कृषि में पूर्वानुमान के लिए मशीन-लर्निंग तकनीकों पर आधारित हाइब्रिड मॉडल का अध्ययन

STUDY ON MACHINE LEARNING TECHNIQUES BASED HYBRID MODEL FOR FORECASTING IN AGRICULTURE

PANKAJ DAS

Doctor of Philosophy

in

Agricultural Statistics

ICAR-Indian Agricultural Statistics Research Institute
ICAR-Indian Agricultural Research Institute
New Delhi- 110012
2019

STUDY ON MACHINE LEARNING TECHNIQUES BASED HYBRID MODEL FOR FORECASTING IN AGRICULTURE

By

Pankaj Das

Thesis submitted to the Faculty of Post-Graduate School,
ICAR-Indian Agricultural Research Institute, New Delhi,
in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

IN

AGRICULTURAL STATISTICS

ICAR-Indian Agricultural Statistics Research Institute
ICAR-Indian Agricultural Research Institute
New Delhi - 110012

Approved by:

Chairman: .....
(Dr. Girish Kumar Jha)

Co-chairman: .....
(Dr. Rajender Parsad)

Member: .....
(Dr. Achal Lama)

Member: .....
(Dr. K. K. Chaturvedi)

Member: .....
(Dr. Dwijesh Chandra Mishra)

Division of Agricultural Economics
ICAR-Indian Agricultural Research Institute
PUSA, New Delhi - 110012

Dr. Girish Kumar Jha
Principal Scientist (Agricultural Statistics)

CERTIFICATE

This is to certify that the work incorporated in the thesis entitled **"Study on Machine Learning Techniques based Hybrid Model for Forecasting in Agriculture"** submitted in partial fulfilment of the requirement for the degree of **Doctor of Philosophy** in **Agricultural Statistics** of the **Post-Graduate School, ICAR-Indian Agricultural Research Institute, New Delhi,** is a record of bonafide research carried out by **Mr. Pankaj Das** under my guidance and supervision and no part of this dissertation has been submitted for any other degree or diploma.

All assistance and help received during the course of this investigation has been duly acknowledged.

Place: New Delhi

Date: 6/2/2020

गिरीश झा

(Dr. Girish Kumar Jha)
Chairman, Advisory Committee

Acknowledgements

This thesis came to fruition due to the knowledge gained over the entire period of my Ph.D. study at ICAR–Indian Agricultural Statistics Research Institute (IASRI), New Delhi during which, I have been in touch with a great number of people whose contribution in varied yet myriad ways has led to the research and making of the thesis which deserve special mention. It is a pleasure to convey my gratitude to all of them by way of my humble acknowledgements.

First and foremost, with reverence, I want to express my deepest sense of gratitude to ***Dr. Girish Kumar Jha****, Principal Scientist, Division of Agricultural Economics, ICAR-Indian Agricultural Research Institute, New Delhi and Chairman of my advisory committee for his initiative, benevolence, endurance, constructive criticism and constant monitoring during the period of my investigation and also in the preparation of this thesis. Above all and most needed, he provided me constant support and encouragement in various ways. I consider myself fortunate in having the privilege of being guided by him. I am really indebted to him.*

I am equally indebted to ***Dr. Rajender Parsad****, Principal Scientist, Division of Design of Experiments, ICAR-IASRI, New Delhi and Co-chairman of my Advisory Committee for his counsel, valuable suggestions and ready to help for accomplishment of my research work.*

I am extremely thankful to ***Dr. Achal Lama****, Scientist, Division of Forecasting and Agricultural Systems Modelling, ICAR-IASRI, New Delhi and member of my Advisory Committee for his useful suggestions and immense support during my research work*

I am also highly grateful to ***Dr. K. K. Chaturvedi****, Scientist, Centre for Agricultural Bioinformatics, ICAR-IASRI, New Delhi and member of my Advisory Committee for his useful suggestion and constant support during my thesis work.*

I am extremely thankful to ***Dr. Dwijesh Chandra Mishra****, Scientist, Centre for Agricultural Bioinformatics, ICAR-IASRI, New Delhi and member of my Advisory Committee for his useful suggestions during my investigation.*

I am also highly grateful to ***Dr. Bishal Gurung****, Scientist, Division of Forecasting and Agricultural Systems Modelling, ICAR-IASRI, New Delhi and member of my Advisory Committee for his useful suggestion and constant support during my thesis work.*

I am also highly grateful to ***Dr. Anshu Bharadwaj****, Principal Scientist, Computer Applications, ICAR-IASRI, New Delhi and member of my Advisory Committee for her useful suggestion and constant support during my thesis work.*

I am also extremely thankful to ***Rajeev Ranjan Kumar Sir****, Scientist, Division of Forecasting and Agricultural Systems Modelling, ICAR-IASRI, New Delhi for his useful suggestions and immense support during my research work.*

I want to express my respect and gratitude to ***my parents, brothers, sisters and all the family members*** *for their sacrifices they did for me. These were the inspirations and moral boosting which gave me sufficient energy to complete this thesis in time.*

I am thankful to my classmate ***Murari, Nitin, Shyam, Sumeet, Rahul, Asif, Animesh, Srikant, Sonober, Somanna, Sushil, Rajeev, Sashank, Priyanka and Arfa*** *for their friendly approach and moral support throughout my entire Ph.D. tenure.*

I do not have any words to express the help and guide given by my seniors specially, ***Chandan Sir, Chiranjit Sir, Himadri Sir, Shwetank Sir, Saurav Sir, Sadikul Sir, Pramod Sir, Harun Sir*** *and my juniors* ***Nobin, Kuldeep, Gopal, Satpati, Amit, Ashraful, Dipankar, Samir, Yeasin, Arpan, Asish, Akhilesh, Ronit, Dilip, Ashit, Sandipan, Kapil, Jitendra, Dipro, Vaiju, Mahalinga, Krishna, Rahul, Vaibhav, Rohit, Nitesh, Amit, Vinayaka, Abhishek, Debopam, Naveen, Vinay, Abhishek and all others*** *for their affectionate support and help.*

I am also thankful to all my seniors, juniors and friends in IARI for their moral support and inspirations.

I take this opportunity to appreciate the help rendered by the staff of PG school and TAC. Special thanks are due to ***Sanjeev ji, Gagan ji,*** *and* ***Sunil ji.***

I extend my thanks to the Director, the Dean and the Associate Dean, ICAR-IARI, New Delhi and the Director, ICAR-IASRI for their helpful attitude and cooperation, throughout the period of study.

I am thankful to ***ICAR*** *and* ***Ministry of Human Resource Development (MHRD)*** *for the financial assistance provided to me in the form of* ***Rajiv Gandhi National Fellowship (RGNF)*** *during the tenure of my study.*

Finally, I would like to thank everybody who was imperative to the successful realization of this thesis, as well as articulate my apology that I could not mention personally one by one.

Date: 06/02/2020

ICAR-IASRI, New Delhi-110012

Pankaj Das

(Pankaj Das)

Contents

Abbreviations

ADF- Augmented-Dickey-Fuller

ANN- Artificial Neural Network

AR- Autoregressive

ARCH- Autoregressive Conditional Heteroscedasticity

ARIMA- Autoregressive Integrated Moving Average

ARMA- Autoregressive Moving Average

BDS- Brock-Dechert-Scheinkman

BF- Basis Function

BP- Backpropagation

DM- Diebold-Mariano

ECT- Error Correction Term

EEMD- Ensemble Empirical Mode Decomposition

EMD- Empirical Mode Decomposition

GARCH- Generalized Autoregressive Conditional Heteroscedasticity

GCV- Generalized Cross Validation

GRSq- General R Square

IMF- Intrinsic Mode Function

LSSVM- Least Squares Support Vector Machine

LSSVR- Least Squares Support Vector Regression

MA- Moving Average

MAD- Mean Absolute Deviation

MAE- Mean Absolute Error

MAPE- Mean Absolute Percentage Error

MARS- Multivariate Adaptive Regression Spline

ME- Maximum Error

MGARCH- Multivariate Generalized Autoregressive Conditional Heteroscedasticity

MLP- Multi Layer Perceptron

MLR- Multiple Linear Regression

NN- Neural Network

PP- Phillips-Perron

RBF- Radial Basis Function

REP- Replication

RMSE- Root Means Square Error

RPROP- Resilient Backpropagation

RSq- R square

SVM- Support Vector Machine

SVR- Support Vector Regression

TDNN- Time Delay Neural Network

VAR- Vector Autoregressive

CHAPTER I

INTRODUCTION

1.1 Background

Forecasting is primarily used to predict the future trend or pattern of data. It has wide domain of applications in various fields like weather forecasting, economic forecasting, energy forecasting, transport forecasting, sales forecasting, technology forecasting, price forecasting and crop yield forecasting *etc.* In India, around 60-70% of total population directly or indirectly depend on agriculture and its allied sectors. Agriculture contributes around 15-16% to Indian Gross Domestic Product (GDP). The contribution of agriculture and its allied sectors to the Indian economy is inescapable. Therefore, it is very important to predict/forecast the price and yield of agricultural commodities.

The agricultural market environment is changing with unprecedented speed both locally and globally. The dynamic nature of market affects farm prices and thereby farm income. Most of the rural farmers are unable to understand and interpret the market and price behaviour to their advantages (Anjaly *et al*., 2010). Thus, market information and intelligence are crucial to enable the farmers and traders in making important decisions about what to grow, when to sell, and where to sell. Besides this, the price instability and uncertainty pose a restriction on decision and policy makers. Hence, agricultural price forecasting plays a vital role for both production and market strategy. Price forecasting enables a match between the supply and demand of the commodity. Price forecasting of an agricultural commodity is a herculean task because it depends on too many factors which cannot be accurately predicted. Nonlinear and nonstationary behaviour are the crucial problems in agricultural price data. Agricultural commodity prices are also volatile in nature due to seasonality, inelastic demand, production uncertainty and also because many agricultural commodities are perishable in nature.

The crop yield is affected by several factors such as physical, economical and technological. Yield modelling are done to improve management techniques and boost actual yields. Improved forecasting models highlight the medium and long term effects of climate change on crop yields. A good forecasting model should be reliable, consistent,

object-oriented, cost effective and sensitive to extreme events (Seid, 2016). In literature, there exist many statistical models as well as artificial intelligence models for agricultural price forecasting and yield prediction.

1.2 Time series models

A data set containing sequence of observations on a single phenomenon observed over time is called time series data. Time series forecasting is an important area of forecasting in which past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. Main goals of time series analysis is to identify the data generating mechanism and forecasting *i.e.* predicting future values of the time series variable. Time series models can be classified into different categories. A general overview of different time series models is given in figure 1.1.

1.2.A Statistical time series model

Statistical time series models are based on probability distribution. In the 19^{th} century the model development was based on deterministic concept. Yule (1927) first introduced the concept of stochasticity in time series. This concept of stochasticity led to the development of statistical models in time series. Statistical time series model can be categorized into two types based on the number of variables involved in the model. They are univariate time series model and multivariate time series model.

1.2.A.1 Univariate time series models

Univariate time series model deals with a set of values of a time series x_t over a time period t *(t = 1,2, ... , n).* Some important univariate time series models are Autoregressive (AR) model, Moving Average (MA) model, Autoregressive Moving Average (ARMA) model, Autoregressive Integrated Moving Average (ARIMA) model, Autoregressive Conditional Heteroskedasticity (ARCH) model, Generalized Autoregressive Conditional

Heteroscedasticity (GARCH) model *etc*. These models are linear in nature. But in real

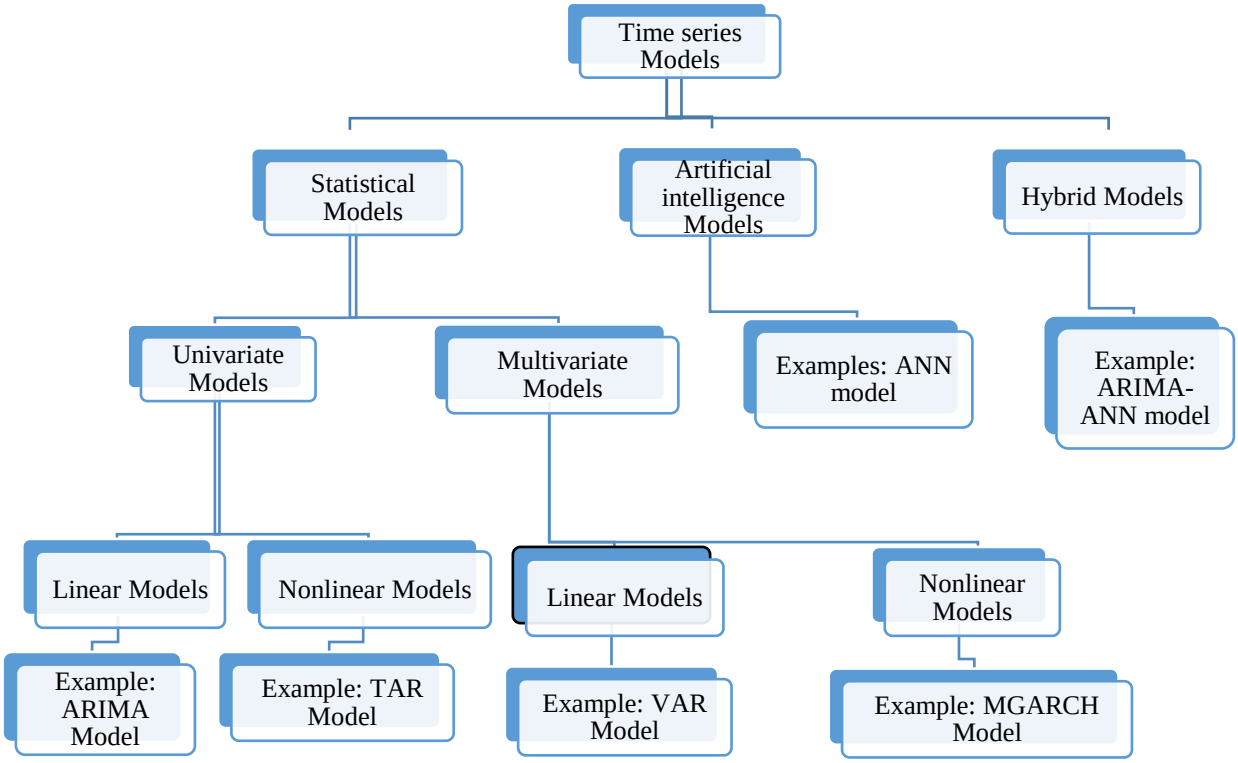


Figure 1.1: An overview of Time series models

world the data are more complex and nonlinear in nature. Based on nonlinearity, several nonlinear time series models were developed. The concept of nonlinear time series model was first introduced by Volterra (1930). Some nonlinear models are Bilinear model, Threshold Autoregressive (TAR) Models and self-exciting TAR (SETAR) model.

1.2.A.2 Multivariate time series models

Multivariate time series models can examine more than one variables simultaneously. Besides the self-dependency, the variables may depend on other variables in multivariate time series model. The multivariate representation of ARIMA or Vector ARIMA (VARIMA) was derived by Quenouille (1957). The assumptions of multivariate models are based on exogeneity and interrelation among variable. Some important multivariate models are:

- Vector Autoregressive (VAR) models
- Multivariate Generalized Autoregressive Conditional Heteroscedasticity (MGARCH) model

1.2.B Artificial Intelligence models

Artificial Intelligence is a branch of applied science that improves the performance of a machine or model by providing the power to mimic like human brain for solving complex problems. Artificial intelligence (AI) models or machine learning models are the models that are developed based on machine learning (ML) algorithms. The ML algorithms use trial and error method to minimize error function. These AI models have better performance compared to the traditional models in case of nonlinear pattern due to the self-adaptive and data driven nature. The models are robust and have generalization power. The AI models first splits the whole dataset into two parts *i.e.* training and testing sets. Training set usually contains 70-80% of data. Training set is used in ML algorithm to build a model. After building the model, the generalization ability of the developed model is checked by testing data set. Some common AI based time series models are:

- Artificial Neural Network (ANN)
- Support Vector Machine (SVM)
- Decision trees
- Bayesian networks

1.2.B.1 Machine Learning (ML) algorithms

The definition of ML is given by Samuel (1959) as “Field of study that gives the computers the ability to learn without being explicitly programmed”. The formal definition of machine learning as given by Mitchell (1997) as “A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*". ML algorithm has been categorized broadly into three classes: supervised learning, unsupervised learning and reinforcement learning. Figure 1.2 visualizes the detailed overview of machine learning.

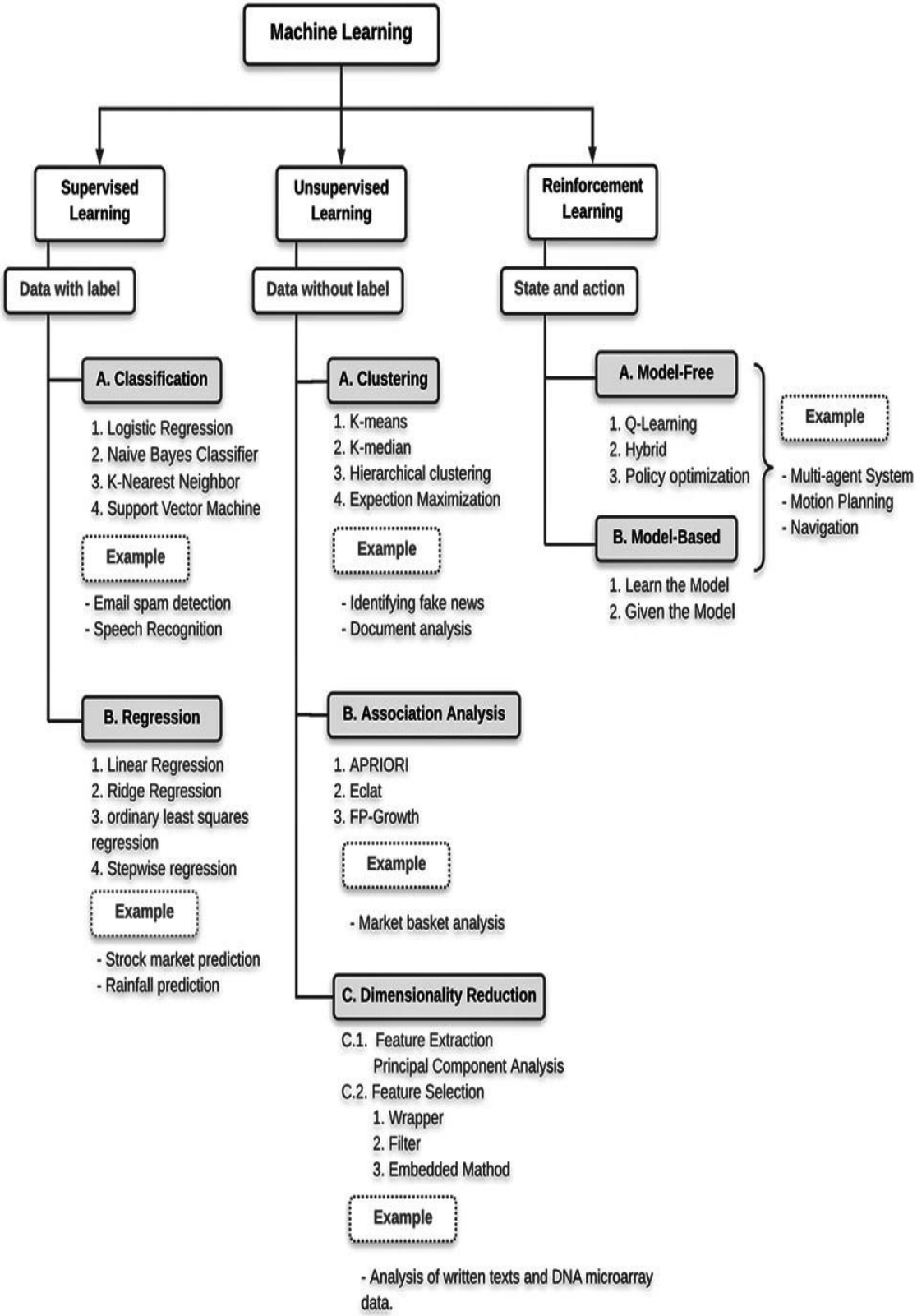


Figure 1.2: An overview of Machine Learning Process (Mitchell, 1997)

1.2.B.1.a Supervised Learning

In supervised learning, input variables are given along with already known output. In short, data are labeled. The prior information about input data pattern is known. The supervised problem can be classified into two categories *i.e.* regression problem and classification problem.

1.2.B.1.b Unsupervised Learning

In unsupervised learning, input variables are provided with little or zero idea about the results of the inputs. Unsupervised learning technique derives a structure/pattern from the inputs without prior information about the relationship among input variables. There is no feedback based on prediction results. In literature, there are three types of unsupervised learning algorithms. They are:

I. Clustering
II. Association analysis
III. Dimensionality reduction

1.2.B.1.c Reinforcement Learning

Reinforcement learning is also known as dynamic programming. It is advanced learning process. The main components of the learning are state, action, agent and interpreter. The agents take actions in an environment and interpreter interpret the results and provide some results of the actions. The specialty of the learning is the feedback of actions to the agents. The reinforcement learning is categorized into model based and model free reinforcement learning.

1.2.C Smoothing of time series data

A time series dataset comprises lots of complex and hidden properties. Therefore, forecasting and estimation of parameters from raw data is quite difficult task. Smoothing helps to remove the irregular roughness from the raw data and make better visualization of data pattern. In the early stage, exponential smoothing was popular method for forecasting. Exponential smoothing is a collection of ad hoc techniques for extrapolating the time series models. Holt (1957) and Winters (1960) developed the modified forms of

exponential smoothing *i.e.* additive (linear) and multiplicative for classification of trends and seasonal patterns in time series datasets. In literature, there are few works in nonlinear smoothing and multivariate exponential smoothing. In recent years, several advanced smoothing techniques like wavelet technique (Anjoy and Paul, 2019), Empirical Mode Decomposition (EMD) (Huang *et al*., 1998) *etc.* have been proposed in the literature. In contrast to wavelet decomposition, empirical mode decomposition does not require to define a filter based function before decomposition. These decomposition techniques are mostly data-driven and free from prior assumptions regarding data generating process. They simply divide original data into different sub series and each of the sub series can further be modeled by any forecasting technique.

1.2.D Hybrid time series model

Hybrid time series model is a combination of two or more methods to use their strengths. It becomes necessary to use hybrid model because a single model may not handle the inherent data patterns like nonlinearity and non-stationarity simultaneously. Besides this, both statistical and AI based time series models have their prerequisite assumptions. These problems demand the necessity for combining two or more forecasting models. In literature, there are many examples for hybrid models. Some hybrid models are GARCH-NN (Donaldson and Kamstra, 1997), ARIMA-ANN (Zhang, 2003), ARIMA-SVR (Che and Wang, 2010) *etc*. The main driving force behind the hybrid modelling is to improve the forecast accuracy. In this study, hybrid models include combination among soft computing methods like EMD, ANN, SVR and Multivariate Adaptive Regression Splines (MARS).

1.3 Motivation

In real world situations, both linear and nonlinear autocorrelation patterns are usually embedded in time series data (Zhang, 2003). This makes it very difficult to model the exact data generating process of the series. Thus, it poses a challenge for the forecaster to select the appropriate model for their unique situations. It has been exhaustively debated in time series literature that no single technique is the best in every situation. This can be attributed to the complex nature of a real-world problem where, any single model fails to

capture different structures equally well. The forecasting performance of traditional time series models are not satisfactory for highly nonlinear and nonstationary data like agricultural prices. Either traditional time series model or machine learning technique singularly cannot capture both linear and nonlinear patterns equally well. Therefore, there is a need of hybrid models *i.e.* combination of time series model and machine learning technique which can deal with both linear and nonlinear pattern and improve forecasting accuracy in agriculture.

The linkages among agricultural commodity prices plays an important role for policy makers. There is a lack of systematic study to assess impact of increase in one commodity prices on prices of other agricultural commodities in India. Hence, development of a Time Delay Neural Network (TDNN) based hybrid forecasting model considering co-integration is needed for commodity price forecasting.

Further, identification of important input variables is a crucial step in determining the accuracy of any forecasting model. MARS, a nonlinear model can be a potential candidate to identify important predictors in case of crop yield, a complex trait. Hence, the hybrid model based on MARS and ANN bears immense potential in improving the yield prediction.

Keeping these research gaps in mind, we undertook the present investigation with following objectives:

1. To develop empirical mode decomposition based support vector regression model for agricultural price forecasting.
2. To develop a time-delay neural network based hybrid forecasting model considering co-integration for commodity prices.
3. To investigate the performance of multivariate adaptive regression splines based neural network for yield prediction.

1.3 Orientation of the Thesis

The entire thesis is divided into five chapters. Chapter 1 introduces the present scenario of forecasting and necessities behind the forecasting in agricultural domain. The basic

times series models and machine learning algorithms have been described in this chapter. In chapter 2, first a detailed description of used datasets is given. Later, this chapter gives an overview of the pervious works which motivated us to frame the objectives of the present study. The detailed representation of general research methodologies and proposed hybrid models have been discussed in chapter 3. Besides the methodologies, the involved tests for the model adequacy checking and accuracy are demonstrated in this chapter. Overall results of proposed hybrid models and the inference of the obtained results have been included in chapter 4. Chapter 5 summarizes the overall obtained results of the present study. Finally, the thesis is concluded with English and Hindi abstracts followed by annexure and bibliography.

CHAPTER II

REVIEW OF LITERATURE

2.1 Background

This chapter provides an overview of the previous works relating to the objectives of the present study. The chapter has three subsections. The first subsection mentioned the previous works related to EMD and SVR methods in price forecasting. The previous works related to co-integration and ANN have been reviewed in second subsection. The third subsection documents the previous investigations relating to MARS, ANN and hybrid models for yield forecasting.

2.2 The reviews related to EMD and SVR models are given below:

Box and Jenkins (1970) developed a coherent, versatile three-stage iterative cycle for time series identification, estimation, and verification (rightly known as the Box–Jenkins approach). The proposed Box-Jenkins approach is the basic fundamental for a model development.

Granger and Joyeux (1980) introduced Autoregressive Fractionally Integrated Moving Average (ARFIMA) model. They used the concept of fractional differencing in replacement of traditional infinite filter concept and generated a time-series class having lower frequencies. The model was designed to provide forecasting properties of long memory.

Engle (1982) introduced the class of Autoregressive Conditional Heteroskedastic (ARCH) models to describe the dynamic changes in conditional variance of a time series. He used the conditional variance as a deterministic (typically quadratic) function of past returns. This model was used on the data set of inflation of United Kingdom to estimate the average and the variance values which disclosed the noticeably increase in estimated variance in the period of seventies that indicates the significant effect of ARCH.

Cheung (1993) explored that in long run forecasting autoregressive fractionally integrated moving average and generalized autoregressive conditional heteroskedasticity models

have limited success rate. They used three different tests i.e. "the Geweke-Porter-Hudak (GPH) test, the Modified Rescaled Range (MRR) test and two Lagrange Multiplier (LM)" type tests for fractional integration. They reported GPH and MRR tests were robust to autoregressive and heteroskedasticity components.

Allen (1994) observed that initially conventional methodologies of economics for price forecasting had major occupancies but later comparison and adoption of more accurate forecast methods have been started for time series observations. He pointed out the issue of little emphasis on the accuracy of models rather than explanation. He also commented that the principles underlying time series models like autoregressive conditional heteroskedasticity model and its generalized form as the GARCH model assume that there are periods of relative high and low volatility, though the underlying unconditional variance remains unchanged.

Donaldson and Kamstra (1997) proposed the GARCH-NN models for modelling volatility of stock market data for London, New York, Tokyo and Toronto. The model was semiparametric and nonlinear in nature. They reported that the both in-sample and out of sample performance of Artificial Neural Network (ANN) based GARCH model was better than the traditional models.

Suykens and Vandewalle (1999) proposed Least Squares Support Vector Regression (LSSVR) to transform inequality constraints into equality constraints by employing a squared loss function.

Huang *et al.* (1998) proposed empirical mode decomposition and the Hilbert spectrum for analyzing nonlinear and nonstationary data. They highlighted the main advantages of the EMD *i.e.* the lack of initial assumptions on the dataset like stationarity or linearity and it does not use a priori determined basis functions.

Zhang (2003) proposed a hybrid ARIMA and Neural Network (NN) model for time series forecasting. He concluded that the proposed hybrid methodology has advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modelling.

Ince and Trafails (2006) proposed a hybrid model based on autoregressive integrated moving average and support vector regression in order to improve forecasting accuracy. The proposed methodology outperformed the Logit/Probit models.

Chen (2007) demonstrated that superiority of support vector regression over the neural network and maximum likelihood estimation.

Brandl *et al.* (2009) used genetic algorithm for variable selection and set their model using SVR methodology. The proposed model outperformed a neural network, an Ordinary Least Squares (OLS) regression and ARIMA model.

Lu *et al*. (2009) highlighted that SVM has generalization capacity to obtain a unique solution. They proposed a new hybrid method combining the Independent Component Analysis (ICA) and SVR methodology for financial time series analysis. The ICA generated the Independent Components (ICs) and SVR forecast the ICs. The results showed the proposed model outperformed the normal SVR model.

Che and Wang (2010) proposed a hybrid model SVRARIMA model for forecasting nonlinear and nonstationary dataset. The proposed hybrid model captured the nonlinearity of the data set using SVR model and then the estimation of the residual regression problem was done using ARIMA method. They applied the proposed methodology to forecast deregulated electricity data. The results showed the superior performance of the SVRARIMA model.

Duan and Stanley (2011) tried to improve the prediction of financial return data series by using SVM method and removing the effect of cross-correlations between different marketplaces of finance. But they concluded that it was hard to improve the financial prediction using forecast model of SVM based on correlated return series data .They highlighted the enhancement of SVM model due to the structural risk minimization principle.

An *et al.* (2012) reported that the hidden pattern and trends of time series can be captured using EMD. They suggested this property of EMD can effectively assist in designing forecasting models for various applications.

Guo *et al.* (2012) decomposed wind speed series using EMD and forecasted them using a feed-forward network. They found that the proposed EMD based feed-forward network was more accurate than conventional forecasting methods.

Wang *et al.* (2012) developed a new approach Ensemble Empirical Mode Decomposition (EEMD) in combination of EMD and classical Hilbert Transformation (HT). They applied the EEMD and EMD in time-frequency analysis of seimic signal data. They concluded that the proposed method EEMD was better than EMD.

Chaâbane (2014) proposed a hybrid model combing ARFIMA and least square SVM. The hybrid model used to forecast electricity market price. They concluded that the hybrid model had better prediction accuracy.

Jha and Sinha (2014) combined ANN and ARIMA model for time series forecasting. The proposed feed-forward time-delay neural network is able to handle nonlinearity feature of dataset. The performance of the proposed TDNN is checked on the monthly Wholesale Price Index (WPI) of oilseed in India. The study showed that the proposed model performed better than the linear models.

Paul (2015) forecasted volatile data combining ARIMAX, GARCH and wavelet approach. The study revealed that the combined method produced significant improvement in forecasting compared to the generic methods.

Paul *et al.* (2015) combined AR and Fractionally Integrated GARCH (FIGARCH) and applied it for forecasting spot price of lentil. The study suggested that the proposed model can be an alternative approach for price forecasting.

Kim and Oh (2016) demonstrated the imputation capability of the EMD for missing values.

Lama *et al.* (2016) explored the superiority of GARCH based time-delay neural networks for forecasting agricultural commodity price volatility. They developed 26 TDNN models with different combinations of parameters.

Komasi and Sharghi (2016) proposed a hybrid wavelet-support vector machine approach for modelling rainfall-runoff process. They suggested that the proposed hybrid model could be the best alternative model for modelling rainfall data.

Ray *et al.* (2016) proposed a new hybrid model combining ARIMA and Wavelet Neural Network (WNN). Forecasting accuracy of the proposed model was better as compared to the existing approach.

Khashei and Hajirahimi (2018) developed a series of hybrid models combining Multilayer Perceptrons (MLPs) and ARIMA. They used a time series data comprising two linear and nonlinear components for the model fitting. They concluded that ANN-ARIMA model had the best forecasting accuracy.

Wu *et al.* (2018) proposed a new EMD based Back Propagation Neural Network (BPNN) model for prediction of sea surface temperature. They employed different EMD algorithms *i.e.* ensemble empirical mode decomposition algorithm and Complementary Ensemble Empirical Mode Decomposition (CEEMD) to decompose the original data series and BPNN method for prediction each decomposed subseries. The study suggested that the CEEMD-BPNN model had better performance than EEMD-ANN model.

Das *et al.* (2019) proposed EMD based SVR model for agricultural price forecasting. They compared EMD-ANN and EMD-SVR models in chilli monthly wholesale price index. The study showed performance of EMD-SVR was better than EMD-ANN model.

Choudhury *et al.* (2019) combined EMD and ANN for potato price index in India. They suggested empirically that the proposed EMD-ANN model was the best alternative for price forecasting.

Suhartono *et al.* (2019) proposed a new hybrid model considering statistical model and machine learning models for predicting Particulate Matter 10 (PM10) at three stations in Surabaya City, Indonesia. They used Time Series Regression (TSR) as a statistical method and Feedforward Neural Network (FFNN) or Long Short-Term Memory (LSTM) as machine learning model. Daily and weekly seasonality of the data are accounted in the study. The results showed that TSR-FFNN model produced more accurate results.

2.3 The reviews related to cointegration and TDNN models are given below:

Engle and Granger (1987) introduced the concept of co-integration for analysis of the long-run equilibrium relationship between economic variables.

Campbell and Shiler (1987) investigated the existence of linear co-integration between aggregate U.S. stock prices and U.S. dividends.

Johansen (1988 and 1992) provided a detailed study on co-integration. The research papers described the statistical analysis of cointegrating vectors and their asymptotic properties.

Riedmiller and Braun (1993) proposed a fast training algorithm Resilient Backpropagation (RPROP). The paper demonstrated the detailed of the algorithm with simulated results.

MacKinnon (1996) derived some numerical distribution functions for unit root and co-integration tests.

Balke and Fomby (1997) developed an estimation method where the dynamics are subjected to three-regime threshold co-integration. They used univariate test for the error correction term.

Zhang *et al*. (1998) reviewed the applications of neural networks in time series forecasting. The study gave the detailed survey for ANN modelling in time series forecasting along with future aspects.

MacKinnon *et al.* (1999) formulated numerical distribution function of likelihood ratio test for co-integration.

Zhang (2003) proposed a hybrid ARIMA and neural network model for time series forecasting. They concluded that the proposed hybrid methodology has the advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modelling.

Aloy *et al.* (2010) used fractional integration and co-integration in stock prices and exchange rates. They found that these variables are integrating order one nonstationary

series, but they are fractionally cointegrated: equilibrium errors exhibit slow mean reversion.

Jha and Sinha (2014) combined ANN and ARIMA model for time series forecasting. The proposed feed-forward TDNN is able to handle nonlinearity feature of dataset. The performance of the proposed TDNN is checked with monthly wholesale price of oilseed in India. The study showed that the proposed model performed better than the linear models.

Lama *et al*. (2016) explored the superiority of GARCH based time-delay neural networks for forecasting agricultural commodity price volatility.

Truchis and Keddad (2016) investigated stock market integration in East Asia by analyzing the co-persistent nature of their observed volatility.

Wegener *et al.* (2016) proposed co-integration based ANN model to forecast government bond yield in Germany and France. The study showed the superior performance of the proposed hybrid approach than the single ANN model.

Jebli and Youssef (2017) investigated the impact of co-integration and Granger causality on Tunisian economy. They investigated short and long-run relationships between per capita carbon dioxide (CO_2) emissions, real GDP, renewable and non-renewable energy consumption, trade openness ratio and Agricultural Value Added (AVA) in Tunisia. The Johansen-Juselius test shows long-run bidirectional causalities between all considered variables.

Kumar and Jha (2017) investigated co-movement and causality between prices of agricultural commodities and energy using Johansen co-integration approach.

Johansen and Nielsen (2018) investigated the nonstationary co-integration in time series data. They demonstrated some asymptotic properties of likelihood function of the fractional Co-integrated Vector Autoregressive (CVAR) model.

Silva *et al.* (2019) developed an early warning system for commodity price. The application was constructed based on ANN. The performance of proposed model was

checked with ANN, SVM and other traditional models. The study showed that the proposed application had high potential in sale forecasting and more accurate prediction.

David *et al.* (2019) worked on the co-integration between ethanol and agricultural commodity price series. They also showed how this relationship effect on the predictability and efficiency of the cointegrated price series.

2.3 The reviews related to MARS and ANN models are given below:

Friedman (1991) introduced the concept of multivariate adaptive regression splines to multidimensional nonlinear dataset and described its application. The study also highlighted how MARS deal with nested variables with an example.

Riedmiller and Braun (1993) proposed a fast training algorithm resilient backpropagation. The paper demonstrated the detail of the algorithm with simulated results.

Friedman and Roosen (1995) worked on MARS for handling nested variables and adjusted missing values from discarded data.

Kuan and Liu (1995) applied feed forward and recurrent neural networks to forecast five exchange rates. Their result demonstrated that neural networks merely exhibit significantly superior abilities of market timing and/or out-of-sample forecasting for two exchange rate series.

Schultz and Wieland (1997) discussed the possibilities of applying neural networks or neural networks in combination with fuzzy techniques in field of agroecological modelling.

Zhang *et al*. (1998) reviewed the applications of neural networks in time series forecasting. The study showed the detailed survey for ANN modelling in time series forecasting along with future aspects.

Uno *et al.* (2005) used artificial neural networks to predict corn yield from compact airborne spectrographic imager data. They used statistical and ANN approaches along with various vegetation indices to develop yield prediction models.

Lee *et al.* (2006) found that MARS performed better than both statistical parametric methods such as linear discriminant analysis or logistic regression and nonparametric approaches such as neural network and support vector machines in mining customer credit.

Crino and Brown (2007) explored a novel procedure for approximating the global optimum in structural design by combining multivariate adaptive regression splines with a Response Surface Methodology (RSM).

Khazaei *et al.* (2008) applied ANN methodology for modelling the correlation between crop yield and 10 yield components of chickpea (*Cicer arietinum L.*). Also, the fuzzy c-means (FCM) clustering technique was used for the classification of 362 chickpea genotypes based on their agronomic and morphological traits. Among the various ANN structures, they found 10–14–3–1 ANN structure with a training algorithm of back-propagation and hyperbolic tangent transfer function in the hidden and output layers performed best.

Higgins *et al.* (2010) developed an ANN model for forecasting maturity of green peas using historical harvest information along with weather and climate forecasts. They implemented and evaluated the ANN in a large pea growing region in Tasmania, Australia. They highlighted the model provided the ability to not only harvest peas closer to their ideal maturity indices but also plan to harvest and transport logistics with a much greater lead time.

Kumar *et al.* (2010) tried to predict wheat yield using fuzzy logic. They reported the superiority of the hybrid model for forecasting wheat yield.

Andrés *et al.* (2011) proposed a hybrid approach using fuzzy c-means clustering and multivariate adaptive regression splines for bankruptcy forecasting. The study demonstrated the performance of proposed model in classification of decisions.

Kumar (2011) used adaptive neuro-fuzzy systems for rice yield forecasting.

Voyant *et al.* (2011) conducted a study on forecasting multivariate series of daily global radiation in France. They used an ad-hoc pre-processing technique

for time series data with the Multi-Layer Perceptron (MLP) as a forecasting technique. They identified the superior forecast accuracy of the ANN model.

Samui and Karup (2011) applied MARS to extract important variables from a data which contain both quantitative and qualitative variables.

Kumar and Kumar (2012) worked specifically on rice yield prediction using fuzzy time series model. They compared the performance of the model with the traditional model and highlighted how fuzzy concept can improve the forecasting accuracy.

Jha and Sinha (2014) combined ANN and ARIMA model for time series forecasting. The proposed feed-forward time-delay neural network able to handle nonlinearity feature of dataset. The performance of the proposed TDNN checked in monthly wholesale price of oilseed in India. The study showed that the proposed model performed better the linear models.

Koc and Bozdogan (2015) discussed the improvement in model selection of MARS algorithm by using information complexity. They introduced a new information criteria called Information-Theoretic Measure of Complexity (ICOMP) criterion for MARS model selection. They suggested ICOMP balanced the model overfitting and underfitting.

Ratjen and Kage (2015) developed the crop–soil model 'HumeWheat' for yield forecasting and parameterized on a broad experimental database including several modern wheat cultivars.

Zhang and Goh (2015) discussed how the MARS model captured the nonlinear relationship among the variables. They demonstrated the performance of MARS model in nonlinear structural problems with practical examples.

Kisi and Parmar (2016) applied least square support vector machine, multivariate adaptive regression splines and M5 model tree to model river water pollution of Yamuna River at Nizamuddin, Delhi in India. They used various water quality parameters as the input variables in the model. They found that the accuracy of LSSVM and MARS were high.

Krishnan *et al.* (2016) developed a web-based application of the crop simulation model 'Web InfoCrop - Wheat' using Visual Studio Express, SQL Server, NET framework 4.0.

This web-based model provides separate modules for input variables, management conditions, and result outputs.

Zhang and Goh (2016) compared the performance of MARS and neural network models for prediction of pile drivability. The study showed the accuracy and computing time of MARS was better than the ANN model.

Deo *et al*. (2017) applied MARS, least square, SVM and M5Tree for drought forecasting at different regions in eastern Australia. The results indicated the performance of different models varying region to region. Seasonal analysis indicated superiority of MARS model.

Garg *et al.* (2017) forecasted rice yield using fuzzy logic and regression model. They tested 4 different types of the fuzzy intervals with 4 degrees of regression equations. The study revealed how the order and prediction interval may affect on model accuracy.

Alam *et al.* (2018) developed an improved ARIMAX model based on ANN and SVM for rice yield forecasting in India. They used ARIMAX model for fitting of the rice yield data with weather variables as exogenous variable. The nonlinear residuals were fitted using ANN and SVM model. The results indicated that the performances of ARIMAX-ANN and ARIMAX-SVM model were superior than the conventional ARIMAX model.

Elavarasan *et al.* (2018) conducted a survey on forecasting crop yield. They integrated different agricultural data like crop protection, crop improvement *etc.* from the available public domain. Then different machine learning techniques like ANN, SVM, Random Forest, Decision Tree *etc.* were applied on the data to find out interrelation among the crop growth factors.

Oguntunde *et al.* (2018) conducted a study to find out the relationship between rice yield and climate variables in Nigeria. They used Multiple Linear Regression (MLR), Principle Component Analysis (PCA) and SVM in 36 years yield data. The results indicated the significance relationship among the variables.

Khaki and Wang (2019) studied the performance of deep learning technique for crop yield prediction. They designed a new technique called Deep Neural Network (DNN) approach and applied the proposed approach in Maize yield prediction. The study reported the DNN

approach had greater accuracy than the conventional methods like ANN. They suggested that combined AI models may produce better results in future.

Kim *et al.* (2019) conducted a comparative study among major crop yield prediction models. They compared major Artificial Intelligence (AI) models for corn and soybean yield.

Pannakkong *et al.* (2019) developed a hybrid model combining ARIMA and ANN model for forecasting cassava export in Thailand. They considered the moving average and seasonal index of the Thailand market in the market data and compared the accuracy of proposed model with the existing forecasting models. The study suggested the hybrid model can be used an alternative forecasting model for Cassava export forecasting.

Widodo *et al.* (2019) modelled average price of Garlic in Indonesia using ARIMA Box-Jenkins methodology. They found that ARIMA (1,1, 0) was best for the dataset based on in-sample and out-sample.

Nagpal and Singh (2019) proposed a new hybrid algorithm for feature selection based on random forest and MARS. The empirical results indicated that the proposed hybrid approach performed better than the traditional methods in disease diagnosis.

CHAPTER III

MATERIALS AND METHODS

This chapter contains two major sections. The first section provides a brief description of data used for the present study. Then the description regarding the used methodologies in analysis of the data has been discussed in the second section. The discussed methodologies involve the machine learning techniques, co-integration, criteria of model selection *etc*.

3.1 Data description

In the present study, three datasets were used to accomplish three objectives. For the first objective, development of empirical mode decomposition based support vector regression model, monthly chili wholesale price index (WPI) dataset was used. The dataset was obtained from the Office of the Economic Advisor, Ministry of Commerce, Government of India (https://eaindustry.nic.in). Figure 3.1 illustrates the monthly data of chili WPI from April, 1994 to May, 2018 containing 290 data points with base year 2004-05.

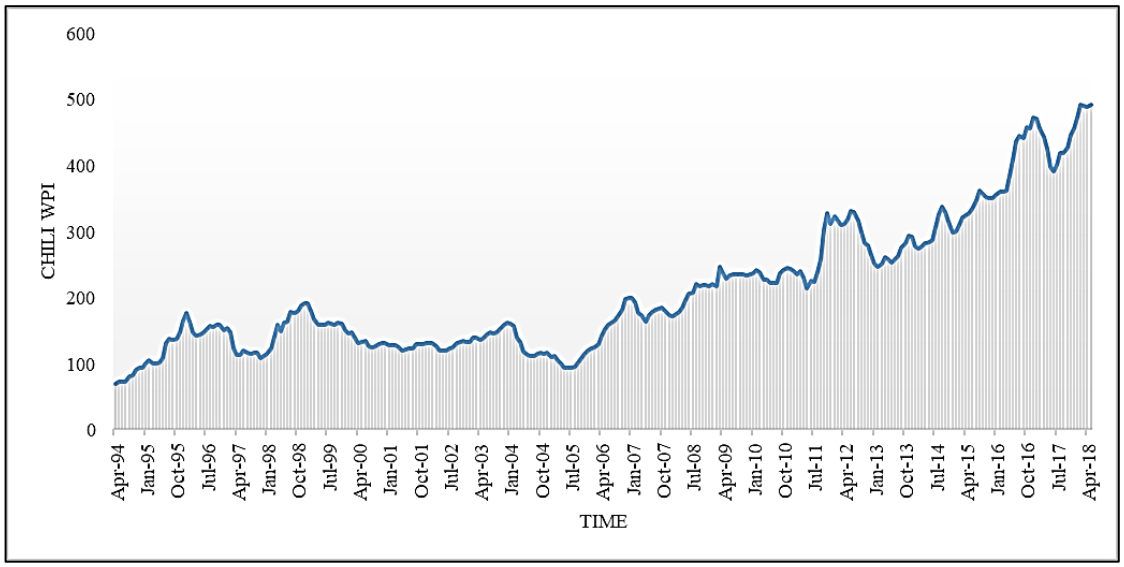


Figure 3.1: Time plot of monthly chili WPI

Under the second objective, monthly price indices of crude oil and fruits starting from April 1994 to March 2014 was used to develop a time-delay neural network based hybrid forecasting model considering co-integration. The price index of crude oil was obtained

from International Monetary Fund website (https://www.imf.org). The monthly price indices of fruit was collected from Office of the Economic Advisor, Ministry of Commerce, Government of India (https://eaindustry.nic.in). The data sets contain 240 data points (April, 1994 to March, 2014) in figure 3.2. Due to discontinuance of monthly price index of crude oil at IMF website in the public domain, both series were used upto March, 2014.

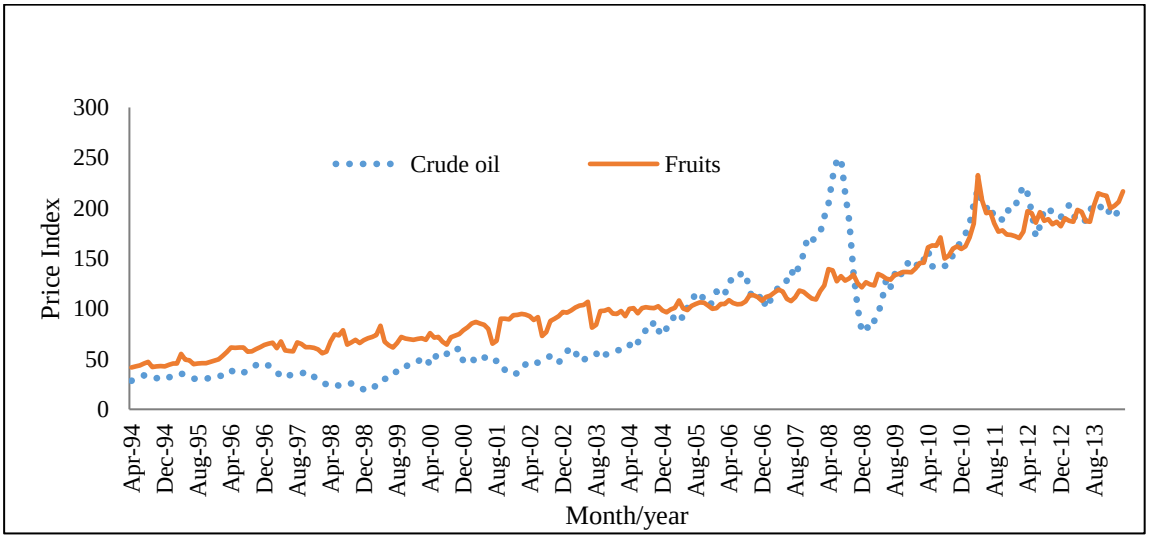


Figure 3.2: Time plot of crude oil vs fruits indexed price (nominal 2004-05=100).

Under third objective, 518 lentil accessions dataset was used to a MARS based hybrid model. Dataset contain 206 exotic collections and 312 indigenous collections. These accessions were grown in augmented block design with five checks during *rabi* season, 2006–07 at ICAR-Indian Institute of Pulses Research, Kanpur. Accessions were evaluated for 21 descriptors, including plant characteristics and seed characteristics following the biodiversity and national Distinctness, Uniformity and Stability (DUS) descriptors guidelines. Ten numerical descriptors were used for detailed analysis (table 3.1).

Table 3.1**:** Variables and their code of the lentil dataset

Code	Name
DF	Days to 50% flowering
PH	Plant height
DM	Days to 90% maturity
SW	100 seed weight

BYP	Biological yield/plant
PB	Primary branch/plant
SB	Secondary branch/plant
PPP	Pods/plant
YPP	Yield/plant
PHLP	Plant height at lowest pod

3.2 Procedure of the study

Fitting of any forecasting model involves various recursive steps and requires careful examination in each individual step.

The basic steps for fitting of the forecasting model are (Box and Jenkins, 1970) as follows:

I. Model identification.
II. Parameter estimation.
III. Diagnostic checking.

The basic process of model development is depicted in the form of a flowchart in the figure 3.3. In the present investigation, an attempt has been made to develop three hybrid models for forecasting agricultural datasets with different characteristics. Under the first objective, empirical mode decomposition based support vector regression model for agricultural price forecasting has been developed. The model captures linear as well as nonlinear patterns of a nonstationary and nonlinear data. EMD is used for smoothing of data and SVR is used for forecasting. Under the second objective, a time-delay neural network based hybrid forecasting model considering co-integration for commodity prices has been developed. Under this objective, a time-delay neural network in combination with the nonlinear error correction model (Kapetanios *et al*., 2006) has been employed to develop data-driven flexible co-integration model. Under the third objective, the performance of nobel Multivariate Adaptive Regression Splines (MARS) based neural network model is evaluated. MARS has been used to extract important variables from multivariate data and prediction is done using ANN. The entire analysis is done using R

software. In the following section the detail of basic test procedures and generic methods are discussed.

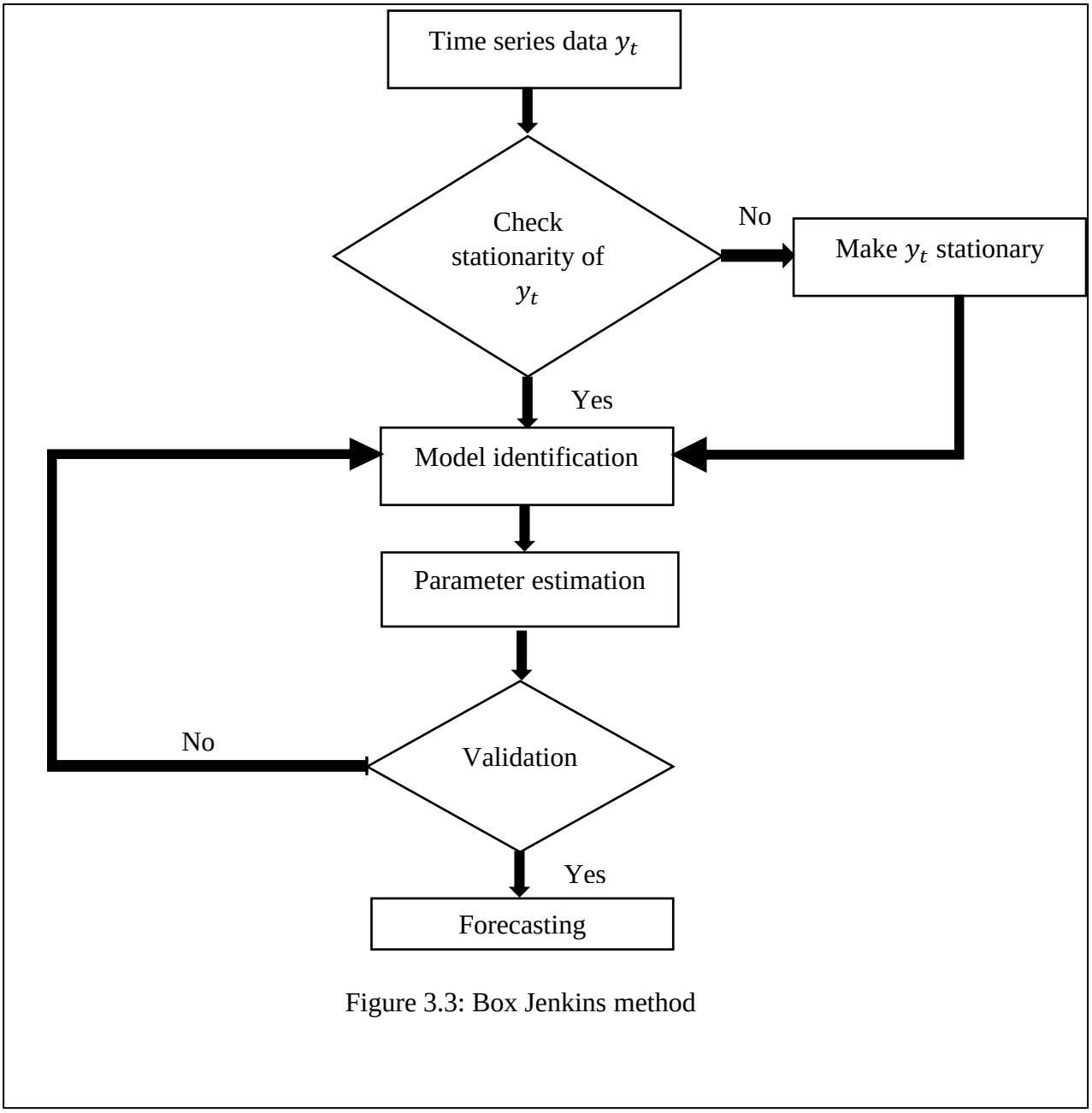


Figure 3.3: Box Jenkins method

3.2.1 Test for stationarity and linearity

In a stochastic process for a time series data the assumption of stationarity hold prime importance. It refers to the mean and variance constant over time and covariance between

the two time periods depends only on the distance or gap or lag between the two time periods then the process is said to be stationary.

In this study, Augmented-Dickey-Fuller (ADF) and Phillips-Perron (PP) tests have been used to test the stationarity in the data.

For a time series data y_t, the regression equation of ADF (Dickey and Fuller,1981) test is represented as follows:

$$\Delta y_t = \alpha_1 + \alpha_2 t + \delta y_{t-1} + \sum_{i=1}^{h} \beta_i \, y_{t-i} + \varepsilon_t \quad \ldots (3.1)$$

where $\Delta y_t = y_t - y_{t-1}$, α_1,α_2 and β_i are parameters of regression model, h is lag length. $\delta = \rho - 1$ and $-1 \leq \rho \leq 1$

In the ADF test, $\delta = 0$ indicates the time series under consideration is nonstationary. In the tests null hypothesis (H_0) and alternative hypothesis (H_1) are as follows:

$H_0 =$ Unit root is present in a time series data (Time series is nonstationary)

$H_1 =$ Time series is stationary

PP (Phillips and Perron, 1988) test detects the non-stationarity of the time series data. PP test is robust with respect to unspecified autocorrelation and heteroscedasticity in error term.

Besides, the nonlinearity of the data series was also checked using Brock-Dechert-Scheinkman (BDS) test (Brock *et al*., 1996). The null hypothesis (H_0) and alternative hypothesis (H_1) of the BDS test are as follows:

$H_0 =$ Time series is linear

$H_1 =$Time series is nonlinear

3.2.2 Empirical Mode Decomposition (EMD)

The empirical mode decomposition method was introduced by Huang *et al*. in 1998. It assumes that the data have many coexisting oscillatory modes of significantly distinct frequencies and these modes superimpose on each other and form an observable time series. EMD decomposes original nonstationary and nonlinear data into a finite and small number of independent sub-series (including intrinsic mode functions and a final residue).

Intrinsic Mode Function (IMF) is the finite additive oscillatory component decomposed by EMD. For example, let y_t is a dataset consisting of high frequency part and low frequency part.

Data = fast oscillations superimposed to slow oscillations

$$y_t = d_t(1) + r_t(1) \quad \ldots (3.2)$$

where $d_t(1)$ = high frequency part *i.e.* IMF and $r_t(1)$ = low frequency part *i.e.* residue.

EMD algorithm iterates over the slow oscillation component considered as a new signal.

$$r_t(1) = d_t(2) + r_t(2)$$

After full decomposition, $y_t = \sum_{i=1}^{n} d_t(i) + r_t \quad \ldots (3.3)$

Data = sum of IMFs + final residue.

Stepwise EMD algorithm procedure is mentioned below:

Step 1: Identify all extrema of y_t

Step 2: Interpolate the local maxima to form an upper envelope *u(x)*

Step 3: Interpolate the local minima to form a lower envelope *l(x)*

Step 4: Calculate the mean envelope: $m_t = [u(x)+l(x)]/2$

Step 5: Extract the mean from the signal: $d_t = y_t - m_t$

Step 6: Check whether d_t satisfies the IMF condition.

YES: d_t is an IMF, stop shifting.

NO: let $y_t = d_t$, keep shifting.

All these steps come under the first iteration in the sifting process for y_t. The sifting process continues till we obtain an IMF. The point of termination of a sifting process is called stopping point k and the iteration is called k^{th} iteration.

3.2.3 Intrinsic Mode Function (IMF)

EMD decomposes the original time series into several intrinsic mode functions and final residue. It is mainly used in computing instantaneous frequency through the Hilbert transformation for a given data series (Huang *et al*., 1998). The whole process of decomposition undergoes through a sifting process. Each of IMFs has different frequencies and different amplitude. IMFs are arranged in a decreasing order with respect to their frequencies *i.e.* first IMF has highest frequency. Each IMF follows two conditions (Huang *et al*., 1998):

1) Difference between number of extrema and number of zero crossing should either be equal or at most one.
2) For an IMF, mean value of envelopes (both upper and lower) should be zero.

For IMF construction at first, instantaneous frequency for a given data series over all time point is calculated. First IMF shows a narrow band and it becomes looser from first IMF to last IMF. The symmetry in an IMF is maintained by the help of second IMF condition.

3.2.3.1 Sifting process

Suppose the original price series $y_t, (t = 1, 2, \ldots, T)$ is decomposed into $n + 1$ modes, including n IMFs $d_t(i), (i = 1, 2, \ldots, n)$ and one residue r_t which is the main trend of series:

$$y_t = \sum_{i=1}^{n} d_t(i) + r_t \qquad \ldots (3.4)$$

All these n IMF are obtained through a sifting process.

3.2.3.2 Finding the first IMF

According to EMD procedure, most varying or frequent oscillatory mode among other modes is the first IMF. The short overview of finding out the IMF is given below.

Let y_t is input signal to initialize sifting process. After first shifting, the input signal decomposed into sub signal

$$y_t = S_t(10) \qquad \ldots (3.5)$$

where sub signal is denoted by $S_t(ij)$, the first index denotes IMF number $i = 1,2,3, \cdots, n$ and the second index denotes iteration number $j = 0,1,2,3, \cdots, k$ of the sifting process. If

T is length of data series, then total number of IMFs will be $log_2 T$. A shifting process uses the aforementioned six steps (3.2.2) of EMD to find a IMF.

IMF can be identified using stopping criteria like:

a) **Standard deviation** – If two components from consecutive iterations are near enough to each other, it means that extracted component is IMF (Huang *et al*., 1998). To find out first IMF, $S_t(1(k-1))$ and $S_t(1(k)$ are respectively $(k-1)^{th}$ iteration and k^{th} iteration. Each iteration consists equal data point *i.e*. T same as original series.

$$SD = \sum_{t=0}^{T} \frac{[S_t(1(k-1)) - S_t(1\,k)]^2}{[S_t(1(k-1))]^2} \quad \ldots (3.6)$$

where SD value lies between 0.2 to 0.3. If $SD \ll 0.2$ then it leads to the mode mixing of IMFs and if $0.3 \ll SD$ than it also creates problem at the time of separation of IMF from an original series.

b) **IMF condition**- An iteration is said to be the last iteration to find out an IMF in a sifting process if the number of extrema and number of zero crossings are equal or differ by at most one. Generally, 4 to 10 iterations are required to achieve the condition (Huang *et al*., 1998).

Based on both the stopping criteria, two possible outputs for $S_t(11)$ are mentioned below-

1) When $S_t(11)$ is satisfying both or any one stopping criterion then it is an IMF *i.e*. first IMF which one extracted from y_t. Here first IMF represented by $d_t(1)$

$$S_t(11) = d_t(1) \quad \ldots (3.7)$$

Here $d_t(1)$ consist of all the important characteristics of first IMF. Next, first residual $r_t(1)$ can be obtained by subtracting first IMF from original series.

$$r_t(1) = y_t - d_t(1) \quad \ldots (3.8)$$

$$r_t(1) = S_t(10) - d_t(1) \quad \ldots (3.9)$$

2) If $S_t(11)$ violates both or any one stopping criterion, it is not an IMF. $S_t(11)$ will be considered as original series and repeat second and third step of sifting process for new input $S_t(11)$ instead of y_t. The second iteration mean envelope is represented by

$$m_t(12) = \frac{S_t(11)^{up} + S_t(11)^{low}}{2} \quad \ldots (3.10)$$

Now subtract $m_t(12)$ from new input $S_t(11)$ i.e.

$$S_t(12) = S_t(11) - m_t(12) \quad \ldots (3.11)$$

Then check whether $S_t(12)$ is not an IMF. The same procedure will continue k times to find out $S_t(1k)$ which satisfy the necessary condition for an IMF.

$$S_t(13) = S_t(12) - m_t(13) \quad \ldots (3.12)$$

$$S_t(14) = S_t(13) - m_t(14) \quad \ldots (3.13)$$

$$S_t(15) = S_t(14) - m_t(15) \quad \ldots (3.14)$$

$$\vdots$$

$$S_t(1k) = S_t(1(k-1) - m_t(1(k)) \quad \ldots (3.15)$$

If first IMF is obtained after k iteration then

$$S_t(1k) = d_t(1) \quad \ldots (3.16)$$

Now after extracting first IMF $d_t(1)$ remaining portion is residue $r_t(1)$ of series. The residue further will be used as original series in sifting process to extract second IMF. For this new input again, repeat all 1 to 5 steps (3.2.2) to extract second IMF $d_t(2)$.

$$r_t(2) = r_t(1) - d_t(2) \quad \ldots (3.17)$$

$$r_t(3) = r_t(2) - d_t(3) \quad \ldots (3.18)$$

$$\vdots$$

$$r_t = r_t(n) = r_t(n-1) - d_t(n) \quad \ldots (3.19)$$

Thus all n IMFs $(d_t(i), i = 1,2,3, \cdots n)$ and one final residual r_t have been obtained. Now original data series y_t can be represented by,

$$y_t = \sum_{i=1}^{n} d_t(i) + r_t \quad \ldots (3.20)$$

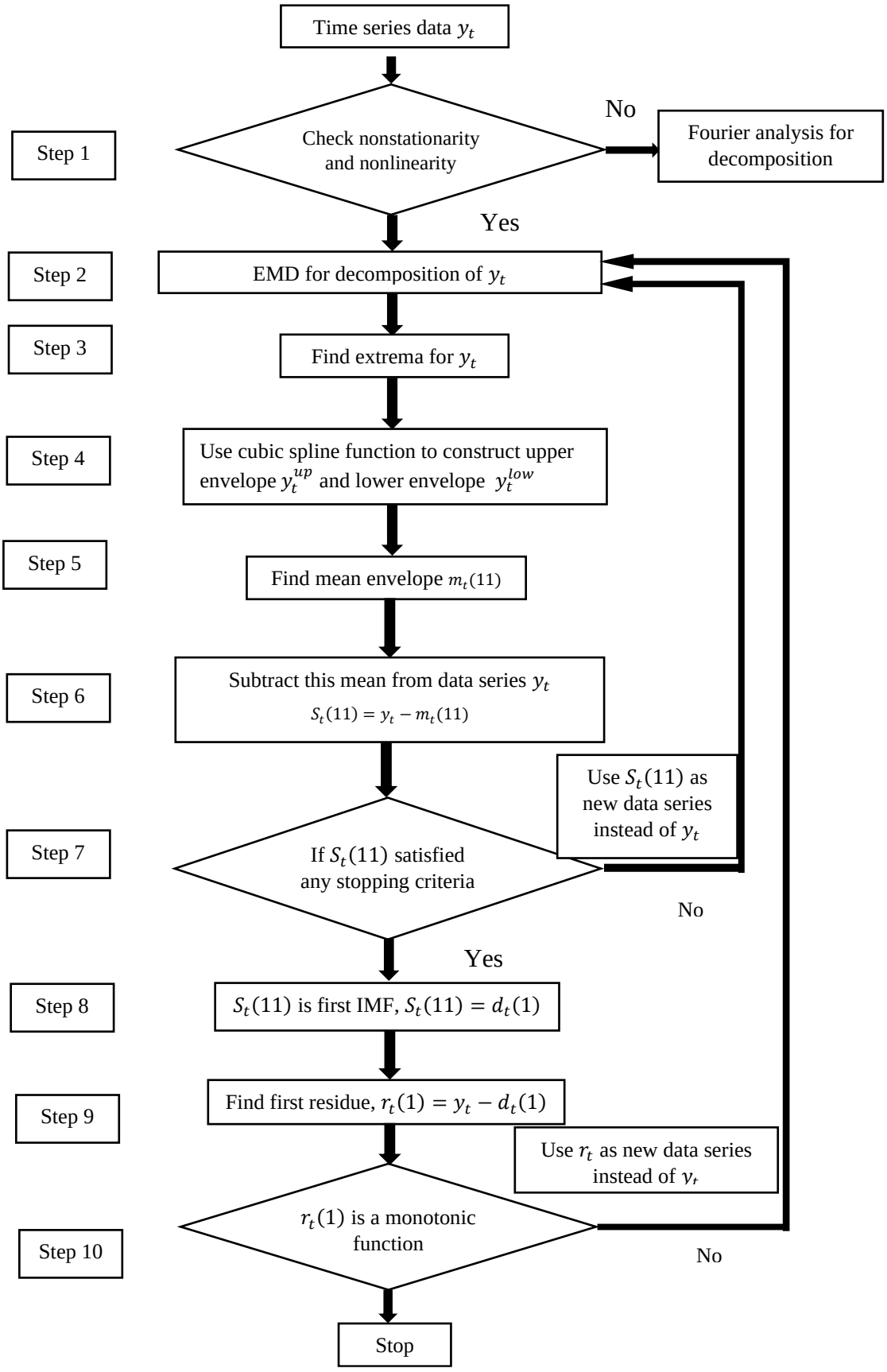


Figure 3.4: Flowchart of EMD process for a data series y_t

The complete procedure of EMD to extract first IMF from a data series y_t is depicted through a flowchart (Figure 3.4).

3.2.4 Support Vector Regression (SVR) model

Support vector machine proposed by Vapnik (1998), is a nonlinear algorithm used in supervised learning frameworks for data analysis and pattern recognition. Vapnik (1998) introduced support vector regression model by incorporating Ɛ-loss function. SVR maps input vectors into a high dimensional space and then fits linear regression in the outer space. The model has been built in two steps *i.e.* the training and the testing. In the training step, the largest part of the dataset has been used for the estimation of the function. In the testing step, the generalization ability of the model has been evaluated by checking the model's performance in the small subset that was left aside during training.

For a given data set $\{(x_1, y_1),...,(x_n, y_n)\}(x_i \in R^K, y_i \in R^1)$, SVR maps the original data into a higher or infinite dimensional space by nonlinear function φ, then seeks mapping function $\phi : R^K \to R^1$.The general formula for linear support vector regression is given as:

$$y = \phi(x) = w^T \varphi(x) + b \qquad \ldots (3.21)$$

where *w* defines weight vector, φ denotes mapping function and *b* is a bias term. LS-SVR is the least square of SVR, where set of linear equations are used to find the solution. The solution of *W* and *b* in above equation can be obtained by solving the following minimization problem (Sermpinis *et al.*, 2014)

$$\min_{W,b,\zeta,\zeta^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\zeta + \zeta^*) \qquad \ldots (3.22)$$

such that $w^T\varphi(x_i) + b - y_i \le \varepsilon + \zeta_i$; $y_i - w^T\varphi(x_i) - b \le \varepsilon + \zeta_i^*$

$\zeta, \zeta^* \ge 0, i = 1,..., N$.

It is a primal function and solution of the function is quite complex in nature. So, the dual of the function can be used. Its dual will be

$$\min_{\alpha,\alpha^*} \frac{1}{2}\sum_{i,j}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)Ker(x_i, x_j) + \varepsilon\sum_{i}^{N}(\alpha_i - \alpha_i^*) + \sum_{i}^{N} y_i(\alpha_i - \alpha_i^*) \quad \ldots (3.23)$$

$$\text{s.t. } \sum_{i}^{N}(\alpha_i - \alpha_i^*) = 0 \text{ and } 0 \le \alpha, \alpha^* \le C, i = 1, ..., N$$

where $Ker(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ is a kernel function. For getting estimated value α, α^* the dual function will be used. Thus, the coefficient *b* will be calculated as

$$\tilde{b} = y_j - \sum_{i}^{N}(\alpha_i - \alpha_i^*)Ker(x_i, x_j) - \varepsilon.; \tilde{\alpha}_i \in (o, C) \quad \ldots (3.24)$$

$$\tilde{b} = y_j - \sum_{i}^{N}(\alpha_i - \alpha_i^*)Ker(x_i, x_j) + \varepsilon. \; ; \; \tilde{\alpha}_i^* \in (o, C) \quad \ldots (3.25)$$

The bias term, *b* in Eq. 3.21 can be accommodated within the kernel function $Ker(x_i, x_j)$ and the regression function is given by:

$$f(x) = \sum_{i}^{N}(\bar{\alpha}_i - \alpha_i^*)K(x_i, x_j). \quad \ldots (3.26)$$

The SVR model (Eq. 3.26) contains three tuning parameters in the $Ker(x_i, x_j)$:

1. € epsilon of the loss function 2. C, the constraints and 3. sigma of the kernel.

3.2.4.1 Least Squares Support Vector Regression (LS-SVR)

Least squares support vector regression proposed by Suykens *et al.* (2002). This is a modified version of SVR and a more simple method than SVR. In LS-SVR model, a set of linear equations are used instead of a quadratic programming problem.

LS-SVR model is represented as:

$$y = w^T \varphi(x) + b \quad \ldots (3.27)$$

with $x \in R^K$ and φ, mapping function $R^n \to R^{n_h}$ to high dimensional feature space.

For a given training set $\{x_k, y_k\}_{k=1}^{N}$, optimization problem

$$\min\{w,e,b\}\, J(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \quad \ldots (3.28)$$

subject to equality constraints

$$y_k = w^T \varphi(x_k) + b + e_k \,;\; k = 1,2,\ldots,N. \quad \ldots (3.29)$$

This is a form of ridge regression. Now incorporating Lagrange multiplier α_k

$$L(w,b,e;\alpha) = J(w,e) - \sum_{k=1}^{N} \alpha_k \{w^T \varphi(x_k) + b + e_k - y_k\} \quad \ldots (3.30)$$

with following conditions of optimality

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k = 0 \\ \dfrac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, \qquad k = 1,2,\ldots,N \\ \dfrac{\partial L}{\partial \alpha_k} = 0 \rightarrow w^T \varphi(x_k) + b + e_k - y_k = 0, \quad k = 1,2,\ldots,N \end{cases} \quad \ldots (3.31)$$

Solution will be $\begin{bmatrix} 0 & \vec{1}^T \\ 1 & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}$ … (3.32)

with $y = [y_1,\ldots,y_N], \vec{1} = [1,\ldots,1], \alpha = [\alpha_1,\ldots,\alpha_N]$. By applying Mercer's condition

$$\Omega_{kl} = \varphi(x_k)^T \varphi(x_l) = K(x_k, x_l), \quad k,l = 1,\ldots,N.$$

The final LS-SVR model can be written as

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x) + b \quad \ldots (3.33)$$

The above mentioned LS-SVR model having two parameters *i.e*. regularization parameter and the kernel function parameter.

In the present study, Ɛ-SVR, specialised form of least squares SVR model with Radial basis function (RBF) kernel was used for nonlinear mapping of dataset.

3.2.4.2 Kernel function:

SVR uses some set of mathematical equations to transform the input dataset into required form. These set of equations is known as Kernel Function. Mathematically a Kernel function can be represented as

$$K(x, y) = \begin{matrix} 1 \; when < f(x), f(y) > \\ 0 \quad otherwise \end{matrix} \qquad \ldots (3.34)$$

Here *K* is the kernel function, *x, y* are *n* dimensional inputs. f is a map from *n*-dimension to *m*-dimension space and <*x*,*y*> denotes the dot product of *x* and *y*. SVR uses kernel function to transform the nonlinearly separable input space into multidimensional feature space. There are different kernel functions based on the algorithms of SVR. Some commonly used kernel functions are

Polynomial kernel: $\phi(x_i, x_j) = (x_i x_j + 1)^d$

RBF kernel: $\phi(x_i, x_j) = \exp(-\lambda \left\| x_i - x_j \right\|^2)$

Sigmoid kernel: $\phi(x_i, x_j) = \tanh(\alpha x_i^T x_j + c)$

Linear kernel: $\phi(x_i, x_j) = 1 + x_i x_j + x_i x_j \min(x_i x_j) - \frac{x_i + x_j}{2} \min(x_i x_j)^2 + \frac{1}{3} \min(x_i x_j)^3$

here x_i and x_j are classes of variables, *d*= degree of polynomial, λ =penalty parameter, α and *c* are some constants.

3.2.5 Proposed EMD-SVR ensemble learning paradigm

Ensemble method is a machine learning approach which combines multiple base models to produce an optimal predictive model. The proposed EMD-SVR consists of three steps

defined in figure 3.5. In the first step, original nonlinear and nonstationary dataset is

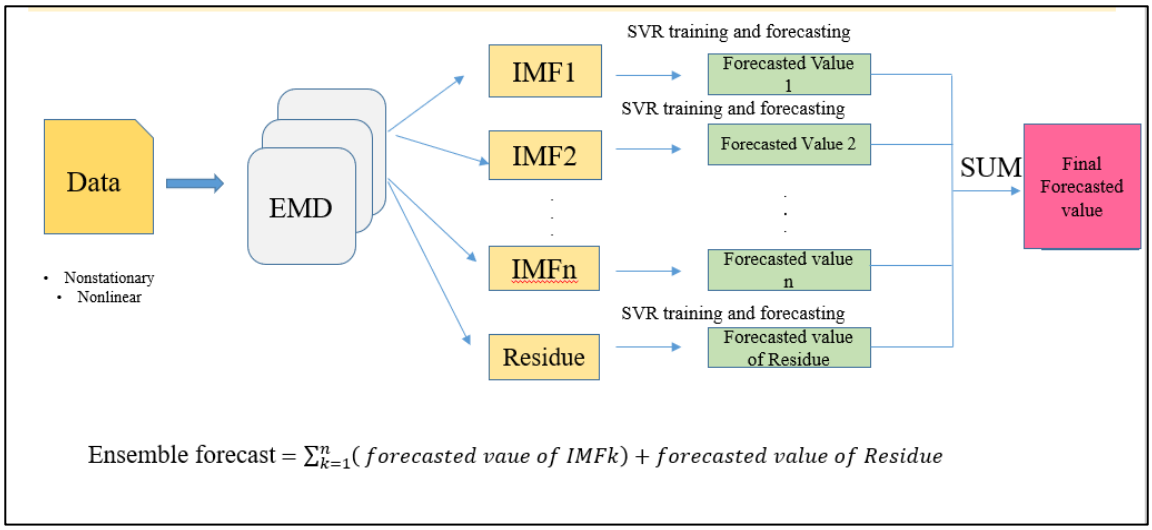


Figure 3.5: Flowchart of proposed EMD-SVR ensemble learning paradigm

decomposed into a finite and often small numbers of independent sub-series by EMD technique. This sub-series contain *n* intrinsic mode functions (IMFs) and a final residue. Secondly, these IMFs and residue is modelled and predicted through SVR. Then, all the forecasted values of the IMFs and residue are summed up to produce ensemble forecast for the original series.

3.2.6 Co-integration

In time series framework, co-integration acts as an alternative to check the spurious regression. For a nonstationary timeseries data, application of linear regression model may sometime cause the spurious regression. For integrated *I(1)*, Granger and Newbold (1974) demonstrated that de-trending does not work to eliminate the problem of spurious regression. Two series with *I(1)* trends can be co-integrated only if there is a genuine relationship between the two.

A nonstationary timeseries y_t with no deterministic component becomes stationary, invertible, ARMA representation after differencing *d* times, is said to be integrated of order *d*, denoted as $y_t \sim I(d)$ (Engle and Granger, 1987). In general, the components of the vector y_t , are said to be co-integrated of order *d*, *b*, denoted $y_t \sim CI(d,b)$, if (i) all components of y_t, are $I(d)$; (ii) there exists a vector α (≠0) so that $z_t = \alpha' y_t \sim I(d-b)$,

b >0. The vector α is called the co-integrating vector (Engle and Granger, 1987). Two time series x_t and y_t are said to be cointegrated if they are individually integrated of order *d (I(d))* and they share a common stochastic trend, so that a linear combination of both variables is *I (d-b)* with b > 0 (Engle and Granger, 1987).

For a two variable system, a typical linear Vector Error Correction Model (VECM) defines the change in one variable to past equilibrium errors, as well as to past changes in both variables. For a multivariate system, a vector time series y_t, has an error correction representation if it can be expressed as

$$A(B)(1-B)y_t = -\gamma z_{t-1} + u_t \qquad \ldots (3.35)$$

where *B* is backshift operator and u_t is a stationary multivariate disturbance, with *A(0) =I, A(1)* has all elements finite, $z_t = \alpha' y_t$ and $\gamma \neq 0$. ECMs directly estimate the speed at which a dependent variable returns to equilibrium after a change in other variables (Engle and Granger, 1987).

Nonlinear vector error correction model for *n×1* vector of *I(1)* stochastic processes z_t can be defined as follows (Kapetanios *et al*., 2006)

$$\Delta \mathbf{z}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{z}_{t-1} + g(\boldsymbol{\beta}'\mathbf{z}_{t-1}) + \sum_{i=1}^{p} \Gamma_i \Delta \mathbf{z}_{t-1} + \boldsymbol{\varepsilon}_t, \qquad t = 1, 2, \ldots, T$$

where $\boldsymbol{\alpha}$ *(n×r),* $\boldsymbol{\beta}$ *(n×r)* and $\boldsymbol{\Gamma}_i$ (*n×n*) are parameter matrices with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of full column rank. *g* is a nonlinear function.

3.2.6.1 Estimation and testing of co-integration

- **Engle and Granger approach** (Engle and Granger, 1987)

To explain the Engle-Granger testing procedure, suppose that two variables y_t and x_t are believed to be integrated of order 1 and to determine whether there exists an equilibrium relationship between the two. Engle and Granger (1987) proposed a four step procedure to determine if two *I*(1) variables are co-integrated of order *CI*(1,1).

(1) Pretest the Variables for their Order of Integration

By definition, the necessary requirement for co-integration is that two variables must be integrated of same order. The Augmented Dickey-Fuller (ADF) test and PP test can be used to infer the number of unit roots in each of the variables. If both variables are stationary, it is not necessary to proceed since standard time series methods apply to stationary variables. If the variables are integrated of different orders, it is possible to conclude they are not co-integrated.

(2) Estimation of long-run equilibrium relationship

If, both variables y_t and x_t are *I*(1), then estimate the long-run equilibrium relationship in the form

$$y_t = \beta_0 + \beta_1 x_t + e_t \qquad \ldots (3.36)$$

If the variables are co-integrated, an ordinary least squares regression yields a super-consistent estimator of the co-integrating parameters β_0 and β_1.

In order to determine if the variables are actually co-integrated, denote the residual sequence from Eq. (3.36) by $\hat{e}_t$. Thus, the $\hat{e}_t$ series are the estimated values of the deviations from the long-run relationship. If these deviations are found to be stationary, then y_t and x_t sequences are co-integrated of order (1,1). It would be convenient if we could perform a Dickey-Fuller test on these residuals to determine their order of integration. Consider the autoregression of residuals:

$$\Delta\hat{e}_t = a_1\hat{e}_{t-1} + \varepsilon_t \qquad \ldots (3.37)$$

Since the $\hat{e}_t$ sequence is a residual from a regression equation, there is no need to include an intercept term; the parameter of interest in Eq. (3.37) is a_1. If we cannot reject the null hypothesis $a_1 = 0$, we can conclude that the residual series contains a unit root. Hence, we conclude that the y_t and x_t sequences are not co-integrated. Given that y_t and x_t were both found to be *I*(1) and that the residuals are stationary, we can conclude that the series are co-integrated of order (1,1).

If the residuals of Eq. (3.37) do not appear to be white noise, an augmented form of the test can be used instead of Eq. (3.37). Suppose that diagnostic checks indicate that the ε_t of Eq. (3.37) exhibits serial correlation. Instead of using the results from Eq. (3.37), estimate the autoregression:

$$\Delta\hat{e}_t = a_1\hat{e}_{t-1} + \sum_{i=1}^{n} a_{i+1}\Delta\hat{e}_{t-i} + \varepsilon_t \qquad \ldots (3.38)$$

Again, if we reject the null hypothesis $a_1 = 0$, we can conclude that the residual sequence is stationary and that the variables are co-integrated.

(3) Estimate the error-correction model

If the variables are co-integrated, the residuals from the equilibrium regression can be used to estimate the error-correction model. If y_t and x_t are $CI(1,1)$, the variables have the error-correction form

$$\Delta y_t = \alpha_1 + \alpha_y[y_{t-1} - \beta_1 x_{t-1}] + \sum_{i=1}^{n}\alpha_{11}(i)\Delta y_{t-i} + \sum_{i=1}^{n}\alpha_{12}(i)\Delta x_{t-i} + \varepsilon_{yt} \qquad \ldots (3.39)$$

$$\Delta x_t = \alpha_2 + \alpha_x[y_{t-1} - \beta_1 x_{t-1}] + \sum_{i=1}^{n}\alpha_{21}(i)\Delta y_{t-i} + \sum_{i=1}^{n}\alpha_{22}(i)\Delta x_{t-i} + \varepsilon_{xt} \qquad \ldots (3.40)$$

where

β_i = the parameters of the co-integrating vector,

ε_{yt} and ε_{zt} are white noise disturbances, and

$\alpha_1, \alpha_2, \alpha_y, \alpha_z, \alpha_{11}(i), \alpha_{12}(i), \alpha_{21}(i),$ and $\alpha_{22}(i)$ are all the parameters .

Engle and Granger (1987) proposed a cross-equation restrictions involved in the direct estimation of Eqs. (3.39) and (3.40). The magnitude of the residual $\hat{e}_{t-1}$ is the deviation from long-run equilibrium in period $(t-1)$. Hence, it is possible to use the saved residuals $\{\hat{e}_{t-1}\}$ as an instrument for the expression $y_{t-1} - \beta_1 x_{t-1}$ in Eqs. (3.39) and (3.40). Thus,

using saved residuals from the estimation of the long-run equilibrium relationship, the error-correcting model is estimated as

$$\Delta y_t = \alpha_1 + \alpha_y \hat{e}_{t-1} + \sum_{i=1}^{n} \alpha_{11}(i)\Delta y_{t-i} + \sum_{i=1}^{n} \alpha_{12}(i)\Delta x_{t-i} + \varepsilon_{yt} \quad \ldots (3.41)$$

$$\Delta x_t = \alpha_2 + \alpha_x \hat{e}_{t-1} + \sum_{i=1}^{n} \alpha_{21}(i)\Delta y_{t-i} + \sum_{i=1}^{n} \alpha_{22}(i)\Delta x_{t-i} + \varepsilon_{xt} \quad \ldots (3.42)$$

Other than the error correction term $\hat{e}_{t-1}$ in Eqs. (3.41) and (3.42) constitute VAR in first differences. All of the procedures developed for a VAR apply to the system represented by the error-correction equations. More importantly, OLS is an efficient estimation procedure since each equation contains the same set of regressors. Since all terms in Eqs. (3.41) and (3.42) are stationary, the test statistics used in traditional VAR analysis are appropriate for Eqs. (3.41) and (3.42).

(4) Assess model adequacy

There are several procedures that can help determine whether the error-correction estimated model is appropriate.

(a) Adequacy of the model can be evaluated by performing diagnostic checks to determine whether the residuals of the error-correction equations have approximate white noise. If the residuals are serially correlated, lag length may be too short. In that case re-estimate the model using appropriate lag lengths that yields serially uncorrelated errors.

(b) The speed of adjustment coefficients α_y and α_x are of particular interest in that they have important implications for the dynamics of the system. If we focus on Eq. (3.42), it is clear that for any given value of $\hat{e}_{t-1}$, a large value of α_x is associated with a large value of Δx_t. If α_x is zero, the change in x_t does not at all respond to the deviation from long-run equilibrium in $(t-1)$.

- **Phillips–Ouliaris approach** (Phillips and Ouliaris, 1990)

Phillips and Ouliaris introduced two residual-based tests, namely the variance ratio test and the multivariate trace statistic (Phillips-Ouliaris, 1990). It is an improved version of Engle-Granger approach, where supplementary variability among the variables are considered. These residual-based tests are used in the same way as the unit root tests, but the data are the residuals from the co-integrating regression. These tests seek to test a null hypothesis of no co-integration against the alternative of the presence of co-integration using scalar unit root tests applied to the residuals. Phillips-Ouliaris methods are based on residuals (differences between the observed and expected values) of the first order autoregression, AR (1) equation. The multivariate trace statistics has the advantage over the variance ratio test in that it is invariant to normalisation of the co-integrating variable, that is, whichever variable is taken to be the dependent variable, the test will yield the same results.

- **Johansen's Test**

Johansen (1992) proposed the method of testing co-integration of multivariate time series. Johansen test is another improved version of Engle-Granger approach based on likelihood-ratio tests. It is either eigen value based or trace based. In general, if there are *n* variables which all have unit roots, there are at most *n – 1* co-integrating vectors. All co-integrating vectors can be estimated through the Johansen test. The hypothesis of the test is similar to that of the Engle and Granger.

For a time series y_t vector auto-regression in levels with the constant can be written as

$$y_t = \sum_{i=1}^{k} A_i y_{t-i} + u_t \qquad \ldots (3.43)$$

where A_i is the coefficients of the model and u_t is the error term at time *t*.

For $k > 1$ the VAR model can be written as

$$\Delta y_t = \mathbf{\Pi} y_{t-1} + \sum_{i=1}^{k-1} \mathbf{\Pi}_i \Delta y_{t-i} + u_t \qquad \ldots (3.44)$$

For *k=1*, the simple model is $\Delta y_t = \mathbf{\Pi} y_{t-1} + u_t$.

The matrix $\mathbf{\Pi}$ can be expressed in terms of vector or matrix of adjustment parameters $\boldsymbol{\alpha}$ and the vector or matrix of cointegrating vectors $\boldsymbol{\beta}$ as

$$\mathbf{\Pi} = \boldsymbol{\alpha}\boldsymbol{\beta}' \qquad \ldots (3.45)$$

In the Eq. 3.44 we check whether $\mathbf{\Pi} = 0$ or not by rank of $\mathbf{\Pi}$.

When rank ($\mathbf{\Pi}$) = 0, the variables are not cointegrated.

When the variables are co-integrated, rank ($\mathbf{\Pi}$) $\neq 0$ *i.e.* rank ($\mathbf{\Pi}$) = the number of co-integrating vectors.

Test of the maximum eigenvalue is a likelihood ratio test. The test statistic is:

$$LR(r_0, r_0 + 1) = -T \ln(1 - \lambda_{r_0+1}) \qquad \ldots (3.46)$$

where *T* is number of observations and λ_i are the eigen value of matrix $\mathbf{\Pi}$. It tests the null hypothesis rank ($\mathbf{\Pi}$) = r_0 against alternative hypothesis rank ($\mathbf{\Pi}$) = $r_0 + 1$.

The test statistics of trace based Johansen test is:

$$LR(r_0, n) = -T \sum_{i=r_0+1}^{n} \ln(1 - \lambda_i) \qquad \ldots (3.47)$$

where *T* is no. of observations and λ_i are the eigen value of matrix $\mathbf{\Pi}$.

In the present study, the Johansen test based on both eigen and trace has been used to find the Error Correction Term (ECT) and co-integrating vectors.

3.2.7 Artificial neural network model

Artificial neural networks are non-linear, data driven self-adaptive approach as opposed to the traditional model based methods (Jha *et al*., 2009). ANNs can identify and learn correlated patterns between input data sets and corresponding target values. ANNs imitate the learning process of the human brain and can process problems involving non-linear and complex data even if the data are imprecise and noisy. Thus, they are ideally suited

for the modelling of agricultural data which are known to be complex and often non-linear. The block diagram of a model of neuron is shown in figure 3.6 on the basis of designing an ANN. Main elements of a neural networks are-

Input layer: Input layer receive the input information (feature) in different form like text, image, number *etc*.

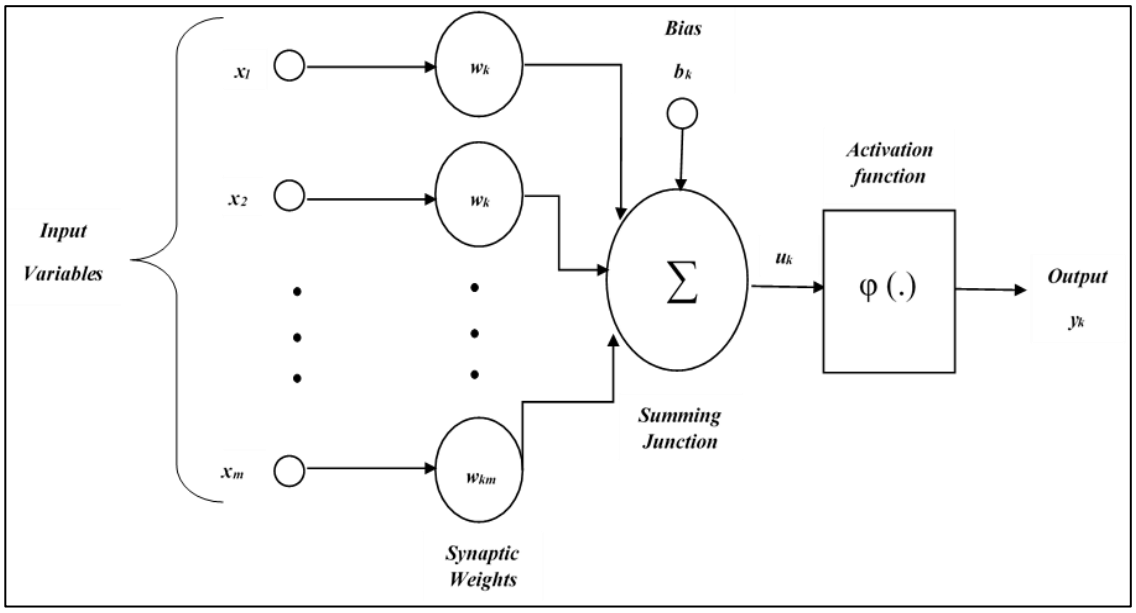


Figure 3.6: Block diagram of model of a neuron

Hidden layer: Hidden layer mainly responsible for all sorts of computation on the input information. The numbers of hidden layer varies algorithm to algorithm.

Output layer: Output layer provides the summarised results after the computation.

Kaastra and Boyd (1996) showed that to form a typical ANN mainly one input, one output and one or at most two hidden layers are sufficient for a univariate time series data.

Haykin (1999) stated mathematically that a neuron *k* can be defined by the following equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad \ldots (3.48)$$

$$y_k = \varphi(u_k + b_k) \qquad \ldots (3.49)$$

Here bias (b_k), has the effect of increasing or lowering the net input of the activation function. x_1, x_2,.. ., x_m are the inputs; w_{k1}, $w_{k2,}$. . ., w_{km} are the weights of the neuron *k*; u_k is the linear combiner output due to input variables; $\varphi(.)$ is the activation function; y_k is

the output of the neuron the weight attached to the connection from j^{th} hidden node to the output node.

3.2.7.1 Activation function

Activation function is a differentiable function that is used for smoothing the result of the cross product of the covariate or neurons and the weights. In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs. Some commonly used activation functions for input *x* are

Sigmoid or Logistic activation function: $(x) = \frac{1}{1+e^{-x}}$; range (0,1)

Hyperbolic tangent function: $(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$; range (-1,1)

Gaussian activation function: $(x) = e^{-x^2}$; range (0,1]

Rectified Linear units (ReLu) : $f(x) = {}^{x\ for\ x>0}_{0\ for\ x\leq 0}$; range [0,∞)

Sinusoid activation function: $(x) = \sin(x)$; range [-1,1]

3.2.7.2 Training Algorithm of ANN

Backpropagation denotes “backward propagation of errors" using gradient descent approach. Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights. It is a generalization of the delta rule for perceptrons to multilayer feedforward neural networks (Bishop, 1995). The backpropagation algorithm can be implemented under the following components:

1. Data should contain input-output pair $(\vec{x}_i, \vec{y}_i)$, where $\vec{x}_i$ is the input and $\vec{y}_i$ is the desired output. For N data of $X = \{(\vec{x}_1, \vec{y}_1), ..., (\vec{x}_N, \vec{y}_N)\}$.
2. Need a feedforward neural network. Let the parameters of the network be denoted by θ. The parameters of interest in backpropagation are the weights w_{ij}^k, node *j* in layer l_k and node *i* in layer l_{k-1} and bias b_i^k the bias for node *i* in layer l_k.

3. Error function $E(X,\theta)=\frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i-y_i)^2$; where $\hat{y}_i$ are the computed output of the network on input $\vec{x}_i$ and y_i is the target value for input-output pair $(\vec{x}_i,\vec{y}_i)$.

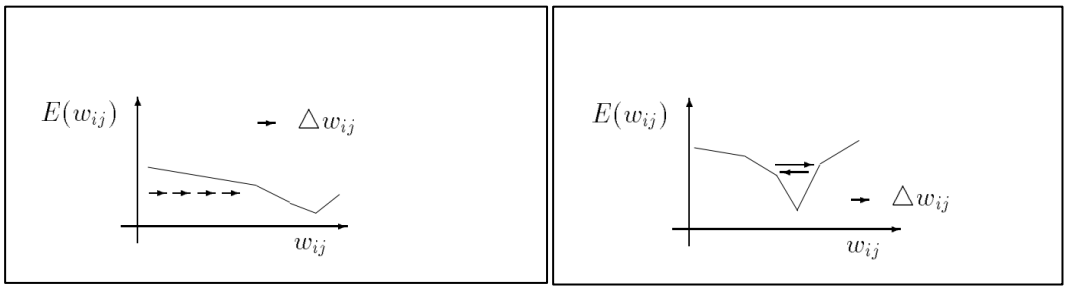
The basic idea of the backpropagation learning algorithm is the repeated application of the chain rule to compute the influence of each weight in the network with respect to an arbitrary error function *E*. (Riedmiller and Braun, 1993).

$$\frac{\partial E}{\partial w_{ij}}=\frac{\partial E}{\partial a_i}\frac{\partial a_i}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \qquad \ldots (3.50)$$

where w_{ij} is the weight from neuron *j* to neuron *i*, a_i is the activation value and net_i the weighted sum of the inputs of neuron *i*. After deriving the partial derivatives, simple gradient descent approach is used to minimize the error function *E* as:

$$w_{ij}(t+1)=w_{ij}(t)-\alpha\frac{\partial E}{\partial w_{ij}}(t) \qquad \ldots (3.51)$$

where α is the learning rate.



(*a*) (*b*)

Figure 3.7: Pictorial representation of learning rate value effect (*a*) small value and (*b*) Large value

The learning rate mainly scale the partial derivates. The choice of the learning rate is an important criterion for the time needed for convergence. If the set is too small, too many steps are required to converge into acceptable solution (figure 3.7). On other side large learning rate leads to fast convergence into acceptable solution. The traditional gradient decent based backpropagation has certain limitations like low convergence rate, parameter adaptation *etc*.

To overcome this problem Riedmiller and Braun (1993) proposed a new adaptive learning algorithm called Resilient Back Propagation (RPROP). It is based on 'Manhattan-learning' rule.

$$\Delta w_{ij} = \begin{cases} -\Delta_0 & ,if \ \frac{\partial E}{\partial w_{ij}} > 0 \\ +\Delta_0 & ,if \ \frac{\partial E}{\partial w_{ij}} < 0 \\ 0 & ,else \end{cases} \qquad \ldots (3.52)$$

where Δ_0 is the update-value and problem dependent which creates to adjust the weights. The RPROP algorithm extracts some more information about the topology of the error function and update the weights more appropriately. For each weight, personal update value Δ_{ij} is identified using local sight of the error function *E*. So, the new learning rule for update values,

$$\Delta w_{ij} = \begin{cases} \Delta_{ij}(t-1) * \eta^+ & ,if \ \frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \Delta_{ij}(t-1) * \eta^- & ,if \ \frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ \Delta_{ij}(t-1) & , else \end{cases} \qquad \ldots (3.53)$$

with $0 < \eta^- < 1 < \eta^+$; η is an increasing-decreasing function between 0 and 1.

The update-value is not influenced by the magnitude of the derivatives, but only by the behaviour of the sign of two succeeding derivatives. The partial derivative of the corresponding weight w_{ij} changes its sign each time. This indicates that the last update was too big and the algorithm has jumped over a local minimum. The advantage of RPROP is that it changes the value of the weight update directly. It only depends on the sign of the partial derivative without reference to its magnitude.

In the present study, Logistic function was used as an activation function and resilient backpropagation algorithm was used to adjust the weights in the multi-layered feedforward networks.

3.2.8 Proposed time-delay neural network (TDNN) model

Under the second objective a time-delay neural network model has been developed. For this, traditional co-integration or error correction model proposed by Granger (1987) has been used. Engle and Granger (1981) concentrated on the classical *I(0)/I(1)* co-integration framework where *d=1* and *b=1* and proposed a two-step method, where the model

$$\begin{cases} (y_t - \beta x_t - \alpha) = \varepsilon_{1,t} \\ \Delta x_t = \varepsilon_{2,t} \end{cases} \quad \ldots (3.54)$$

where Δ indicates first differences, $\varepsilon_{1,t}$ and $\varepsilon_{2,t}$ are zero mean *I(0)* residuals and is $\vec{\beta} = (1, -\alpha, -\beta)$ the co-integration vector with an intercept α. In the second step the error correction model

$$\Delta y_t = \phi_0 + \sum_{j}^{p} \phi_j \Delta y_{t-j} + \sum_{h=0}^{q} \theta_h \Delta x_{t-h} + \lambda \varepsilon_{1,t-1} + u_t \quad \ldots (3.55)$$

can be estimated with λ as the adjustment coefficient, ϕ_0 as constant and ϕ_j, θ_h as coefficients of short-run relationship with *p* lags of endogenous and *q* lags of exogenous variable. Thus the *Eq. 3.54* can be written as:

$$\begin{cases} \Delta^{d-b}(y_t - \beta x_t - \alpha) = \varepsilon_{1,t} \\ \Delta^{d} x_t = \varepsilon_{2,t} \end{cases} \quad \ldots (3.56)$$

where $\Delta^d = (1-B)^d$ and *B* is a backshift operator *i.e.* $Bx_t = x_{t-1}$; *0<b<1*.

Nonlinear error correction model was defined by Kapetanios *et al*. (2006)

$$\Delta Y_t = \phi_0 + \Lambda \vec{\beta}' y_{t-1} + \Psi(\vec{\beta}' y_{t-1}) + \sum_{i}^{p} \phi_i \Delta y_{t-1} + U_t \quad \ldots (3.57)$$

Where Y_t is a matrix containing the time series y_t, ϕ_0 is a vector of constant, ϕ_i is a matrix of the coefficients of lagged endogenous variables, Λ is the adjustment coefficient and U_t is a matrix of white noise residuals and ψ is a nonlinear function. Instead of a

defined functional form of ψ a NN function η has been used. The modified equation can be written as:

$$\Delta^d Y_t = \eta(\Delta^d Y_{t-1}, ..., \Delta^d Y_{t-p}, \Delta^{d-b}\vec{\beta}' y_{t-1}) + U_t \quad \ldots (3.58)$$

Here all variables are assumed to be stationary.

The general expression of a TDNN with single hidden layer is given by (Jha and Sinha, 2014) -

$$\hat{y}_t = g\left(\alpha_0 + \sum_{j=1}^{q} \alpha_j f\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right)\right) \quad \ldots (3.59)$$

where $\hat{y}_t$ is the predicted value for y_t at time t, p input and q hidden nodes, i, j is i^{th} node of input layer, j^{th}node of hidden layer respectively, y_{t-i} ; $(i = 1,2, \ldots p)$ are network input nodes. β_{ij} $(j = 1, 2, \ . \ . \ ., q)$ refer to the connection weight between i^{th} and j^{th}neuron. α_j refer to the weight between j^{th} neuron of hidden node and output node. α_0 and β_{0j} are bias terms for output layer and j^{th} hidden node. f and g are respectively hidden and output layer activation functions, mainly Logistic $f(v_j) = \frac{1}{1+e^{-v_j}}$ and g is an identity function. The above mentioned TDNN model will be used to for forecasting.

3.2.9 Multivariate Adaptive Regression Spline (MARS) model

Multivariate adaptive regression spline is a nonparametric technique developed by Friedman (1991). The main purpose of this technique is to predict the values of a continuous dependent variable, $y(n\times 1)$ from a set of independent explanatory variables, $X(n\times p)$. The MARS model can be represented as:

$$y = f(X) + e \quad \ldots (3.60)$$

where e is an error vector of dimension $(n\times p)$. $f(x)$ is basically sum of intercept and basis functions generated by MARS algorithm.

MARS is based on local regression modelling. It uses spline functions to approximate complex nonlinear relations. MARS divides whole data region into several subdomains. At each of the subdomain, a regression line is used to fit a response variable. For a spline of degree q, each segment is a polynomial function. MARS uses two-sided truncated

power functions as spline basis functions. These function can be described by the following equations (Sekulic and Kowalski, 1992):

$$[-(x-t)_+^q = \left\{ \begin{array}{ll} (t-x)^q, & \text{if } x < t, \\ 0, & \text{otherwise} \end{array} \right\} \quad \ldots (3.61)$$

$$[+(x-t)_+^q = \left\{ \begin{array}{ll} (t-x)^q, & \text{if } x \geq t, \\ 0, & \text{otherwise} \end{array} \right\} \quad \ldots (3.62)$$

where $q(\geq 0)$ is the power of splines to determine the degree of smoothness of the resultant function estimate.

The MARS model for a dependent (outcome) variable *y*, and *M* terms, can be summarized in the following equation ((Friedman and Roosen, 1995):

$$\hat{y} = \hat{f}_M(\vec{X}) = c_0 + \sum_{m=1}^{M} c_m B_m(\vec{X}) \quad \ldots (3.63)$$

where $\hat{\vec{y}}$ is the dependent variable predicted by the MARS model, c_o is a constant, $B_m(\vec{X})$ is the m^{th} basis function, and c_m is the coefficient of the m^{th} basis function. The basis function may be a single spline basis function. Both the variables to be introduced into the model and the knot positions for each individual variable have to be optimized. For a data set $\vec{X}$ containing *n* objects and *p* explanatory variables, there are $N = n \times p$ pairs of spline basis functions, given by Eqs. (3.61) and (3.62), with knot locations x_{ij} ($i = 1, 2, \ldots, n; j = 1, 2, \ldots, p$). In the present study, x=206 entries are exotic collections and 312 are indigenous collections including 59 breeding lines, *p*=9 and *n*=518.

In this study, we followed a two-step procedure to construct the final model (figure 3.8). First, a two-at-a-time forward stepwise procedure is implemented in order to select the consecutive pairs of basis functions of the model (Friedman and Roosen, 1995). This forward stepwise selection process leads to a very complex and over-fitted model. The fitted model has poor predictive abilities for new objects.

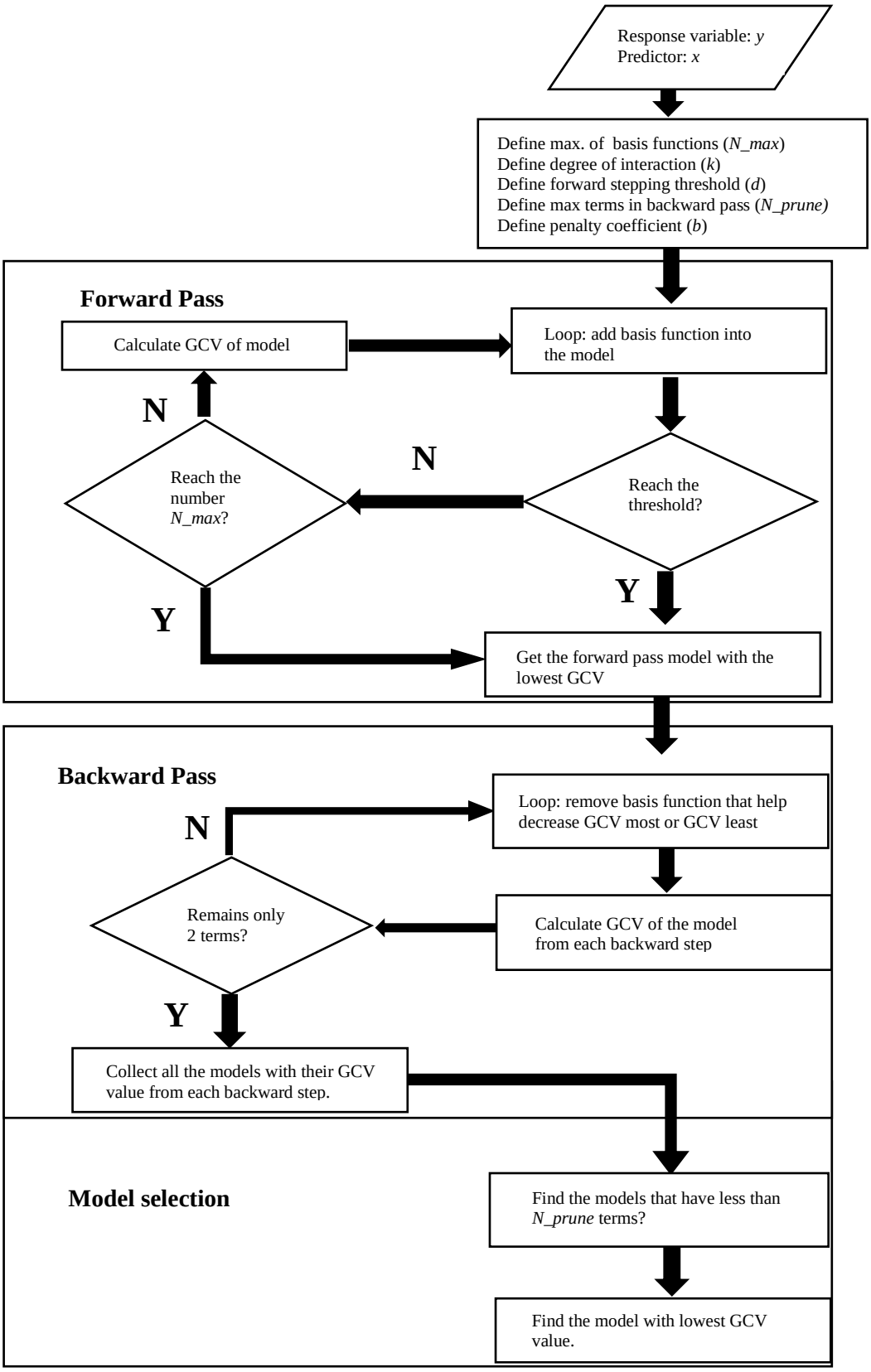


Figure 3.8: Flowchart of the MARS modelling process

Second a backward stepwise procedure is used to improve the prediction. This procedure removed the redundant basis functions one at a time. Moreover, to determine which basis functions should be included in the model.

MARS utilizes Generalized Cross-Validation (GCV) (Sekulic and Kowalski, 1992). The GCV is the mean squared residual error divided by a penalty which is dependent on model complexity. The GVC criterion is defined as follows

$$GCV(M) = \frac{\frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{f}_M(\vec{x}_i))^2}{(1-\frac{C(M)}{n})^2} \quad \ldots (3.64)$$

where *C(M)* is a complexity penalty that increases with the number of basis functions in the model and which is defined as:

$$C(M) = (M+1) + dM \quad \ldots (3.65)$$

where *M* is the number of basis functions and the parameter *d* is a penalty for each basis function included into the model. *d* is a smoothing parameter.

3.2.9 Proposed MARS based hybrid model approach

In the given study, MARS based hybrid models have been developed by combining it with ANN and SVM. The basic idea was the variable selection ability of MARS algorithm and prediction ability of ANN/SVM. First, MARS algorithm is used to extract important variables among the independent variables. Then these selected variables were taken as input variables to predict dependent variable.

MARS model: $\hat{\tilde{y}} = c_0 + \sum_{m=1}^{M} c_m B_m(\vec{X})$

The parameters are same as defined in *Eq.* 3.63

Modified ANN model: $\tilde{u}_k = \sum_{j=1}^{m} w_{kj}\tilde{x}_j$

$$\tilde{y}_k = \varphi(\tilde{u}_k + b_k) \quad \text{... (3.66)}$$

Here $\tilde{x}_j$ are the selected important variables through MARS. $\tilde{u}_k$ is the linear combiner output due to input variables; $\varphi(.)$ is the activation function; $\tilde{y}_k$ is the output of the neuron the weight attached to the connection from j^{th} hidden node to the output node.

Modified SVM model: $\tilde{y} = \phi(\tilde{x}) = W.\varphi(\tilde{x}) + b$ ... (3.67)

Here $\tilde{x}_j$ are the selected important variables through MARS. $\tilde{y}_k$ is the output of modified SVM model. Other parameters are same as defined in Eq. (3.25). Schematic diagram was also presented in figure 3.9.

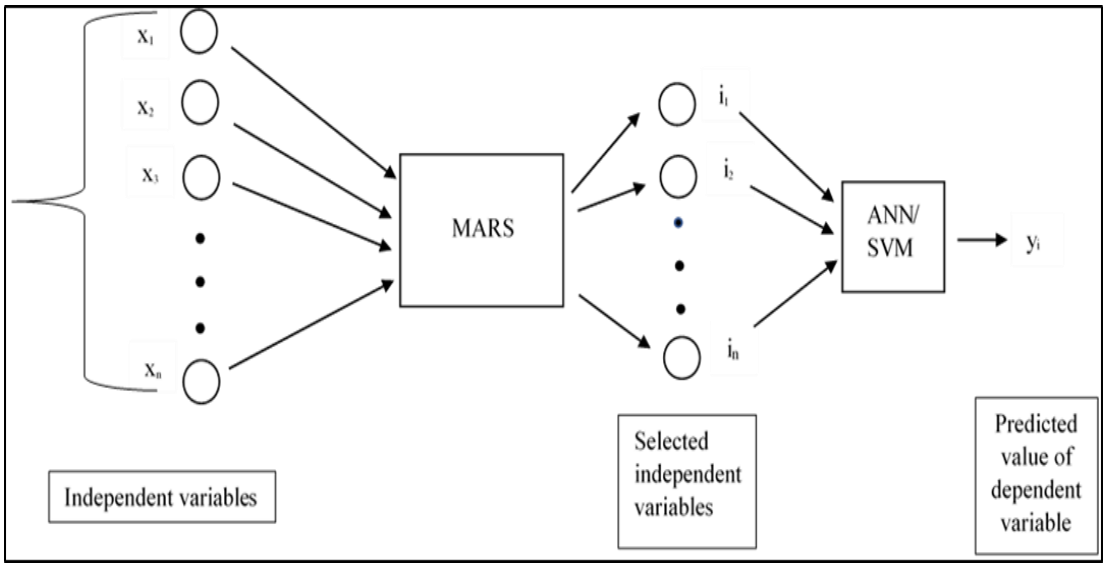


Figure 3.9: MARS based ANN/SVM hybrid framework

3.3 Forecasting evaluation criteria

Forecasting accuracy is crucial for finding out the practicability of developed hybrid models. To compare the model accuracy of the three developed hybrid models, the following evaluation criteria have been used.

3.3.1 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}$$

here y_i and $\hat{y}_i$ are the actual value and predicted value of response variable.

3.3.2 Mean Absolute Deviation (MAD)

$$MAD = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N}$$

here y_i and $\hat{y}_i$ are the actual value and predicted value of response variable.

3.3.3 Mean Absolute Percentage Error (MAPE)

MAPE is defined as

$$MAPE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i| / y_i}{N}$$

3.3.4 Maximum Error (ME)

$$ME = \max \sum_{i=1}^{N}|y_i - \hat{y}_i|$$

here y_i and $\hat{y}_i$ are the actual value and predicted value of response variable.

3.4 Normality assumption of residuals

Detailed analysis of residuals is important for checking model adequacy. Normality of the residuals are tested with the help of many test like, in this study, we used J.B. test.

3.4.1 Jarque–Bera test

Jarque and Bera proposed the goodness-of-fit test (Jarque and Bera, 1987). The test is negative in nature. The test statistic of the test is

$$JB = \frac{n}{6}(S^2 + \frac{1}{4}(K-3)^2)$$

Here n = numbers of observations. S= skewness of sample and K= kurtosis of sample.

$$s=\frac{\hat{\mu}_3}{\hat{\sigma}^3}=\frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^3}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^{\frac{3}{2}}} \text{ and } K=\frac{\hat{\mu}_4}{\hat{\sigma}^4}=\frac{\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^4}{\left(\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2\right)^2}$$

here $\hat{\mu}_3$ and $\hat{\mu}_4$ are the estimates of third and fourth central moments, respectively, $\bar{x}$ is the sample mean, and $\hat{\sigma}^2$ is the estimate of the second central moment, the variance.

The hypothesis of the test are

H_0: The sample follows normal distribution.

H_1: The sample does not follow normal distribution.

3.5 Lag section criterion

The following criteria are used for selecting appropriate lags for VAR model.

3.5.1 Akaike's Information Criterion (AIC)

AIC can be defined as:

$$\text{AIC} = -\ln L + p$$

where L is the likelihood function for model with p parameters and ln is natural logarithm.

3.5.2 Schwarz Information Criterion (SIC)

The Bayesian information criterion (BIC) or Schwarz criterion (also SBC, SBIC) is another criterion for lag selection. The SIC can be written as:

$$\text{SIC} = -2L_p + p\ln n$$

where n is the sample size, L_p is the maximized log-likelihood of the model and p is the number of parameters in the model and ln is natural logarithm.

3.6 Accuracy test for the proposed hybrid models

In the present investigation, the accuracy of the proposed hybrid models are compared with the generic forecasting methodologies. Diebold-Mariano (DM) test is used to evaluate the accuracy of proposed hybrid models.

3.6.1 Diebold Mariano (DM) test

Diebold and Mariano (1995) proposed the test which is based on error function. Let y_t is a time series dataset and h-step forecasted value of the dataset from i^{th} model the is $\hat{y}_{i,t}^{h}$ (i = 1, 2, 3,…m), m is the number of the forecasting models. The h-step forecasting errors, $e_{i,t}^{h}$ is:

$$e_{i,t}^{h} = y_t^{h} - \hat{y}_{i,t}^{h}$$

The accuracy of each forecast is measured by the loss function (L).

Diebold Mariano test statistic can be written as (Das *et al.*, 2016):

$$DM = \frac{\bar{d}}{\frac{\sqrt{2\pi \hat{f}_d(0)}}{T}}$$

Here d = $L(e_{1t})$ − $L(e_{2t})$, $\bar{d}$ =average distance between loss function. $2\pi \hat{f}_d(0)$ is a consistent estimator of the asymptotic variance of $\sqrt{Td}$ and T = number of data points.

The test hypothesis are

$$H_0 : E\left[L(e_{t+h}^{1})\right] = E\left[L(e_{t+h}^{2})\right]$$

$$H_1 : E\left[L(e_{t+h}^{1})\right] \neq E\left[L(e_{t+h}^{2})\right]$$

If the p-value lower than .01 and 0.05, test is significant and reject the null hypothesis.

CHAPTER IV

RESULTS AND DISCUSSION

This chapter contains the detailed results of the research work carried out to accomplish three specific objectives of the study. The research work was performed on three different types of agricultural datasets. The detail of the datasets is already discussed in the previous chapter 3.1. The first portion of the present chapter highlights the results of EMD based SVR model, then the results of the co-integration based TDNN model are discussed. Finally, the results of MARS based hybrid models are highlighted.

4.1 Results of EMD based SVR model

Under the first objective, we proposed a hybrid method by combining empirical mode decomposition with support vector regression for agricultural price forecasting. The developed methodology was implemented in R Studio. The packages "EMD" (Kim and Oh, 2009) and "e1071" (Meyer *et al*., 2018) were customized and used for EMD and SVR fitting respectively. At first, the behaviour of the dataset was studied. The descriptive statistics, stationarity test and normality of data were analysed (Table 4.1). The data had a total of 290 points consisting of a maximum value of 971.3 and minimum value 136.5. The statistics obtained through Augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) tests were insignificant *i.e.* null hypothesis of unit root test cannot be refused. The results indicated that the given dataset was nonstationary. Jarque-Bera test (Table 4.1) indicated the nonnormality of data.

Table 4.1: The descriptive statistics, stationarity test and normality test of data

Observations	**Minimum**	**Maximum**	**Mean**	**Standard deviation**	**Skewness**	**Kurtosis**
290	136.5	971.3	412.8	201.415	1.042	3.252
Augmented Dickey-Fuller Test (p value)			**Phillips-Perron Test (p value)**		**Jarque-Bera test (p value)**	
0.924			0.821		<0.01	

Brock-Dechert-Scheinkman (Brock *et al.,* 1996) test was used in the dataset for checking the nonlinearity of data. The results of the BDS test (Table 4.2) indicated that the test statistics were far bigger than the critical values. It provided an evidence to reject the null hypothesis that the price series is linearly dependent. The results obtained from various tests revealed that the monthly chili WPI dataset to be nonlinear and nonstationary. These characteristics of the dataset enabled us to implement and evaluate the performance of the proposed EMD-SVR model with popular existing models.

Table 4.2: Brock- Dechert-Scheinkman (BDS) test

Embedding dimension				**Conclusion**
2		**3**		
Statistics	**Probability**	**Statistics**	**Probability**	Nonlinear
66.089	< 0.001	106.523	< 0.001	
51.709	< 0.001	62.327	< 0.001	
40.525	< 0.001	42.372	< 0.001	
35.129	< 0.001	34.132	< 0.001	

4.1.2 EMD decomposition

The EMD, as an adaptive decomposition technique is quite effective in extracting characteristic information from nonlinear and nonstationary time series. Detail of EMD are discussed in section 3.2.2. EMD methodology has been employed to decompose the series. The original series has been decomposed into four IMFs and one final residue using EMD (figure 4.1). It has been observed that the frequencies and amplitudes of IMFs were different and independent to each other. Thus, the different hidden oscillatory modes in the original datasets were separated by EMD. Each decomposed IMF contains certain characteristics of the dataset which needs to be modelled and forecasted using appropriate model.

Table 4.3: Unit root test of decomposed components of chili dataset

IMFs	**Phillips-Perron Test (p-value)**	**Augmented Dickey-Fuller Test (p-value)**	**Remarks**
1	<0.01	<0.01	Stationary
2	<0.01	<0.01	Stationary
3	<0.01	<0.01	Stationary
4	<0.01	<0.01	Stationary
Residue	0.99	0.985	Nonstationary

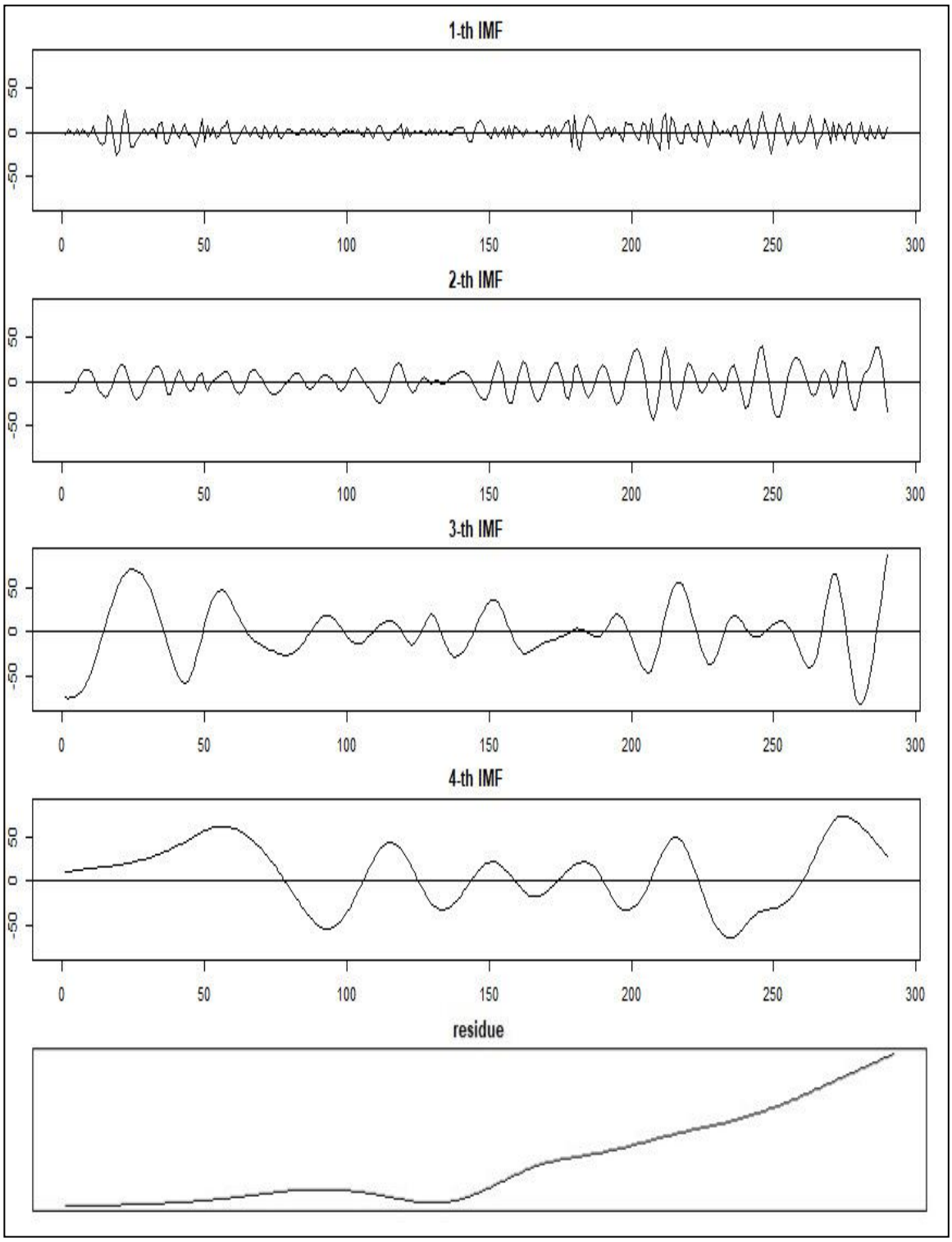


Figure 4.1: Decomposed components of monthly chili WPI

After decomposition, it is essential to check the stationarity of IMFs and residue (Table 4.3), the results of the test indicated that all IMFs were stationary. But the final residue was nonstationary because it was the remaining portion of the data which cannot be decomposed by EMD algorithm. As stationarity is one of important assumptions for forecasting, hence, the nonstationary residue cannot be used in forecasting. The residue was transformed into stationary by differencing. The stationary decomposed parts *i.e.* IMFs and the differenced residue were used for forecasting.

4.1.3 SVR training and forecasting

Each component (IMFs and residue) was modelled and forecasted by the SVR model defined in previous chapter section 3.2.4.1. The SVR model was preferred over other machine learning algorithms due to its capability to handle nonlinear systems as well as its suitability for small sample size. To begin with the implementation of SVR, we need to divide the dataset into training and testing sets. The training set is used for model building purpose whereas, the testing set allows us to understand the generalization ability of the developed model. In our study, we have used 80% of data as training set and the remaining 20% as testing set. The developed SVR model for each decomposed component (IMFs and residue) was used to forecast the respective components. Then all the forecasted values of IMFs and residue were summed up to get an ensemble forecast of the data. Iterative 8-step and 6-step ahead prediction was utilized to measure out-of-sample predictability of EMD-SVR model (Table 4.4). The model predicted one-step ahead each time and for the next time period, step prediction is added to the current output. In this study, we have employed a new variant of SVR called LS-SVR and have optimized the results by choosing appropriate kernel function. Before moving towards selection of the kernel function, it is important to find the optimal parameter combination of the model. We have used the grid search method to obtain the final combination of the parameters value. Next, we tried implementing the LS-SVR using Radial Basis Function (RBF), polynomial, linear and sigmoid kernel functions to our data. The best result was obtained using RBF kernel function. To overcome the problem of overfitting 10-fold cross-validation was also done. This measure ensures the appropriateness of the developed LS-SVR model for the data. Graphical representation of the forecasted values of each IMFs and the final residue values through SVR is given in figure 4.2. The red line indicates

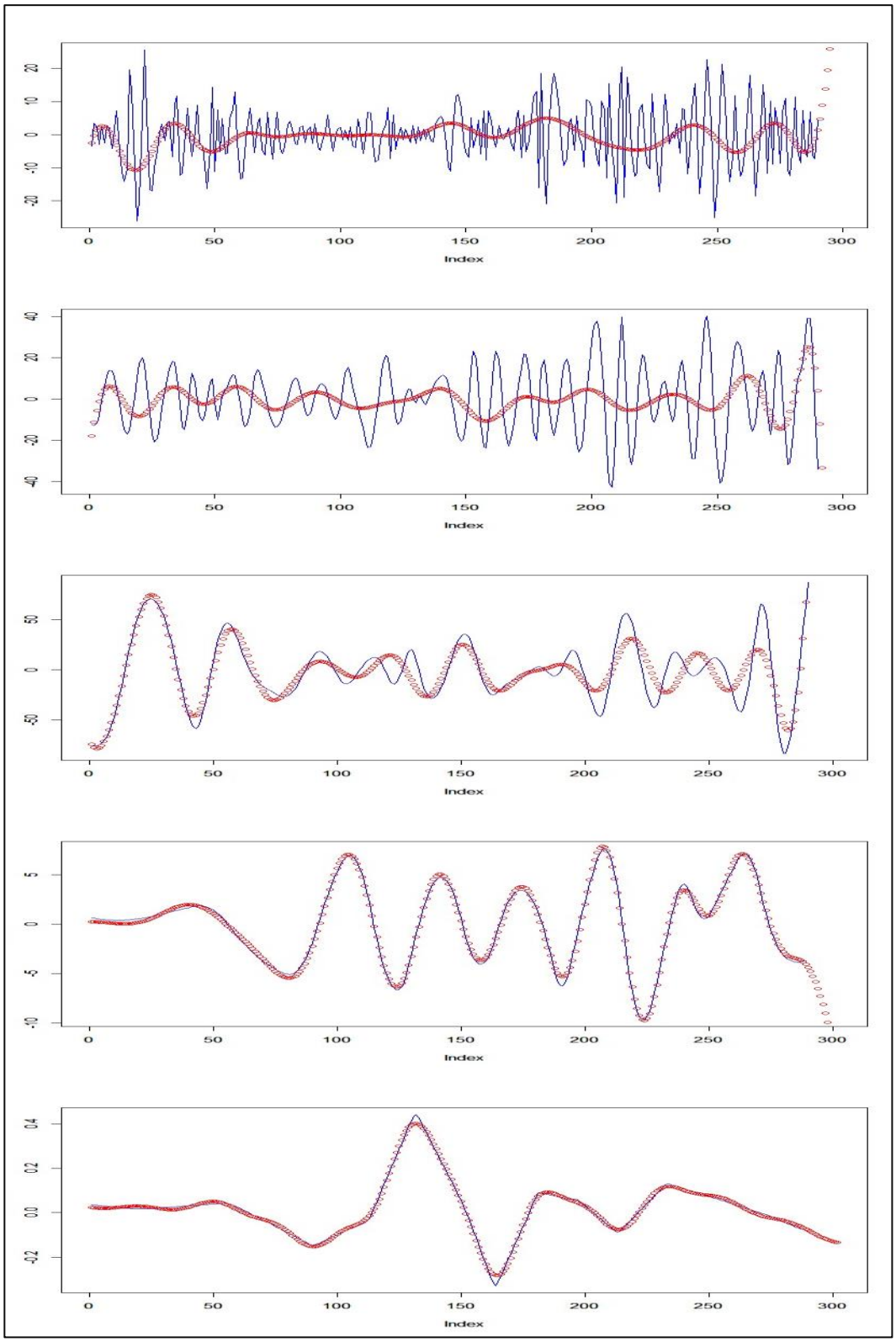


Figure 4.2: Forecast of each decomposed component using SVR

prediction of SVR and the blue line denotes the general series.

Table 4.4: Forecasting performance of EMD-SVR and SVR model

Method	Forecast period	RMSE	MAPE	MAD	ME
EMD-SVR	F_1-F_6	27.228	0.021	19.627	58.519
	F_1-F_8	37.483	0.033	31.420	68.734
SVR	F_1-F_6	73.373	0.078	72.580	69.984
	F_1-F_8	78.918	0.050	46.041	116.450

The forecasting accuracy of the EMD-SVR model was superior compared to the standard SVR model (Table 4.4) for both the forecasting horizons. This result strongly recommended that one can improve the accuracy of SVR by incorporating EMD. The accuracy of 8-step ahead EMD-SVR forecasting was slightly lower than 6-step EMD-SVR forecasting which is obvious because the performance deteriorates as forecasting horizon increases.

4.1.3 TDNN training and forecasting

TDNN being a machine learning technique and capable of handling nonlinear and complex data, we found this model apt for our comparison purpose with LS-SVR model. TDNN model was implemented to different IMFs and final residue using the methodology as described in section 3.2.7. For this purpose, “nnfor” R-package was customized and implemented to our data. Backpropagation training algorithm was used for TDNN fitting. In practice, TDNN with a small number of parameters namely input lags and hidden nodes often performs better in out of sample forecasting. This may be because over-fitting is a common problem in case of neural network modelling with a large number of parameters. In this case, we varied input lags and hidden nodes from one to five. TDNN model with three input lags and four hidden nodes was found the most suitable model for the given dataset in terms of accuracy criterion. The other parameters like maximum number of iterations for neural network was fixed at 200. We averaged the results of 26 neural networks for getting the final output. The number of neural networks to be averaged was selected on the basis of minimum error criterion. We tried averaging 10 to 50 neural networks and obtained the best result on 26 number of neural networks.

4.1.4 Performance comparison of fitted models

The performance (in-sample and out-sample) of EMD-TDNN and EMD-SVR model was compared with individual TDNN and SVR model (Table 4.5 and 4.6). Both the in-sample and out-sample performance of EMD-SVR was relatively superior as compared to other competing models. EMD based TDNN and SVR model outperformed the individual models like TDNN and SVR. The reason behind the poor performance can be attributed mainly to the fact that the singular ANN and SVR model could not handle nonstationary behaviour of the given dataset. On the other hand, the hybrid models EMD-TDNN and EMD-SVR performed better due to the ability to capture both nonlinearity and nonstationarity pattern of the dataset.

Table 4.5: In-sample performance of fitted models

F_1-F_6	**TDNN**	**SVR**	**EMD-TDNN**	**EMD-SVR**
RMSE	82.68	73.37	54.91	27.23
MAPE	0.07	0.08	0.05	0.02
MAD	72.54	72.58	49.95	19.63
ME	71.61	69.98	83.75	58.52

Table 4.6: Out-sample performance of fitted models

	TDNN	**SVR**	**EMD-TDNN**	**EMD-SVR**
RMSE	89.91	78.92	48.69	37.48
MAPE	0.09	0.05	0.05	0.03
MAD	90.40	46.04	42.34	31.42
ME	121.61	116.45	93.75	68.74

Further Diebold-Mariano (Diebold and Mariano, 1995) test was employed to judge the accuracy of EMD-SVR model compared to EMD-TDNN model. The null hypothesis of the DM test was that both models have the same accuracy. Table 4.7 showed the result of the DM test. Results clearly indicated that the proposed EMD-SVR model was superior to existing EMD-TDNN model in terms of all the criteria. The novelty of the proposed ensemble approach is that it can handle nonlinear and nonstationary data which is difficult for the traditional time series methods. Our empirical findings suggest that the proposed EMD-SVR model can be considered as an alternative tool for agricultural price forecasting.

Table 4.7: Results of DM test

DM value	p value	Remarks
5.48	<0.01	The accuracy of EMD-SVR is superior to EMD-ANN.

4.2 Results of Co-integration based TDNN model

To evaluate the performance of the proposed TDNN model, two data series *i.e.* monthly price index of crude oil and fruits starting from April 1994 to March 2014 were taken. The price index of crude oil was obtained from International Monetary Fund website and the monthly price index of fruits was collected from Office of the Economic Advisor, Ministry of Commerce, Government of India. The summary statistics of the datasets are given in the Table 4.8. It was observed that both the price series were positively skewed. Further Jarque-Bera test included to check the normality assumption of the data series, showed the variables (Table 4.9) follow non-normal distribution and were leptokurtic in nature.

Table 4.8: Descriptive statistics of the monthly data series

	Crude oil	Fruits
Mean	94.89	106.80
Median	63.09	99.25
Maximum	249.66	232.70
Minimum	19.54	41.58
Std. Dev.	64.18	46.85
CV (%)	67.64	43.88
Skewness	0.63	0.72
Kurtosis	-1.03	-0.42

Table 4.9: Normality test of data series

Jarque-Bera Test			
	JB value	**p-value**	**Remarks**
Crude Oil	68.01	<0.01	Variables follow non-normal distribution and are leptokurtic in nature
Fruits	242.02	<0.01	Variables follow non-normal distribution and are leptokurtic in nature

4.2.1 Co-integration test of the data

In case of nonstationarity of the time series, co-integration provides an appropriate statistical technique to investigate if there is a statistically significant relationship between the time series. Accordingly, first step is determination of nonstationarity nature of the price series. Nonstationarity behaviour of the crude oil and fruit price series were checked using the ADF and PP test (Table 4.10 (a)). The results highlighted that both the monthly price data series were nonstationary at level and stationary after first differencing. This results indicated that both the data series were integrated of order one *I(1)* and suitable for co-integration analysis.

Table 4.10 (a): Stationarity test of data series

Series		**Augmented Dickey-Fuller**		**Phillip-Perron**	
		t-statistic	**Prob.**	**t-statistic**	**Prob.**
Crude oil	Level	-1.167	0.689	-1.156	0.693
	1st difference	-14.086	<0.001	-9.952	<0.001
Fruits	Level	-0.926	0.779	0.928	0.996
	1st difference	-13.166	<0.001	-22.229	<0.001

Table 4.10 (b): Brock- Dechert-Scheinkman (BDS) test for data series

Series	**Embedding dimension**				**Conclusion**
	2		**3**		
	Statistics	**Probability**	**Statistics**	**Probability**	Nonlinear
Fruits	112.794	< 0.001	199.717	< 0.001	
	51.822	< 0.001	63.787	< 0.001	
	40.849	< 0.001	43.339	< 0.001	
	37.268	< 0.001	36.669	< 0.001	
Crude oil	63.750	< 0.001	102.304	< 0.001	Nonlinear
	83.517	< 0.001	106.389	< 0.001	
	53.964	< 0.001	58.675	< 0.001	
	40.097	< 0.001	40.129	< 0.001	

The nonlinearity of data series was checked using BDS test (Table 4.10 (b)). The results indicated both the fruits and crude oil data series were nonlinear in nature. The co-integration of the price series was checked using the Johansen's co-integration test. For the optimal lag length for the Johansen's co-integration test, the vector autoregression model was applied at one to ten lags. For selecting optimal length, four criteria were used *i.e.* AIC, Hannan-Quinn Information Criterion (HQIC), SIC and Final Prediction Error (FPE) (Table 4.11).

Table 4.11: Lag selection criteria for Johansen test

	Fruits vs Crude oil
AIC	3
HQ	3
SC	3
FPE	3

Based on the information criteria, the lag length of 3 was fixed for the model with fruit as the dependent variable and crude oil as independent variable. All the computations were carried out in R Studio and the required packages were "vars" and "urca". Both Johansen trace-based as well as maximum eigen value based tests were used to find co-integration. The tests indicated that there was co-integration among crude oil price and fruit price (Table 4.12).

Table 4.12: Johansen co-integration rank test

Fruits vs Crude oil	**Test statistic**	**Prob.**	**Remarks**
λ_{trace}			
$H_0: r = 0 \; vs \; H_1: r \geq 1$	12.31	0.14	
$H_0: r \leq 1 \; vs \; H_1: r \geq 2$	0.62	0.42	*r=1* not rejected.
λ_{max}			Co-integration occurs.
$H_0: r = 0 \; vs \; H_1: r \geq 1$	11.68	0.12	
$H_0: r \leq 1 \; vs \; H_1: r \geq 2$	0.62	0.42	

Table 4.13: Estimated parameter value of fitted VAR models

Model	**Regressors**	**Parameter estimates**	**t-test**	**p-value**
Fruits vs Crude Oil	β	-0.282	-5.66	<0.01
	ECT_{t-1}	-0.13	-4.40	<0.01

The detailed estimated parameters of VAR models are given in Table 4.13 with speed adjustment factor and the cointegrating vector *β.* The estimated value of β was -0.282 for

fruits vs crude oil model. This provided strong evidence of long-run relationship among the data series.

4.2.2 Proposed TDNN model

The estimated value of the error correction term was used for building the proposed time-delay neural network model. In the previous chapter section 3.2.8, the detailed of the proposed TDNN model was discussed. The main concept of the proposed TDNN model was the development of a model that uses co-integration behaviour among the data series when the data is nonlinear and nonstationary. The estimated ECT of the fitted VAR models was incorporated in TDNN model as auxiliary information (figure 4.3). The fitting of the TDNN model was done in R-Studio with the help of package "nnet".

4.2.3 Performance of the proposed TDNN model

For training of the proposed model, first 228 observations were used and remaining last 12 observations were kept to check the generalization power of the model. As mentioned earlier, in practice, a simple neural network structure with small number of parameters are preferred due to better generalization ability for out of the sample data. Accordingly, we varied input lags from one to five and number of hidden nodes from one to ten. TDNN model with three input lags and five hidden nodes was found as the best model in terms of overall accuracy criteria such as the root mean squared error, mean absolute error etc. Repeats were tried from 10 to 30 for obtaining the best forecast from TDNN model. Repeat means the number of networks that were averaged for getting the output. In this study, repeats = 26 *i.e*. 26 neural networks were averaged to get the desired forecast. Further, the estimated value of ECT (Table 4.13) was taken as xreg *i.e*. exogenous regressor variable. Box-Cox transformation was also used for the model. Maximum numbers of iterations "maxit" for neural network fitting was checked from 100 to 250. The best result was obtained at 200 iterations. The parameter values used for fitting are reported in Table 4.14. The required R code was generated by modifying the existing R packages for neural networks.

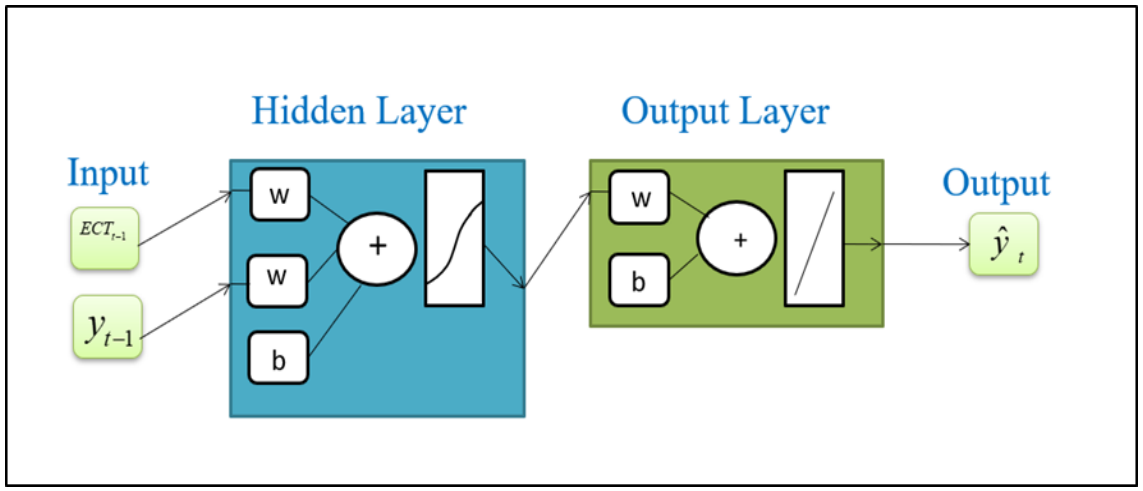


Figure 4.3: Proposed TDNN model framework

Table 4.14: Parameter values used for TDNN model fitting

	Lag	Hidden Node	Repeats	xreg	Maxit	Box-Cox Parameter value
Fruits	3	5	26	-0.13	200	0.5

4.2.3.1 Fruits price index forecasting using proposed TDNN model

The proposed TDNN model was fitted to fruit price index data set. Table 4.15 describes the in-sample performance of the ECT based TDNN model and usual TDNN model. It has been noticed that ECT based TDNN model perform better as compared to the standard TDNN model. The percentage of improvement concerning MAPE for the TDNN model with ECT was 0.3727.

Table 4.15: In-sample performance of proposed model for fruit data

Training set (228 obs)	ME	RMSE	MAE	MPE	MAPE	% of improvement
TDNN	0.0034	5.5894	3.6673	-0.3630	3.9050	0.3727
TDNN with ECT	0.0035	5.6163	3.6658	-0.3640	3.8905	

Further, the generalization power of this model was checked using a hold-out sample of last 12 observations. The results of the TDNN and TDNN model with ECT are given in

Table 4.16. The performance measures showed that the proposed TDNN model with ECT performed better than the standard TDNN model. The percentage of improvement for ECT based TDNN has also improved for out-sample.

Table 4.16: Out-sample performance of proposed model for fruit data

Testing set (12 obs)	ME	RMSE	MAE	MPE	MAPE	% of improvement
TDNN	0.0210	6.1190	4.0091	-0.3370	3.9870	3.27
TDNN with ECT	0.0301	6.0040	3.9880	-0.3220	3.9543	

The out-sample performance of the fitted models is also shown in Figure 4.4. The black dotted line described the original time series values. The square lines (red coloured) and circular dotted (green coloured) denote the forecasted value of TDNN with ECT model and standard TDNN model respectively. Figure 4.4 indicated that the forecasted values of the proposed TDNN model with ECT were closer to the original data point, while predicted values of standard TDNN model deviated from the original data points. Hence, the prediction of the proposed model is more accurate than the standard TDNN model.

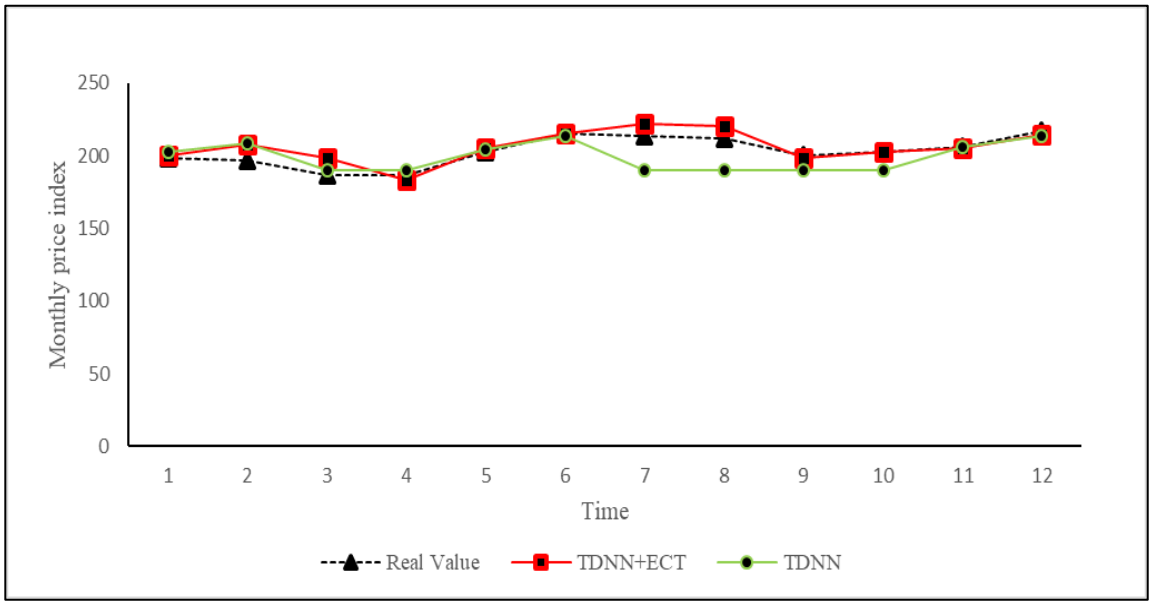


Figure 4.4: Original values with forecasted values of proposed TDNN model for fruit price

Table 4.17: Forecasted value and prediction interval of fruits data through the proposed model

Point	Forecasted value	Lo 95	Hi 95
April,2014	222.13	222.06	222.18
May, 2014	217.21	217.12	217.29
June, 2014	198.12	198.08	198.17
July, 2014	202.45	202.41	202.50
Aug., 2014	206.06	205.99	206.13
Sept., 2014	215.25	215.19	215.32
Oct., 2014	221.81	221.75	221.88
Nov., 2014	217.99	217.84	218.15
Dec.,2014	202.38	202.33	202.43
Jan.,2015	205.64	205.56	205.72
Feb., 2015	215.13	215.06	215.20
April, 2015	221.91	221.85	221.97

Generally, for a nonstationary time series, differencing of the data series lead to loss of some information. The ECT from co-integration model helps to improve the performance of the TDNN model by incorporating this lost information. As a result, the proposed TDNN model with ECT generated superior result than the single TDNN model. So, the TDNN model with ECT was used for further forecasting. The 12-step forecasted value of fruits price index using the proposed TDNN was reported in Table 4.17. The predictions intervals at 95% are also mentioned in Table 4.17. Pictorial representation of the forecasted value along with the original fruits price series are given in figure 4.5. The blue line denotes 12 step ahead forecast value of the series.

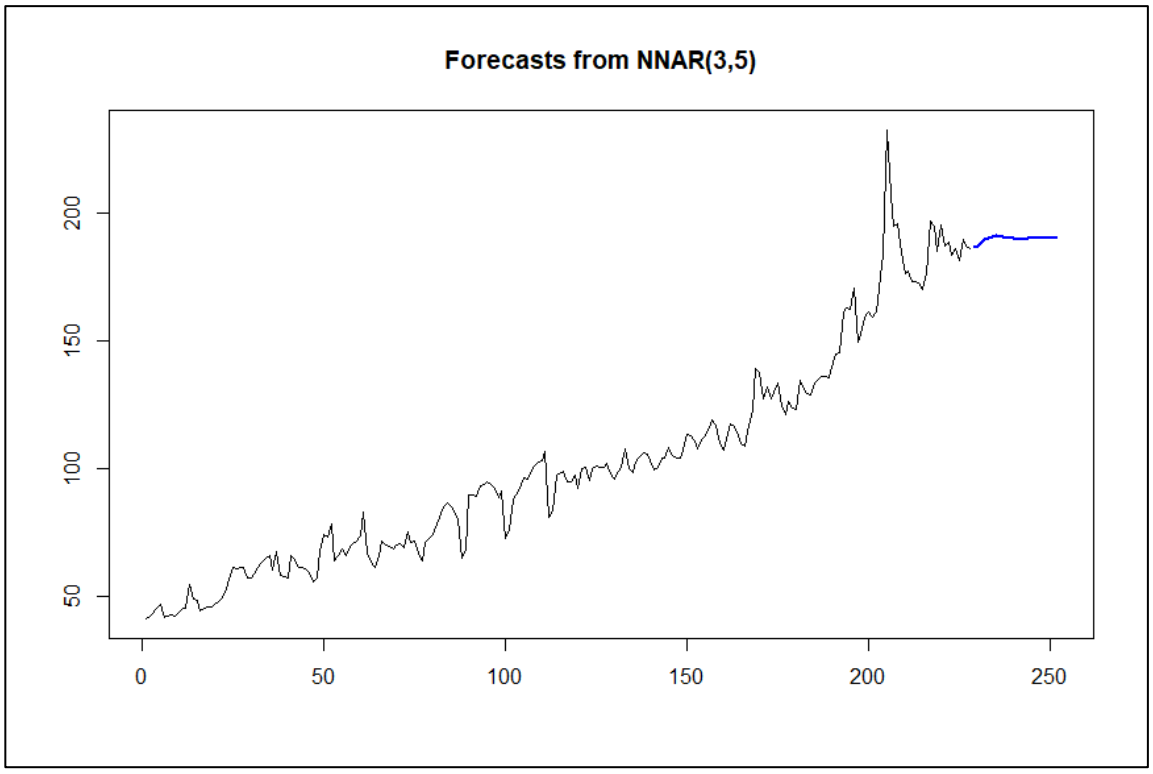


Figure 4.5: 12 step ahead forecast value of fruits using proposed model

4.3 Results of MARS based hybrid model

In this study, ANN, MARS and SVR model were fitted with the help of R. Fraction of cases selected at random for testing was 0.20 *i.e.* 80% of the data was randomly used for the training and rest 20% for the testing.

4.3.1 Hybrid modelling Approach

Under third objective, an effort was made to formulate two new hybrid models, combining MARS with ANN as well as MARS with SVR. The crop yield is multifactorial in nature. So, selection of important variables from the set of effecting variables is very crucial for precise yield prediction. Mostly, literature in crop yield prediction selects important input variables on the basis of theoretical knowledge. In this study, we have used machine learning technique for variable selection based on statistical loss function from a set of relevant variables in our dataset instead of hand picking of the input variables. The choice of MARS for variable selection is based on empirical evidence of the ability of the method in the existing literature. In the hybrid model approach, firstly MARS model was

employed on the data to find out the important independent variables. Then prediction of the dependent variable (yield) based on important variables using either ANN or SVR model was done. Figure 4.6 gives a pictorial representation of the proposed MARS based ANN framework. $x_1, x_2, \ldots, x_n$ were the independent variables in which MARS model of order 3 is fitted to extract the important variables. $i_1, i_2, \ldots i_n$ were the selected variables by the MARS model which are reported in Table 4.24. These selected variables were then used to predict the dependent variables using ANN with 4 hidden nodes. On similar lines we also attempted to build a MARS based SVR framework in this study.

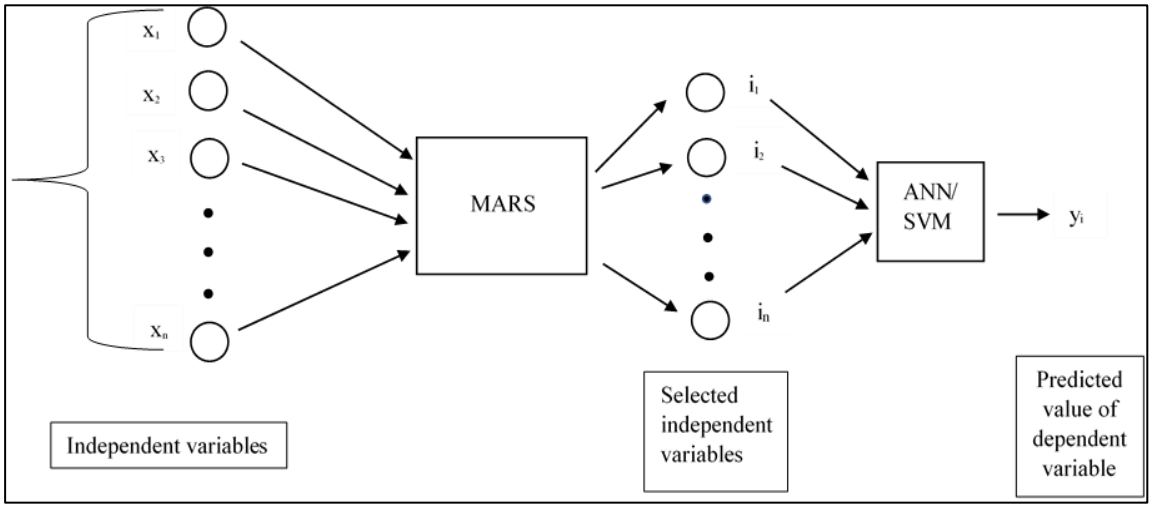


Figure 4.6: MARS based ANN/SVR hybrid framework

4.3.2 Model identification

4.3.2.1 MARS model identification

The basic aim of model fitting is to find a model that describes the exact relationship between the target variable and the explanatory variables. In the study, yield per plant (YPP) was taken as a target variable. Based on cross-validation performance, we had tried different methods for fixing the number of model terms upper bound (N*max*) and the penalty coefficient (*b*). It was observed that N*max* =20 and b =3 provided the best fit for the model. In the MARS model fitting, MARS degree largely influences the performance of model fitting and forecasting. Hence, we critically examined the performance of the model on different degrees (df 1, 2 and 3). Table 4.18-4.20 described the basis functions for the MARS model of degree 1, 2 and 3 respectively. It was found that the model

complexity increased as the MARS degree increased. To compare the model accuracy of the three MARS model, RMSE, MAD, MAPE and ME were computed.

Table 4.18: List of basis function for MARS model of degree 1

BASIS FUNCTION	DEFINITION	COEFFICIENT
	(Intercept)	-1.2315
BF1	h(Ph-20.6)	1.5540
BF2	h(23-Ph)	1.3914
BF3	h(Ph-23)	-1.5418
BF4	h(118-DM)	-0.1284
BF5	h(DM-118)	-0.0304
BF6	h(2.9-SW)	-0.9129
BF7	h(6.7-SB)	-0.5892
BF8	h(PPP-52.7)	0.0283
BF9	h(PPP-162)	-0.0172
BF10	h(7.3-PHLP)	0.1993
BF11	h(BYP-15.9)	0.2371

Table 4.19: List of basis function for MARS model of degree 2

BASIS FUNCTION	DEFINITION	COEFFICIENT
	(Intercept)	1.841
BF1	h(20.6-Ph)	-8.362
BF2	h(52.7-PPP)	-0.150
BF3	h(PPP-52.7)	0.027
BF4	h(BYP-13.2)	0.273
BF5	h(20.6-Ph) * SW	4.022
BF6	PB * h(52.7-PPP)	0.043
BF7	h(2.9-SW) * h(PPP-52.7)	-0.012
BF8	h(PPP-52.7) * h(8-PHLP)	0.011
BF9	h(PPP-52.7) * h(BYP-16.2)	-0.001
BF10	h(PPP-52.7) * h(16.2-BYP)	0.001
BF11	h(8.3-PHLP) * h(BYP-13.2)	-0.203

Table 4.20: List of basis function for MARS model of degree 3

BASIS FUNCTION	DEFINITION	COEFFICIENT
	(Intercept)	1.846
BF1	h(20.6-Ph)	-8.431
BF2	h(52.7-PPP)	-0.149
BF3	h(PPP-52.7)	0.027
BF4	h(BYP-13.2)	0.291
BF5	h(20.6-Ph) * SW	4.057
BF6	PB * h(52.7-PPP)	0.043
BF7	h(2.9-SW) * h(PPP-52.7)	-0.012
BF8	h(PPP-52.7) * h(8-PHLP)	0.011
BF9	h(PPP-52.7) * h(BYP-16.2)	-0.001
BF10	h(PPP-52.7) * h(16.2-BYP)	0.002
BF11	h(8.3-PHLP) * h(BYP-13.2)	-0.208
BF12	h(72-DF) * h(PPP-52.7) * h(PHLP-8)	-0.001

Table 4.21 presents the RMSE, MAD, MAPE and ME of the three MARS models. It has been found that the MARS model with interaction terms tends to perform better than the model without interaction (MARS model with degree 1). MARS model with degree of 3 had smaller RMSE, MAD and ME value compared to the MARS model with degree of 1 and 2. In Figure 4.7, a detail of the MARS model with degree of 3 was given. Therefore, the MARS model with degree of 3 was selected for the model fitting.

Table 4.21: Performance measures for different MARS models

Degree	RMSE	MAD	MAPE	ME
1	0.597	0.513	0.179	6.149
2	0.449	0.487	0.157	4.239
3	0.436	0.484	0.157	4.216

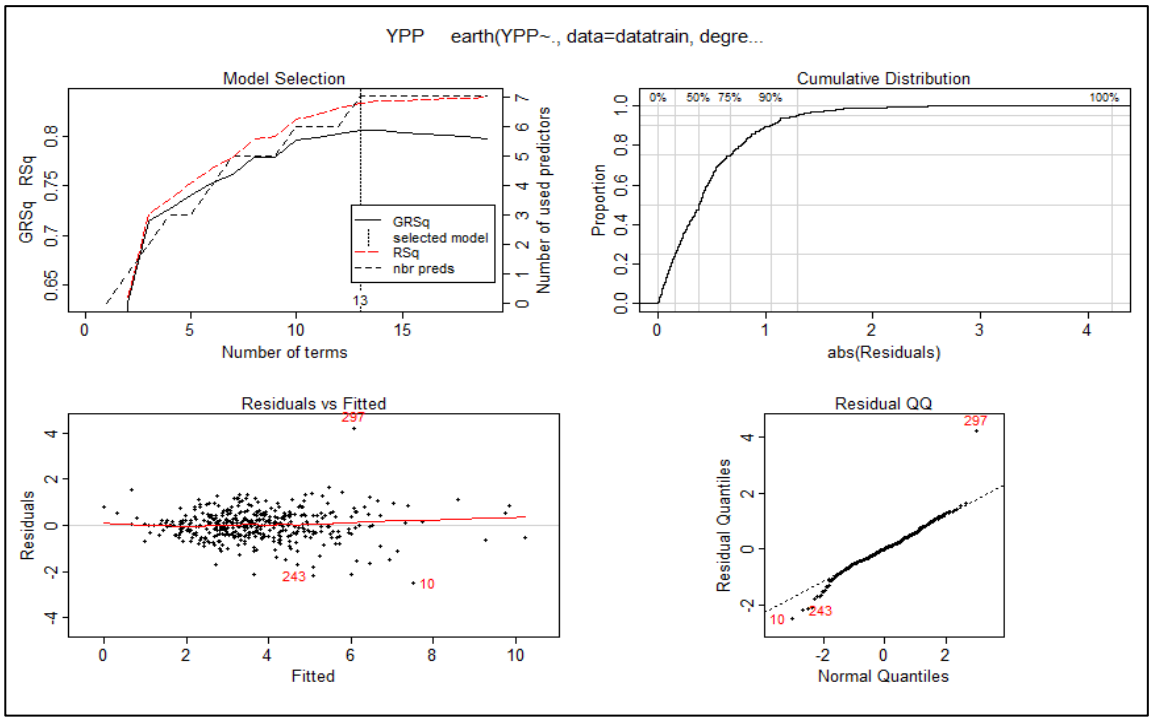


Figure 4.7: Details of MARS model of degree 3

The normality assumption of residuals of selected MARS model was also tested using the Shapiro-Wilk test and Anderson-Darling test. The *p*-value of the Shapiro-Wilk test and the Anderson-Darling test was <0.001 for both test statistic. Hence the selected model was adequate for the given dataset.

4.3.2.2 ANN model identification

For ANN model fitting, the same lentil dataset was used. 80% of the dataset was used for training and the remaining 20% for testing. For model building purpose, we have varied our model architecture with 1 to 5 hidden nodes with single hidden layer. Neuralnet package of R was used for ANN fitting. Two types of resilient backpropagation were used viz "rprop+" and "rprop-". rprop+ is defined as a direct adaptive method for faster backpropagation learning. rprop− is defined as advanced supervised learning in multi-layer perceptrons from backpropagation to adaptive learning algorithms. Stepmax is the maximum steps for the training of the neural network. Reaching this maximum leads to halting of the neural network's training process. Rep is the number of repetitions for the neural network's training and the threshold is a numeric value specifying the threshold for the partial derivatives of the error function as stopping criterion. The schematic

representation of fitted ANN model with weights is shown in Figure 4.8. In the present +study, neural network models are fitted with rep=1 to 3, stepmax = 1e+05 to 1e+08 and threshold =0.01. It was observed that the error rate of the fitted ANN model was minimax at stepmax=1e+07 with threshold=0.01 and rep=3 while most of the other models with different values of stepmax, rep and threshold did not meet the convergence criteria. So,

Table 4.22: Performance measures for different number of nodes in ANN models

No. of nodes in Hidden Layer	RMSE	MAD	MAPE	Error rate (best replication)
1	1.613	1.217	0.461	538.833
2	1.613	1.217	0.461	538.833
3	1.613	1.217	0.461	538.833
4	0.963	0.629	0.183	143.767
5	1.151	1.019	0.352	238.833

the entire analysis was conducted at stepmax=1e+07, rep=3 and threshold=0.01. Table 4.22 summarised the error rate and performance measures of fitted ANN with a different number of nodes in the hidden layer. ANN model with 1, 2, 3 and 5 hidden nodes had the same performance. But ANN model with 4 hidden nodes gave the best result. So, the best-fitted replication in the ANN model with 4 hidden nodes was taken for yield forecasting.

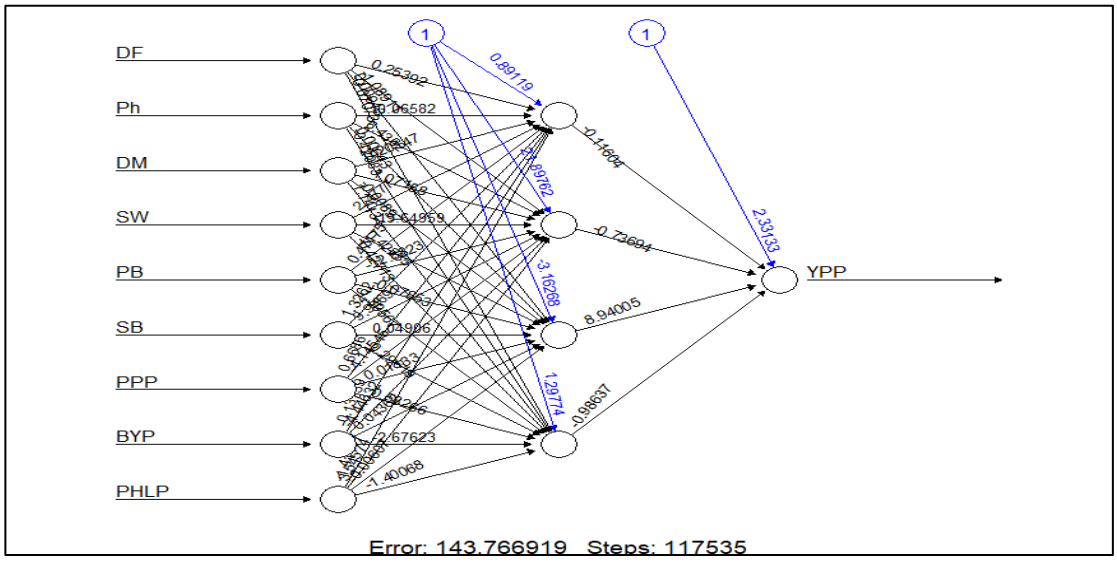


Figure 4.7: Schematic diagram of fitted neural network structure

4.3.2.3 SVR model identification

In this study R package, "e1071" was customized and implemented. This package provides several options for SVR model fitting using different types of kernel functions like linear, radial basis, sigmoid and polynomial. Although the most often used and recommended function is radial basis. But, it is recommended to select the appropriate kernel function for the given dataset. So, SVR has been fitted using the four different kernel basis functions and the best model was selected on the basis of performance measures. In SVR model, besides kernel function other important parameters like epsilon factor, cross-validation and type of regression largely influence the model performance. Thus, we tried a range of values of these parameters for selecting the most efficient model for our dataset. The epsilon factor also known as tolerance factor gives more accurate model at lower values. For our dataset, we found the value of epsilon factor at 0.1 and three degree of parameters provide the best result. The resultant model was validated using 10-fold cross-validation method. Further, we focussed our attention on selection of the appropriate kernel function. The support vectors produced using RBF, linear and polynomial were 323, 321 and 329 respectively. In the training phase, performance measures like RMSE, MAD, MAPE and ME were computed for choosing the best forecasting model. From Table 4.23, it has been observed that the SVR model with radial basis kernel function provided the best result on the basis of parsimonious representation. Hence, the SVR model with radial basis kernel function was employed for yield forecasting.

Table 4.23: Performance measures for different kernel function in SVR

Kernel function	RMSE	MAD	MAPE	ME	No. of Support vectors
Radial basis	0.6474	0.3602	0.1089	7.1265	323
Linear	0.8599	0.5231	0.1586	8.8763	321
Polynomial	0.827	0.5253	0.1537	6.8826	329
Sigmoid	0.8269	0.5253	0.1537	6.8825	403

MARS model of degree 3 selects 7 variables among 9 predictors that affects yield of lentil data (Table 4.24). These 7 predictors are further used as input variable in ANN and SVR for predicting yield of lentil.

Table 4.24: Important predictor identified by MARS model of degree 3

Variable	GCV	RSS
PPP	100	100
SW	46.5	48.6
Ph	31.2	33.9
BYP	31.1	33.9
PHLP	25.5	28
PB	18.4	20.1
DF	6.6	7.9

4.3.2 Performance comparison of fitted models

Performance of fitted models was compared on fit statistics values for both in-sample and out-sample forecasting. The details of residuals obtained were also analysed to check the model adequacy. Table 4.25 and Table 4.26 depicted the value of the statistic of fitted models. It was observed that among the individual models MARS performed the best followed by SVR and ANN. MARS being a nonparametric form of piece-wise nonlinear regression, a model is fitted locally to each sub-region of the available range of dataset. The low performance of ANN may be due to over-fitting problem. However, both proposed hybrid models performed better than individual models. As indicated earlier, in case of hybrid models, nonlinear forecasting techniques like ANN and SVR were used with relevant explanatory variables extracted with the help of MARS model. Parsimony and two stage building of the model may be the reason of better performance of the hybrid models.

Table 4.25: In-sample performance measures of the fitted models

Method	RMSE	MAD	MAPE	ME
ANN	0.9827	0.6288	0.1828	8.1055
SVR	0.6474	0.3602	0.1089	7.1265
MARS	0.4356	0.4842	0.1565	4.2157
MARS-ANN	0.0802	0.0607	0.2478	0.3918
MARS-SVR	0.0826	0.0579	0.1834	0.8498

In out-sample forecasting, SVR performed the best followed by ANN and MARS model. However, both the hybrid models performed better than the individual models even in out of sample forecasting. The reason behind the superior performance of the MARS based hybrid models as compared to the individual models such as MARS, SVR and ANN can be largely due to the enhanced feature extraction capability of MARS model coupled with nonlinear adaptive learning feature of ANN and SVR. Although, the individual models like ANN and SVR have the capability of modelling nonlinear complex data structure but for efficiently forecasting such data structure, selection of important input variables hold pivotal position. This important aspect was contributed by the MARS model in the hybrid setup hence increasing its efficiency in modelling and forecasting.

Table 4.26: Out-sample performance measures of the models

Method	RMSE	MAD	MAPE	ME
ANN	0.8142	0.6435	0.2308	2.4871
MARS	0.9415	0.6147	0.2769	5.3540
SVR	0.6853	0.4902	0.2707	2.6435
MARS-ANN	0.0802	0.0579	0.2214	0.7085
MARS-SVR	0.0658	0.0579	0.1626	0.2206

Besides, these hybrid models have good generalization ability as evident from the out-of-sample performance. Further, the DM test was done to identify the best model between MARS-ANN and MARS-SVR. The null hypothesis of the test was MARS-ANN and MARS-SVR had the same accuracy and alternative MARS-ANN model had better accuracy than MARS-SVR model. Table 4.27 showed the significance of the DM test. The MARS-ANN model was superior to the MARS-SVR.

Table 4.27: Results of DM test

DM value	p value	Remarks
4.185	<0.01	The accuracy of MARS-ANN is better than MARS-SVR.

4.4 Conclusion

In view of the complexity of agricultural price movement and the uncertainty of agricultural price forecasting, this study proposed a hybrid combination of the EMD which decomposed the price series into nonlinear and time varying components that are then used to train an SVR model to produce out of sample ensemble price forecasts. The results clearly demonstrated the superior performance of the proposed EMD-SVR model in comparison to individual models such as TDNN and SVR for agricultural price forecasting. The accuracy of EMD-SVR is even higher as compared to the EMD-TDNN model. Under the second objective, an effort was made to improve the prediction ability of the standard TDNN model by incorporating the error correction terms obtained from the co-integrated series. Finally, MARS based hybrid models were proposed for accurate crop yield prediction. In the first stage, important input variables were selected using MARS model instead of hand picking of variables on theoretical knowledge. In the second stage, nonlinear prediction techniques like ANN and SVR were used for yield prediction using the selected variables. The proposed MARS based hybrid models performed better as compared to the individual models such as MARS, SVR and ANN, may be largely due to the enhanced feature extraction capability of MARS model coupled with nonlinear adaptive learning feature of ANN and SVR.

CHAPTER V

SUMMARY

The scientific and effective forecasting method is helpful to correctly guide producers and policy makers to match the supply and demand of the agricultural production and facilitate decision-making process of the government. Agricultural price forecasting is not an easy task due to its dependency on many extraneous factors. Nonlinearity and nonstationary behaviour of data series are crucial problems in the agricultural price forecasting. Agricultural commodity prices are volatile in nature due to seasonality, inelastic demand, production uncertainty *etc.* Traditionally, time series forecasting has been dominated by linear methods like ARIMA (Box and Jenkins, 1970) and nonlinear models such as SETAR, STAR, *etc.* because they are well understood and effective on many situations. These traditional methods suffer from some limitations, such as linear models focus on linear relationships, fixed temporal dependence, *etc.* and nonlinear models require the specific nonlinear relation of data generating process to be known *a priori.* On the other hand, Artificial Intelligence (AI) models with its flexible functional designs and powerful self-learning capabilities have recently become a great alternative for time series data forecasting. But, it has been observed that a single model is not sufficient to deal with complex real world systems such as agricultural data which contains unknown mixed patterns. Taking into account of these factors, there is an ever increasing need of using hybrid models to improve the accuracy of predictions. In order to improve forecast accuracy, hybridization is a good idea because they can capture various patterns in the data concurrently. Hybrid models are more robust as they often compliment the advantages of the individual techniques involved and improve the forecasting accuracy.

In the context of above advantages of hybrid models, we proposed three different hybrid models under three objectives for the present investigation. In the first objective, empirical mode decomposition based support vector regression model has been developed. The performance of the proposed hybrid model was evaluated on chili wholesale price index in terms of forecasting accuracy. Time delay neural network based on ECT considering co-integration (TDNN-ECT) has been developed in the second objective. In the third

objective, the performance of proposed MARS based ANN model has been evaluated for yield prediction.

The entire thesis is divided into five chapters. First chapter starts with a brief description of agricultural price forecasting and different statistical as well as machine learning models. This chapter also highlights the drawbacks of traditional forecasting models and problem in agricultural forecasting. Supervised, unsupervised and reinforcement learning algorithms of machine learning techniques and brief about hybrid models have been discussed in this chapter. The chapter ends with the motivation of the study and orientation of the thesis.

Chapter II provides review of literature related to the present study. A broad review regarding the three objectives are mentioned in this chapter.

Chapter III deals with the methodologies and description of the data used in this study. The basic common tests like stationarity test, nonlinearity tests and others have been described. This chapter also provide the details of the proposed EMD-SVR, TDNN with ECT and MARS-ANN hybrid models. Forecasting evaluation criteria like RMSE, MAD, MAPE and ME have been described.

In chapter IV, all the results of the study are given with tables and figures. Empirical results indicated that the proposed EMD-SVR model perform better compare to the existing TDNN, SVR and EMD-TDNN models. Time delay neural network model based on ECT considering co-integration (TDNN-ECT) has been demonstrated on the co-integrated series of crude oil and fruit WPI pair. Investigation revealed that MARS based ANN model outperform than the existing ANN, MARS and SVR models for yield prediction. The performance of models is compared in respect of RMSE, MAPE, MAD and ME. DM test has been conducted to check the accuracy of all the developed hybrid models. The results indicate superiority of the developed hybrid models in their respective situations. They are capable to handle nonstationary and nonlinearity problems simultaneously. This study can be extended by employing advanced EMD techniques like Complementary Ensemble Empirical Mode Decomposition (CEEMD) algorithm which can further improve the forecasting accuracy of the hybrid model.

Study on machine learning techniques based hybrid model for forecasting in agriculture

Abstract

Agricultural datasets are mostly nonlinear, nonstationary and leptokurtic in nature. These properties of dataset pose a variety of problems in forecasting. Precise forecasting helps both farming community and policy makers to undertake informed decisions. Literature suggests that each of the forecasting models has their own limitations. A single forecasting model is not able to handle problems like nonstationary and nonlinearity simultaneously. Accordingly, the present study proposes three different hybrid models *i.e.* empirical mode decomposition based support vector regression (EMD-SVR), time-delay neural network with error correction term (TDNN-ECT) and multivariate adaptive regression splines based artificial neural network (MARS-ANN) models. The novelty of these models lies in the fact that they can handle both nonstationary and nonlinear features of dataset simultaneously. In EMD-SVR model, the nonstationary and nonlinear dataset is decomposed into different intrinsic mode functions and final residue through EMD method. Then the decomposed components are forecasted using SVR model and finally, all forecasted values are summed up to produce the final forecast. In the second model, TDNN-ECT uses the error correction term from the two co-integrated series as auxiliary variable. The auxiliary information in the form of ECT improves the forecasting accuracy. Further, selection of important input variables is a crucial step in determining the accuracy of any forecasting model. Hence, MARS-ANN hybrid model was developed in which the MARS algorithms was employed to extract important factors determining crop yield and the extracted factors were used for yield prediction using ANN methodology. The performance of proposed hybrid models is evaluated with individual forecasting models using three different agricultural datasets. The performance measures like RMSE, MAD, MAPE and ME are used to evaluate the model. The results indicated that the performance of the proposed hybrid models are substantially superior as compared to the individual forecasting model.

Key words: *Co-integration, Nonlinearity, Nonstationary, EMD, SVR, TDNN, and MARS.*

कृषि में पूर्वानुमान के लिए मशीन-लर्निंग तकनीकों पर आधारित हाइब्रिड मॉडल का अध्ययन

सार

ज्यादातर कृषि आकड़े नॉनलिनियर,नॉनस्टेसनरी एवं लेप्टोकर्टिक प्रकृति के होते हैं। आकड़ों की यह प्रकृति, पूर्वानुमान में बहुत सारी समस्याएँ पैदा करता है। सटीक पूर्वानुमान से कृषि समुदाय से जुड़े लोगो एवं नीति निर्माताओं, दोनों को भविष्य की योजना बनाने में मदद मिलती है। साहित्य बताता है कि प्रत्येक पूर्वानुमान मॉडल की अपनी धारणाएं हैं। एक एकल पूर्वानुमान मॉडल एक साथ नॉन-लिनियर एवं नॉन-स्टेसनरी जैसी दो समस्याओं से निपटने में सक्षम नहीं होते हैं। इन सब बिंदुओं को ध्यान में रखते हुए, वर्तमान अध्ययन में तीन अलग-अलग हाइब्रिड मॉडल यानी EMD-SVR, TDNN मॉडल के साथ ECT और MARS-ANN मॉडल प्रस्तावित किए गए हैं। इन मॉडलों की विशेषता यह है कि ये नॉन-लिनियर एवं नॉन-स्टेसनरी संबंधी समस्या दोनों को एक साथ संभाल सकते हैं। EMD-SVR मॉडल में, नॉन-लिनियर एवं नॉन-स्टेसनरी डेटासेट को EMD विधि के माध्यम से अलग-अलग आईएमएफ और अंतिम अवशेष में विघटित किया जाता है। तब विघटित घटकों को SVR मॉडल का उपयोग करके पूर्वानुमानित किया जाता है और अंत में, सभी पूर्वानुमानित मूल्यों को अंतिम पूर्वानुमान तैयार करने के लिए अभिव्यक्त किया जाता है। दूसरे मॉडल में, TDNN मॉडल ECT को दो संयोगित श्रृंखला से सहायक चर के रूप में उपयोग करता है। ईसीटी के संदर्भ में सहायक जानकारी पूर्वानुमान सटीकता में सुधार करती है। MARS-ANN हाइब्रिड मॉडल, MARS और ANN पद्धति के संयोजन से विकसित किया गया है। MARS एल्गोरिथ्म डेटासेट से महत्वपूर्ण चर निकालता है और इन निकाले गए चर का उपयोग ANN विधि का उपयोग करके भविष्यवाणी किया जाता है। प्रस्तावित हाइब्रिड मॉडलों के प्रदर्शन का मूल्यांकन तीन अलग-अलग कृषि डेटासेट का उपयोग करते हुए जेनेरिक पूर्वानुमान मॉडल के साथ किया गया है। RMSE, MAE, MAPE और ME जैसे प्रदर्शन उपायों का उपयोग मॉडल का मूल्यांकन करने के लिए किया गया है। परिणाम बताते हैं कि प्रस्तावित हाइब्रिड मॉडल का प्रदर्शन सामान्य पूर्वानुमान मॉडल की तुलना में काफी बेहतर है। इस अध्ययन से हम कृषि डेटासेट पर प्रस्तावित हाइब्रिड मॉडल के उपयोग की सुझाव करते हैं।

मुख्य शब्द: *लेप्टोकर्टिक, नॉनलाइनियरिटी, नॉनस्टेसनरी, ईएमडी, एसवीआर, टीडीएनएन, ईसीटी, मार्स, एएनएन, आईएमएफ और कॉइनटेग्रेटेड सीरीज़।*

ANNEXURE

1. R code related to EMD based support vector regression model

```
#Import of dataset

data=read.table (file = "clipboard", sep = "\t", header = TRUE)

#install of required packages

install.packages(timeseries)

install.packages(tseries)

install.packages(moments)

install.packages(Emd)

#loading of packages

Library (tseries)

Library (timeseries)

Library (forecast)

plot.default(data)

xt=as.matrix(data)

xt=as.vector(xt)

#data import

ori=read.table(file = "clipboard", sep = "\t", header = TRUE)

ori=unclass(ori)

#framing of imported dataset

DF <- data.frame(nd,ori)
```

```
colnames(DF)<-c("x","y")

ylim <- c(min(DF$y), max(DF$y))

xlim <- c(min(nd1),max(nd1))

#plotting of dataset

plot(DF$y, col="blue", ylim=ylim, xlim=xlim, type="l")

prognoza

t=read.table(file = "clipboard", sep = "\t", header = TRUE)

y=ts(t)

x=ts(ori)

plot(x)

par(new=TRUE)

plot(y)

plot(prognoza, col="red", ylim=ylim, xlim=xlim)

#summary of original data set

library(moments)

summary(xt)

sd(xt)

skewness(xt)

kurtosis(xt)

# stationary test of dataset

adf.test(xt)

pp.test(xt)

bds.test(xt)
```

```
library(EMD)
data2=emd(data)
par(mfrow=c(data2$nimf,1), mar=c(2,2,2,2))
rangeimf=range(data2$imf)
for(i in 1:try$nimf){plot(data2$imf[,i], type="l", xlab="",
                          ylab="", ylim=rangeimf, main=
                            paste(i, "-th IMF", sep="")); abline(h=0)
}
#plotting of extracted components
plot(try$residue, xlab="", ylab="",main="residue", type="l",axes=FALSE);
box(try);
#imf visualization
imf=data$imf
#saving of output
write.csv(imf,"imf function.csv")
#residue extraction
residue=data$residue
#saving of residue
write.csv(residue,"residue.csv")
#isolation of each imf
imf1=imf[,1]
imf2=imf[,2]
imf3=imf[,3]
```

```
imf4=imf[,4]
#install of required packages
install.packages(e1071)
install.packages(kernab)
#Required packages loading
library(e1071)
library(kernab)
#data import
k<-read.csv("numeric variable.csv", header=TRUE)
head(k,5)
#divide training and testing
train=k[1:415,]
test=k[416:518,]
head(test)
library(e1071)
#Fit a model. The function syntax is very similar to lm function
model_svm <-svm(model, data=train)
summary(model_svm)
# #Use the predictions on the data
#in sample
predictedY <- predict(model_svm,traindata)
# summarize accuracy
mse <- mean((traindata$yt - predictedY)^2)
```

```
rmse<- sqrt(mse)

print(paste(rmse))

# R square value

R21 <- 1 - (sum((traindata$yt-predictedY)^2)/sum((traindata$yt-mean(traindata$yt))^2))

print(paste(R21))

#mean absolute deviation (MAD)

MAD1=(sum(abs(traindata$yt-predictedY))/nrow(traindata))

print(paste(MAD1))

#Mean absolute percent error (MAPE)

d=sum((abs(traindata$yt-predictedY))/traindata$yt)

MAPE1=d/nrow(traindata)

print(paste(MAPE1))

#maximum error

ME1=max(abs(traindata$yt-predictedY))

print(paste(ME1))

#out sample

predicted_out <- predict(model_svm,testdata)

# summarize accuracy

mse_out <- mean((testdata$yt - predicted_out)^2)

rmse_out<- sqrt(mse_out)

print(paste(rmse_out))

# R square value

cor(testdata$yt,predicted_out)^2
```

```
R2_out <- 1-(sum((testdata$yt-predicted_out)^2)/sum((testdata$yt-mean(testdata$yt))^2))

print(paste(R2_out))

#mean absolute deviation (MAD)

MAD_out=(sum(abs(testdata$yt-predicted_out))/nrow(testdata))

print(paste(MAD_out))

#Mean absolute percent error (MAPE)

d_out=sum((abs(testdata$yt-predicted_out))/testdata$yt)

MAPE_out=d_out/nrow(testdata)

print(paste(MAPE_out))

#maximum error

ME_out=max(abs(testdata$yt-predicted_out))

print(paste(ME_out))

# residuals checking

res=resid(model_svm)

shapiro.test(res)

ad.test(res)

re1=predicted_out-testdata$yt

shapiro.test(re1)
```

2. R code related to TDNN model considering co-integration

```
# my data analysis

#setting of working Directory

setwd("path")

getwd()
```

```
#cointegration and vecm code

#install of required packages

install.packages(urrca)

install.packages(vars)

install.packages(moments)

#Required packages loading

library(urca)

library(vars)

library(moments)

#import of data

coin=read.table(file = "clipboard", sep = "\t", header = TRUE)

#conversion of time series data

crudeoil=ts(coin)

#Nonstationarity checking of datasets

bds.test (crudeoil)

bds.test(fruit)

# to view dataset

head(coin)

head(fruit)

#Johansen Test on Simulated Data

library("urca")

set.seed(123)

z <- rep(0, 10000)
```

```
for (i in 2:10000) z[i] <- z[i-1] + rnorm(1)

p <- q <- r <- rep(0, 10000)

p <- 0.3*z + rnorm(10000)

q <- 0.6*z + rnorm(10000)

r <- 0.2*z + rnorm(10000)

jotest=ca.jo(data.frame(p,q,r), type="trace", K=2, ecdet="none", spec="longrun")

summary(jotest)

s = 1.000*p + 1.791324*q - 1.717271*r

plot(s, type="l")

library("tseries")

adf.test(s)

library("quantmod")

#Johansen Test on Financial Data

#EWA and EWC, representing baskets of equities for the Australian and Canadian economies

#IGE, which contains a basket of natural resource stocks

#The logic is that all three should in some part be affected by stochastic trends in commodities and thus may form a cointegrating relationship.

getSymbols("EWA", from="2006-04-26", to="2012-04-09")

getSymbols("EWC", from="2006-04-26", to="2012-04-09")

getSymbols("IGE", from="2006-04-26", to="2012-04-09")

#to create new variables to hold the backward-adjusted prices

ewaAdj = unclass(EWA$EWA.Adjusted)
```

```
ewcAdj = unclass(EWC$EWC.Adjusted)

igeAdj = unclass(IGE$IGE.Adjusted)

# perform the Johansen test on the three ETF daily price series and output the summary of the test:

jotest=ca.jo(data.frame(ewaAdj,ewcAdj,igeAdj), type="trace", K=2, ecdet="none", spec="longrun")

summary(jotest)

jotest@teststat[2]

jotest@teststat[1]

jotest@teststat[3]

 jotest@cval

#vecm through tsDyn package

library(tsDyn)

vecm_t <- VECM(coin, lag=1, estim="ML")

summary(vecm_t)

#forecasting

newDat <- tail(as.data.frame(coin),2)

vecm.eg <-VECM(coin, lag=1)

predict(vecm.eg, newdata=newDat, n.ahead=12)

return

#finding  of residuals

res=resid(vecm.eg)

#plotting of residuals
```

```
plot(res)

#plotting of residuals in boxplot

boxplot(res)

#histogram of the residuals

hist(res)

#time series plot of the residuals

plot.ts(res)

# Ljung test of the residuals

Box.test(res[,2],lag=6,type="Ljung")

#TDNN model fitting

# package download for NARS

install.packages (nnet)

install.packages (nnfor)

install.packages (tseries)

install.packages (forecasting)

#loading of packages

Library(tseries)

Library(nnet)

Library(nnfor)

Library (forecasting)

#fitting of model

fit.y0=nnetar(data)

#view of fitted model
```

```
fit.y0

# checking of performance data

accuracy(fit.y0)

#forecasting of dataset

fcast0=forecast(fit.y0, h=12)

#view of forecasted values

fcast0

#saving of output results

write.table(as.data.frame(fcast0),file = "testingprediction intervalwith subset ect.csv",  sep = ",", col.names = FALSE, append = T)

#plotting of forecasted data points

plot(fcast0)

lines(y0.ts)
```

3. R code related to MARS based ANN model

```
setwd("path")

#reuired packages

install.packages (TeachingDemos)

install.packages (plotrix)

install.packages (plotmo)

install.packages (forecasting)

#loading of packages

library(TeachingDemos)

library(plotrix)
```

```
library(plotmo)

library(forecasting)

p<-read.csv(file.choose())

#Random sampling

samplesize = 0.80 * nrow(p)

set.seed(100)

index = sample( seq_len ( nrow ( p ) ), size = samplesize )

# Create training and test set

datatrain = p[index, ]

datatest = p[-index, ]

#regression model fitting

reg=lm(model.,data=datatrain)

#summary of fitted model

summary(reg)

#plotting of Mars model

O2=plotrix(model,data=datatrain,degree = “reqired degree”)

#summary of fitted model

summary(O2)

#plotting of fitted model

plot (O2)

# neural network model

setwd("path")

#data import in csv format
```

```
data=read.csv(file.choose())

#required packages/libraries

#install.packages("ggplot2")

#install.packages("neuralnet ")

#install.packages("MASS")

#install.packages("boot")

#install.packages("plyr")

#load libraries

library(MASS)

library(neuralnet)

library(boot)

library(plyr)

library(ggplot2)

#to know data description

head(data)

summary(data)

#correlation matrix of data variables

cor(data)

#grapical representation of data

plot(data)

# scatter plot (for variable)

ggplot2::ggplot(data=data) +

  geom_point(mapping = aes(x = calories, y = rating,color = "red"))
```

```
# Random sampling for data splitting

samplesize = 0.80 * nrow(data)

set.seed(100)

index = sample( seq_len ( nrow ( data ) ), size = samplesize )

# Creation of training and test set

datatrain = data[ index, ]

datatest = data[ -index, ]

#Scale data for neural network (minimax normalization)

max = apply(data , 2 , max)

min = apply(data, 2 , min)

scaled = as.data.frame(scale(data, center = min, scale = max - min))

## Fit neural network

# load library

library(neuralnet)

# creating training and test set

trainNN = scaled[index , ]

testNN = scaled[-index , ]

# fit neural network

set.seed(100)

NN3 = neuralnet(, trainNN, hidden = 3 , linear.output = T )

NN5 = neuralnet(dependent variable ~ independent variables, trainNN, hidden = 5 , linear.output = T )
```

```
NN10 = neuralnet(dependent variable ~ independent variables, trainNN, hidden = 10 , linear.output = T )

# plot neural network

plot(NN3)

plot(NN5)

plot(NN10)

## Prediction using neural network

predict_testNN3 = compute(NN3, testNN[,c(1:5)])

predict_testNN10 = compute(NN10, testNN[,c(1:5)])

predict_testNN5 = compute(NN5, testNN[,c(1:5)])

#rescaling of output

predict_testNN3 = (predict_testNN3$net.result * (max(data$ypp) - min(data$ypp))) + min(data$rating)

predict_testNN5 = (predict_testNN5$net.result * (max(data$ypp) - min(data$ypp))) + min(data$rating)

predict_testNN10 = (predict_testNN10$net.result * (max(data$ypp) - min(data$ypp))) + min(data$rating)

#PLOT OF ANN

plot(datatest$ypp, predict_testNN3, col='blue', pch=16, ylab = "predicted NN3", xlab = "real ypp")

abline(0,1)

plot(datatest$ ypp, predict_testNN5, col='blue', pch=16, ylab = "predicted NN5", xlab = "real ypp ")

abline(0,1)
```

```
plot(datatest$ypp, predict_testNN10, col='blue', pch=16, ylab = "predicted rating NN10", xlab = "real ypp ")

abline(0,1)

# Calculate Root Mean Square Error (RMSE)

RMSE.NN3 = (sum((datatest$ ypp - predict_testNN3)^2) / nrow(datatest)) ^ 0.5

RMSE.NN10 = (sum((datatest$ ypp - predict_testNN10)^2) / nrow(datatest)) ^ 0.5

RMSE.NN5 = (sum((datatest$ ypp - predict_testNN5)^2) / nrow(datatest)) ^ 0.5

print(paste(RMSE.NN3,RMSE.NN5,RMSE.NN10))

#mean absolute deviation (MAD)

MAD10=(sum(abs(datatest$rating-predict_testNN10))/nrow(datatest))

MAD3=(sum(abs(datatest$rating-predict_testNN3))/nrow(datatest))

MAD5=(sum(abs(datatest$rating-predict_testNN5))/nrow(datatest))

print(paste(MAD3,MAD5,MAD10))

#Mean absolute percent error (MAPE)

d=sum((abs(datatest$rating-predict_testNN10))/datatest$ypp)

MAPE10=d/nrow(datatest)

d3=sum((abs(datatest$ypp-predict_testNN3))/datatest$ypp)

MAPE3=d3/nrow(datatest)

d5=sum((abs(datatest$ypp-predict_testNN5))/datatest$ypp)

MAPE5=d5/nrow(datatest)

print(paste(MAPE3,MAPE5,MAPE10))

#maximum error

ME10=max(abs(datatest$rating-predict_testNN10))
```

```
ME3=max(abs(datatest$rating-predict_testNN3))

ME5=max(abs(datatest$rating-predict_testNN5))

print(paste(ME3,ME5,ME10))

## Cross validation of neural network model

# install relevant libraries

install.packages("boot")

install.packages("plyr")

# Load libraries

library(boot)

library(plyr)

# Initialize variables

set.seed(50)

k = 100

RMSE.NN = NULL

List = list( )

# Fit neural network model within nested for loop

for(j in 10:65){

  for (i in 1:k) {

    index = sample(1:nrow(data),j )

    trainNN = scaled[index,]

    testNN = scaled[-index,]

    datatest = data[-index,]

    NN3 = neuralnet(ypp ~ ,trainNN, hidden = 3, linear.output= T)
```

```
   predict_testNN3 = compute(NN3,testNN[,c(1:5)])

   predict_testNN3=(predict_testNN3$net.result*(max(data$ypp)-
min(data$ypp)))+min(data$ypp)

   RMSE.NN [i]<- (sum((datatest$ypp - predict_testNN3)^2)/nrow(datatest))^0.5

  }

  List[[j]] = RMSE.NN

}

Matrix.RMSE = do.call(cbind, List)

## Variation of mean RMSE

install.packages("matrixStats")

library(matrixStats)

mean = colMeans(Matrix.RMSE)

X = seq(10,65)

plot (mean~X, type = "l", xlab = "length of training set", ylab = "mean RMSE", main =
"Variation of RMSE with length of training set")
```

BIBLIOGRAPHY

Abdulkarim, S. A. and Engelbrecht, A. P. (2019). Time Series Forecasting Using Neural Networks: Are Recurrent Connections Necessary? *Neural Processing Letters*, **50(2)**, 1-33.

Alam, W., Ray, M., Kumar, R. R., Sinha, K., Rathod, S. and Singh, K. N. (2018). Improved ARIMAX model based on ANN and SVM approaches for forecasting rice yield using weather variables. *Indian Journal of Agricultural Sciences*, **88(12)**, 1909-1913.

Allen, P. G. (1994). Economic forecasting in agriculture. *International Journal of Forecasting*, **10**, 81-135.

Aloy, M., Mohamed, B., Gente, K. and Anne, P. (2010). Fractional integration and cointegration in stock prices and exchange rates. *Economics Bulletin*, **30(1)**, 115-129.

An, X., Jiang, D., Zhao, M. and Liu, C. (2012). Short time prediction of wind power using EMD and chaotic theory. *Communication in Nonlinear Science and Numerical Simulation*, **17(2)**, 1036-1042.

Andrésa, J. D., Lorca, P., Juez, F. J. C. and Fernando, S. L. (2011). Bankruptcy forecasting: A hybrid approach using Fuzzy c-means clustering and Multivariate Adaptive Regression Splines (MARS). *Expert Systems with Applications*, **38(3)**, 1866-1875.

Anjaly, K. N., Surendran, S., Babu, S. K. and Thomas, J. K. (2010). *Impact Assessment of Price Forecast: A Study of Cardamom Price Forecast by AMIC, KAU*. NAIP on Establishing and Networking of Agricultural Market Intelligence Centres in India. College of Horticulture, Vellanikkara.

Anjoy, P. and Paul, R. K. (2019). Comparative performance of wavelet-based neural network approaches. *Neural Computing and Applications*, **31(8)**, 3443-3453.

Bailey, T. M. (2015). Convergence of Rprop and variants. *Neurocomputing*, **159**, 90-95.

Balke, N. S. and Fomby, T. B. (1997). Threshold cointegration. *International Economic Review*, **38**, 627– 645.

Banerjee, P., Arčabić, V. and Lee, H. (2017). Fourier ADL cointegration test to approximate smooth breaks with new evidence from crude oil market. *Economic Modelling*, **67**, 114-124.

Bishop, M. C. (1995). *Neural networks for Pattern recognition*. Oxford University Press, New York.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, **31**, 307-327.

Box, G. and Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.

Brandl, B, Wildburger, U. and Pickl, S. (2009). Increasing of the fitness of fundamental exchange rate forecast models. *International Journal of Contemporary Mathematical Sciences*, **4(16)**, 779-798.

Brock, W. A., Scheinkman. J. A., Dechert, W. D. and LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*. **15**, 197-235.

Chaâbane, N. (2014). A hybrid ARFIMA and neural network model for electricity price prediction. *International Journal of Electrical Power & Energy Systems*, **55**, 187–194.

Campbell, J. Y. and Shiller, R. J. (1987). Cointegration and tests of present value models. *Journal of Political Economy*, **95**, 1062–1088.

Castle, J. L., Nicholas, W. P. F. and Hendry, D. F. (2010). Forecasting with equilibrium-correction models during structural breaks. *Journal of Econometrics*, **158** (**1**), 25-36.

Chattopadhyay, S. and Chattopadhyay, M. (2007). A Soft Computing technique in rainfall forecasting. *International Conference on IT, HIT*.

Che, J. and Wang, J. (2010). Short-term electricity prices forecasting based on support vector regression and Auto-regressive integrated moving average modeling. *Energy Conversion and Management*, **51(10),** 1911–1917.

Chen, C. F., Lai., M. C. and Yeh, C. C. (2012). Forecasting tourism demand based on empirical mode decomposition and neural network. *Knowledge-Based System*, **26**, 281-287.

Chen, S. Y. (2007). Forecasting exchange rates: a new nonparametric support vector regression. *The Journal of Quantitative and Technical Economics*, **5**, 142-150.

Cheung, Y-W. (1993). Tests for fractional integration: a monte carlo investigation. *Journal of Time Series Analysis*, **14(4)**, 331-345.

Choudhury, K., Jha, G. K., Das, P. and Chaturvedi, K. K. (2019). Forecasting Potato Price using Ensemble Artificial Neural Networks. *Indian Journal of Extension Education*, **55(1)**, 73-77.

Chuang, C. C. and Lee, Z. J. (2011). Hybrid robust support vector machines for regression with outliers. *Applied Soft Computing*, **11**(**1**), 64-72.

Clemen, R. (1989). Combining forecasts: a review and annotated bibliography with discussion. *International Journal of Forecasting*, **5**, 559-608.

Crino, S. and Brown, D. (2007). Global optimization with multivariate adaptive regression splines. *IEEE Transactions on Cybernetics*, **37(2)**, 333-340.

Das, P., Paul, A. K. and Paul, R. K. (2016). Non-linear mixed effect models for estimation of growth parameters in goats. *Journal of the Indian Society of Agricultural Statistics*, **70(3)**, 205–210.

Das, P., Jha, G. K., Lama, A. and Kumar, M. (2019). Price forecasting using EMD SVR hybrid model: An ensemble learning paradigm. *Eighth International Conference on Agricultural Statistics*, New Delhi, India.

David, S. A., Inácio, C. M. C. and Machado, A. J. T. (2019). Quantifying the predictability and efficiency of the cointegrated ethanol and agricultural commodities price series. *Applied Sciences*, **9(24)**, 1-14.

Deo, R. C., Kisi, O. and Singh, V. P. (2017). Drought forecasting in eastern Australia using multivariate adaptive regression spline, least square support vector machine and M5Tree model. *Atmospheric Research*, **184**, 149-175.

DES (Directorate of Economics and Statistics) Cost of cultivation of principal crops in India (various issues), Government of India, New Delhi.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, **84**, 427–431.

Dickey, D. A. and Fuller, W. A. (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, **49**, 1057–1072.

Diebold, F. and Mariano, R. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, **13**, 253–265.

Donaldson, R. G. and Kamstra, M. (1997). An artificial neural network-GARCH model for international stock return volatility. *Journal of Empirical Finance*, **4**, 17–46.

Duan, W. Q. and Stanley, H. E. (2011). Cross-correlation and predictability of financial return series. *Physica A*, **390(2)**, 290-296.

Elavarasan, D., Vincent, D. R., Sharma, V., Zomaya, A. Y. and Srinivasana, K. (2018). Forecasting yield by integrating agrarian factors and machine learning models: A survey. *Computers and Electronics in Agriculture*, **155**, 257-282.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. *Econometrica*, **50**, 987-1008.

Engle, R and Granger, C. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica*, **55**, 251–276.

Engle, R. F. and Yoo, B. S. (1987). Forecasting and testing in co-integrated systems. *Journal of Econometrics*, **35**, 143–159.

Ericsson, N. R. and MacKinnon J. G. (2002). Distributions of error correction tests for cointegration, *Econometrics Journal*, **5**, 285–318.

Fanellia, L. and Paruolo, P. (2010). Speed of adjustment in cointegrated systems. *Journal of Econometrics*, **158**(**1**), 130-141.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, **19**, 1-141.

Friedman, J. H. and Roosen, C. B. (1995). An introduction to multivariate adaptive regression splines. *Statistical Methods in Medical Research*, **4**, 197–217.

Fuller, W. A. (1976), *Introduction to Statistical Time Series*. New York, Wiley.

Garg, B., Aggarwal, S. and Sokhal, J. (2017). Crop yield forecasting using fuzzy logic and regression model. *Computers and Electrical Engineering*, **2**, 1-21.

Gooijer, J. G. D. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, **22**, 443– 473.

Granger, C. W. J. (1981). Some properties of time series data and their use in econometric model specification. *Journal of Econometrics*, **16**, 121-130.

Granger, C. W. J. and Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics,* **2(2)**, 111-120.

Granger, C. W. J. and Joyeux, R. (1980). An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, **1**, 15-29.

Guo, Z., Zhao, W., Lu, H. and Wang, J. (2012). Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model. *Renewable Energy*, **37(1)**, 241-249.

Hansen, B. E. and Seo, B. (2002). Testing for two-regime threshold cointegration in vector error correction models. *Journal of Econometrics*, **110**, 293–318.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Ontario.

Higgins, A., Prestwidge, D., Stirling, D. and Yost, J. (2010). Forecasting maturity of green peas: An application of neural networks. *Computers and Electronics in Agriculture*, **70(1)**, 151-156.

Holt, C. C. (1957). Forecasting seasonals and trends by exponentially weighted averages. O.N.R. Memorandum 52/1957, Carnegie Institute of Technology. Reprinted with discussion in 2004. *International Journal of Forecasting*, **20**, 5-13.

Hora, J. and Campos, P. (2015). A review of performance criteria to validate simulation models. *Expert Systems*, **32** (**5**), 578–595.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L., Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998). The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceeding of the Royal Society London A*, **454**, 909-995.

Ince, H. and Trafalis, T. (2006). A hybrid model for exchange rate prediction. *Decision Support Systems*, **42(2)**, 1054-1062.

Jacobs, R. (1998). Increased rates of convergence through learning rate adaptation. *Neural Networks*, **1**(**4**), 24-29.

Jarque, C. M. and Bera, A. K. (1987). A test for normality of observations and regression residuals. *International Statistical Review*, **55(2)**, 163–172.

Jebli, M. B. and Yousse, S. B. (2017). Renewable energy consumption and agriculture: evidence for co-integration and Granger causality for Tunisian economy. *International Journal of Sustainable Development and World Ecology*, **24(2)**, 149-158.

Jha, G. K., Thulasiraman, P. and Thulasiram, R. K. (2009). PSO based neural network for time series forecasting. *IEEE International Joint Conference on Neural Networks*, USA, 1422-1427.

Jha, G. K. and Sinha, K. (2014). Time-delay neural networks for time series prediction: An application to the monthly wholesale price of oilseeds in India. *Neural Computing and Applications*, **24**, 563-571.

Johansen, S. (1988). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, **12**, 231-254.

Johansen, S. (1992). Cointegration in partial systems and the efficiency of single equation analysis. *Journal of Econometrics*, **52**(**3**), 389-402.

Johansen, S. and Juselius, K. (1990). Maximum likelihood estimation and inference on cointegration with application to the demand for money. *Oxford Bulletin of Economics and Statistics*, **52**, 169-210.

Johansen, S. and Nielsen, M. O. (2018). Nonstationary cointegration in the fractionally cointegrated VAR model. *Journal of Time Series Analysis*, **40(4)**, 1-25.

Kaastra, I. and Boyd, M. (1996) Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, **10**, 215–236.

Kapetanios G, Shin, Y and Snell, A. (2006). Testing for cointegration in nonlinear smooth transition error correction models. *Econometric Theory*, **22**, 279–303.

Karlov, D. N., Zueva, V. N., Trukhan, D. A. and Belykh, A. A. (2019). Application of high-speed algorithms for training neural networks for forecasting financial markets. *Broad Research in Artificial Intelligence and Neuroscience*, **10**, 61-66.

Khashei, M. and Hajirahimi, Z. (2018). A comparative study of series ARIMA/MLP hybrid models for stock price forecasting. *Communication in Statistics-Simulation and Computation*, **48**(**9**), 1-16.

Khandelwal, I., Adhikari, R. and Verma, G. (2015). Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Procedia Computer Science*, **48**, 173-179.

Khaki, S. and Wang, L. (2019). Crop yield prediction using deep neural networks. *Frontiers in Plant Science*, **10**, 621-630.

Khazaei, J., Naghavi, M. R., Jahansouz, M. R. and Salimi-Khorshidi, G. (2008). Yield estimation and clustering of chickpea genotypes using soft computing techniques. *Agronomy Journal*, **100**, 1077-1087.

Kim, D. and Oh, H-S. (2009). EMD: A package for empirical mode decomposition and Hilbert spectrum. *The R Journal*, **1**, 41-45.

Kim, N., Ha, K. J., Park, N. W., Cho, J., Hong, S. and Lee, Y. W. (2019). A comparison between major artificial intelligence models for crop yield

prediction: Case study of the midwestern United States, 2006–2015. *International Journal of Geo-Information*, **8(5)**, 240-246.

Kim, D. and Oh, H. S. (2016). Empirical mode decomposition with missing values. *SpringerPlus*, **5**(**1**), 1-15.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, **55**, 307-319.

Kisi, O. and Parmar, K. S. (2016). Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution. *Journal of Hydrology*, **534**, 104-112.

Koc, E. K and Bozdogan, H. (2015). Model selection in multivariate adaptive regression splines (MARS) using information complexity as the fitness function. *Machine Learning*, **101(1-3)**, 35–58.

Komasi, M. and Sharghi, S. (2016). Hybrid wavelet-support vector machine approach for modelling rainfall-runoff process. *Water Science & Technology*, **73(8)**, 1937-1953.

Krishnan, P., Sharma, R. K., Dass, A., Kukreja, A., Srivastav, R., Singhal, R. J., Bandyopadhyay, K. K., Lal, K., Manjaiah, K. M., Chhokar, R.S. and Gill, S. C. (2016). Web-based crop model: Web InfoCrop – Wheat to simulate the growth and yield of wheat. *Computers and Electronics in Agriculture*, **127**: 324–335.

Kuan, C. M. and Liu, T. (1995). Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks. *Journal of Applied Econometrics*, **10(4)**, 347-364.

Kumar, P. (2011). Crop yield forecasting by adaptive neuro fuzzy inference system. *Mathematical Theory Model***, 1**, 1-7.

Kumar, R. R. and Jha, G. K. (2017). Examining the co-movement between energy and agricultural commodity prices in India. *Journal of the Indian Society of Agricultural Statistics*, **70(3)**, 241-252.

Kumar, S. and Kumar, N. (2012). Two factor fuzzy time series model for rice forecasting. *International Journal of Computational Mathematical Science*, **4**, 56–61.

Kumar, N., Ahuja, S., Kumar, V. and Kumar, A. (2010). Fuzzy time series forecasting of wheat production. *International Journal of Computer Science and Engineering*, **2**, 635–640.

Lama, A. (2013). A study on agricultural commodity price volatility using dynamic neural networks. Published M.Sc. Thesis, I.A.R.I. New Delhi, India.

Lama, A., Jha, G., Gurung, B., Paul, R. K., Bharadwaj, A. and Parsad, R. (2016). A comparative study on time-delay neural network and GARCH models for forecasting agricultural commodity price volatility. *Journal of the Indian Society of Agricultural Statistics*, **70(1)**, 7-18.

Lee, H., Lee, J. and Im, K. (2015). More powerful cointegration tests with non-normal errors. *Studies in Nonlinear Dynamics & Econometrics*, **19**(**4**), 397-413.

Lee, T. S., Chiu, C. C., Chou, Y. C. and Lu, C. J. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis*, **50**, 1113-1130.

Li, Y., He, Y., Su, Y. and Shu, L. (2016). Forecasting the daily power output of grid-connected photovoltaic system based on MARS. *Applied Energy*, **180**, 392-401.

Li, X., Song, Y., Yao, Z. and Xiao, R. (2018). Forecasting China's CO_2 emissions for energy consumption based on cointegration approach. *Discrete Dynamics in Nature and Society,* **18**, 1-9.

Liang, Y. (2009). Combing seasonal time series ARIMA method and neural networks with genetic algorithms for predicting the production value of the mechanical industry in Taiwan. *Neural Computation & Application*, **18**, 833-841.

Lu, C. J., Lee, T. S. and Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector machine. *Decision Support Systems*, **47(2)**, 115-125.

MacKinnon, J. G. (1996). Numerical distribution functions for unit root and cointegration tests. *Journal of Applied Econometrics*, **11**, 601–618.

MacKinnon, J. G., Haug, A. A. and Michelis, L. (1999). Numerical distribution functions of likelihood ratio tests for cointegration, *Journal of Applied Econometrics*, **14**, 563–577.

Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, N. J., Parzen, E. and Winkler, R. (1982). The accuracy of extrapolation (time series) methods: results of a forecasting competition. *Journal of Forecasting***, 1**, 111-153.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C. C. and Lin, C. C. (2018). e1071: A package for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering and naive Bayes classifier. *The R Journal*.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill. 2nd ed.

Montomery, D. C., Peck, E. A. and Vining, G. G. (2006). *Introduction to linear regression analysis*. 4th ed. Wiley-Interscience.

Moore, K. J., Kurt, M., Eriten, M., McFarland, D. M., Bergman, L. A. and Vakakisa, A. F. (2018). Wavelet-bounded empirical mode decomposition for measured time series analysis. *Mechanical Systems and Signal Processing*, **99**, 14-29.

Moosa, I. A. and Vaz, J. J. (2016). Cointegration, error correction and exchange rate forecasting. *Journal of International Financial Markets, Institutions and Money*, **44**, 21-34.

Nagpal, A. and Singh, V. (2019). Coupling multivariate adaptive regression spline (MARS) and random forest (RF): A hybrid feature selection method in action. *International Journal of Healthcare Information Systems and Informatics*, **14(1)**, 30-35.

Oguntunde, P. G., Lischeid, G. and Dietrich, O. (2018). Relationship between rice yield and climate variables in southwest Nigeria using multiple linear regression and support vector machine analysis. *International Journal of Biometeorology*, **62(3)**, 459-469.

Pannakkong, W., Huynh, V. and Sriboonchitta, S. (2019). A novel hybrid autoregressive integrated moving average and artificial neural network model for cassava export forecasting. *International Journal of Computational Intelligence Systems*, **12**(**2**), 1047-1061.

Parmezan, A. R. S., Souza, V. M. A. and. Batista, G. E. (2019). Evaluation of statistical and machine learning models for time series prediction: identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences*, **484**, 302-337.

Paul, R. K. (2015). ARIMAX-GARCH-WAVELET model for forecasting volatile data. *Model Assisted Statistics and Applications*, **10(3)**, 243-252.

Paul, R. K., Rathod, S., Singh, K. N., Mehar, S. K. and Ray, M. (2015). An improved autoregressive fractionally integrated moving average (ARFIMA) model using artificial neural network (ANN) to forecast spot prices of mustard. *International Conference on Recent Advances in Mathematics, Statistics and Computer Science,* Central University of South Bihar, India.

Phillips, P. C. B. (1987). Time series regression with a unit root. *Econometrica*, **55**, 277–301.

Phillips, P. C. B. and Ouliaris, S. (1990). Asymptotic properties of residual based tests for cointegration. *Econometrica*, **58**, 165–193.

Phillips, P. C. B. and Perron, P. (1988). Testing for a unit root in time series regression. *Biometrika*, **75**, 335-346.

Plakandaras, V., Papadimitriou, T. and Gogas, P. (2015). Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting*, **34**, 560-573.

Prasad, R., Deo, R. C., Li, Y. and Maraseni, T. (2018). Soil moisture forecasting by a hybrid machine learning technique: ELM integrated with ensemble empirical mode decomposition. *Geoderma*, **330**, 136-161.

Quenouille, M. H. (1957). *The analysis of multiple time-series* (2nd ed. 1968). London, Griffin.

Ratjen, A. M. and Kage, H. (2015). Forecasting yield via reference and scenario calculations. *Computers and Electronics in Agriculture*, **114**, 212–220.

Ray, M., Rai, A., Ramasubramanian, V. and Singh, K. N. (2016). ARIMA-WNN hybrid model for forecasting wheat yield time-series data. *Journal of the Indian Society of Agricultural Statistics*, **70(1)**, 63–70.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, San Francisco, 586-591.

Samuel, A. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, **3(3)**, 210-229.

Samui, P. and Karup, P. (2011). Multivariate adaptive regression splines and least square support vector machine for prediction of undrained shear strength of clay. *Applied Metaheuristic Computing*, **3(2),** 33-42.

Schultz, A. and Wieland, R. (1997). The use of neural networks in agroecological modelling. *Computers and Electronics in Agriculture*, **18**, 73-90.

Seid, M. (2016). Crop forecasting: Its importance, current approaches, ongoing evolution and organizational aspects. *FAO report*.

Sekulic, S. and Kowalski, B. R. (1992). MARS: A tutorial. *Journal of Chemometrics*, **6**, 199-216.

Sermpinis, G., Stasinakis, C., Theofilatos, K. and Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting*, **33**, 471–487.

Silva, J., Herazo, J. C. M., Millan, R. H. R., Lezama, O. B. P., Gamero, W. M. and Varela, N. (2019). Early warning method for the commodity prices based on artificial neural networks: SMEs case. *Procedia Computer Science,* **151**, 1243-1248.

Silveman, B. W. (1985). Some aspects of the spines smoothing approach to nonparametric regression curve fitting. *Journal of Royal Statistical Society*, series B, **47**, 1-52.

Suhartono, S., Prabowo, H., Prastyo, D. and Lee, M. H. (2019). New hybrid statistical method and machine learning for PM10 prediction. *Soft Computing in Data Science*. 1st ed.

Sugiyama, M. and Kawanabe, M. (2012). *Machine Learning in Non-Stationary Environments-Introduction to Covariate Shift Adaptation.* The MIT Press, Cambridge, Massachusetts, London, England. 2nd ed.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifier. *Neural Processing Letters*, **9(3)**, 293-300.

Suykens, J. A. K., Van, G. T., De Brabanter, J., De, M. B., and Vandewalle, J., (2002). *Least Squares Support Vector Machines*. World Science, Singapore.

Truchis, G. D. and Keddad, B. (2016). On the risk co-movements between the crude oil market and U.S. dollar exchange rates. *Economic Modelling*, **52(A)**, 206-215.

Uno, Y., Prasher, S. O., Lacroix, R., Goel, P. K., Karimi, Y., Viau, A. and Patel, R. M. (2005). Artificial neural networks to predict corn yield from compact airborne spectrographic imager data. *Computers and Electronics in Agriculture*, **47**, 149-161.

Vapnik, N. V. (1998). *Statistical Learning Theory*. Wiley-Interscience. 1st ed.

Volterra, V. (1930). *Theory of functionals and of integro-differential equations*. New York: Dover.

Voyant, C., Museli, M., Paoli, C. and Nivet, M. (2011). Optimization of an artificial neural network dedicated to multivariate forecasting of daily global radiation. *Energy*, **36**, 348-359.

Wang, T., Zhang, M., Yu, Q. and Zhang, H. (2012). Comparing the application of EMD and EEMD on time-frequency analysis of seimic signal. *Journal of Applied Geophysics*, **83**, 29-34.

Wegener, C., Spreckelsen, C. V., Basse, T. and Mettenheim, H. J. V. (2016). Forecasting government bond yields with neural networks considering cointegration. *Journal of Forecasting*, **35**, 86–92.

West, K. D. (1988). Asymptotic normality, when regressors have a unit root. *Econometrica*, **56**, 1397–1417.

Widodo, A., Kusdarwati, H. and Samingun, H. (2019). Modeling average prices of garlic in Indonesia. *Journal of Engineering and Applied Sciences*, **14(21)**, 7943-7950.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, **6**, 324–342.

Wu, Z., Jiang, C., Mack, C., Deng, B. and Chen, J. (2018). Hybrid improved EMD-BPNN model for the prediction of sea surface temperature. *Ocean Science Discussions*, **14**, 1-20.

Yang, H. F. and Chen, Y. P. P. (2019). Hybrid deep learning and empirical mode decomposition model for time series applications. *Expert Systems with Applications*, **120**, 128-138.

Yule, G. U. (1927). On the method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society London, Series A*, **226**, 267– 298

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, **50**, 159-175.

Zhang, G. P., Zhang, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, **14**, 35-62.

Zhang, Q. M. and Zhang, G. P. (2008). Trend time series modeling and forecasting with neural networks. *IEEE Transactions on Neural Networks*, **19**, 808-816.

Zhang, W. and Goh, A. T. C. (2015). Nonlinear structural modeling using multivariate adaptive regression splines. *Computers and Concrete,* **16(4)** 569-585.

Zhang, W. and Goh, A. T. C. (2016). Multivariate adaptive regression splines and neural network models for prediction of pile drivability. *Geoscience Frontiers*, **7**, 45-52.