction performance of the selected model using sentiment score from the news headlines and market technical indicator from historical price data. To know the relationship between the news and market movement, we used the sentiment score calculated from news and normalized it from 0 to 1, and the market price up and down by converting daily closing price to price signals.

Research question 3: Which is the best way to use the two main market analysis approaches in the market price prediction?

To investigate whether the market analysis approaches can be used individually or in combination, the experimental results from the model selection experiment were used.

4.1. Model Comparison and Selection (RQ1)

The study aims at analyzing and predicting the commodity market price using the best fit deep learning approach. To do so, we have used four deep learning approaches, RNN, LSTM, CNN and CNN-LSTM to analyze and predict the historical price data of 3 commodities (coffee, sesame and haricot bean) which are the most commonly traded commodities (see **Chapter 3**). The data set encompassed the trading days of 13 years i.e. from 16th Nov, 2008 to 15th Nov, 2020. The experiments are performed on a system with Intel Core i5 (2.7 GHz.) processor and 8GB RAM, running Microsoft Windows 10 (64-bit) operating system. The models were built in Python language (version 3.7.0) utilizing Keras as the high-level API developed for Google's TensorFlow.

For training purpose, we used 80% of each the data from the three categories which are DOHCL, DOHCL+CTI and DOHCL+CTI+NEWS DATA on different window size as well as the length of days. We split the testing data into validation and out of sample to prevent over and under fitting.

4.1.1 Recurrent Neural Network Building and Prediction

The first deep learning model that we build and used for Ethiopian agricultural market price data analysis and prediction is the Simple Recurrent Neural Network (Simple RNN). This class of network is one of the deep learning approaches under the family of Recurrent neural network (see more detail in **chapter 2**).

To build Simple RNN model for our research work, we used a stacked RNN way of model building. All the hyper parameters for this model was taken from the result of the automated hyper parameter tuning algorithm called Keras-tuner. It is a python library that lets you pick the optimal combination of hyper parameters for your model (Yang et al., 2020). The activation function used is 'Relu', and learning rate 0.001, epochs 20, batch size 32 and hidden layers ranging from 45 to 50. The time stamp used in model building is 30. The model that we built is depicted in the figure 4.1 below.

```
1 model,train_error,val_error = fit_model(train,val,timesteps,h1,lr,batch_size,num_epochs)
2 plot_error(train_error,val_error)
```

Congratulation!!! Ahmed you biuld a simple RNN model.....
Model: "sequential_1"

Layer (type)	Output Shape	Param #
simple_rnn_3 (SimpleRNN)	(None, 30, 4)	36
simple_rnn_4 (SimpleRNN)	(None, 30, 50)	2750
simple_rnn_5 (SimpleRNN)	(None, 45)	4320
dense_1 (Dense)	(None, 1)	46

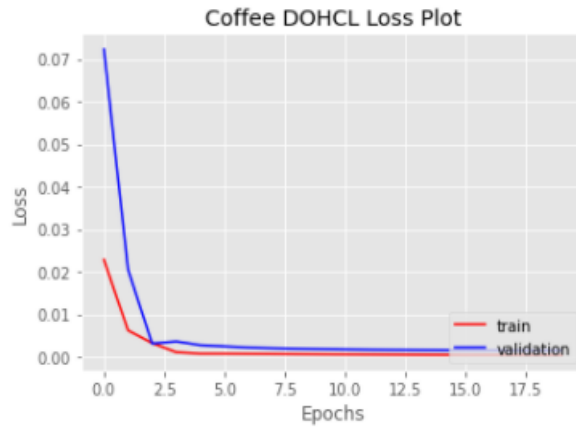
=====
Total params: 7,152
Trainable params: 7,152
Non-trainable params: 0

Figure 4.1 Shows the RNN model structure we build during our experiment

After we build the recurrent network as shown in figure 4.1, we train it using the training data taken from the three category of each commodity datasets. In table 4.1 we have the MSE, MAE, directional accuracy (DA), time it takes for training(T(s)) and R-squared error

values obtained for training coffee, sesame and haricot bean for the period of 5th October 2008 to 30th December 2017.

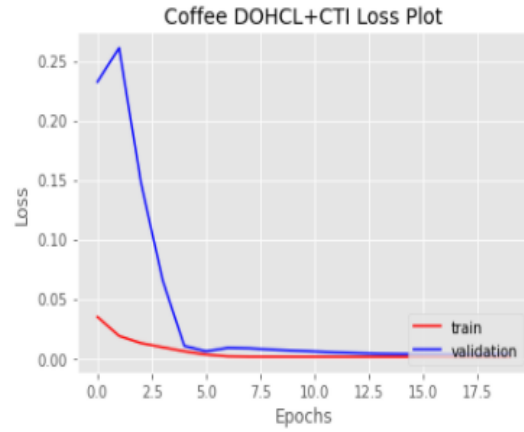
Epoch 20/20
70/70 - 3s - loss: 4.0167e-04 - val_loss: 0.0015



Wall time: 1min 19s

(a)

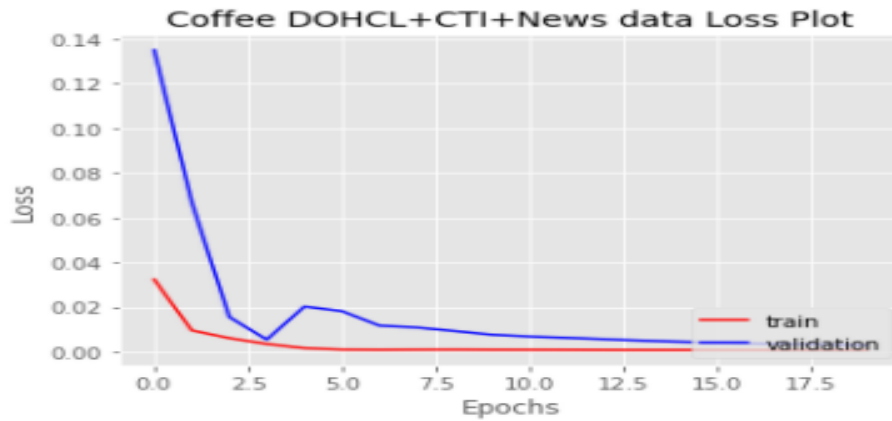
Epoch 20/20
71/71 - 2s - loss: 9.9176e-04 - val_loss: 0.0038



Wall time: 57.4 s

(b)

Epoch 20/20
71/71 - 1s - loss: 7.3252e-04 - val_loss: 0.0029



Wall time: 36.4 s

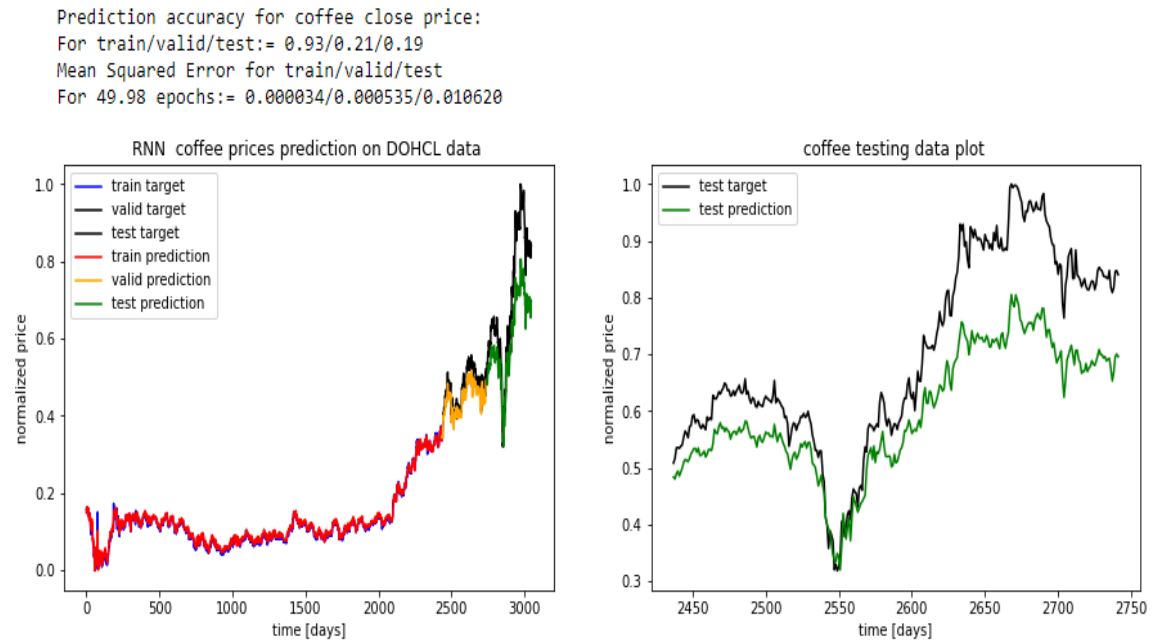
(c)

Figure 4.2a shows the validation and training loss of RNN for coffee

From figure 4.1 we can see that for epochs 0.0 to around 2, the train loss is very small than the validation loss. This shows that during the first training time there is an under fitting of the prediction model. However, when the epoch increases from the 2.5 to 15 then 20, the

train and validation loss becomes close each other and up to 15 epochs the validation loss is greater than the train loss. This show that the predictive mode goes in the right way. In figure 4.1b we can see that, until epoch 6.8 the validation and train losses are far apart which indicates that there is an over fitting for training the model using the DOHCL and CTI. However, after the epoch 7.5 the two losses became closer and closer.

If we compare this lose function with figure 4.1a. we can conclude that using only DOHCL dataset results accurate prediction performance. The time it takes for training is also greater than the first one and this indicated that it takes more computational resources. Figure 4.1c shows the training and validation loss of RNN model when it trained using DOHCL, CTI and News dataset. As we can see, under-fitting is happened from epoch 2.5 to 12.5. the train loss of RNN model based on all the three dataset is higher than loss in figure 4.1b and less than train loss of RNN using only DOHCL. The prediction and actual values of coffee price is depicted in figure 4.2a, b, and c below.



(a)

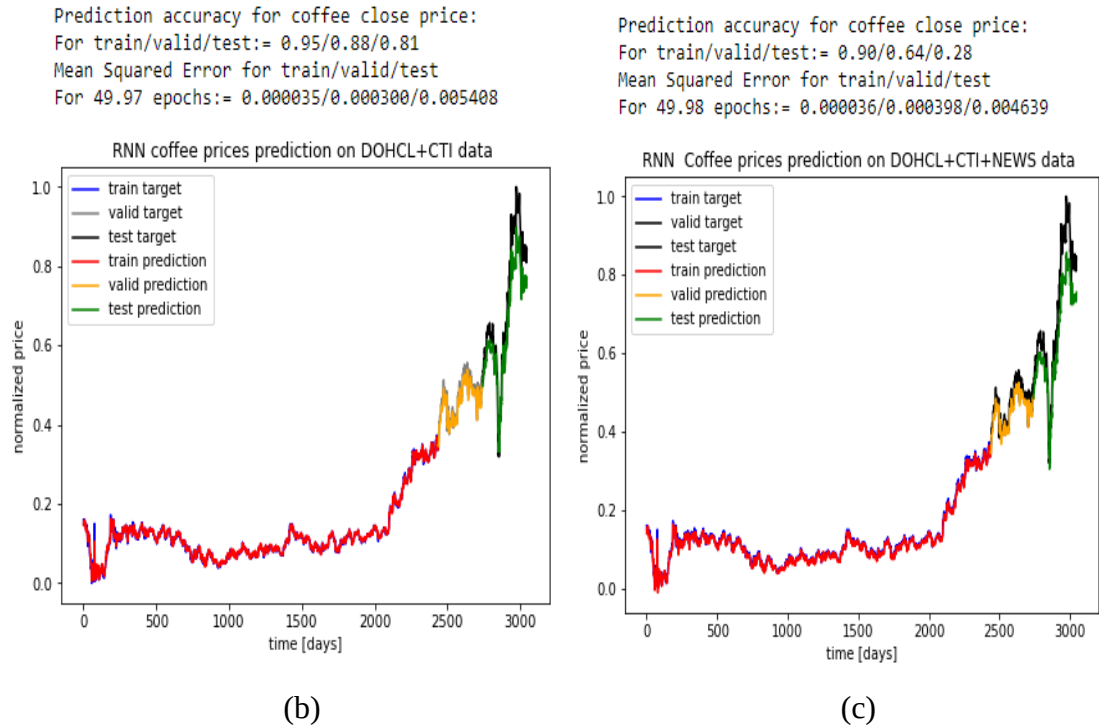
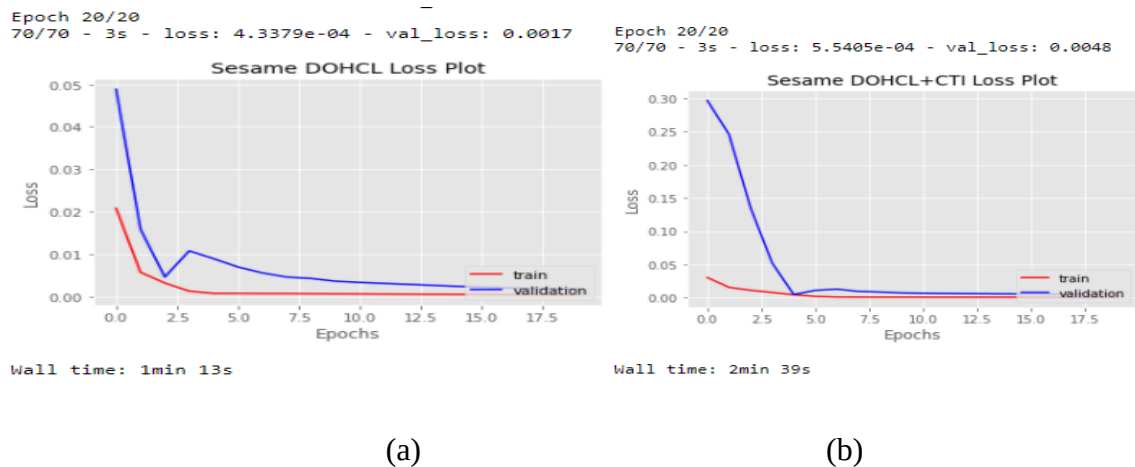
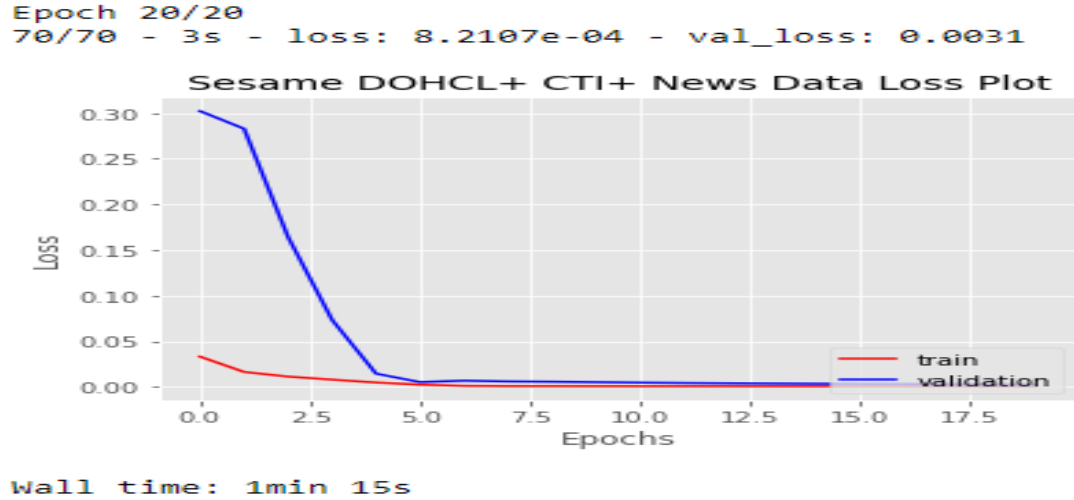


Figure 4.3 Shows the predictive and real price for coffee using RNN

From figure 4.3, it is clearly seen that there is a large decrease in price of coffee around 2800 to 2900 due to the pandemic corona virus spread in Ethiopia. As it is shown in figure 4.2a, RNN model failed to capture the patterns from the time series data during the last marketing periods. However, after it is trained with the selected commodity technical indicators it tries capture the sequence of the time series data. According to figure 4.2c, News data contribute little to the increment of the accuracy of prediction model.





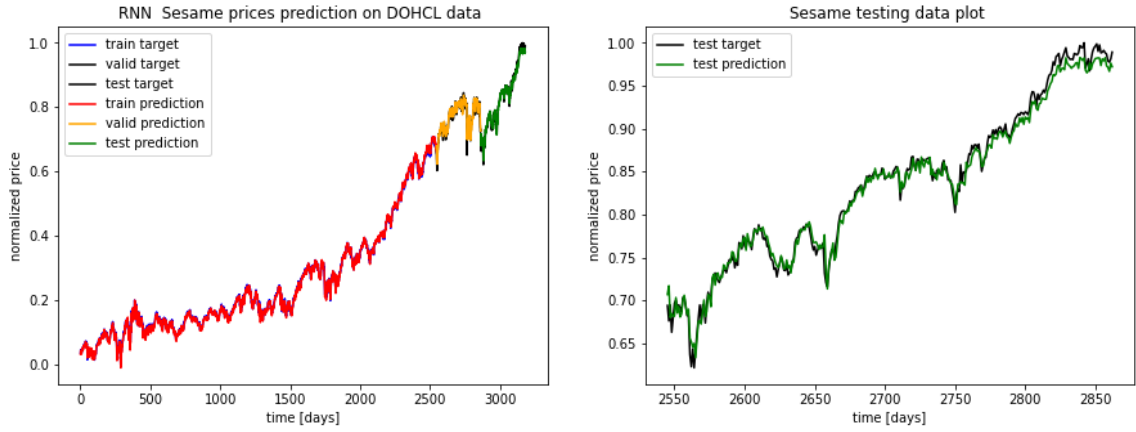
(c)

Figure 4.4 Show the training and validation loss of RNN for sesame

In case of Sesame, as we can see from the figures depicted above the RNN model performs best when both historical price data and technical indicators however when we add the news data it failed to capture the data pattern for the first fifteen epochs. This indicates that the model goes under-fitting during the first 15 epochs. From this we can conclude that in terms of sesame market price, the market technical indicators have a significant impact in the prediction and analysis of the market price.

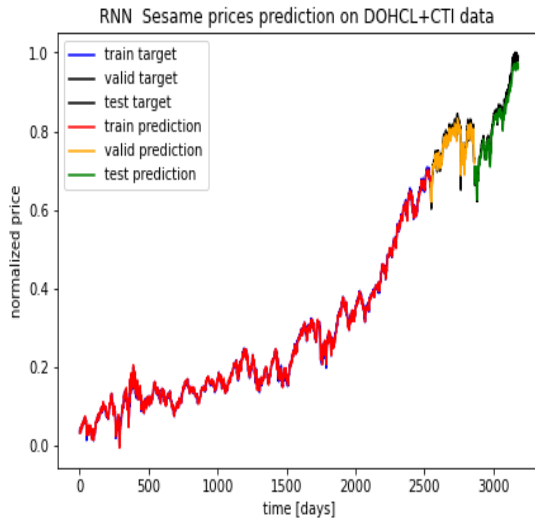
Visualization of the prediction, validation and testing using RNN model for Sesame market price data is shown in the figure 4.5. Figure 4.5a shows the prediction result of RNN model using only DOHCL whereas figure 4.5b&c shows the prediction result of RNN model when it trained using DOHCL+CTI and DOHCL+CTI+News data respectively.

Prediction accuracy for Sseame close price:
For train/valid/test:= 0.83/0.52/0.48
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000041/0.000204/0.000293



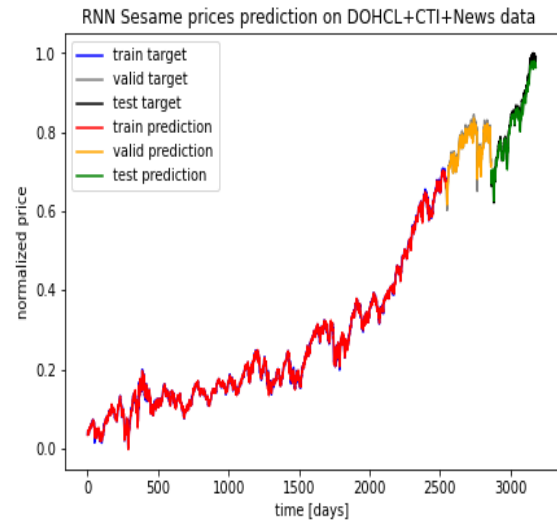
(a)

Prediction accuracy for sesame close price:
For train/valid/test:= 0.78/0.60/0.53
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000051/0.000182/0.000184



(b)

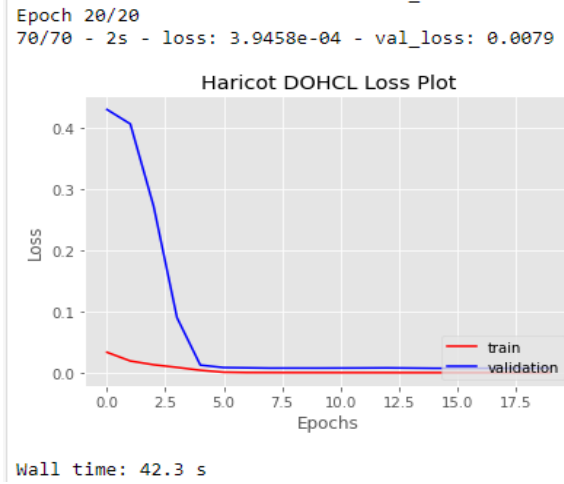
Prediction accuracy for sesame close price:
For train/valid/test:= 0.79/0.57/0.57
Mean Squared Error for train/valid/test
For 49.99 epochs:= 0.000044/0.000187/0.000259



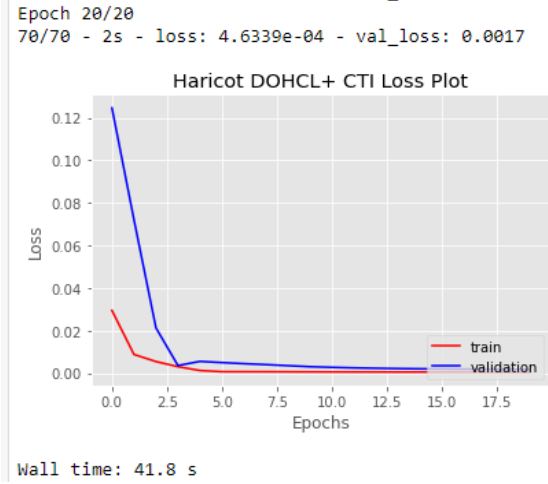
(c)

Figure 4.5 Shows the prediction result of RNN for Sesame price

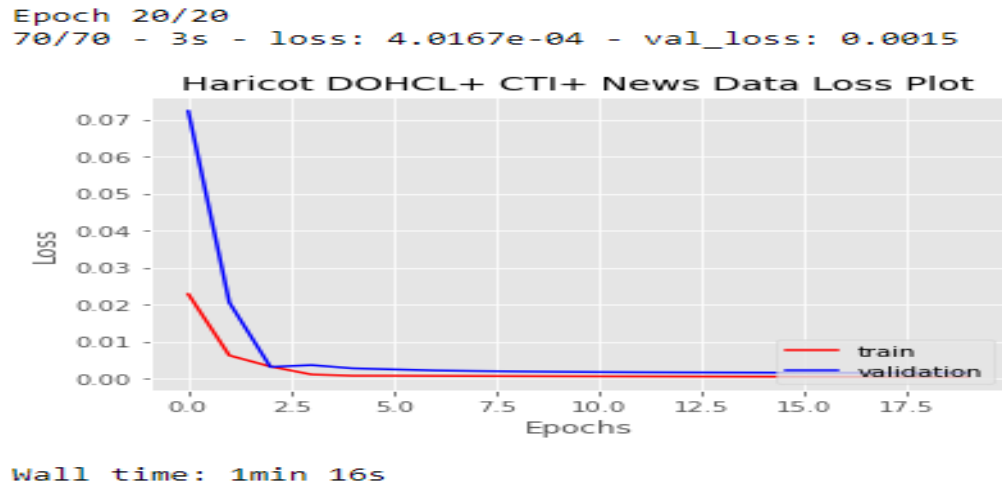
Like coffee and Sesame, the data of haricot bean divided in to three categories, which consists of DOHCL, DOHCL+CTI and DOHCL, CTI and News data. The experimental result that we incurred is depicted in the immediate section below.



(a)



(b)

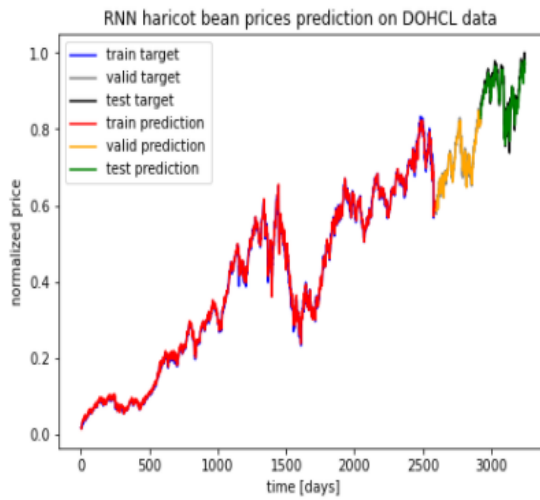


(c)

Figure 4.6 shows the train and validation loss of RNN model on haricot

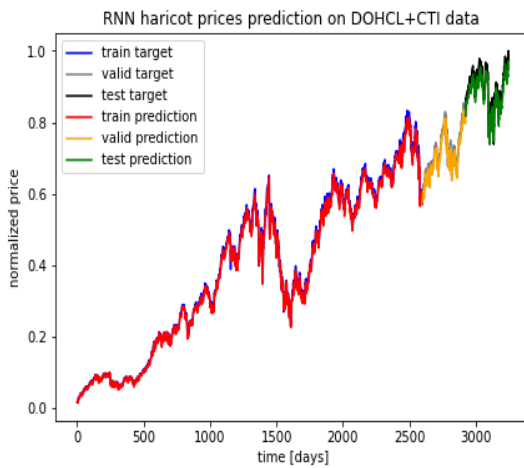
From figures plotted above, we can see that the training and validation loss of the model follows almost the same trend. Specially after epoch 2.5, both training and validation loss goes to almost zero and there is very small under-fitting of the model and this shows us that the predictive model is going in the right way. The prediction, validation and testing result for each training, validation and testing data are plotted in figure 4.6 below.

Prediction accuracy for haricote close price:
 For train/valid/test:= 0.57/0.70/0.65
 Mean Squared Error for train/valid/test
 For 49.99 epochs:= 0.000092/0.000143/0.000306



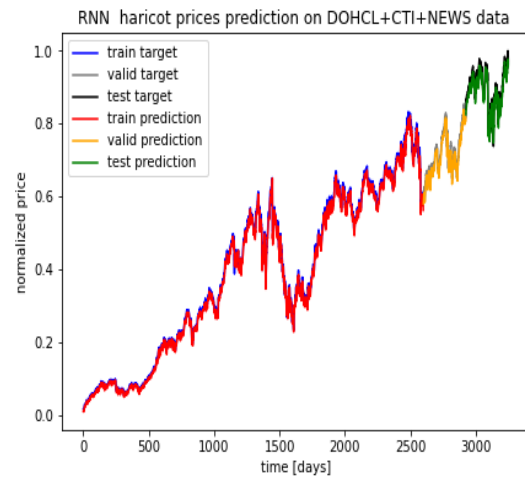
(b)

Prediction accuracy for haricote close price:
 For train/valid/test:= 0.58/0.71/0.65
 Mean Squared Error for train/valid/test
 For 49.97 epochs:= 0.000147/0.000157/0.000202



(b)

Prediction accuracy for haricote close price:
 For train/valid/test:= 0.53/0.71/0.65
 Mean Squared Error for train/valid/test
 For 49.97 epochs:= 0.000094/0.000156/0.000215



(c)

Figure 4.7 Shows the prediction of haricot bean price using RNN

When the RNN mode trained on DOHCL its predictive accuracy was 58% as shown in the figure 4.7a. However, when the selected commodity technical indicator feature added and trained again the accuracy decrease to 57%. This tells us the contribution of CTI for sesame

market price prediction is negative. During sesame market price prediction news data has little contribution as shown in the figure 4.7c

4.1.2 Long-Short Term Memory Model Building and Prediction

The other deep learning model that we build during our market price data analysis and prediction experiment is Long-Short Term Memory (LSTM) network. It is a class of RNN network in which the vanishing gradient problem is solved (see detail in **chapter Two**). During our model building all the hyper parameters are the result of the Keras tuner. The activation function is Relu on both the first layer and also on the second layer and we used the Adam optimizer. The learning rate 0.0001, epoch 20, batch size 34, time stamp 30 and for our model construction the first layer has 4 layers since we used the OHCL four input dimension but in the second layer we used 50 and next 45 layers. The dense layer is one. The final constructed model is shown in figure 3.7 below.

```

1 | model,train_error,val_error = fit_model(train,val,timesteps,h1,lr,batch_size,num_epochs)
2 | plot_error(train_error,val_error)

#####
Congratulation!! Ahmed your model is build successfully.....
#####
Model building start.....
Model: "sequential_2"

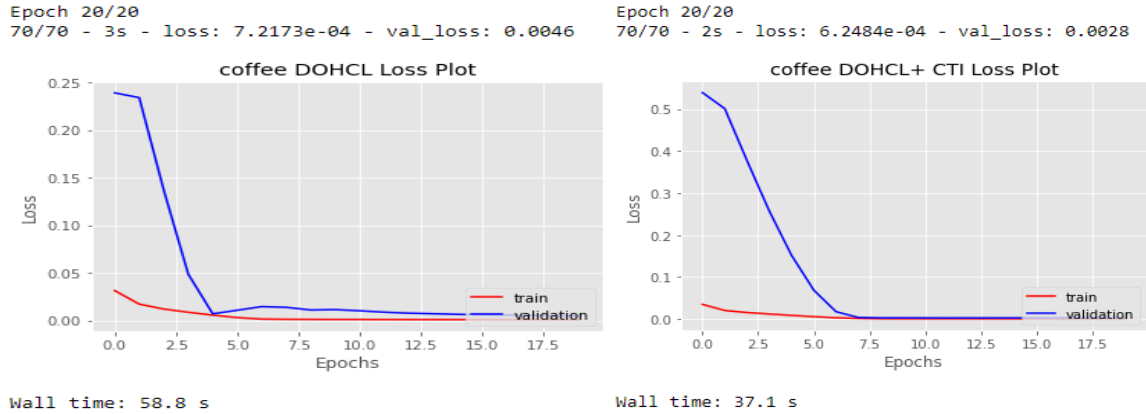
Layer (type)                 Output Shape              Param #
=====
lstm_6 (LSTM)                 (None, 30, 4)            144
lstm_7 (LSTM)                 (None, 30, 50)           11000
lstm_8 (LSTM)                 (None, 45)               17280
dense_2 (Dense)               (None, 1)                46
=====
Total params: 28,470
Trainable params: 28,470
Non-trainable params: 0

```

Figure 4.8 shows the build LSTM model

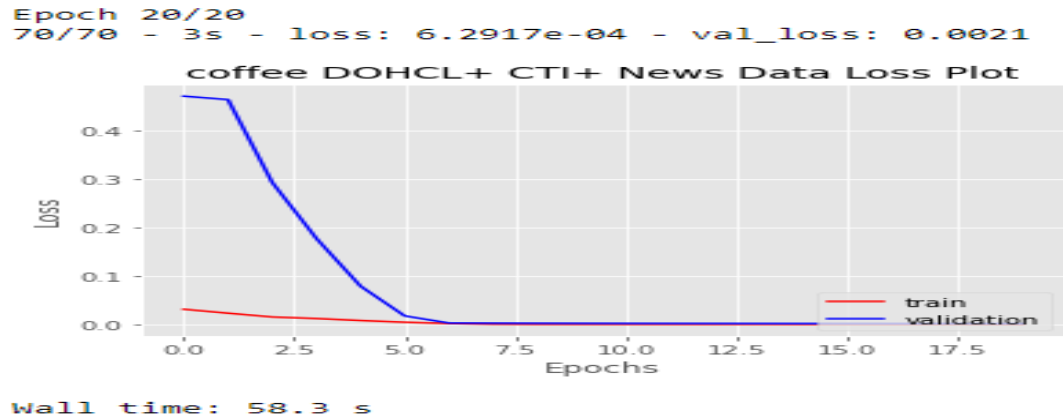
Like the previous model RNN, after we build our model, we train it using the training data taken from the three category of each commodity datasets. It has three categories of one

with only the price data (DOHCL), the second is DOHCL plus the technical indicators computed from the historical price data and the third one contains DOHCL, CTI and News dataset. In table 4.2 we have the MSE, MAE, directional accuracy (DA), time it takes for training (T(s)) and R-squared error values obtained for training coffee, sesame and haricot bean for the period of 5th October 2008 to 30th December 2017. The validation and training losses are depicted in the following figures.



(a)

(b)

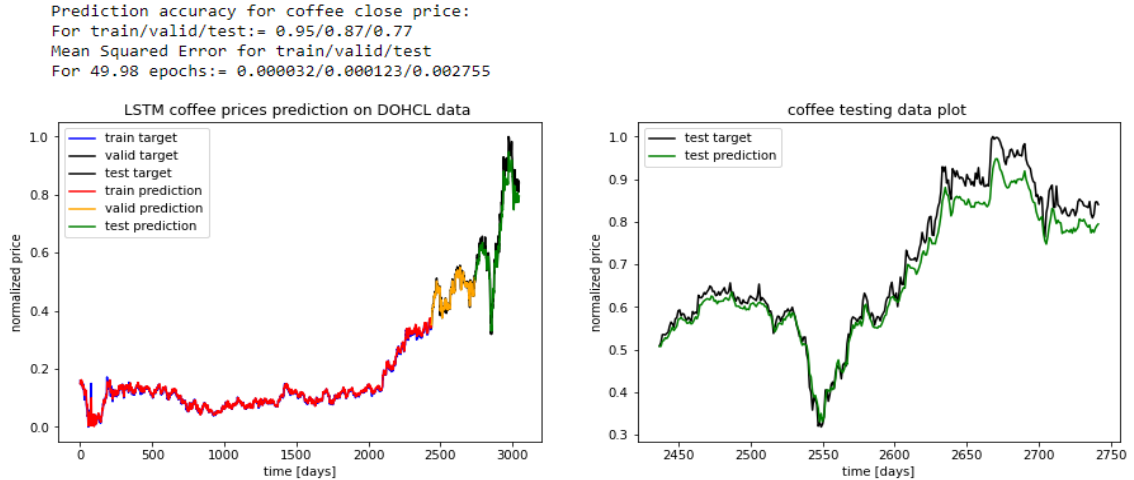


(c)

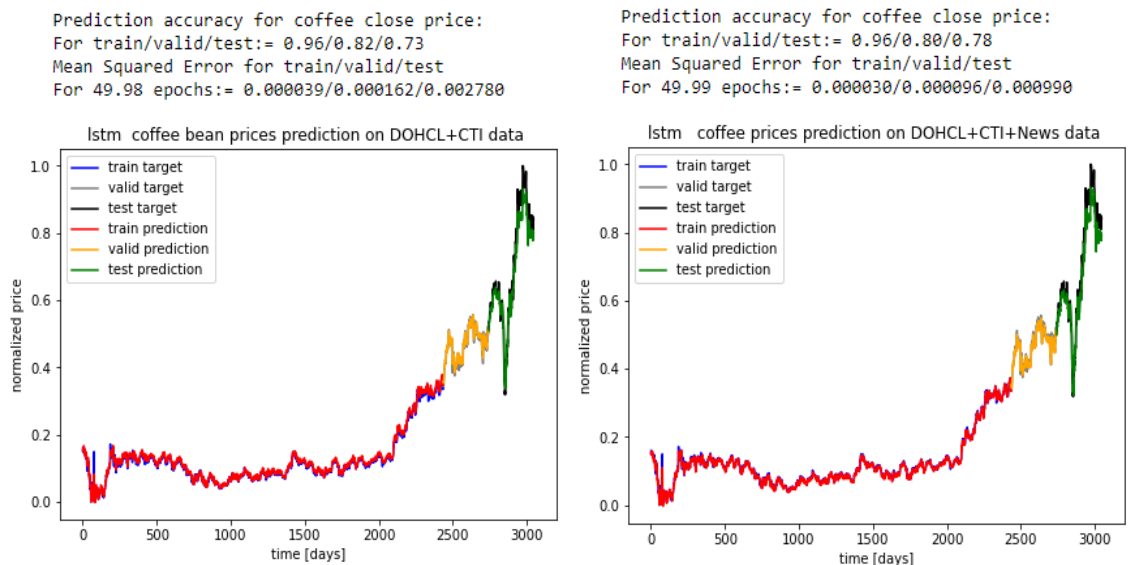
Figure 4.9 shows the training and validation loss of LSTM

From the figure 4.8 above, it is clearly seen that when the epochs increase the train and validation loss becomes closer and closer to 0. Specially on figure 4.8b and 4.8c after epoch 7.5 or around, the validation and the training losses are almost they are equal. This shows

that in our prediction model there is no under fitting or over fitting happened except for figure 4.8a. The prediction result incurred using LSTM model is shown in the figure 4.10 for each category of training and testing dataset



(a)



(b)

(c)

Figure 4.10 Shows the prediction result of LSTM

From figure 4.10, we can see that our prediction model can predict the coffee price date with greater accuracy than the previous predictive model of RNN. When the model trained

using both DOHCL and CTI its accuracy increased by 1%. The commodity technical indicators show a little contribute for the improvement of the predictive model whereas the data that comes from the new article does not bring any change to the predictive model.

The other experiment that we conducted was on sesame dataset to see the predictive performance our LSTM model. The training, validation and testing losses of the LSTM model is presented in figure 4.11.

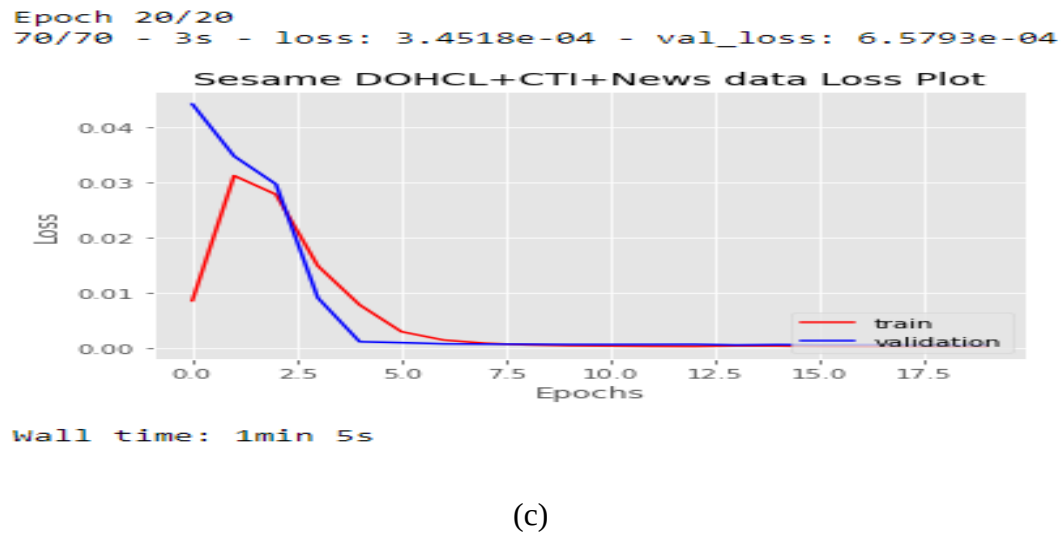
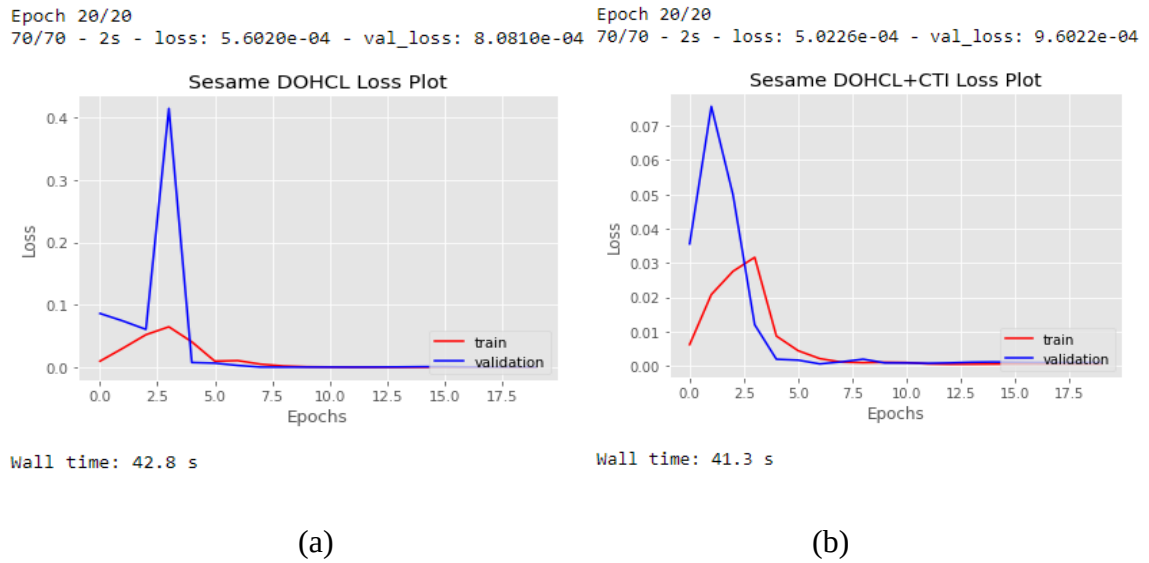
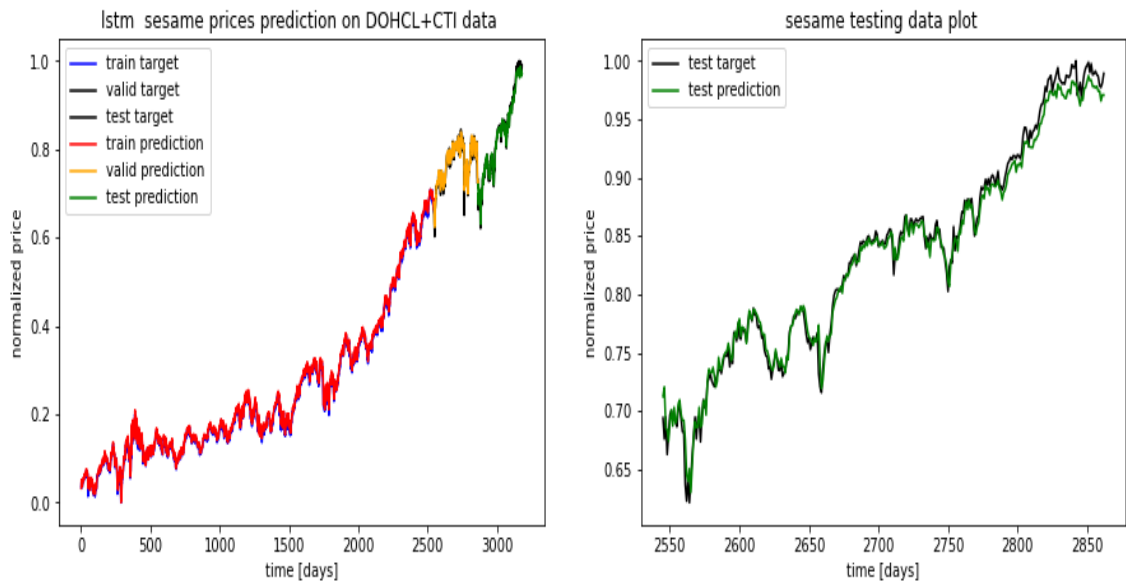


Figure 4.11 Shows training and validation loss of LSTM model on Sesame

In figure 4.11, LSTM model resulted both in over fitting and under fitting on the first 7 epochs. This is due to the reason that deep learning models need large amount of data for training to capture sufficient information from the data. The number of records in our dataset is very small compared to the standard dataset the deep learning algorithms need. After 7 epochs the training and validation losses goes together and close to zero. This shows how much is the predicted values far from the real value.

Here below are the plotted graphs of LSTM trained, validated and tested using sesame price and news headlines.

```
Prediction accuracy for sesame close price:
For train/valid/test:= 0.88/0.48/0.46
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000044/0.000177/0.000279
```



(a)

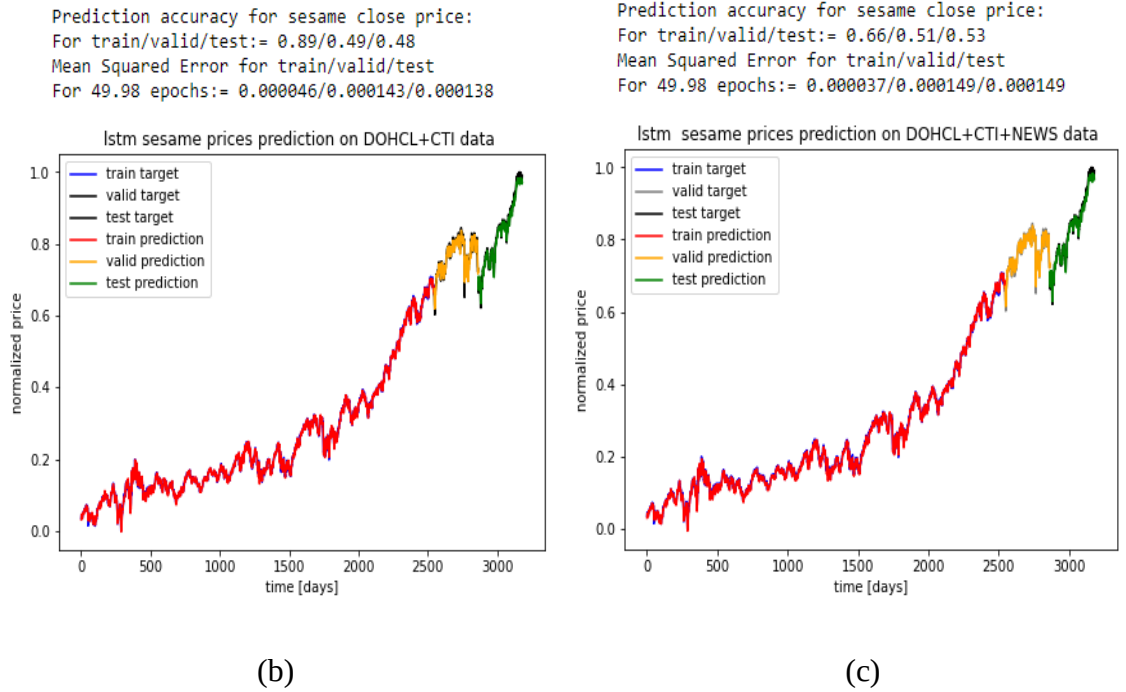
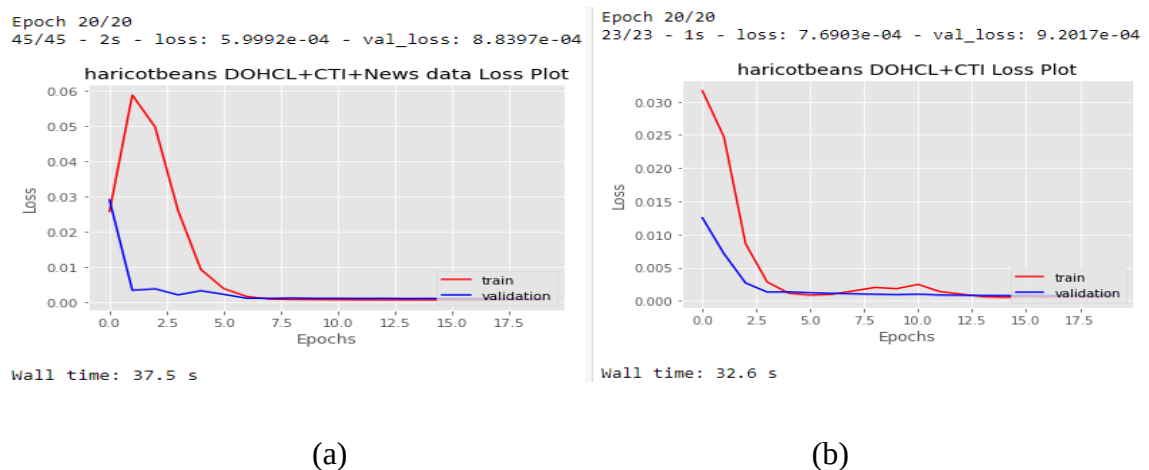
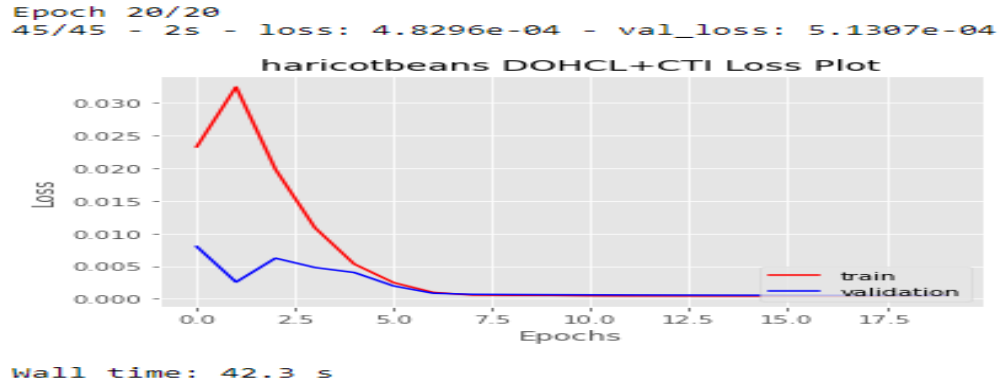


Figure 4.12 shows the prediction of Sesame price using LSTM model

As it is clearly seen from figure 4.5 prediction and analysis of sesame market price using DOHCL and technical indicators resulted in a higher accuracy whereas news data resulted in decreasing the accuracy of the model. From this, we can have understood features in news dataset are not relevant for sesame market prediction.

The third predicted and analyzed commodity was haricot beans and its training, validation and testing losses are depicted in the figure 4.13

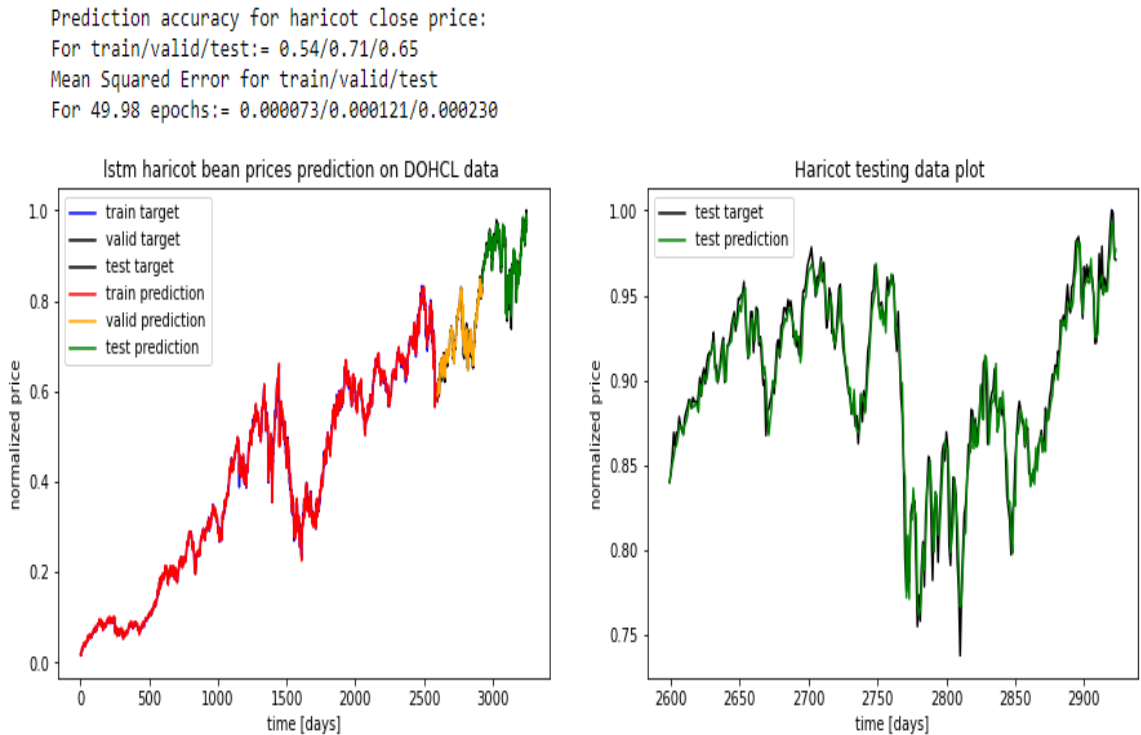




(c)

Figure 4.13 Shows the training and validation loss LSTM on Haricot Beans

From figure 4.13, it is clearly seen that for the first 5 epochs there the predictive mode goes under-fitting. Then when the epoch increases both the training and validation loss close to zero. This indicates that the predictive model goes in the right way. The plotted predicted and real values of haricot bean using our LSTM model is depicted in the figure 4.14.



(a)

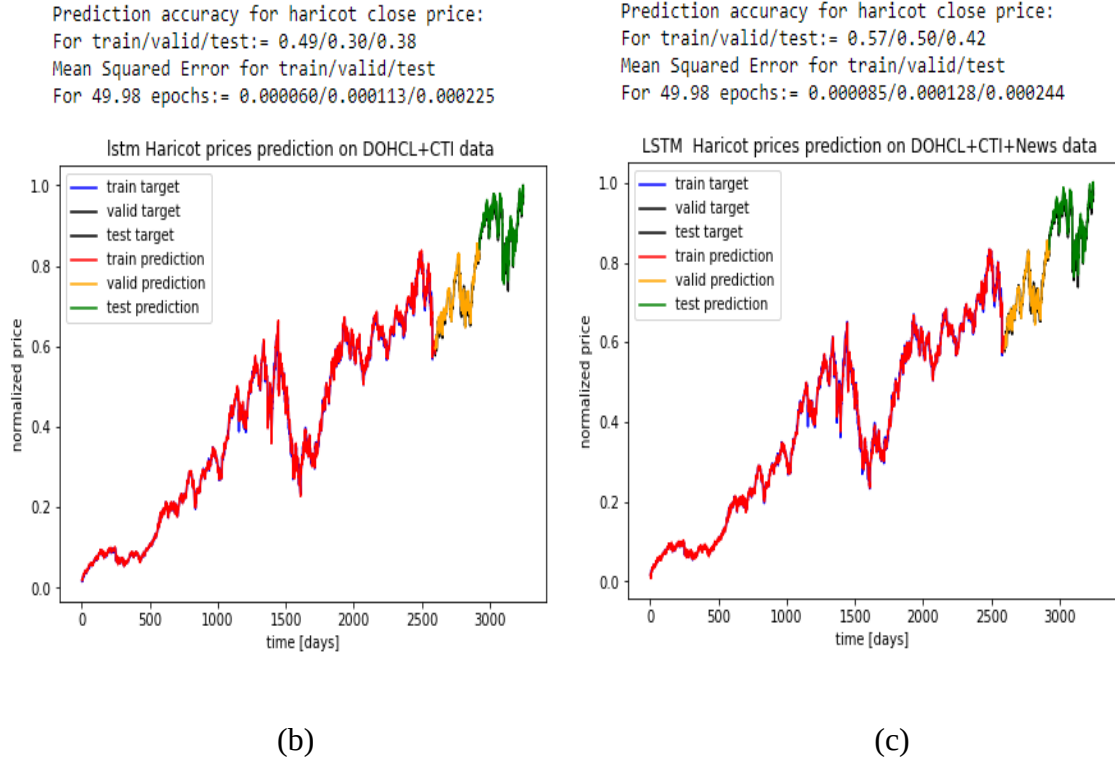


Figure 4.14 shows the prediction result of LSTM trained on Haricot beans.

Our LSTM model did not perform well on the haricot bean dataset. It failed to capture the sequence of patterns found in the dataset specially for the first 1000 days in training data and 2900 to the end for testing. The accuracy of the model decreases when the CTI features used as an input together with DOHCL, this shows that during prediction CTI features were irrelevant for the model.

4.1.3 Convolutional Neural Network Model Building and Prediction

The third deep learning model included in our research experiment is Convolutional neural network (CNN). It is a type of neural networks which have been inspired by the neurological concept of receptive fields and shows much success in the domain of computer vision and image classification (see **chapter two**). To build our CNN model all the hyper parameters are determined by the automated hyper parameter optimizer called Keras tuner. The max-pooling and conv1D has 256 units or hidden layer. For epoch 50 0.001 learning rate, Relu as an activation function, dense layer of 1 for out put and a dropout of 0.1 is used. The input dimension is 4 (OHCL) and we trained the model using the three

category of training dataset. The basic structure of the CNN that we build is depicted below in the figure 4.15 followed by its losses and predicted results graphs.

The created CNN model trained on the given three category of the training dataset and tested. During training the CNN model, the training loss, training accuracy, time it takes for training and other performance metrics are considered to make a comparative analysis with rest of the model that we developed and used in our thesis work.

The created model was trained, validated and tested using the three categorical features from each commodity dataset. The train and validation of coffee price incurred from the experiment is depicted in the figure 4.16.

```

1 model,train_error,val_error = fit_model(train,val,timesteps,h1,lr,batch_size,num_epochs)
2 plot_error(train_error,val_error)

```

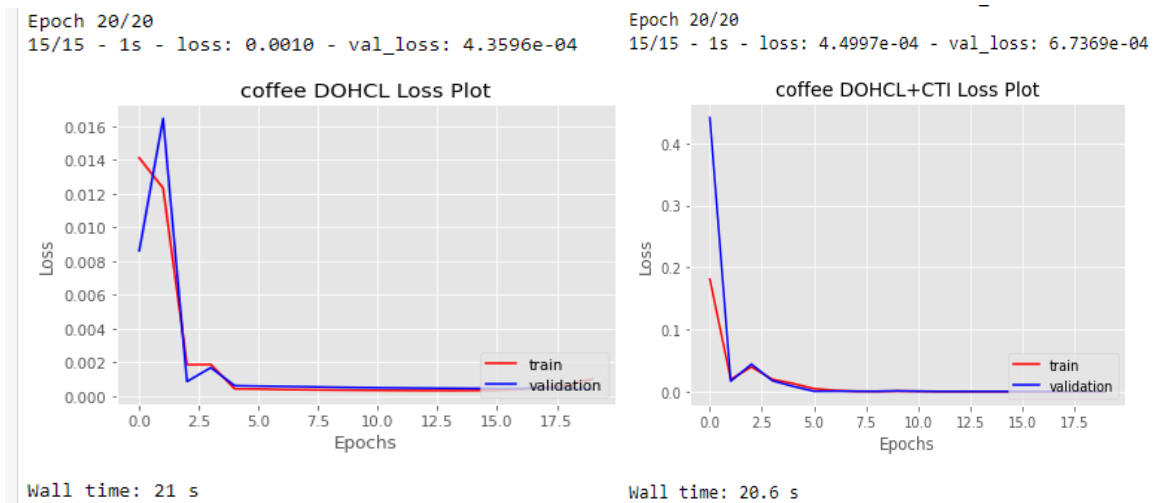
Ahmed the model is created. Congratulation.....
Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 29, 256)	2304
max_pooling1d (MaxPooling1D)	(None, 14, 256)	0
flatten (Flatten)	(None, 3584)	0
dense (Dense)	(None, 50)	179250
dense_1 (Dense)	(None, 45)	2295
dense_2 (Dense)	(None, 1)	46
dense_3 (Dense)	(None, 1)	2

=====
Total params: 183,897
Trainable params: 183,897
Non-trainable params: 0
=====

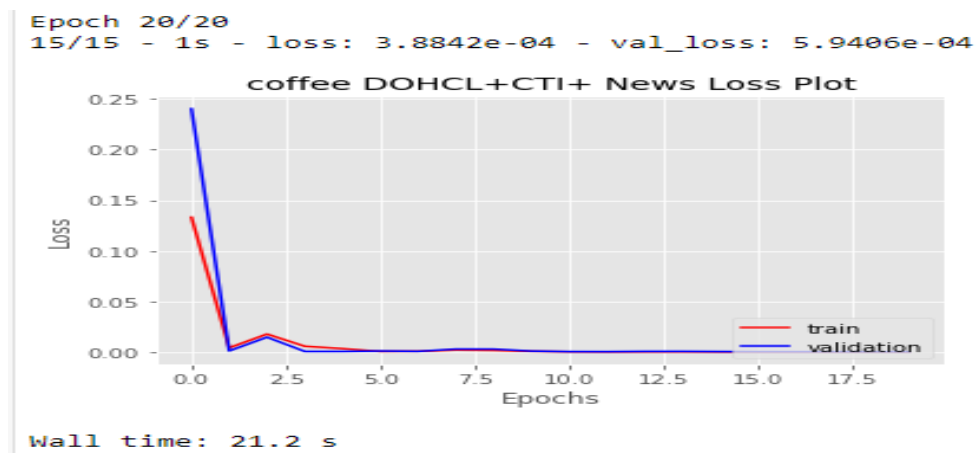
None
Ahmed train your model here
Epoch 1/20
70/70 [=====] - 4s 39ms/step - loss: 0.0154 - val_loss: 0.0077
- . . . -

Figure 4.15 Shows the basic CNN model created



(a)

(b)

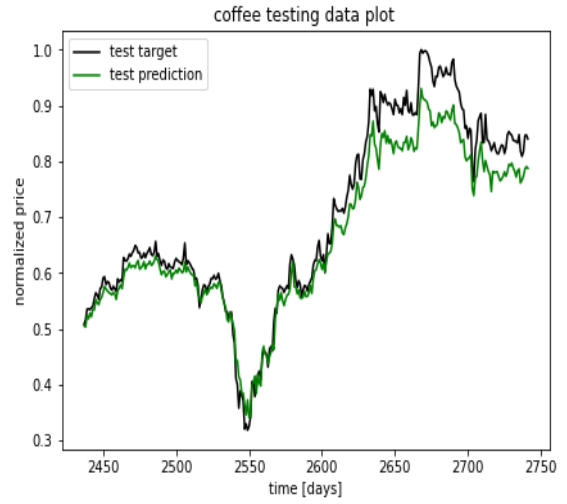
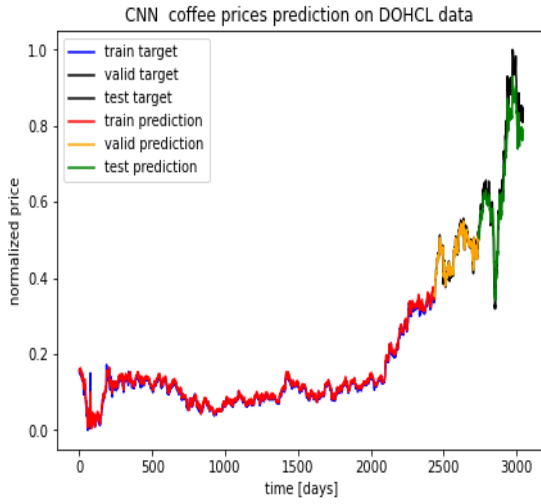


(c)

Figure 4.16 shows the training and validation loss of CNN for coffee

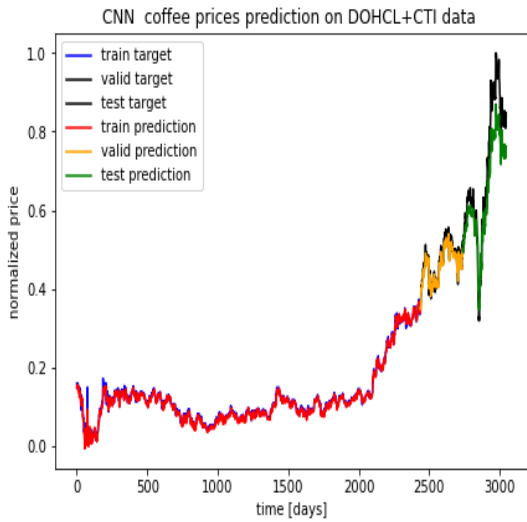
In deep learning model, over fitting happened when the validation loss is less than the training loss of the given model and the under-fitting will be happened when the training loss is less and far from the validation loss. The best fit model is a model where its validation loss is equal to the training loss. In figure 4.16a both over fitting and under-fitting was happened until epoch 3. After epoch three our model becomes stable and goes in the right way.

Prediction accuracy for coffee close price:
For train/valid/test:= 0.96/0.89/0.80
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000030/0.000304/0.006107



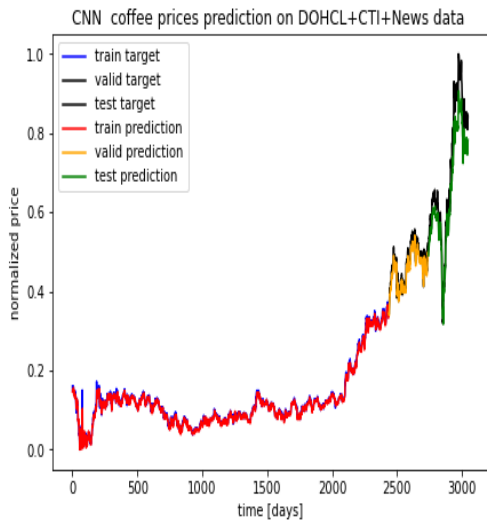
(a)

Prediction accuracy for coffee close price:
For train/valid/test:= 0.94/0.87/0.42
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000035/0.000373/0.004478



(b)

Prediction accuracy for coffee close price:
For train/valid/test:= 0.95/0.89/0.72
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000040/0.000404/0.003706

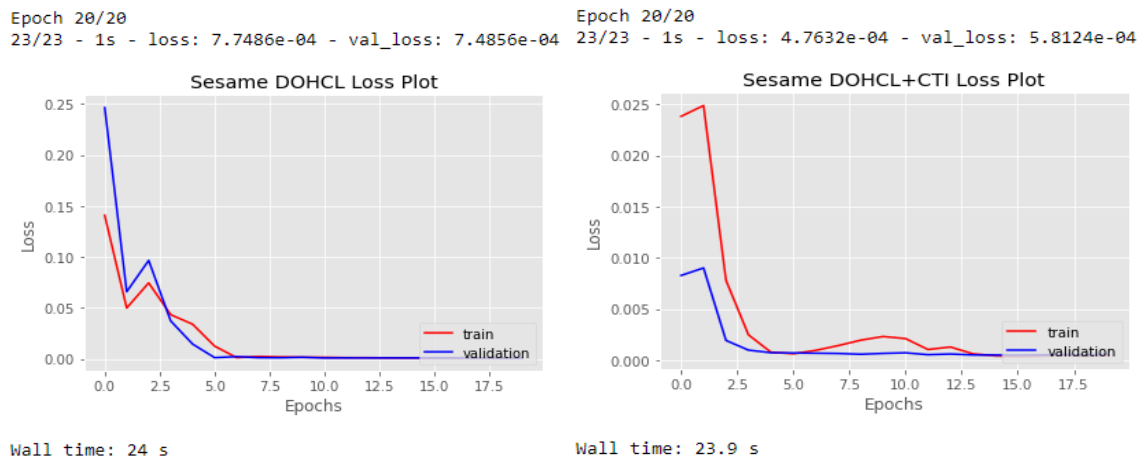


(c)

Figure 4.17 shows the predicted and real values of coffee using CNN

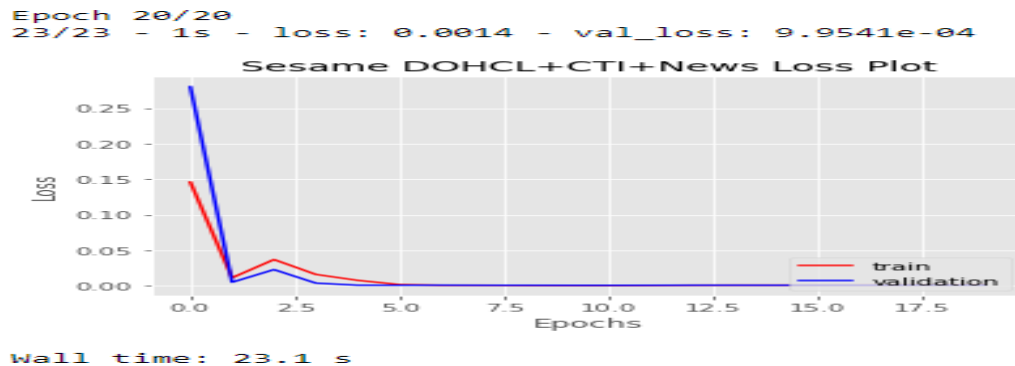
In figure 4.17, it is clearly seen that CNN model able to catch almost all the training data. However, for validation and testing, it failed to capture the sequence of patterns in the dataset. In figure 4.17b, the model performed well on the validation dataset whereas, in figure 4.17c, it was successful in identifying the real time series patterns. This shows that the input features from commodity technical indicators and news data contributes a lot to the improvement of the model's prediction performance.

The second experiment was conducted on Sesame price using our CNN model. The model was trained, validated and tested on the three categories of the input features. The training and validation loss of the model is depicted in the figure 4.18 followed by the graph of predicted and real values of the dataset in the figure 4.19.



(a)

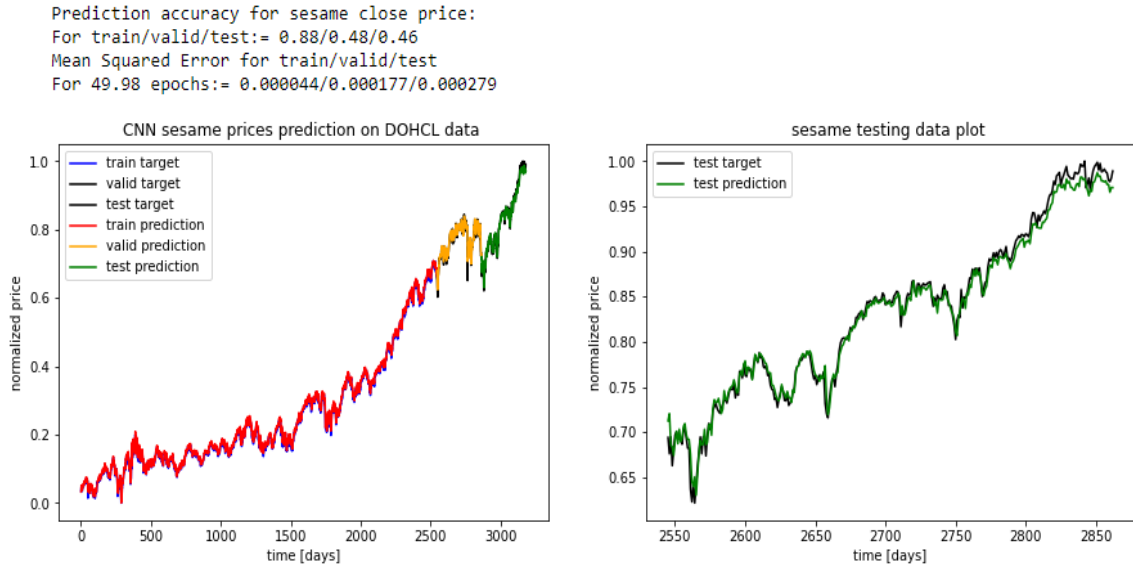
(b)



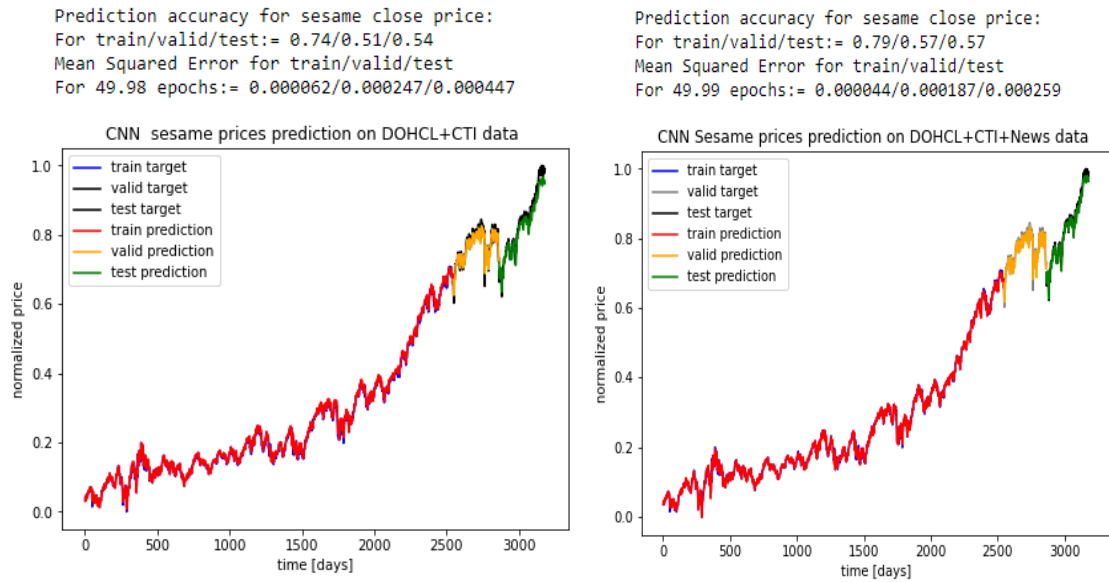
(c)

Figure 4.18 Shows the training and validation loss of Sesame using CNN

However, for figure 4.18a and 4.18c, after epoch 5, the two losses are overlapped and there is not any under or over fitting except in figure 4.18 b which takes 15 epochs to become the best fit model.



(a)



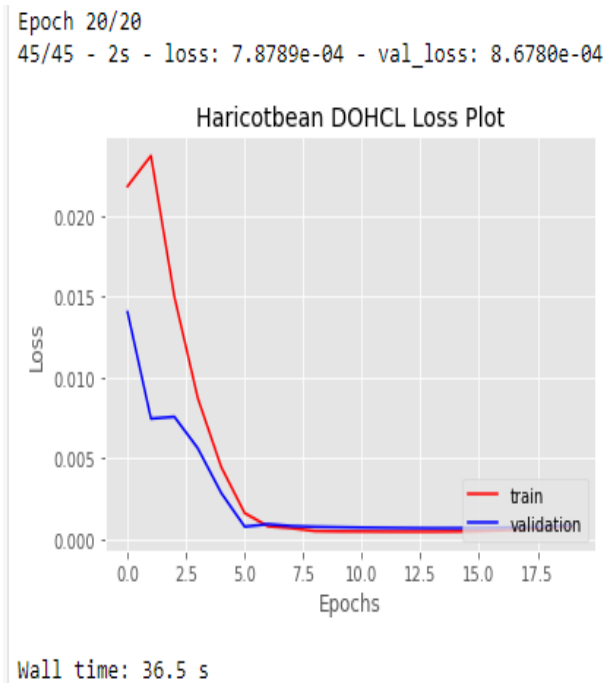
(b)

(c)

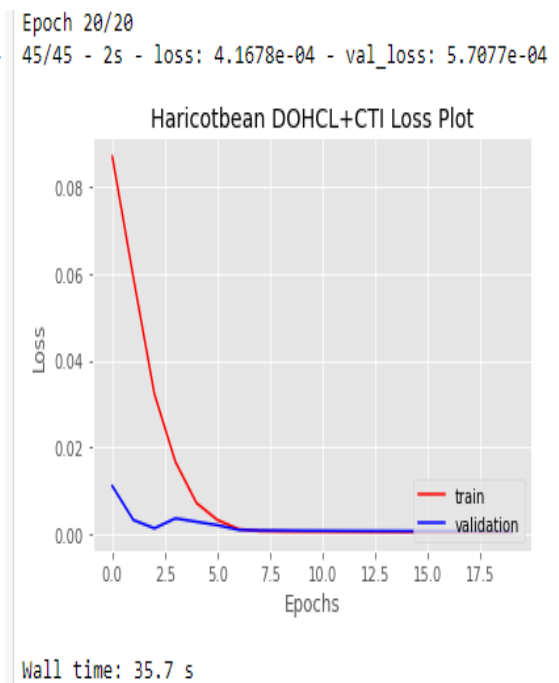
Figure 4.19 show the predicted and real values of sesame using CNN

From figure 4.19 we can see that there is 13% change in the model prediction performance when the model trained using both DOHCL and CTI. From this, we can have understood that the commodity technical indicator features contribute a lot to the model performance. In figure 4.19b, there is no change in accuracy of the mode mean that input feature of news data is neural to the model.

Our CNN model also used to in the analysis and prediction of haricot beans market price. The input features related to Haricot beans were given to our CNN model as an input features and after trained using the training data it was validated and tested using validation and testing dataset of haricot beans. The experimental results are depicted in the figure 4.20 and figure 4.21

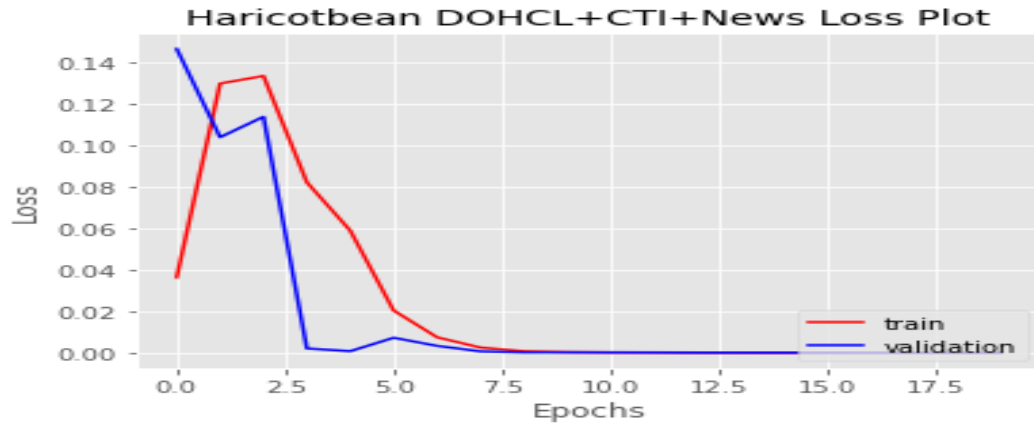


(a)



(b)

Epoch 20/20
45/45 - 2s - loss: 4.8255e-04 - val_loss: 5.7221e-04



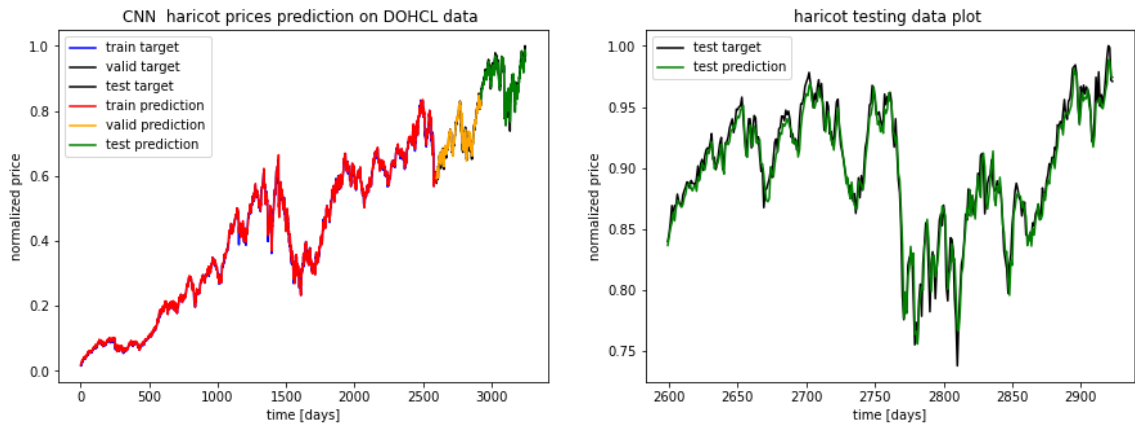
Wall time: 37.6 s

(c)

Figure 4.20 show the training and validation losses of CNN for Haricot Bean

From the figure 4.20, it is clearly seen that the CNN model resulted in an over fitting for the first 5.5 epochs. In figure 4.20 both over fitting and under-fitting was seen for the first few epochs. After 7.5 epoch the CNN model performed well for all input features.

Prediction accuracy for haricot close price:
For train/valid/test:= 0.54/0.70/0.62
Mean Squared Error for train/valid/test
For 49.97 epochs:= 0.000069/0.000122/0.000290



(a)

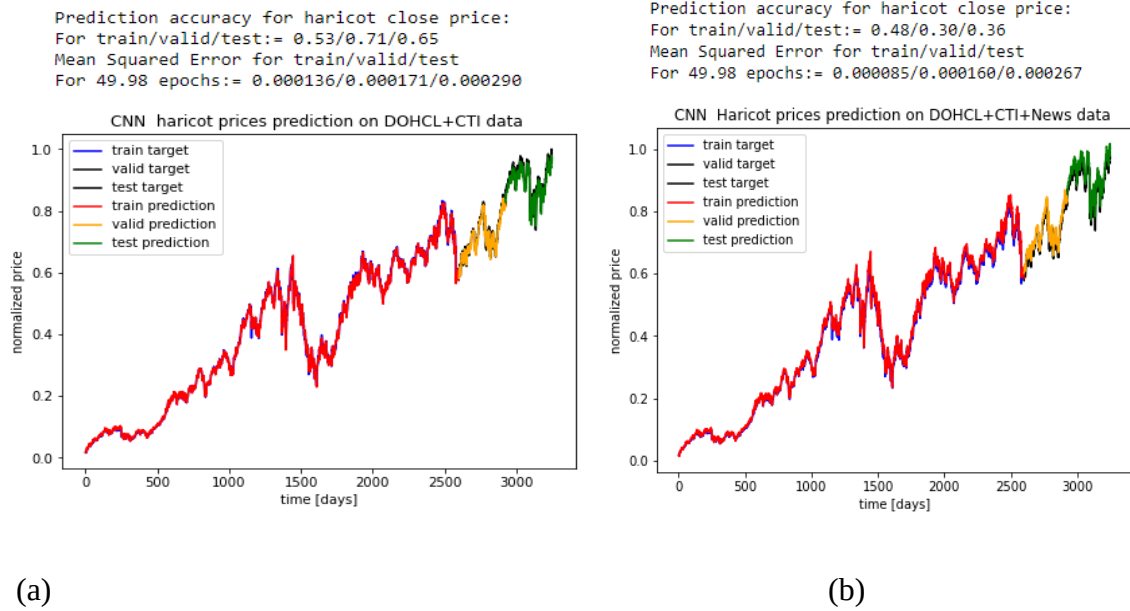


Figure 4.21 Shows the predicted and real value of haricot bean using CNN

As it clearly seen from the figure, our CNN model prediction performance was not good at all. The model can predict with a very low accuracy specially when it was trained using DOHCL. From the figure 4.21b, when the training repeated using CTI as additional input feature the model performance increased by 21%. This show that the input features of technical indicators contribute a lot to the improvement of the performance of the model. For haricot bean price prediction input features from News data did not affect the performance of the prediction model as it is shown in the figure 4.21c.

4.1.4 CNN-LSTM Model Building and Prediction

The last deep learning model that we build to make market price analysis and prediction was the hybrid of convolutional and long-short term memory. Our main goal of building CNN-LSTM model is the take the best advantages of both models. as it is discussed in the article review section (see **chapter two**) CNN model is popular in extraction of important feature from the given data and LSTM is designed for handling both linear and non-linear property of the time series data and it avoids the vanishing gradient problem.

During model construction, we used 64 input layers with kernel size of 3 and Maxpooling with size 2. Then, the LSTM layer, which has 100 hidden neurons, takes the output of the

Maxpooling layer as an input. Next, the one-dimensional convolution with 32 hidden layers and kernel size of 2 was applied on the output of LSTM layer and Maxpooling layer comes again with pool size of 2. We used 1 unit for dense layer and tanh as an activation function.

```
1 model,train_error,val_error = fit_model(train,val,timesteps,h1,lr,batch_size,num_epochs)
2 plot_error(train_error,val_error)
```

Ahmed! You build your model successfully. here it is
Model: "sequential_1"

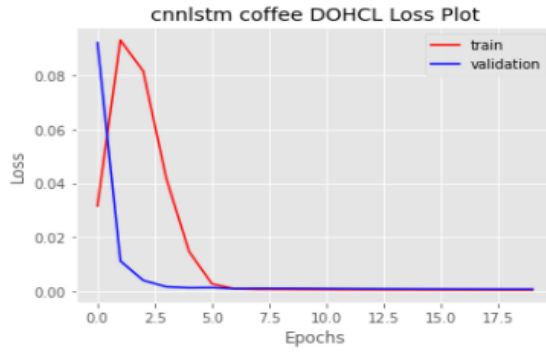
Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 30, 64)	832
max_pooling1d_2 (MaxPooling1D)	(None, 15, 64)	0
lstm_1 (LSTM)	(None, 15, 100)	66000
dropout_1 (Dropout)	(None, 15, 100)	0
conv1d_3 (Conv1D)	(None, 15, 32)	9632
max_pooling1d_3 (MaxPooling1D)	(None, 7, 32)	0
flatten_1 (Flatten)	(None, 224)	0
dense_1 (Dense)	(None, 1)	225
activation_1 (Activation)	(None, 1)	0

Total params: 76,689
 Trainable params: 76,689
 Non-trainable params: 0

Figure 4.22 Show basic CNN-LSTM model

Using the model depicted in the figure 4.22, we conducted a price prediction experiment on coffee price. The model trained, validated and tested using the training, validation and testing dataset of coffee price. The result obtained from the experiment is depicted in the following figure. First let we see the training and validation losses that are depicted in the figure 4.23.

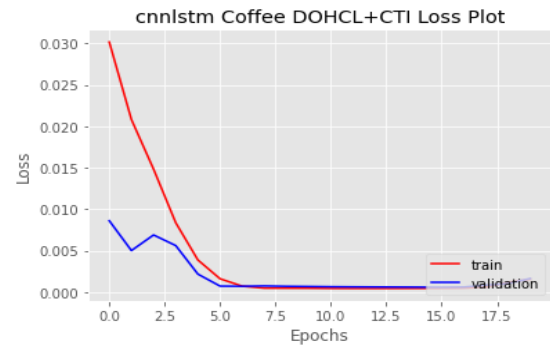
Epoch 20/20
45/45 - 2s - loss: 4.4156e-04 - val_loss: 6.8006e-04



Wall time: 55.2 s

(a)

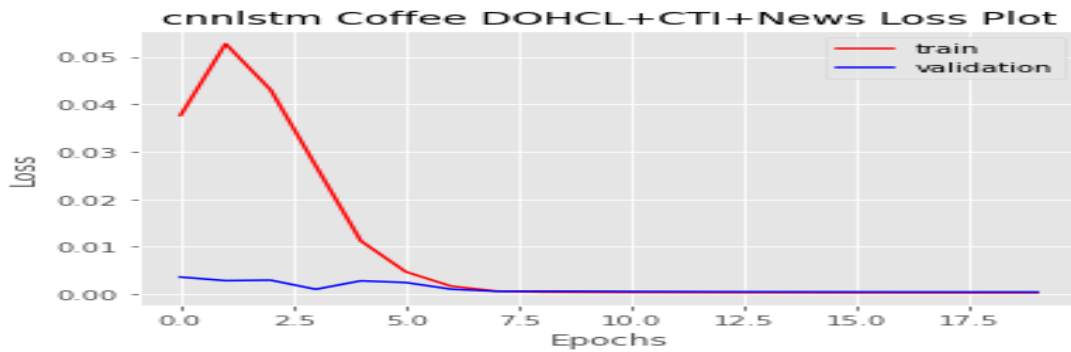
Epoch 20/20
45/45 - 2s - loss: 0.0015 - val_loss: 0.0016



Wall time: 40.2 s

(b)

Epoch 20/20
45/45 - 2s - loss: 3.3830e-04 - val_loss: 4.6496e-04



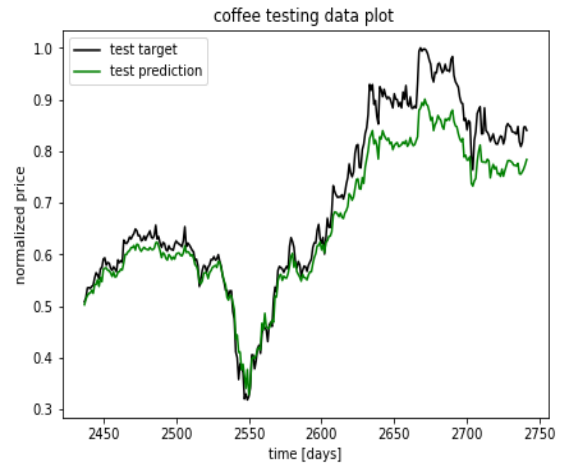
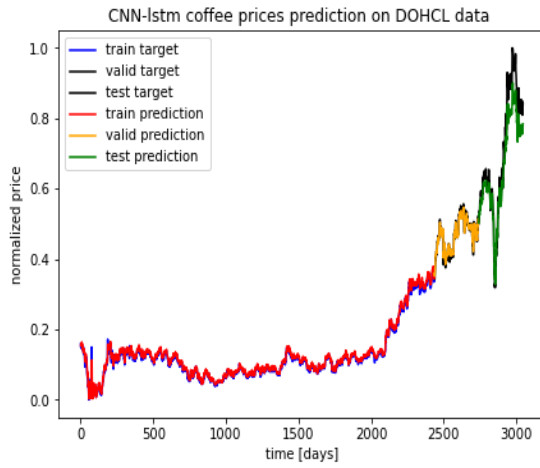
Wall time: 40.6 s

(c)

Figure 4.23 shows the training and validation loss of CNN-LSTM on coffee price

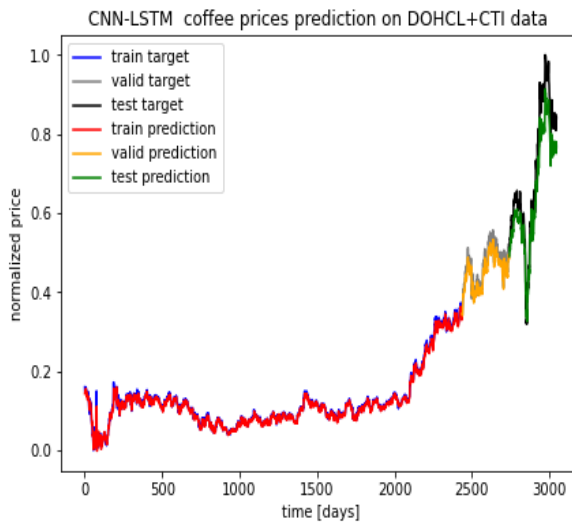
From the figure 4.23, We notice that the training loss and validation loss are not correlated for the first 5 epochs. This means the as the training loss increases and then is decreasing, the validation loss also decreases over the iterations as shown in the figure 4.23a and 4.23b. This means that the model is not exactly improving, but is instead overfitting the training data. However, after 5 epochs both the training and validation loss decreasing. there is clearly a health correlation between training loss and the validation loss.

Prediction accuracy for coffee close price:
For train/valid/test:= 0.96/0.89/0.80
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000035/0.000255/0.004614



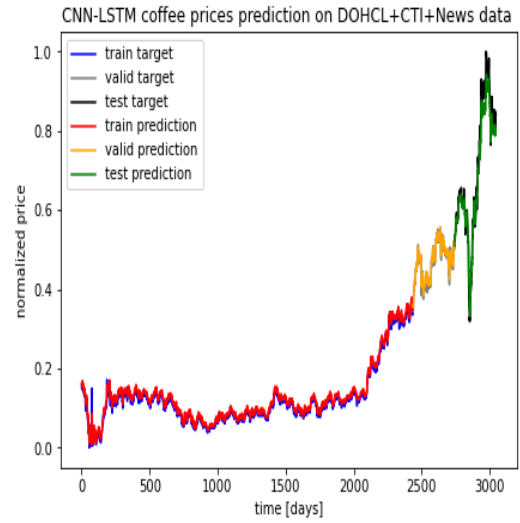
(a)

Prediction accuracy for coffee close price:
For train/valid/test:= 0.95/0.87/0.81
Mean Squared Error for train/valid/test
For 49.98 epochs:= 0.000043/0.000533/0.005879



(b)

Prediction accuracy for coffee close price:
For train/valid/test:= 0.96/0.89/0.81
Mean Squared Error for train/valid/test
For 49.99 epochs:= 0.000051/0.000460/0.003477

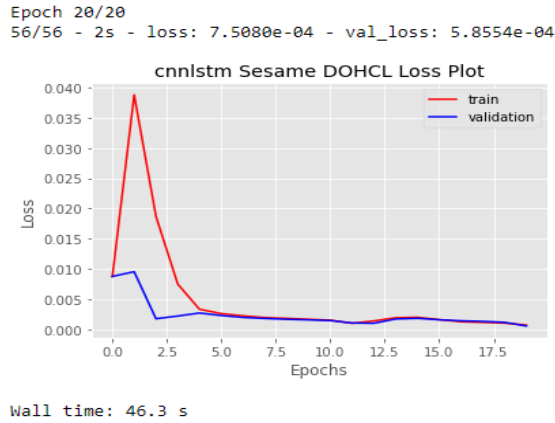


(c)

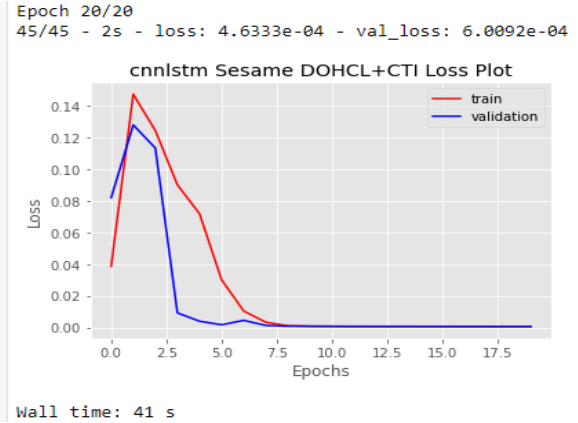
Figure 4.24 Show the predicted and real of coffee dataset using CNN-LSTM

From the graph 4.24, we noticed that the input features of commodity technical indicators and new data contribute a lot to the improvement of the prediction accuracy of the model.

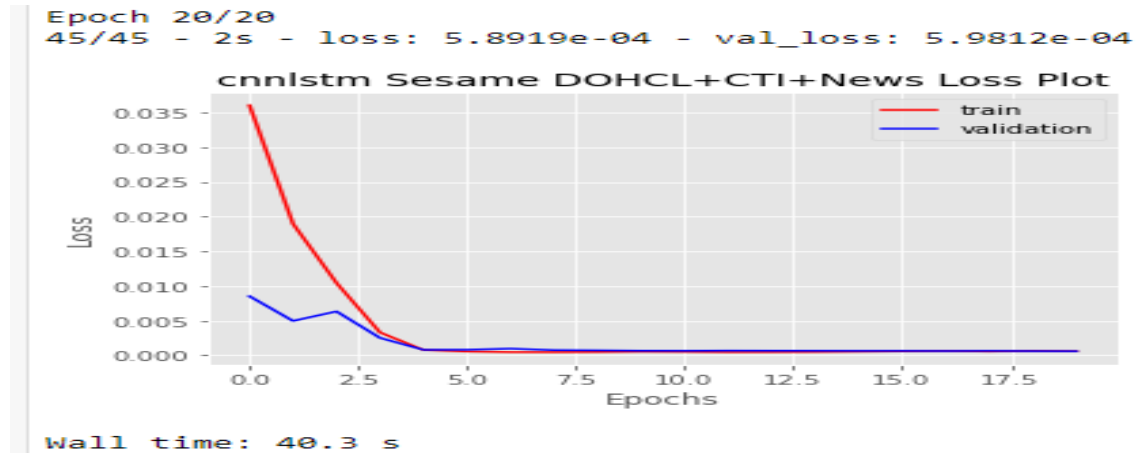
The other experiment was on the sesame price using CNN-LSTM model. It was trained, validated and tested using sesame daily price, technical indicators and market new data. The train loss and validation loss incurred from the experiment is depicted in the figure 2.25 below.



(a)



(b)

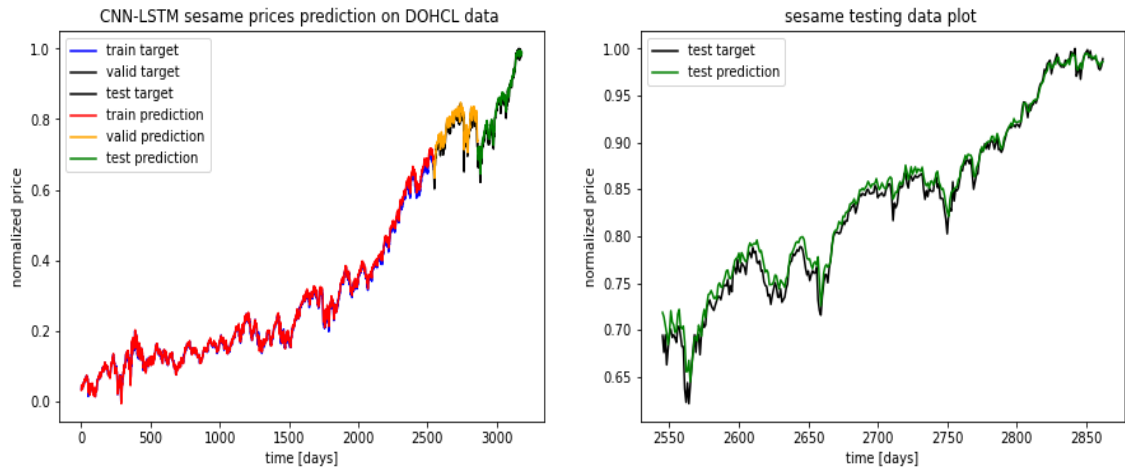


(c)

Figure 2.25 train and validation loss of CNN-LSTM on sesame

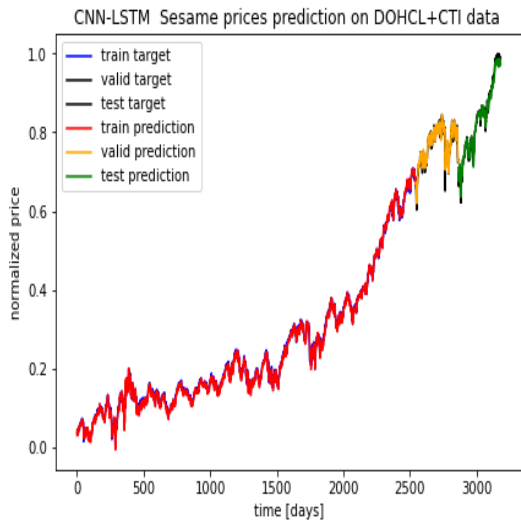
Form the experimental result depicted in the figure 4.25, we notice that the training loss increases and then decreases in the figure 4.25a and 4.25b for epochs of 2.5. this mean that the train loss and validation loss have no correlation and indicates that our model is not healthy. However, when the epochs increase both the train and validation loss decreases and close to zero.

Prediction accuracy for sesame close price:
 For train/valid/test:= 0.86/0.51/0.47
 Mean Squared Error for train/valid/test
 For 49.98 epochs:= 0.000053/0.000330/0.000436



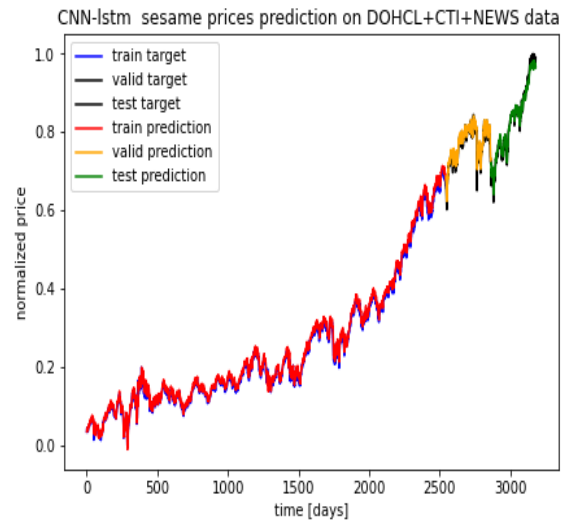
(a)

Prediction accuracy for sesame close price:
 For train/valid/test:= 0.70/0.55/0.49
 Mean Squared Error for train/valid/test
 For 49.98 epochs:= 0.000050/0.000353/0.000646



(b)

Prediction accuracy for sesame close price:
 For train/valid/test:= 0.73/0.52/0.54
 Mean Squared Error for train/valid/test
 For 49.98 epochs:= 0.000096/0.000249/0.000146

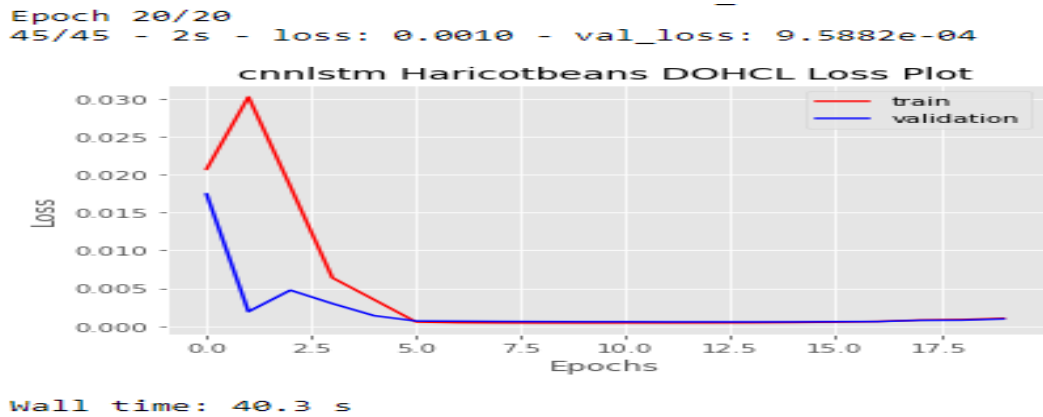


(c)

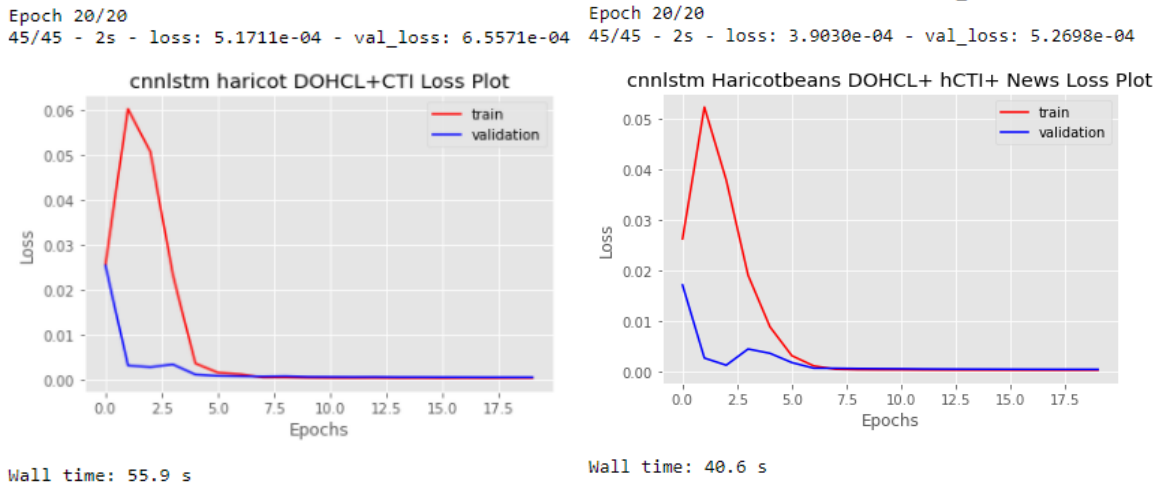
Figure 4.26 shows the predicted and real dataset of sesame using CNN-LSTM

As it is clearly show in the figure 4.26, the mean squared error incurred from the CNN-LSTM model using DOHCL was 0.000037 or 3.7×10^{-5} . The accuracy also has 1% change when the model was trained with DOHCL and CTI dataset.

The last experiment conducted using CNN-LSTM model was on the haricot daily market price dataset. The, was splited in to train, validation and testing dataset. The model trained, validated and tested on train validation and test dataset. The train and validation losses of the CNN-LSTM trained and validated is depicted in the figure 4.27 below.



(a)



(a)

(b)

(c)

Figure 4.27 Train and Validation loss of CNN-LSTM on haricot bean dataset

In figure 4.27a and c the train loss increases once and decrease while the validation loss decrease, increase and then decrease for epochs 0 to 2.5. This means the training and validation loss has no correlation. After epoch 5.0 the correlation between train and validation loss becomes high. In figure 4.27b the training and validation losses have inverse

relation for the first 2.7 epochs. However, after epoch 5, the two losses show high correlation.

The predicted and real dataset of haricot bean using CNN-LSTM model is shown in the figure 2.28. there are three figures for each category of the input features and for each category of the input feature we plotted the following three figures.

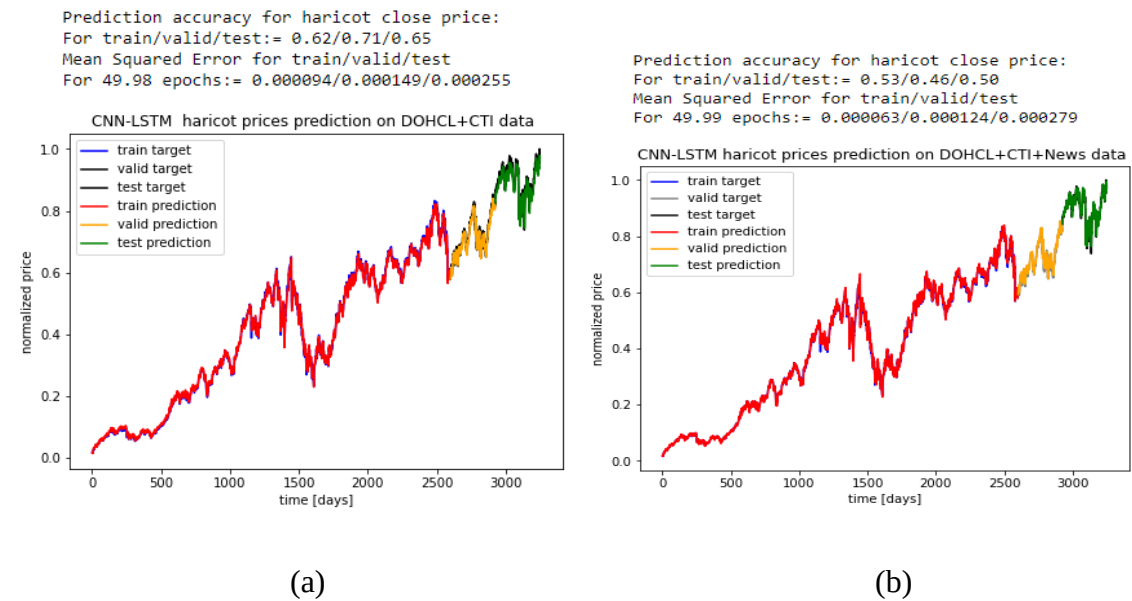
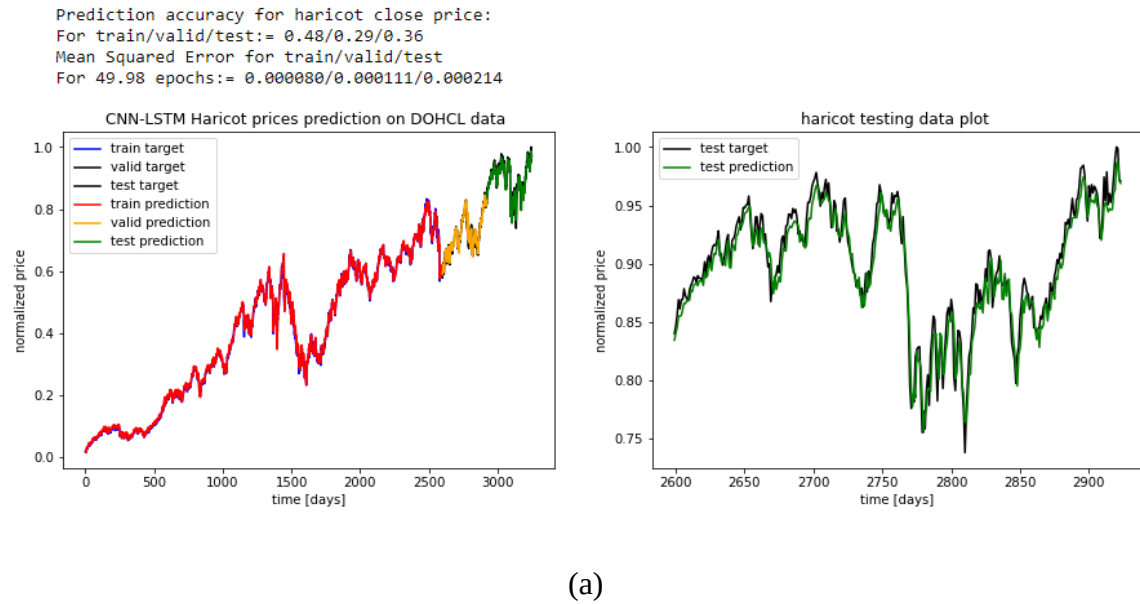


Figure 4.28 predicted and real dataset of Haricot bean using CNN-LSTM

During prediction of haricot bean using CNN-LSTM model the mean squared errors are 8×10^{-5} , 1.1×10^{-4} and 1.2×10^{-4} on the training, validation and testing dataset. This shows that how much is the predicted and real datasets are close each other. but the accuracy of the model was not that much satisfactory. This may due to the nature of the dataset.

Finally, to answer the first research question, the overall experimental results from the above-mentioned deep learning models is tabulated in the table 4.1. The table shows the result of prediction performance measures of mean squared error, mean absolute error, directional accuracy and the wall time for execution for the selected deep learning models on the three commodities that we used in our experiment.

Table 4.1 Shows performance measures of four models on the three commodities.

Model	Commodity	Input features	MSE (10^{-4})	MAE	DA (%)	T(s)
RNN	Coffee	DOHCL	5.3	0.15	89	79
		DOHCL+CTI	3.0	0.05	89	57.4
		DOHCL+CTI+News	3.9	0.04	89	36.4
	Sesame	DOHCL	2.0	0.31	55	73
		DOHCL+CTI	1.8	0.05	53	15.9
		DOHCL+CTI+News	1.8	0.025	78	75
	Haricot beans	DOHCL	1.4	0.46	70	42.3
		DOHCL+CTI	1.5	0.055	51	41.8
		DOHCL+CTI+News	1.5	0.02	71	76
LSTM	coffee	DOHCL	1.2	0.15	75	58.8
		DOHCL+CTI	1.6	0.81	87	37.1
		DOHCL+CTI+News	0.96	0.11	88	58.3
	Sesame	DOHCL	1.7	0.32	52	42.8
		DOHCL+CTI	1.43	0.21	51	41.3
		DOHCL+CTI+News	1.49	0.11	52	65
	Haricot Bean	DOHCL	1.2	0.46	70	37.5
		DOHCL+CTI	1.3	0.41	62	32.6
		DOHCL+CTI+News	2.4	0.01	61	42.3
	Coffee	DOHCL	3.0	0.12	75	21
		DOHCL+CTI	3.7	0.56	88	20
		DOHCL+CTI+News	4.0	0.012	89	21

CNN	Sesame	DOHCL	1.7	0.52	54	24
		DOHCL+CTI	2.4	0.91	58	23.9
		DOHCL+CTI+News	1.8	0.041	56	23.1
	Haricot bean	DOHCL	1.2	0.72	50	36.5
		DOHCL+CTI	1.7	0.35	40	35.7
		DOHCL+CTI+News	1.06	0.125	66	37.6
CNN-LSTM	Coffee	DOHCL	2.5	0.95	68	55.2
		DOHCL+CTI	5.3	2.01	89	40.2
		DOHCL+CTI+News	4.6	0.91	88	40.6
	Sesame	DOHCL	3.3	0.12	49	46.3
		DOHCL+CTI	3.5	0.115	50	41
		DOHCL+CTI+News	2.4	0.11	50	40.3
	Haricot bean	DOHCL	1.1	0.56	71	40.3
		DOHCL+CTI	1.4	0.49	71	55.4
		DOHCL+CTI+News	1.1	0.15	71	40.6

From table 4.1 we can see that the lowest MSE is obtained when the LSTM model was trained using coffee price data of DOHCL+CTI and NEWS dataset together. The MAE and directional accuracy also highest compared with others. In case of Coffee the lowest MSE and the highest directional accuracy with minimum training time is obtained when the model was trained with DOHCL, CTI and News dataset which is 9.6×10^{-5} , 0.05, 93 and 1.11 respectively. For haricot bean the lowest MSE and Highest directional accuracy is recorded when the model was trained using only DOHCL dataset.

Since the experiment is prediction, to make a comparative analysis and select the best fit regression model that we used in our experiment, we used the mean squared error (MSE) from the 4.1. MSE measures the average of squares of the errors. The difference between the actual value and the predicted value is termed as an error. The mean squared error incurred from the previous experiment is recorded in the table 4.2. Each columns of the table contain the mean squared error of each models and rows contain the input features for the three selected commodities.

Table 4.2 shows the mean squared (MSE) error incurred during the experiments

	Inputs	RNN	LSTM	CNN	CNN-LSTM
Coffee	DOHCL	5.3×10^{-4}	1.2×10^{-4}	3.0×10^{-4}	2.5×10^{-4}
	DOHCL+CTI	3×10^{-4}	1.6×10^{-4}	3.7×10^{-4}	5.3×10^{-4}
	DOHCL+CTI+News	3.9×10^{-4}	0.96×10^{-4}	4.0×10^{-4}	1.6×10^{-4}
Sesame	DOHCL	2.0×10^{-4}	1.7×10^{-4}	1.7×10^{-4}	3.3×10^{-4}
	DOHCL+CTI	1.8×10^{-4}	1.43×10^{-4}	2.4×10^{-4}	3.5×10^{-4}
	DOHCL+CTI+News	1.8×10^{-4}	2.4×10^{-4}	1.8×10^{-4}	2.4×10^{-4}
Haricot bean	DOHCL	1.4×10^{-4}	1.2×10^{-4}	1.2×10^{-4}	1.10×10^{-4}
	DOHCL+CTI	1.5×10^{-4}	1.3×10^{-4}	1.7×10^{-4}	1.4×10^{-4}
	DOHCL+CTI+News	1.5×10^{-4}	2.4×10^{-4}	1.6×10^{-4}	1.2×10^{-4}

In the table 4.2, it is clearly seen that all the selected deep learning model perform well on the Ethiopian commodity price dataset. The highest mean squared error was recorded for models RNN and CNN-LSTM. Both models have 5.3×10^{-4} mean squared error. The model RNN show pretty good performance on the sesame and haricot bean dataset. Model CNN also good for haricot bean. From the four deep learning model LSTM scored the lowest mean squared error in all datasets and here the first research question fired and now it possible to say LSTM model is the best fit model for Ethiopian agricultural market price analysis and prediction.

4.2 Comparison of Technical and Fundamental Indicators

To investigate the second research question, we used the selected best fit model and conducted an experiment on 600 days of each selected commodity dataset. During the experiment the selected model was trained using three datasets of one only the historical price data, theses with commodity technical indicators and the last was on DOCHL with news dataset only. the experiment was conducted for all the three commodities. The first experimental result is shown in the figure 4.29.

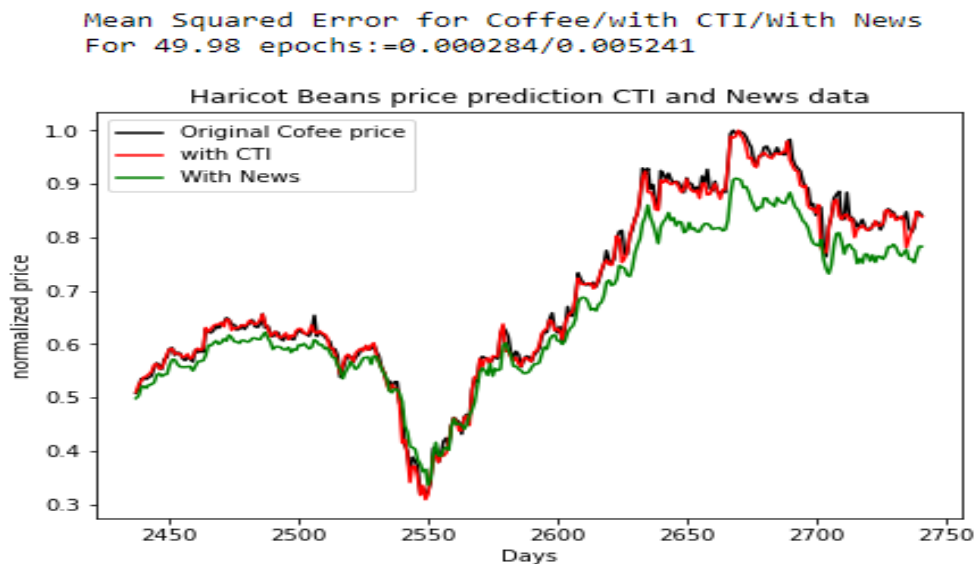


Figure 4.29 show the prediction of coffee with CTI and News data using LSTM

To select and prioritize important features and the determinant market analysis approaches first the sentiment score from news headlines combined with the closing price of the historical price data based on the date of publication. Then the closing price of the dataset was used as a target for prediction. Second the technical indicators from historical price data with only closing price of the data was combined and then the closing price of the data was used as a target for prediction. From the figure 4.29, it is clearly seen that prediction with CTI is more accurate than prediction with news sentiment scores. This means that it is better to use CTI for accurate prediction of Ethiopian commodity market price. In this way the second research question was answered.

The following figure shows the prediction comparison on sesame price using commodity technical indicators and news sentiment scores separately.

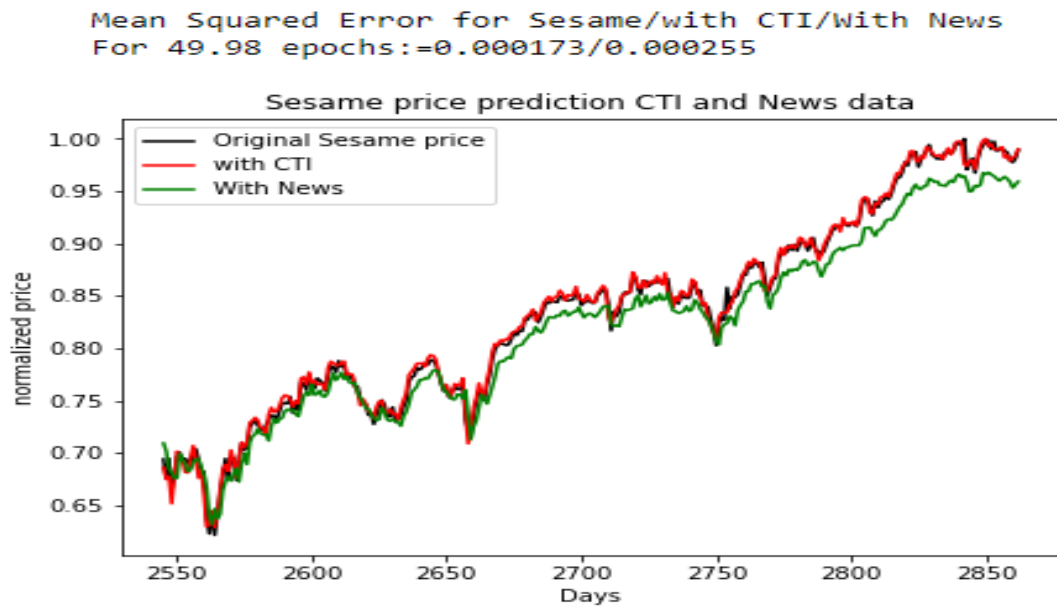


Figure 4.30 shows prediction of sesame with CTI and News

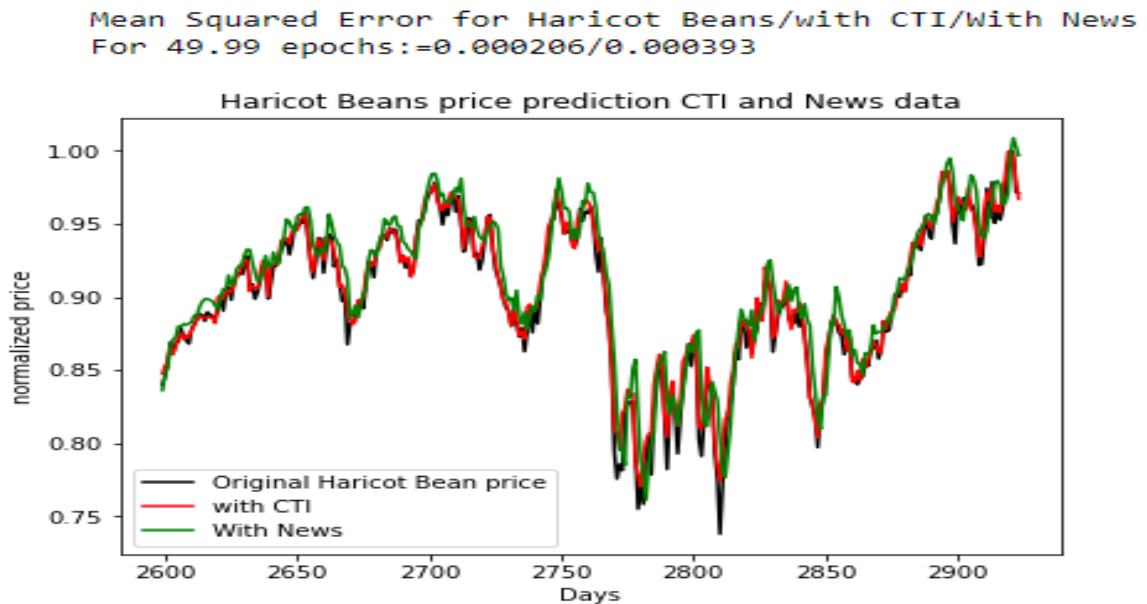


Figure 4.31 Shows prediction of Haricot beans price using with news and with CTI separately

From the figure 4.29, figure 4.30 and figure 4.31 it is possible to conclude that, from the two market prediction and analysis approaches, the technical market analysis is better.

To investigate whether the individual or combined use of the two market analysis is best approaches for accurate market price prediction, the following table was formulated from the table 4.1.

Table 4.3 shows MSE incurred for each input features

Model	Commodity	Features	MSE (10^{-4})
LSTM	Coffee	DOHCL+CTI	1.6
		DOHCL+NEWS	5.2
		DOHCL+CTI+News	0.96
	Sesame	DOHCL+CTI	1.7
		DOHCL+NEWS	2.5
		DOHCL+CTI+News	1.49
	Haricot bean	DOHCL+CTI	1.2
		DOHCL+ NEWS	3.9
		DOHCL+CTI+News	1.5

In the table 4.3, it is clearly seen that the prediction model has lowest MSE score when the model was trained using feature of both technical and fundamental indicators. The directional accuracy in table 4.1 also shows that the accuracy of the prediction model increases when it was trained with combined features of both technical and market analysis. This means that using the combined features of the two market analysis approaches is best way for the analysis and prediction of the Ethiopian commodity market.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5 Conclusion and Recommendation

As it is discussed, the aim of this thesis is analyzing and predicting the Ethiopian agricultural market price using selected deep learning algorithms. Keep this in mind, four deep learning models were built and experiments conducted on both numerical analysis and textual analysis. The technical indicators calculated from the market price data and the sentiment scores from news data were used as an input to the models. In the experiment, each model was trained, validated and tested using the training, validation and testing data from the three selected commodities. From the models (RNN, LSTM, CNN and CNN-LSTM), the best fit model for Ethiopian agricultural market price prediction is LSTM network which score MSE of 0.96×10^{-4} . In addition, two experiments were conducted to explore the determinant market analysis approach and how to be used, individual or combined, to get the best prediction. The result showed that the CTI features are determinant feature for price prediction. However, based on the last experiment, combined features from technical and fundamental analysis is a better way to get an accurate market price prediction.

In this thesis work nothing was done for additional high-level feature extraction from the denoised data. Interested researchers can use a stacked auto encoder decoder to extract high-level features from technical indicators. For fundamental analysis, only the financial news headlines were used. However, other microeconomic features such as governmental policy change, natural disaster and political instability may contribute a lot for market prediction. So, this can be a good start for them. The impact of news articles on the market prediction was investigated only for its publication date. For future work, for how many days the news affects the market should be investigated.

Reference

- Agustin, I. N., & Batam, U. I. (2019). THE INTEGRATION OF FUNDAMENTAL AND TECHNICAL ANALYSIS IN PREDICTING THE STOCK, *18*, 93–102.
- B, N. N., & Mohan, B. R. (2019). *Study of Stock Return Predictions Using Recurrent Neural Networks with LSTM*. Springer International Publishing.
<https://doi.org/10.1007/978-3-030-20257-6>
- Berhane, T., Shibabaw, N., Shibabaw, A., Adam, M., & Muhamed, A. A. (2018). Forecasting the Ethiopian Coffee Price Using Kalman Filtering Algorithm, *9*(3), 302–305. <https://doi.org/10.5814/j.issn.1674-764x.2018.03.010>
- Beyaz, E., Tekiner, F., Zeng, X., & Keane, J. A. (2020). Comparing Technical and Fundamental indicators in stock price forecasting Comparing Technical and Fundamental indicators in stock price forecasting.
<https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00262>
- Borovkova, S., & Tsiamas, I. (2019). An Ensemble of LSTM Neural Networks for High-Frequency Stock Market Classification, 1–27. <https://doi.org/10.1002/for.2585>
- Bulti, A. G., & Ray, A. (2019). Commodity Market Price Analysis and Prediction using Machine Learning Framework, *(8)*, 2822–2828.
- Chen, S., & He, H. (2018). Stock Prediction Using Convolutional Neural Network.
<https://doi.org/10.1088/1757-899X/435/1/012026>
- Chen, Z., Pang, M., Zhao, Z., Li, S., Miao, R., Zhang, Y., ... Feng, X. (2020). Feature selection may improve deep neural networks for the bioinformatics problems 1 Introduction. <https://doi.org/10.1093/bioinformatics/btz763/5583689>
- Faculty, G., & Technology, I. (2019). Department of Information Technology Sesame Price Prediction Using Artificial Neural Network University of Gondar Faculty of Informatics Department of Information Technology Sesame Price Prediction Using Artificial Neural Network.

- Gao, P., Zhang, R., & Yang, X. (2020). The Application of Stock Index Price Prediction with Neural Network. <https://doi.org/10.3390/mca25030053>
- Ghosh, A., Bose, S., Maji, G., & Debnath, N. C. (2019). Stock Price Prediction Using LSTM on Indian Share Market, 63, 101–110.
- Guo, T., & Aberer, K. (2017). HYBRID NEURAL NETWORKS OVER TIME SERIES FOR, (c), 1–8.
- Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred : CNN-based stock market prediction using a diverse set of variables CNNpred : CNN-based stock market prediction using a diverse set of variables, (April). <https://doi.org/10.1016/j.eswa.2019.03.029>
- Jha, G. K., & Sinha, K. (2013). Agricultural Price Forecasting Using Neural Network Model : An Innovative Information Delivery System, 26(2), 229–239.
- Karim, R., Bruce, N. D. B., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1837–1844.
- Krishna, V. (2018). ScienceDirect ScienceDirect NSE Stock Stock Market Market Prediction Prediction Using Using Deep-Learning Deep-Learning Models Models. *Procedia Computer Science*, 132(Iccids), 1351–1362. <https://doi.org/10.1016/j.procs.2018.05.050>
- Li, C. (2019). Multi-factor Based Stock Price Prediction Using Hybrid Neural Networks with Attention Mechanism. *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCoM/CyberSciTech)*, 961–966. <https://doi.org/10.1109/DASC/PiCom/CBDCoM/CyberSciTech.2019.00176>
- Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. *Information Processing and*

- Management*, 57(5), 102212. <https://doi.org/10.1016/j.ipm.2020.102212>
- Liu, Z., & Sullivan, C. J. (2018). Author ' s Accepted Manuscript Prediction of Weather Induced Background Networks Reference : To appear in : Radiation Physics and Chemistry. *Radiation Physics and Chemistry*.
<https://doi.org/10.1016/j.radphyschem.2018.03.005>
- Long, W., Lu, Z., & Cui, L. (2019). Knowledge-Based Systems Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164, 163–173. <https://doi.org/10.1016/j.knosys.2018.10.034>
- Mathematics, A. (2018). *ijpam.eu*, 118(22), 139–144.
- Moghar, A. (2020). ScienceDirect ScienceDirect April Using Stock Market Prediction LSTM Neural Network a LSTM Recurrent Stock Market Prediction Using. *Procedia Computer Science*, 170, 1168–1173.
<https://doi.org/10.1016/j.procs.2020.03.049>
- Mosavi, A., Ardabili, S., & Varkonyi-koczy, A. R. (2019). List of Deep Learning Models, (August), 1–28. <https://doi.org/10.20944/preprints201908.0152.v1>
- Muanenda, S., & Yohannes, E. G. (2018). Modeling and Forecasting the Volatility of the Export Price of Sesame in Ethiopia, 9(19), 7–13.
- Nhita, F., Saepudin, D., Paramita, A., Marliani, S., & Wisesty, U. N. (2019). Price Prediction for Agricultural Commodities in Bandung Regency Based on Functional Link Neural Network and Artificial Bee Colony Algorithms, 0–5.
<https://doi.org/10.3844/jcssp.2019.1390.1395>
- Ouyang, H., Wei, X., & Wu, Q. (2019). Agricultural commodity futures prices prediction via long- and short-term time series network. *Journal of Applied Economics*, 22(1), 468–483. <https://doi.org/10.1080/15140326.2019.1668664>
- Qiu, Y., Shan, H. Y., & Wei, L. (2020). A novel hybrid model based on recurrent neural networks for stock market timing. *Soft Computing*. <https://doi.org/10.1007/s00500->

- Rajeswari, S., & Suthendran, K. (2019). Developing an Agricultural Product Price Prediction Model using HADT Algorithm, (1), 569–575.
<https://doi.org/10.35940/ijeat.A1126.1291S419>
- Rikumahu, B. (2019). Prediction of Agriculture and Mining Stock Value Listed in Kompas100 Index Using Artificial Neural Network Backpropagation. In *2019 7th International Conference on Information and Communication Technology (ICoICT)* (pp. 1–5). IEEE.
- Sabu, K. M., & Kumar, T. K. M. (2020). ScienceDirect Predictive Predictive analytics analytics in in Agriculture : Agriculture : Forecasting Forecasting prices prices of of Arecanuts Arecanuts in Kerala in Kerala. *Procedia Computer Science*, 171(2019), 699–708. <https://doi.org/10.1016/j.procs.2020.04.076>
- Sai, K., & Vanukuru, R. (2018). Stock Market Prediction Using Machine Learning, (November). <https://doi.org/10.13140/RG.2.2.12300.77448>
- Singh, A., & Shakya, S. (2020). ScienceDirect ScienceDirect Analysis of look back period for stock price prediction with Analysis of look back period price prediction with RNN variants : A case study for on stock banking sector of NEPSE RNN variants : A case study on banking sector of N. *Procedia Computer Science*, 167(2019), 788–798. <https://doi.org/10.1016/j.procs.2020.03.419>
- Singh, R. (2017). Stock prediction using deep learning, 18569–18584.
<https://doi.org/10.1007/s11042-016-4159-7>
- Varun, R., Neema, N., Sahana, H. P., Sathvik, A., & Muddasir, M. (2019). Agriculture Commodity Price Forecasting using ML Techniques, (2), 729–732.
<https://doi.org/10.35940/ijitee.B1226.1292S19>
- Xu, B., Zhang, D., Zhang, S., & Li, H. (2018). Stock Market Trend Prediction Using Recurrent Convolutional Neural Networks, 166–177. <https://doi.org/10.1007/978-3-319-99501-4>

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks, (July 2019). <https://doi.org/10.1007/s00521-019-04212-x>

Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). ScienceDirect Stock Market Prediction Based on Generative Adversarial Network Stock Market Prediction Based on Generative Adversarial Network. *Procedia Computer Science*, 147, 400–406. <https://doi.org/10.1016/j.procs.2019.01.256>