"COMPARISON OF DEEP LEARNING ARTIFICIAL INTELLIGENCE AND STATISTICAL MODELS FOR PRICE FORECASTING OF BANANA IN GUJARAT"

A THESIS SUBMITTED TO THE
ANAND AGRICULTURAL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

Master of Science

(AGRICULTURE)

IN

AGRICULTURAL STATISTICS

BY

VINIYA GOSWAMI
B. Sc. Agriculture
(Registration No: 2010119102)

B. A. COLLEGE OF AGRICULTURE
ANAND AGRICULTURAL UNIVERSITY
ANAND-388 110

2021

"COMPARISON OF DEEP LEARNING ARTIFICIAL INTELLIGENCE AND STATISTICAL MODELS FOR PRICE FORECASTING OF BANANA IN GUJARAT"

BY

VINIYA GOSWAMI

B. Sc. Agriculture

(Registration No: 2010119102)

कृणवन्तो राष्ट्रं कृषि संपन्नम्

B. A. COLLEGE OF AGRICULTURE

ANAND AGRICULTURAL UNIVERSITY

ANAND-388 110

2021

VINIYA GOSWAMI M. Sc. (Agri.) AGRICULTURAL STATISTICS 2021

COMPARISON OF DEEP LEARNING ARTIFICIAL INTELLIGENCE AND STATISTICAL MODELS FOR PRICE FORECASTING OF BANANA IN GUJARAT

Name of the student
Viniya Goswami

Major guide
Prity Kumari

DEPARTMENT OF AGRICULTURAL STATISTICS
B. A. COLLEGE OF AGRICULTURE
ANAND AGRICULTURAL UNIVERSITY
ANAND- 388110

ABSTRACT

Price as well as other forecast viz., production, productivity, yield, weather data, etc. well in advance play a decisive role for governmental agencies in formulating policies regarding export-import, food grain procurement and distribution, price policies and for countering storage and other marketing issues and is also beneficial for farmers as it helps in pre-planning the activities and avert unwanted situations.

Horticulture is a vital and a significant constituent in the Indian economy. It represents a substantial component of the total agricultural production of our country. The scenario of horticultural crops in India has become very encouraging during the last decade. During 2020, the production of horticulture crops in India was 320.77 million tonnes from an area of 26.46 million hectares. It is further expected to grow in financial year 2021.

In the present study titled "Comparison of deep learning artificial intelligence and statistical models for price forecasting of banana in Gujarat", evaluation of price forecasts for banana has been undertaken for Rajpipla market, Narmada as it accounts for more than 75% of total arrival in the market of whole Gujarat. The secondary data on prices of banana for Gujarat has been taken from Agricultural Marketing website http://agmarknet.nic.in/ from January 2009 to December 2019, with an aim to forecast for year 2020 and to identify the most accurate price forecasting technique out of ARIMA, SARIMA, ARCH/GARCH, ANN and RNN.

ARIMA (5, 1, 0) model was fitted for prediction, but could not perform well for forecasting for the year 2020. Similarly, SARIMA (0, 1, 2) x (0, 1, 2, 4) model was fitted out of various models tried, but performed poorly while forecasting. AR (1,0) + EGARCH (0,1) model was fitted which performed better than the above two models, but still could not provide a satisfactory forecast. ANN with 2 hidden layers and 10 neurons in each was created for price forecasting and was able to provide a satisfactory forecast, but was outperformed by RNN which provided the most satisfactory result out of all.

Further to assess the accuracy, Root Mean Square Error (RMSE) and Mean Absolute Per cent Error (MAPE) were calculated, in which RNN performed best with RMSE - 84.60 and MAPE - 9.58 and SARIMA (0, 1, 2) x (0, 1, 2, 4) performed poorly with RMSE and MAPE 208.84 and 74.13 respectively.

In conclusion, after the comparison of various models attempted in this study to forecast prices of Banana for Rajpipla market, Narmada, Gujarat for the year 2020, the Recurrent Neural Network (RNN) on the basis of RMSE and MAPE values performed better as compared to all the other models studied. This model not only provides better forecasting accuracy but also generates scope for using such techniques in forecasting variables such as prices of agricultural commodities which has high fluctuations owing to various factors.

Dr. Prity Kumari
Assistant Professor & Head,
Department of Basic Science,
College of Horticulture,
Anand Agricultural University
Anand 388110
Gujarat, India

CERTIFICATE

This is to certify that the thesis entitled **"Comparison of deep learning artificial intelligence and statistical models for price forecasting of banana in Gujarat"** submitted by **Viniya Goswami (Reg. No. 2010119102)** in partial fulfilment of the requirements for the award of the degree of **MASTER OF SCIENCE (AGRICULTURE)** in the subject of **AGRICULTURAL STATISTICS** of **B. A. College of Agriculture, Anand Agricultural University**, **Anand** is a record of bonafide research work carried out by her under my personal guidance and supervision and the thesis has not previously formed the basis for award of any degree, diploma or other similar title.

Place: Anand **(Prity Kumari)**

Date: **/** /2021 Major Guide

DECLARATION

This is to declare that the whole of the research work reported herein for the partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE (AGRICULTURE)** in **AGRICULTURAL STATISTICS** by the undersigned is the result of investigation done by me under the guidance and supervision of **Dr. Prity Kumari**, Assistant Professor & Head, Department of Basic Science, College of Horticulture, Anand Agricultural University, Anand and no part of the work has been submitted for any other degree so far.

Place: Anand **(Viniya Goswami)**

Date: / /2021

COUNTER SIGNED
BY

(PRITY KUMARI)
Assistant Professor & Head
Department of Basic Science
College of Horticulture
Anand Agricultural University
Anand - 388110
Gujarat

ACKNOWLEDGEMENT

I am beholden to my highly benevolent and merciful god whose grace always bestows upon me. The very idea of this work makes me amused in thanking to all those who helped me in achieving this milestone of my academic pursuit. Although thanks convey modicum of deep sense of gratitude from the core of my heart yet there is no better way to express it.

I feel immense pleasure and deep sense of gratitude, I take this privilege to express my indebtness to my major advisor ***Dr. Prity Kumari, Assistant professor and Head****, Department of Basic Science, College of Horticulture, AAU, Anand for her invaluable guidance and constant encouragement with constructive suggestions throughout the course of this investigation as well as the preparation of this manuscript.*

My grateful thanks go to the members of my advisory committee, ***Dr. B. L. Dudhat*** *(Minor adviser)* ***Assistant professor and Head****, Department of Social Science, College of Horticulture, AAU, Anand****, Dr. D. J. Parmar,*** *Associate Professor, B. A. College of Agriculture, AAU, Anand,* ***Dr. Nitin Patel****, Assistant Professor, B. A. College of Agriculture, AAU, Anand.*

I wish to extend my thanks to the facilities provided by ***Dr. K. B. Kathiria****, Hon. Vice-Chancellor and* ***Dr. M. K. Jhala****, Director of Research & Dean, Faculty of P. G. Studies, AAU and* ***Dr. Y. M. Shukla****, Principal and Dean, BACA, AAU, Anand. My respects are due to the* ***Dr. P. R. Vaishnav,*** *Professor (P) and Head (Retd.), Department of agricultural statistics, B. A. College of Agriculture, AAU, Anand and* ***Dr. A. D. Kalola****, Associate Professor and Head, Department of agricultural statistics, B. A. College of Agriculture, Anand Agricultural University, Anand.*

I would like to extend a multitude of thanks to my seniors ***Vishnurekha, Hiral Gundaniya*** *and* ***Raj Popat*** *who have inspired me and helped me in all respects during this study to which I remain grateful forever.*

I also take the opportunity to acknowledge the help rendered and time devoted during my research by my friends ***Shiva, Pavani, Jyotshna, Pooja, Samrin, Disha, Hemangi, Ashish, Sagar, Rahul, Jay*** *and all my classmates, also my childhood friends* ***Neelu Pareek, Harish Giri, Pawan Dangi, Pratibha Yadav, Sanya Kalra*** *for helping me directly or indirectly for their good companionship.*

I acknowledge my gratitude to ***Kartik Choubisa, Jugal Paliwal, Bhavya Dave*** *and* ***Parth Sharma*** *for their sustained help and cooperation during my research work.*

I express my heartiest feelings through words to my beloved parents ***Smt. Neelam Goswami*** *and* ***Shri. Parmeshwar Puri Goswami*** *without whose moral support, sacrifice and blessings my dream would not have been a reality. Love and encouragement showered by my loving Grandparents* ***Mrs. Sushila*** *and* ***Mr. Nathu Giri,*** *and my loving sisters* ***Tanvi*** *and* ***Sakshi*** *and my cousins* ***Himani, Ritika*** *and* ***Siddharth*** *has energized to mount the climax of this exploration. I would also like to thank my uncle* ***Mr. Shravan Puri*** *and* ***Mr. Jitendra Giri*** *for their endless support and incessant help. With all my love and respects, I dedicate this work to them.*

I also want to express my love and gratitude towards ***BTS****, who inspired me throughout these years and motivated me to become a better version of myself, encouraged me to work hard each day and work towards my goal. Also my special thanks to* ***Naruto****, who encouraged me to believe in myself.*

I would like to thank and appreciate myself, for working hard and securing All India Rank during ICAR-JRF exam, and getting into the college I dreamt of. Also incessantly working hard, improvising myself every day, and for continuously moving on.

Place: Anand

Date: /09/2021 ***(Viniya Goswami)***

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

%	Per cent
&	Ampersand
/	Per
@	At the rate of
AAU	Anand Agricultural University
ACF	Autocorrelation Function
ADF test	Augmented Dickey-Fuller test
AI	Artificial Intelligence
AIC	Akaike Information Coefficient
ANN	Artificial Neural Network
AR	Autoregressive
ARCH	Autoregressive Conditional Heteroscedasticity
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
B.A.C.A	Bansilal Amrutlal College of Agriculture
BIC	Baysian Information Coefficient
CPU	Central Processing Unit
EGARCH	Exponential Generalized Autoregressive Conditional Heteroscedasticity
et al.	And others
etc.	Etcetera
FAO	Food & Agriculture Organization
Fig.	Figure
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
GPU	Graphics Processing Unit
IGARCH	Integrated Generalized Autoregressive Conditional Heteroscedasticity
LM test	Lagrange Multiplier test
LSTM	Long Short-Term Memory
MA	Moving Average
MAPE	Mean Absolute Per cent Error
MLP	Multi Layered Perceptron
MSE	Mean Squared Error
NN	Neural Network
PACF	Partial Autocorrelation Function
q	Quintal(s)
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
₹	Rupee(s)
SARIMA	Seasonal Autoregressive Integrated Moving Average
SD	Seasonal Differencing
TF	TensorFlow

CONTENTS

1. INTRODUCTION

Horticulture is a vital and significant constituent in the Indian economy. It represents a substantial component of the total agricultural production of our country. Its importance can be elucidated by its benefits like, high export value, high returns per unit area, effective utilization of wasteland, provision of raw materials for industries, wholesome engagement by a grower/ labourer, production of more food energy per unit area than that of field crops, better utilisation of undulating lands and stabilization of farmer's revenue by generating employment opportunities through processing, floriculture, olericulture, seed production, mushroom cultivation, nursery preparation, post-harvest production etc. (Ravichandra N. G. 2014).

The scenario of horticultural crops in India has become very encouraging. During 2020, the production of horticulture crops in India was 320.77 million tonnes from an area of 26.46 million hectares. It is further expected to grow at the rate of 1.8% in Financial year 2021 with area expected at 27.17 million hectares and production 326.58 million tonnes, according to the first estimates given by the Government of India https://pib.gov.in/PressReleasePage.aspx?PRID=1717447 .

In the present study, evaluation of price forecasts for BANANA will be undertaken. Banana (*Musa* sp.) is one of the healthiest fruits on earth and for this reason is popular worldwide. It's year round availability and affordability makes it the favourite fruit among all classes of people. It also has good export potential. Important cultivars that are grown in India include Robusta, Monthan, Poovan, Dwarf Cavendish, Nendran, Red banana, Basrai, Ardhapuri, Nyali, Safed Velchi Rasthali, Karpurvalli, etc.

Approximately, about 5.6 million hectares of land are dedicated to banana production globally. The banana industry has achieved rapid improvements in productivity, with the average yield increasing from around 14 tonnes per hectare in 1993 to 20 tonnes per hectare in 2017. Bananas are principally produced in Asia, Latin America and Africa. India is the biggest producer and exporter of banana globally with 884 thousand hectares area under cultivation and production of 30.80 million tonnes, contributing around 26% to the global banana basket (FAOSTAT, 2020).

Table 1.1: Global scenario of major Banana producing countries (2019)

Sr. No.	COUNTRY	PRODUCTION (MT)
1	India	30.5
2	China	12
3	Indonesia	7.2
4	Brazil	6.8
5	Ecuador	6.6

(Source: FAOSTAT, 2020)

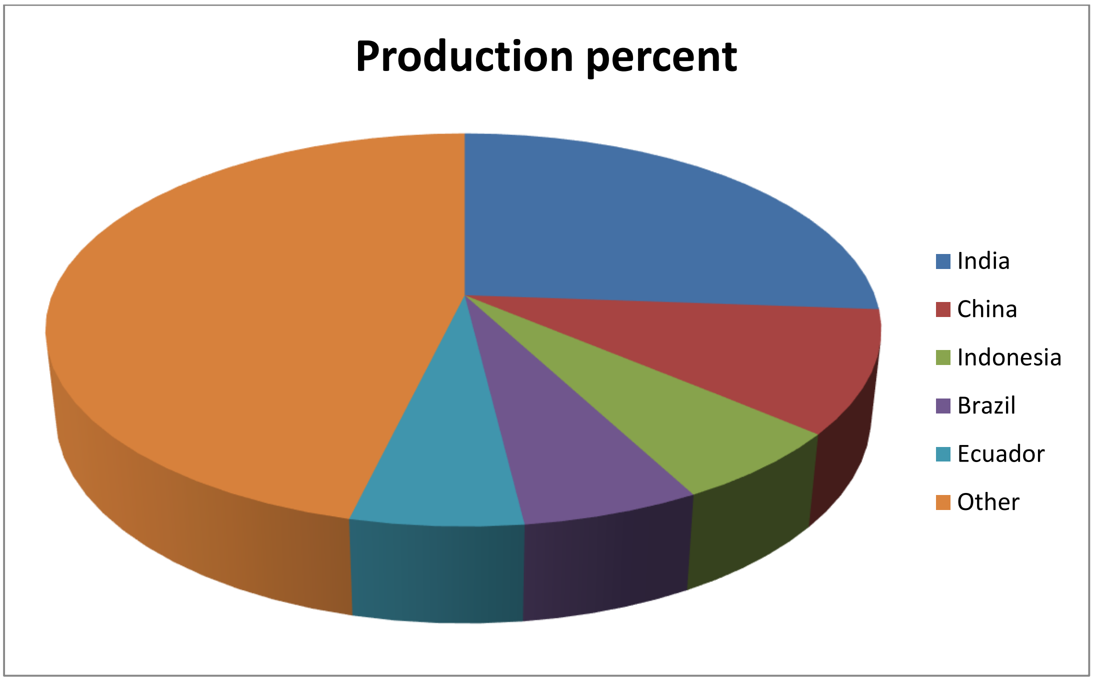

Fig. 1.1: Global banana production (2019)

Table 1.2: Indian scenario of major Banana producing states (2020)

Sr. No.	STATE	PRODUCTION (metric tonnes)
1	Tamil Nadu	5136.2
2	Gujarat	4627.5
3	Maharashtra	3600.0
4	Andhra Pradesh	3242.0
5	Karnataka	2529.6

(Source: National Horticulture Board)

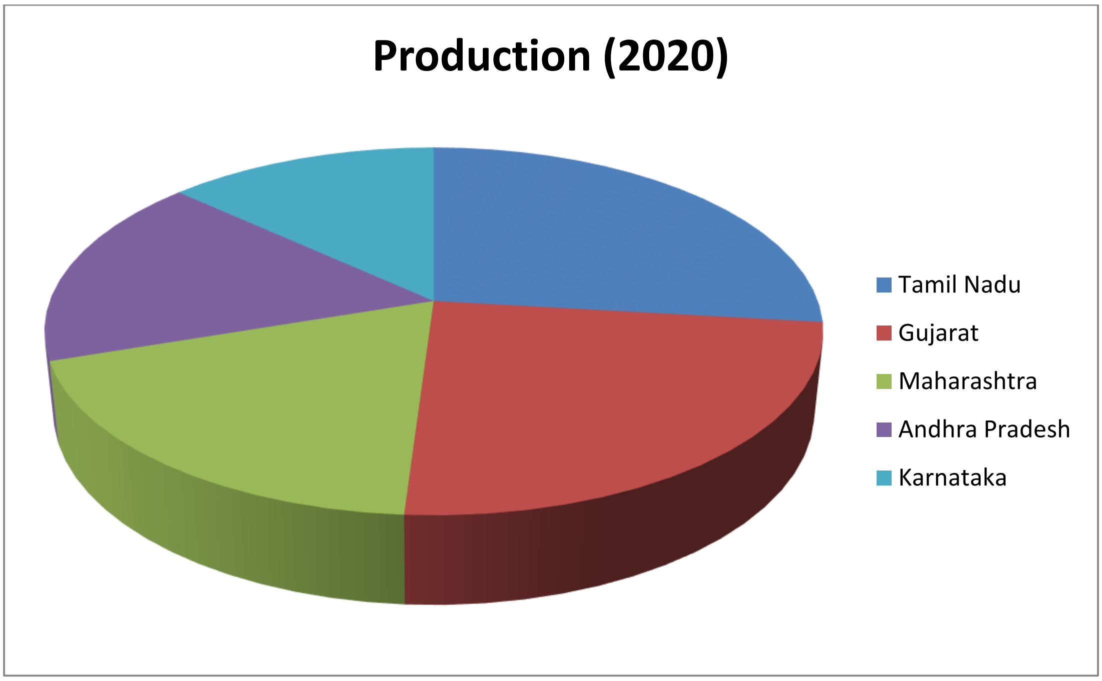

Fig. 1.2: Topmost banana producing states in India (2020)

According to statistics provided by Directorate of Horticulture Gujarat 2020, the area under banana production is 69.537 thousand ha with production 4627.52 metric tonnes https://doh.gujarat.gov.in/Images/directorofhorticulture/pdf/statistics/Area-Production-2019-20.pdf .

As per the market arrival of data, Rajpipla market (Narmada) covers 78% of the total arrival in the market of whole Gujarat and based on this fact Rajpipla market is considered as deciding market for price of banana for the entire state of Gujarat (Agricultural Marketing website, 2020).

Price forecasting is an integral part of commodity trading and price analysis. Quantitative accuracy with small errors, along with turning point forecasting power is important for evaluating forecasting models. Currently, 20–30% of India's banana production is estimated to go waste owing to natural calamities such as droughts, floods, attacks by pest and diseases and post-harvest losses. Also India being the largest producer of bananas, exports only 0.38% of its total production. http://assets.fsnforumhlpe.fao.org.s3-eu-west-1.amazonaws.com/public/files/Food_losses_waste/India%20banana%20case_final.pdf

Therefore, forecasting of price is very helpful in designing a cropping plan and is thus beneficial to farmers as well. It will also aid in improvising the export potential of banana in India.

Naturally forecasting is one of the main objectives of time series analysis having the art of saying what will happen in the future rather than why. One can use a forecasting model based on the experience and external information. Among the most important and widely used time-series models is Auto Regressive Integrated Moving Average (ARIMA). Recently, Artificial Neural Network (ANN) model has attracted much attention as an alternative technique for estimation and forecasting in economics and finances (Zhang *et. al*., 1998).

With this point of view, the present investigation entitled “Comparison of deep learning artificial intelligence and statistical models for price forecasting of banana in Gujarat” is envisaged with the following objectives:

Objectives:

1. To study deep learning artificial intelligence for price forecasting of banana.

2. To study different statistical models for price forecasting of banana.

3. To identify and suggest the most accurate price forecasting technique.

2. REVIEW OF LITERATURE

In the current chapter, a comprehensive review of published literature has been presented so as to understand the notions and approach followed in earlier studies. This would facilitate the researchers to steer the study in the right direction, collect the appropriate data and to draw meaningful results out of it. This will also envision addressing the gap in literature. Keeping in view, the objectives of my study, the reviews have been presented.

Ray (1974) worked on the short run price forecasting and has revealed that most of the model proprietors practiced adjusting the forecasts from their model before they release forecasts. The forecasts from the models can be adjusted by changing the values of constant terms in the equations and by adjusting the values of the exogenous variables used for the forecasts. The adjusted forecasts of the model proprietors were on average more accurate than the non-adjusted forecasts from the models.

Brandt and Bessler (1983) worked on seven forecasting approaches and examined their performances over 24 quarters from 1976 to 1981. These methods included exponential smoothing; an autoregressive integrated moving average process, an econometric model, expert judgment, and a composite forecasting approach. The application gave result which supported previous findings in the forecasting literature and suggested that forecasting methods can provide valuable information to the decision makers.

Kastens *et. al*. (1998) worked on future based price forecasts for agricultural producers and businesses and determined the forecasting accuracy of five competing naive and futures-based localized cash price forecasts. The third week's price each month from 1987-96 was forecasted from several vantage points. Commodities examined included those relevant to Midwest producers: the major grains, slaughter steers, slaughter hogs, several classes of feeder cattle, cull cows, and sows. Relative forecasting accuracy across forecast methods was compared using regression models of forecast error. The traditional forecast method of deferred futures plus historical basis had the greatest accuracy-even for cull cows. Adding complexity to forecasts, such as including regression models to capture nonlinear bases or biases in futures markets, did not improve accuracy.

Yin (1999) conducted timber price forecasts with univariate autoregressive-integrated-moving-average, or ARIMA, models employing the standard Box-Jenkins modelling strategy. They found that most of the selected pine pulpwood and saw timber markets can be evaluated using ARIMA models, and that short-term forecasts, especially those of one-lead forecasts, were fairly accurate.

Haofei *et. al.* (2007) compared the predictive performance of ARIMA, artificial neural network and the linear combination models for forecasting wheat price in Chinese market. Empirical results showed that the combined model can improve the forecasting performance significantly in contrast with its counterparts in terms of the error evaluation measurements. The ANN model was overall the best model, and can be used as an alternative method to model Chinese future food grain price.

Paul and Himadri (2009) worked on forecasting of India's spices export data set, which exhibits a volatile behaviour,. They used Box-Jenkins Autoregressive integrated moving average (ARIMA) approach and Generalized autoregressive conditional heteroscedastic (GARCH) nonlinear time-series model. Lagrange multiplier test was used for testing the presence of Autoregressive conditional heteroscedastic (ARCH) effects. Comparative study of the fitted ARIMA and GARCH models is resulted in superiority of GARCH model over ARIMA approach for the data under consideration.

Li *et. al.* (2010[a]) worked on price forecasting of tomato crop in China . A feed-forward ANN model was developed for short-term price forecasting of tomato and in comparison with time series model ARIMA was used in this study. The results showed that ANN model evidently outperformed the time series model in forecasting the price before one day or one week. A good correlation between the modelled and the real prices was observed from the feed-forward ANN model, with a relative error less than 5.0%.

Li *et. al.* (2010[b]) worked on wholesale price forecasting models of tomatoes, based on random features of daily price fluctuation of agro-product market as well as ADF test and ARCH effect test based on price series data. They employed ARIMA, ARCH and GARCH to establish daily, and applied the models to forecast the tomato price. The result shows that mean absolute percentage error (MAPE) of the three daily price forecasting models is less than 2%, among which the highest accuracy in forecasting is GARCH model.

Jha and Sinha, (2013) worked on agricultural price forecasting using artificial neural network (ANN) model and stated that forecasts of food prices are intended to be useful for farmers, policymakers and agribusiness industries. In this study, the superiority of ANN over linear model methodology had been demonstrated using monthly wholesale price series of soyabean and rapeseed-mustard. The empirical analysis indicated that ANN models were able to capture a significant number of directions of monthly price change as compared to the linear models. The study aimed at developing a user-friendly ANN based decision support system by integrating linear and nonlinear forecasting methodologies.

Kumari *et. al.* (2013) worked on forecasting of productivity and pod damage by *Helicoverpa armigera* using Artificial Neural Network model in Pigeon-pea (*Cajanus cajan*). The feed-forward neural network with supervised learning was proposed to forecast the damage. It has been inferred that Levenberg Marquardt algorithm gave the best performance in the prediction of damage (26.29%) and productivity (1137.40 kg/ha) of long duration pigeon-pea for NEPZ in India for the year 2012-13.

Kumar *et. al.* (2013) used ANN methodology for forewarning *Alternaria blight* and Powdery mildew in mustard for maximum disease severity, crop age at first appearance of disease and crop age at maximum disease severity as response variables and weather indices as predictors for three locations namely Bharatpur, Dholi and Berhampurr. In this study, two types of neural network architectures namely Multilayer perceptron (MLP) and Radial basis function (RBF) were attempted and compared with weather indices based regression model and it has been found that a MLP performed best in terms of mean absolute percentage error (MAPE).

Kumar *et. al.* (2013) studied the trends of temperature of Nagapattinam District, a main shrimp farming area in Tamil Nadu. Surface weather data of the study area were obtained from Indian Meteorological Department, Pune .Time series forecasting model was built for monthly maximum temperatures ,Autoregressive integrated moving average model (ARIMA) (2, 0, 1) (1, 1, 1) s was found to be suitable model with highest R^2 value of 0.94 and lowest Root Mean Square Error.

Model predictions were reassessed with Willmott's index and found to be reasonably good (0.95) for the study area.

Luo *et. al*. (2013) worked on vegetables price forecast. In order to find an effective method, this paper considered the seasonal variations, and used the seasonal auto regressive integrated moving average model (SARIMA) to forecast the cucumber price. The experimental results indicated that the SARIMA (1,0,1) $(1,1,1)_{12}$ fitted the cucumber market prices exactly in the previous months. Its average fitting error was 17%. The forecast data of twelve months in 2011 was in line with the actual trend. Its average error reaches 25%. The SARIMA model was feasible for short-term warning of vegetable price.

Mishra and Singh (2013) worked on forecasting price of groundnut oil in Delhi, for the purpose they used times series namely ARIMA (Autoregressive Integrated Moving Average) methodology given by Box and Jenkins and this approach has been compared with ANN (Artificial Neural Network) methodology. The results showed that forecast by ARIMA model was found to be the best. The mean squared error, root mean square error and mean absolute per cent error were all lower on average for the ARIMA forecast than for the neural network.

Bhardwaj *et. al.* (2014) worked on empirical investigation of ARIMA and GARCH models in agricultural price forecasting and dealt with time series models which are non-structural-mechanical in nature. Augmented Dickey Fuller (ADF) test was used for testing the stationarity of the series. ARCH-LM test was used for testing the volatility. It was found that ARIMA model cannot capture the volatility present in the data set whereas GARCH model has successfully captured the volatility. Root Mean square error (RMSE), Mean absolute error (MAE) and Mean absolute prediction error (MAPE) were computed. It showed that GARCH was a better model than ARIMA for estimating daily price of Gram.

Kumari *et. al.* (2014) developed different Autoregressive Integrated Moving Average (ARIMA) models to forecast the rice yield by using time series data of sixty two years. The performance of these developed models were assessed with the help of different selection measure criteria and the model having minimum value of these criteria considered as the best forecasting model. Based on findings, it has been observed that out of eleven ARIMA models, ARIMA (1, 1, 1) was the best fitted model in predicting efficiently the rice yield as compared to others.

Eni (2015) obtained historical data of average monthly rainfall in Warri Town of Nigeria for the period 2003-2012 for the purpose of model identification and those of 2013 for forecast validation of the identified model. The chosen model was the Seasonal ARIMA (1,1,1) (0,1,1) process which met the criterion of model parsimony with RSS value of 81.098, AIC value of 281.312 and 6 SBC value of 289.330. Model adequacy showed that the seasonal ARIMA model was appropriate for rainfall in the study area. They used the model to forecast rainfall for 2013 and the result compared very well with the observed empirical data for 2013.

Lama *et. al.* (2015) used the autoregressive integrated moving-average (ARIMA) model, generalized autoregressive conditional heteroscedastic (GARCH) model and exponential GARCH (EGARCH) model along with their estimation procedures for modelling and forecasting of three price series, namely domestic and international edible oils price indices and the international cotton price 'Cotlook A' index. The Augmented Dickey-Fuller (ADF) and Philips Peron (PP) tests were used for testing the stationarity of the series. Lagrange multiplier test was applied to detect the presence of autoregressive conditional heteroscedastic (ARCH) effect. A comparative study of the above three models was done in terms of root mean square error (RMSE) and relative mean absolute prediction error (RMAPE). The study has revealed that the EGARCH model outperformed the ARIMA and the GARCH models in forecasting the international cotton price series primarily due to its ability to capture asymmetric volatility pattern.

Mitra and Paul (2017) worked on hybrid time series models for forecasting agricultural commodity prices and stated that agricultural price forecasting has become a promising area of research in recent times. The hybrid methodology namely ARIMA GARCH and ARIMA-ANN had been applied for modelling and forecasting of wholesale potato price in Agra market of India. A comparative assessment had been made in terms of Mean absolute percentage error (MAPE) and Root mean square error (RMSE) among the hybrid and their individual counterpart as far as forecasting is concerned. It was observed that ARIMA-ANN hybrid model outperforms the other combinations and individual counterpart.

Samal (2017) worked on price discovery of cotton futures market in India and evaluated the efficiency of Indian cotton futures prices using Vector Auto Regression (VAR) model and granger causality tests. The Augmented Dicky-Fuller test was initially applied to check stationarity in futures and spot prices. The results showed that both the variables were stationary at level. The VAR model suggested that lag value of futures had more influence on spot price of cotton. The causality test further indicated that futures markets have negligible ability to predict subsequent spot prices for cotton.

Jadhav *et. al.* (2017) demonstrated the utility of price forecasting of farm prices and validating the same for major crops namely Paddy, Ragi and Maize in Karnataka state using the time series data. The results were obtained from the application of univariate ARIMA techniques to produce price forecasts for cereal and precision of the forecasts were evaluated using the standard criteria of MSE, MAPE and Theils U coefficient criteria. The values of MSE, MAPE and Theils U were relatively lower, indicating validity of the forecasted prices of the three crops.

Naveena and Subedar (2017) worked on hybrid time series modelling for forecasting the price of washed coffee (Arabica plantation coffee) in India and used ARIMA, ANN and Hybrid ARIMA-ANN models to analyse the past behaviour of a time series data. Forecasting performance of these models were evaluated and compared using common criteria's such as Root Mean Square Error, Mean Absolute Percentage Error. Hybrid ARIMA-ANN model was best compared to other models, for forecasting of Indian Arabica Plantation coffee price.

Darekar and Reddy (2017[a]) worked on forecasting of common paddy prices in India. Time-series data on monthly average prices of paddy from January 2006 to December 2016 were collected from AGMARK. ARIMA (Box-Jenkins) model was employed to predict the future prices of paddy. The performance of fitted model was examined by computing measures of goodness of fit viz.. AIC, BIC and MAPE. The ARIMA model was the most representative model for the price forecast of paddy in overall India.

Darekar and Reddy (2017[b]) worked on cotton price forecasting in major producing states. The time series data on monthly price of cotton required for the study was collected from the AGMARKNET website from January, 2006 to

December, 2016 to forecast prices for Kharif 2017-18 year harvest months. ARIMA model was employed to predict the future prices of cotton. Model parameters were estimated using the R programming software. The performance of fitted model was examined by computing measures of goodness of fit viz., AIC, SBC and MAPE. Forecasted values showed that market prices of cotton would be in the range of 4600-4900 per quintal in kharif harvesting season, 2017-18.

Darekar and Reddy (2017^c) worked on price forecasting of pulses with an aim of forecasting the pigeon pea prices using the time series data of monthly average prices (January 2006 to December 2016) to predict harvest prices during 2017-18 in major pigeon pea producing states Maharashtra, Uttar Pradesh, Rajasthan, Madhya 8 Pradesh and Gujarat. The study used ARIMA models for price forecast. To test the reliability of model MAPE, AIC, and BIC criterion were used. The model was validated for the year 2016-17. The forecast showed that market prices of pigeon pea, would be in the range of Rs. 4,300-7,600 per quintal during November to January, 2017-18.

Kumarmahto *et. al.* (2019) used Auto Regressive Integrated Moving Average (ARIMA) model under Time series analysis for price forecasting of sunflower seeds in Andhra Pradesh. The data from 1st Jan, 2011 to 31st Dec 2015 was used for training purpose and the data from 1st Jan, 2016 to 31st Dec 2016 for testing purpose. Based on the training data, ARIMA(1, 1, 2) selected as best model. Mean Average Percentage Error (MAPE) for the selected model was calculated as 2.30%.

Weng *et. al.* (2019) used ARIMA model, back propagation (BP) network method, and RNN method to forecast the price of agricultural products (cucumber, tomato, and eggplant) in short term (several days) and long term (several weeks or months). ARIMA gave good performance for average monthly data but not for daily data. Instead, the neural network methods (including BP network and RNN) can predicted well daily, weekly, and monthly trend of price fluctuation.

Sabu and Kumar (2020) worked on monthly prices of arecanut in Kerala using time-series and machine learning models. The models SARIMA, Holt-Winter’s Seasonal method, and LSTM 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.

3. MATERIALS AND METHODS

A dataset that contain a sequence of observations on a single phenomenon observed over a period of time is known as Time Series Data. For example, daily maximum-minimum temperatures, annual crop production, productivity and yield etc. are all datasets for a time series. Time series forecasting is a significant area in which past observations of same variable are utilised and analysed to develop a model that describes the relationship, to extrapolate the series into the future. In literature work, the time series models are commonly categorized into two broad groups namely Linear models and Non-Linear models, which have been used in the present study.

This chapter deals with the details of software used for analysis, database used and the methods applied for price forecasting of banana and checking the accuracy of the fitted models.

3.1 SOFTWARE - Python

The software tool utilised for the analysis work is python and has been discussed below.

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991. Python is an easy-to-use programming language, the standard library of which offers a wide range of facilities. It is also interoperable with a vast selection of other libraries, modules and even entire application development frameworks, which makes it an excellent tool for use in multitude fields of study, specifically Data Science and Machine Learning. Its additional library NumPy supports multi-dimensional arrays and adds high-level mathematical functions useful for data science https://www.numpy.org/ .

3.1.1 TensorFlow

TensorFlow developed by the Google Brain team is an open-source framework especially used for machine learning applications. It is available for different operating systems and mobile computing platforms. Implemented functions and modules are extended by a module tensorflow.contrib which is not included in the main repository. However users can contribute with own code or use functions of this module. TensorFlow can also run on multiple CPUs and GPUs which fastens the training. The platform also provides checkpoints to save and restore TensorFlow (TF) models built with Estimators and a utility called TensorBoard for visualising computation graphs, parameters distributions and other features. https://www.tensorflow.org/.

Python offers a wide variety of features. There are a number of python packages and libraries. The libraries that will be used in the proposed research work are as follows and presented in Table 3.1 (Stancin and Jovic, 2019).

Table 3.1: Libraries used for preparing models

Library	**Application**
Numpy	• Numpy was developed in 2005 by Travis Oliphant. • Widely used package for scientific computing with python. • It has many tools which can be used to integrate C/C++ code.
Matplotlib	• Matplotlib is a Python implementation of the MATLAB-like plots and is written on a low level, with a lot of possibilities for customization. • It produces figures and graphics which can be used in a variety of interactive environments across platforms.
SciPy	• It is a package for mathematics, science and engineering. • It contains modules for optimization, integration, interpolation, special functions, fast Fourier transform, signal and image processing, Ordinary Differential Equation solvers and other tasks common in science and engineering.
Keras	• Open source python library for neural network and other machine learning models.

Pandas	• "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. • It is used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.
Seaborn	• It is a library based on Matplotlib, used for more attractive visualisations of statistical graphs.
Scikit-learn	• It is an open-source software library that provides tools for data analysis. • There are classes for preprocessing and overall ML problems, such as classification, regression, clustering or dimensionality reduction.
StatsModels	• It is a Python module that provides functions and classes for estimation of many different statistical models, as well as for conducting statistical tests, and data exploration.

3.2 DATABASE

In the present study, the secondary data on prices of banana for Gujarat will be taken from Agricultural Marketing website http://agmarknet.gov.in/ for Rajpipla market (Narmada), Gujarat from 2009-2019. The reason for selecting Rajpipla Market is based on the market arrival data and Rajpipla market (Narmada) covers 78% of the total arrival in the market of whole Gujarat (Agriculture Marketing website, 2020).

3.3 METHODOLOGY

The following analytical models were utilized in the present study:

3.3.1 Statistical Models

A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data.

3.3.1.1 Autoregressive Integrated Moving Average (ARIMA) Models

The Box-Jenkins procedure

Box-Jenkins (1970) introduced the following procedure of fitting of mixed Auto Regressive Integrated Moving Average (ARIMA) model to a given data set. The core objective of fitting the ARIMA model is to identify the stochastic process of the time-series and predict the future values precisely. These techniques have also been useful in several situations which involve the building of models for discrete time series and dynamic systems. Auto Regressive (AR) models were first familiarized by Yule in 1926. These were consequently supplemented by Slutzky in 1937, who presented Moving Average (MA) schemes. Wold (1938) combined both AR and MA schemes and showed that ARMA processes to model all stationary time series as long as the appropriate order of p: number of AR terms, and q: number of MA terms stands.

Prior to discussing the ARIMA model developing, few concepts of linear time-series analysis, such as stationarity, seasonality and a brief reference to the most classical common types of time-series forecasting methods are discussed.

3.3.1.1.1 Stationarity and Non-stationarity

Time-series is said to be stationary if its generating process is based on a constant mean and constant variance with its autocorrelation function (ACF) essentially constant through time. Otherwise it is called non-stationary series. A statistical test for stationarity has been presented by Dickey and Fuller (1979). The test is applied for the parameter p in the auxiliary regression.

$$\Delta_1 \, y_t = \, py_{t-1} \, + \, \alpha_1 \, \Delta_1 \, y_{t-1} + \, \varepsilon_t \quad \ldots(1)$$

Where Δ_1 signifies the differencing operator i.e. $\Delta_1 \, y_t = y_t - \, y_{t-1}$ the relevant Null hypothesis is $H_0 : p = 0$ against the alternative hypothesis $H_1 : p < 0$.

Acceptance of null hypothesis of series is stationary. Usually, differencing is applied until the ACF shows an interpretable pattern with only few significant autocorrelations.

3.3.1.1.2 Seasonality

In adjunct to the trend, which has now been provided for, stationary series generally exhibits seasonal behaviour where certain basic pattern tends to be repeating at regular seasonal intervals. The seasonal pattern may more over display constant change over time as well. Just as regular differencing was applied to the overall trending series, seasonal differencing (SD) is applied to seasonal non-stationarity and as well as autoregressive and moving average tools are available with the overall series. They are available for seasonal phenomenon using seasonal autoregressive parameters (SAR) (Shruthi 2015).

3.3.1.1.3 Autocorrelation

The most important tools for studying dependence in the time series data are the sample autocorrelation function. The correlation coefficient between any two random variables X and Y, that measures the strength of linear dependence between X and Y, always take values between -1 and +1. If stationarity is assumed and autocorrelation function p_k for a set of lags K = 1, 2 … is approximated by simply computing the sample correlation coefficient between the pairs, k units apart in time. The correlation coefficient between Y_t and Y_{t-k} is called the lag-k autocorrelation or serial correlation coefficient of Y_t and is denoted by the symbol p_k, which under the assumption of weak stationarity is defined as:

$$p_k = \frac{\sum_{t-k+1}^{T}(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t-1}^{T}(Y_t - \bar{Y})^2} = \frac{\gamma_k}{\gamma_0} \quad ; \text{For } k = 1, 2, \ldots \qquad \ldots(2)$$

Where, γ_k = cov(Y_t, Y_{t-k})

γ_0= variance of time-series.

It ranges from -1 to + 1 Box and Jenkins has suggested that maximum number of useful p_k are roughly N/4 where N is the number of periods upon which information of Y_t is available (Hanumanthaiah, 2018).

3.3.1.1.4 Partial – Autocorrelation Function (PACF)

The correlation coefficient between two random variable Y_t and Y_{t-k} , after removing the effect of the intervention is called PACF at lag K and is denoted by, (Hanumanthaiah, 2018) :

$$\emptyset_{00} = 1 \qquad \emptyset_{11} = p_1$$

$$\emptyset_{kk} = \frac{p_k - \sum_{j=1}^{k-1} \emptyset_{k-1.j} \quad _{k-j}}{1 - \sum_{j=1}^{k-1} \emptyset_{k-1.j}\, p_j}, \text{ k = 2,3….} \qquad \ldots(3)$$

Where, $\emptyset_{k.j} = \emptyset_{k-1.j} - \emptyset_{k.k}\, \emptyset_{k-1.k-1}$

3.3.1.1.5 White noise

An essential case of stationary process is called white noise. For a white noise series, all the ACF’s are zero or close to zero (Brockwell and Davis, 2016).

If $\{e_t\}$ is normally distributed with mean zero and variance σ^2 and no autocorrelation, then it is said to be Gaussian white noise.

3.3.1.1.6 ARIMA Model

A generalization of the ARMA model that integrates a wide class of non-stationary time-series is attained by introducing differencing into the model. The simplest example of a non-stationary procedure which reduces to a stationary one after differencing is random walk. Though in contrast to the regression models, it is one of the most traditional methods of non-stationary time-series analysis. The ARIMA model allows to be explicated by its lagged or past values and stochastic error terms. These models are often referred to as “mixed models”. Although this makes the forecasting method more complicated, but the structure may indeed stimulate the series better and produce a more accurate forecast. Pure models imply that the structure consists only of AR or MA parameters and not both. The models developed by this approach are usually called ARIMA models because they use a combination of autoregressive (AR), integration (I) - referring to the reverse process of differencing to produce the forecast, and moving average (MA) operations. An ARIMA model is usually stated as ARIMA (p, d, q). An autoregressive integrated moving average is expressed in the form (Hanumanthaiah, 2018):

If $y_t = \nabla^d y_t = (1-B)^d y_t$ then …(4)

$$y_t = \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \ldots + \emptyset_p w_{t-p} + \varepsilon_t - \emptyset_1 \varepsilon_{t-1} - \emptyset_2 \varepsilon_{t-2} \ldots - \emptyset_p \varepsilon_{t-p} \quad ..(5)$$

If {y_t} follows the ARIMA (p, q) model and { y_t } is an ARIMA (p, d, q) process. For practical purposes, we usually take d = 1 or 2 at most. Above equation is also written as:

$$\emptyset(B) y_t = \theta_0 + \theta(B)\varepsilon_t \quad \ldots(6)$$

Where $\emptyset(B)$ is a stationary aggressive operator, $\theta(B)$ is a stationary moving average operator, and ε_t is a white noise and θ_0 is a constant.

3.3.1.1.7 Box-Jenkins forecasting model is carried out in three stages:

3.3.1.1.7.1 Identification

The primary step, in the modelling process is to check for the stationarity of the series, because the estimation procedures are available only for stationary series. If the original series is non-stationary, then first of all, it should be made stationary.

The next step in the identification procedure is to find the initial values for the orders of seasonal and non-seasonal parameters, p, q, and P, Q. They could be obtained by looking for significant autocorrelation and partial autocorrelation coefficients. For example, if second order autocorrelation coefficient is significant, then an AR(2), or MA(2) or ARMA(2) model can be tried to start with.

3.3.1.1.7.2 Estimation of parameters

At the estimation stage, one or more models are tentatively chosen that seem to provide statistically adequate representations of the available data. Then we attempt to obtain precise estimates of parameters of model by the method of least squares due to residuals as advocated by Box-Jenkins. At the estimation phase, stationarity and invertibility are checked for the coefficient obtained at the same time checking is also done in order to know, whether the model fit the data satisfactorily or not?

To check the closeness of fit, Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) were calculated.

3.3.1.1.7.3 Diagnostic checking

The best model is obtained by following diagnostics.

3.3.1.1.7.3.1 Akaike Information Criteria (AIC) / Bayesian Information Criteria (BIC)

$$AIC = (-2LogL + 2(p + q + P + Q + 1)) \quad \ldots(7)$$

$$AIC = (-2LogL + 2m) \quad \ldots(8)$$

Where m = p + q + P + Q + 1 and L is the likelihood function. Since -2LogL is approximately equal to $\{n(1 + \log 2\pi) + n\log\sigma^2\}$ where σ^2 is the model MSE. Thus AIC can be written as AIC = $\{n\ (1 + \log 2\pi) + n\log\sigma^2 + 2m\}$ and because the first term is a constant, it is usually omitted while comparing between the models.

3.3.1.1.7.3.2 Plot Residual ACF

Once the appropriate ARIMA model has been fitted, one can examine the goodness of fit by means of plotting the ACF of residuals of the fitted model. If most of the sample autocorrelation coefficients of the residuals are within the limits $\pm$1.96/N where, N is the number of observations upon which the model is based then the residuals are the white noises indicating that the model is a good fit (Shruthi, 2015).

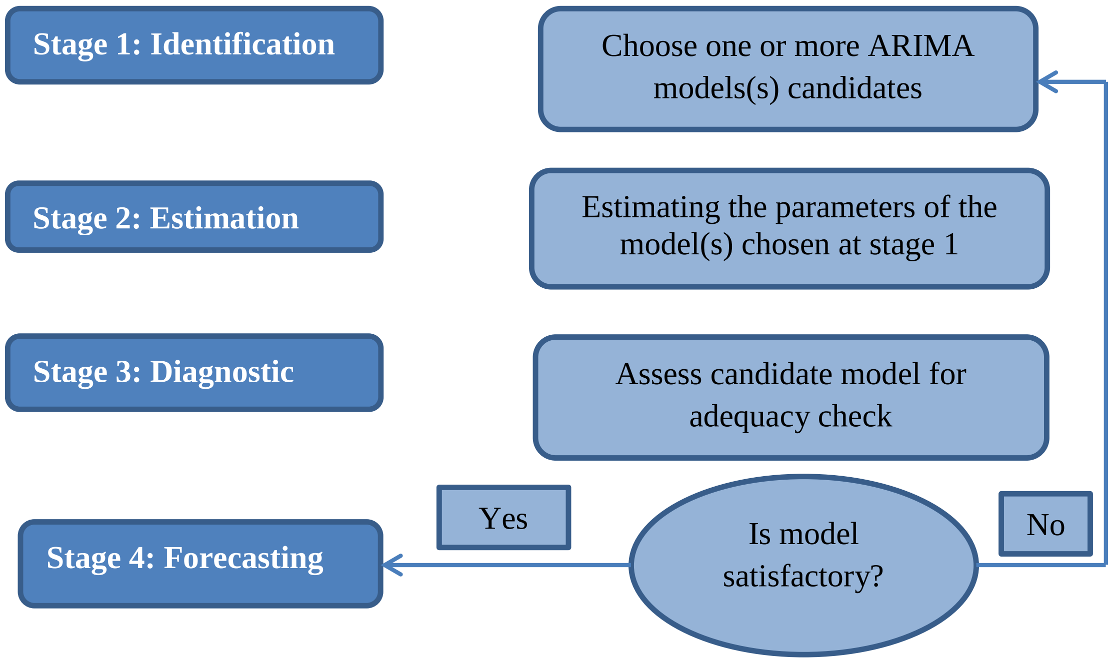

Fig. 3.1: Schematic representation of Box-Jenkins methodology for time series forecasting

3.3.1.2 Seasonal Auto Regressive Integrated Moving Average (SARIMA) Model

The ARIMA model belongs to non-seasonal and non-stationary data. Box-Jenkins generalized this model to deal with seasonality as SARIMA. In this model seasonal differencing of appropriate order is used to remove non-stationarity from the series (Shruthi,2015).

SARIMA (p, d, q) * (P, D, Q)^S model in terms of Lag polynomial is given below:

$$\Phi_P(L^S)\varphi_P(L)(1-L)^d(1-L^S)^D\ y_t\ =\ \Theta Q(L^S\ )\theta_q\ (L)\varepsilon_t \quad \ldots(9)$$

$$\text{i.e. } (L^S)(L)z_t = \Theta Q(L^S\ )\theta q\ (L)\varepsilon_t \qquad \ldots(10)$$

Here z_t is the seasonally differenced series.

3.3.1.3 Autoregressive Conditional Heteroscedasticity (ARCH) Model

Most of the statistical tools have been designed to model the conditional mean of a random variable. The tools described in this section differ by modelling the conditional variance, or the volatility of the variable.

Autoregressive Conditional Heteroscedasticity (ARCH) models have been specifically designed so as to model and forecast the conditional variances. The variance of the dependent variable is to be modelled as a function of past or previous values of the dependent variable and independent or exogenous variables. ARCH models were initially introduced by Engle (1982) and generalized as GARCH (Generalized ARCH) by Bollerslev (1986). These models have wide application in time series analysis.

3.3.1.3.1 Generalized Autoregressive Conditional Heteroscedasticity (GARCH) Models

The key feature of GARCH models is that, the conditional variance of the disturbance of the Y_t sequence comprises on an ARMA process. Hence, it is to be expected that the residuals from a fitted ARMA model should display this characteristic pattern.

Let us suppose that the disturbance term of Y_t sequence, e_t has a conditional variance of σ_t^2.Then for GARCH model (p, q) (Bera and Higgins, 1993):

$$\sigma_t^2 = \alpha_\circ + \alpha_1 e_{t-q}^2 + \cdots + \alpha_q e_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \cdots + \beta_p \sigma_{t-p}^2 \quad \ldots(11)$$

Where the inequality restrictions,

$\alpha_\circ > 0$

$\alpha_i \geq 0$ for i = 1,….,q

$\beta_i \geq 0$ for i = 1,….,p

are imposed to ensure that the conditional variance is strictly positive.

3.3.1.3.2 Exponential Generalized Autoregressive Conditional Heteroscedasticity (EGARCH) Models

The exponential GARCH (EGARCH) may generally be specified as:

$$\varepsilon_t = \sigma_t z_t \quad \ldots(12)$$

$$ln\sigma^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j ln\sigma_{t-j}^2 \quad \ldots(13)$$

This model differs from GARCH variance structure because of the log of variance. The following specification also has been used in the financial literature (Ali, G., 2013).

$$\varepsilon_t = \sigma_t z_t$$

$$ln\sigma^2 = \omega + \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \lambda_j \ln\left(\sigma_{t-j}^2\right) + \sum_{i=1}^{p} \gamma_i \left(\frac{|\varepsilon_{t-i}|}{\sigma_{t-i}} - \sqrt{\frac{2}{\pi}}\right) \quad \ldots(14)$$

The fact that the EGARCH model captures that is not contemplated by the GARCH model, which is the empirically observed fact that negative shocks at time t−1have a stronger impact in the variance at time t than positive shocks. This asymmetry is called leverage effect. The effective coefficient associated with a negative shock is γ−α, while the effective coefficient associated with a positive shock is γ+α.

3.3.1.3.3 Integrated Generalized Autoregressive Conditional Heteroscedasticity (IGARCH) Models

GARCH models apply both an autoregressive and moving average structure to the variance, σ^2. The integrated GARCH (IGARCH) is specified as:

$$\varepsilon_t = \sigma_t z_t$$

$$\sigma^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \qquad \ldots(15)$$

The sum of coefficients is restricted to 1. The exogenous variable can be easily reflected in the various specifications of GARCH models just by addition of $x_t\bar{\beta}$ (Ali, G., 2013).

3.3.2 Deep Learning Artificial Intelligence

Deep learning also known as deep structured learning is part of a broader family of machine learning methods based on neural networks with representation learning.

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the learning algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Recurrent nets have shown light on sequential data such as text and speech (LeCun *et. al.*, 2015).

3.3.2.1 Deep Learning over traditional methods

The shift from traditional methods to deep learning was necessitated by a number of factor (Mahapatra, 2018).

- **Data dependencies**

  As the world today is moving towards “Big Data Era”, traditional methods are not as efficient as deep learning algorithms. Thus the need to shift towards deep learning comes into picture.

- **Execution time**

  Usually, a deep learning algorithm takes a long time to train, because there are so many parameters to be considered. Whereas traditional learning models comparatively takes much less time to train. But this is completely reversed on testing time as deep learning takes very less time to run while test time increases on increasing the size of data for traditional methods.

- **Problem Solving approach**

  When solving a problem using traditional methods, it is generally recommended to break the problem down into different parts, solve them individually and combine them to get the result, while in Deep learning, it advocates to solve the problem end-to-end.

- **Feature engineering**

  Feature engineering is a process of putting domain knowledge into the creation of feature extractors to reduce the complexity of the data and make patterns more visible to learning algorithms to work. This process is difficult and expensive in terms of time and expertise. In traditional methods, most of the applied features need to be identified by an expert and then hand-coded as per the domain and data type. Deep learning algorithms try to learn high-level features from data. Therefore, deep learning reduces the task of developing new feature extractor for every problem.

3.3.2.2 Deep Learning utility

1. Deep learning is ideal for predicting outcomes whenever there is a lot of data to learn from.
2. Complex problems and things that would be vastly expensive to solve with human decision making can be easily and effectively solved with it. Eg. – image processing

3.3.2.3 Hyper-parameters

While parameters are the variables that are involved to estimate an optimisation algorithm and are also updated during training, hyper-parameters are not directly estimated from the data. They are external to the model and are set before the learning phase commences. The positioning of hyper-parameters has an enormous impact on how the model learns and what the predictions are going to be (Reimers & Gurevych, 2017).

The hyperparameters that were used in the model are:

• Number of hidden layers and number of hidden neurons

• Activation function

• Learning rate

• Objective function

• Optimiser

• Batch size

• Train ratio

• Gradient clipping

• Number of epochs.

3.3.2.3.1 Description of the hyper-parameters

1. Hidden layer and neurons

It never emerges and hence called hidden layer. All the processes in the training phase are carried out in this layer. The number of layers depends on the architecture to be designed for the available dataset but generally it consists of a hidden layer, preferably it should lie between input layer and output layer (Hsieh, 2009).

2. Activation Function

Activation function transmits information from the input layer to output layer after the passage of activity – a certain threshold. An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network. Various types of activation function are listed in table 3.2 (Sharma S., 2017).

Table 3.2: Activation Function

Activation Function	**Equation**	**Description**
Linear/Identity	$f(s) = s$	• The function is a line or

		linear. Therefore, its output will not be confined between any range. • Range = (-∞, ∞)
ReLU	f(s) = max{0, s}	• Most commonly used activation function in deep learning, but the issue is that all the negative value becomes zero immediately which decreases the ability of the model to fit or train from the data properly. • Range = [0, ∞)
Leaky ReLU	f(s)=max(0.01*s , s).	• It is an attempt to solve the dying ReLU problem. • Range = (-∞, ∞)
Sigmoid	$\mathrm{f(s)} = \frac{1}{1+e^{-s}}$	• Especially used for models where we have to predict the probability as an output • Range = [0, 1]
TanH	$\mathrm{f(s)} = \frac{\sinh(s)}{\cosh(s)}$ $= \frac{\mathrm{e^s - e^{-s}}}{\mathrm{e^s + e^{-s}}}$	• Used for classification between two classes. • Range = [-1 to 1]
Softmax	$\sigma(\bar{Z}_i) = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}}$	• Calculates probabilities distribution of the event over 'n' different events. • Range = [0, 1]

3. Learning rate

Learning rate controls the convergence of the algorithm – the lower it is, the longer the training usually takes. A too high learning rate can cause the gradient to “overshoot” the minimum, or even diverge (Nielsen M. A., 2015).

4. Objective function

Objective function is a function we want to maximise or minimise and it serves as the evaluation of how well the model fits the data. The parameters of the objective function are all the learnable parameters of the model. Having the objective function L, where θ is the set of all the weights and biases in the model, the aim is to find these weights and biases, so that the condition of Equation 16 is met (Hastie et. al., 2009).

$$\arg\frac{min}{\theta}(L(\theta)) \qquad \text{….(16)}$$

The result of an objective function is a scalar value, which in minimising optimisation problems is called a loss or an error and the target is to reduce it. To find if the model learns, it is best to display the loss on each iteration, and overall this value should decrease.

When dealing with objective functions in a supervised learning algorithm, they can be divided into two groups, regression losses and classification losses and should be chosen with respect to the problem as well as to the data. We face a regression problem, having time series as an input. The measures of estimators’ quality for these types of problems are SSE (Sum of Squared Errors) Eq. 17, MSE (Mean Squared Error) Eq. 18 and MAD (Mean Absolute Deviation) Eq. 19.

$$SSE = \sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \qquad \text{…(17)}$$

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \qquad \text{…(18)}$$

$$MAD = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t| \qquad \text{…(19)}$$

Because we want to know how far on average our predicted values are, we prefer MSE and MAD. MSE penalises the outliers more than MAD because of the square, so the objective function in our model will be MSE.

5. Optimiser

To correctly update the parameters, an optimisation algorithm is used. Adaptive Moment Estimation (Adam) has been used in the present study as an optimiser for numerically optimising neural networks. Adam is an optimiser that enhances its predecessors. It changes the learning rate for each parameter, performing larger updates for frequently occurring features and smaller updates for more frequent ones. It also prevents a decay of the learning rate which could cause the model to stop learning. Moreover, Adam stores momentum changes for each parameter separately. The first moment (the mean) m_t and the second moment (the uncentered variance) v_t (initialised as vectors of zeros) of the gradients g_t are calculated and these values are then used to update the parameters.

6. Batch size

Batch size is the number of samples that are fed to the model during one iteration. Batch size can radically affect how quickly the model converges as well as the results. Large batch size can cause the model's ability to generalise being lost. To have the fastest learning, the samples should be shuffled, i.e. randomly chosen. The more dissimilar the examples in one batch are, the faster the model learns (LeCun *et. al.*, 2012).

7. Train ratio

It can be defined as the ratio of training data to the whole dataset. In some cases, the data are split beforehand, but with a time series, usually a whole dataset is obtained. In this situation it depends on the target of building the model and it is defined without using any hyperparameter optimization.

8. Gradient clipping

To provide the numerical stability of training deep neural network models, a method called Gradient Clipping is used. This method rescales the norm of a gradient when it exceeds a threshold. This gives us a new hyperparameter – the threshold ξ (Hochreiter & Schmidhuber, 1997).

9. Number of epochs

In one epoch, the whole training dataset is passed forward and backpropagated through the model only once. Defining how many epochs are the correct amount so that the model fits the data well, but is not overfitted, can be challenging. The model is underfitted when it does not capture the relationship between the independent (x) and dependent (y) variables, with the possible reason being that it is too simple. The opposite is overfitting, when the model is too fitted on the training data that it is unable to generalise to the test data, which leads to making poor predictions. This may be due to a too long training (Goodfellow *et.al.*, 2016).

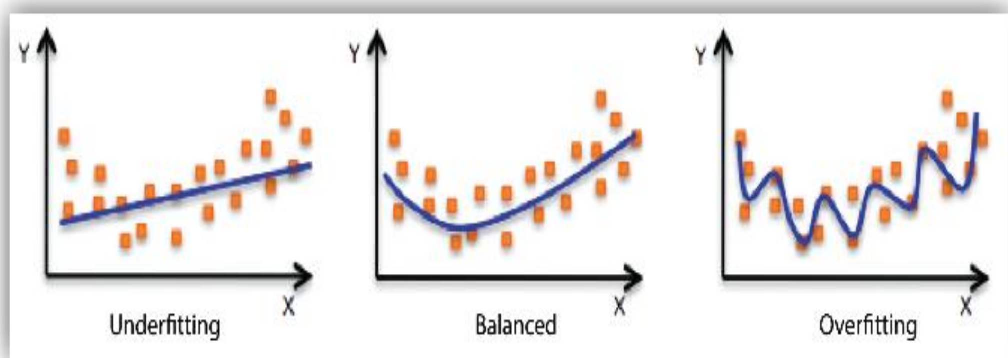

Fig. 3.8: Neural Network curve fitting

3.3.2.4 Neural Networks

Neural networks are a set of algorithms that are modelled loosely after the human brain, and are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labelling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated. Some problems that can be solved with ANN are prediction, classification, optimization and pattern recognition. On the basis of the abilities they have, results of ANN learning can be used to find solutions to a problem (Pal, 2019).

3.3.2.4.1 Structure Of Neural Networks:

Input layer

It contains nodes that each store an unchanged input value in the training phase and can only be changed if new input values are given. The nodes in this layer depend on the number of inputs of a pattern.

Hidden layer

All the processes in the training phase are carried out in this layer. It is located between the input and output of the algorithm, in which the function applies weights to the inputs and directs them through an activation function as the output (Hsieh, 2009).

Output layer

The output layer in an artificial neural network is the last layer of neurons that produces given outputs for the program.

3.3.2.5 Flowchart for NN Modelling

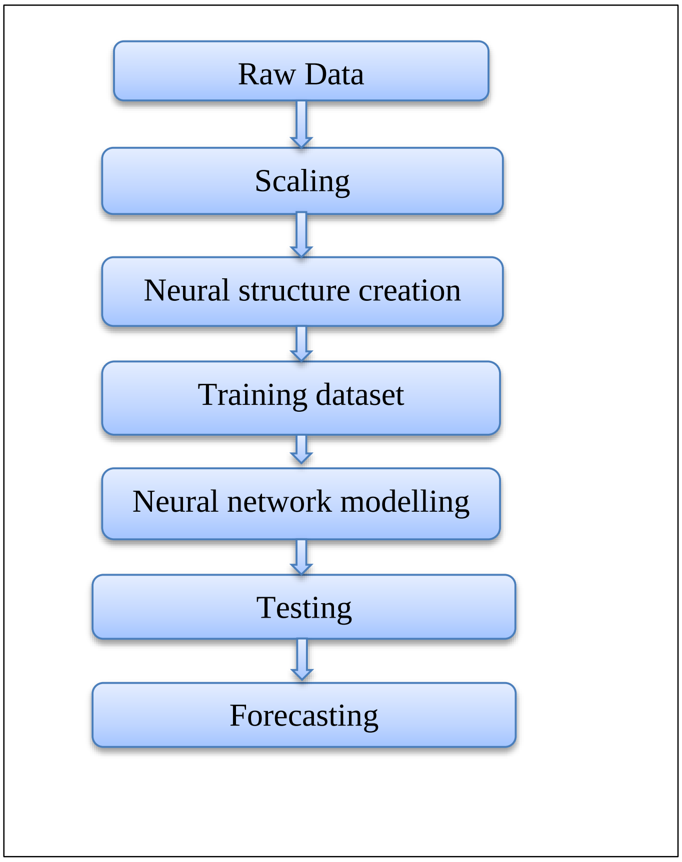

Fig. 3.9: Flowchart of neural network forecasting

- **Scaling**

All variables should be standardized to ensure that they receive equal attention during the training process. Without standardization, input variables measured on different scales will dominate the training procedure to a greater or lesser extent because initial weights within a network are randomised to the same finite range. Data standardization is also important for the efficiency of training algorithms. All variables will be scaled in the range of 0 to 1.

Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \qquad \ldots.(20)$$

- **Training and Testing**

Training set is the dataset that is used to adjust the weights on the neural networks. It can be a set of examples that is used for learning i.e. to fit the parameters {i.e., weights} of the classifier or regressor. Validation set is the dataset that is used to minimize over fitting. If the accuracy over the training data set increases, but the accuracy over validation dataset stay the same or decreases, then it is over fitting the neural network and training should stop. A set of example used to tune the parameters (i.e., architecture, not weights) of classifier or regressor, for example to choose the number of hidden units in a neural network. Testing set is the dataset that is used only for testing the final in order to confirm the actual predictive power of the network (Pal, 2019).

Training stops when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- The maximum amount of time is exceeded
- Performance is minimized to the goal
- The performance gradient falls below minimum gradient value.

Training set can be made easily from the time-series. Certain number of measured values is used as inputs and the value that is to be predicted is used as required output. Input part of the time-series is called window, the output part is the predicted value. By shifting the window over time-series the items of training set are made. It is advised to leave some part of time-series for testing, i.e., not to use this part during learning, but to use it for forecast, how successfully the network learned to predict our unseen data. The training set obtained in this way can be then adjusted for the needs of a particular neural network.

3.3.2.6 Artificial Neural Network (ANNs)

Artificial Neural Network is a biological computer program designed as to stimulate in the way as like human brain processes the information. Artificial neural network gather their information by detecting the pattern of given data and associations between given data and learn from experiences and not from the programming. ANNs are produced from hundreds of units, processing element that is, artificial neurons are associated with coefficients i.e. weights, and comprises of neural structures and organised in layers. The computations of neurons get powerful when the neurons are connected through a network, where each processing elements have weighted inputs, transfer functions and an output. Nature of neural networks is judged by the architecture, learning rule and the transfer functions of its neurons. The weights are adjustable. The parameter and in this sense, the neural networks are parameterized systems. The weighed sum of given inputs are used to activate the neurons. The active signals are passed through the transfer function. This produces a single output of neurons. The transfer functions introduce nonlinearity to the networks. During the ANN training, inter-units connection is optimised until the errors in the prediction are minimised and networks reach the optimised level of accuracy. Once these networks are trained and tested, it will be given as new input information to predict the outputs. ANNs represent the promising modelling techniques, especially for the set of data, those have nonlinear relationships.

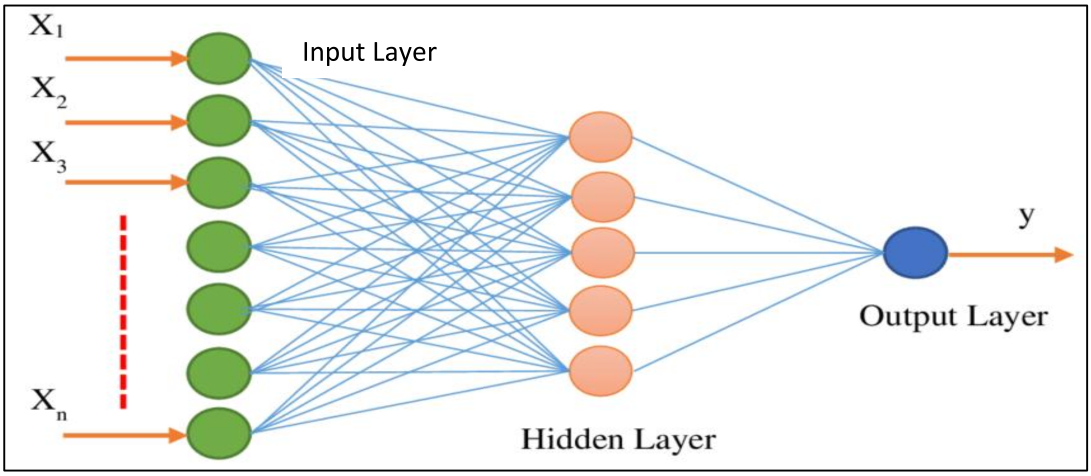

Fig. 3.10: Structure of ANN

Features of ANN (Kumar, 2010)

- ANNs are inherently non-linear.
- ANNs are data-driven and self-adaptive in nature.
- ANNs are universal functional approximators.

The output of the ANN model is computed using the following mathematical expression:

$$y_t = a_\circ + \sum_{j=1}^{q} a_j g\left(\beta_{\circ j} + \sum_{j=1}^{q} \beta_{ij} y_{t-i} n\right) + \varepsilon_t, \quad \ldots(21)$$

Here y_{t-i} (i = 1, 2, ... , p) are the p inputs and y_t is the output.

The integers p, q are the number of input and hidden nodes respectively.

a_j (j = 0,1,2,………q) and β_{ij} (i = 0,1,2,………q) are the connection weights and ε_t is the random shock. $a_\circ$ and $\beta_{\circ j}$ are the bias terms and g(x) is the nonlinear activation function (Pal, 2019).

3.3.2.6.1 Features of Artificial Neural Network

3.3.2.6.1.1 Pattern of connections between the neurons.

3.3.2.6.1.2 Learning algorithm

3.3.2.6.1.1 Pattern of connections between the neurons.

The pattern of connections between the neurons and the propagation of the data is what constitute the topology of the networks. On basis of pattern connections, main neural networks are:

Feed forward neural network

- Single layer Feed forward neural network

It is a simple design of neural networks that allows only unidirectional forward connections among nodes. That is why it is called feed forward neural network or perceptron.

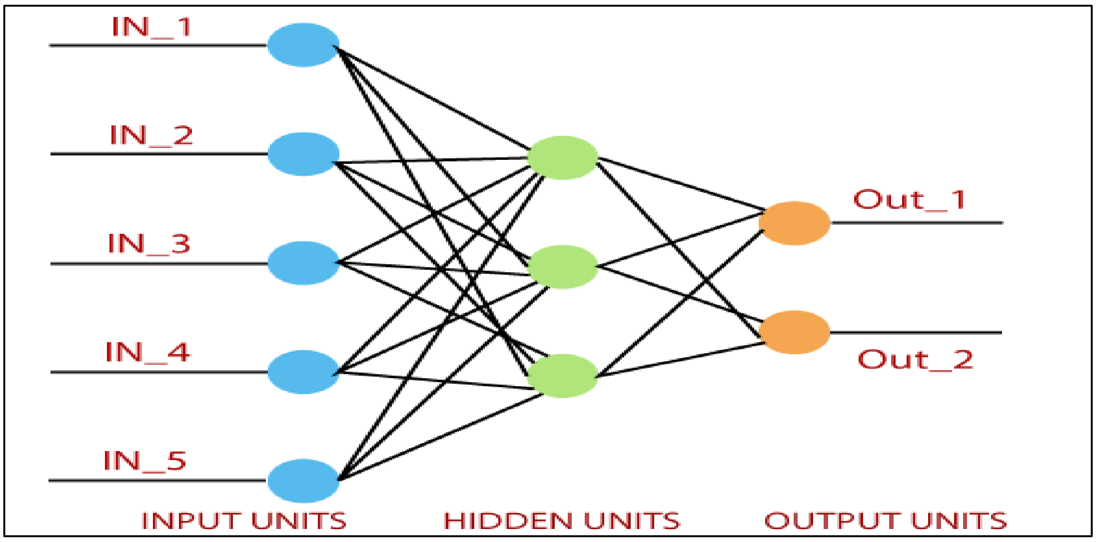

Fig. 3.11: Single-layer feed-forward Neural Network

The simplest type of Feed – Forward Neural Network, consists of only one layer of p nodes associated with a group of n input (Fig. 3.11) and is commonly used in all architectures.

- Multi-layer feed-forward Neural Network

Networks that contain more than one layer of hidden layers, which allow unidirectional forward connections of inputs and outputs, are called multi-layered perceptron or Multi-layer feed forward neural network. A multi-layered perceptron consists of a set of input terminals, an output neural layer, and a number of layers of hidden nodes between the input terminals and the output layer (Fig. 3.12).

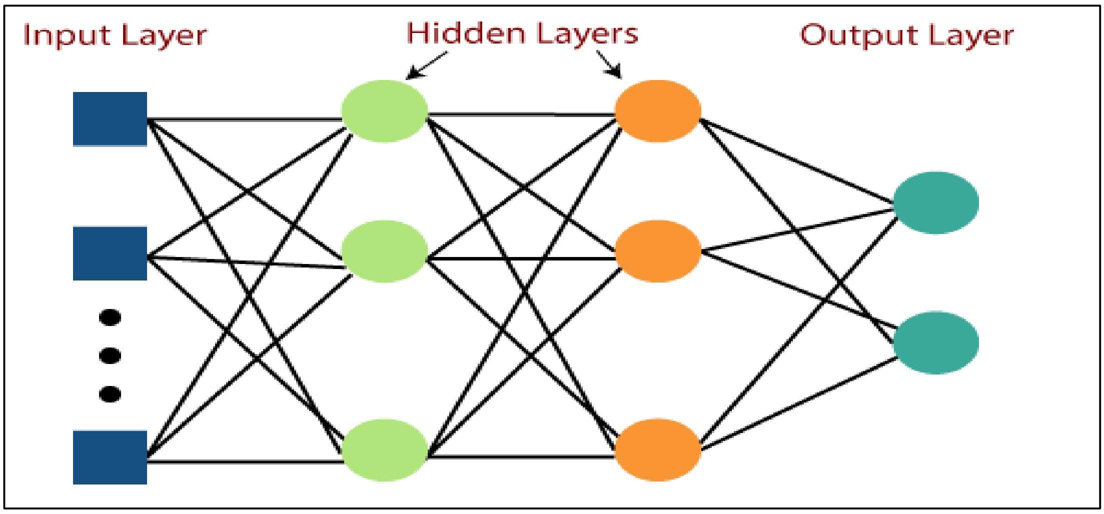

Fig. 3.12: Multi-layer feed-forward Neural Network

In multi-layer feed forward network, information is transmitted from the input layer to output layer, as in the case in the human brain where signals go in one direction. Feed forward networks use any Boolean function and are guaranteed to reach stability provided that the number of hidden neurons is sufficiently large. (McCulloch and Pitts, 1990)

3.3.2.6.1.2 Learning Algorithms

The important aspect of ANN is its learning and training algorithm which is a procedure for updating and determining weights of the connections. Learning algorithm is a procedure of modifying weights and biases of network i.e., method of driving next changes that might be made in ANN, while training algorithm is a procedure whereby network is actually adjusted to do a particular job (Pal, 2019).

There are 3 major learning paradigms; supervised learning, unsupervised learning and reinforcement learning described below:

3.3.2.6.1.2.1 Supervised Learning

Supervised Learning is a technique that sets parameters of an Artificial Neural Networks from training data. The task of the learning ANN, is to set, the value of its parameters for any valid input value after having seen output value. The training data consist of pairs of input and desired output values that are traditionally represented in data vectors.

3.3.2.6.1.2.2 Unsupervised Learning

Unsupervised Learning is a machine learning technique that sets parameters of an ANN based on given data and a cost function which has to be minimised. Unsupervised learning is mostly used in applications that fill within the domain of estimation problems such as statistical modelling, compression, filtering, blind source separation and clustering.

3.3.2.6.1.2.3 Reinforcement Learning

Reinforcement learning is an area of machine learning for taking a suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation.

Main points in reinforcement learning –

- Input: The input should be an initial state from which the model will start.

- Output: There are many possible output as there are variety of solution to a particular problem
- Training: The training is based upon the input, the model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn from its experience.
- The best solution is decided based on the maximum reward.

3.3.2.6.2 Cross Validation

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. It is a validation technique in which data is randomly divided into k section and in each part classification process or regression is done (Salman *et. al.,* 2018).

3.3.2.7 Recurrent Neural Networks (RNNs)

In working with traditional Neural Networks, the assumption has always been that all inputs (and outputs) are independent of each other. This however was not true for many tasks. Recurrent Neural Networks works in a similar way, so that the recurrent part of RNNs is as a result of its ability to perform the same task for every element of a sequence with the output being dependent on the previous computations. In theory, RNNs ways of operating can be illustrated as RNNs having memory for storing unbounded history on previously processed elements. So at every point in time, this stored history is used to predict the next output from the process (Pant, N., 2017).

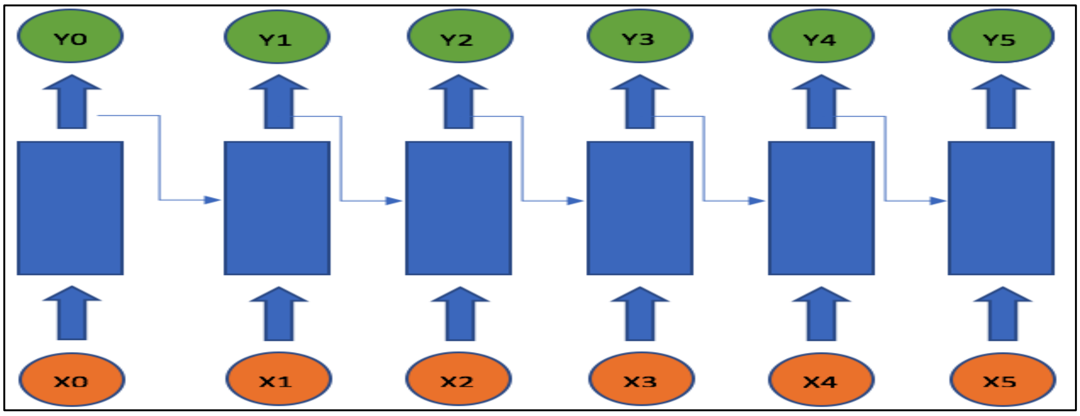

Fig. 3.13: Recurrent Neural Network Model

$$Y_t = \tanh(wY_{t-1} + ux_t) \qquad \ldots(22)$$

Where, Y_t = final output

Y_{t-1}= previous output

w= weight multiplying to Y_{t-1}

x_t= current input

u = weight multiplying to x_t

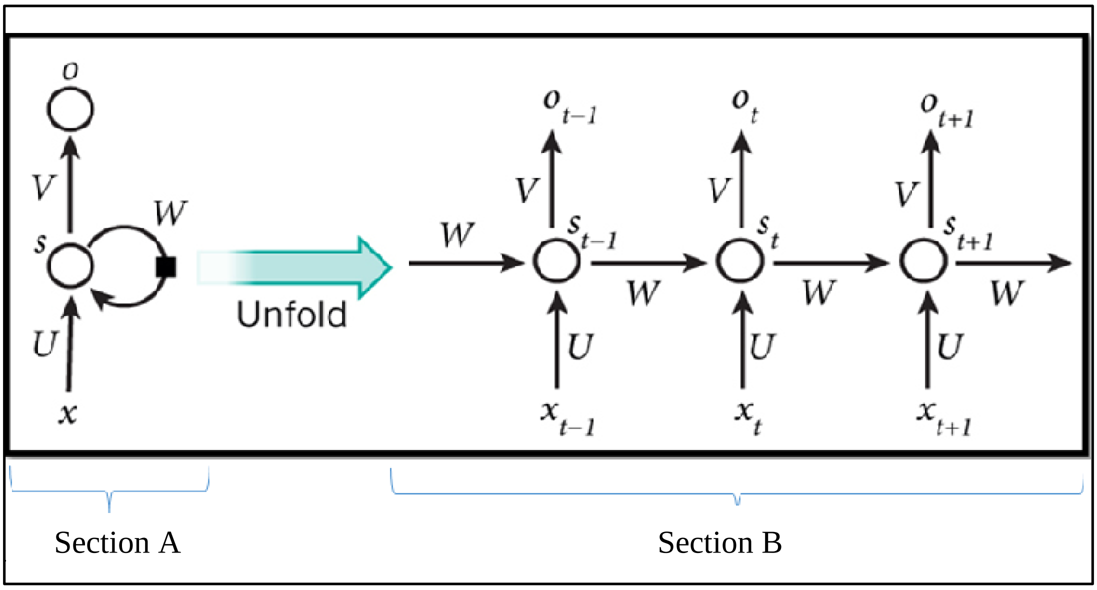

Fig. 3.14: The state of Recurrent Neural Network

Section A illustrates the folded state of RNNs and section B illustrates the unfolded state of RNN into a network. The unfolded state of RNN simply illustrates the network of complete sequence. From the section B, we can see that it is a three layer neural network. And this can be referred to as deep learning neural network because it has more than 1 hidden layer. From figure above, below are the meanings of the various parameters used in RNN computations as shown in Fig. 3.14.

- U; V and W, represents the weight of neurons.

For instance, W represents the weight of neurons between hidden state S. V represents the weight of neurons between hidden states S and the output O. U represents the weight of neurons between the inputs X and the hidden state S.

A major difference in RNN as compared to the traditional deep neural network is that, in RNN, all the three weights at any point in the operation of the RNNs, will have the same value, but the values will be different in the case of traditional neural network. This is because the same task is performed at each step with different inputs parameters. This reduces the total number of parameters RNN will need to learn. To update these weights, we use Back Propagation Through Time (BPTT) (Bediako, 2017).

- X_t is the input at time step t.
- S_t is the hidden state at time step t. It's the memory of the network. S_t is calculated based on the previous hidden state (S_{t-1}) and the input at the current state.

  $S_t = f\,(UX_t + WS_{t-1})$ …(23)

The function f usually is a non-linearity such as tanH or ReLU. S_{t-1}, which is required to calculate the first hidden state, is typically initialized to all zeroes.

- O_t is the output at step t .

$O_t = \text{softmax}(V\,S_t)$ …(24)

From figure, Error calculation for the RNN unfolding is as follows:

- $E_{Total} = \sum \frac{1}{2}\,(Given_{output} - Actual_{output})^2$ …(25)

3.3.2.7.1 Sequence problems

As LSTM name suggest sequential learning, so the simplest machine learning problem involving a sequence is a one to one problem.

Fig 3.15 shows a few examples of how the structure of the network may be designed, depending on whether the input or output (or both) is a sequence. Orange circle represent input vectors, blue rectangles represent hidden vector and green circle represent output vectors RNN blocks.

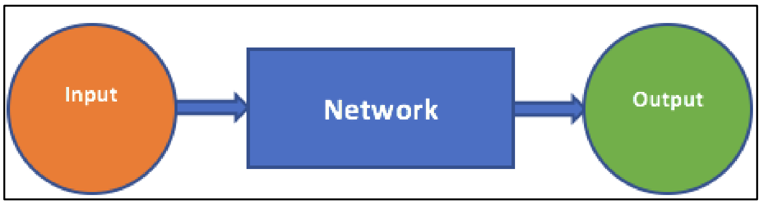

Fig. 3.15: One to One sequential Network

It is also known as the one to many problems. The one to many problem (Fig. 3.16) starts like the one to one problem where we have an input to the model and the model generates one output. However, the output of the model is now fed back to the model as a new input. The model now can generate a new output and it can continue like this indefinitely. This is why these are known as recurrent neural networks (Pant, N., 2017).

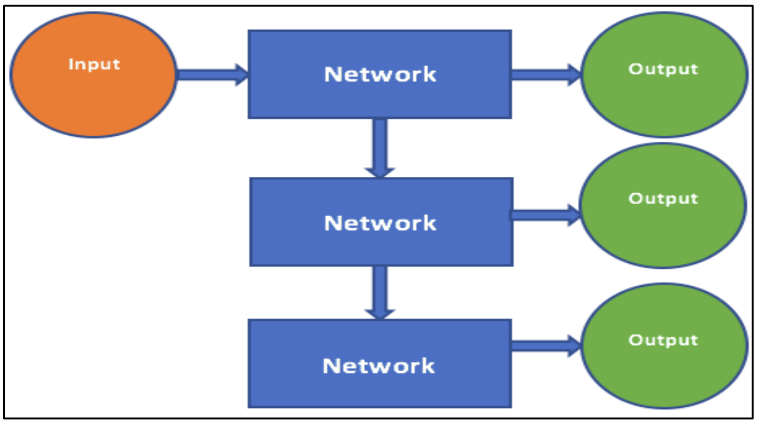

Fig. 3.16: One to Many sequential networks

A recurrent neural network deals with sequence problems because their connections form a directed cycle. In other words, they can retain state from one iteration to the next by using their own output as input for the next step. In programming terms this is like running a fixed program with certain inputs and some internal variables. The simplest recurrent neural network can be viewed as a fully connected neural network if we unroll the time axes.

3.3.2.7.2 Recurrent Neural Training

A deep recurrent neural network can be built up by simply stacking units to one another. A simple recurrent neural network works well only for a short-term memory. This network is not suitable for the situation when there is longer time dependency. So for overcoming the problem we use Long Short-Term Neural Network (LSTM).

3.3.2.7.2.1 Long Short-Term Neural Network

Long-Short-Term Memory Recurrent Neural Network belongs to the family of deep learning algorithms. It is a recurrent network because of the feedback connections in its architecture (Fig. 3.17). It has an advantage over traditional neural networks due to its capability to process the entire sequence of data. Its architecture comprises the input gate, output gate and forget gate. These gates provide read, write and reset operations on the network memory unit to retain and update the status of memory unit (Bai, X., 2018).

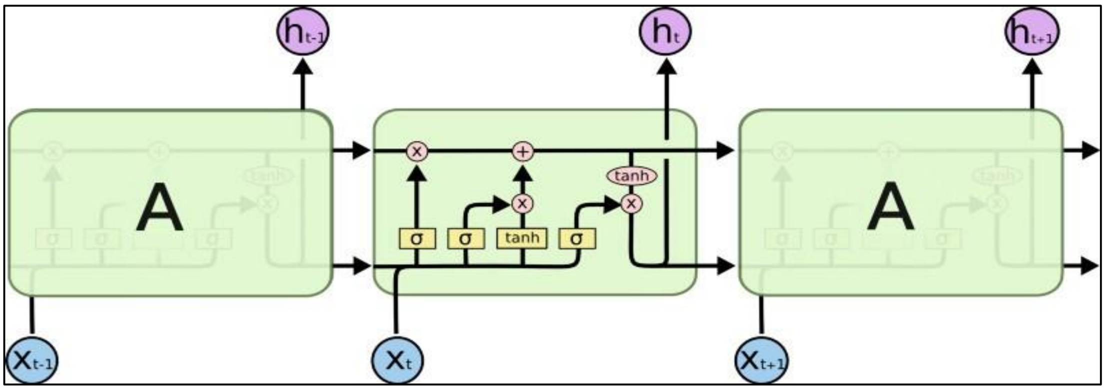

Fig. 3.17: LSTM Architecture

This model is organized in cells which include various operations. LSTM has an internal state variable, which is passed from one cell to another cell and is modified by Operation Gates.

1. **Forget Gate**

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \qquad \ldots(26)$$

It is a sigmoid layer that takes the output at *t-1* and the current input at time *t* and concatenates them into a single tensor and applies a linear transformation followed by a sigmoid. Because of the sigmoid, the output of this gate is between 0 and 1. This number is multiplied with the internal state and that is why the gate is called a forget gate. If $f_t=0$ then the previous internal state is completely forgotten, while if $f_t=1$ it will be passed through unaltered (Gers *et. al.,* 2000).

2. **Input Gate**

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \qquad ….(27)$$

The input gate takes the previous output and the new input and passes them through another sigmoid layer. This gate returns a value between 0 and 1. The value of the input gate is multiplied with the output of the candidate layer.

$$C_t = tanh(W_c . [h_{t-1}, x_t] + b_c) \qquad ….(28)$$

This layer applies a hyperbolic tangent to the mix of input and previous output, returning a candidate vector to be added to the internal state.

The internal state is updated with this rule:

$$C_t = f_t \times C_{t-1} + i_t \times C_t \qquad …(29)$$

The previous state is multiplied by the forget gate and then added to the fraction of the new candidate allowed by the output gate (Bai, X., 2018).

3. **Output Gate**

$$O_t = \sigma(W_O . [h_{t-1}, x_t] + b_O) \qquad …(30)$$

$$h_t = \sigma_t \times \tanh C_t \qquad …(31)$$

This gate controls how much of the internal state is passed to the output and it works in a similar way to the other gates.

These three gates described above have independent weights and biases, hence the network will learn how much of the past output to keep, how much of the current input to keep, and how much of the internal state to send out to the output (Bai, X., 2018).

3.3.3 Evaluation of Forecasting Methods

Evaluation of these forecasting methods has been done using different standard criterion like Root Mean Square Error (RMSE), Mean Absolute Per cent Error (MAPE) (Kumar, 2010).

3.3.3.1 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is defined as the square root of mean square error which is sum of squared errors divided by total numbers of observations. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(F_t - A_t)^2} \qquad \ldots(32)$$

3.3.3.2 Mean Absolute Per cent Error (MAPE)

Mean Absolute Per cent Error (MAPE) is a measure of accuracy of a method for constructing fitted time series values in statistics, specifically in trend estimation. It usually expresses accuracy as a percentage and is defined by the formula:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{(F_t - A_t)}{A_t}\right| \times 100 \qquad \ldots(33)$$

Where F_t is a forecasted value for time t, A_t is the actual value for time t, n is the total number of forecasts.

4. RESULTS AND DISCUSSION

Price as well as other forecast viz., production, productivity, yield, weather data, etc. well in advance play a decisive role for governmental agencies in formulating policies regarding export-import, food grain procurement and distribution, price policies and for countering storage and other marketing issues and is also beneficial for farmers as it helps in pre-planning of activities and avert unwanted situations. Monthly average price (₹/quintal) has been taken for Banana crop for eleven years from Agriculture Marketing website for Rajpipla market, Narmada, Gujarat from January 2009 to December 2019.

4.1 DATASET DESCRIPTION

The time-series dataset for the present study was taken as monthly average price from January 2009 to December 2019, a total of 132 data points, for Rajpipla market, Narmada, Gujarat (http://agmarknet.gov.in/). Table 4.1 presents the mean, standard deviation, minimum, maximum and quartiles for the time-series dataset.

Table 4.1: Time series data description for banana price in Rajpipla market

Statistic	₹/q
Number of observations	132
Mean	806.76
Standard deviation	384.83
Minimum value	183.33
1st quartile	379.57
2nd quartile	800.00
3rd quartile	1200.00
Maximum value	1533.33

The Department of Agriculture & Cooperation, Ministry of Agriculture launched "National Horticulture Mission"(NHM), during the X Five year Plan with effect from 2005-06, for the holistic development of horticulture sector duly ensuring forward and backward linkages, with the active participation of all the stake-holders including farmers and private entrepreneurs.

Banana production in 2005-06 was 1888 MT which rose to 29780 MT in 2010-11. Thus a robust push was given to Banana production nationally which resulted in improved and higher prices from 2010 onwards.

Time-series plot for average monthly prices for banana in Rajpipla market, Narmada, Gujarat from January 2009 to December 2019 is shown in Fig. 4.1. A visual inspection of Fig. 4.1 indicated towards non-stationary nature of the time-series. But for confirmation, Auto correlation function (ACF) and Partial Auto correlation function (PACF) were generated.

The histogram of time series data has been shown in Fig. 4.2. It can be seen that the time series data does not follow the normal distribution.

For detection of possible presence of seasonality, trend, time varying variance and other non-linear phenomena, the time plot (Fig. 4.3) of the banana price was generated. This helped us in determining possible order of differencing.

The Seasonal decomposition of time series data for banana price of Rajpipla market, Narmada, Gujarat is shown in Fig. 4.3. The impression of the series is that it exhibits a regular seasonal pattern and deviation from linearity is indicated from the trend curve. Some of the autocorrelations of the residuals appear to be rather large in magnitude, but the autocorrelation function as a whole exhibits no discernible pattern.

4.2 FORECASTING OF BANANA PRICE USING ARIMA MODEL

For fitting the ARIMA model, the three stages of modelling viz; identification, estimation and diagnostic checking as suggested by Box and Jenkins has been undertaken. The analysis of dataset and results of price forecast of Banana using ARIMA model is presented below.

4.2.1 Identification of model

The plots of Auto correlation function (ACF) and Partial Auto correlation function (PACF) were generated for determining the stationarity.

On the basis of ACF and PACF plots as shown in Fig. 4.4 and 4.5 respectively, the time-series under consideration is non-stationary in nature, which is further affirmed by the results of Augmented Dickey-Fuller (ADF) unit root test given in Table 4.2.

Augmented Dickey-Fuller (ADF) test was carried out considering the null hypothesis,

H_0: Time series is non-stationary.

Table 4.2: Augmented Dickey-Fuller (ADF) test for banana prices of Rajpipla market

	t-statistic	Probability
ADF test value	1.90	0.33

The results of ADF test in Table 4.2 reveals that the p value = 0.33 is coming out to be non-significant at 5% level of significance, so we accept the null hypothesis, that our series is non-stationary.

Once the series is identified as non-stationary, then first order differencing of the series was done so as to transform it into stationary series as shown in Fig. 4.6. Time plot of the differenced series evidently demonstrates that the series is now stationary.

The plot of ACF and PACF of differenced series is presented in Fig. 4.7 and 4.8 respectively. The price series became stationary after first differencing.

Table 4.3: Augmented Dickey-Fuller (ADF) test for differenced time-series

	t-statistic	Probability
ADF test value	7.63	<0.01

Results of Augmented Dickey-Fuller (ADF) test shown in Table 4.3 revealed that the original series which was non-stationary became stationary after first order differencing. The p-value came out to be less than 0.01, at the 5% level of significance, which means the difference is significant and thus we reject the null hypothesis, that is now the series became stationary.

4.2.2 Estimation of parameters for ARIMA (5, 1, 0) model

Out of various parametric combinations, ARIMA (5, 1, 0) was found to be the best and the values of parameter co-efficient, its standard error, z-value and p-value are presented in Table 4.4.

Table 4.4: Parameter estimates for ARIMA (5, 1, 0) model

Parameter	Coefficient	Standard Error	z-value	P>|z|	Conclusion
ar.L1	-0.19	0.08	2.39	0.01	S
ar.L2	-0.03	0.06	0.56	0.57	NS
ar.L3	-0.08	0.14	0.56	0.57	NS
ar.L4	-0.19	0.09	1.99	0.04	S
ar.L5	0.11	0.01	6.88	0.03	S

Table 4.4 reveals that 1^{st}, 4^{th} and 5^{th} parameters were found significant at 5% level of significance while 2^{nd} and 3^{rd} are non-significant.

4.2.3 Performance measure and diagnostic check for ARIMA model

Table 4.5 indicates the values of Akaike information criteria (AIC) and Bayesian information criteria (BIC). The best models were selected based on AIC and BIC values. Lower the value of AIC and BIC, better is the model to forecast. Out of various models tried, ARIMA (5, 1, 0) was found with lowest AIC and BIC value and was considered as best fitted model for forecasting banana price for Rajpipla market, Narmada, Gujarat.

Table 4.5: AIC and BIC test statistic value for fitted ARIMA model

Fitted model	(5, 1, 0)
AIC	1702.86
BIC	1720.11

4.2.4 Portmanteau-Quenouille (PQ) test

PQ test is used for testing of autocorrelation in the residuals of a model, that is, it tests whether any of a group of autocorrelations of the residual time series is different from zero.

Table 4.6: Portmanteau-Quenouille test of ARIMA (5, 1, 0) model

Order	PQ test statistic	p-value
4	0.60	0.86
8	0.88	0.89
12	1.43	0.91
16	1.65	0.97
20	2.52	0.99

The adequacy of model was judged on the basis of Q statistic. The values of the test are shown in Table 4.6, it was observed as non-significant at 5% level of significance, revealing that there is no auto-correlation among the residuals of fitted model.

4.2.5 Lagrange Multiplier Test

The LM test is a specific linearity test in the sense that it has a linear time series model (usually an AR (MA) type model) under the null hypothesis and a specified nonlinear time series model as the alternative hypothesis. It is basically for detecting nonlinearities in the time series.

Table 4.7: Lagrange Multiplier Test of ARIMA (5, 1, 0) model

Order	LM test statistic	p-value
4	312.9	<0.01
8	137.2	<0.01
12	75.2	<0.01
16	46.8	<0.01
20	28.0	<0.01

The linearity of fitted model was judged based on Lagrange's Multiplier test. The values of the test statistic at different lags are shown in Table 4.7 and it was observed as highly significant at 5% level of significance, which denotes that there is strong presence of ARCH impact in the banana price series.

4.2.6 Cross validation check of ARIMA model

The data was randomly divided into training and testing part and for testing performance of the model, the same was applied to unseen data and its prediction was noted along with their respective observed value. The observed and predicted values are given in Table 4.8 and also the graph for same is shown in Fig. 4.9.

Table 4.8: Banana price prediction on testing dataset by ARIMA (5, 1, 0) model

Sr. No.	Observed value	Predicted value
1	183.33	377.84
2	187.04	473.61
3	194.00	748.49
4	225.20	361.23
5	227.75	373.71
6	228.80	154.24
7	350.00	563.70
8	650.00	680.51
9	647.00	726.16
10	795.45	756.77
11	800.00	793.40
12	830.76	800.00

On the basis of observed and predicted values from testing dataset, RMSE and MAPE value were calculated for ARIMA (5, 1, 0) model. RMSE and MAPE values were 209.78 and 65.79 respectively.

4.2.7 Price Forecasting of Banana by ARIMA (5, 1, 0) model

The forecasted banana price for Rajpipla market, Narmada, Gujarat using ARIMA technique is presented in Table 4.9. Table below shows the actual value and the forecasts of banana price as fitted by ARIMA (5, 1, 0) model from January 2020 to December 2020 and graph for actual price vs forecasted price is shown in Fig. 4.10.

Table 4.9: Actual value and forecasts of banana price by ARIMA (5, 1, 0) model

Month & Year	Actual price (₹/q)	Forecast price (₹/q)	Forecast error (%)
Jan-2020	1187.68	233.50	80.33
Feb-2020	1133.15	227.77	79.89
Mar-2020	1092.35	235.12	78.47
Apr-2020	564.41	238.02	57.82
May-2020	566.66	234.10	58.68
Jun-2020	510.50	235.67	53.83
Jul-2020	564.77	233.15	58.71
Aug-2020	666.42	234.20	64.85
Sep-2020	815.25	235.03	71.17
Oct-2020	920.65	234.30	74.55
Nov-2020	900.00	235.80	73.80
Dec-2020	900.00	234.33	73.96
p value <0.01			

The p value for difference between forecasts and actual price was <0.01 as shown in Table 4.9, depicting significant difference, also the accuracy of price forecast was examined by calculating forecast error per cent for the given model. For the fitted ARIMA (5, 1, 0) model, the forecast error per cent was more than 50%, indicating poor forecast by the model.

4.3 FORECASTING OF BANANA PRICE USING SARIMA MODEL

The analysis of dataset and results of price forecast of banana using SARIMA model is presented below.

4.3.1 Identification of model

After trying out various models, SARIMA (0, 1, 2) x (0, 1, 2, 4) was found performing better in the present case.

4.3.2 Estimation of parameters for SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Out of various parametric combinations, SARIMA (0, 1, 2) x (0, 1, 2, 4) was found to be the best and the values of parameter co-efficient, its standard error, z-value and p-value are presented in the Table 4.10.

Table 4.10: Parameter estimates for SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Parameter	Coefficient	Standard error	Z	P>|z|	Conclusion
ma.L1	-0.20	0.09	2.15	0.03	S
ma.L2	-0.03	0.09	0.36	0.71	NS
sma1	-1.15	0.10	10.81	<0.01	S
sma2	0.25	0.10	2.41	0.01	S

Table 4.10 reveals that 1^{st}, 3^{rd} and 5^{th} parameters were found significant at 5% level of significance while 2^{nd} parameter ma.L2 was non-significant.

4.3.3 Performance measure and diagnostic check for SARIMA model

Table 4.11 indicates the values of Akaike information criteria (AIC) and Bayesian information criteria (BIC). The best models were selected based on AIC and BIC values. Lower the value of AIC and BIC, better is the model to forecast. Out of various models tried, SARIMA (0, 1, 2) x (0, 1, 2, 4) was found with lowest AIC and BIC value and was considered as best fitted for forecasting banana price for Rajpipla market, Narmada, Gujarat.

Table 4.11: AIC and BIC test statistic value for fitted SARIMA model

Fitted model	(0, 1, 2) x (0, 1, 2, 4)
AIC	1671.54
BIC	1685.77

4.3.4 Portmanteau-Quenouille (PQ) test

PQ test is used for testing of autocorrelation in the residuals of a model, that is, it tests whether any of a group of autocorrelations of the residual time series is different from zero.

Table 4.12: Portmanteau-Quenouille test of SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Order	PQ test statistic	p-value
4	1.42	0.84
8	1.81	0.98
12	2.53	0.99
16	3.09	1.00
20	4.59	1.00

The adequacy of model was judged on the basis of Q statistic. The values of the test are shown in Table 4.12, it was observed as non-significant at 5% level of significance, revealing that there is no auto-correlation among the residuals of fitted model.

4.3.5 Lagrange Multiplier Test

The LM test is a specific linearity test in the sense that it has a linear time series model (usually an AR (MA) type model) under the null hypothesis and a specified nonlinear time series model as the alternative hypothesis. It is basically for detecting nonlinearities in the time series.

Table 4.13: Lagrange Multiplier Test of SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Order	LM test statistic	p-value
4	213.7	<0.01
8	92.8	<0.01
12	52.3	<0.01
16	32.9	<0.01
20	19.4	<0.01

The linearity of fitted model was judged based on Lagrange’s Multiplier test. The values of the test statistic at different lags are shown in Table 4.13 and it was observed as highly significant at 5% level of significance, which denotes that there is strong presence of ARCH impact in the banana price series.

4.3.6 Cross validation check of SARIMA model

The data was randomly divided into training and testing part and for testing performance of the model, the same was applied to unseen data and its prediction was noted along with their respective observed value. The observed and predicted values are given in Table 4.14 and also the graph for same is shown in Fig. 4.11.

Table 4.14: Banana price prediction on testing dataset by SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Sr. No.	Observed value	Predicted value
1	211.08	245.53
2	187.04	280.91
3	225.20	295.45
4	238.80	342.77
5	225.63	335.41
6	235.58	384.06
7	227.75	373.71
8	258.23	447.35
9	254.50	490.02
10	260.00	554.64
11	243.06	582.01
12	250.80	637.20

On the basis of observed and predicted values from testing dataset, RMSE and MAPE value were calculated for SARIMA (0, 1, 2) x (0, 1, 2, 4) model. RMSE and MAPE values were 208.84 and 74.13 respectively.

4.3.7 Price Forecasting of banana by SARIMA (0, 1, 2) x (0, 1, 2, 4) model

The forecasted banana price at Narmada market, using SARIMA technique is presented in Table 4.15. Table below shows the actual value and the forecasts of banana price as fitted by SARIMA (0, 1, 2) x (0, 1, 2, 4) model from January 2020 to December 2020 and graph for Actual price vs forecasted price is shown in Fig. 4.12.

Table 4.15: Actual value and forecasts of banana price by SARIMA (0, 1, 2) x (0, 1, 2, 4) model

Month & Year	Actual price (₹/q)	Forecast price (₹/q)	Forecast error (%)
Jan-2020	1187.68	305.94	74.24
Feb-2020	1133.15	297.67	73.73
Mar-2020	1092.35	322.60	70.46
Apr-2020	564.41	338.45	40.03
May-2020	566.66	303.71	46.40
Jun-2020	510.50	295.44	42.12
Jul-2020	564.77	320.37	43.27
Aug-2020	666.42	336.22	49.54
Sep-2020	815.25	301.48	63.01
Oct-2020	920.65	293.21	68.15
Nov-2020	900.00	318.14	64.65
Dec-2020	900.00	333.99	62.89
p value <0.01			

The p value for difference between forecasts and actual price was <0.01 depicting significant difference, also the accuracy of price forecast was examined by calculating forecast error per cent for the given model. For the fitted SARIMA (0, 1, 2)x(0, 1, 2, 4) model, the forecast error per cent was between 40-75% indicating poor forecast by the model.

4.4 FORECASTING OF BANANA PRICE USING ARCH/GARCH MODEL

The analysis of dataset and results of price forecast of banana using ARCH/GARCH model is presented below.

4.4.1 Estimation of parameter for GARCH model

Out of various parametric combinations, AR (1,0) + EGARCH (0,1) was found to be the best and the values of parameter co-efficient, its standard error, z-value and p-value are presented in Table 4.16.

Table 4.16: Parameter estimates for fitted AR (1,0) + EGARCH (0,1) model

Fitted Model: AR (1,0) + EGARCH (0,1)				
Parameter	**Coefficient**	**Standard error**	**t-value**	**p-value**
Omega	0.62	1.40	0.66	0.51
Gamma [1]	0.14	7.06	2.02	0.04
Beta [1]	0.31	0.13	6.54	<0.01

Table 4.16 reveals that except the constant term of variance model, the Gamma parameter and Beta parameter of error variance at one lag were statistically significant at 5% level of significance.

4.4.2 Performance measure and diagnostic check for GARCH model

Table 4.17 indicates the values of Akaike information criteria (AIC) and Bayesian information criteria (BIC). The best models were selected based on AIC and BIC values. Lower the value of AIC and BIC, better is the model to forecast. Out of various models tried, AR (1,0) + EGARCH (0,1) was found with lowest AIC and BIC value and was considered as best fitted for forecasting banana price for Rajpipla market, Narmada, Gujarat.

Table 4.17: AIC and BIC values of AR (1,0) + EGARCH (0,1)

Fitted model	AR (1,0) + EGARCH (0,1)
AIC	1701.06
BIC	1715.44

4.4.3 Lagrange Multiplier Test

The LM test is a specific linearity test in the sense that it has a linear time series model (usually an AR (MA) type model) under the null hypothesis and a specified nonlinear time series model as the alternative hypothesis. It is basically for detecting nonlinearities in the time series.

Table 4.18: Lagrange Multiplier Test of AR (1,0) + EGARCH (0,1)

Order	LM test statistic	p-value
4	321.9	<0.01
8	136.8	<0.01
12	73.6	<0.01
16	46.9	<0.01
20	28.6	0.07

The linearity of fitted model was judged based on Lagrange's Multiplier test. The values of the test are shown in Table 4.18 and it was observed as highly significant at 5% level of significance, inferring presence of non-linearity in the dataset

4.4.4 Cross validation check of GARCH model

The data was randomly divided into training and testing part and for testing performance of the model, the same was applied to unseen data and its prediction was noted along with their respective observed value. The observed and predicted values are given in Table 4.19 and also the graph for same is shown in Fig. 4.13.

Table 4.19: Banana price prediction on testing dataset by AR (1,0) + EGARCH (0,1)

Sr. No.	Observed value	Predicted value
1	210.08	245.53
2	197.84	260.91
3	235.20	275.45
4	228.80	304.77
5	223.63	325.41
6	244.02	374.06
7	227.75	403.71
8	268.23	457.35
9	254.50	482.02
10	250.00	504.64
11	238.96	512.01
12	298.80	537.20

On the basis of observed and predicted values from testing dataset, RMSE and MAPE value were calculated for AR (1,0) + EGARCH (0,1) model. RMSE and MAPE values were 171.90 and 60.91 respectively.

4.4.5 Price Forecasting of banana by GARCH model

The forecasted banana price at Narmada market, using GARCH technique is presented in Table 4.20. Table below shows the actual value and the forecasts of banana price as fitted by AR (1,0) + EGARCH (0,1) model from January 2020 to December 2020 and graph for Actual price vs forecasted price is shown in Fig. 4.14.

Table 4.20: Actual value and forecasts of banana price by AR (1,0) + EGARCH (0,1) model

Month & Year	Actual price (₹/q)	Forecast price (₹/q)	Forecast error (%)
Jan-2020	1187.68	585.91	50.66
Feb-2020	1133.15	636.30	43.84
Mar-2020	1092.35	681.69	37.59
Apr-2020	564.41	722.58	28.02
May-2020	566.66	759.42	34.01
Jun-2020	510.50	798.96	56.50
Jul-2020	564.77	737.83	30.64
Aug-2020	666.42	776.70	16.54
Sep-2020	815.25	810.57	0.57
Oct-2020	920.65	864.44	6.10
Nov-2020	900.00	893.30	0.74
Dec-2020	900.00	932.17	3.57
p value = 0.27			

The p value came out to be 0.27 for difference between forecasts and actual price, depicting non-significant difference, also the accuracy of price forecast was examined by calculating forecast error per cent for the given model. For the fitted AR (1,0) + EGARCH (0,1) model, the forecast error per cent though shows a declining pattern, but still could not provide a satisfactory forecast.

4.5 DEEP LEARNING FOR FORECASTING OF BANANA PRICE

4.5.1 Forecasting Of Banana Price Using Artificial Neural Network (ANN)

4.5.1.1 Development of ANN architecture for price forecasting of banana in Rajpipla market

Preprocessing of data were done through MinMaxScaler function which is imported from Sklearn library and reverse transformation of the data was done by using function scaler.inverse_transform()

After preprocessing of data, Sequential model was defined.

Out of various parametric combinations, following architecture (Table 4.21) and hyperparameters (Table 4.22) were considered to be the best with respect to the data and algorithm during programming and tuning through python software 3.8.

Table 4.21: Sequential Model architecture

```
Model: "sequential_27"
_________________________________________________________________
Layer (type)                 Output Shape
Param #
=================================================================
dense_27 (Dense)             (None, 3)                 12
_________________________________________________________________
dense_28 (Dense)             (None, 10)                40
_________________________________________________________________
dense_29 (Dense)             (None, 10)                110
_________________________________________________________________
dense_30 (Dense)             (None, 1)                 11
=================================================================
Total params: 173
Trainable params: 173
Non-trainable params: 0
```

Table 4.22: Hyperparameters Of ANN

Layers	Neurons	Activation Function
Input	03	LekyReLU
Hidden1	10	LekyReLU
Hidden2	10	LekyReLU
Output	1	Linear
Other hyperparmeters		
Learning Rate	0.0001	
Objective function	minimise the error	
Loss function	Mean square error (MSE)	
Optimiser	Adaptive Moment Estimation (Adam)	
Batch size	20	
Epochs	450	
validation_split	0.2	

4.5.1.2 Cross validation check of ANN model

The data was randomly divided into training and testing part and for testing performance of the model, the same was applied to unseen data and its prediction was noted along with their respective observed value. The observed and predicted values are given in Table 4.23 and also the graph for same is shown in Fig. 4.15.

Table 4.23: Banana price prediction on testing dataset by ANN model

Sr. No.	Observed value	Predicted value
1	710.41	915.43
2	500.00	582.35
3	647.00	720.28
4	974.46	1074.10
5	250.00	420.62
6	710.41	915.43
7	361.65	366.50
8	1229.24	1276.93
9	838.33	884.93
10	1161.90	1196.61
11	710.41	915.43
12	500.00	582.35

On the basis of observed and predicted values from testing dataset, RMSE and MAPE value were calculated for ANN model. RMSE and MAPE values were 116.99 and 16.56 respectively.

4.5.1.3 Price Forecasting of banana by ANN model

The forecasted banana price at Narmada market, using ANN technique is presented in Table 4.24. Table below shows the actual value and the forecasts of banana price as fitted by ANN model from January 2020 to December 2020 for Rajpipla market, Narmada, Gujarat and graph for Actual price vs forecasted price is shown in Fig. 4.16.

Table 4.24: Actual value and forecasts of banana price by ANN model

Month & Year	Actual price (₹/q)	Forecast value (₹/q)	Forecast error (%)
Jan-2020	1187.68	654.15	44.92
Feb-2020	1133.15	628.22	44.55
Mar-2020	1092.35	618.19	43.40
Apr-2020	564.41	598.41	6.02
May-2020	566.66	588.25	3.81
Jun-2020	510.50	545.40	6.83
Jul-2020	564.77	570.23	0.96
Aug-2020	666.42	647.02	2.91
Sep-2020	815.25	818.57	0.04
Oct-2020	920.65	925.31	0.50
Nov-2020	900.00	932.16	3.57
Dec-2020	900.00	933.08	3.67
p value = 0.06			

The p value came out to be 0.06 for difference between forecasts by ANN and actual prices, showing no significant difference, also the accuracy of price forecast was examined by calculating forecast error per cent for the given model. For the fitted ANN model, the forecast error per cent though high initially shows a drastic declining pattern after the March month, thus can be taken into consideration for developing into a reliable model.

4.5.2 FORECASTING OF BANANA PRICE USING RECURRENT NEURAL NETWORK (RNN)

4.5.2.1 Development of RNN architecture for price forecasting of banana in Rajpipla market

Preprocessing of data was done through MinMaxScaler function which was imported from the library Scikit-learn. Then from keras.preprocessing.sequence, TimeseriesGenerator was imported and a function was created to define generator of this time-series data.

Following are the inputs used to create TimeseriesGenerator.

- n_input = 3; where n_input is the number of lags which were considered as input values,
- n_features = 1; where n_features is 1 as we are using uni-variate time-series data.
- batch_size = 1; where batch_size is the number of batches we have used to train the model.

After creation of generator, Sequential model was defined.

Out of various parametric combination, following architecture (Table 4.25) and hyperparameters (Table 4.26) were set with respect to the nature of data and algorithm during programming and tuning through Python.

Table 4.25: Sequential Model architecture of simple lstm (RNN)

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_10 (LSTM)                (None, 10)                   480
_________________________________________________________________
dense_9 (Dense)              (None, 1)                     11
=================================================================
Total params: 491
Trainable params: 491
Non-trainable params: 0
_________________________________________________________________
```

Table 4.26: Hyperparameters of simple lstm (RNN)

Layers	Neurons	Activation Function
Input	02	-
Hidden	10	LekyReLU
Output	1	Linear
Other Hyperparameters		
Learning Rate	0.0001	
Objective function	minimise the error	
Loss function	Mean square error (MSE)	
Optimiser	Adaptive Moment Estimation (Adam)	
Batch size	10	
Epochs	500	
Validation_split	0.2	

4.6.2.2 Cross validation check of RNN model

The data was randomly divided into training and testing part and for testing performance of the model, the same was applied to unseen data and its prediction was noted along with their respective observed value. The observed and predicted values are given in Table 4.27 and also the graph for same is shown in Fig. 4.17.

Table 4.27: Banana price prediction on testing dataset by RNN model

Sr. No.	Observed value	Predicted value
1	654.99	535.53
2	830.76	798.81
3	795.45	768.79
4	800.00	707.09
5	900.00	881.23
6	933.33	847.68
7	793.75	698.13
8	699.99	607.03
9	866.65	818.81
10	755.54	885.66
11	499.99	571.61
12	350.41	280.50

On the basis of observed and predicted values from testing dataset, RMSE and MAPE value were calculated for RNN model. RMSE and MAPE values were 84.60 and 9.58 respectively.

4.6.2.3 Price Forecasting of Banana by RNN model

The forecasted Banana price at Narmada market, using RNN technique is presented in Table 4.28. Table below shows the actual value and the forecasts of banana price as fitted by RNN model from January 2020 to December 2020 for Rajpipla market, Narmada, Gujarat, and graph for Actual price vs forecasted price is shown in Fig. 4.18.

Table 4.28: Actual value and forecasts of banana price by RNN model

Month & Year	Actual price (₹/q)	Forecast value (₹/q)	Forecast error (%)
Jan-2020	1187.68	1172.21	1.30
Feb-2020	1133.15	1142.67	0.84
Mar-2020	1092.35	1100.98	0.79
Apr-2020	564.41	837.20	48.33
May-2020	566.66	568.71	0.36
Jun-2020	510.50	515.86	1.04
Jul-2020	564.77	554.99	1.73
Aug-2020	666.42	671.19	0.71
Sep-2020	815.25	810.86	0.53
Oct-2020	920.65	913.10	0.82
Nov-2020	900.00	896.02	0.44
Dec-2020	900.00	903.43	0.38
p value = 0.177			

The p value came out to be 0.177 for difference between forecasts by RNN and actual prices, showing no significant difference accuracy of price forecast was examined by calculating forecast error per cent for the given model. For the fitted RNN model, the forecast error per cent was below 2% except for 1 month, which shows RNN provides a promising model for Banana price forecast for year 2020.

4.6 FORECAST ACCURACY

Table 4.29 gives the Root Mean Square Error (RMSE) and Mean Absolute Per cent Error (MAPE) values for different fitted models.

Table 4.29: Comparison of different models based on RMSE and MAPE values

Model	RMSE	MAPE
ARIMA	209.78	65.79
SARIMA	208.84	74.13
ARCH/GARCH	171.90	60.91
ANN	116.99	16.56
RNN	84.60	9.58

Amongst the fitted model, RNN has the least RMSE value while ARIMA showed highest, whereas the lowest MAPE value amongst the fitted models was demonstrated by RNN followed by ANN while SARIMA has highest. So, RNN proved to be the best fitted model out of all.

5. SUMMARY AND CONCLUSION

In the present study, forecasting models based on ARIMA, SARIMA, ARCH/GARCH, ANN and RNN were developed for the price forecasting of Banana for year 2020 for Rajpipla market of Narmada district in Gujarat. The fact that price forecasting of horticultural commodities is always and will remain difficult task because such data are greatly affected by trade, both at national and international level, political and even natural shocks. The year of 2020, which showed way more different pattern owing to covid-19 induced lockdown, import-export restrictions, thus affecting the overall price pattern.

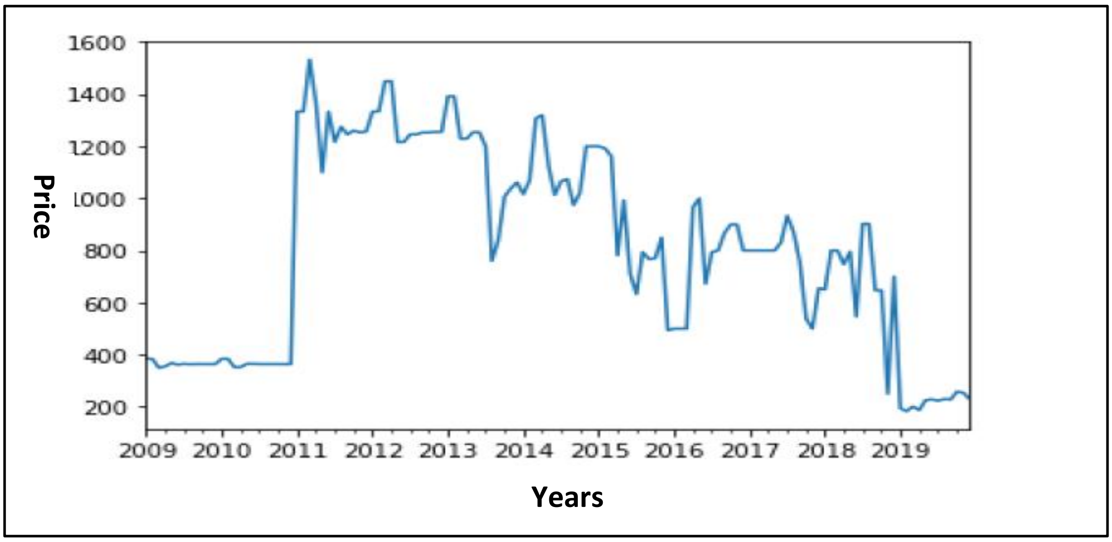

Fig. 5.1: Original dataset pattern

Results of the present study showed that RNN model forecast were considerably more accurate when compared to ARIMA, SARIMA, ARCH/GARCH and ANN. Monthly data for about 10 years was taken from 1st January 2009 to 31st December 2019 (Fig. 5.1) randomly, for training the models and 12 dataset were taken randomly for testing the model. On the basis of Test value, price for January - December 2020 was forecasted.

For evaluating the forecast accuracy of models, Root Mean Square Error (RMSE) and Mean Absolute Per cent Error (MAPE) test was carried out.

5.1 Following objectives were undertaken:

1. To study different statistical models for price forecasting of banana.

2. To study deep learning artificial intelligence for price forecasting of banana.

3. To identify and suggest the most accurate price forecasting technique.

5.1.1 To study different statistical models for price forecasting of banana.

In the present study, forecasting models based on ARIMA, SARIMA and ARCH/GARCH were developed to forecast the prices for the year 2020. RMSE and MAPE were used to evaluate forecasting models.

Following conclusions were drawn:

- For the Rajpipla market, ARCH/GARCH model performed better than ARIMA and SARIMA on the basis of lower values of RMSE and MAPE.

5.1.2 To study deep learning artificial intelligence for price forecasting of banana.

In the present study, forecasting models based on ANN and RNN were developed to forecast the prices for the year 2020. RMSE and MAPE were used to evaluate forecasting models.

Following conclusions were drawn:

- For the Rajpipla market, RNN model performed better than ANN on the basis of lower values of RMSE and MAPE.

5.1.3 To identify and suggest the most accurate price forecasting technique.

Table 5.1: RMSE and MAPE values for fitted models

Model	RMSE	MAPE
ARIMA	209.78	65.79
SARIMA	208.84	74.13
ARCH/GARCH	171.90	60.91
ANN	116.99	16.56
RNN	84.60	9.58

Table 5.1 depicts the RMSE and MAPE values for various models studied. RNN has least RMSE and MAPE value while ARIMA has highest RMSE.

In conclusion, after the comparison of various models attempted in this study to forecast prices of Banana for Rajpipla market, Narmada, Gujarat for the year 2020, the Recurrent Neural Network (RNN) on the basis of RMSE and MAPE values performed better as compared to all the other models studied.

Therefore RNN might be suitable for forecasting Banana prices and can provide a basis for policy makers and farmers to optimise their decision making process in producing higher profits.

However, the model still has some shortcomings such as it considers its own price lag value and fails to integrate other factors such as weather conditions, arrival of commodity, government export-import policies and variety of commodity into the forecast. So for our future research work in Deep Learning, we can utilise these factors mainly to increase the accuracy of the forecast and in generating more reliable predictions.

REFERENCES

Agriculture marketing website. Retrived from http://agmarknet.gov.in/ . Assessed on August 2020.

Ali, G. (2013). EGARCH, GJR-GARCH, TGARCH, AVGARCH, NGARCH, IGARCH and APARCH models for pathogens at marine recreational sites. *Journal of Statistical and Econometric Methods*, *2* (3), 57-73.

Bai, X. (2018). Text classification based on LSTM and attention. *13th International Conference on Digital Information Management (ICDIM,)* 29-32.

Bediako, P. K. (2017). Long Short-Term Memory Recurrent Neural Network for detecting DDoS flooding attacks within Tensor Flow Implementation framework.

Bera, A. K. & Higgins, M. L. (1993). ARCH models: properties, estimation and testing. *Journal of economic surveys*, *7* (4), 305-366.

Bhardwaj, S. P., Paul, R. K., Singh, D. R. & Singh, K. N. (2014). An empirical investigation of ARIMA and GARCH models in agricultural price forecasting. *Economic Affairs*, *59* (3), 415-428.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, *31* (3), 307-327.

Box, G. E., Jenkins, G. M. & Reinsel, G. (1970). *Time series analysis: forecasting and control.* Holden-day San Francisco.

Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brandt, J. A. & Bessler, D. A. (1983). Price forecasting and evaluation: An application in agriculture. *Journal of Forecasting*, *2* (3), 237-248.

Brockwell, P. J. & Davis, R. A. (2016). *Introduction to time series and forecasting*. Springer

Darekar, A. & Reddy, A. (2017[a]). Forecasting of common paddy prices in India. *Journal of Rice Research*, *10* (1), 71-75.

Darekar, A. & Reddy, A. A. (2017[b]). Cotton price forecasting in major producing states. *Economic Affairs*, *62* (3), 373-378.

Darekar, A. & Reddy, A. A. (2017[c]). Price forecasting of pulses: case of pigeonpea. *Journal of Food Legumes*, *30* (3), 212-216.

Dickey, D. A. & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, *74* (366a), 427-431.

Directorate of Horticulture, Gujarat, 2020. Accessed on June 2021.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the econometric society*, 50, 987-1007.

Eni, D. (2015). Seasonal ARIMA modelling and forecasting of rainfall in Warri Town, Nigeria. *Journal of Geoscience and Environment Protection*, *3* (6), 91.

FAOSTAT. (2020) Food and Agricultural Organisation of United Nations. Retrieved from http://www.fao.org/faostat/en/#search/banana .

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

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT press.

Hanumanthaiah, R. (2018). *Development of Time Series Models for Forecasting of Prices and Arrivals of Black pepper in Karnataka*. (Unpublished master's thesis) Institute of Agricultural Sciences, Banaras Hindu University, Varanasi, Uttar Pradesh.

Haofei, Z., Guoping, X., Fangting, Y. & Han, Y. (2007). A neural network model based on the multi-stage optimization approach for short-term food price forecasting in China. *Expert Systems with Applications*, *33* (2), 347-356.

Hastie, T., Tibshirani, R. & Friedman, J. (2009). The elements of statistical learning: data mining, inference and prediction. *Springer*.

Hochreiter, S. & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 473-479.

Hsieh, W. W. (2009). Machine learning methods in the environmental sciences: Neural networks and kernels. *Cambridge university press*.

Jadhav, V., CHINNAPPA, R. B. & Gaddi, G. M. (2017). Application of ARIMA model for forecasting agricultural prices. *Journal of Agricultural Science and Technology*, 19 (4), 981-992.

Jha, G. K. & Sinha, K. (2013). Agricultural price forecasting using neural network model: An innovative information delivery system. *Agricultural Economics Research Review*, *26* (2), 229-239.

Kastens, T. L., Jones, R. & Schroeder, T. C. (1998). Futures-based price forecasts for agricultural producers and businesses. *Journal of Agricultural and Resource Economics*, 23 (1) 294-307.

Kumar, A. H. S. (2010). *Evaluation of statistical models in price forecasting – a case of timber trade*. (Unpublished master's thesis). University of Agricultural Sciences, Dharwad, Karnataka (India).

Kumar, A., Agrawal, R. & Chattopadhyay, C. (2013). Weather based forecast models for diseases in mustard crop. *Mausam*, *64* (4), 663-670.

Kumar, J. A., Muralidhar, M., Jayanthi, M. & Kumaran, M. (2013). Trend analysis of weather data in shrimp farming areas of Nagapattinam district of Tamil Nadu. *Journal of Agrometeorology,* 15 (2), 129-134.

Kumari, P., Mishra, G. C. & Srivastava, C. P. (2013). Forecasting of productivity and pod damage by *Helicoverpa armigera* using artificial neural network model in pigeonpea (*Cajanus cajan*). *International Journal of Agriculture, Environment and Biotechnology*, 6 (2), 335-340.

Kumari, P., Mishra, G. C., Pant, A. K., Shukla, G. & Kujur, S. N. (2014). Autoregressive Integrated Moving Average (ARIMA) approach for prediction of rice (*Oryza sativa*) yield in India. *The Bioscan*, *9* (3), 1063-1066.

KumarMahto, A., Biswas, R. & Alam, M. A. (2019). Short term forecasting of agriculture commodity price by using ARIMA: based on Indian market. *International Conference on Advances in Computing and Data Sciences*, 452-461.

Lama, A., Jha, G. K., Paul, R. K. & Gurung, B. (2015). Modelling and forecasting of price volatility: an application of GARCH and EGARCH models. *Agricultural Economics Research Review*, *28* (1), 73-82.

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. *Neural networks: Tricks of the trade* Springer, 9-48.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, *521* (7553), 436-444.

Li, G. Q., Xu, S. W., Li, Z. M. & DONG, X. X. (2010[a]). Study on super short-term forecasting for market price of agro-products - Based on modern times series modelling of daily wholesale price of tomatoes. *Journal of Huazhong Agricultural University*, *6*, 40-45.

Li, G. Q., Xu, S. W. & Li, Z. M. (2010[b]). Short-term price forecasting for agro-products using artificial neural networks. *Agriculture and Agricultural Science Procedia*, *1*, 278-287.

Luo, C. S., Zhou, L. Y. & Wei, Q. F. (2013). Application of SARIMA model in cucumber price forecast. *Applied Mechanics and Materials*, 373, 1686-1690.

Mahapatra, S. (2018). Why deep learning over traditional machine learning. *Towards Data Science*.

McCulloch, W. S. & Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, *52* (1), 99-115.

Mishra, G. C. & Singh, A. (2013). A study on forecasting prices of groundnut oil in Delhi by ARIMA methodology and artificial neural networks. *Agris on-line Papers in Economics and Informatics*, *5* (3), 25-34.

Mitra, D. & Paul, R. K. (2017). Hybrid time-series models for forecasting agricultural commodity prices. *Model Assisted Statistics and Applications*, *12* (3), 255-264.

National Horticulture Board website. Retrived from http://nhb.gov.in/. Assessed on June 2021.

Naveena, K. & Subedar, S. (2017). Hybrid time series modelling for forecasting the price of washed coffee (Arabica Plantation Coffee) in India. *International Journal of Agriculture Sciences*, *ISSN*, 9 (10) 0975-3710.

Nielsen, M. A. (2015). *Neural networks and deep learning.* San Francisco, CA: Determination press, (25).

Pal, V. (2019). *Price forecasting of brinjal and chilli (green) – a statistical evaluation*. (Unpublished master's thesis), B. A. College of Agriculture Anand Agricultural University, Anand, Gujarat, India.

Pant, N. (2017). A guide for time series prediction using recurrent neural networks (LSTMS). *Stats and Bots.*

Paul, R. K. & Himadri, G. (2009). GARCH nonlinear time series analysis for modelling and forecasting of India's volatile spices export data. *Journal of the Indian Society of Agricultural Statistics*, *63* (2), 123-131.

Ravichandra, N. G. (2014). Horticulture and its role in the national economies. *Horticultural Nematology*, 1-3.

Ray, C. F. (1974). An evaluation of a short-run forecasting model. *International Economic Review*, 15 (2), 285-303.

Reimers, N. & Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. 1707.06799.

Sabu, K. M. & Kumar, T. M. (2020). Predictive analytics in agriculture: Forecasting prices of arecanuts in Kerala. *Procedia Computer Science, 171*, 699-708.

Salman, N., Lawi, A. & Syarif, S. (2018, November). Artificial neural network back propagation with particle swarm optimization for crude palm oil price prediction. *Journal of Physics: Conference Series*, 1114 (1), 012088.

Samal, G. P. (2017). Price discovery efficiency of cotton futures market in India. *Agricultural Economics Research Review, 30* (2), 235-244.

Sharma, S. (2017). Activation functions in neural networks. *Towards Data Science, 6* (12), 310-316.

Shruthi, J. (2015). An analysis of price forecasting techniques for onion and tomato crops. (Unpublished master's thesis). University of Agricultural Sciences, Dharwad, Karnataka (India).

Slutzky, E. (1937). The summation of random causes as the source of cyclic processes. *Econometrica: Journal of the Econometric Society*, 5 (2), 105-146.

Stančin, I., & Jović, A. (2019). An overview and comparison of free Python libraries for data mining and big data analysis. *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 977-982.

Subhasree, M. & Priya, C. A. (2016). Forecasting vegetable price using time series data. *International Journal of Advanced Research*, *3*, 535-641.

Weng, Y., Wang, X., Hua, J., Wang, H., Kang, M. & Wang, F. Y. (2019). Forecasting horticultural products price using ARIMA model and neural network based on a large-scale data set collected by web crawler. *IEEE Transactions on Computational Social Systems*, 6 (3), 547-553.

Wold, H. (1938). *A study in the analysis of stationary time series* (Doctoral dissertation, Almqvist & Wiksell).

Yin, R. (1999). Forecasting short-term timber prices with univariate ARIMA models. *Southern Journal of Applied Forestry*, *23* (1), 53-58.

Yule, G. U. (1926). Why do we sometimes get nonsense-correlations between Time-Series? - a study in sampling and the nature of time-series. *Journal of the royal statistical society*, *89* (1), 1-63.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International journal of forecasting*, *14* (1), 35-62.

Websites referred

- http://agmarknet.gov.in/
- http://assets.fsnforumhlpe.fao.org.s3-eu-west-1.amazonaws.com/public/files/Food_losses_waste/India%20banana%20case_final.pdf
- https://doh.gujarat.gov.in/Images/directorofhorticulture/pdf/statistics/Area-Production-2019-20.pdf
- https://www.numpy.org/
- https://www.tensorflow.org/
- https://agmarknet.gov.in/
- http://nhb.gov.in/
- http://www.fao.org/faostat/en/#search/banana
- https://pib.gov.in/PressReleasePage.aspx?PRID=1717447

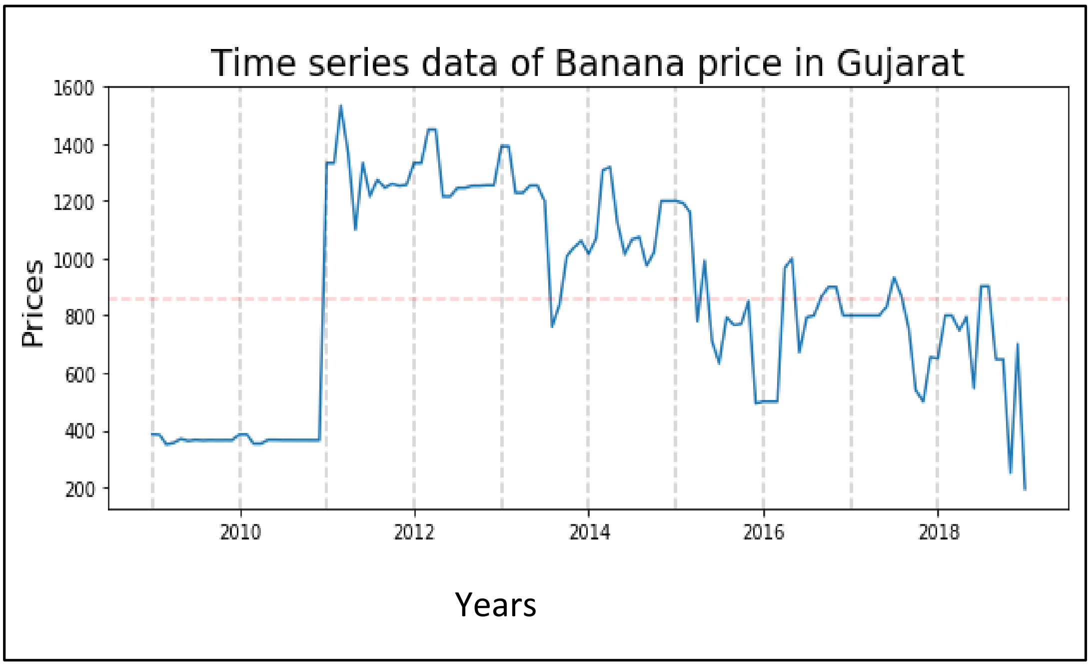

Fig. 4.1: Time series data for banana price in Rajpipla market, Narmada, Gujarat from January 2009 to December 2019

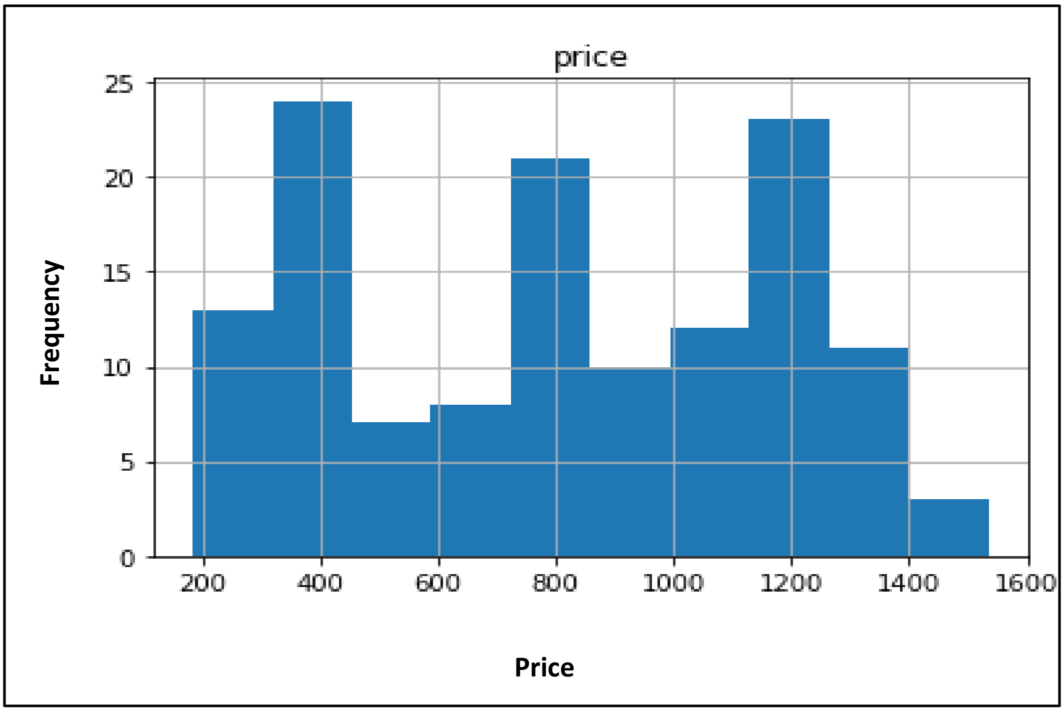

Fig. 4.2: Histogram of time series data for banana price in Gujarat from January 2009 to December 2019

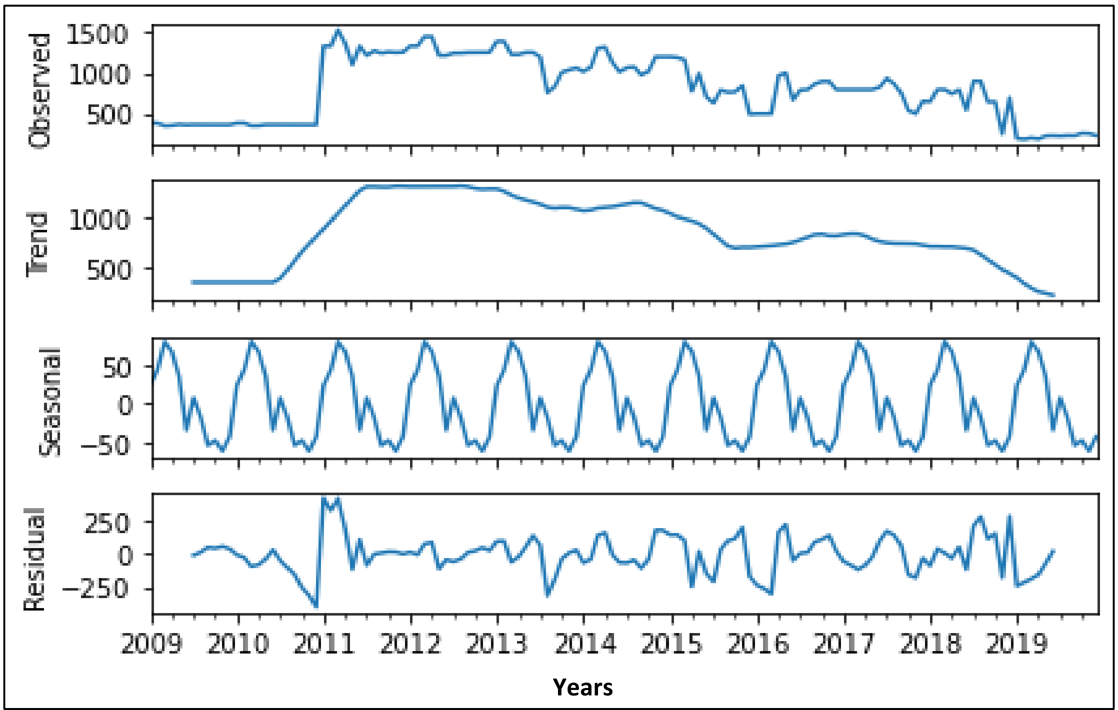

Fig. 4.3: Seasonal decomposition of time series data for banana price in Gujarat

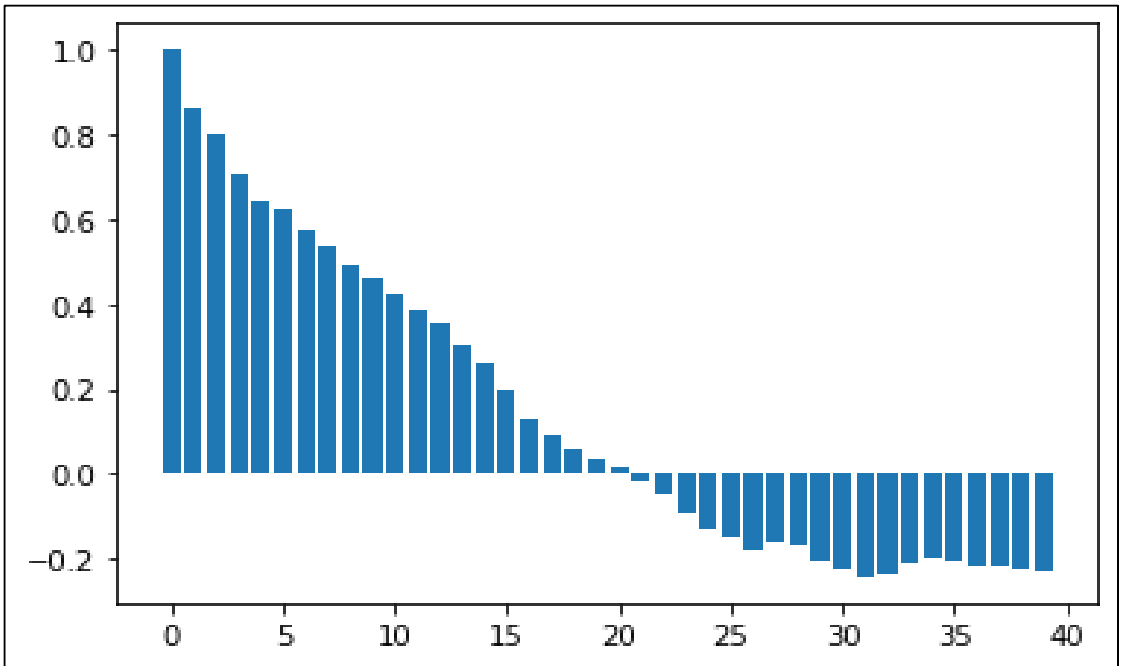

Fig. 4.4: ACF lags of banana price of Rajpipla market

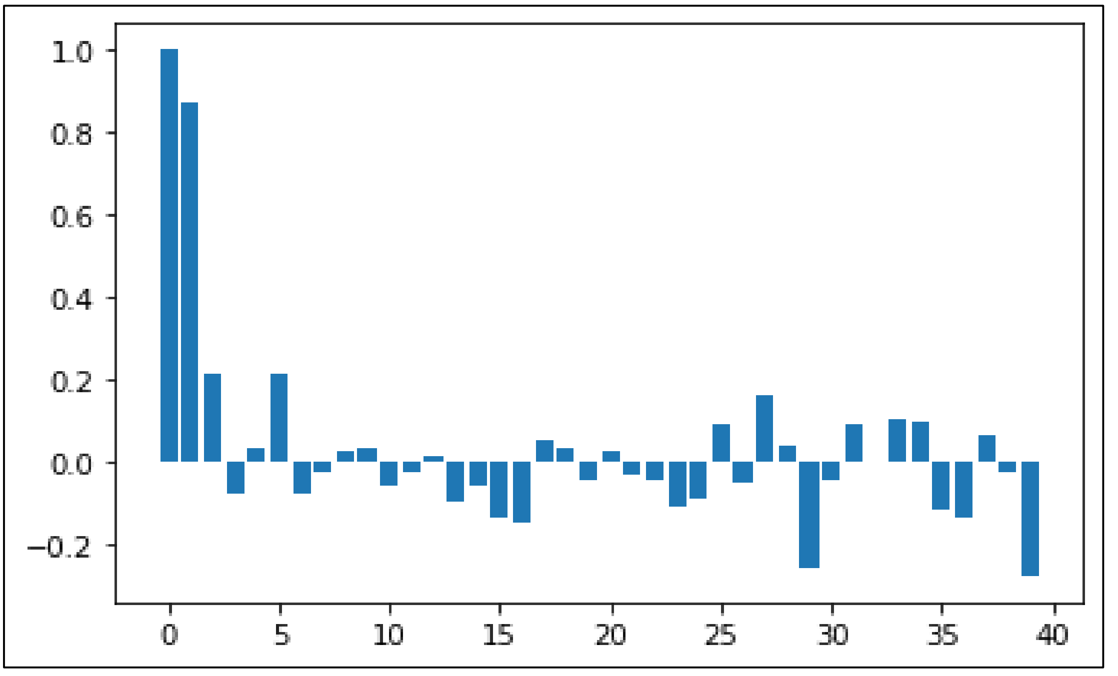

Fig. 4.5: PACF lags of banana price of Rajpipla market

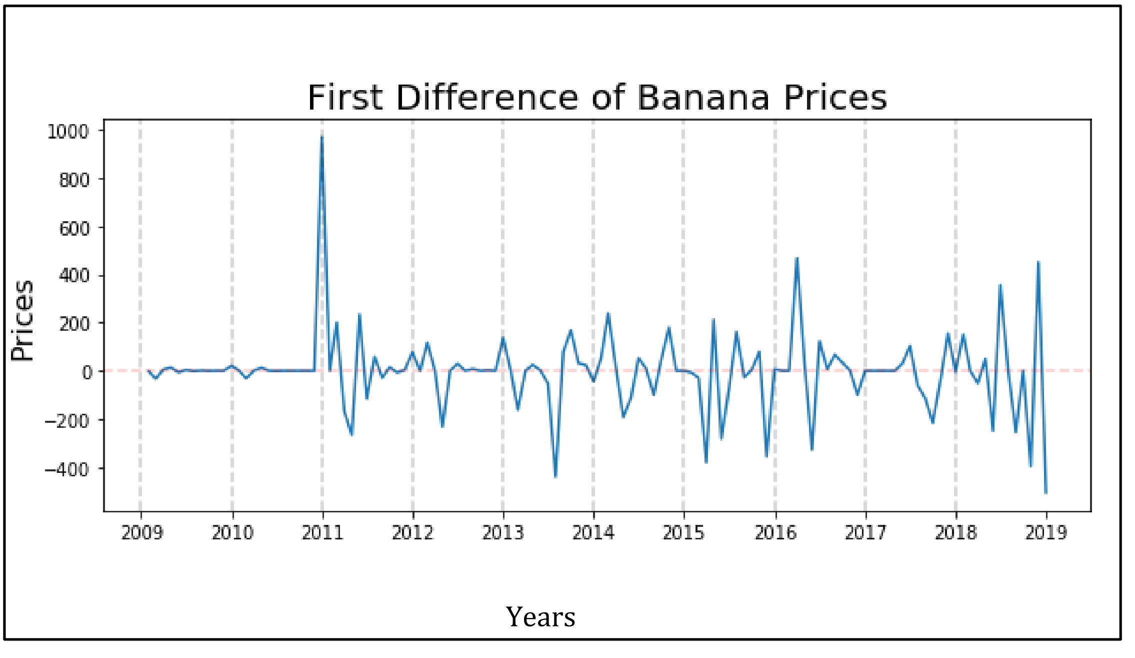

Fig. 4.6: Time series data for banana price after first differencing

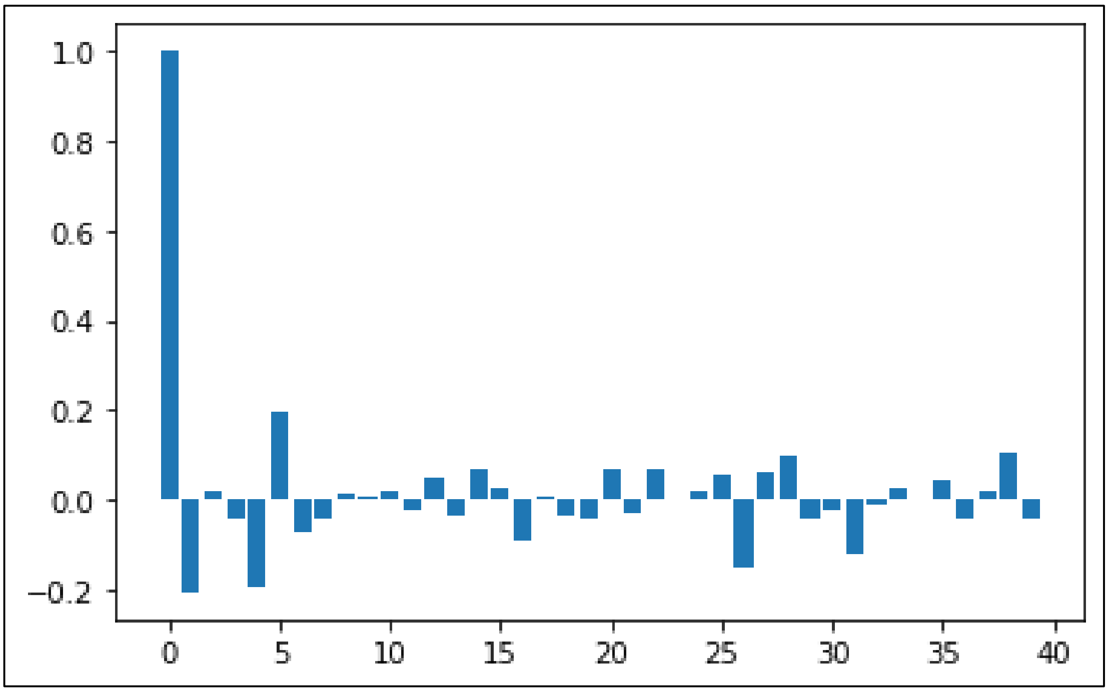

Fig. 4.7: ACF of time series data for banana price after first differencing

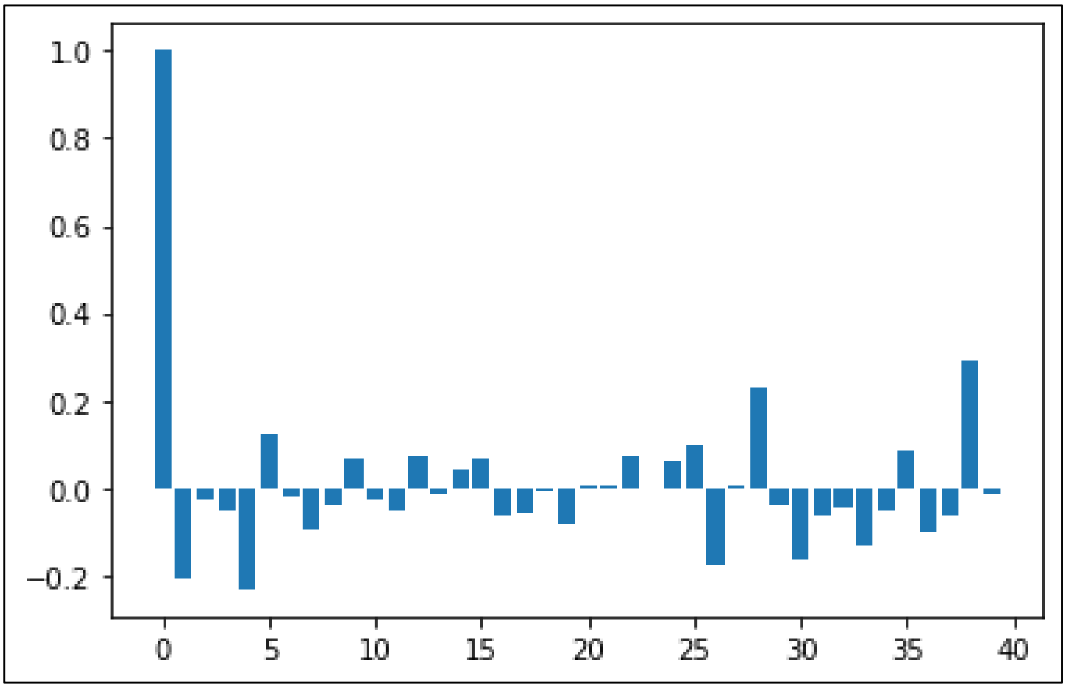

Fig. 4.8: PACF of time series data for banana price after first differencing

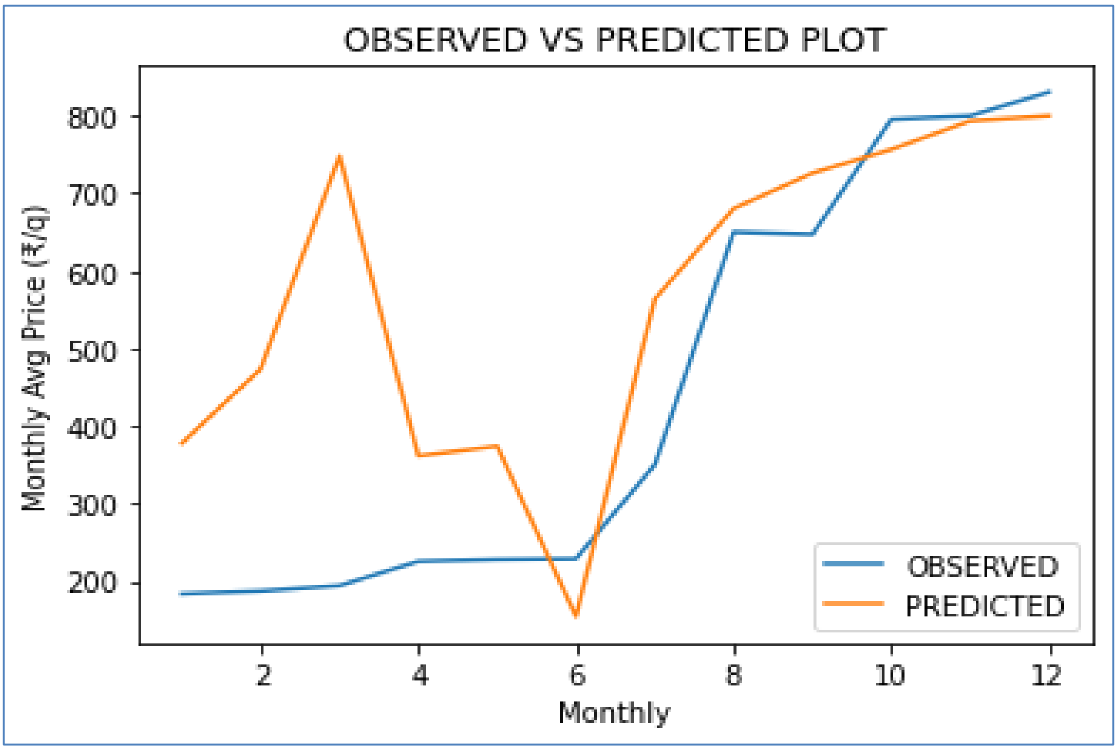

Fig. 4.9: Observed vs Predicted graph for ARIMA model

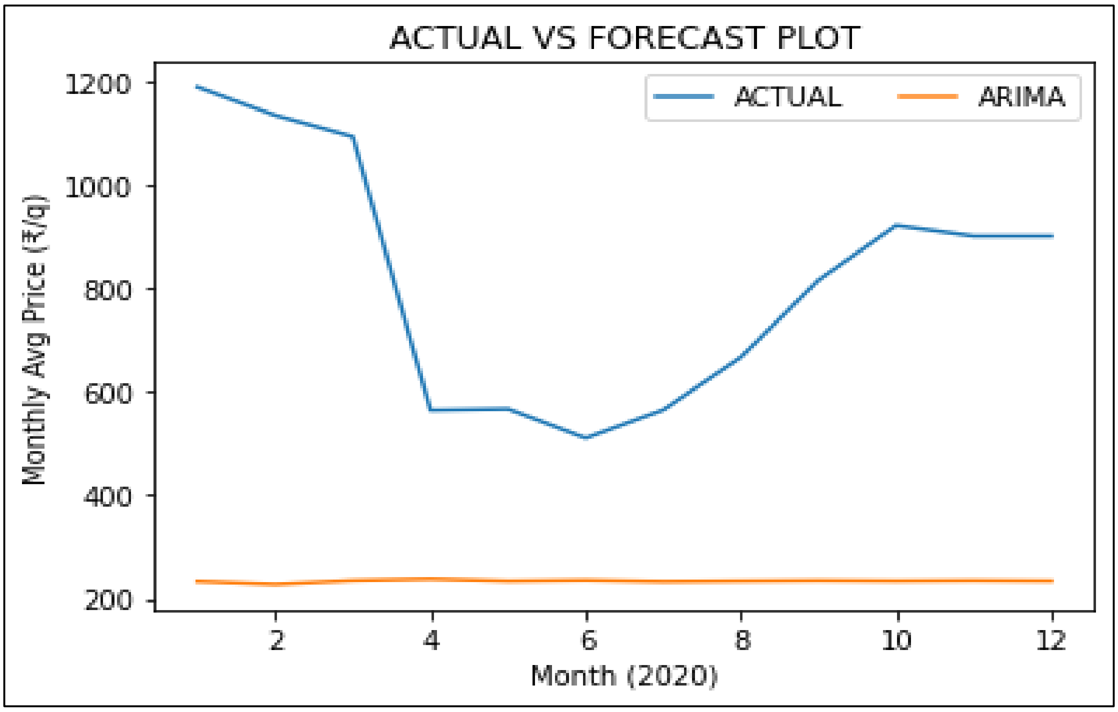

Fig. 4.10: Actual vs Forecast graph for ARIMA model

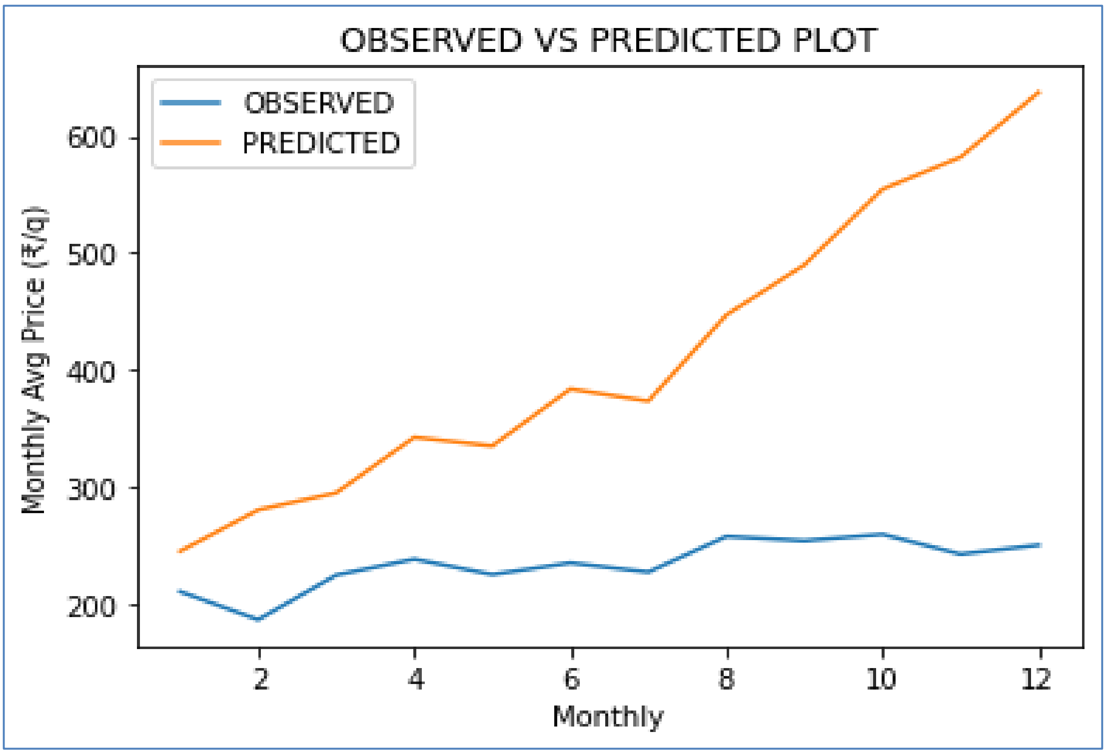

Fig. 4.11: Observed vs Predicted graph for SARIMA model

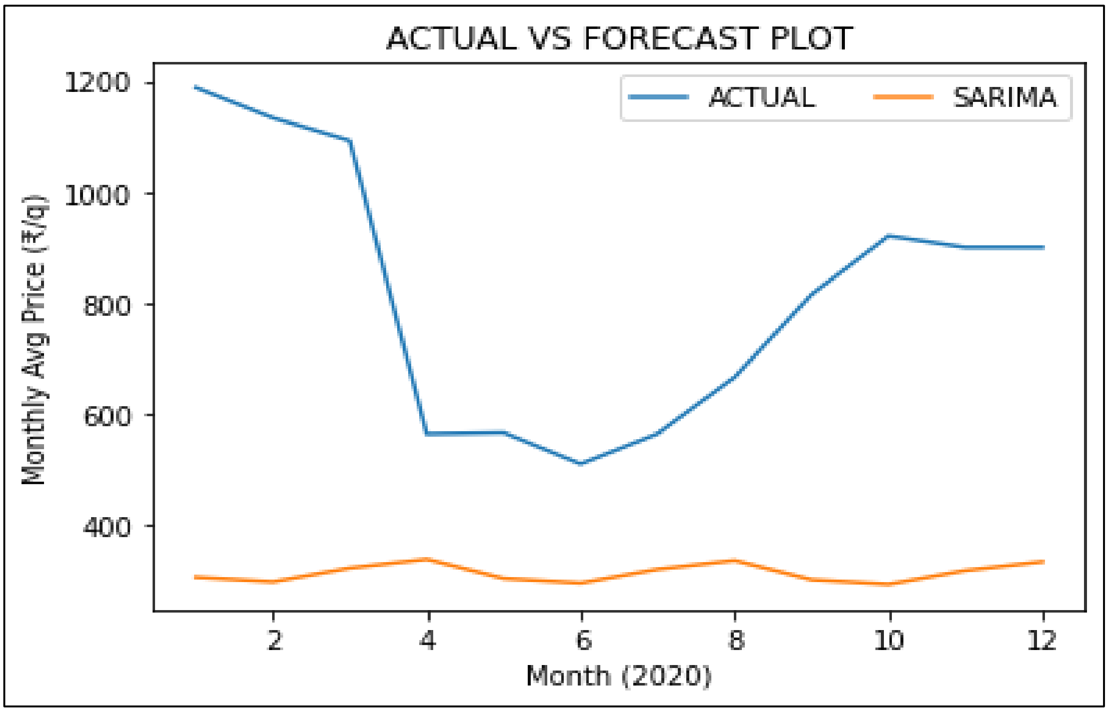

Fig. 4.12: Actual vs Forecast graph for SARIMA model

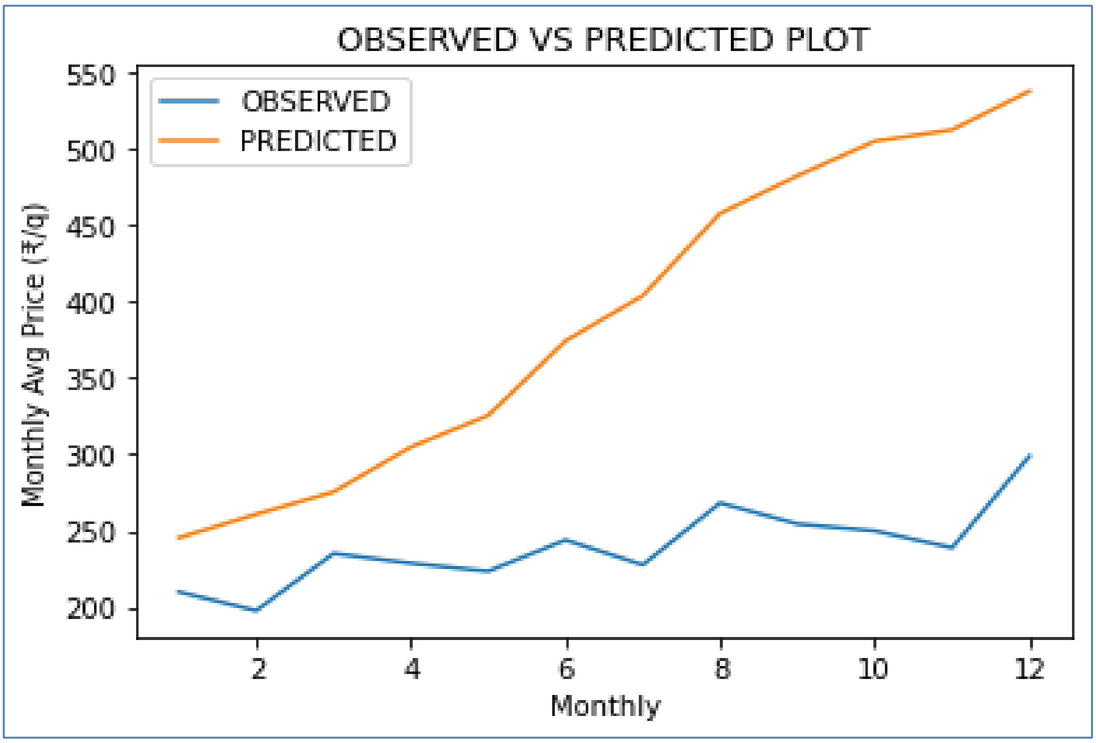

Fig. 4.13: Observed vs Predicted graph for ARCH/GARCH model

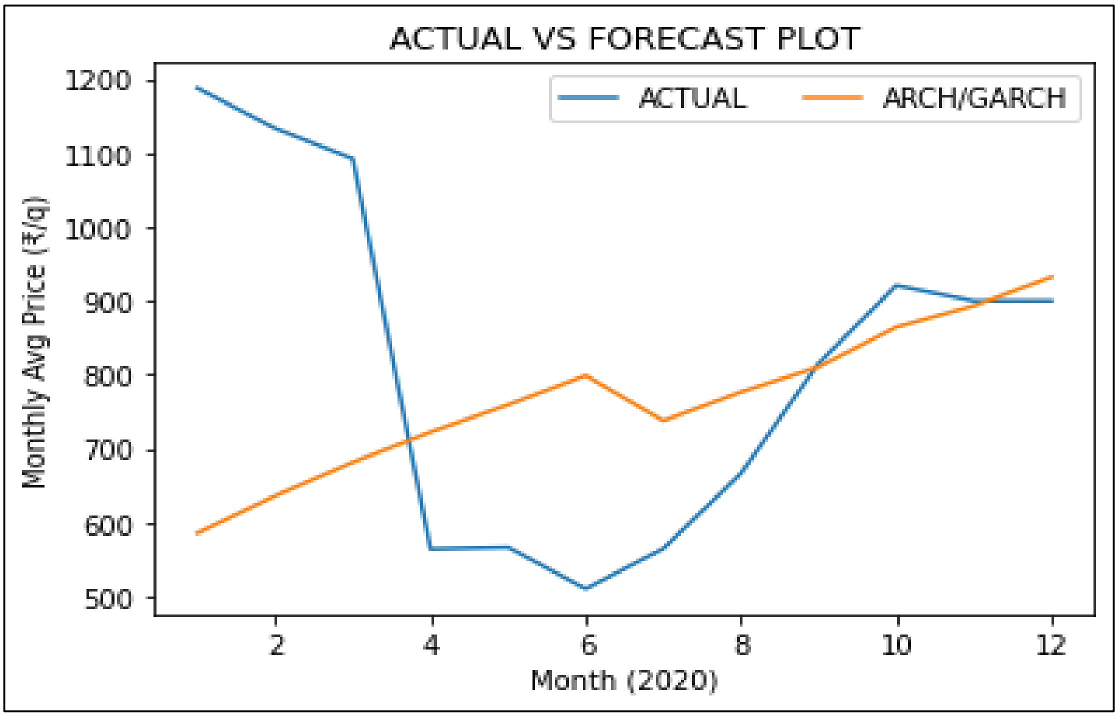

Fig. 4.14: Actual vs Forecast graph for ARCH/GARCH model

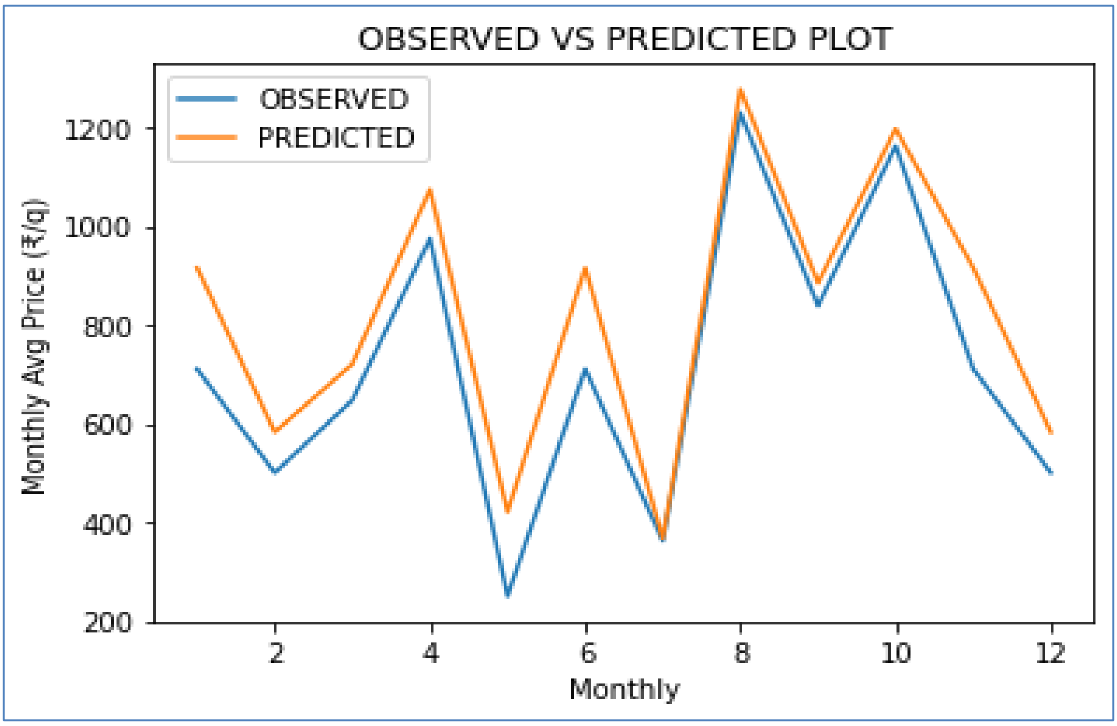

Fig. 4.15: Observed vs Predicted graph for ANN model

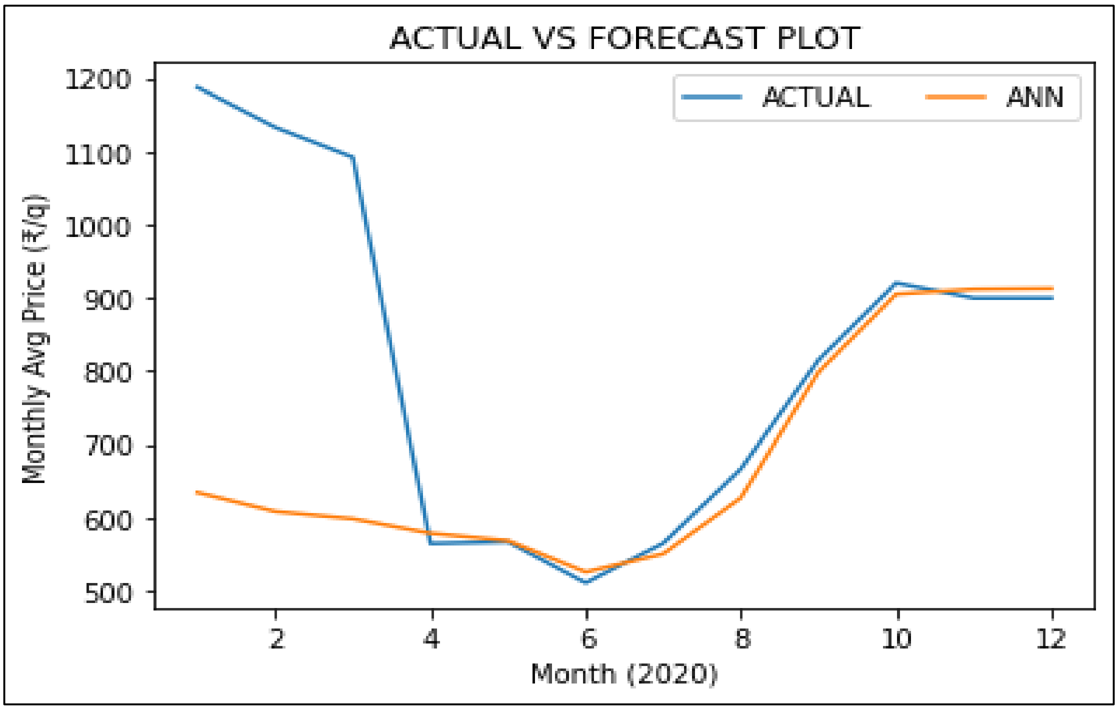

Fig. 4.16: Actual vs Forecast graph for ANN model

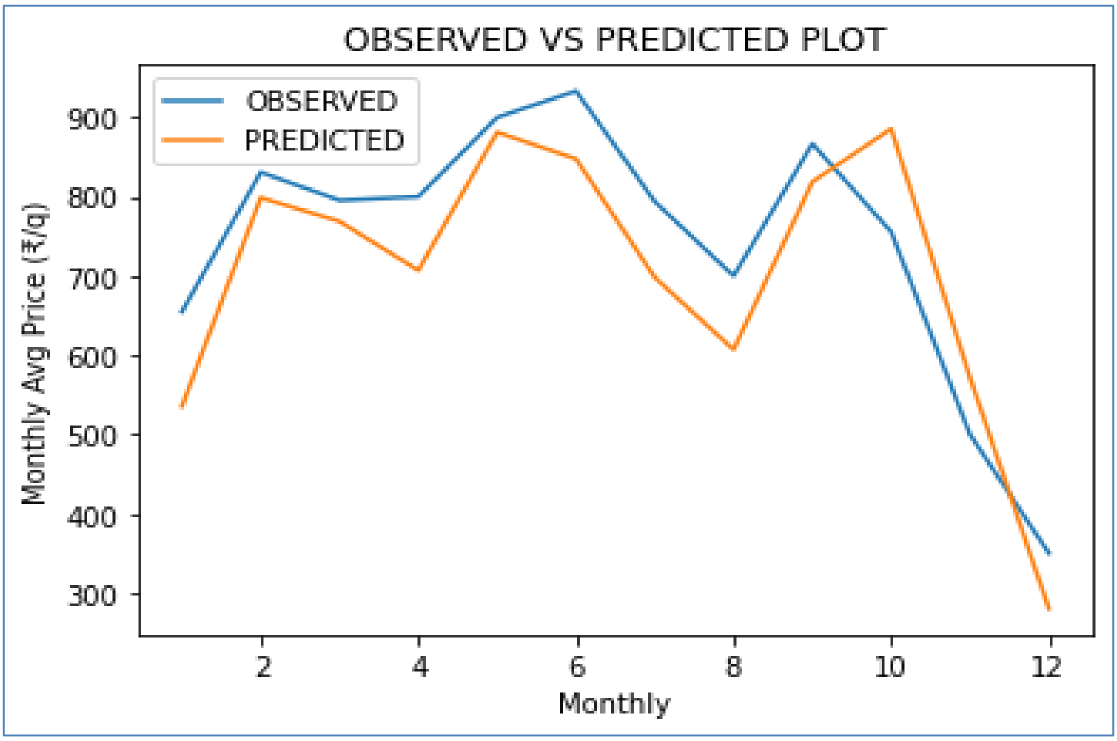

Fig. 4.17: Observed vs Predicted graph for RNN model

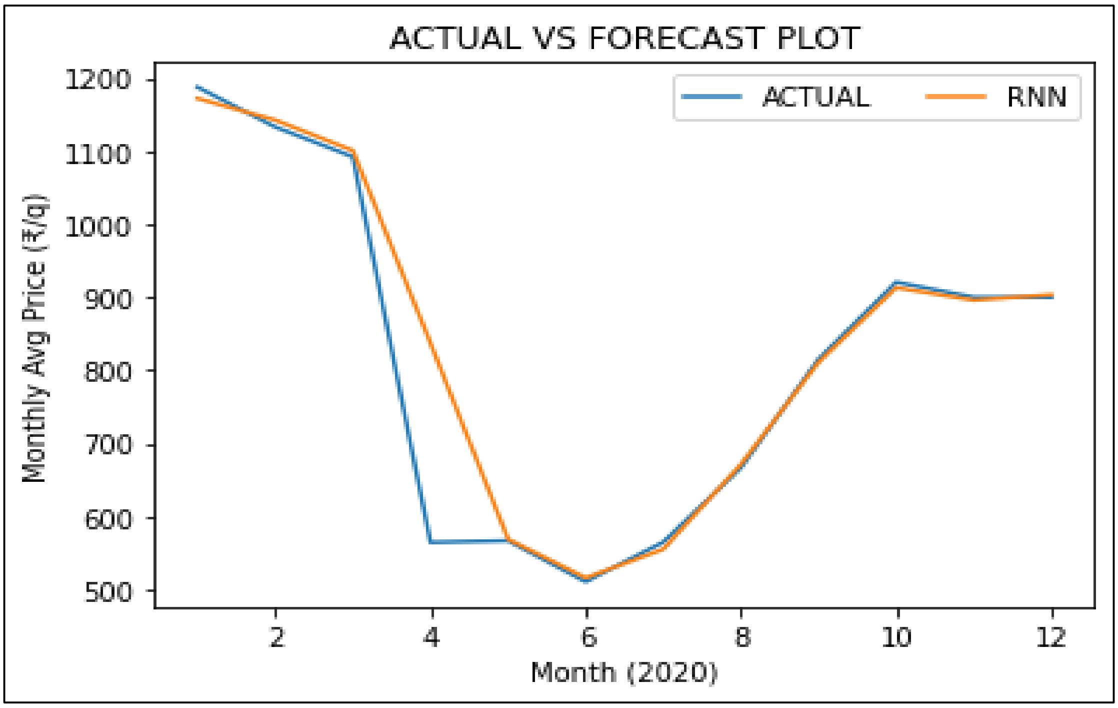

Fig. 4.18: Actual vs Forecast graph for RNN model

Curves for Activation function

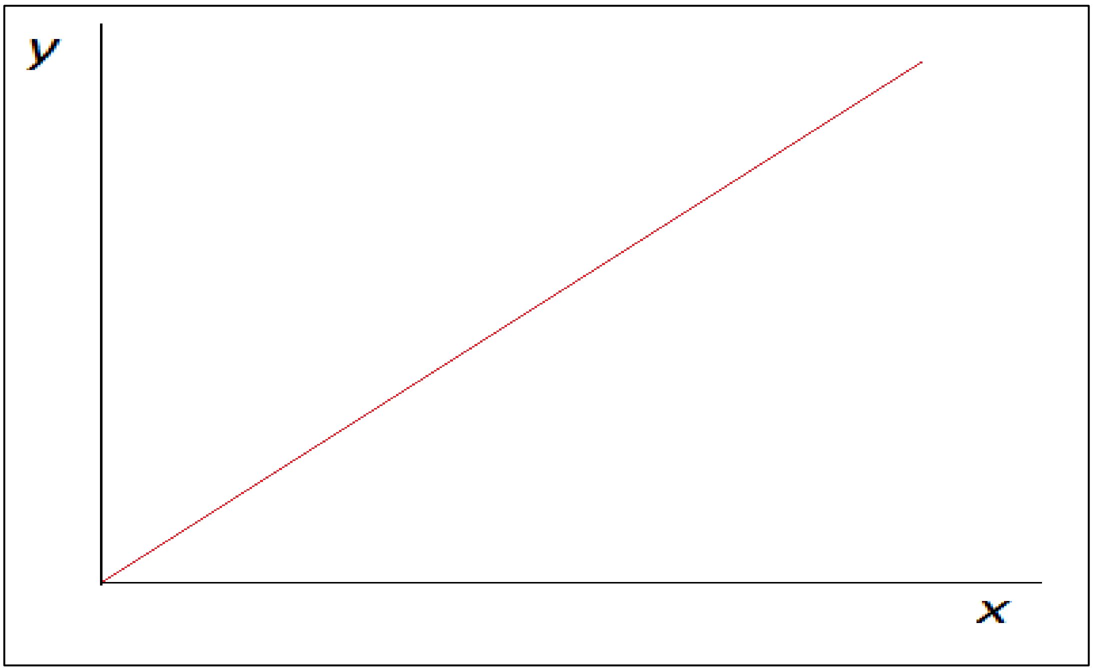

Fig. 3.2: Linear/Identity function

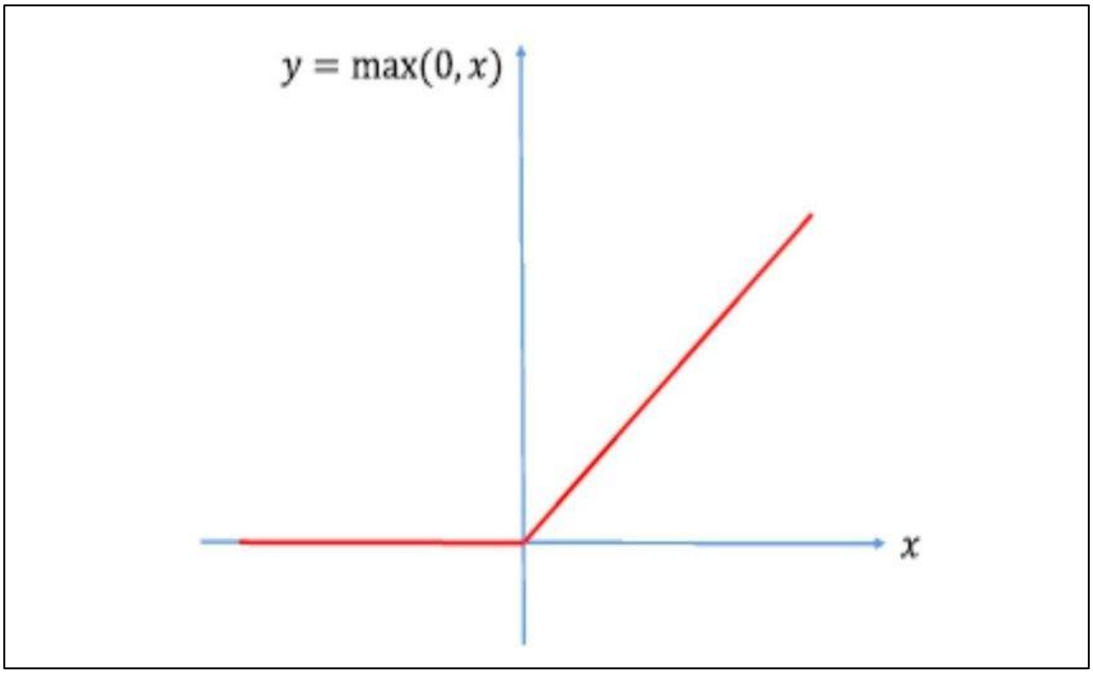

Fig. 3.3: ReLU function

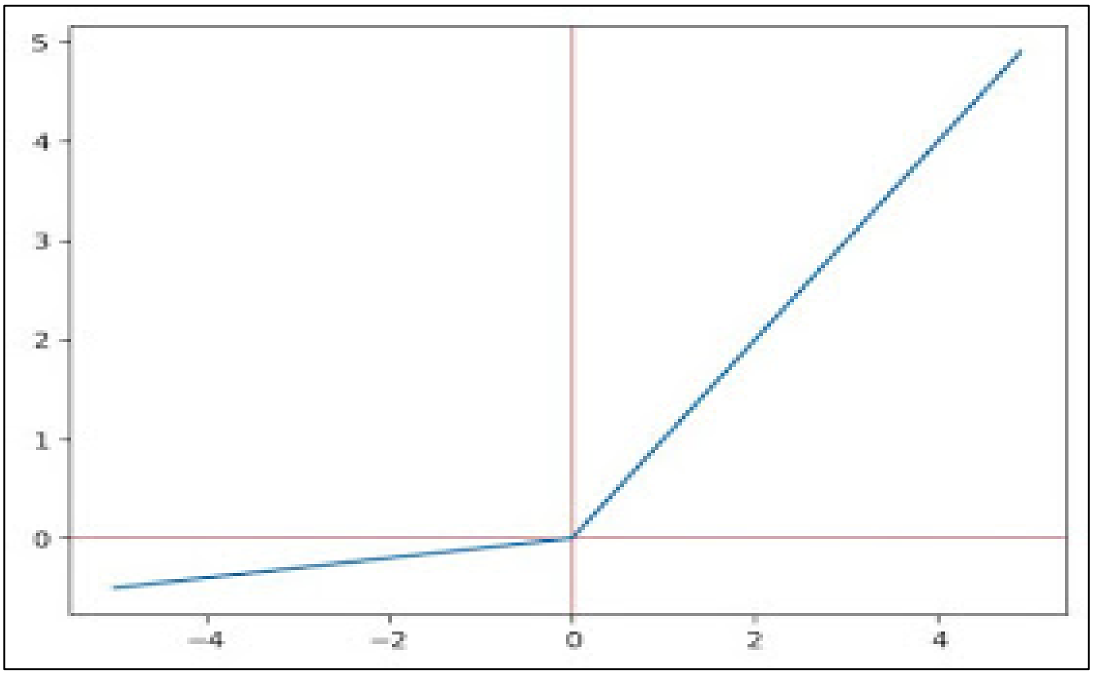

Fig. 3.4: Leaky ReLU function

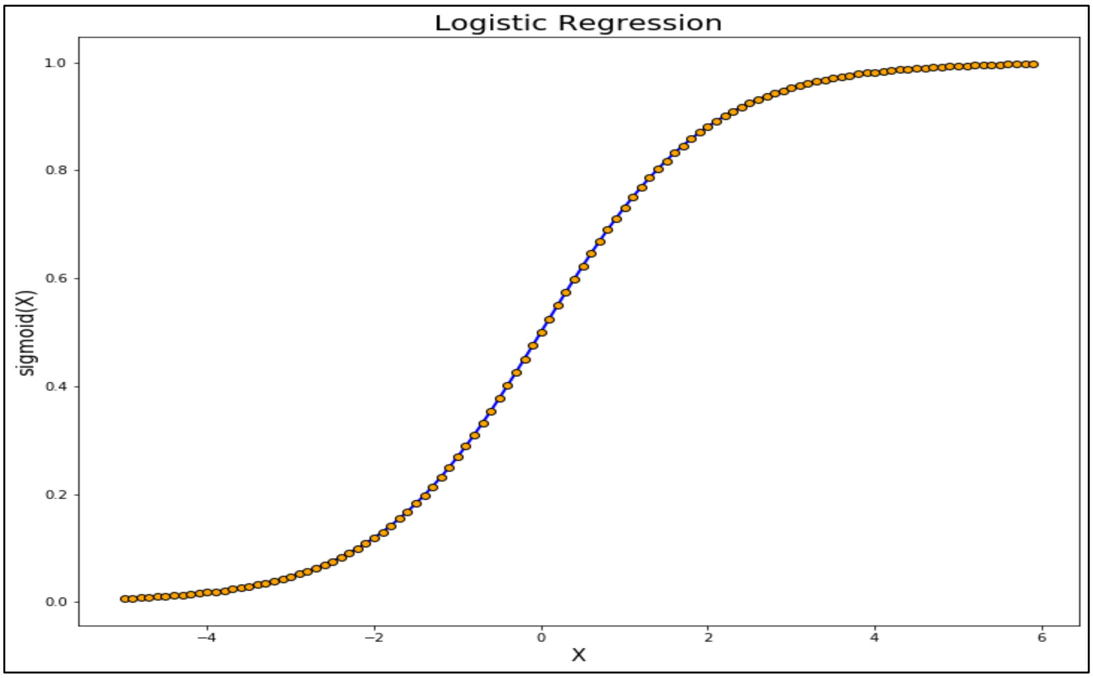

Fig. 3.5: Sigmoid function

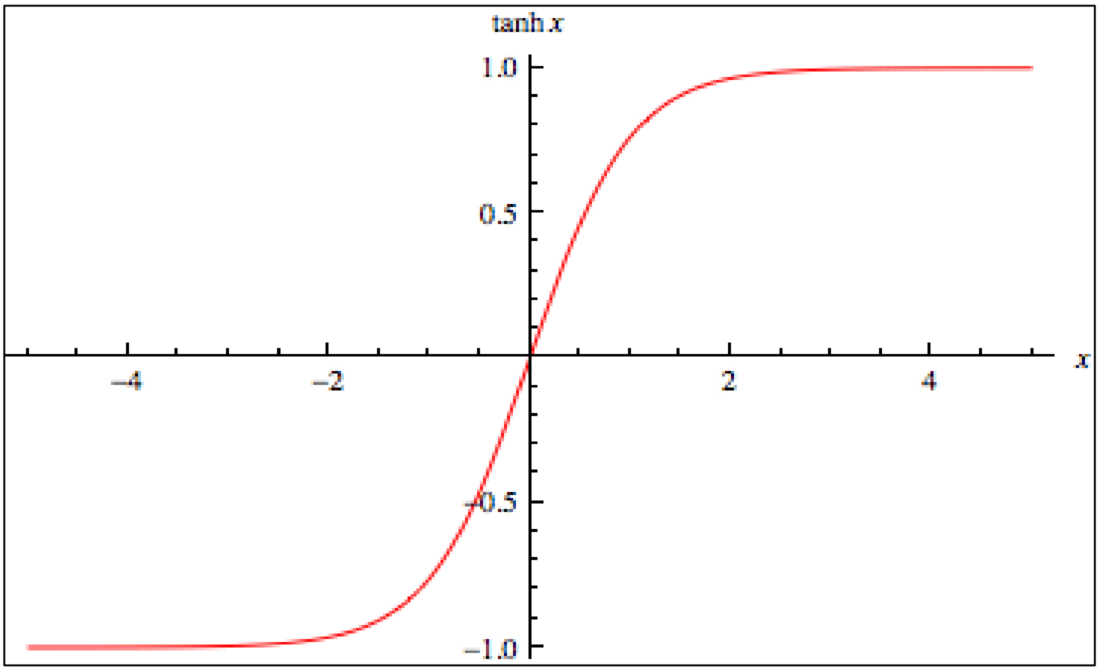

Fig. 3.6: TanH function

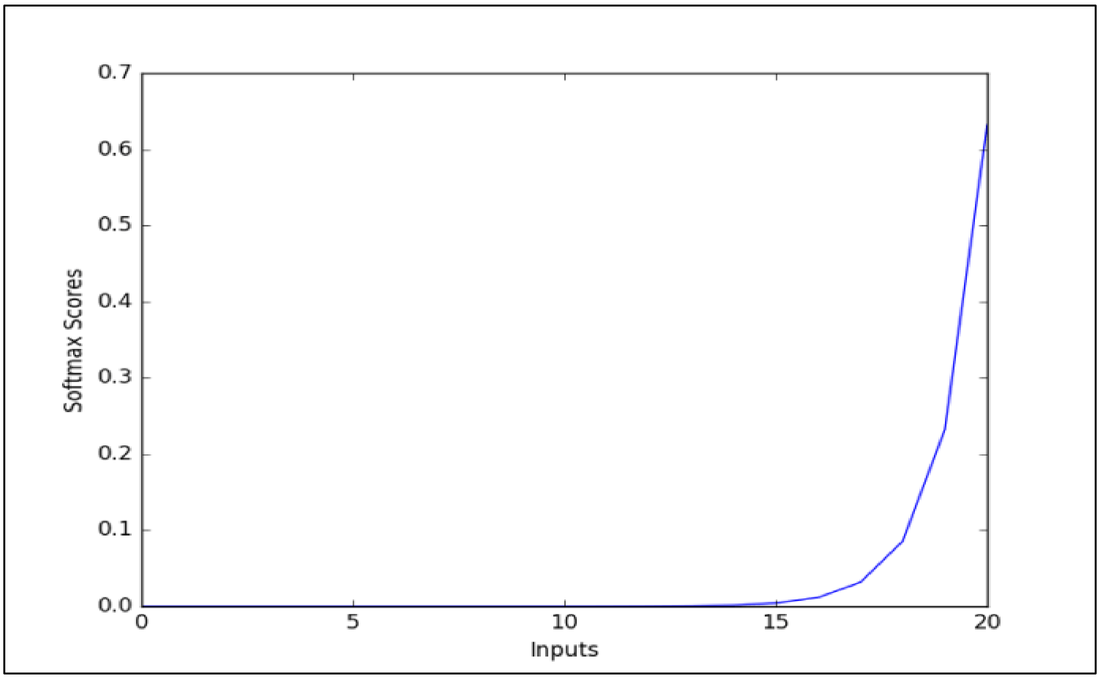

Fig. 3.7: Softmax function

CERTIFICATE

This is to certify that I have no objection for supplying to any scientist one copy of any part of this thesis for rendering reference service in a library of documentation centre.

Place: Anand

Date: / 09 /2020 **(Viniya Goswami)**