Neural Comput & Applic
DOI 10.1007/s00521-012-1264-z

ORIGINAL ARTICLE

Time-delay neural networks for time series prediction: an application to the monthly wholesale price of oilseeds in India

Girish K. Jha · Kanchan Sinha

Received: 2 May 2012 / Accepted: 30 October 2012

Abstract Agricultural price forecasting is one of the challenging areas of time series forecasting. The feed-forward time-delay neural network (TDNN) is one of the promising and potential methods for time series prediction. However, empirical evaluations of TDNN with autoregressive integrated moving average (ARIMA) model often yield mixed results in terms of the superiority in forecasting performance. In this paper, the price forecasting capabilities of TDNN model, which can model nonlinear relationship, are compared with ARIMA model using monthly wholesale price series of oilseed crops traded in different markets in India. Most earlier studies of forecast accuracy for TDNN versus ARIMA do not consider pretesting for nonlinearity. This study shows that the nonlinearity test of price series provides reliable guide to post-sample forecast accuracy for neural network model. The TDNN model in general provides better forecast accuracy in terms of conventional root mean square error values as compared to ARIMA model for nonlinear patterns. The study also reveals that the neural network models have clear advantage over linear models for predicting the direction of monthly price change for different series. Such direction of change forecasts is particularly important in economics for capturing the business cycle movements relating to the turning points.

Keywords ARIMA · Price forecasting · Time-delay neural networks

G. K. Jha (✉) · K. Sinha
Indian Agricultural Research Institute, Delhi, India
e-mail: girish.stat@gmail.com

1 Introduction

Forecasts of agricultural prices are intended to be useful for farmers, governments and agribusiness industries. The ability to accurately forecast the price of agricultural commodities is therefore an important concern in both policy and business circles. Price forecasts are largely made by using time series approaches. In time series modeling, past observations of the same variable are collected and analyzed to develop a model describing the underlying relationship. The developed model is then used to extrapolate the time series into the future. In the last few decades, much effort has been devoted to the development and improvement of time series forecasting models.

One of the most important and widely used time series models is the autoregressive integrated moving average (ARIMA) model. The popularity of the ARIMA model is due to its statistical properties as well as the well-known Box–Jenkins methodology in the model building process. However, the major limitation of ARIMA model is the preassumed linear form of the model. That is, a linear correlation structure is assumed among the time series values, and therefore, no nonlinear patterns can be captured by the ARIMA model. Although linearity is a useful assumption and a powerful tool in many areas, it became increasingly clear in the early 1980s that the approximation of the linear models to complex real-world problem is not always satisfactory. For example, sustained animal population size cycles (the well-known Canadian lynx data) and sustained solar cycles (annual sunspot numbers) are not suitable for linear models. The last two decades has seen the development of a substantial literature dealing with testing and modeling nonlinearity of time series data. Several nonlinear time series models, such as the bilinear model [12] and the threshold autoregressive (TAR) model

Published online: 15 November 2012

[22], have been developed. These nonlinear models are "model-driven approaches" in which we first identify the type of relation among the variables (model selection) and afterward estimate the selected model parameters.

The recent upsurging research activities in artificial neural network (ANN) as well as their numerous successful forecasting applications suggest that they can also be an important candidate for time series forecasting [5, 8, 16, 24]. As opposed to the traditional model-based methods, ANN is a data-driven, self-adaptive, nonlinear, nonparametric statistical method in that there are few a priori assumptions about the models for problems under study [2]. ANN models can be useful for nonlinear processes that have an unknown functional relationship and as a result are difficult to fit [6]. The process of constructing the relationship between the input and output variables is addressed by certain general purpose learning algorithm. ANN modeling has attracted as a new technique for estimation and forecasting in many fields of study including agriculture, economics and statistics. Investigators have been attracted by ANN's freedom from restrictive assumption such as linearity that is often needed to make the traditional mathematical models tractable. Most uses of ANN in economics have so far been in financial market, in part because traditional approaches have had low explanatory power and in part because the ANN approaches require abundant data. The use of the ANN model in applied work is generally motivated by a mathematical result stating that under mild regularity conditions, a relatively simple ANN model is capable of approximating any Borel-measurable function to any given degree of accuracy [11]. Such an approximator would still contain a finite number of parameters. However, despite the popularity and the sheer power of these models, the empirical forecasting performance of neural network models has been rather inconclusive [19, 24].

Neural networks and traditional time series techniques have been compared in several studies. Sharda and Patil [20] use 75 out of the 111 time series from the well-known M-competition [24] as test cases and find that the neural networks perform as well as the automatic Box–Jenkins procedure. Kohzadi et al. [17] demonstrate that the neural networks are superior to ARIMA methods for forecasting commodity price. In the context of economic data, Swanson and White [21] investigate the performance of neural network models in forecasting nine quarterly seasonality-adjusted US macro-econometric time series, finding that they generally outperform traditional economic approaches even where there is no explicit nonlinearity. Hervai et al. [15] consider linear and ANN models for forecasting seasonally unadjusted monthly data on European industrial production series and conclude that linear models generally produce more accurate post-sample forecast than neural network models at horizons of up to a year, in terms of root mean square error. Zou et al. [25] employ ARIMA model and neural networks to predict average soil water content and mention the better performance of latter on the basis of statistical parameters. Crone et al. [5] report that neural networks are capable of handling complex data, including short and seasonal time series, beyond prior expectations. However, most earlier studies of forecast accuracy for neural networks versus linear models do not consider pretesting for nonlinearity. This will enable us to examine whether nonlinearity tests of the series provide any indication to post-sample forecast accuracy of the models. Besides, literature suggests that the performance of nonlinear model should be evaluated on the basis of percentage of forecasts that correctly predict the direction of change instead of measures based on error terms. The above facts clearly indicate that there is a lack of systematic investigation on forecasting agricultural price series using the neural network models. Hence, in this paper, an effort is made to evaluate the suitability of a time-delay neural network as a price forecasting model in agriculture in comparison with the Box–Jenkins methodology using monthly wholesale price series of major oilseed crops of India.

Moreover, the oilseed price series were deliberately chosen for the study because the domestic edible oil price is very responsive to international price as huge quantity is imported to meet domestic requirement of our country. The rest of the paper is organized as follows. The details of time series forecasting models used in this paper are described in Sect. 2. Empirical results obtained from real data are given in Sect. 3. Finally, Sect. 4 concludes the paper.

2 Time series forecasting models

2.1 The ARIMA model

In an autoregressive integrated moving average model, time series variable is assumed to be a linear function of past actual values and random shocks. An ARIMA (p, q) model is defined by the following equation [1]:

$$y_t = \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots + \emptyset_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (1)$$

that is,

$$\left(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p\right) y_t = \left(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q\right) \varepsilon_t \quad (2)$$

or

$$\emptyset(B) y_t = \theta(B) \varepsilon_t \quad (3)$$

where B is the backshift operator defined by $By_t = y_{t-1}$, $\emptyset(B)$ and $\theta(B)$ are polynomials of degree p and q in B. y_t is the price at time period t, and ε_t is a random error at time period t and is assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2. A generalization of ARIMA models, which incorporates a wide class of nonstationary time series, is obtained by introducing 'differencing' in the model. Thus, an ARIMA model that can represent homogeneous nonstationary behavior is written as follows:

$$\emptyset(B)(1-B)^d y_t = \theta(B)\varepsilon_t \tag{4}$$

that is,

$$\emptyset(B)z_t = \theta(B)\varepsilon_t \tag{5}$$

where $z_t = \nabla^d y_t$, $\nabla = (1-B)$ is the differencing operator. In general, an ARIMA model is characterized by the notation ARIMA (p, d, q) where p, d and q denote orders of autoregression, integration (differencing) and moving average, respectively. ARIMA is a parsimonious approach which can represent both stationary and nonstationary processes. When $d = 0$, the ARIMA (p, d, q) model reduces to ARIMA (p, d) model. In practice, d is usually 0, 1 or at most 2.

The first step in the process of ARIMA modeling is to check for the stationarity of the series as the estimation procedure is available only for stationary series. A series is said to be stationary if its statistical characteristics such as the mean and the autocorrelation structure are constant over time. Stochastic trend of the series is removed by differencing, while logarithmic transformation is employed to stabilize the variance. After the appropriate transformation and differencing, multiple ARIMA models are chosen on the basis of autocorrelation function (ACF) and partial autocorrelation function (PACF) that closely fit the data. Then, the parameters of the tentative models are estimated through any nonlinear optimization procedure such that an overall measure of errors is minimized or the likelihood function is maximized. Lastly, diagnostic checking for model adequacy is performed for all estimated models through plot of residual ACF and via portmanteau test like Box–Pierce and Ljung–Box tests. In this study Ljung–Box test is used. The most suitable ARIMA model is selected using the smallest Akaike Information Criterion (AIC) or Schwarz–Bayesian Criterion (SBC) value and the lowest root mean square error (RMSE).

2.2 The neural network models

Neural network models are computational methods that mimic the behavior of the human brain's central nervous system. They are considered as a class of generalized nonlinear, nonparametric, data-driven statistical methods. A general neural network architecture consists of an input layer that accepts external information, one or more hidden or middle layer that provide nonlinearity to the model and an output layer that provides the target value. Each layer contains one or more nodes. All the layers in a multilayer neural network are connected through an acyclic arc.

Time series data can be modeled using neural network in two possible ways. The first way is to explicitly represent time in the form of recurrent connections from output nodes to the preceding layer [10]. The second way is to provide the implicit representation of time, whereby a static neural network-like multilayer perceptron is bestowed with dynamic properties [14]. A neural network can be made dynamic by embedding either long-term or short-term memory, depending on the retention time, into the structure of a static network. For temporal data processing, we require some form of short-term memory to make neural network dynamic. One simple way of building short-term memory into the structure of a neural network is through the use of time delay, which can be implemented at the input layer of the neural network. An example of such architecture is a time-delay neural network (TDNN), presented in Fig. 1, has been employed for the present study.

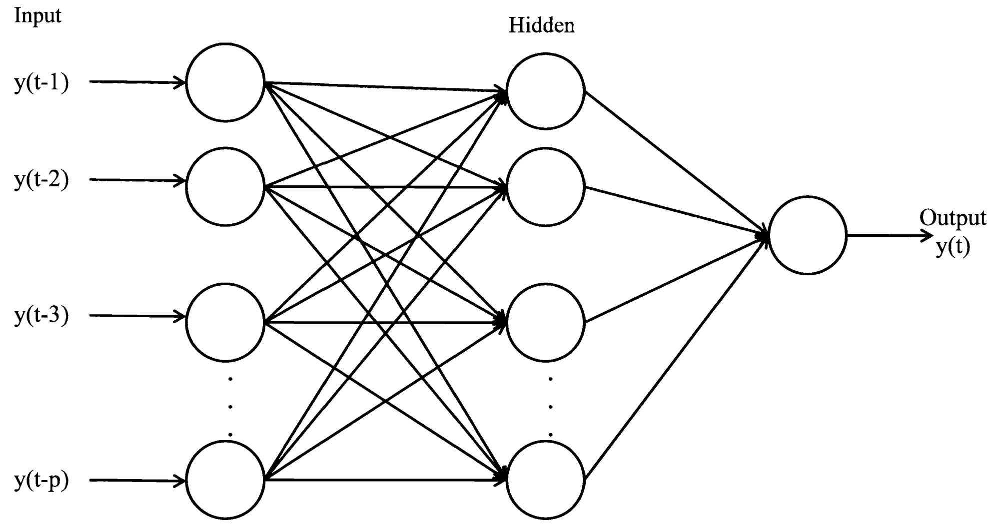
The neural network structure for a particular problem in time series prediction includes determination of the number of layers and total number of nodes in each layer. This is usually determined through experimentation of the given data as there is no theoretical basis for determining these parameters. As mentioned earlier, it has been proved that neural networks with one hidden layer can approximate any nonlinear function given sufficient number of nodes at hidden layer and adequate data points for training. In this study, we use neural network with one hidden layer. In time series analysis, the determination of the number of input nodes which are lagged observations of the same variable plays a crucial role as it helps in modeling the autocorrelation structure of the data. The determination of the number of output nodes is relatively easy. In this study, one output node is used, and multistep ahead forecasting is done using the iterative procedure as used in Box–Jenkins method. This involves use of forecast value as an input for forecasting the future value. It is always better to select the model with small number of nodes at hidden layer as it improves the out-of-the sample forecasting performance and also avoids the problem of overfitting.

The general expression for a multilayer feed-forward time-delay neural network is given by

$$y_{t+1} = g\left(\sum_{j=0}^{q} \propto_j f\left(\sum_{i=0}^{p} \beta_{ij} y_{t-i}\right)\right) \tag{6}$$

where $y_{t+1} = \ln(y_{t+1}/y_t)$. f and g denote the activation function at hidden and output layer, respectively, p is number of input nodes (tapped delay), q is the number of

Fig. 1 Time-delay neural network (TDNN) with one hidden layer



hidden nodes, β_{ij} is the weight attached to the connection between ith input node and the jth node of hidden layer, α_j is the weight attached to the connection from jth hidden node to the output node, and y_{t-i} is the ith input (lag) of the model. Each node of the hidden layer receives the weighted sum of all inputs including a bias term for which the value of input variable will always take a value one. This weighted sum of input variables is then transformed by each hidden node using the activation function f which is usually nonlinear sigmoid function. In a similar fashion, the output node also receives the weighted sum of the output of all hidden nodes and produces an output by transforming the weighted sum using its activation function g. In time series analysis, f is often chosen as logistic sigmoid function and g as an identity function. For p tapped delay nodes, q hidden nodes, one output node and biases at both hidden and output layers, the total number of parameters (weights) in a three-layer feed-forward neural network is $q(p + 2) + 1$.

For a univariate time series forecasting problem, past observations of a given variable serve as the input variables. The neural network model attempts to map the following function

$$y_{t+1} = f\left(y_t, y_{t-1} \ldots y_{t-p+1}, w\right) + \varepsilon_{t+1} \qquad (7)$$

where y_{t+1} pertains to the observation at time $t + 1$, p is the number of lagged observation, w is the vector of network weights, and ε_{t+1} is the error term at time $t + 1$. Hence, the neural network acts like nonlinear autoregressive model.

In this paper, we divide the data series into two parts: training set and testing set. The training set is used for parameter estimation as well as to measure network generalization, and testing set provides out-of-the sample performance. The procedure of early stopping is adopted for terminating the estimation process. The 20 percent of the training set is randomly held as a validation set to optimize the model complexity. The error measured with validation set is monitored during the training process which often shows a decrease at first followed by an increase as the network starts to overfit. The training is stopped at the point of smallest validation error in order to obtain a network having good generalization performance.

Suppose a time series data contain N observations y_1, y_2, …,y_N and out of N data points, n observations y_1, y_2, …, y_n are available for training purposes, and the model contains p lagged observations as input variable then $n - p$ patterns will be available for training the network for one-step ahead forecasting. This implies that y_1, y_2, … , y_p will serve as first input patterns for predicting the target output y_{p+1}. The last training pattern will be y_{n-p}, y_{n-p+1}, … ,y_{n-1} for predicting the target output y_n. Once the number of layers and total number of nodes in each layer has been determined, the network is ready for training, a parameter estimation process. The objective of training is minimization of an error function that measures the misfit between the predicted value and the actual value for any given value of w. The error function which is widely used is given by the sum of the squares of the error between the predicted value $\hat{y}_t$ for time t and the corresponding target value y_t at time t, so that we minimize

$$E(w) = 1/2 \sum_{t=p+1}^{n} [y_t - \hat{y}_t]^2 \qquad (8)$$

where the factor 1/2 is included for mathematical simplification. The error surface for multilayer feed-forward neural network with nonlinear activation function is complex and believed to have many local and global minima.

2.3 Nonlinearity test

In this study, we apply a nonlinearity test given by McLeod and Li [18]. This test is designed to test the null hypothesis of linearity against different types of possible nonlinearity. The test is based on the autocorrelations of the squared residuals. In this test, the residuals are obtained after fitting the ARIMA model to the difference time series of n sample observations. The test statistic is given as [18]

$$Q = n(n+2)\sum_{i=1}^{h}\frac{r^2(i)}{n-i} \quad (9)$$

where $r(i)$ is the autocorrelations of the squared residuals and h is the number of autocorrelations. Under the null hypothesis of linearity, this statistic is asymptotically distributed as a chi-square distribution with h degrees of freedom.

3 Empirical results

3.1 Data and implementation

The study uses the monthly wholesale price (Rs. per quintal) of major oilseeds crops of India, viz., soybean, groundnut and rapeseed and mustard traded in Indore (Madhya Pradesh), Rajkot (Gujarat) and Delhi markets, respectively, to evaluate the prediction power of two approaches. The basic characteristics of the price series used in the experiments are presented in Table 1. Data on rapeseed and mustard and groundnut are obtained from the various issues of "Agricultural Prices in India" published by the Directorate of Economics and Statistics, Government of India, while data on soybean are collected from the Web site of the Soybean Processors Association of India (SOPA), Indore, Madhya Pradesh. In this study, all estimation and forecasting of ARIMA model is done using SAS/ETS 9.2, while the neural network modeling is implemented using neural network toolbox of MATLAB 7.10. The price data are monthly averages. As mentioned earlier, in all cases we divide the data into two sets, namely training set and the testing set. The last 12-month price data are retained for testing purpose. The training set will be the one used for the modeling procedure and in-sample prediction, and the testing set will be kept for post-sample forecasting. The training set for the soybean series, groundnut series and rapeseed and mustard series consists of about 216, 260 and 360 observations, respectively.

In order to compare the performance of TDNN and ARIMA model, we follow the same five-step modeling procedures, namely preprocessing, identification, estimation, diagnostic checking and evaluation, for all three price series. Although neural networks are said to be universal approximators, it has been reported that they are not able to model a nonstationary time series data efficiently [23]. Therefore, data preprocessing becomes essential before training, so that a stationary time series should be used for modeling. We apply the natural choice of logarithmic transformation in this context for each series and then the augmented Dickey–Fuller test to test for unit roots. After transformation, each series is differenced in order to make the series stationary in mean as price data are trended and nonstationary in nature. We find the ARIMA structure of differenced series, based on the autocorrelation function (ACF), the partial autocorrelation function (PACF) and the AIC.

We find the best time-delay neural network with single hidden layer for this study. Following the previous studies [15, 24], the logistic and identity function have been used as activation function for the hidden nodes and output node, respectively. We focus primarily on one-step ahead forecasting, and the multistep ahead forecasting is done using iterative procedure so only one output node is employed. Hence, the model uncertainty is associated only with the number of tapped delay (p), which is the number of lagged observations in this case, and the number of hidden layer nodes (q). The number of tapped delay and hidden nodes was determined with the help of experimentation. We use multiple retries, with different random starting points, in order to avoid local minima and find the global minimum. We varied the number of input nodes from 1 to 6 and the number of hidden nodes from 2 to 10 with an increment of 2 with basic cross-validation method. Thus, different numbers of neural network models were

Table 1 Characteristics of the price series used in experiments

Crop	Market	Sample size	Start date	End date	Mean (Rs/quintal)	SD (Rs/quintal)
Soybean	Indore	228	October 1991	September 2010	1256.36	472.54
Groundnut	Rajkot	272	May 1988	December 2010	1520.04	617.98
Rapeseed & mustard	Delhi	372	January 1980	December 2010	1288.38	741.33

tried for each series before arriving at the final structure of the model. Essentially, the process of exploration and exploitation is carried out to obtain the best model for the given series. There are many variations of the backpropagation algorithm used for training feed-forward networks. In this study, the Levenberg–Marquardt algorithm [13] which has been designed to approach second-order training speed without computing the Hessian matrix has been employed. It has been shown [9] that this algorithm provides the fastest convergence for moderately sized feed-forward neural network used on function approximations problems. A typical TDNN structure with one hidden layer is denoted by I:Hs:Ol, where I is the number of nodes in the input layer, H the number of nodes in the hidden layer, O the number of nodes in the output layer, s denotes the logistic sigmoid transfer function, and l indicates the linear transfer function. The forecasting ability of both models is assessed with respect to two common performance measures, viz., the root mean squared error (RMSE) and the mean absolute deviation (MAD). The RMSE measures the overall performance of a model and is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{10}$$

where y_t is the actual value for time t, $\hat{y}_t$ is the predicted value for time t, and n is the number of predictions. The second criterion, the mean absolute deviation, is a measure of average error for each point forecast and is given by:

$$\text{MAD} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{11}$$

where the symbols have the same meaning as above.

In order to examine whether nonlinearity test provides any reliable guide for post-sample forecast accuracy, we apply nonlinearity test [18] to all series in this study. This test is designed to test the null hypothesis of linearity against different types of possible nonlinearity. This test is based on the autocorrelations of the squared residuals. In this study, autocorrelations up to 24 lags are used for computing the test. Further, Clements and Smith [4] argue that the value of nonlinear model forecast may be better reflected by the direction of change. Accordingly, we also compute the percentage of forecasts that correctly predict the direction of change as part of post-sample forecast accuracy.

4 Discussion

The first and foremost step in time series analysis is to plot the data. Figure 2a shows the time series plot of average monthly price of soybean from October 1991 to September 2010. We can see that there is a positive trend over time which indicates the nonstationary nature of the series. Similar trend was observed in case of groundnut and rapeseed and mustard. As mentioned earlier, we applied natural logarithmic transformation to the data to stabilize the variance. Logarithmic transformation is used for data which can take on both small and large values and is characterized by an extended right-hand tail distribution. Logarithmic transformation is one of the data processing techniques which also convert multiplicative or ratio relationship to additive which is believed to simplify and improve neural network training. We note that the series is nonstationary even after logarithmic transformation and shows a stochastic upward trend. We apply the augmented Dickey–Fuller test for each level and transformed series to test for unit root, and the results are provided in Table 2. The table values clearly show the nonstationarity of level and transformed series. This is also inferred from the ACF and PACF plot of level and transformed series. Therefore, we use first differencing for all price series. The first differenced series are found to be stationary as indicated in Fig. 2b and Table 2; hence, further differencing was not required. Here, differencing is preferred over detrending considering the stochastic trend of price data. The ACF and PACF of difference series do not show strong and consistent seasonal pattern. The absence of strong seasonality is further confirmed by obtaining seasonal indices (see Table 3) through multiplicative model of decomposition analysis.

After logarithmic transformation and first differencing, we model the relative change in the price series which also has meaningful economic interpretation. We obtain the best ARIMA model for each series based on the lowest AIC and BIC information criteria as well as lowest RMSE and MAD value. We select ARIMA (1, 1, 0), ARIMA (0, 1, 1) and ARIMA (2, 1, 0) model for soybean, groundnut and rapeseed and mustard series, respectively. Due importance is given to the well-behaved residuals while selecting the best model.

We find the best time lagged neural network with single hidden layer for each series by conducting an experiment with the basic cross-validation method. Out of a total of 24 neural network structures, a neural network model with two input nodes and three hidden nodes (2:3s:1l) performs better than other competing model in respect of out-of-the sample forecasting accuracy measures for soybean series. In case of groundnut data, a neural network with one tapped delay and nine hidden nodes (1:9s:1l) provides the best test results out of a total of 21 neural network structures. Similarly, a TDNN with two lagged observations as input node and eight hidden nodes (2:8s:1l) shows minimum training and testing RMSE. The issue of finding a parsimonious model is taken into account while selecting the

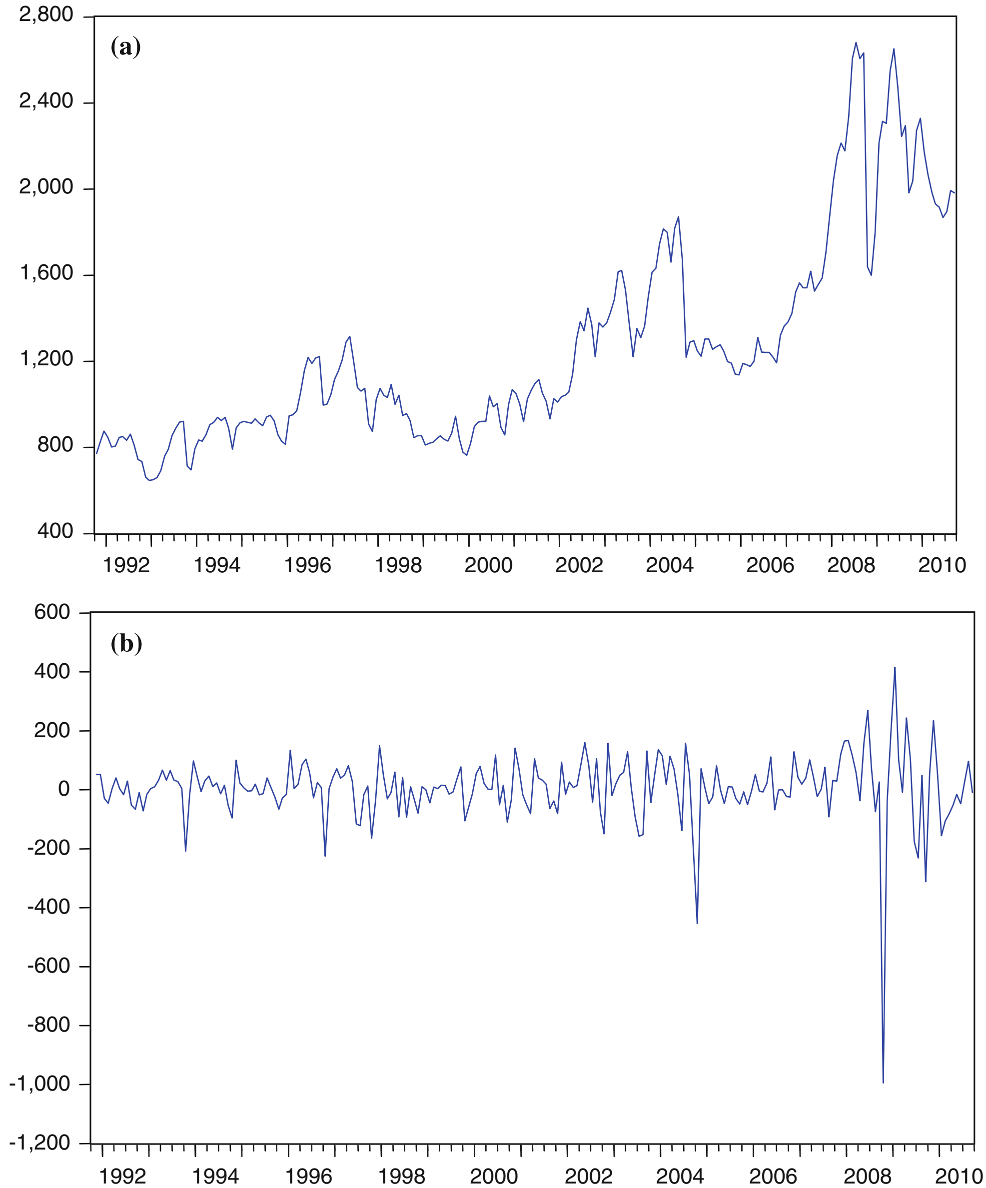


Fig. 2 Soybean monthly price data (Rs/quintal) from October 1991 to September 2010. **a** Raw price data, **b** first differenced price series

Table 2 Augmented Dickey–Fuller stationarity test for different series

Null hypothesis	Level series		Logarithmic transformed series		First difference of transformed series	
	t-statistic	*P* value	*t*-statistic	*P* value	*t*-statistic	*P* value
Soybean series has a unit root	−1.951	0.308	−1.557	0.502	−11.666	<0.0001
Groundnut series has a unit root	−0.879	0.794	−1.237	0.658	−17.595	<0.0001
Rapeseed and mustard series has a unit root	−0.737	0.830	−1.321	0.621	−17.428	<0.0001

best model for each price series. The parsimonious models not only have the recognition ability but also have the more important generalization ability. Further, nonlinearity test statistic is computed for each series in order to decipher true nonlinear pattern. Table 4 shows the results of McLeod and Li nonlinearity test. Table values reveal strong rejection of linearity only in case of rapeseed and mustard. In this study, our interest centers on short-term forecasting, and hence, we consider forecast horizon of up to a year. In terms of the forecast horizon, we include results for 1, 3, 6 and 12 months ahead forecast. Table 5 gives comparative results for the best ARIMA and TDNN

Table 3 Seasonal indices of wholesale price of different oilseed crops

Month crop	January	February	March	April	May	June	July	August	September	October	November	December
Soybean	0.98	0.99	1.00	1.04	1.06	1.06	1.04	1.04	1.01	0.90	0.92	0.95
Groundnut	0.99	0.99	1.00	1.03	1.01	1.02	1.03	1.05	0.99	0.95	0.96	0.98
Rapeseed & mustard	1.00	0.95	0.92	0.93	0.95	0.99	1.04	1.04	1.03	1.04	1.06	1.04

Table 4 McLeod and Li nonlinearity test for different series

Series	Value	*P* value
Soybean series	9.73	0.99
Groundnut series	25.60	0.37
Rapeseed & mustard series	87.82	<0.0001

Table 5 Comparative results of the series in terms of post-sample RMSE ratio of TDNN and ARIMA model

Series	1 Month ahead	3 Months ahead	6 Months ahead	12 Months ahead
Soybean	6.24	1.03	0.85	0.56
Groundnut	1.49	1.39	2.69	0.97
Rapeseed & mustard	1.43	0.08	0.19	0.17

models with respect to the post-sample RMSE ratio, which is computed as RMSE (TDNN)/RMSE (ARIMA). We can see from Table 5 that for all series the ratio is smaller than 1 for the forecast horizon of 12 months, suggesting a better performance of TDNN over ARIMA model. However, we observe that ARIMA model performs better than TDNN model for the forecast horizon of 1 month as RMSE ratio is more than one with respect to all series. It is obvious from the Table that, in general, TDNN model performs better in case of 6 and 12 months ahead forecasting, while ARIMA models dominate in case of 1- and 3-month forecast horizons. At this juncture, it is worth mentioning that for all cases the best neural network model in terms of test RMSE is obtained for a forecasting horizon of 12 months, and the same model is used for other forecast horizons. In this context, several researchers [15, 21] have recommended that a specific neural network model should be selected for each forecast horizon which implies that p and q may vary over forecast horizon. This will in general improve the performance of TDNN model with respect to each forecast horizon. The multiple model approach is not of much advantage in case of ARIMA model [3].

As indicated earlier, we also compare the prediction abilities of both models with respect to the mean absolute deviation (MAD), which provide similar trend as RMSE for all series with respect to all forecasting horizons. Hence, the results related to MAD measure are not presented in the text. Table 5 reveals that the test RMSE ratio is smaller than 1 for all horizons except 1 month for rapeseed and mustard series. This clearly suggests better performance for TDNN over ARIMA model for a truly nonlinear series. Hence, nonlinearity test provides a fairly good indication to post-sample forecast accuracy for neural network models. As mentioned earlier, forecast accuracy is measured here by the RMSE and by the percentage of forecasts that correctly predict the direction of change by having forecast and the actual value of the same sign. With 1 year of post-sample data, we have 12 one-step ahead forecast errors. The number of forecast errors decreases as the forecast horizon increases, so we calculate direction of change only for forecast horizon of 1, 3 and 6 months with 12, 10 and 7 forecast errors, respectively. We repeat the best model with regard to different series for five times in order to compute the percentage of forecasts of correct sign, and the results are given in Table 6. The implications of the direction of change results of Table 6 are, however, different from the results based on RMSE. At horizon 1, 3 and 6 months, the neural network model always has a larger percentage of correct sign than the ARIMA model for all (both linear and nonlinear) series. In this context, Dacco and Satchell [7] have shown that RMSE type measures may be inappropriate for nonlinear models, since these measures can imply that the nonlinear model is less accurate than a linear one even when the nonlinear model is the true data-generating process. In effect, the nonlinear model may generate more variation in forecast values than a linear model, and hence, it may be unduly penalized for errors that are large in magnitude. Thus, the results relating to direction of change imply that the relative forecasting performance of TDNN and ARIMA models crucially depends on how performance is measured.

Table 6 Post-sample percentage of forecasts of correct sign

Series	1 Month ahead		3 Months ahead		6Months ahead	
	ARIMA	TDNN	ARIMA	TDNN	ARIMA	TDNN
Soybean	42	55	46	54	57	60
Groundnut	53	58	54	62	51	66
Rapeseed & mustard	50	67	44	68	49	71

5 Conclusions

This study has compared ARIMA and TDNN model both in terms of modeling and forecasting using monthly wholesale price data of oilseed crops namely soybean, groundnut, rapeseed and mustard traded in Indore, Rajkot and Delhi markets of India. The aim of the study pertains to short-term price forecasting up to 1 year with multiple forecast horizons, namely 1, 3, 6 and 12 months. The TDNN models provide better forecast accuracy in terms of conventional RMSE as compared to ARIMA model for nonlinear relationship. We find that nonlinearity of series plays a fairly good role in providing reliable guide to post-sample forecast accuracy of ARIMA and TDNN models in terms of RMSE for these price series. Our study clearly suggests that before adopting any nonlinear model, one need to check whether the series is indeed nonlinear. For rapeseed and mustard series which is nonlinear in nature, TDNN performance was better than ARIMA for all forecast horizons except for 1 month ahead. It may be because we find best neural network model for 12-month forecast horizon; hence, it has been suggested in the literature that optimum network should be selected for each forecast horizon. However, TDNNs perform substantially better than linear models in predicting the direction of change for these series and hence may be preferred than linear models in the context of predicting turning point, which is more relevant in case of price forecasting.

References

1. Box GEP, Jenkins GM, Reinsel GC (1994) Time series analysis: forecasting and control. Pearson Education, Upper Saddle River
2. Cheng B, Titterington DM (1994) Neural networks: a review from a statistical perspective. Stat Sci 9:2–54
3. Clements MP, Hendry DF (1996) Multi-step estimation for forecasting. Oxf Bull Econ Stat 58:657–684
4. Clements MP, Smith J (1997) The performance of alternative forecasting methods for SETAR models. Int J Forecast 13:463–475
5. Crone SF, Hibon M, Nikolopoulos K (2011) Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. Int J Forecast 27:635–660
6. Darbellay GA, Slama M (2000) Forecasting the short-term demand for electricity: do neural networks stand a better chance? Int J Forecast 16:71–83
7. Dacco R, Satchell S (1999) Why do regime-switching models forecast so badly. J Forecast 16:1–16
8. De Gooijer JG, Hyndman RJ (2006) 25 years of time series forecasting. Int J Forecast 22:443–473
9. Demuth H, Beale M (2002) Neural network toolbox user's guide. Mathworks, Natic
10. Elman J (1990) Finding structure in time. Longtive Sci 14:179–211
11. Fine TL (1999) Feed forward neural network methodology. Springer, New York
12. Granger CWJ, Andersen AP (1978) Introduction to bilinear time series models. Vandenhoeck and Ruprecht, Gottingen
13. Hagan MT, Menhaj M (1994) Training feed-forward networks with the Marquardt algorithm. IEEE Trans Neural Netw 5:989–993
14. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Upper Saddle River
15. Hervai S, Osborn DR, Birchenhall CR (2004) Linear versus neural network forecast for European industrial production series. Int J Forecast 20:435–446
16. Jha GK, Thulasiraman P, Thulasiram RK (2009) PSO based neural network for time series forecasting. In: Proceedings of the international joint conference on neural networks. Atlanta, USA, 1422–1427
17. Kohzadi N, Boyd MS, Kermanshahi B, Kaastra I (1995) A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing 10:169–181
18. McLeod AI, Li WK (1983) Diagnostic checking ARMA time series models using squared residual autocorrelations. J Time Ser Anal 4:269–273
19. Rojas I, Valenzuela O, Rojas F, Guillen A, Herrera LJ, Pomares H, Marquez L, Pasadas M (2008) Soft-computing techniques and ARMA model for time series prediction. Neurocomputing 71:519–537
20. Sharda R, Patil RB (1992) Connectionist approach to time series prediction: an empirical test. J Intell Manuf 3:317–323
21. Swanson NR, White H (1997) Forecasting economic time series using adaptive versus non-adaptive and linear versus nonlinear economic models. Int J Forecast 13:439–461
22. Tong H, Lim KS (1980) Threshold autoregressive, limit cycles and cyclical data. J R Stat Soc Ser B Methodol 42:245–292
23. Zhang G, Qi Min (2005) Neural network forecasting for seasonal and trend time series. Eur J Oper Res 160:501–514
24. Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14:35–62
25. Zou P, Yang J, Fu J, Liu G, Li D (2010) Artificial neural network and time series models for predicting soil salt and water content. Agric Water Manag 97:2009–2019