RESEARCH ARTICLE Bhartiya Krishi Anusandhan Patrika, Volume 39 Issue 3-4: 245-253 (September and December 2024)

Performance Comparison of Time Delay Neural Network and Support Vector Regression for Forecasting of Jute Prices in Coochbehar District, West Bengal

Chowa Ram Sahu[1], Satyananda Basak[1]

10.18805/BKAP737

ABSTRACT

Background: In recent times, Machine Learning approaches have gained significant traction in modelling non-linearity features in the field of time series forecasting. In the present investigation, the nonstationary, nonlinearity and non-normality features of the jute price series at the three different commodity markets are dealt using the Machine Learning models.

Methods: An attempt has been made to explore efficient Machine Learning (ML) techniques *e.g.*, Time-Delay Neural Network (TDNN) and Support Vector Regression (SVR) for modelling weekly jute prices in different markets of the Coochbehar district (West Bengal). The nonlinearity pattern of the price series is tested by the Brock-Dechert-Scheinkman (BDS) test.

Result: The results of present study show that nonlinearity is present in the jute price series. Accordingly, the TDNN and SVR models have been applied for modelling and forecasting the nonlinearity of the jute prices. Finally, the TDNN(1:1s:1*l*), TDNN(10:1s:1*l*) and TDNN (2:1s:1*l*) models are outperformed SVR models in the Coochbehar, Baxirhat and Tufanganj markets, respectively in terms of minimum RMSE (157.61, 176.32 and 136.03), MAE (94.11, 95.67 and 90.45) and MAPE (1.51, 1.54 and 1.41) criteria. Hence, the TDNN model is regarded as the best optimal model for forecasting the jute prices in the Coochbehar district.

Key words: Forecasting, Jute prices, Machine learning, Nonlinearity, SVR, TDNN.

INTRODUCTION

Agricultural commodity prices play an important role in the whole national economy of India. Commodity price projections and forecasts are critical for market participants making production and marketing strategies as well as for policy makers managing commodity programs (Das *et al.,* 2022) and assessing the market impact of national or international events. Jute (*Corchorus capsularis* L.), also referred to as the "Golden Fiber" is an important traditional cash crop in India. Among the states, West Bengal ranks first in area and production of jute in the country with a total area of 0.52 million hectares (75.67 per cent) and a total production of 8.36 million bales (81.00 per cent) having 2900 kg/hectare productivity during the year 2021-22 (Fourth Advance Estimates), followed by Assam (0.91 million bales production) and Bihar (0.82 million bales production) (Directorate of Economics and Statistics, DA and FW). Jute is one of the second most important natural fiber in terms of global consumption after cotton. India is the world's largest producer of raw jute with a domestic consumption of 8.2 million bales (Jute Corporation of India, Kolkata), accounting for more than 75% of total jute production of 10.32 million bales during the year 2021-22 (Fourth Advance Estimates) (Directorate of Economics and Statistics, DA and FW). In terms of area, production and quality of jute fibre produced, the importance of Coochbehar district is next to Murshidabad and Nadia districts. So, there is a need to forecast the price of jute using suitable statistical models. The real-world price data of agro-products and its underlying market changes are often nonlinear in nature and therefore, linear models may not be suitable when market changes frequently (Kumar *et al.,* 2022). In recent times, Machine Learning (ML) techniques have gained significant traction in modelling non-linear patterns and generating non-linear forecasts in the arena of timeseries forecasting. In the present investigation, an attempt has been made to explore efficient ML algorithms *e.g.,* Artificial Neural Network (ANN) and Support Vector Regression (SVR) for forecasting Jute prices in the three markets of Coochbehar district, West Bengal, India. In the present investigation, an attempt has been made to develop the optimal machine learning model for forecasting of jute prices in Coochbehar district of West Bengal. The ANN and SVR models have been successfully applied by

[1]Department of Agricultural Statistics, Uttar Banga Krishi Viswavidyalaya, Coochbehar-736 165, West Bengal, India.

Corresponding Author: Chowa Ram Sahu, Department of Agricultural Statistics, Uttar Banga Krishi Viswavidyalaya, Coochbehar-736 165, West Bengal, India.
Email: chowasahu100@gmail.com

How to cite this article: Sahu, C.R and Basak, S. (2024). Performance Comparison of Time Delay Neural Network and Support Vector Regression for Forecasting of Jute Prices in Coochbehar District, West Bengal. Bhartiya Krishi Anusandhan Patrika. 39(3-4): 245-253. doi: 10.18805/BKAP737.

Submitted: 06-05-2024 **Accepted:** 25-10-2024 **Online:** 27-12-2024

Kohzadi *et al.* (1996); Jha and Sinha (2012); Amin (2020); Kumar *et al.* (2022); Alparslan and Uçar (2023) and Bouteska *et al.* (2023) for forecasting of different commodity prices. Several comparative studies by Paul *et al.* (2022); Sharifzadeh *et al.* (2019); Madhu *et al.* (2021) and Msiza *et al.* (2007) *etc.* have established the superiority of artificial neural network for forecasting over different Machine Learning (ML) techniques.

MATERIALS AND METHODS

Description of data

In order to carry out our analysis, historical weekly jute prices data for Coochbehar market, Baxirhat market and Tufanganj market of Coochbehar district have been taken from the Agricultural Marketing Information Network (https://agmarknet.gov.in) portal. The data for the Coochbehar market has been collected from January, 2009 to December, 2022 (672 weeks) and for Baxirhat and Tufanganj markets from January, 2010 to December, 2022 (624 weeks). The first 80% observations are used for the model building purpose and the rest 20% observations are used for model validation. In the present study, statistical analyses have been carried out using the powerful software "RStudio" (https://www.rstudio.com).

Non-linearity test

In the present study, we have applied a nonlinearity test to determine the existence of nonlinear dynamics given by Brock *et al.* (1996). The BDS test is a non-parametric test having the null hypothesis (H_0): Data is independently and identically distributed (*i.i.d.*). *i.e.*, data is linear against the alternative hypothesis (H_A): Data is not independently and identically distributed indicating presence of nonlinearity in the data.

Time- delay neural network (TDNN)

Artificial neural network (ANN) is a nonlinear data driven self-adaptive approach and is powerful tools for performing nonlinear modelling without a priori knowledge about the relationships between input and output variables. Artificial Neural Network is an information processing system which is inspired by the models of biological neural network (Sivanandam *et al.,* 2008). It is an adaptive system that changes its structure or internal information that flows through the network during the training time (Sheela and Deepa, 2013). One of the most dominant advantages of ANN models over other nonlinear statistical models is that ANN is universal approximator that is able to approximate a large class of functions with a high degree of accuracy (Zhang and Qi, 2005). There are several types of neural networks: Feed forward NN, Radial Basis Function (RBF) NNand Recurrent neural network. A single hidden layer feedforward ANN with one output node is most prominent network used in time series modelling and forecasting (Zhang *et al.,* 1998; Zhang, 2003; Anjoy *et al.,* 2017; Ray *et al.,* 2023). Time-Delay Neural Network (TDNN) has gained significant traction in modelling non-linear patterns and generating non-linear forecasts in the arena of time series forecasting (Coban and Tezcan, 2022; Zhong *et al.,* 2022). The major advantage of this model is that it does not require any presumption about the considered time series data; rather, the pattern of the data is quite important to it, usually referred as the data-driven approach. The structure between output (Y_t) and the inputs ($Y_{t-1}Y_{t-2},...,Y_{t-p}$) by multilayer feed-forward time-delay neural network ($p \times q \times 1$) is given with the following mathematical representation:

$$Y_t = \phi_0 + \sum_{j=1}^{q} \phi_j g\left(\theta_{0j} + \sum_{i=1}^{p} \theta_{ij} y_{t-i}\right) + \varepsilon_t$$

Where,

$\phi_j (j = 0,1,2,...,q)$= Weights between hidden layer and output layer.

$\theta_{ij} (i = 0,1,2,...,p\ ;\ j = 1,2,...,q)$= Weights connect input layer and hidden layer.

ϕ_0 and θ_{0j}= Bias terms.

p= Number of input nodes.

q= Number of hidden nodes.

ε_t= White noise.

g(.)= Hidden layer activation function.

A graphical presentation of Time-delay neural network (TDNN) with one hidden layer is given in Fig 1. For a

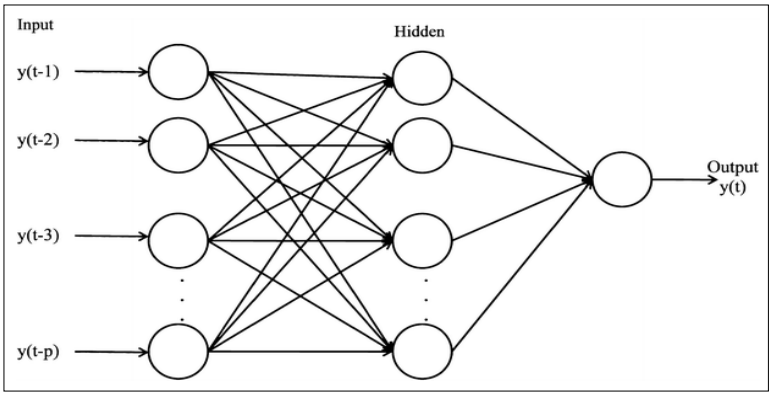


Fig 1: Time-delay neural network (TDNN) with one hidden layer (Jha and Sinha, 2012).

univariate time series forecasting problem, lagged values of the time series can be used as inputs to a neural network. Each node of the hidden layer receives the weighted sum of all the inputs including a bias term for which the value of input variable will always take a weight one. This weighted sum of input variables is then transformed by each hidden node using the activation function g(.) which is a non-linear function. In a similar manner, the output node also receives the weighted sum of the output of all the hidden nodes and produces an output by transforming the weighted sum its activation function *f*(.).

The activation function (transfer function) determines the relationship between inputs and outputs of a node and a network. In general, the activation function introduces a degree of nonlinearity that is valuable for most ANN applications (Zhang *et al.,* 1998). In time series analysis, hidden layer activation function is often chosen as the Logistic (Sigmoid) function and output nodes, as an Identity function (Jha and Sinha, 2012). Sigmoid or Logistic activation function for hidden layer is written as an equation (Kumar *et al.,* 2020; Priyadarshi *et al.,* 2023):

$$s(y)= \frac{1}{1+e^{-y}};\text{range } (0,1)$$

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. 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. Hence, a fashionable approach is trial and error until obtaining the appropriate parameters (Kaastra and Boyd, 1996; Zhang, 1998).

Support vector regression (SVR)

Support vector machine (SVM) is a machine learning algorithm introduced by Vapnik (1995) for classification problems. It was originally developed for pattern recognition thereafter promoted to support vector regression (Vapnik *et al*., 1997) for regression problems by incorporating $\in$-insensitive loss function to penalize data when they are greater than . The support vector regression (SVR) is a method that can handle overfitting so that it produces good performance in regression and prediction of time series (Setiawan *et al.,* 2021). The SVR technique provides a nonlinear mapping function to map the training dataset into a high dimensional feature space (Yeh *et al.,* 2011).

Considering a data set of *N* elements;

$$D = \{(X_i, Y_i)\}_{i=1}^{N}$$

Where

$X_i \in R^n$ input vector, $Y_i \in R$ is scalar output corresponding to x_i.

The general formula for linear support vector regression is given as:

$$f(x) = W^T \varphi(x) + b$$

Where,

w= Weight vector N dimension,

$\varphi(x)$= Function that maps to the feature space with dimensions.

b= Biased.

The solution of W and b in above equation can be obtained by solving the following minimization problem (Sermpinis *et al.,* 2014):

$$\underset{w,\, b,\, \xi^*,\, \xi}{\text{Min}}\ R_{\varepsilon}(w,\xi,\xi^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

With the constrains:

$$y_i - w^T\varphi(x_i) - b \le \varepsilon + \xi_i; \quad i = 1,2,...,N$$

$$w^T\varphi(x_i) + b - y_i \le \varepsilon + \xi_i^*; \quad i = 1,2,...,N$$

$$\xi_i, \xi_i^* \ge 0; \quad i = 1,2,...,N$$

Where,

y_i= Actual value of *i* period.

$\varphi(x_i)$= Estimated value of period.

This regularized risk function minimizes both empirical error and regularized term simultaneously and implements structural risk minimization (SRM) principle to avoid under and over fitting of training data. The first term $\|w\|^2$ in the objective function, employing the concept of maximizing the distance of two separated training data. It is used to regularize weight sizes to penalize large weightsand to maintain regression function flatness. The second term.

$$C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

Penalizes training errors of *f* (*x*) and *y* by using the ε-insensitive loss function. C > 0 is a constant known as the penalty parameter determines how much the error deviation is from the tolerable limit ε. Training errors above ε are denoted as ξ_i , whereas training errors below are denoted as ξ_i^* The SVR illustration is given in Fig 2.

The formula above is a Convex Linear Programming. NLP Optimization Problem which functions to minimize the quadratic function to be converted into a constraint. This limitation can be solved by using the Lagrange Multiplier function. The process of deriving formulas is very long and complicated. After going through mathematical stages, a new equation is obtained with the function (Muthiah *et al*., 2021):

$$f(x) = \sum_{i=1}(\alpha_i - \alpha_i^*) \cdot K(x_i, x_j) + b$$

Where,

α_i^*, α_i = Lagrangian multipliers.

x_i= Support vector.

x_j= Test vector.

The above functions can be used to solve linear problems. The function $K(x_i, x_j)$ is called the kernel function and the value of the kernel equals the inner product of two vectors, x_i and x_j, in the feature space respectively; that is:

$$K(X_i, X_j) = \varphi(X_i)^T \varphi(X_j)$$

The radial basis functions (RBF) kernel function is often used for forecasting (Lin *et al*., 2007; Cho, 2024). A nonlinear RBF kernel function defined as follows (Ghanbari and Goldani, 2022):

$$K(X_i, X_j) = \exp(-\gamma \| X_i - X_j \|^2)$$

Where,

γ = RBF width parameter.

The value of this parameter with the trade-off between the minimum fitting error and estimated function is determined. The SVR model with RBF contains three tuning parameters in the K (x_i, x_j) : 1. The parameter of loss function ε 2. Penalty factor C and 3. The parameter of kernel function γ.

2.2.5 Accuracy measures =

Root mean square error (RMSE) =

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

Mean absolute error (MAE)

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

Mean absolute percentage error (MAPE)

$$\frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Where,

y_i and $\hat{y}_i$ = Actual value and predicted value of response variable.

RESULTS AND DISCUSSION

This section presents the details of modelling and forecasting of jute price data for three different markets of Coochbehar district of West Bengal, employing the TDNN as well as SVR models. In time series analysis, the first and foremost step is to plot the data. Here, we have plotted the price series data of jute for all of the three markets showing structural breaks and its confidence intervals (Fig 3). Table 1 provides a summary statistic of the data series, where the Coochbehar market has experienced a lowest minimum price of Rs. 1569.44 per quintal and the Baxirhat market has experienced a highest maximum price of Rs.

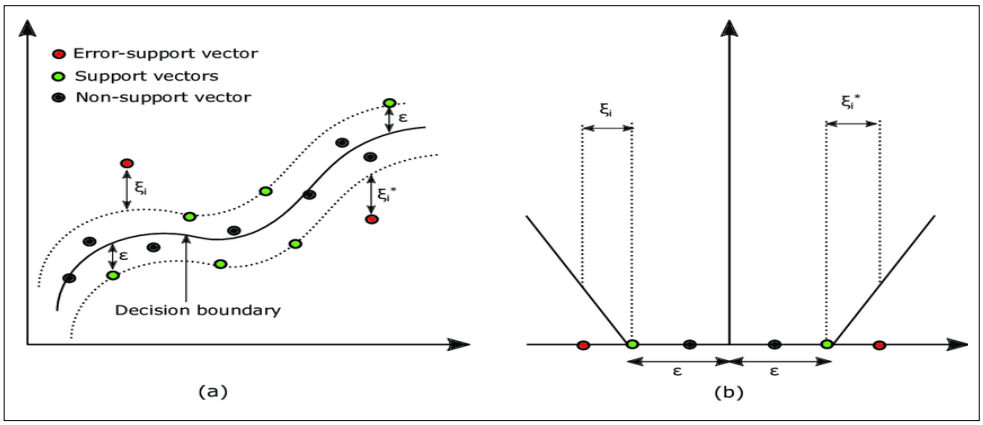


Fig 2: Schematic to illustrate (a) the conceptual structure of the support vector regression, (b) the loss function. (https://doi.org/10.3390/rs13193794).

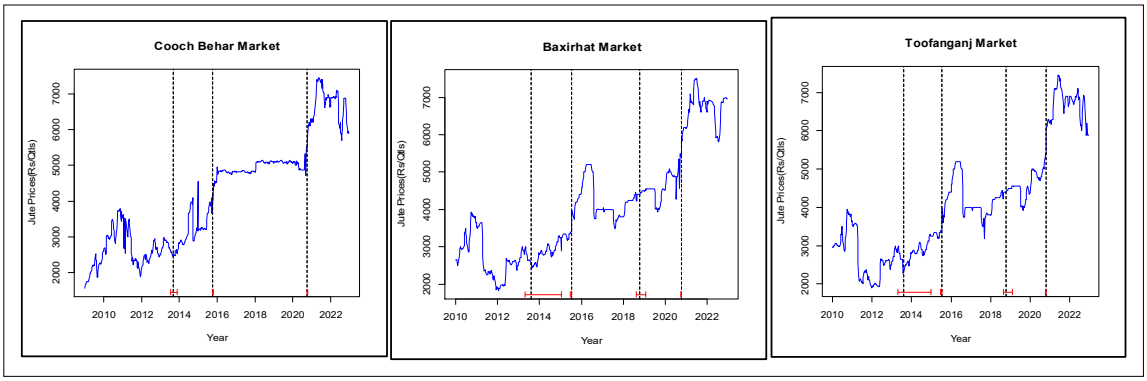


Fig 3: Weekly jute price series for all the market, including all structural breaks with confidence intervals.

7514.86 per quintal. The district market of Coochbehar has received maximum mean and median prices of Rs. 4187.65 per quintal and Rs.4648.44 per quintal respectively. The higher variability (coefficient of variation) in jute price series is observed in Coochbehar market of 36.26%, indicating presence of higher volatility as compared to the subdivisional markets *viz* Tufanganj and Baxirhat. In addition, skewness and kurtosis statistics for all the markets show that the price series are not normally distributed.

Non-linearity test

The first and most important step prior to modelling the TDNN and SVR models is to check the linearity of the data. Therefore, this study used the BDS non-linearity test, using tolerance distances ε of 0.5, 1.0, 1.5 and 2-times of standard deviations and embedding dimensions m of 2, 3 and 4 to determine whether the data under consideration has a nonlinearity pattern. The results of the BDS test for all the markets are shown in Tables 2. It is found that BDS statistic values are highly significant (p<2.2e-16) for all the markets. Hence, the table reveals a strong rejection of the null hypothesis that the price series is linearly dependent *i.e.*, all the markets have a nonlinear trend. This indicates that the traditional model cannot be used to predict the data since it is incapable of describing the nonlinear temporal pattern. However, TDNN (Jha and Sinha, 2012; Kumar *et al.,* 2022) and SVR (Alparslan and Uçar, 2023) may be used for better price forecasting of jute because of their data-driven modelling techniques.

Time delay neural network (TDNN)

To establish TDNN and SVR models, the data series is divided into two sets: the training set consisting of 80% observations and the testing set consisting of 20% observations. First, the models are fitted using the training data setand thereafter, they are used to predict over the validation period. In this study we have selected the time delay neural network (TDNN) model with single hidden layer. The optimal number of lagged observations (p) and number of hidden nodes (q) are critical in TDNN. Although there are no documented theories for determining the appropriate number of lagged observations and number of hidden nodes. Hence, in order to determine the optimal values, it is important to carry out trials. We have used multiple starts with different random starting points. The range of input lags from 1 to 10 and the range of hidden nodes from 1 to 6 have been explored to determine the optimal TDNN, *i.e.*, total of 60 neural network structures

Table 1: Summary statistics of jute price data of Coochbehar district.

Summary statistics	Markets		
	Coochbehar	Baxirhat	Tufanganj
Minimum	1569.44	1836.47	1900
Maximum	7448.30	7514.86	7454.86
Mean	4187.65	4102.24	4086.36
Median	4648.44	4000	4000
Standard deviation	1518.44	1452.80	1440.16
CV (%)	36.26	35.42	35.24
Skewness	0.29	0.66	0.65
Kurtosis	2.07	2.55	2.57

Table 2: Brock- Dechert-Scheinkman (BDS) test for nonlinearity.

Dimensions	Epsilon	Markets					
		Coochbehar		Baxirhat		Tufanganj	
		Test statistic	p-value	Test statistic	p-value	Test statistic	p-value
m=2	eps[1]	356.80	< 2.2e-16	318.3553	< 2.2e-16	345.509	< 2.2e-16
m=3	eps[1]	609.50	< 2.2e-16	627.1346	< 2.2e-16	683.917	< 2.2e-16
m=4	eps[1]	1120.28	< 2.2e-16	1361.716	< 2.2e-16	1501.912	< 2.2e-16
m=2	eps[2]	583.26	< 2.2e-16	258.6214	< 2.2e-16	238.6388	< 2.2e-16
m=3	eps[2]	782.95	< 2.2e-16	342.0036	< 2.2e-16	315.5326	< 2.2e-16
m=4	eps[2]	1086.82	< 2.2e-16	469.1708	< 2.2e-16	433.3449	< 2.2e-16
m=2	eps[3]	144.59	< 2.2e-16	115.361	< 2.2e-16	112.2714	< 2.2e-16
m=3	eps[3]	165.44	< 2.2e-16	126.0395	< 2.2e-16	122.1834	< 2.2e-16
m=4	eps[3]	193.29	< 2.2e-16	139.3052	< 2.2e-16	134.5856	< 2.2e-16
m=2	eps[4]	82.20	< 2.2e-16	77.652	< 2.2e-16	77.1845	< 2.2e-16
m=3	eps[4]	82.60	< 2.2e-16	76.1226	< 2.2e-16	75.7223	< 2.2e-16
m=4	eps[4]	83.44	< 2.2e-16	74.8992	< 2.2e-16	74.5852	< 2.2e-16

are tried for each of the market. The sigmoid activation function in the hidden nodes and the identity function in the output nodes have been used in the process of training TDNN. The TDNN model with one hidden layer is represented as *I*: *Hs*: *OI*, where I is the number of nodes in the input layer, *H* is the number of nodes in the hidden layer, *O* is the number of nodes in the output layer, *s* denotes the sigmoid (logistic) transfer function and L indicates the linear transfer function. Out of a total 60 neural network structures, a TDNN model for the Coochbehar market with one input node and one hidden node (1:1s:1*l*) identified better than other competing models based on the smallest values of RMSE, MAE and MAPE in respect of the training data set. Likewise, the best TDNN for Baxirhat market is (10:1s:1*l*) and for Tufanganj market is (2:1s:1*l*) (Table 3).

Support vector regression (SVR)

We have also employed the support vector regression model for all of the three markets and optimized the results by choosing radial basis kernel function. Tuning of the model is very much important to get better prediction through optimized parameters. Therefore, The SVR model with the radial basis kernel functions are needed to determine the three parameters *i.e.*, C, ε and the kernel

Table 3: TDNN models for jute price data of Coochbehar district.

Markets	Model	Training set			
		No. of parameters	RMSE	MAE	MAPE
Coochbehar	(1:1s:1*l*)	4	151.49	59.86	1.94
Baxirhat	(10:1s:1*l*)	13	84.78	45.17	1.39
Tufanganj	(2:1s:1*l*)	5	93.49	47.30	1.47

Table 4: Support vector regression for jute price data of Coochbehar district.

Markets	Cost (C)	Gamma (γ)	Epsilon (ε)	Training set			
				No. of support vectors	RMSE	MAE	MAPE
Coochbehar	25	5	0.1	138	144.71	114.09	3.11
Baxirhat	55	2	0.1	124	117.17	95.99	2.56
Tufanganj	5	3	0.1	124	113.97	96.34	2.61

Table 5: Comparison of prediction performance.

Markets	TDNN testing set			SVR testing set		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Coochbehar	157.61	94.11	1.51	160.56	134.82	3.71
Baxirhat	176.32	95.67	1.54	170.34	123.81	3.29
Tufanganj	136.03	90.45	1.41	172.67	127.44	3.30

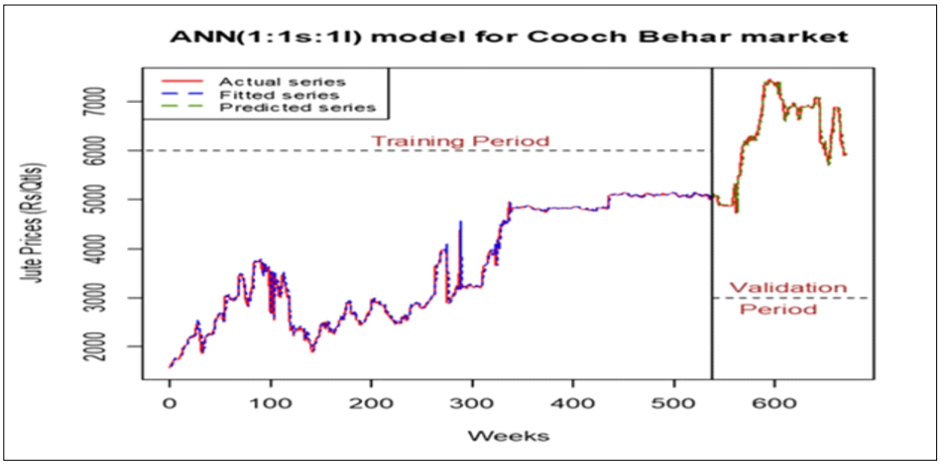


Fig 4: Plot of TDNN(1:1s:1*l*) model with training and validation periods.

parameter γ. In this study, these parameters are determined *via* trial-and-error methodfor all the markets. To overcome the problem of overfitting 10-fold cross-validation is also done. For this purpose, the training set is divided into 10 distinct subsets using 10-fold cross validation. Then, during the entire training process, each subset is used for training and the rest nine are used for validation. To obtain the optimal values of *C*, the cross-validation task was performed by varying the values of *C* from 5 to 100 with a stepping factor of 5. We tried with different settings of hyperparameter γ, from 1 to 10 with a stepping factor of 1. The value of ε is varied from 0.01 to 0.1 with a stepping factor of 0.01 and from 0.1 to 1 with a stepping factor of 0.1. Table 4 displays the optimized parameters with the smallest RMSE, MAE and MAPE for SVR model after sufficient tuning of SVR model and these parameters have been utilized to build the SVR model. From the table, it is found that the optimum values of C, γ and ε for Coochbehar market are 25, 5 and 0.1, respectively, for the Baxirhat market are 55, 2 and 0.1, respectively and for the Tufanganj market are 5, 3 and 0.1, respectively.

The evaluation of forecasting performance has been done for the test set as an out of-sample period of 20% observations for all the markets. Table 5 represents the results of the models based on the three different accuracy performance measures: MAE, RMSE and MAPE. Comparison of the validation results of the TDNN and SVR models from Table 5 indicates that both models are likely to perform well in the forecasting phase. However, the TDNN model produces the lowest MAE, RMSE and MAPE for all the markets and hence, is the most accurate one as compared to SVR model as predictions indicate that there are narrow differences between the actual and predicted values of jute prices (Fig 4-6). Hence to forecast the weekly Hence to forecast the weekly jute prices TDNN models (1:1s:1*l*), (2:1s:1*l*) and (10:1s:1*l*) are found to be the most accurate model for the Coochbehar, Tufanganj and Baxirhat markets respectively.

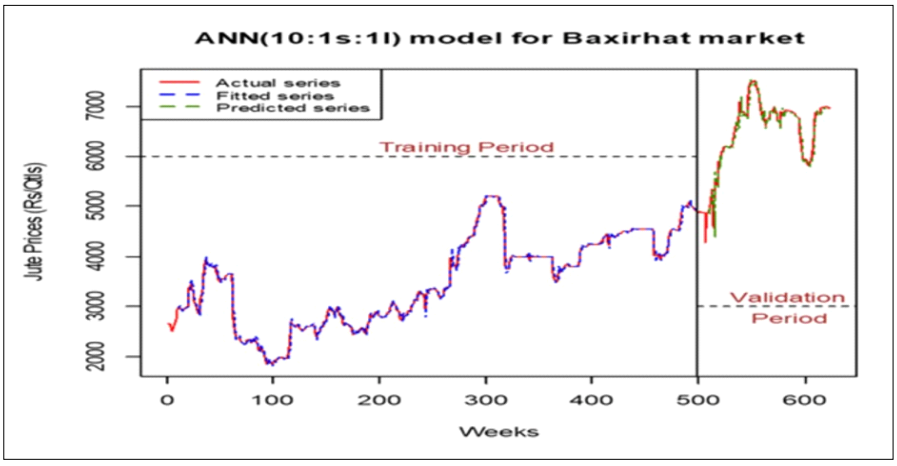


Fig 5: Plot of TDNN(10:1s:1*l*) model with training and validation periods.

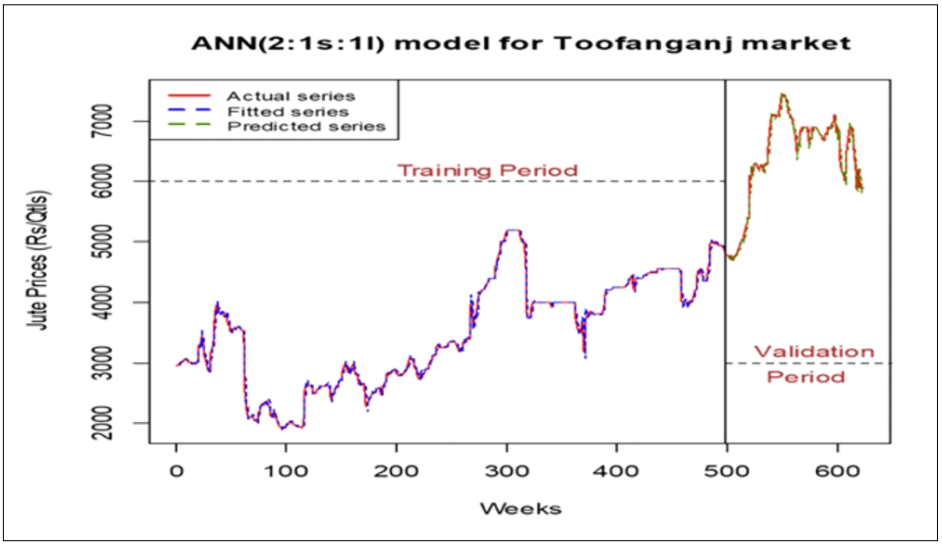


Fig 6: Plot of TDNN(2:1s:1*l*) model with training and validation periods.

CONCLUSION AND IMPLICATIONS

The aim of this paper has been to introduce an appropriate model for modelling and forecasting the nonlinearity pattern of weekly jute prices in the Coochbehar, Baxirhat and Tufanganj markets of Coochbehar district, West Bengal. For this purpose, BDS test is employed to detect the nonlinearity pattern in the jute price series for all the markets. The presence of nonlinearity for the jute prices indicates that it would be better to develop and employ the TDNN and SVR models. A comparative study has been made between the forecasting performances of the TDNN and SVR models based on the minimum RMSE, MAE and MAPE criteria. The results of this investigation indicate that the TDNN model out performs the SVR model to forecast the jute prices for all the markets. The information obtained from this study can be utilized for price forecasting and agriculture planning with regard to the jute crop of Coochbehar district.

Conflict of interest

The authors declare that they have no conflict of interest.

REFERENCES

Alparslan, S. and Uçar, T. (2023). Comparison of commodity prices by using machine learning models in the COVID-19 era. *T*urkish Journal of Engineering. 7(4): 358-368.

Amin (2020). Predicting Price of Daily Commodities using Machine Learning. International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT). doi: 10.1109/3ICT51146.2020.9312012.

Anjoy, P. Paul, R.and Sinha, K., Paul, A.and Ray, M. (2017). A hybrid wavelet based neural networks model for predicting monthly WPI of pulses in India. Indian Journal of Agricultural Sciences. 87(6): 834-839. https://doi.org/10.56093/ijas.v87i6.71022.

Bouteska, A., Hájek, P., Fisher, B.and Abedin, M. (2023). Nonlinearity in forecasting energy commodity prices: Evidence from a focused Time-delayed neural network. Research in International Business and Finance. 64: 1-15. https://doi.org/10.1016/j.ribaf.2022.101863.

Brock, W., Dechert, W., Scheinkman, J.and LeBaron, B. (1996). A test for independence based on the correlation dimension. Econometric Reviews. 15(3): 197-235. https://doi.org/10.1080/07474939608800353.doi:10.18805.

Coban, M. and Tezcan, S.S. (2022). Feed-forward neural networks training with hybrid taguchi vortex search algorithm for transmission line fault classification. Mathematics. 10:3263. doi: https://doi.org/10.3390/math10183263.

Das, P., Jha, G. K.and Lama, A. (2022). “EMD-SVR” hybrid machine learning model and its application in agricultural price forecasting. Bhartiya Krishi Anusandhan Patrika. 37(1): 1-7. doi: 10.18805/BKAP385.

Ghanbari, M. and Goldani, M. (2022). Support vector regression parameters optimization using golden sine algorithm and its application in stock market. Advances in Mathematical Finance and Applications. 7(2): 477-487. doi: 10.22034/AMFA.2021.1936352.1623.

Jha, G. and Sinha, K. (2012). 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. doi: 10.1007/s00521-012-1264-z.

Kaastra, I., Boyd, M.S., (1995). Forecasting futures trading volume using neural networks. Journal of Futures Markets. 15(8): 953-970.

Kohzadi, N., Boyd, M.S., Kermanshahi, B.and Kaastra, I. (1996). A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing. 10(2): 169-181. https://doi.org/10.1016/0925-2312(95) 00020-8.

Kumar, A., Babu, B. M., Satishkumar, U.and Reddy, G. V. (2020). Comparative study between wavelet artificial neural network (WANN) and artificial neural network (ANN) modelsfor groundwater level forecasting. Indian Journal of Agricultural Research. 54(1): 27-34. doi: 10.18805/IJARe.A-5079.

Kumar, S., Kashish, A., Singh, P., Gupta, A., Sharma, I. and Vatta, K. (2022). Performance comparison of ARIMA and Time Delay Neural Network for forecasting of potato prices in India. Agricultural Economics Research Review. 35 (2): 119-134. doi: 10.5958/0974-0279.2022.00035.0.

Lin, K., Lin, Q., Zhou, C., Yao, J. (2007). Time Series Prediction Based on Linear Regression and SVR*. Third International Conference on Natural Computation, Haikou, China.

Madhu, B., Rahman, Md. A., Mukherjee, A., Islam, Md. Z., Roy, R. and Ali, L.E. (2021). A comparative study of support vector machine and artificial neural network for option price prediction. Journal of Computer and Communications. 9:78-91.https://doi.org/10.4236/jcc.2021.95006.

Msiza, I.S., Nelwamondo, F.V. and Marwala, T. (2007). Artificial Neural Networks and Support Vector Machines for Water Demand Time Series Forecasting. IEEE International Conference on Systems, Man and Cybernetics. doi: 10.1109/ICSMC.2007.4413591.

Muthiah, H., Sa’adah, U. and Efendi, A. (2021). Support Vector Regression (SVR) Model for Seasonal Time Series Data. Proceedings of the Second Asia Pacific International Conference on Industrial Engineering and Operations Management Surakarta, Indonesia.

Paul, R.K., Yeasin, M., Kumar, P., Kumar, P., Balasubramanian, M. and Roy, H.S., *et al.* (2022). Machine learning techniques for forecasting agricultural prices: A case of brinjalin Odisha, India. PLoS ONE. 17(7): 1-17https://doi.org/10.1371/journal. pone.0270553.

Priyadarshi, M.B., Sharma, A., Chaturvedi, K.K., Bhardwaj, R., Lal, S.B., Farooqi, M. S. and Singh, M. (2023). Comparing various machine learning algorithms for sugar prediction in chickpea using near-infrared spectroscopy. Legume Research-An International Journal. 46(2): 251-256. doi: 10.18805/LR-4931.

Ray, M., Singh, K., Pal, S., Saha, A., Sinha, K. and Kumar, R. (2023). Rainfall prediction using time-delay wavelet neural network (TDWNN) model for assessing agrometeorological risk. Journal of Agrometeorology. 25(1): 151-157. doi: https://doi.org/10.54386/jam.v25i1.1895.

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.

Setiawan, I.N., Kurniawan, R., Yuniarto, B., Caraka, R.E. and Pardamean, B. (2021). Parameter optimization of support vector regression using Harris hawks optimization. Procedia Computer Science. 179: 17-24.

Sharifzadeh, M., Sikinioti-Lock, A., Shah, N. (2019). Machine-Learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regressionand Gaussian Process Regression. Renewable and Sustainable Energy Reviews. 108: 513-538. https://doi.org/10.1016/j.rser.2019.03.040.

Sheela, K. and Deepa, S.N. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Mathematical Problems in Engineering. 1-11. https://doi.org/10.1155/2013/425740.

Sivanandam, S.N., Sumathi, S. and Deepa, S.N. (2008). Introduction to Neural Networks Using Matlab 6.0, Tata McGraw Hill, 1st edition.

Vapnik, V. (1995). The nature of statistical learning theory. New York: Springer-Verlag.

Vapnik, V., Golowich, S.E. and Smola, A.J. (1997). Support vector method for function approximation, regression estimation and signal processing. Advances in Neural Information Processing Systems. 9: 281-287.

Yeh, C.Y., Huang, C.W. and Lee, S.J. (2011). A multiple-kernel support vector regression approach for stock market price forecasting.Expert Systems with Applications. 38: 2177-2186. doi: 10.1016/j.eswa.2010.08.004.

Zhang, G.P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 50: 159-175. doi: 10.1016/S0925-2312(01)00702-0.

Zhang, G.P. and Qi, M. (2005). Neural network forecasting for seasonal and trend time series. European Journal of Operational Research. 160(2): 501-514. doi: 10.1016/j.ejor.2003.08.037.

Zhang, G., Patuwo, B.E. and Hu, M.Y. (1998). Forecasting with artificial neural networks: the state of the art. International Journal of Forecasting. 14(1): 35-62. https://doi.org/10.1016/S0169-2070(97)00044-7.

Zhong, C., Lou, W. and Wang, C. (2022). Neural network-based modeling for risk evaluation and early warning for large-scale sports events. Mathematics.10: 3228. https://doi.org/10.3390/math10183228.