Coconut Price Prediction in Sri Lanka Using Supervised Machine Learning Approach (LSTM)

G.A.S.M Padmasiri

2021

Coconut Price Prediction in Sri Lanka Using Supervised Machine Learning Approach (LSTM)

A dissertation submitted for the Degree of Master of Business Analytics

G.A.S.M Padmasiri

University of Colombo School of Computing

2021

DECLARATION

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: G.A.S.M Padmasiri

Registration Number: 2018/BA/024

Index Number: 18880242

Signature: Date:13/09/2021

This is to certify that this thesis is based on the work of

Mr. G.A.S.M Padmasiri

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. Rushan Abeygunawardana

Signature: Date:13/09/2021

DEDICATION

My dissertation is dedicated to my parents and family members. My loving parents, whose words of encouragement and insistence on perseverance are still in my ears, owe me a special debt of gratitude. This dissertation is also dedicated to the many friends and family members who assisted me throughout the process.

ACKNOWLEDGEMENTS

I received a lot of support and encouragement while completing this dissertation. I'd like to begin by thanking Dr.Rushan Abegunawardana, my supervisor, for his assistance in creating the study themes and methods. Your helpful feedback inspired me to think more clearly and increase the quality of my work.

I would want to offer my appreciation for my Silvermill colleagues' exceptional cooperation. Mr. Kelum Wickramasinha, my Silvermill section head, deserves special recognition. I would like to thank you for your patience and for all of the opportunities you have given me to continue my studies and also want to thank the Coconut Development Authority of Sri Lanka for providing the dissertation's data source.

In addition, I would want to thank my parents for their sound advice and sympathetic ear. Finally, without the help of my friends, who provided stimulating conversations as well as pleasurable distractions from my studies. I would not have been able to complete this dissertation.

ABSTRACT

Among Sri Lankans Coconut is a very prominent food with versatile usages. Coconut price is a significant factor but unfortunately the coconut price is fluctuating unpredictably because of several factors like rainfall and Soil Conditions. The goal of this project is to create a machine-learning model that uses supervised learning to estimate the price of coconut in Sri Lanka. This model can be used by the Coconut Manufacturing sector and Sri Lankan farmers to forecast prices and take required production actions. Coconut Development Authority information was selected for this research. The review of the literature assisted in identifying past research in relation to previously built crop price prediction models. The weekly coconut price data set was used to create and test the supervised learning model, and it is accessible from the Coconut Development Authority Web Site. For the modeling approach, a cross-industry data mining standard method was applied. It entails the processes of business comprehension, data comprehension, data preparation, modeling, and development. By analyzing the dataset and literature reviews it was decided to use the univariate Time series. long-short term memory neural network (LSTM) was selected as best fit for the scenario because it has eliminated the long-term dependency problem. Long Short-term memory has some variants named as Vanilla LSTM, Stacked LSTM and Bi-directional LSTM. As a first step the data was checked for the null values and prepared the dataset for model creation. The dataset was divided to two sectors as train and test data and Variants LSTM models are created and tested for different parameters. 50 to 250 epochs were changed during the model training and recorded the results. These results were evaluated using mean squared error and mean absolute error. According to the results that evaluated by mean squared error and mean absolute error measures shows that the Bi-directional LSTM was shows the best results. Finally using the results, it was concluded that for the Coconut price prediction Bi-directional LSTM is ideal. finally, for access the results to the users the power bi tool was used and published the results through web portal.

Table of Contents

List of Figures

List of Tables

CHAPTER 1
INTRODUCTION

Coconut is one of main commercial crop in the world. Sri Lanka produce around 2500 – 3000 million nuts per annum and hugely consume in domestic household and rest is use for to manufacture export value added Coconut products. The main issue coconut industry face is the coconut price variations. The goal of this project is to create a machine-learning approach that uses supervised learning to estimate the price of coconut in Sri Lanka. This model can be used by the Coconut Manufacturing sector and Sri Lankan farmers to forecast prices and take required production decisions. Coconut Development Authority publicly available data was selected for this research. The review of the literature assisted in identifying past research in relation to previously built crop price prediction models. The weekly coconut price data set was used to create and test the supervised learning model. The coconut price will be predicted using supervised learning methods such as the long-short term memory neural network (LSTM). For the modeling approach, a cross-industry data mining standard method was applied. It entails the processes of business comprehension, data comprehension, data preparation, modeling, and development. Here is how the research will be conducted.

1.1 Motivation

Because of its several uses coconut is referred to as "kupruka" in Sri Lanka. The word "Kapruka" refers to the significance of the coconut tree and its seeds. Coconut is a popular food in Sri Lankan households and has a high export value. Coconut growers, coconut exporters, and coconut product manufacturers all rely heavily on the price of coconuts. However, the price of coconut is volatile due to a variety of variables. It has been a well-known reality in Sri Lanka for a long time. Because of the volatility in the price of coconuts, it is difficult to plan for manufacturing, inventory management, and setting selling pricing for manufacturer items. It also has a significant impact on local consumers, as the price of coconut has risen significantly in recent months. Manufacturers, consumers, and policymakers all require a reliable coconut price prediction model in order to forecast the price of coconuts and make informed management decisions. Because of the great development of the Data Science Sector, machine learning algorithms may now be used for prediction.

1.2 Research Aim and Objectives

1.2.1 Aim

The Aim of this research is to help the Coconut consumers to understanding the future coconut prices for their consumptions. Coconut growers, coconut exporters, and coconut product manufacturers can take benefits by taking their raw material Coconut Stock.

1.2.2 Objectives

Innovative applications may be built with the enormous development of machine learning techniques, and the major goal of this research is **to Predict the Coconut price in Sri Lanka from a machine learning model.**

1.3 Statement of the problem

Coconut deal with several prices in the market channels such as retail prices, wholesale prices, auction prices. All these prices are undergone to fluctuations because of various factors. Coconut Price variations making a great risk of financing, inventory process, making future contracts, and in international trade and other associated actions for Coconut Consumers.

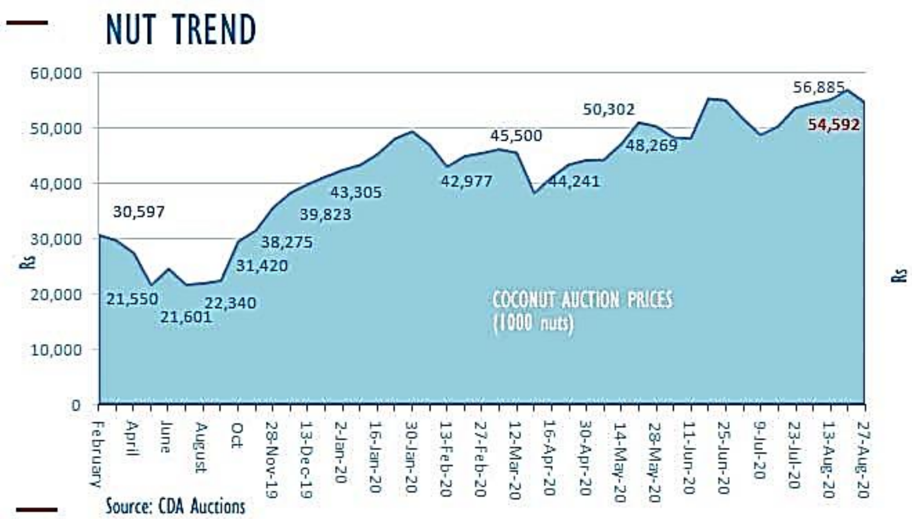

Figure 1.1: Nut Price Fluctuation

Source: CDA Auction

According to the Figure 1.1 The Y Axis shows the Coconut price for 1000 nuts and the X Axis shows the Date. Figure 1.1 shows that the Coconut price has a rising trend but when comparing months and weeks however, it is impossible to predict the next price. To overcome this problem currently Statistical forecasting methods are using like Autoregressive Integrated

Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving-Average (SARIMA). There are Some major drawbacks of ARIMA (Autoregressive Integrated Moving Average) forecasting, some of the classic model identification strategies for finding the proper model from a class of probable models are difficult to understand and frequently computationally ten time expensive, which is one of the fundamental limitations of ARIMA forecasting. This is a subjective procedure, and the accuracy of the chosen model can be affected by the forecaster's talent and experience. Furthermore, like with other forecasting systems, ARIMA models are inherently backward looking. As a result, long-term forecasts tend to be straight lines, and they are poor at projecting series with turning points. Therefore, new methods are needed to predict the prices.

1.4 Background of the Study

Sri Lanka is a legendary island for the Coconut among the worlds. It is a commodity of economic, social and cultural values for Sri Lankans. Agriculture plays a main role in the Sri Lankan market and has been the third most significant commercial crop of Sri Lanka. Coconut shows a significant role in many developing countries and directly affecting economy of people and the Country. Because of its unique qualities Coconut has become one of the prominent and key plantation crops with versatile consumes.

1.4.1 Coconuts Background

Coconut is also called Cocos nucifera. It is the fruit of the coconut palm.("coconut | Description, Uses, & Facts," n.d.). It contains large endosperm. Because of Coconut endosperm it is different from other fruits. It contains a significant volume of coconut water or coconut juice, which is a clear liquid. Coconuts that are mature and ripe can be eaten as seeds, or they can be processed for oil and plant milk from the skin, it produce charcoal and fibrous husk from the hard shells, and coir.("Coconut," 2021). Following Figure 1.2 and 1.3 Shows the Coconut Out and inside view.

Figure 1.2: Coconut Outside view
Source: www.veggies.ind.in

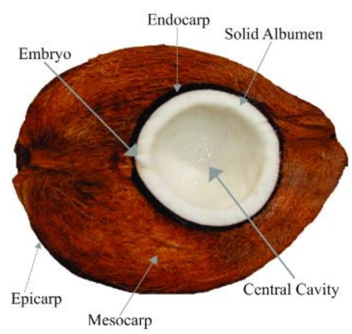

Figure 1.3: Inside Coconut View
Source: www.researchgate.net

Sri Lankans mainly used Coconut Milk and Coconut Oil for the food preparation, and It is nearly 65 percent of the output. The remainder is exported in the form of value-added kernel products. They are listed below.

- Desiccated Coconut
- Coconut Oil
- Copra
- Coconut Cream
- Coconut Milk Powder
- Coconut Water
- Coconut Shell Products
- Coconut Fiber Products

Following Figures Shows the value-added Coconut products.

Figure 1.4: Coconut Products

Source: www.coconutboard.com

To make desiccated coconut and coconut oil from raw Coconut, Coconut product manufacturers employ a procedure that starts with the husking of the coconut and the removal of the coconut water (dewatering), followed by the removal of the shell (shelling) and pairing to obtain the white meat (white coconut), which is used as a material for Desiccated Coconut and Coconut oil.

Coconut oil can be produced using both traditional and modern processes. To obtain oil, milk produced from grated coconut kernels is cooked in the traditional technique. In recent years, a bridge press and mechanical grater have been used to somewhat automate the conventional procedure. Wet processing is the contemporary method of extracting oil from fresh coconut kernels.("CDB - Coconut Products," n.d.)

Coconuts are an essential dietary source of nutrients. This substance contains 363 to 669 calories per 100 grams. Between 4.8 and 10.8 glaucids and about 9 g of fiber are generated by the same number. The importance of potassium (35-650 mg per 100 g of product) and chlorine are among the minerals found in this tropical fruit (122-190 mg for 100 g of fresh produce)("Listado de hortalizas," n.d.).

Coconut oil is swiftly digested and ingested at a rate of 95-98 percent faster than butter fat due to the low molecular weight of fatty acids. It contains almost no cholesterol, like other vegetable oils. Coconut milk is rich in fiber, which helps to preserve potassium levels. Because of its role in maintaining intracellular equilibrium, potassium is one of the most essential compounds for human metabolism. Potassium is a required part of our diet. Muscle fatigue, mental illness, and heart disorders are all signs of a lack of it. The pulp and milk help to avoid the symptoms of a potassium deficiency, such as neuromuscular impairment (weakness or paralysis), respiratory system disorders, and cardiovascular disease.

Because of its good fragrance and purity as an edible oil, hair oil, and baby oil, virgin coconut oil is favored. It is added to a baby's body to shield them from skin diseases. This oil has a longer shelf life due to its low FFA content.

1.4.2 Exports Income for Coconut Products in Sri Lanka

By offering a varied range of products to the global market, the coconut and coconut-based goods business contributes significantly to foreign exchange revenues. Coconut exports contribute more to the Sri Lankan economy, while agricultural export contributions have increased significantly in the export market. These exports make a major contribution to foreign exchange. Following Figure 1.5 Shows the export performance in year 2010 to 2020 and it has slightly increasing trend and it is a good sign Sri Lankan economy.

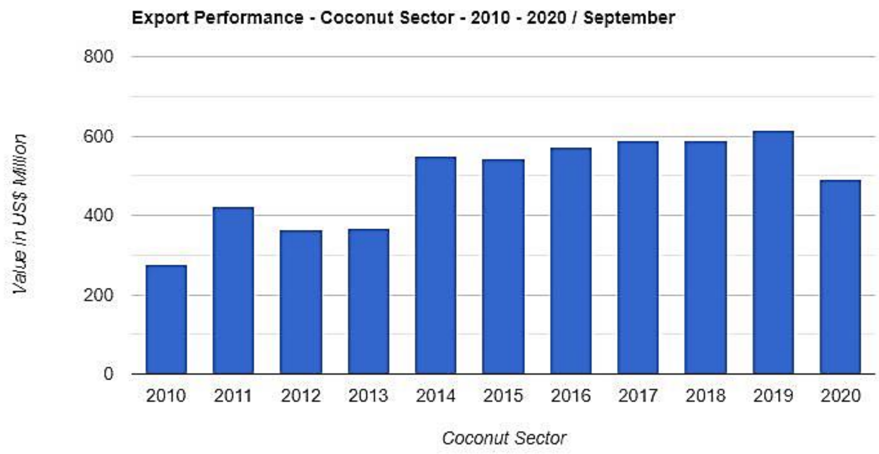

Figure 1.5: Coconut Export Income

Source: Sri Lankan Export Development Board

Sri Lanka is one of the world's leading coconut producers. It has a sizable market share in the global coconut market, owing to its availability of Desiccated Coconut (DC).

Coconut cultivation covers a total of 394836 hectares, with large estate holdings accounting for 18% of the total and king coconut production accounting for 10% of the total. About 140,000 people are working in this sector across Sri Lanka. The coconut-intensive regions of the West and Northwestern provinces are known as the "Coconut Triangle." Kurunegala, Puttalam, and Gampaha are the three major coconut-producing districts. (“Sri Lanka Export Development Board - Sri Lanka Business Portal,” n.d.).

There are three important coconut kinds in Sri Lanka. Coconut trees come in three varieties: talls (Typica), dwarfs (Nana), and king coconut (Aurantiaca). There are many variations within each variety, each with its own set of characteristics. Talls, also referred as Sri Lanka Big, are grown on nearly all of Sri Lanka's coconut farms. Except for beverage applications, dwarfs are not commercially produced (as copra from them is in short supply and of low quality).

Sri Lanka has a sizable domestic demand for coconut-based products, accounting for nearly two-thirds of the country's annual production of 1.5 billion nuts. Coconut consumption per capita in Sri Lanka is 110 kg per year, which has an impact on raw material supply as well as promoting the industry.

The Coconut Development Authority, Coconut Cultivation Board, and Coconut Research Institute are the three principal government entities in charge of production and quality improvement, supply development, and research, respectively. Farmers, Desiccated Coconut Producers, Copra Producers, Coconut Oil Producers, Fiber Millers, Shell Charcoal Producers, Value Added Fiber Goods Producers, Activated Carbon Producers, and Exporters are the major stakeholders in the coconut business. (“Sri Lanka Export Development Board - Sri Lanka Business Portal,” n.d.)

1.4.3 Coconut yield and affecting factors.

Following Graph shows the Coconut Yield in Sri Lanka over time. It shows seasonal patterns with the yearly coconut production.

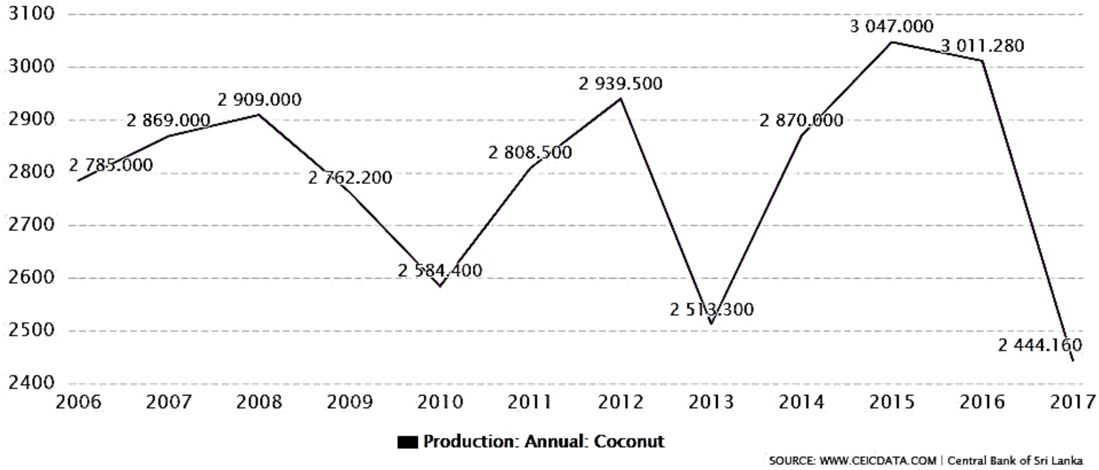

Figure 1.6: Coconut Yield Yearly

Source: Central Bank of Sri Lanka

According to the Figure 1.6 in year 2006, 2010, 2013, 2017 Coconut production has decreased and year 2008, 2012, 2015 have the highest Coconut Production. Coconut production pre year is called the Coconut yield (no. of nuts per year). The amount and distribution of rainfall, as well as the year-round temperature, have a major effect on coconut yield. Soil is another factor and the Coconut disease influence on the Coconut Yield (Samarasinghe et al., 2018).

For Any Agricultural crops, the main factors that affecting is the rainfall apart form that the temperature soil condition and fertilizer usage pest attack will affect the yield. When consider about Coconut drought and high temperatures are two big stress factors that have a direct negative effect on nut yield. Abiotic stress affects the coconut as a result of temperature rises caused by climate change and changes in rainfall patterns. According to the Figure 1.6 it can say that these factors are influence the coconut yield. Coconut may thrive in climates with 1300 mm or more of annual rainfall, high humidity, temperatures between 27 and 30 °C, and moderately to well-aerated soils. (Samarasinghe et al., 2018).

Coconut has an erratic development pattern, although it does produce an inflorescence at each leaf axil at intervals of 25 to 30 days, depending on the temperature and the age of the palm. Some axils, however, fail to throw off inflorescences due to abortion of inflorescences generated within the leaf axil. Depending on genetic and environmental conditions, the total number of female flowers in a coconut inflorescence might range from zero to a few hundreds. A typical inflorescence, on the other hand, has tens of thousands of male flowers. Initial nut set (female flowers changed into button nut three months after an inflorescence emerged) in coconut may be low due to unfavorable climatic conditions such as high temperature, low light, and moisture tension. Under unfavorable conditions, female flower and immature fruit abortion is a common occurrence in coconut.

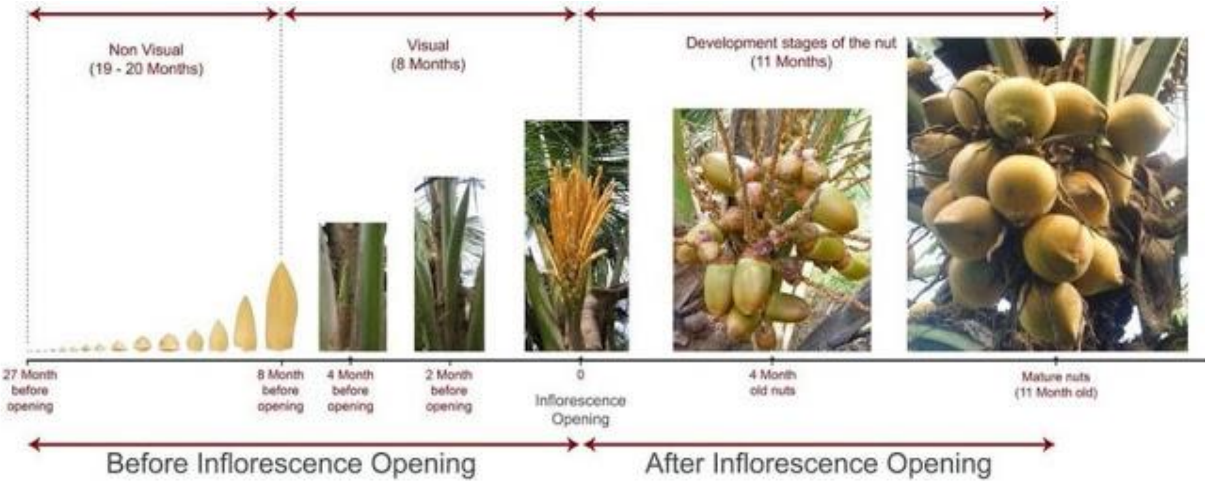

Figure 1.7: Coconut Inflorescence Opening

Source: Coconut Research Institute, Sri Lanka

Figure 1.7 Shows the steps of the Coconut fruit development. Mature coconuts develop approximately eleven months after inflorescence opening. Since young nuts are susceptible to climatic variation, The most essential period is thought to be the first three months after the inflorescence opens. (Ranasinghe et al., 2015)

1.5 Scope of the Study

This research is focused to predict the coconut price using finest machine learning techniques. To select the best algorithm literature reviews will be examined which done for previews similar domains by the researchers. Sri Lanka Coconut Authority weekly auction prices will be used to build the model.

The following methods and procedures are planning to use in the research,

(1) Data collection:

> The data using for the create models are taken from the Sri Lanka Coconut Authority during the period 2010- 2020, which is count 490 weeks, the data set is divided into two parts, one is 85% used for training, and other is used 15% for testing.

(2) Data analysis and preprocessing:

> Using the Python development environment, the time series data was summed and exhibited in statistical approaches such as tables and graphs to find out all the specifics about the data.

(3) Implementation:

> Python and the Keras open-source software package, which provides a Python interface for artificial neural networks, are used to create the suggested prediction system.

(4) Performance evaluation:

> The performance of the proposed prediction system is evaluated using the obtained results comparing with actual data points.

1.6 Feasibility Study

Fulfill the objective of this research Data is available form year 2010 to 2020. (490 weekly data points). The data used to create the coconut price prediction model is collected from Coconut Development Authority and used the Coconut Auction Price for this prediction. Coconut Auction was held on every week Thursday. With the Collected data it was only possible to create a time series model. For Predicting the coconut price machine learning algorithms are available and from existing algorithms after literature review best suitable algorithm can be selected for predict. Much Research have studied the time series analysis and publish their finding for Literature review. Python programming language will be used to preprocessing analyze and develop the model for prediction. These technologies can easily learn and can be done the research.

As a Limitation to the research, it can be shows that the data is only available for year 2010 to 2020. Another Limitation to the research is time frame that allocated for research and Accuracy of the Model.

1.7 Structure of the Dissertation

This research contains five main chapters including the Introduction chapter. Introduction chapter describe about the research background research problem and the outline of the research. Other chapters are organized as below. Chapter 02 is based on the similar research information’s which previously done by the researchers to predict the prices. Chapter 03 the concepts related to the methods and techniques used with respect to the introductory and advanced analysis in the study are broadly described. Chapter 04 Implementation and results evaluation describes the Methodological Approach, Data Collection, Method of Analysis. Chapter 05 describes the Conclusion and future works.

CHAPTER 2
LITERATURE REVIEW

There are various works that illustrate how various researchers have explored the application of forecasting in various scenarios and different situations. Many have contributed to research in the price prediction sector and applied the knowledge on Agriculture for enhance the productivity and profit. Thera are mainly two approaches to the Time series price prediction they are classical methods and the machine learning methods.

2.1 Classical Methods

Classical method also can identify as Statistical method that using the Statistical techniques for predict the prices. There are some researchers used Statistical methods like Autoregression (AR),Moving Average (MA),Autoregressive Moving Average (ARMA),Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving-Average (SARIMA),Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX),Vector Autoregression (VAR) ,Vector Autoregression Moving-Average (VARMA) ,Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX),Simple Exponential Smoothing (SES) and Holt Winter's Exponential Smoothing (HWES) for price predictions.

Peiris et al., (2007) has presented research named Use of seasonal climate information to predict coconut production in Sri Lanka. He has used log–linear trend model and the regression model to predict the coconut production. The prediction of Annual coconut production for 2003 and 2004 provided errors of 6.5 and 7.0%. Alibuhtto, (2013) was using ARIMA model to predict the fresh Coconut export in Sri Lanka. The ARIMA (1, 1, 1) model was identified as the best fitted to forecast the fresh coconuts export from Sri Lanka. Udumulla, (2018) Also has research about export Coconut products price forecasting. He mainly focusses on desiccated Coconut oil and nuts. He observes that the best model for the DC prices is ARIMA (2,2,1), Oil price is SARIMA (0, 1, 1) (0,0,1) and for coconut nuts prices is ARIMA (1,1,0). Since there is a strong correlation between prices, he considers vector auto regression (VAR) model to improve the forecasts. Among several plausible models VAR of order 3 results in the best model. VAR (3) model shows that the prices of coconut nuts have a strong relationship among DC and oil prices. Rani and Chohan, (2018) also predicting the price if coconut oil with respect to time they conclude that that ARIMA (1,1,3) and (2,1,3) are very close to each other so They use both models for forecasting purposes. They check the accuracy by using MAE, MAPE, and RMSE. From the above study, it is found that ARIMA (2, 1, 3) is more efficient than ARIMA (1, 1,3).

Liu and Shao, (2016) research about India's Tea Price Based on ARMA Model they use Indian tea's auction price in the year 2013 to 2014 the prediction error was small. The test results were satisfactory and meet the forecast requirements but Villaren M. Vibas and Raqueño, (2019) conduct a research on Retail Price Movements of Basic Fruit and Vegetable. He used ARIMA, SARIMA, and ARIMAX for his research and it was found that for different fruits different algorithms was fit as an example ARIMAX emerged as the finest model for banana and mango. For vegetable commodities, the best model to use for estimating monthly prices of cabbage was ARIMAX, SARIMA model was fit for pechay and tomatoes.

Uddin et al., (2011) has presented Statistical Analysis and Trend on maximum temperature using SARIMA model. He was able to fit SARIMA model to predict the temperature. Sajal and Rahman, (2012) was predicting groundwater fluctuation with time series analysis. They also applied the Box and Jenkins univariate stochastic models widely known as ARIMA to simulate groundwater table fluctuations in all monitoring wells under consideration in the study area. The best models predicted data is compared to the observed time series, and the accuracy is assessed using several error parameters. In terms of numerical accuracy, the simulation indicated that ARIMA models produce reasonably accurate projections. The results also revealed that the anticipated data for each monitoring well closely matches the actual data.

For predicting the Year-Long Monthly Rainfall Forecasting for a Coastal Environment of Bangladesh Mahmud et al., (2015) has used ARIMA model. It was found effective Hence it is recommended that stochastic model can be used in similar area to forecast rainfall also Mahmud et al., (2016) found that SARIMA model also fitted satisfactorily. Not only for rainfall but also SARIMA model has used to forecast the humidity by Hossain et al., (2016) another research has used to compare Seasonal ARIMA and Exponential smoothing by Lwesya and Kibambila, (2017) for forecasting Tourist Arrivals in Tanzania. In a similarly organized scenario, the results suggest that Seasonal ARIMA(4,1,4)(3,1,4)12 and Holt-Winters multiplicative smoothing methods are successful in forecasting visitor arrivals in Tanzania. When the two models were evaluated in different topologies, however, the Holt-

Winters multiplicative smoothing approach outperformed Seasonal ARIMA(4,1,4)(3,1,4)12. This indicates that the Holt-Winters multiplicative smoothing method with Alpha (0.01), Delta (0.11), and Gamma (0.11) is more effective in forecasting visitor arrivals in Tanzania in the short term and can be used to aid tourism planning operations.

Etuk et al., (2017) perform A Simulating Model for Daily Uganda Shilling-Nigerian Naira Exchange Rates using SARIMA and it was observed SARIMA(0,1,0)x(1,1,0) model is more adequate model to forecast exchange rates. MLMCE, India and Krishnan, (2019) presented Electricity bill price forecasting with ARIMA model. The numerical results reveal that the proposed framework outperforms other benchmark methods in terms of accuracy.

2.2 Machine Learning Methods

With the vast development of the machine learning techniques researchers are focused to use innovative technologies for prediction using powerful machine learning tools. There have been numerous studies on the use of neural networks in time series price prediction. Artificial Neural network (ANN), Back propagation neural network (BPN), Classification and Regression Trees (CART), Group Method of Data Handling (GMDH), K Nearest Neighbor (KNN), Multilayer Perceptron (MLP), Particle Swarm Optimization (PSO), Support Vector Machine (SVM), Support Vector Regression (SVR), Long Short-Term Memory (LSTM) are some of useful methods used by researchers.

It was notice that using machine learning techniques many researches are done to predict the Stock price. Haider Khan et al., (2011) also presented Price Prediction of Share Market using Artificial Neural Network (ANN). They introduce a method which can predict share market price using Backpropagation algorithm and Multilayer Feedforward network. Ayodele A and Charles K, (2012) also predict stock price using Neural Network with Hybridized Market Indicators. They used three-layer (one hidden layer) multilayer perceptron models (a feedforward neural network model) trained with backpropagation algorithm. The empirical results obtained showed high level of accuracy for daily stock price prediction with hybridized approach performing better than technical analysis approach. Hegazy et al., (2013) also use Particle swarm optimization (PSO) and least square support vector machine (LS-SVM) for Stock Market Prediction. PSO algorithm selects best free parameters combination for LS-SVM to avoid over-fitting and local minima problems and improve prediction accuracy. The proposed model was applied and evaluated using thirteen benchmark financials datasets and compared with artificial neural network with Levenberg-Marquardt (LM) algorithm. The obtained results showed that the proposed model has better prediction accuracy and the potential of PSO algorithm in optimizing LS-SVM. Qiu et al., (2020) also forecast stock price using long-short term memory neural network. They compared the results with the other three models, including the LSTM model, the LSTM model with wavelet denoising and the gated recurrent unit(GRU) neural network model on S&P 500, DJIA, HSI datasets. Results from experiments on the S&P 500 and DJIA datasets show that the coefficient of determination of the attention-based LSTM model is both higher than 0.94, and the mean square error of model is both lower than 0.05. Shen and Shafiq, (2020) also predict the stock market price using long short-term memory (LSTM) model. They conducted comprehensive evaluations on frequently used machine learning models and conclude that proposed solution outperforms due to the comprehensive feature engineering. The system achieves overall high accuracy for stock market trend prediction. With the detailed design and evaluation of prediction term lengths, feature engineering, and data pre-processing methods. Yang et al., (2020) presented research on Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory. In the research They designed a CNN-based module for feature extraction. Finally, they employed a LSTM network for stock price movement direction prediction. Extensive experiments demonstrated that the deterministic trend signals and the ranked stock indices in the three-dimensional input tensor play a significant role in improving the prediction performance. Moreover, the result of comparing with several state-of-the-art models showed the superiority of the proposed model in predicting direction of the stock price movement. Obthong et al., (2020) presented very useful Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques. In the report they compare each of machine learning methods like ANNs (Artificial Neural Networks), BPNN (Back propagation neural network), BSM (Black Scholes Model), CART (Classification and Regression Trees), FCM (Fuzzy c means), GAs (Genetic Algorithms), GMDH (Group method of data handling), GP (Gaussian Processes), GRNN (Generalized Regression Neural Network), HMM (Hidden Markov Model), Hierarchical clustering, LSTM (Long Short-Term Memory), MLP (Multilayer Perceptron), QRNN (Quantile Regression Neural Network), RNN (Recurrent neural networks), SOM (Self organizing maps), SVM (Support Vector Machine), SVR (Support Vector Regression) and finally the survey presented The number of ML algorithms and techniques in terms of types of input, purposes, advantages and disadvantages. According to this survey it was highlighting that LSTM model can predict stock prices more accurately.

Agriculture sector is another field that can practice Machine Learning techniques. Kajeswari and Suthendran (2019) used the HADT Algorithm to create an Agricultural Product Price Prediction Model. The HADT forecast model's result is more encouraging and exact in predicting agricultural commodity prices than other contemporary decision tree models. Rachana et al., (2019) used Naïve Bayes Algorithm and K Nearest Neighbor technique to predict the Price of the Crop. Sabu and Kumar, (2020) forecast prices of Arecanuts in Kerala using LSTM neural Network and found that LSTM was the best model to fit however Samuel, (2020) used Logistic Regression, Decision Trees, XGBoost, Neural Nets to predict Crop price according to his findings XGBoost is the suitable technique for price prediction.

Limsombunc et al.,(2004) forecast house price using artificial neural network model. the empirical evidence presented in this study supports the potential of neural network on house price prediction. Sangrody et al.,(2018) presented Long Term Forecasting using Machine Learning Methods. The performance of the most commonly used machine learning approaches in load forecasting was investigated in this study. Feedforward artificial neural network (ANN), support vector regression (SVR), recurrent neural network (RNN), k-nearest neighbors (KNN), generalized regression neural network (GRNN), and Gaussian Process Regression are some of the approaches available (GPR). New England Network is the case study, and its monthly energy usage from 2000 to May 2016 is used to train and validate load forecasting algorithms. Weather indicators (HDD and CDD), a dummy variable of month number, and the moving average of the target variable before 2011 are used as inputs to the load forecasting models. The findings of the forecasting models represented by MAPE demonstrate that, while all LF approaches perform well on both the training and validation data sets, the feedforward ANN method outperforms the others. Do and Yen, 2019 presented predicting commodity prices in market using neural network they use group method of data handling (GMDH) technique in one day-ahead forecasting the market prices. The results showed that the proposed model based on GMDH technique outperforms than other methods in prediction of commodity prices. Nasser and Al-Shawwa, 2019 use Multilayer Perceptron Topology for predicting the price range of a mobile phone. Test data evaluation shows that the ANN model is able to correctly predict the mobile price range with 96.31 accuracy. Singh, 2020 used Enriched RF (Random Forest) to forecast Land Price. It was works well with numeric features. Sun and Xu, 2021 presented Carbon price prediction based on modified wavelet least square support vector machine. A single model cannot meet the prediction accuracy anymore. Since this is the case, this paper puts forward a novel hybrid forecasting model, consisting of the ensemble empirical mode decomposition (EEMD), the linearly decreasing weight particle swarm optimization (LDWPSO), and the wavelet least square support vector machine (wLSSVM). Innovatively, wLSSVM is utilized in the field of carbon price prediction for the first time. Firstly, EEMD decomposes the raw carbon price into several stable subsequences and a residual. Then, the inputs of each sequence are determined by the partial auto-correlation function (PACF). Next, wLSSVM optimized by LDWPSO forecasts each sequence separately. Finally, the final prediction result is obtained by adding all prediction results. For the purpose of verifying the effectiveness and superiority of the EEMDLDWPSO-wLSSVM model, a total of 12 models were built to compare their performance in three regions of Guangdong, Hubei, and Shanghai respectively from three evaluating indicators: MAPE, RMSE, and R2. All the predicted results showed that the model presented in this paper has the best forecasting performance among all the model.

2.3 Combination of Both Classical and Machine Learning Methods.

de Faria et al., 2009 perform Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods. The objective is to compare the forecasting performance of both methods on this market index, and in particular, to evaluate the accuracy of both methods to predict the sign of the market returns. Also the influence on the results of some parameters associated to both methods is studied. results show that both methods produce similar results regarding the prediction of the index returns. On the contrary, the neural networks outperform the adaptive exponential smoothing method in the forecasting of the market movement, with relative hit rates similar to the ones found in other developed markets. Aras et al., (2017) research on sales forecasts for a global furniture retailer operating in Turkey. They use using state space models, ARIMA and ARFIMA models, neural networks, and Adaptive Network-based Fuzzy Inference System (ANFIS). Also, the forecasting performances of some widely used combining methods were evaluated by comparison with the weekly sales data for ten products. According to this journal, this study is the first time that the recently developed state space models, also called ETS (Error-Trend-Seasonal) models, and the ANFIS model have been tested within combining methods for forecasting retail sales. Analysis of the results of the single models in isolation indicated that none of them outperformed all the others across all the time series investigated. However, the empirical results suggested that most of the combined forecasts examined could achieve statistically significant increases in forecasting accuracy compared with individual models. Sabu and

Kumar, 2020 presented Forecasting prices of Arecanuts in Kerala using time-series and machine learning models. The models SARIMA, HoltWinter's Seasonal method, and LTM neural network were used, and their performance was evaluated based on the RMSE value on the arecanut dataset with prices from 2007 to 2017. LSTM neural network model was found to be the best model that fits the data.

2.4 Research Gap and Conclusion.

This Chapter will discuss about the previous studies done by the researchers related to the price prediction, Time series, Machine Learning Methods for price prediction. Price predictions has become very demanding area of research due to the applications such as stock price forecasting, business planning, weather forecasting, resources allocation, and numerous others. Price predictions can be effectively achieve using Statistical methods and machine learning methods. When considering the data inputs, it can be univariate or multivariate attributes can be used for analysis. univariate involves the analysis of a single variable while multivariate analysis examines two or more variables. Most multivariate analysis involves a dependent variable and multiple independent variables. When considering the input data if input data has a natural order, it can be called as a time series data. Time series analysis can be used to analyze time series data. Exponential smoothing, Autoregression (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) represent some classical time series forecasting techniques. On the other hand, linear regression is a forecasting model that does not require time series data. Finding correlations and modeling the relationship between one or more dependent or independent variables is the goal of regression analysis.

Most of the research that done in the agriculture section has done by the classical Statistical methods using ARMA, ARIMA, SARIMA and exponential Smoothing Methods. But it was found that Machine Learning methods are more accurate than the Classical methods.

With the vast development of the machine learning techniques powerful machine learning tools are introduced for price predictions Artificial Neural network (ANN), Back propagation neural network (BPN), Classification and Regression Trees (CART), Group Method of Data Handling (GMDH), K Nearest Neighbor (KNN), Multilayer Perceptron (MLP), Particle Swarm Optimization (PSO), Support Vector Machine (SVM), Support Vector Regression (SVR), Long Short-Term Memory (LSTM) are some of useful methods.

According to the literature review conducted here, it was found that there is no research on the LSTM model in the Sri Lankan Coconut Price Prediction but Machine learning approaches have been discovered to be employed in stock price predictions, exchange rate predictions, and other industries other than agriculture. It has a significant gap in that this machine learning technology has not been utilized to predict in the agriculture sector, and it has also been discovered that RNN machine learning algorithms have a higher accuracy percentage. Finally, a Long Short-term Memory (LSTM) model is chosen for forecasting Coconut Price based on all of the literature reviews and a specified data set.

CHAPTER 3

METHODOLOGY

3.1Systematic Approach

CRISP-DM is a common analytics method known as cross-industry data mining standard method. It is an open standard process model that explains how data prediction can continue. This process standard will be used for the research. This method will be help to discovery of trends, relationships, and observations in broad data sets that help businesses measure and manage where they are now and forecast where they will be in the future.

The six most important steps in the data mining process are listed below.

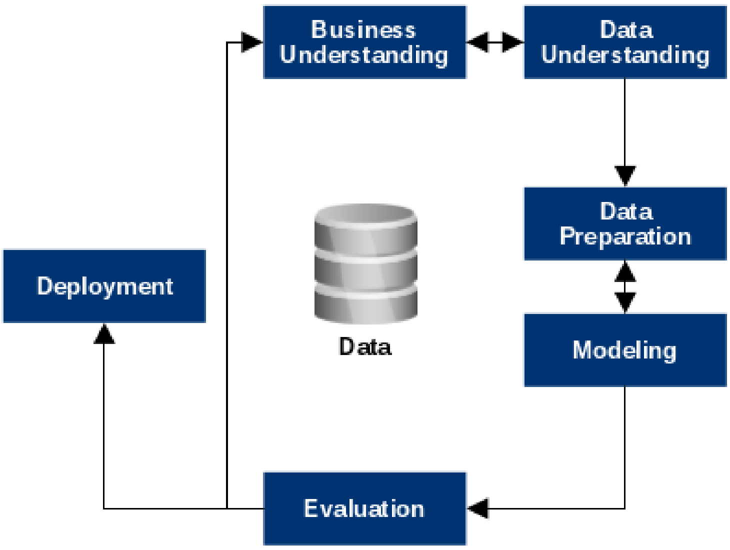

Figure 3.1: CRISP-DM Process Model

Source: (Gluesenkamp, 2019)

3.1.1 Business understanding

During the process of getting to know business, it is necessary to thoroughly comprehend business goals and determine what the company's requirements are. Then, evaluate the current situation by identifying the tools, assumptions, limitations, and other critical factors to consider. Establish data mining targets to achieve the business object based on the business objectives and current circumstances.

This process is successfully completed by finding the most current problem in the country. The Coconut price is gone high in this year unexpectedly and many organizations and individuals have affected by this situation. This proposed model will help to avoid the unexpected Coconut Prices.

3.1.2 Data understanding

To help get acquainted with the data, the data understanding process begins with initial data collection from available data sources. In order to complete the data collection successfully, certain essential tasks such as data loading and data integration must be completed. The data must then be analyzed by answering data mining questions, which can be accomplished by querying, reporting, and visualization.

Data was selected from the Coconut Development Authority (CDA) Auction. CDA Auction was held every Thursday weekly from year 2010 to 2020 weekly data are available for the analysis.

COLOMBO COCONUT AUCTION AVERAGE PRICES					
Year	Month	Auction Date	Qty Offered (nuts)	Qty Sold (nuts)	Fresh Nut Price (Rs. Per 1000 nuts)
2020					
2020	January	2	1,274,374	1,134,618	42,392.68
		9	901,611	783,792	43,305.56
		16	414,263	322,352	45,265.05
		23	552,299	503,068	48,077.60
		30	667,831	492,809	49,382.44
Total / Avg			3,810,378	3,236,639	45,684.67
2020	February	6	996,490	701,222	46,969.33
		13	432,796	310,791	42,977.15
		20	699,523	610,403	44,881.60
		27	709,150	417,066	45,454.64
Total / Avg			2,837,959	2,039,482	45,070.68

Figure 3.2: Coconut Development Authority Coconut Prices

Source: https://cda.gov.lk

3.1.3. Data preparation

In most cases, data preparation takes up about 90% of the project's time. The final data set is the product of the data preparation process. After identifying available data sources, they must be chosen, cleaned, constructed, and formatted into the desired format. During this process, a more in-depth data exploration task may be carried out in order to spot trends based on market knowledge.

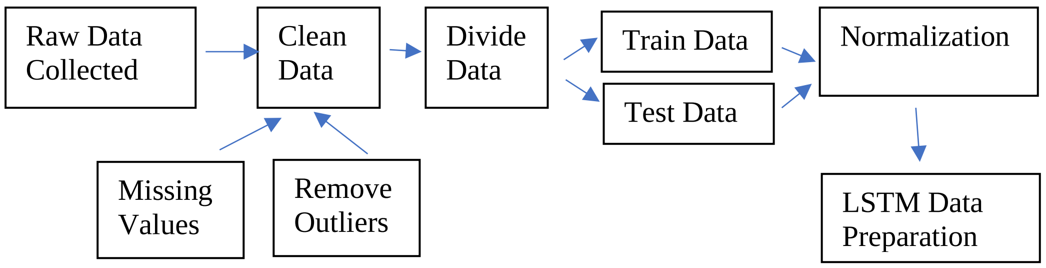

Figure 3.3: Data Preparation Activities

Selected data was preprocessed for the univariate time series analysis and got date and price columns by python pandas library using python pandas check for missing values and remove null values then divide the data set to Train and Test data.

When training a neural network, such as a Long Short-Term Memory recurrent neural network, the data for the sequence prediction problem would almost certainly need to be scaled. When a network is fitted to data that is unscaled and has a number of values. Large inputs can slow down network learning and convergence, and in some cases prevent the network from learning the problem effectively. There are two of them when scaling of series that may want to consider normalization and standardization. Both of these can be accomplished with Python's scikit-learn machine learning library.

The LSTM model will train a function that transfers a set of previous observations as input to a new observation as output. As a result, the series of observations must be converted into several examples for the LSTM to learn from. For the one-step prediction that is being learnt, can partition the sequence into numerous input/output patterns called samples, where three-time steps are used as input and one time step is used as output.

3.4 Modeling

First, the modeling techniques to be used with the prepared data set must be chosen. The test scenario must then be created in order to verify the model's accuracy and validity. The prepared data set is then used to construct one or more models. Finally, models must be carefully evaluated by stakeholders to ensure that generated models meet business objectives.

LSTM Model is selected for the predicting Coconut price from the Literature review according to the data characteristics.

3.4.1 Neural Networks

The term "Neural Networks" refers to a technological idea that capturing the scope of machine learning and the biological brain. Network of neurons or nodes are the basic build unit in Neural Network. The network's nodes receive information as input from the network's edges. The weight set by the network's constructor is multiplied by the data inputted into the nodes. The weight is used to balance the importance of several computing outputs from a particular node. The result is combining with the non-linear function, which is referred to as the activation function. The activation function, which is usually a tan(h) function, controls how much of the weighted result from the neurons is contributed to the output, which can be the final output or an output that is passed on to other neurons or nodes in the network.

3.4.2 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are a type of neural network that has the ability to remember, making them more similar to how humans process information and making them a useful tool for solving a variety of scientific problems. The data is processed independently in a typical neural network. Traditional neural networks are inadequate to RNN.

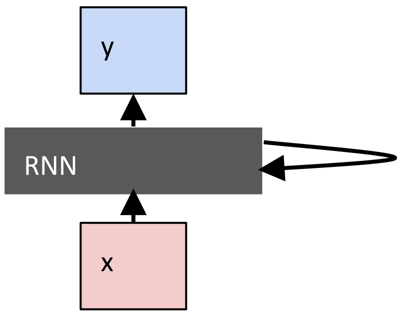

Figure 3.4: Recurrent Neural Network

3.4.3 Long-term Dependency

Long-term dependency is a challenge in RNN that appears when the network needs to make a prediction that requires context. The need to recognize the context can be handled in a regular RNN, but this is reliant on how far back the memory needs to preserve the context information. It is possible for the RNN to forecast a context in a basic sentence where the RNN must look back a series of words. It is far more challenging in a circumstance where the algorithm is asked to remember a paragraph. In theory, if the parameters are adjusted appropriately by hand, this can still be done using an RNN. Fortunately, scientists can use an LSTM instead of putting their temporal data into the adjusting parameters of a traditional RNN.

3.4.4 LSTM Model (Long Short-Term Memory)

Hochreiter and Schmidhuber (1997) proposed LSTM in 1997, and it quickly gained popularity, particularly for time series prediction issues. LSTM, which is a modified RNN approach, works well on a wide range of situations and is currently frequently utilized. Gate units and memory cells in the neural network design solves the problem of figuring out how to remember data over time. Cell states in memory cells store data that has been recently encountered. When information reaches a memory cell, the outcome is determined by a mixture of cell states, and the cell state is then updated. If the memory cell receives any new information, the output is processed using both the new information and the refreshed cell state. LSTM is designed to keep track of a situation with a long-term dependency. It is not something they learn via difficulty, rather it is their default behavior to remember knowledge for lengthy periods of time.

Standard RNNs have a basic structure that contains a single tanh layer in its rehashing module. The LSTM chain structure is similar to that of a regular RNN, but the rehashing module has a distinct structure. Data to or from the cell state can be included or excluded using the LSTM. The structures known as gates are in charge of managing these data. A gate is a method of determining whether or not data can enter the cell state. A sigmoid function and a point-wise multiplication algorithm are combined to form Gate. The sigmoid function can produce any value between 0 and 1. This number regulates data flow in such a way that an estimated zero means "don't pass anything" and an estimated one means "pass everything." Different gates are employed in the LSTM model to pass our recently experienced data from one cell to the next.

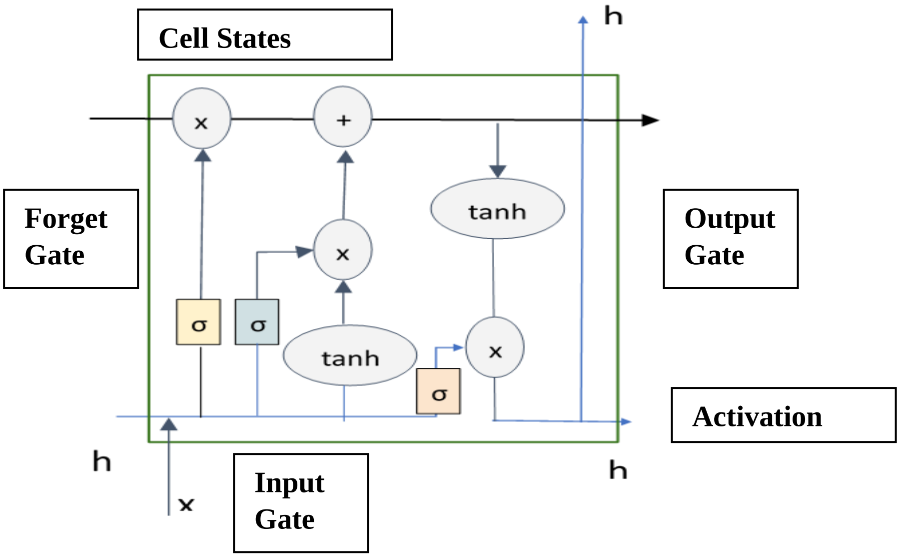

Figure 3.5: LSTM Cell Architecture

Update, forget, and output gates are the three types of gates. The memory of the LSTM model is controlled by these cells. Both activation values and candidate values were employed in LSTM. As a result, the cell produces two outputs from the LSTM: one is activation, and the other is candidate value. The data is transferred through the highest point, the level line. Cell state is the name given to this level line. As a result, the status of a cell is comparable to that of a transport line. With just modest linear cooperation, it flows straight down the entire chain. It can just pass data without any modifications.

Vanilla LSTM, Stacked LSTM and Bidirectional LSTM are some variations in LSTM model structure when using for univariate time series forecasting.

Vanilla LSTM

Vanilla LSTM is a prediction model with a single hidden layer of LSTM units and an output layer. The initial LSTM paper specified this as the standard and simplest LSTM model. Simple sequence problems can be solved with it.

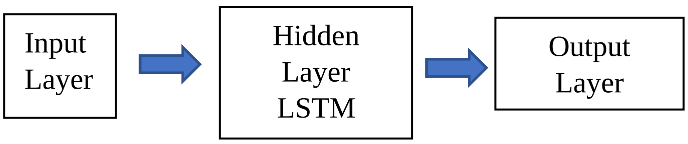

Figure 3.6 Simple Vanilla LSTM Architecture

Stacked LSTM

A Stacked LSTM model is created by stacking multiple hidden LSTM layers one on top of the other. An LSTM layer requires a three-dimensional input, and by default, LSTMs produce a two-dimensional output as a result of the sequence's conclusion.

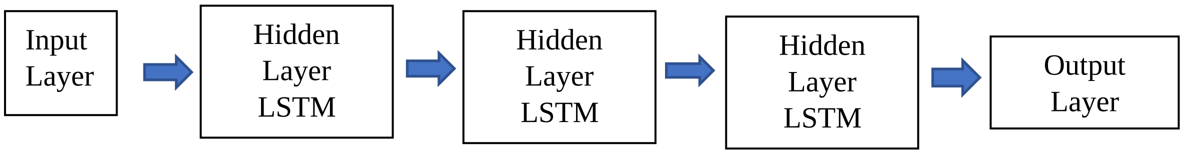

Figure 3.7 Stacked LSTM Architecture

Bidirectional LSTM

Allowing the LSTM model to learn the input sequence both forward and backwards and concatenate both interpretations can be advantageous in some sequence prediction tasks. A Bidirectional LSTM is what this is termed. By wrapping the first hidden layer in a wrapper layer termed Bidirectional, may create a Bidirectional LSTM for univariate time series forecasting.

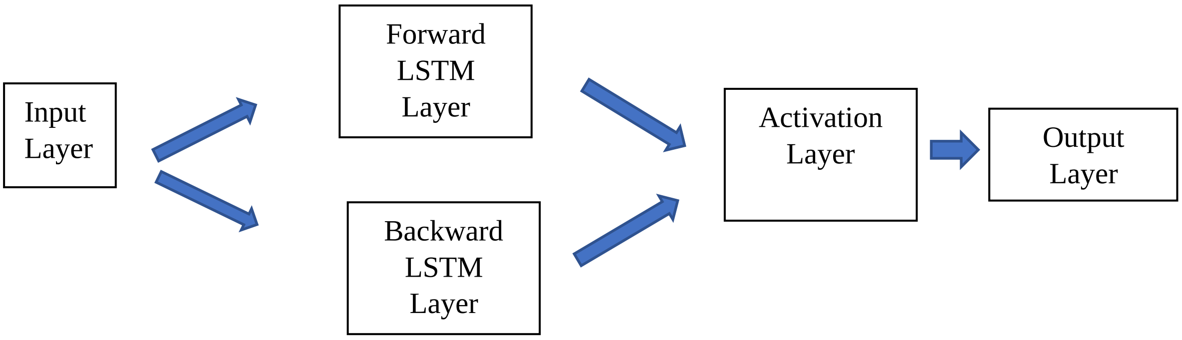

Figure 3.8: Bidirectional LSTM Architecture

3.4.5 Proposed Model

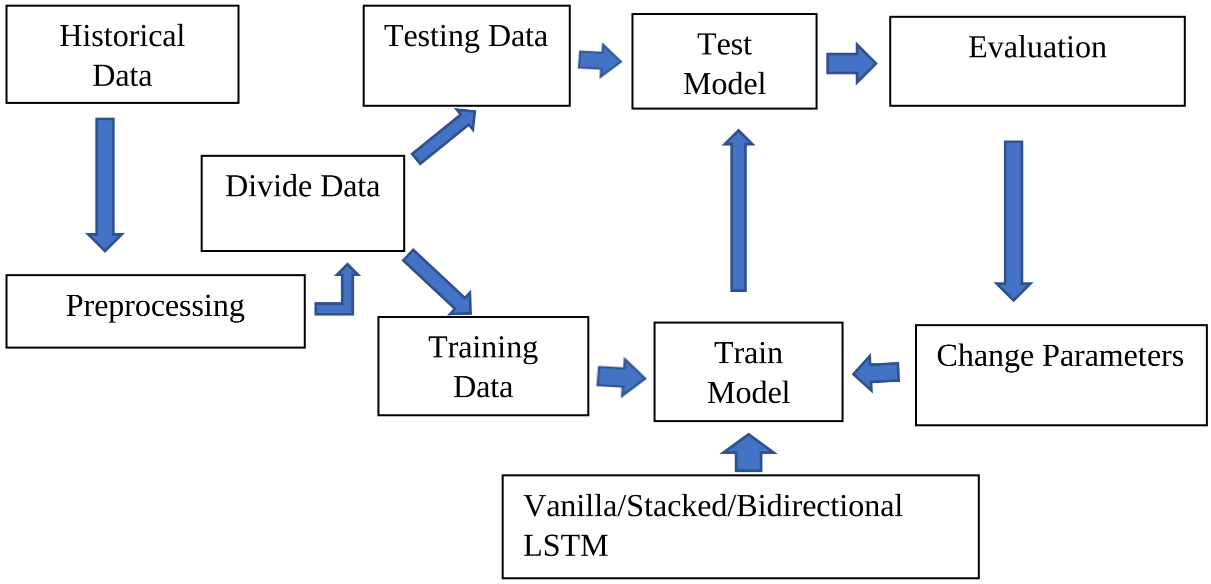

Figure 3.9: Proposed Model

Figure 3.9 shows the Research plan this plan will be execute throughout the project for predict the Coconut Price. Data will be preprocessed, and the Date and Price Column will be selected for LSTM Analysis. Data will be divided to two parts as train and test data Train data will be used to train the Model and Test Data will be used for evaluating purposes.

The steps in the LSTM Train Model are as follows.

Input : Historical Coconut Price data

Output : Prediction for Coconut Price

01.Start

02.Coconut Price data is taken and Stored in a NumPy array of 3 dimensions (N,W,F)

N : Number of training sequence

W: Sequence length

F: No of features of each sequence

03. A network Structure is built with [l,a,b,l] dimensions, where there is a l input layer, a neuron in next layer, b neuron in subsequent layer and a single layer with a linear activation function.

04. Train the Constructed network on data.

05. Use the output of the last layer as prediction of next time step.

06. Repeat 04 and 05 until optimal convergence is reached.

07. Obtain prediction by providing test data as input to the network.

08. Evaluate accuracy by comparing prediction made with actual data.

09. End.

3.4.6 LSTM Model Parameters

```
tf.keras.layers.LSTM(
    units,
    activation="tanh",
    recurrent_activation="sigmoid",
    use_bias=True,
    kernel_initializer="glorot_uniform",
    recurrent_initializer="orthogonal",
    bias_initializer="zeros",
    unit_forget_bias=True,
    kernel_regularizer=None,
    recurrent_regularizer=None,
    bias_regularizer=None,
    activity_regularizer=None,
    kernel_constraint=None,
    recurrent_constraint=None,
    bias_constraint=None,
    dropout=0.0,
    recurrent_dropout=0.0,
    return_sequences=False,
    return_state=False,
    go_backwards=False,
    stateful=False,
    time_major=False,
    unroll=False,
    **kwargs
)
```

- **units**: Positive integer, dimensionality of the output space.
- **activation**: Activation function to use. Default: hyperbolic tangent (tanh). If you pass None, no activation is applied (ie. "linear" activation: a(x) = x).
- **recurrent_activation**: Activation function to use for the recurrent step. Default: sigmoid (sigmoid). If you pass None, no activation is applied (ie. "linear" activation: a(x) = x).
- **use_bias**: Boolean (default True), whether the layer uses a bias vector.
- **kernel_initializer**: Initializer for the kernel weights matrix, used for the linear transformation of the inputs. Default: glorot_uniform.
- **recurrent_initializer**: Initializer for the recurrent_kernel weights matrix, used for the linear transformation of the recurrent state. Default: orthogonal.
- **bias_initializer**: Initializer for the bias vector. Default: zeros.
- **unit_forget_bias**: Boolean (default True). If True, add 1 to the bias of the forget gate at initialization. Setting it to true will also force bias_initializer="zeros".
- **kernel_regularizer**: Regularizer function applied to the kernel weights matrix. Default: None.
- **recurrent_regularizer**: Regularizer function applied to the recurrent_kernel weights matrix. Default: None.
- **bias_regularizer**: Regularizer function applied to the bias vector. Default: None.
- **activity_regularizer**: Regularizer function applied to the output of the layer (its "activation"). Default: None.
- **kernel_constraint**: Constraint function applied to the kernel weights matrix. Default: None.
- **recurrent_constraint**: Constraint function applied to the recurrent_kernel weights matrix. Default: None.
- **bias_constraint**: Constraint function applied to the bias vector. Default: None.
- **dropout**: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the inputs. Default: 0.
- **recurrent_dropout**: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the recurrent state. Default: 0.
- **return_sequences**: Boolean. Whether to return the last output. in the output sequence, or the full sequence. Default: False.
- **return_state**: Boolean. Whether to return the last state in addition to the output. Default: False.
- **go_backwards**: Boolean (default False). If True, process the input sequence backwards and return the reversed sequence.
- **stateful**: Boolean (default False). If True, the last state for each sample at index i in a batch will be used as initial state for the sample of index i in the following batch.
- **time_major**: The shape format of the inputs and outputs tensors. If True, the inputs and outputs will be in shape [timesteps, batch, feature], whereas in the False case, it will be [batch, timesteps, feature]. Using time_major = True is a bit more efficient because it avoids transposes at the beginning and end of the RNN calculation. However, most TensorFlow data is batch-major, so by default this function accepts input and emits output in batch-major form.
- **unroll**: Boolean (default False). If True, the network will be unrolled, else a symbolic loop will be used. Unrolling can speed-up a RNN, although it tends to be more memory-intensive. Unrolling is only suitable for short sequences.

Figure 3.10 LSTM Class Parameters

Source: https://keras.io/api/layers/recurrent_layers/lstm

Batch size, Epoch, Dense Layer, Dropout, and Loss function, cost function have been considered when training an LSTM model.

Batch Size

The Batch Size is the number of observations that the model must process before adjusting the parameter weights. batch size was selected as 32.

epoch

The training algorithm's whole traverse through the given dataset is referred to as an epoch. epochs were increased from 50 to 250 to compare the models.

Hidden Layers

One cannot say how many LSTM hidden layers are ideal for a specific sequence prediction task or LSTM architecture, just like we cannot say how many memory cells are best. When you have a lot of data, it is often wiser to go deeper.

Dense layer

Fully connected layer is another name for dense layer. It is the layer in which each neuron is connected to all of the neurons in the layer above it.

Dropout

Dropout is a training approach in which neurons are neglected at random during training, in other words, they are “dropped out” at random. As a result, their contribution to downstream neuron activation is removed temporally on the forward pass, and no weight changes are added to the neuron on the backward pass.

Loss function

A loss function is a function that measures a penalty such as square loss and is defined on a data point, prediction, and label. The sum of loss functions over the training set is the cost function. The Mean Squared Error is a representation of a cost function.

Activation function

Activation functions are an important aspect of a neural network's design. In a neural network, an activation function specifies how the weighted sum of the input is turned into an output from a node or nodes in a layer. To allow a neural network to simulate non-linear processes, activation is essential between matrix multiplications. The activation function of the hidden layer impacts how well the network model learns the training dataset. The activation function employed in the output layer determines the type of predictions the model can make.

The most common activation functions used in RNN modules are sigmoid, tanh, Relu.

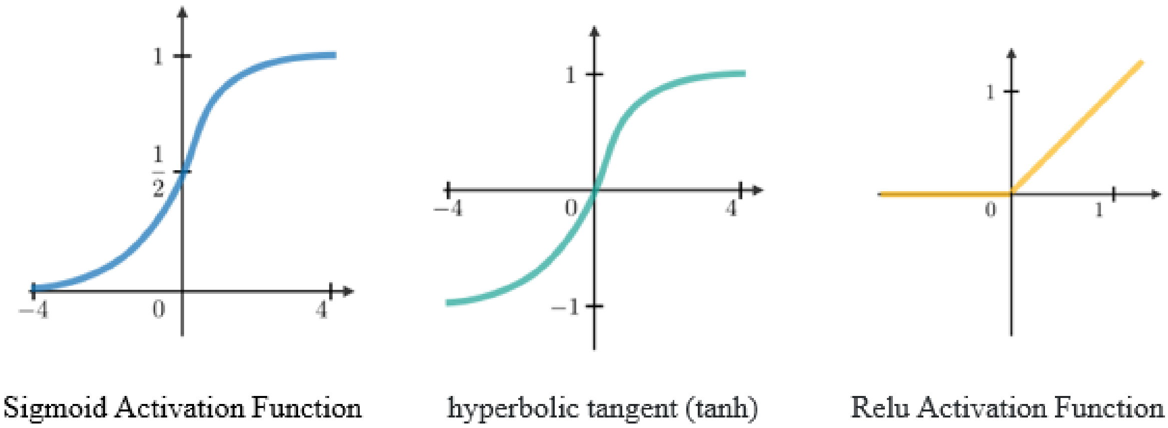

Figure 3.11 Activation Functions

Source: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture

The sigmoid activation function is also known as the logistic function. The same function is used in the logistic regression classification strategy. The function takes any real number as input and outputs a number between 0 and 1. The bigger the input (more positive), the closer the output gets to 1.0, and the smaller the input (more negative), the closer it gets to 0.0.
The hyperbolic tangent activation function is also known as the Tanh (also "tanh" and "TanH") function. It looks similar to the sigmoid activation function and even has the same S-shape. The function takes any real number as input and outputs a number between -1 and 1. The output gets closer to 1.0 as the input gets larger (more positive), while the output gets closer to -1.0 as the input becomes smaller (more negative).

The rectified linear activation function, or ReLU activation function, is the most commonly used function for hidden layers. It's popular because it's simple to use and successful at overcoming the limitations of other popular activation functions such as Sigmoid and Tanh. Although it is less prone to vanishing gradients, which prevent deep models from being trained, it can nevertheless suffer from saturated or "dead" units.

There are already a lot of non-linearities in a typical LSTM cell. In a single LSTM layer with one hyperbolic tangent (tanh) function and three sigmoid functions, the cell output is already the multiplication of two activations (a sigmoid and a hyperbolic tangent). There is no need to put another activation layer after the LSTM cell in this situation. If we connect the hidden output of one layer to the input of the stacked layer, the input will be routed through the sigmoid and hyperbolic tangents of the stacked layer's forget/input/output gates first.

Optimization Algorithm

The Adam method is an excellent default implementation of gradient descent. This is because it employs a specific learning rate for each parameter (weight) in the model automatically, combining the best features of the AdaGrad and RMSProp approaches. In addition, Adam's Keras implementation employs the best-practice initial settings for each of the setup parameters with different gradient descent algorithms. However, it is debatable whether Adam is the best gradient descent strategy for the model. Model performance should be evaluated.

Learning Rate

The learning rate determines how frequently the weights are updated in response to the calculated gradient at the conclusion of each batch. This can have a significant impact on the trade-off between the model's ability to learn the problem fast and effectively. Consider experimenting with different learning rates and momentum values using the basic stochastic gradient descent (SGD) optimizer.

The number of epochs and the learning rate are inextricably linked (number of passes through the training samples). In general, the lower the learning rate (for example, 0.0001), the more training epochs are needed. Because this is a linear connection, the converse is true: for higher learning rates, fewer epochs are required (e.g 0.1)

3.5 Evaluation

The model outcomes must be analyzed in the light of the business goals in the evaluation phase. New business requirements could arise during this process as a result of new trends discovered in the model results or other factors. Data mining is an iterative method for gaining market understanding. To proceed to the deployment process, must make a yes or no decision in this stage. Following Methods will be used to evaluate the model.

Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is a measure used to determine how reliable predictions or forecasts are in comparison to actual results. The mean absolute error is the average of the absolute errors, where yi is the estimate and fi is the true value, as the name means.

Mean Absolute Percentage Error (MAPE)

In statistics, such as trend estimation, the Mean Absolute Percentage Error (MAPE), also known as the Mean Absolute Percentage Deviation (MAPD), is an indicator of a forecasting method's prediction accuracy. Accuracy is normally expressed as a percentage.

Mean Squared Error (MSE)

The sum of the squares of the errors, or the difference between the estimator and the expected value, is calculated by the Mean Squared Error (MSE) of an estimator. MSE is a risk function that corresponds to the estimated value of a squared error or quadratic loss. If > y is a vector of n projections, and y is the vector of observed values for the function's inputs.

Root Mean Square Error (RMSE)

The difference between what a model predicts and what is actually observed is expressed as a percentage. The number of samples is multiplied by the sum of the squares of the differences between the predicted and actual values.

3.6 Deployment

The data mining knowledge or information must be provided in a way that enables stakeholders to use it when they need it. Depending on the business requirements, the implementation step may be as simple as producing a report or as complex as a repeatable data mining method across the enterprise.

During the deployment phase, plans for deployment, repair, and monitoring, as well as possible support, must be established. The final report must outline the project's experiences and review the project to assess what needs to be improved and what lessons must be learned from the project's perspective.

Power Bi Analytic tool was used to visualize predicted data. Using Power BI python Script feature. Using this tool, the user can filter the data visualize it as needed.

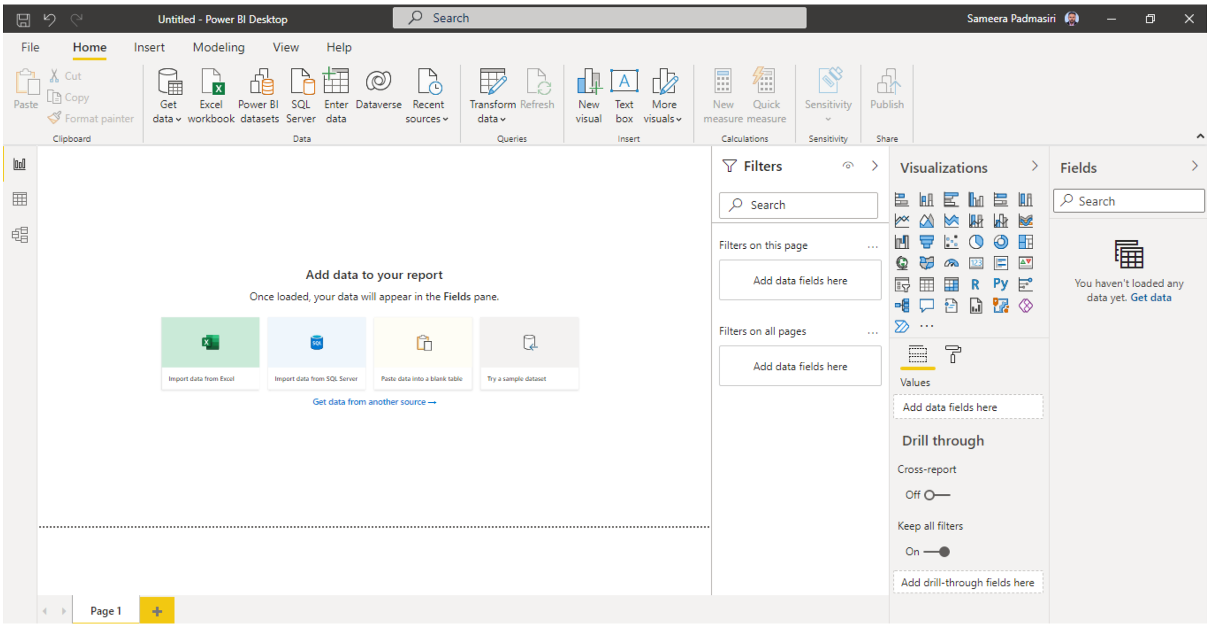

Figure 3.12: Power BI Desktop Interface

Figure 3.12 Shows the Power BI Desktop interface using this desktop version python script can be run and build python models. Using the Python Script models finally predicted data can be visualize using visualizations.

CHAPTER 4
IMPLEMENTATION AND RESULTS EVALUATIONS

This chapter will focus on explanatory analysis, model implementation and results evaluations and deployment. Explanatory analysis will analysis about raw data that used to create the LSTM model. Model implementation will discuss the model creation functions and their usage. Finally, results evaluation will be discussed about the model evaluation methods.

4.1 Explanatory Analysis

Explanatory Analysis describe the data numerically and graphically. Data is the most important factor in any data analysis task to understand the dataset python data visualization capabilities has used and generated the histogram and summery statistics.

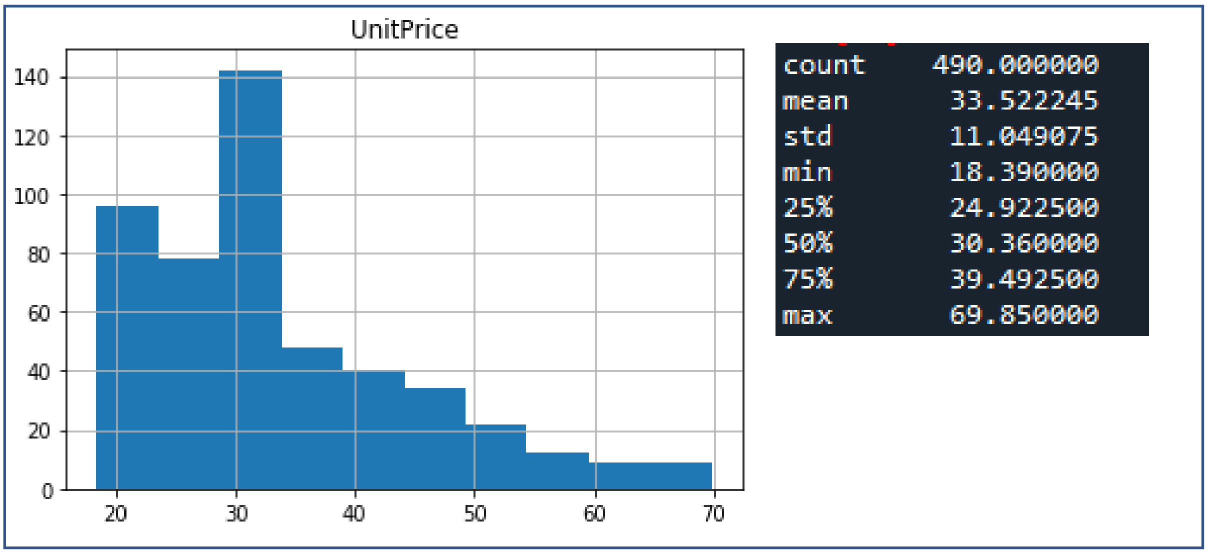

Figure 4.1 Coconut Price Histogram

Figure 4.1 Shows the frequency distribution of the Coconut price. Most data falls to the right side so it is a right skewed histogram with mean 33.52 and having 11.04 standard deviation. Maximum value of the dataset is 69.85 and the minimum value is 18.39 from this information it is showing that the Coconut price was between 18 and 69 Rupees in the given period. Average Coconut price is 33.52.

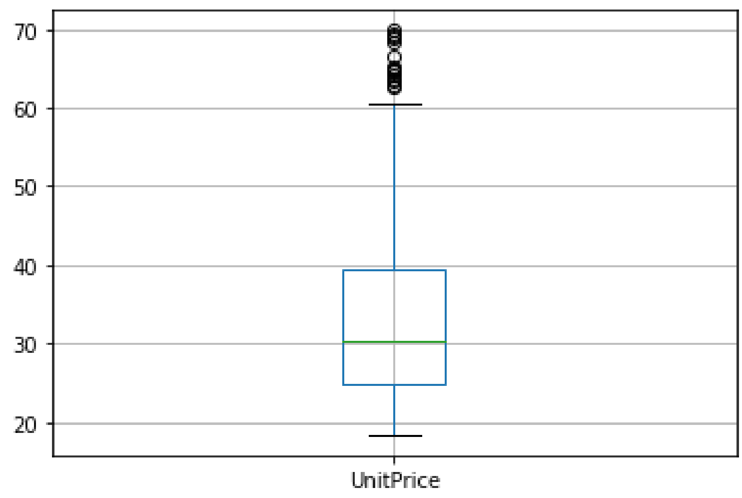

Figure 4.2: Coconut Price Box Plot

Figure 4.2 Coconut Price Box plot showing the Coconut price from 2010 to 2020. Box plot showing minimum, mean and maximum coconut prices as 18.39,33.52 and 69.85. According to the box plot it can be seen that no outlier values are included.

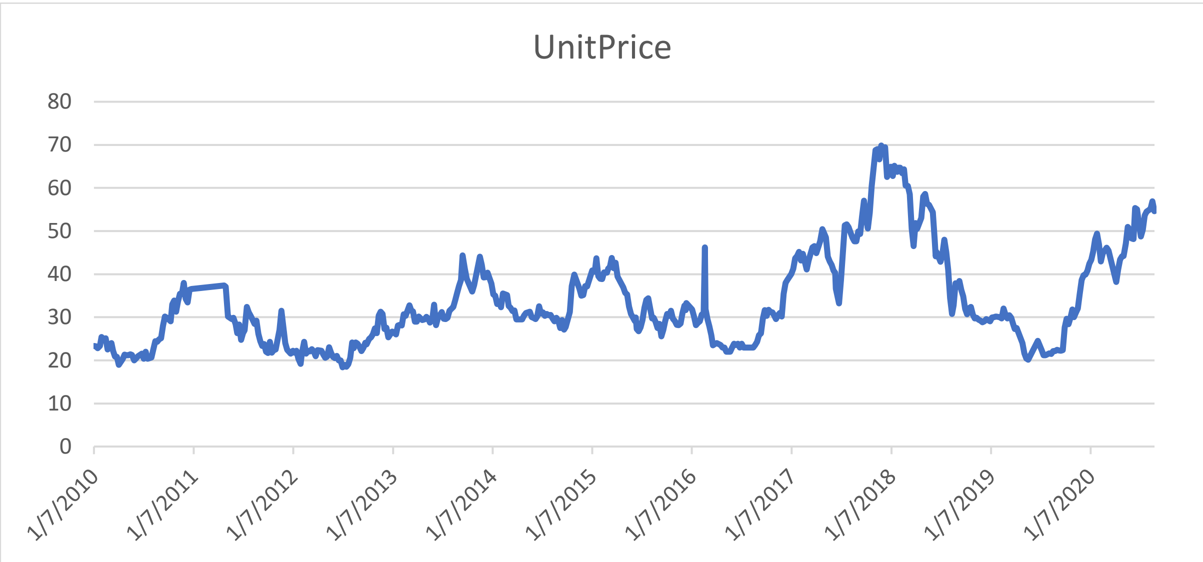

Figure 4.3: Coconut Price Weekly Fluctuation

The Figure 4.3 shows the Coconut price between year 2010 to 2020. In the Line chart X Axis Showing the Corresponding years and Y Axis Shows the Coconut price in Rupees. By Looking at the graph it can be see the positive trend with the year and the price fluctuations and also cyclic patterns are visible. It can be monitor that in 2018 the Coconut price is highest than the other years. It is needed to do the more analysis to check for seasonal patterns.

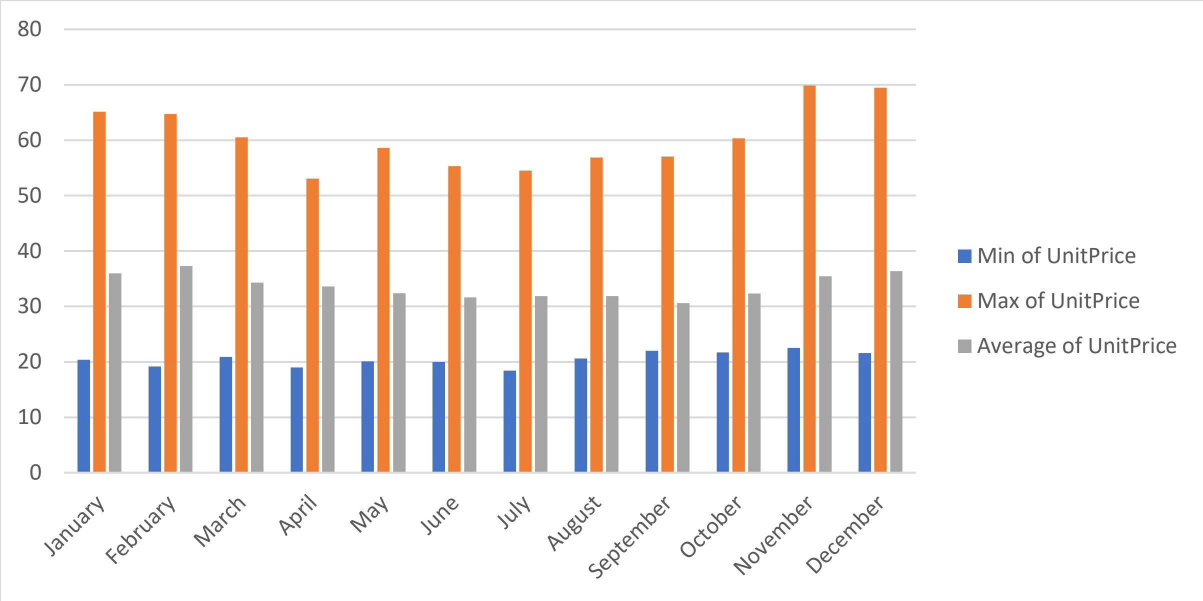

Figure 4.4: Coconut price variations with month

Figure 4.4 Shows the Coconut price variations corresponding to months. X Axis shows the months from January to December. Y Axis shows the Coconut prices from year 2010 to 2020. Bars shows the Average, minimum and maximum prices of coconut for 10 years with months. From this chart it can be conclude that from January to April maximum coconut price is decreasing and from September to December maximum Coconut price is increasing monthly but with the average and minimum monthly coconut price in 10 years are not showing huge variations. By Analyzing all the three prices it can be monitor that in the middle months the coconut has a low price than end and beginning of the years.

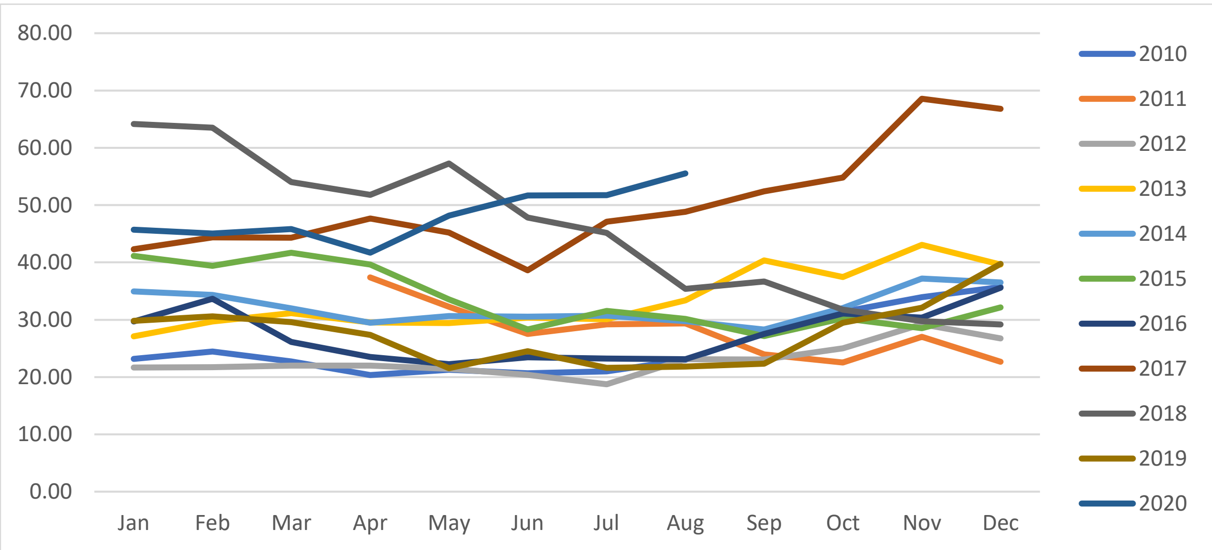

Figure 4.5: Coconut Price Monthly Fluctuation

Figure 4.5 Shows the Coconut price from year 2010 to 2020 with corresponding months and corresponding years. Line charts legends are showing the years and X Axis showing the months and Y Axis is displaying the Coconut price in Rupees. Different line colors show the years wise coconut prices. In the most of years it is showing that price is down in middle of the year and when comes to November month it is increasing the price. With this chart it can be seen that some seasonal patterns are monitored because same patterns are occurred in cyclic way as an example every year November month showing a coconut price increase.

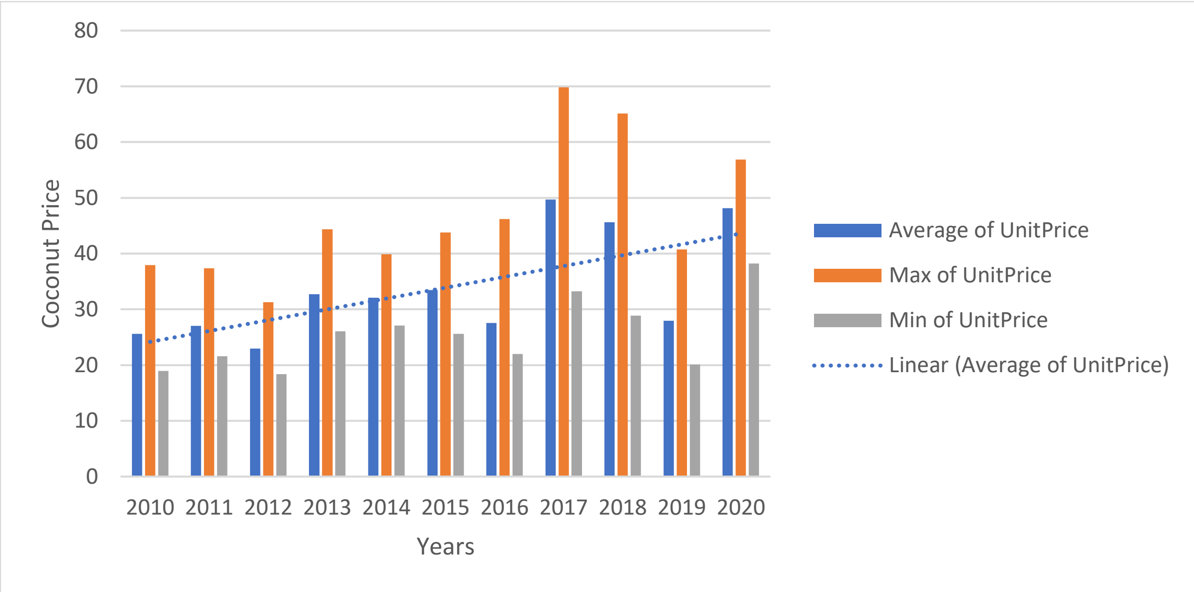

Figure 4.6: Coconut Prices Between Years

Figure 4.6 Shows the Coconut price yearly fluctuation with average, minimum and maximum Coconut prices. The X Axis shows the years, and the Y Axis shows the Coconut Price. Different Color bars are showing the average maximum and minimum Coconut price of corresponding years. By looking at the bar chart it can be monitor that year 2017 had the maximum coconut price and year 2010 had the minimum coconut price but if focus on the average coconut price it shows a positive trend with years.

Average of UnitPrice	Years										
Months	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Jan	23.19		21.64	27.10	34.96	41.15	29.71	42.30	64.16	29.79	45.69
Feb	24.48		21.70	29.69	34.33	39.38	33.67	44.38	63.53	30.60	45.07
Mar	22.70		21.99	31.15	31.96	41.69	26.11	44.34	54.05	29.61	45.82
Apr	20.35	37.39	21.97	29.51	29.48	39.66	23.52	47.65	51.81	27.36	41.71
May	21.25	32.27	21.44	29.44	30.67	33.58	22.25	45.19	57.26	21.55	48.18
Jun	20.68	27.53	20.38	30.36	30.53	28.30	23.46	38.62	47.86	24.51	51.68
Jul	20.97	29.20	18.74	30.09	30.72	31.52	23.22	47.13	45.17	21.60	51.76
Aug	23.00	29.35	23.12	33.39	29.78	30.13	23.13	48.84	35.40	21.85	55.53
Sep	27.61	23.93	23.08	40.37	28.29	27.16	27.54	52.42	36.67	22.34	
Oct	31.37	22.54	25.02	37.45	32.05	30.27	31.10	54.80	31.77	29.47	
Nov	33.94	27.02	29.22	43.07	37.19	28.51	30.38	68.56	29.76	32.12	
Dec	35.71	22.67	26.74	39.63	36.50	32.14	35.58	66.80	29.18	39.75	

Figure 4.7: Monthly Coconut Price

Figure 4.7 Showing the Average Monthly Coconut price with Corresponding years by looking at the figure it can be seen that some missing values on the year 2011. Missing values should be addressed when we apply the model. Using deletion or imputation methods can used for null value in this study deletion method is used.

4.2 Data Preparation

After Analyze the raw data next step is to prepare the raw data for model building. Data preparation stage consist of data clean tasks, handling missing values tasks, construct data tasks, Integrated data, outlier removal, feature selection and feature scaling tasks but for the selected data it should be use null value handling, data division for train and test data and feature scaling tasks.

4.2.1 Handling Null Values

Data should be clean before model creation the following process is used to check and remove the null values.

```
df=pd.read_csv('checknull.csv')

df.isnull().values.any()
df.info()
df = df.dropna()
```

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       490 non-null    object
 1   UnitPrice  490 non-null    float64
```

Figure 4.8: Data preparation results

“Isnull” function is used to check for the null values and “info” function gives the Figure 4.8 Output results. “dropna” funtion used to rows wise deletion of null values.

4.2.2 Data Divide and Feature Scalling

```
#Divide data train 90% to test 10%
training_set = df.iloc[:450, 1:2].values
test_set = df.iloc[450:, 1:2].values

# Normalize using Feature Scaling
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

Figure 4.9 Data Split and Feature Scalling

Figure 4.9 Shows the Data divide and feature scaling codes. “MinMaxScalar” function is used to normalize the data.

4.2.3 Data Reconstruct

For the Time series data should be a sequence it must consists past observation as input to output observations. and therefore, the data should reconstruct to match the LSTM models.

```
# Creating a data structure with 20 time-steps and 1 output
X_train = []
y_train = []
for i in range(20, len(training_set)):
    X_train.append(training_set_scaled[i-20:i, 0])
    y_train.append(training_set_scaled[i, 0])

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

Figure 4.10: Reconstruct Data

The data for an LSTM model must be in a certain format, which is commonly a 3D array created in 20 timesteps and converted to an array using NumPy. Then, at each step, create a 3D dimension array with x train samples, 20 timestamps, and one feature.

4.3 Long Short-Term Models Implementation

Keras library was used to develop LSTM model as shown in the Figure 4.11. It is needed to import a handful of Keras modules in order to build the LSTM. The initialization of the neural network should be done in a specific order. Dense allows you to add a densely linked neural network layer to your model. Dropout for adding dropout layers that prevent overfitting, LSTM for adding the Long Short-Term Memory layer. In keras model build as a sequence of layers by adding layers.

```
from keras.models import Sequential
from keras.layers import Bidirectional
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
```

Figure 4.11 Keras LSTM Libraries

4.3.1 Vanilla LSTM

```
model = Sequential()
model.add(LSTM(50, input_shape=(X_train.shape[1], 1)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
history=model.fit(X_train, y_train ,epochs = 150 , batch_size = 32)
```

Figure 4.12 Vanilla LSTM model creation

As Figure 4.12 shows first add the LSTM Layer. LSTM Layer needs parameters to be set. As for the hidden nodes 50 is selected by analyzing previous model Creations. “input_shape” function need to input parameters time steps value and the number of features values. dense layer is used to provide a one-unit output. After that, use the popular adam optimizer to assemble our model and set the loss to mean squared error. The mean of the squared errors will be computed as a result of this. The model was then tweaked to run on 50 to 250 epochs at a time, with a batch size of 32.

4.3.2 Stacked LSTM

```
model = Sequential()
#Adding the first LSTM layer and some Dropout regularisation
model.add(LSTM(units = 50,return_sequences = True, input_shape = (X_train.shape[1], 1)))
model.add(Dropout(0.2))
# Adding a second LSTM layer and some Dropout regularisation
model.add(LSTM(units = 50, return_sequences = True))
model.add(Dropout(0.2))
# #Adding a third LSTM layer and some Dropout regularisation
model.add(LSTM(units = 50, return_sequences = True))
model.add(Dropout(0.2))
# Adding a fourth LSTM layer and some Dropout regularisation
model.add(LSTM(units = 50))
model.add(Dropout(0.2))
#Adding the output layer
model.add(Dense(units = 1))

# Compiling the RNN
model.compile(optimizer = 'adam', loss = 'mse',metrics=['accuracy'] )
# Fitting the RNN to the Training set
history=model.fit(X_train, y_train, epochs = 150, batch_size = 32)
```

Figure 4.13 Stacked LSTM model creation

As Figure 4.14 shows to avoid overfitting, add the LSTM layer first, followed by a few Dropout layers. The LSTM layer is added with the following inputs. Return sequences set to True to controls whether to return the final output in the output sequence, or the entire sequence input shape as the shape of our training set. Another three LSTM Layers are added for stacked LSTM to check how the model is responding. Dropout layer was set to 0.2, which means that 20% of the layers will be dropped. After that, we add the Dense layer, which defines a single unit output. After that, we use the popular adam optimizer to assemble our model and set the loss to mean squarred error. The mean of the squared errors will be computed as a result of this. The model was then adjusted to run on 50 to 250 epochs with a batch size of 32.

4.3.3 Bi-directional LSTM

```
model = Sequential()
model.add(Bidirectional(LSTM(50, input_shape=(X_train.shape[1], 1))))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mae',metrics=['accuracy'])

history=model.fit(X_train, y_train, epochs = 150, batch_size = 32)

score=model.evaluate(X_train, y_train, batch_size = 32,verbose=1)
print(' mean_squared_error:', score)
```

Figure 4.14 Bi-directional LSTM model creation

As figure 4.14 shows bidirectional LSTM layer is added with the 50 hidden nodes. The parameters time steps and number of features must be entered into the “input shape” function. To get a one-unit output, a dense layer is used. After that, create model with the popular Adam optimizer and set the loss to mean squared error. As a result, the mean of the squared errors will be calculated. After then, the model was adjusted to run on 50 to 250 epochs at a time with a batch size of 32.

```
Epoch 147/150
430/430 [==============================] - 1s 1ms/step - loss: 0.0298
Epoch 148/150
430/430 [==============================] - 1s 1ms/step - loss: 0.0301
Epoch 149/150
430/430 [==============================] - 1s 1ms/step - loss: 0.0300
Epoch 150/150
430/430 [==============================] - 1s 1ms/step - loss: 0.0295
430/430 [==============================] - 1s 3ms/step
 mean_squared_error: 0.029388378101379373
Model: "sequential_24"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
bidirectional_7 (Bidirection (None, 100)              20800
_________________________________________________________________
dense_21 (Dense)            (None, 1)                 101
=================================================================
Total params: 20,901
Trainable params: 20,901
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.15 Bi-directional LSTM model results

Figure 4.15 shows the model building and output results of bidirectional LSTM model. According to the Figure 4.15 mean squared error value shows as 0.02938. No of epoch is 150. These results can be generated using model summary function.

4.4 LSTM models Evaluations

Long Short-term memory model's variants were used to produce the results by modifying the parameters and layers of the model. For each modifications different mean squared error values were detected between the real data and predicted data in model building stage. Smaller Mean squared values and smaller Mean absolute vales gives the best results.

Table 4.1: A review of the MSE between Vanilla, Stacked LSTM and Bi-LSTM model for several epochs.

No of Epochs	Vanilla LSTM (MSE)	Stacked LSTM (MSE)	Bi-Directional LSTM (MSE)
50	0.00272	0.00568	0.00331
100	0.00199	0.00263	0.00175
150	0.00176	0.00209	0.00157
200	0.00171	0.00221	0.00170
250	0.00169	0.00222	0.00191

Table 4.1 Shows the Mean Squared error results for different LSTM Variants used in the study. Results were generated by changing the epochs values 50 to 250 the epochs values are selected to compare the epochs. Table Column shows the different LSTM Variants, and the Rows are Showing epochs values. With batch Size 32 and using 50 epochs the Vanilla LSTM MSE value is 0.00272 Stacked LSTM MSE Value is 0.00568 Bidirectional LSTM value is 0.00331 by comparing MSE values in 50 epochs Vanilla LSTM gives the best results. In the 100 epochs the Vanilla LSTM MSE value is 0.00199, Stacked LSTM MSE Value is 0.00263, Bidirectional LSTM MSE value is 0.00175 by comparing these results it shows that Bidirectional LSTM gives the best model for 100 epochs. In the 150 epochs the Vanilla LSTM MSE value is 0.00176, Stacked LSTM MSE Value is 0.00209, Bidirectional LSTM MSE value is 0.00157 by comparing these results it shows that Bidirectional LSTM gives the best model for 150 epochs. In the 200 epochs the Vanilla LSTM MSE value is 0.00171, Stacked LSTM MSE Value is 0.00221, Bidirectional LSTM MSE value is 0.00170 by comparing these results it shows that Bidirectional LSTM gives the best model for 200 epochs. In the 250 epochs the Vanilla LSTM MSE value is 0.00169, Stacked LSTM MSE Value is 0.00222, Bidirectional LSTM MSE value is 0.00191 by comparing these results it shows that Bidirectional LSTM gives the best model for 250 epochs. When considering all the values the 150 epochs Bidirectional LSTM gives the best results according to this table.

By considering the table 4.1 monitoring at mean squared values individually it can be seen that Vanilla LSTM mean squared values are reducing further when epochs are increasing therefore to check the loss of Vanilla LSTM further analysis perform using Figure 4.16.

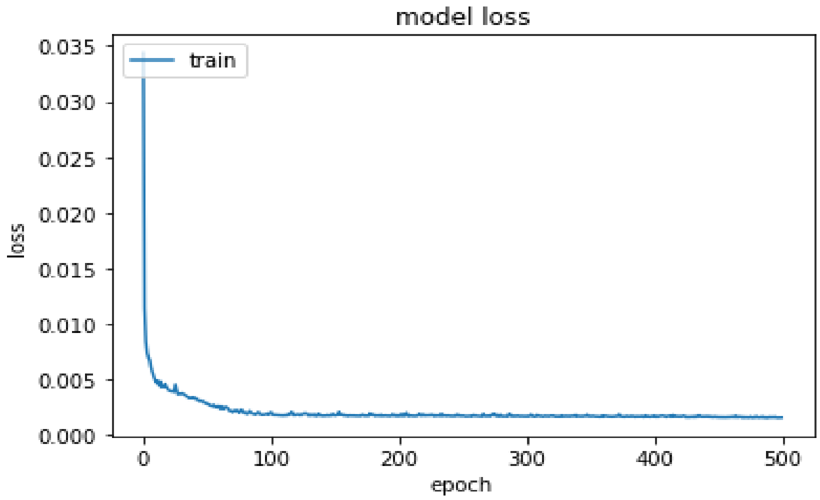

Figure 4.16: Vanilla LSTM Loss further analysis

Figure 4.16 Shows the relation between mean squared error and the epochs. X Axis shows the no of epochs and Y Axis show the loss value (mean squared error). When increasing the number of epochs, the loss is decreasing. It has massive loss on the start and when increasing after 100 epochs the loss value doesn’t show major fluctuation. Loss value has begun in 0.035 and has decreased to the 0.035.

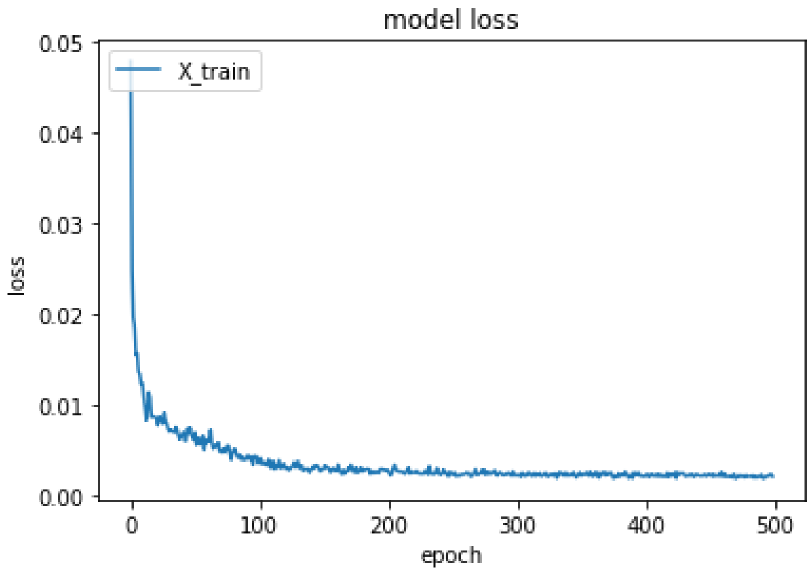

Figure 4.17: Stacked LSTM Loss further analysis

Figure 4.17 Shows the relationship of stacked LSTM model with mean squared error and epochs. It shows a massive dropdown of loss value from 0 to 100 but not like vanilla LSTM it shows the Loss value fluctuations between 0 to 100 after the 150 epochs the loss values are showing the same value less than 0.01.

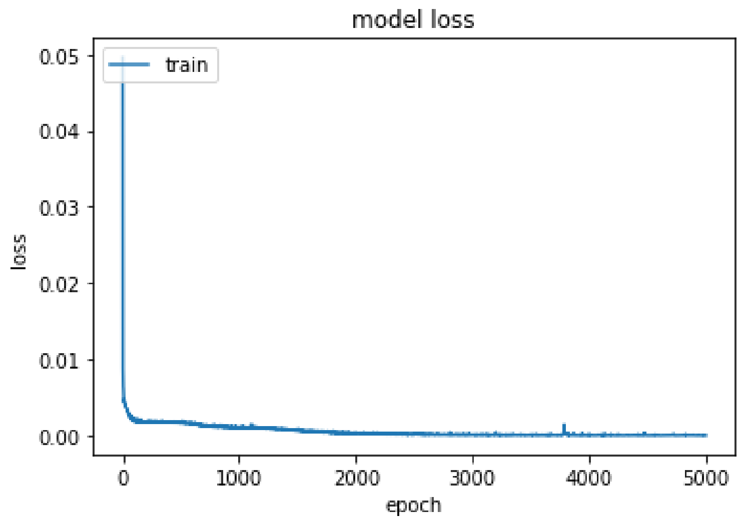

Figure 4.18: Bi-directional LSTM further analysis

Figure 4.18 Shows the relationship between mean squared values and the epochs values for Bidirectional LSTM model. X Axis shows the epoch values, and the Y axis shows the mean squared values. In the start of the LSTM model, it is showing huge dropdown of loss value. Mean squared values are showing between 0.05 and 0 values for bidirectional LSTM model. As a result of further analysis, the mean squared value get straight line after 3000 epochs. Using the Figure 4.18 the best epochs to build the prediction model shows as the 3000 but when we use more epochs the model building time will be increase.

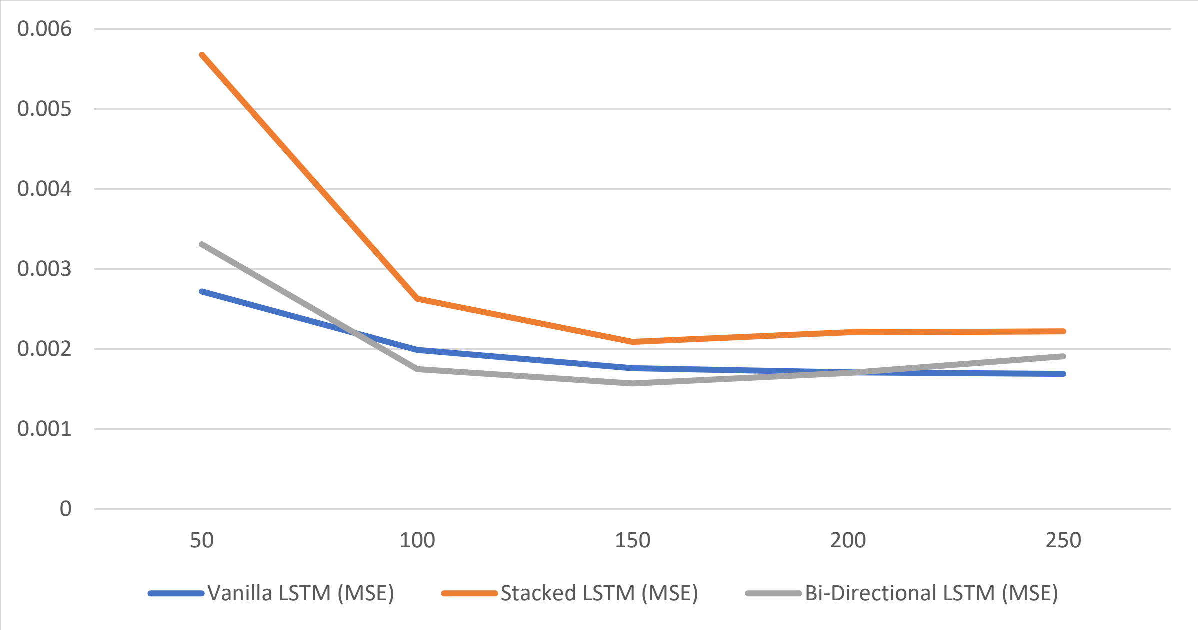

Figure 4.19: MSE between Vanilla, Stacked LSTM and Bi-LSTM model

Figure 4.19 Shows the Mean squared values of different LSTM models. From the X Axis showing the epochs values and from Y Axis it is showing the Mean squared values. By further examining the Table 4.1 and Figure 4.19, it should be mentioned that the number of epochs increases the accuracy of the results, however some places in the training model have a under fitting problem. The accuracy of the Vanilla LSTM model increased as the epochs increased, but the accuracy of the Stacked LSTM model did not increase after 150 epochs. However, the Bidirectional LSTM model recorded more accuracy than the Vanilla and Stacked LSTM models, with 150 epochs giving the best accuracy for Bidirectional LSTM model. For all the models the best result was offered by the 150 epochs. 150 epochs can be identified as the best prediction giving model training runs.

Table 4.2: A review of the Mean Squared Error and Mean Absolute error between Vanilla, Stacked LSTM and Bi-LSTM model for 150 epochs.

Evaluation Method	Vanilla LSTM	Stacked LSTM	Bi-Directional LSTM
MAE	0.02949	0.03066	0.02944
MSE	0.00176	0.00209	0.00157

Table 4.2 Shows the MAE and MSE Values for the LSTM Variants. Using the above 4.1 Table 150 epochs has selected as best epochs parameter. By looking at the MAE values the Bidirectional LSTM has the best results.

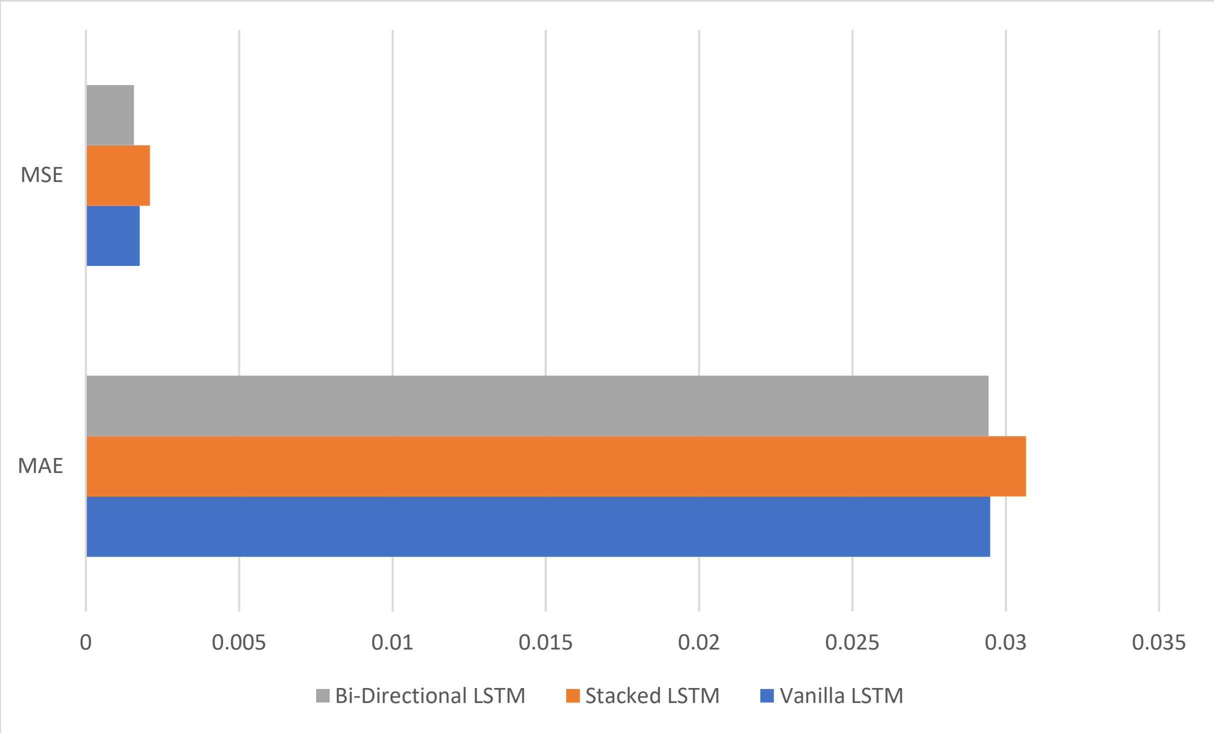

Figure 4.20: Mean Absolute Error and Mean Standard Error for LSTM Variations 150 epoch

Figure 4.20 is a bar chart it Shows the Value of MES and MAE for the LSTM variants. Stacked LSTM Shows more error values than the Vanila and Bidirectional LSTM with MAE value 0.03066 and MSE value 0.00209 vanila LSTM Shows the MAE value 0.02946 and 0.00176 Bidirectional LSTM Shows value MAE value 0.02944 and 0.00157. According to the Table 4.2 and Figure 4.6 Bar chart comparing Mean Absolute error and Mean Standard error. It suggests that more than Vanilla LSTM and Stacked LSTM Bi-Directional LSTM Perform well with Current Dataset for 150 epochs.

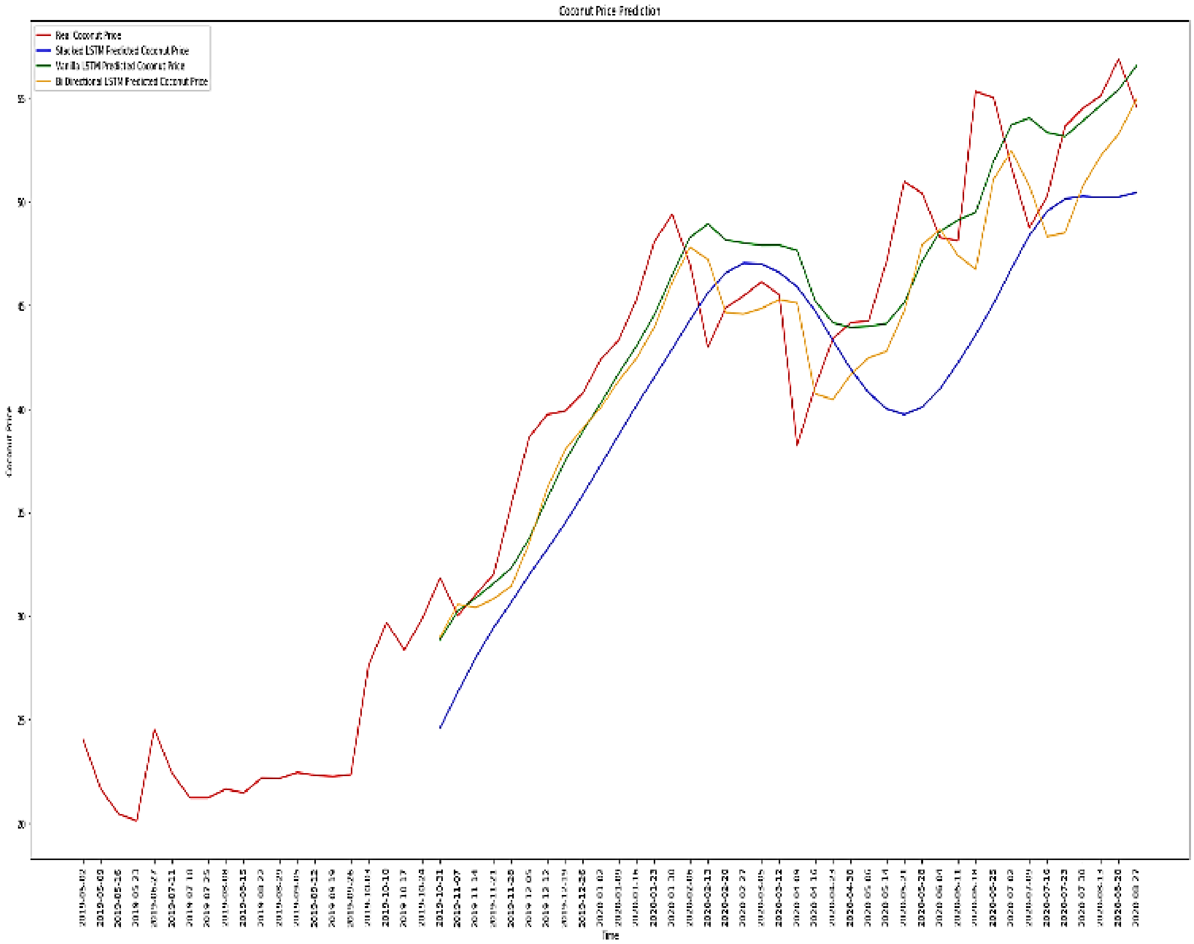

Figure 4.21: LSTM model results prediction for 50 epochs

Figure 4.21 is a line chart the X Axis shows the Dates and the Y Axis Shows the Coconut Price. Different Colored lines has used to show draw the line comparing real Coconut price, Vanilla LSTM Predicted price, Stacked LSTM Predicted Price and for the Bidirectional LSTM Prediction price. The LSTM model prediction results batch size was selected as 32 has been used for training. Blue line shows the Stacked LSTM model, green line shows the Vanilla LSTM model and orange line Shows the Bidirectional LSTM model. the Mean Squared error was 0.00272 for Vanilla LSTM. With the same parameters the Stacked LSTM mean squared error was 0.00568 and with same parameters 0.00331 was Bidirectional LSTM Model mean squared error. According to these results it was Vanilla LSTM which gives best result in 50 epochs.

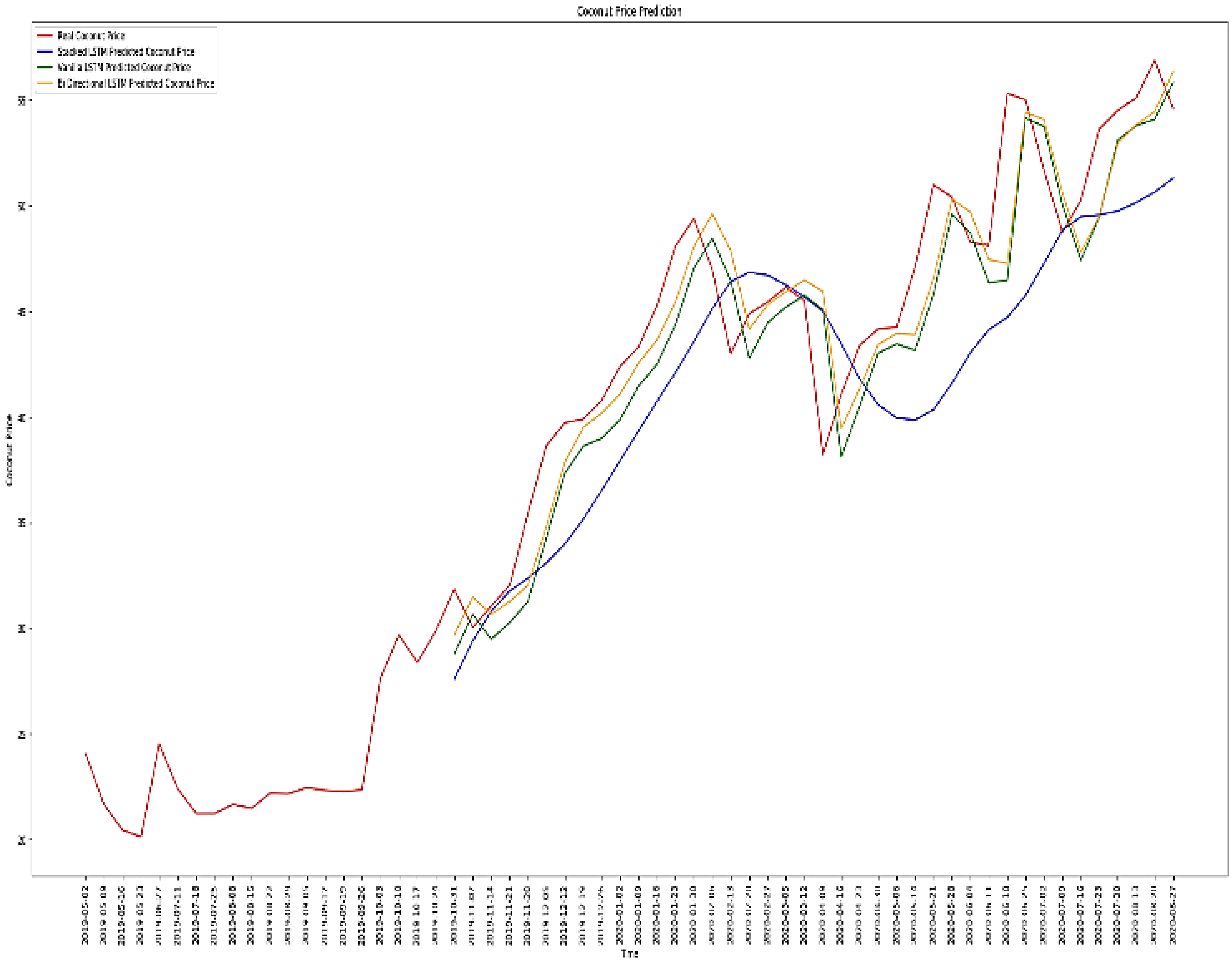

Figure 4.22: LSTM model results prediction for 100 epochs

Figure 4.22 Shows the predicted results of LSTM models. X Axis shows the date value, and the Y Axis shows the Coconut price. Different Colored lines have used to show the LSTM variants Coconut prices. The train algorithms will run 100 times in train data set according to the results the Vanila LSTM mean squared error value is 0.00199 Stacked LSTM MSE value is 0.00263 and Bidirectional LSTM model mean squard error is 0.00175 the lowest value is the best accuracy model with 100 epochs the Bidirectional LSTM model is the best.

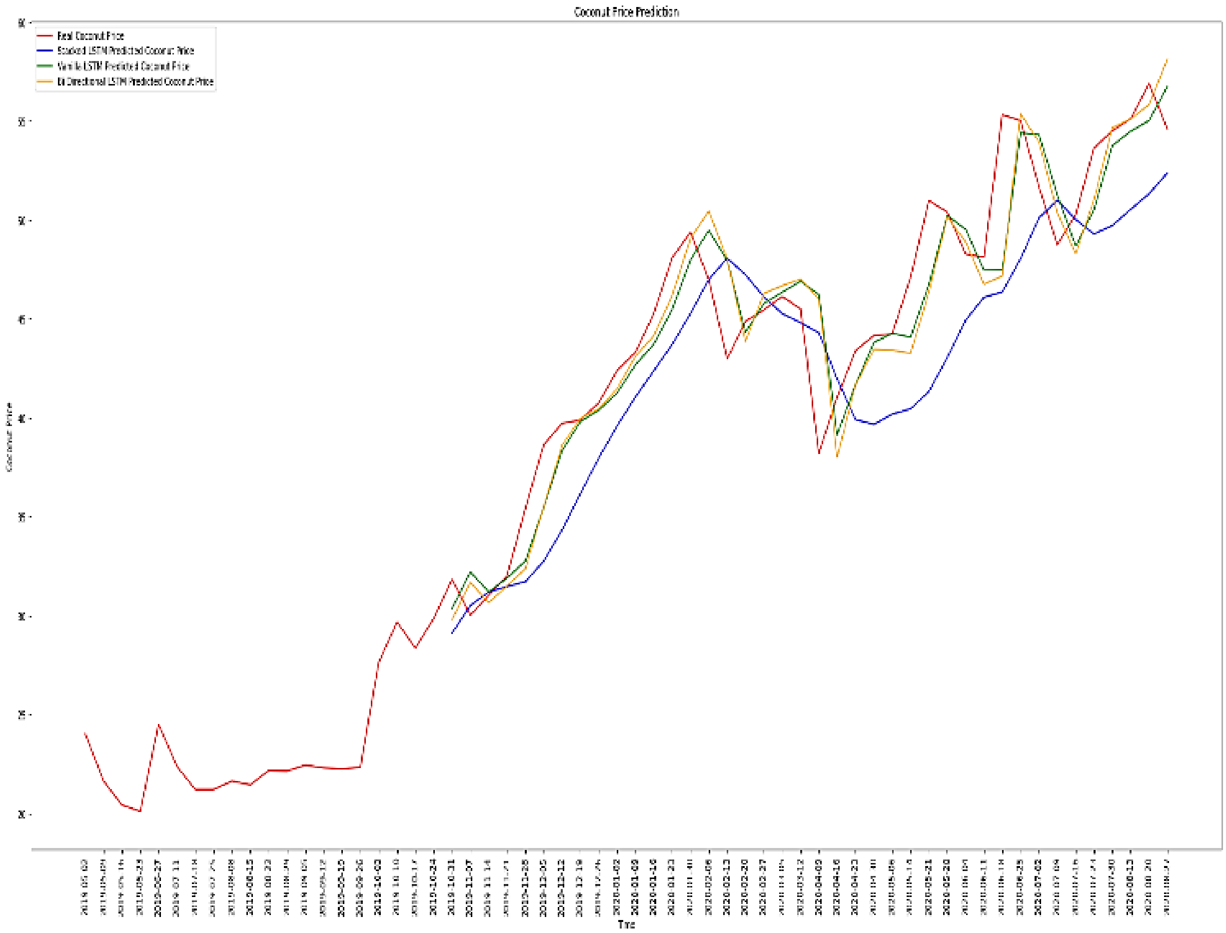

Figure 4.23: LSTM model results prediction for 150 epochs

Figure 4.23 represent the model train results for Vanilla, Stacked and Bidirectional LSTM models. X Axis shows the date value, and the Y Axis shows the Coconut price. Different Colored lines have used to show the LSTM variants Coconut prices. the mean squared error value for the models are 0.00176, 0.00209, 0.00157 the lowest value is the 0.00157 which is Bidirectional Model therefore the 150 epochs best model is Bidirectional Model.

Figure 4.24: LSTM model results prediction for 200 epochs

Figure 4.24 Shows the Prediction results for LSTM models blue line shows the Stacked LSTM model green line shows the Vanilla LSTM model and orange line Shows the Bidirectional LSTM model. Mean Squared errors for the LSTM model are for Vanilla LSTM 0.00171 and for Stacked LSTM 0.00221 and 0.00170 for the Bidirectional model. In the 200 epochs the vanilla LSTM and Bidirectional Model showing the similar results.

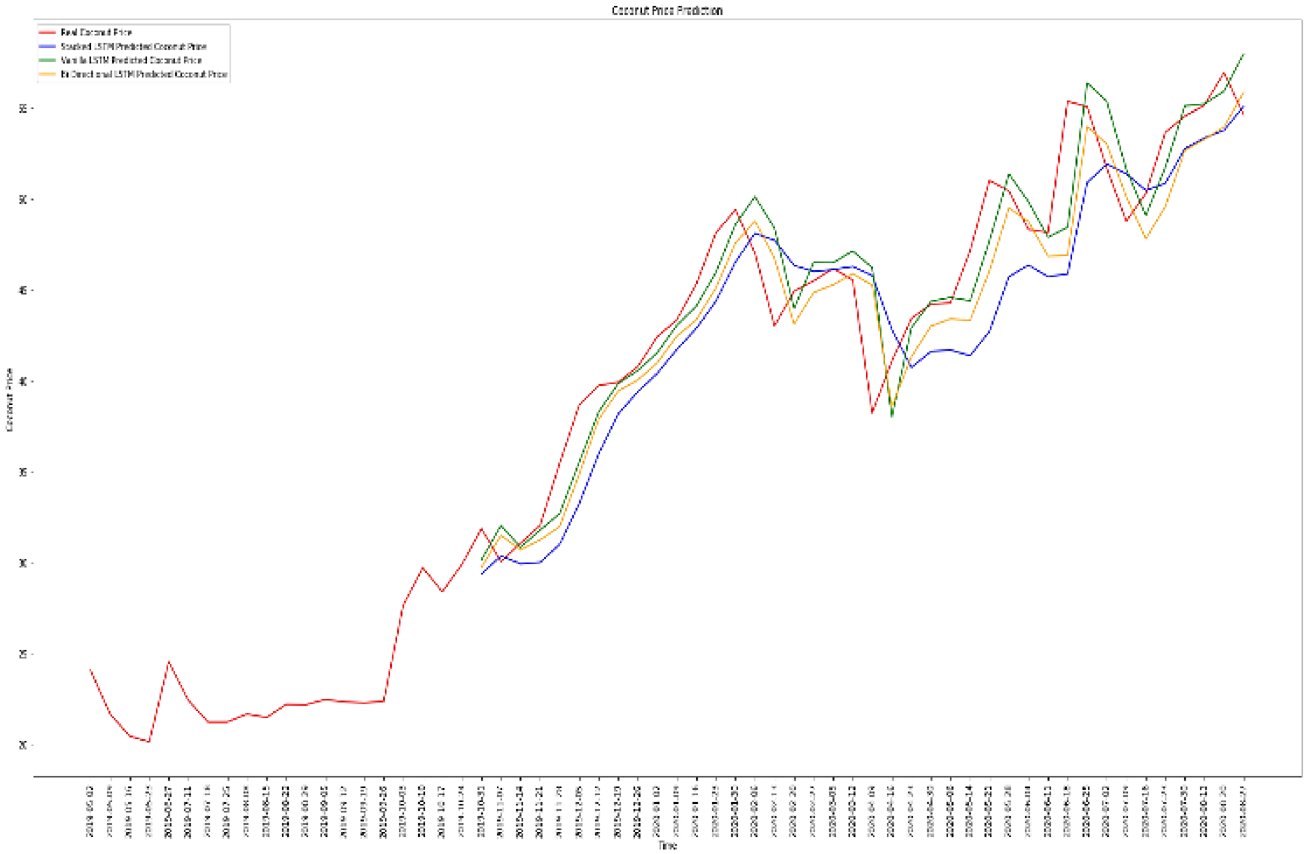

Figure. 4.25: LSTM model results prediction for 250 epochs

Figure 4.25 shows the results of LSTM models in 250 runs. The X Axis shows the Date and Y Axis shows the price. Using the line chart, we can monitor that with the date the Coconut price as increased. By examining the Bidirectional LSTM Line, it is visual showing the Aline with the real data by examining the mean squared error for the Vanilla LSTM is 0.00169 Stacked LSTM is 0.00222 and 0.00191 for the Bidirectional LSTM in the 250 epochs vanilla LSTM is the best model match.

Table 4.3 Predicted Results for the Bi-directional LSTM Model

Date	Real Price	Predicted Price			Squared Error		
		Bidirectional LSTM	Vanila LSTM	Stacked LSTM	Bidirectional LSTM	Vanila LSTM	Stacked LSTM
1/2/2020	42.39	42.04	41.50	40.83	0.12	0.79	2.45
1/9/2020	43.31	42.82	43.26	42.56	0.24	0.00	0.56
1/16/2020	45.27	43.73	44.26	43.53	2.36	1.02	3.01
1/23/2020	48.08	45.69	46.28	45.49	5.73	3.25	6.70
1/30/2020	49.38	48.56	49.19	48.21	0.68	0.04	1.38
2/6/2020	46.97	49.65	50.42	49.30	7.19	11.93	5.44
2/13/2020	42.98	45.84	47.68	47.04	8.17	22.08	16.51
2/20/2020	44.88	42.93	43.56	43.82	3.81	1.74	1.12
2/27/2020	45.45	45.54	46.06	46.04	0.01	0.37	0.35
3/5/2020	46.13	46.07	46.55	46.45	0.00	0.17	0.10
3/12/2020	45.5	46.29	46.96	46.85	0.63	2.14	1.81
4/9/2020	38.22	44.55	46.06	46.08	40.11	61.45	61.71
4/16/2020	41.05	37.83	38.04	39.48	10.40	9.06	2.46
4/23/2020	43.39	41.33	41.86	42.06	4.26	2.34	1.76
4/30/2020	44.16	43.84	44.18	44.16	0.10	0.00	0.00
5/6/2020	44.24	43.70	44.35	44.40	0.29	0.01	0.03
5/14/2020	47.1	43.58	44.14	44.24	12.40	8.75	8.19
5/21/2020	50.98	46.68	47.20	46.95	18.48	14.28	16.27
5/28/2020	50.4	50.56	51.13	50.48	0.03	0.53	0.01
6/4/2020	48.27	49.22	49.93	49.45	0.91	2.77	1.40
6/11/2020	48.12	46.91	47.64	47.82	1.46	0.23	0.09
6/18/2020	55.32	47.38	48.01	48.31	62.98	53.49	49.09
6/25/2020	55.02	55.44	56.21	55.66	0.17	1.41	0.41
7/2/2020	51.67	54.06	55.04	54.04	5.72	11.33	5.64
7/9/2020	48.74	49.98	51.14	51.26	1.54	5.75	6.35
7/16/2020	50.26	48.05	48.66	49.44	4.87	2.55	0.67
7/23/2020	53.63	52.29	51.00	51.39	1.79	6.90	5.00
7/30/2020	54.51	53.74	54.58	54.33	0.59	0.01	0.03
8/13/2020	55.11	53.98	54.91	54.43	1.27	0.04	0.47
8/20/2020	56.89	54.50	55.33	54.94	5.73	2.44	3.81
8/27/2020	54.59	55.69	57.30	56.75	1.20	7.37	4.66
		Mean Squared Error			**6.56**	**7.56**	**6.69**

Table 4.3 is showing the Real Price of Coconut and Predicted price of Coconut with all the LSTM Variants models with mean squared errors of test data. This predicted price is generated by using the Vanilla, Stacked and Bi-directional models. Using this table comparing the test values and actual values the mean squared values are shown near to each other, but bidirectional

model shows the less value 6.56 than the other Stacked and vanilla LSTM models which MSE were 6.69 and 7.56. it can be decided that bidirectional LSTM is more suitable to predict the coconut price for this data set.

4.5 Deployment

Purpose of the deployment of the created model to give access to the users to use the predicted results. The best way the use can access the result is a web portal then from the web portal any user can login from any place and using any device. To succeed this purpose Power Bi Analytic tool was used to visualize and predict the data. Using Power BI python Script feature. The model was built inside the Power BI tool.

Using HTML technologies, the main page of the web portal is built. The Predicted data was published in web site for user access. User Should have the Username and Password to login the web site. Admin user has to assign the username and password for the user if someone needed to access the web site.

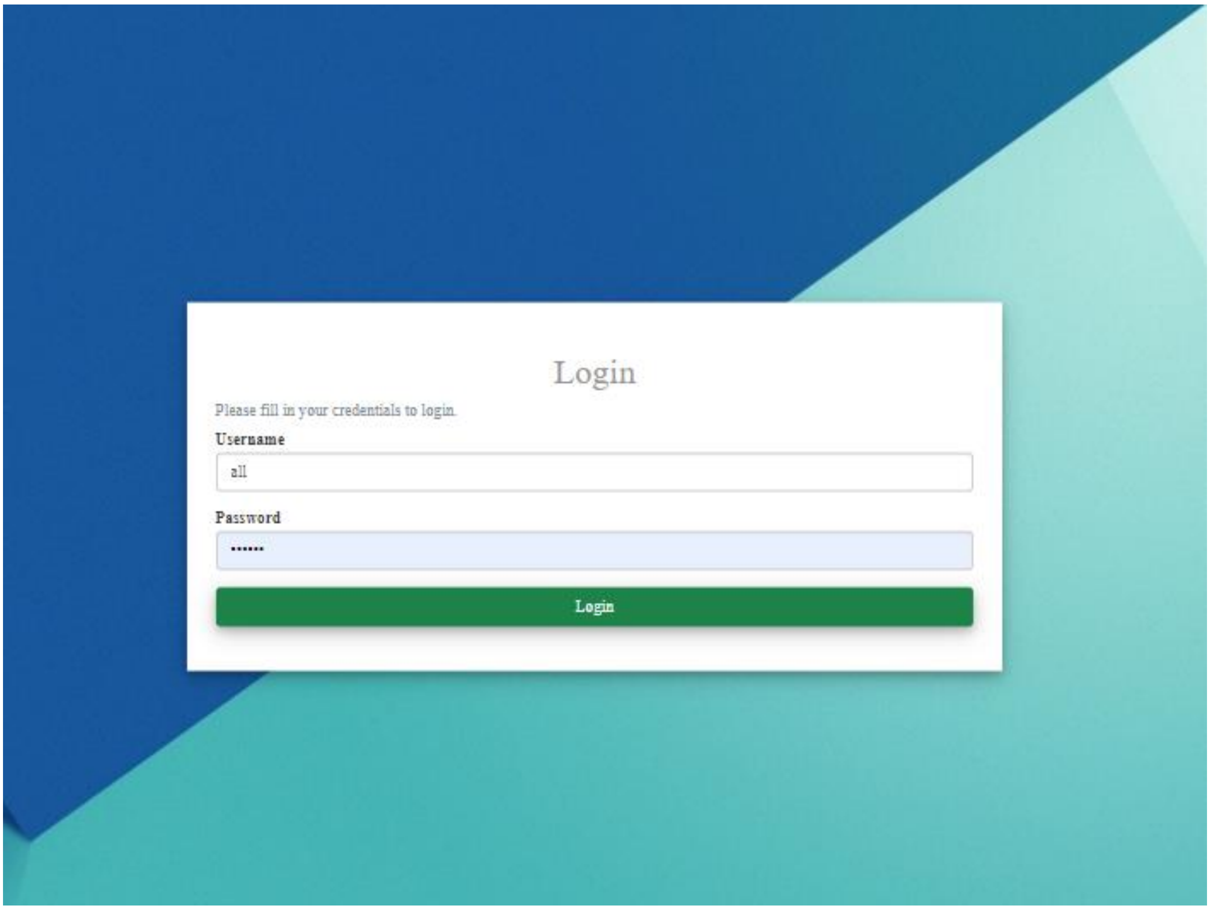

Figure 4.26 Web site Login Page

Figure 4.26 Shows the web portal used to login to the Coconut price prediction module. The user has need to login to the System using web link. When user enter to the site after username is inserted if username is correct user will have permission to access the predicted results.

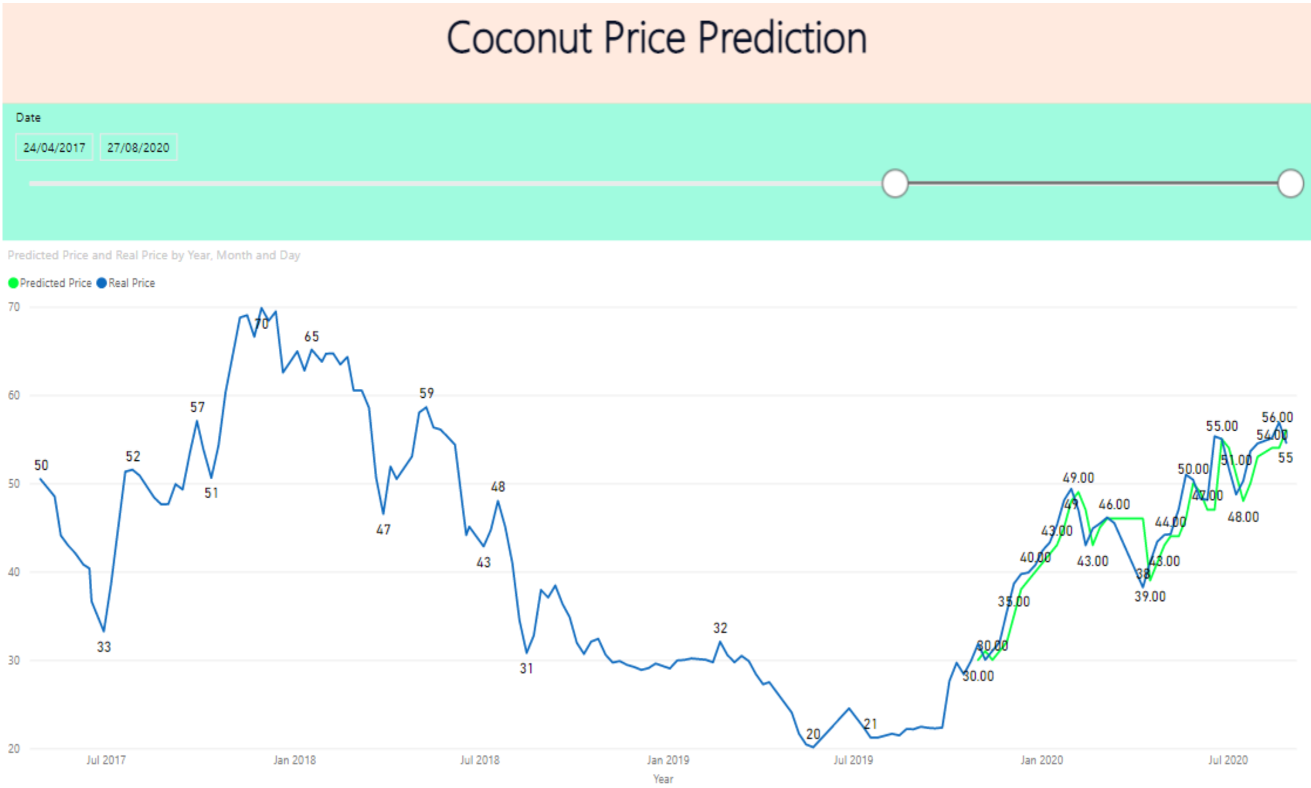

Figure 4.27: Power BI Coconut Price Prediction Model

Figure 4.27 Shows the Coconut Price Prediction module. There is a Date filter on the top of the screen and the line chart gives the real coconut price data and predicted price data. Blue line shows the real data and orange line shows the predicted data. From the line chart the users can visually get an idea how model works. The User can check the price for a week or a month or to a year. The user can access to this web portal, and they can know the predicted price easily. They can do their Coconut Stock planning by referencing this module.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

This study was focused to predict the Coconut price of Sri Lanka using machine learning approach. Moving averages, linear regression, K-Nearest Neighbors, ARIMA, and Prophet are some more strategies for predicting Coconut prices. According to the literature reviews of past researchers in similar domain. Univariate Time series technologies was studied and realized that the Deep Learning algorithms have a considerable influence on modern technology, particularly in the development of distinct time series-based prediction models, as shown in this study. Deep Learning have the highest level of accuracy when it comes to predicting the price of Various Predictions. By studying the past research works it was observed that when compared to other models like regression analysis, Recurrent Neural Networks, Long Short-term Memory which using Deep Learning Algorithms has become more accurate because of it can keep more memory thanks to the architecture that address the problem like long term dependency. Vanilla LSTM, Stacked LSTM, and Bidirectional LSTM are among the Deep Learning models that can be utilized for Coconut price prediction with suitable parameter adjustments. Adjusting these parameters is critical when developing any form of prediction model, as the accuracy of the prediction is heavily reliant on them. As a result, the Vanilla LSTM, Stacked LSTM, and Bidirectional LSTM models all require this parameter adjustment. When the same parameters are used in both models with the 150 epochs the Bidirectional LSTM model produces lower MSE value 0.00157 than the Other LSTM Variants models which Vanilla and Stacked Variant produces 0.00176 and 0.00209.

Individuals and businesses of Coconut consumers can utilize the proposed Bidirectional LSTM prediction model to forecast the Coconut price in Sri Lanka. This can provide significant financial benefits to Coconut consumers. Consumers can plan the production or usage according to the predicted Coconut price.

5.2 Future Works

Although univariate time series data was employed in this study, multivariate data can be used to estimate the price of coconuts. Rainfall data, Temperature, Export Demand, Cultivation Land Size and Coconut Yield data are also significant for the price of coconuts. Since the research can continue to use multivariate prediction in the future and hoping to generate more accurate results using the multivariate prediction. Multivariate predictions also can be done using LSTM models with the Same Steps followed in this study.

The result of the study is published in the web site using power bi tool in the future it can be more developed by connecting the real data source and schedule the model predicting task automatically by scheduling the power bi service. Then the user can enjoy with the UpToDate results.

APPENDICES

Development Tools and Technology

Tensorflow and Keras library has used as the development framework for Long Short-term Memory model. In the field of data science, frameworks are quite important. Frameworks are a set of packages and libraries that make it easier to construct a specific type of application by simplifying the overall programming experience. When it comes to Deep Learning, Keras and TensorFlow are two of the most prominent frameworks.

TensorFlow

TensorFlow is an open-source machine learning platform that runs from start to finish. It is a vast and adaptable ecosystem of tools, libraries, and other resources that provide high-level APIs for workflows. The framework provides different levels of concepts from which to pick while building and deploying machine learning models. TensorFlow allows you to construct and train models at different levels of abstraction. TensorFlow makes it simple to train and deploy models, regardless of the language or platform used. TensorFlow provides flexibility and control for the development of complicated topologies with features like the Keras Functional API and Model Subclassing API.

Keras

Keras is a high-level neural network library that uses TensorFlow, CNTK, and Theano as its foundation. Keras is a deep learning framework that allows for quick prototyping and runs on both CPU and GPU. This framework is developed in Python code, which makes it simple to debug and extend. Keras features a straightforward, consistent interface that is geared for typical use scenarios and gives clear and actionable feedback in the event of a user error. Keras models are created by putting together adjustable building components with few constraints. Keras makes it simple to create bespoke building blocks for new ideas and research. Keras delivers uniform and straightforward APIs that reduce the amount of user activities required for common use cases while still providing clear and responsive feedback in the event of a user error.

Pandas

Pandas libraries are used in the Python programming language, It helps user to used inbuilt functions. pandas use as a data manipulation and analysis software suite. It mostly consists of data structures and methods for manipulating numerical tables and time series.

Scikit-Learn

Scikit-learn, sometimes known as sklearn, is a Python machine learning library that is free to use. It is meant to interact with the Python numerical and scientific libraries NumPy and SciPy, and features support vector machines, random forests, gradient boosting, k-means, and DBSCAN, among other classification, regression, and clustering techniques.

Numpy

NumPy is a Python library that adds support for multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to work with them.

Reference List

Alibuhtto, M.C., 2013. Modelling Fresh Coconuts Export using Time Series Approach. Sri Lanka 5.

Aras, S., Deveci Kocakoç, İ., Polat, C., 2017. COMPARATIVE STUDY ON RETAIL SALES FORECASTING BETWEEN SINGLE AND COMBINATION METHODS. Journal of Business Economics and Management 18, 803–832. https://doi.org/10.3846/16111699.2017.1367324

Ayodele A, A., Charles K, A., 2012. Stock Price Prediction using Neural Network with Hybridized.pdf.

CDB - Coconut Products [WWW Document], n.d. URL https://www.coconutboard.gov.in/CoconutProducts.aspx (accessed 08.27.20).

Coconut, 2021. . Wikipedia.

coconut | Description, Uses, & Facts [WWW Document], n.d. . Encyclopedia Britannica. URL https://www.britannica.com/plant/coconut (accessed 8.22.20).

Etuk, E.H., Mazi, Y.D., Aboko, I.S., 2017. A Simulating Model For Daily Uganda Shilling-Nigerian Naira Exchange Rates 1, 6.

Gluesenkamp, D., 2019. Data science career reflection based on CRISP-DM process model [WWW Document]. Medium. URL https://towardsdatascience.com/data-science-career-reflection-based-on-crisp-dm-process-model-aedd8542b019 (accessed 3.2.21).

Haider Khan, Z., Sharmin Alin, T., Hussain, A., 2011. Price Prediction of Share Market Using Artificial Neural Network "ANN." IJCA 22, 42–47. https://doi.org/10.5120/2552-3497

Hernández, J., López, D., Vera, N., 2018. Primary user characterization for cognitive radio wireless networks using long short-term memory. International Journal of Distributed Sensor Networks 14, 155014771881182. https://doi.org/10.1177/1550147718811828

Hossain, Md.M., Rahman, Md.A., Islam, Md.Z., Majumder, A.K., 2016. Forecasting of Humidity of Some Selected Stations from the Northern Part of Bangladesh: An Application of SARIMA Model. American Journal of Environmental Sciences 12, 182–192. https://doi.org/10.3844/ajessp.2016.182.192

Kajeswari, S., Suthendran, K., 2019. Developing an Agricultural Product Price Prediction Model using HADT Algorithm. IJEAT 9, 569–575. https://doi.org/10.35940/ijeat.A1126.1291S419

Listado de hortalizas [WWW Document], n.d. URL http://www.frutas-hortalizas.com/Hortalizas/ (accessed 9.22.20).

Lwesya, F., Kibambila, V., 2017. A Comparative Analysis of the Application of Seasonal ARIMA and Exponential Smoothing methods in short run Forecasting Tourist Arrivals in Tanzania. European Journal of Business and Management 14.

Mahmud, I., Bari, S., Hussain, M.M., 2015. Year-Long Monthly Rainfall Forecasting for a Coastal Environment of Bangladesh 6.

MLMCE, India, Krishnan, A., 2019. ELECTRICITY BILL PRICE FORECASTING WITH ARIMA MODEL USING THE CONCEPT OF MACHINE LEARNING. IJISCS 8, 139–142. https://doi.org/10.30534/ijiscs/2019/33822019

Peiris, T.S.G., Hansen, J.W., Zubair, L., 2007. Use of seasonal climate information to predict coconut production in Sri Lanka. Int. J. Climatol. 28, 103–110. https://doi.org/10.1002/joc.1517

Rachana, P., Rashmi, G., Shivani, D., Shruthi, N., 2019. 20-CROP PRICE FORECASTING SYSTEM USING SUPERVISED MACHINE.pdf.

Ranasinghe, C., Silva, L., Premasiri, R., 2015. Major determinants of fruit set and yield fluctuation in coconut (Cocos nucifera L.). J. Natn. Sci. Foundation Sri Lanka 43, 253. https://doi.org/10.4038/jnsfsr.v43i3.7955

Rani, S., Chohan, K.E., 2018. PREDICTING THE CHANGE IN PRICE OF COCONUT OIL WITH RESPECT OF TIME 9, 10.

Sabu, K.M., Kumar, T.K.M., 2020. Predictive analytics in Agriculture: Forecasting prices of Arecanuts in Kerala. Procedia Computer Science 171, 699–708. https://doi.org/10.1016/j.procs.2020.04.076

Sajal, A., Rahman, M., 2012. A Stochastic Modelling Technique for Predicting Groundwater Table Fluctuations with Time Series Analysis. IJASER 1. https://doi.org/10.6088/ijaser.0020101024

Samarasinghe, C.R.K., Meegahakumbura, M.K., Dissanayaka, H.D.M.A.C., Kumarathunge, D., Perera, L., 2018. Variation in yield and yield components of different coconut cultivars in response to within year rainfall and temperature variation. Scientia Horticulturae 238, 51–57. https://doi.org/10.1016/j.scienta.2018.03.058

Shen, J., Shafiq, M.O., 2020. Short-term stock market price trend prediction using a comprehensive deep learning system. J Big Data 7, 66. https://doi.org/10.1186/s40537-020-00333-6

Sri Lanka Export Development Board - Sri Lanka Business Portal [WWW Document], n.d. URL https://www.srilankabusiness.com/ (accessed 2.22.21).

Uddin, N., Rahaman, T., Rahman, M.S., 2011. STATISTICAL ANALYSIS AND TREND ON MAXIMUM TEMPERATURE OF BOGRA STATION, BANGLADESH 9.

Udumulla, N.M., 2018. Forecasting Export Prices of Sri Lankan Coconut Products Using Multivariate Time series. 6.

Understanding LSTM Networks -- colah's blog [WWW Document], n.d. URL https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 09.22.20).

Villaren M. Vibas, Raqueño, A.R., 2019. A Mathematical Model for Estimating Retail Price Movements of Basic Fruit and Vegetable Commodities Using Time Series Analysis. https://doi.org/10.5281/ZENODO.3333529

Dr. Rushan Abeygunawardana

to me

6:47 AM (6 hours ago)

Dear Coordinator/MBA, UCSC

This is to certify that the thesis (Titled, Coconut Price Prediction Using Supervised Machine Learning Approach (LSTM)) is based on the work of Mr. G.A.S.M. Padmasiri (Index no.:18880242) under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Thanks

Rushan Abeygunawardana

==

Dr. Rushan Abeygunawardana (Ph.D)

Senior Lecturer (Grade I),
Department of Statistics,
Faculty of Science,
University of Colombo,
Sri Lanka.

==

Supervisor's Comments	This is to certify that this thesis is based on the work of Mr. G.A.S.M. Padmasiri under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.			
Supervisor Recommendation	√	Recommend to submit		Do not recommend to submit
Name	Dr. Rushan Abeygunawardana		**Signature**	
			Date	**14.09.2021**

***Main Supervisor must be a UCSC senior academic staff member**