Statistics and Applications {ISSN 2454-7395 (online)}
Volume 21, No. 2, 2023 (New Series), pp 237–259
http://www.ssca.org.in/journal

Agricultural Price Forecasting Based on Variational Mode Decomposition and Time-Delay Neural Network

Kapil Choudhary[1], Girish K. Jha[2], Ronit Jaiswal[1], P. Venkatesh[2] and Rajender Parsad[1]

[1]*ICAR-Indian Agricultural Statistics Research Institute, PUSA, New Delhi-110012*
[2]*ICAR-Indian Agricultural Research Institute, PUSA, New Delhi-110012*

Received: 24 May 2022; Revised: 16 March 2023; Accepted: 21 March 2023

Abstract

Agricultural commodities prices are very unpredictable and complex, and thus, forecasting these prices is one of the research hotspots. In this paper, we propose a new hybrid VMD-TDNN model combining variational mode decomposition (VMD) and time-delay neural network (TDNN) to improve the accuracy of agricultural price forecasting. Specifically, the VMD decomposes a price series into a set of intrinsic mode functions (IMFs), and the obtained IMFs are modelled and forecasted separately using the TDNN models. Finally, the forecasts of all IMFs are combined to provide an ensemble output for the price series. VMD overcomes the limitation of the mode mixing and end effect problems of the empirical mode decomposition (EMD) based variants. The prediction ability of the proposed model is compared with TDNN, and EMD based variants coupled with TDNN model using international monthly price series of maize, palm oil, and soybean in terms of evaluation criteria like root mean squared error, mean absolute percentage error and, directional prediction statistics. Additionally, Diebold-Mariano test and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), a ranking system, are used to evaluate the accuracy of the models. The empirical results confirm that the proposed hybrid model is superior in terms of evaluation criteria and improves the prediction accuracy significantly.

Key words: Agricultural price forecasting; Empirical mode decomposition; Intrinsic mode function; Time-delay neural network; Variational mode decomposition.

1. Introduction

Price forecasting of agricultural commodities is a challenging task as there are several unpredictable factors, both natural and man-made, which influence the production and price of the commodities. Thus, the price series become inherently nonstationary and nonlinear in nature posing a severe threat to food security in developing countries, see FAO (2011). Accurate and reliable agricultural price forecasts are thus very necessary not only

Corresponding Author: Girish K. Jha
Email: girish.stat@gmail.com

for mitigating the threat of food security but also to balance the demand with supply, ensure remunerative prices to farmers, and the welfare of the consumers, see Jaiswal *et al.* (2022). A thorough review of the existing literature confirms that significant efforts have been done to improve price forecasting using various time series models. The various time series models developed for price forecasting can be broadly classified into two categories, *i.e.* statistical models and artificial intelligence (AI) models. Among statistical models, autoregressive integrated moving average (ARIMA), see Box *et al.* (2015), and constituent models, see Hayat and Bhatti (2013); Jadhav *et al.* (2017), are most frequently used as prediction models. However, ARIMA models assume linear relationships among data points, despite real-world agricultural price data being usually nonlinear. As a result, the ARIMA model is unable to capture the hidden patterns in the agricultural price series effectively, leading to unsatisfactory forecasting results.

In recent years, artificial neural network (ANN) in the category of AI models has become the most efficient modelling method in dealing with the complex nature of time series. ANN has been widely utilized to model nonlinear time series with minimal assumptions and high prediction accuracy due to its self-learning capabilities, see Zhang *et al.* (1998). ANN has been effectively employed as a universal function approximator in a wide range of research areas like electricity price forecasting, exchange forecasting, wind speed forecasting, solar energy forecasting, *etc.* In agricultural price forecasting, Jha and Sinha (2014)used the time-delay neural network (TDNN) model to predict monthly wholesale prices of different oilseeds and concluded that the ANN-based forecasting model outperforms the ARIMA model in terms of prediction accuracy. Similarly, Xu and Zhang (2021) investigated both univariate and bivariate neural network modelling for corn cash prices and found that simple neural networks with twenty hidden nodes and two lags provided better forecasting accuracy for short-term forecasting. Despite the better prediction performance of ANN-based models in many areas, their accuracy is still not satisfactory when dealing with nonstationary and nonlinear time series data. However, the accuracy can be further increased using the hybridization technique, *i.e.* combining different models according to their strength and producing a synergetic effect. In a hybrid model class, the decomposition-and-ensemble-methodology is the most promising one, see Qian *et al.* (2019). This methodology follows the principle of "divide and conquer" whereby using some techniques, a complex series is divided into a number of simple subseries such that each subseries now has better characterization and thus can be easily captured, resulting in better forecasting accuracy.

For the decomposition of any nonlinear and nonstationary time series, empirical mode decomposition (EMD), see Huang *et al.* (1998), and its variants like ensemble empirical mode decomposition (EEMD), see Wu and Huang (2009), and complementary ensemble empirical mode decomposition with adaptive noise (CEEMDAN), see Torres *et al.* (2011), are commonly used. The essence of the EMD and its variants is that they decompose a time series into a set of subseries (modes) called intrinsic mode functions (IMFs) and residual. These IMFs and residual are further modelled by any of the forecasting techniques like TDNN. For instance, Yu *et al.* (2008) evaluated EMD based feed-forward neural network (FNN) for crude oil predictions and concluded that decomposition-based hybrid models outperform standalone forecasting models. Choudhary *et al.* (2019) used EEMD for decomposing the daily potato price series of two different markets. Fang *et al.* (2020) applied EEMD to different agricultural commodities for decomposition, whereas ARIMA, neural network (NN) and support vector machine (SVM) models for predicting the decomposed components. Prasad *et al.* (2018) demonstrated the superiority of a hybrid model that combines CEEMDAN and an extreme learning machine (ELM) to forecast soil moisture.

However, EMD and its variants have major drawbacks, such as the frequent appearance of mode mixing, noise sensitiveness and end effects, leading to meaningless subseries that negatively impact the precision of decomposition. In order to address these limitations, variational mode decomposition (VMD) is proposed as an adaptive, non-recursive and multiresolution decomposition technique by Dragomiretskiy and Zosso (2014). VMD decomposes original time series into a set of distinct independent IMFs based on their central frequencies. The VMD proved its superiority over EMD based decomposition in the different areas of time series forecasting, see Bisoi *et al.* (2019); Dragomiretskiy and Zosso (2014); Lahmiri (2016); Liu *et al.* (2018). Therefore,in view of the superior performance of VMD as a unique data-adaptive decomposition technique and the advantageous properties of TDNN for forecasting any nonlinear series, a novel hybrid VMD-TDNN model is proposed for agricultural price forecasting.

The main idea of our study is to utilise a new adaptive multiresolution technique in the context of modelling and predicting nonstationary and nonlinear agricultural price series. However, the most significant contributions of this paper are as follows. First, a novel agricultural price forecasting framework is proposed by combining VMD with the TDNN model. VMD is a decomposition technique that breaks a highly complex agricultural price series into several uniform subseries with stable fluctuations. VMD has the advantage of being noise robust as it denoises a time series using simulated harmonic functions. In this context, for an agricultural price series that is known to be very noisy, the VMD is more suitable for its better characterization leading to faster convergence and better predictive accuracy. Second, for empirical evaluation of our proposed model, we use three real agricultural price series to test how well the proposed model can tackle high-frequency events such as fluctuations in fuel prices, strikes, *etc.* and also the low-frequency events such as lower production, higher export, optimum rainfall, *etc.* Third, it is seen that many scientists believe that machine learning is a black box, and the results obtained from any machine learning technique are either not trustworthy or are not able to provide proper interpretations otherwise. Another reason for this belief is that the datasets and the code for machine learning-related forecasting research are often not publicly available, making it difficult for the forecasting community to adapt such research and verify the claimed performance. Thus, considering one of the major goals of our study to make this work replicable by the whole research fraternity for practical forecasting tasks, we use the datasets available in the public domain and for each of the hybrid models used in this study, we develop and publish packages namely, “eemdTDNN” and “vmdTDNN”, see Choudhary *et al.* (2021, 2022), in CRAN. Fourth, we compare the prediction accuracy of the VMD-TDNN with different decomposition-based techniques, and the empirical evidence shows that the VMD-TDNN model outperforms EMD and its variants based hybrid models in terms of each evaluation criteria. Finally, for the robust validation and to check the superiority of the forecasting ability of the developed model, we use Diebold-Mariano test for checking the significant improvement achieved by it. Further, we also use Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) for ranking the models based on overall performance.

The remainder of the paper is organised as follows: Section 2 describes the proposed VMD-TDNN hybrid model for agricultural price forecasting in detail. For empirically evaluating the proposed model, three internationally traded agricultural commodities, namely maize, soybean oil and palm oil monthly price series, are described in Section 3. Section 4 concludes the work.

2. Methodology

2.1. Variational mode decomposition

VMD, see Dragomiretskiy and Zosso (2014), is a novel data-adaptive decomposition technique that overcomes the limitations of traditional frequency-based decomposition techniques. This technique effectively improves the end effect, mode mixing, recursive sifting process, sensitivity to noise, a fixed number of modes, and other shortcomings of EMD variants, see Wu and Huang (2009). The algorithm used in VMD is non-recursive as it extracts modes concurrently, assuming limited bandwidth of central frequency for each IMF. Moreover, the modes obtained after VMD have a particular property called sparsity which means each mode is mostly compact around a centre pulsation in the frequency spectrum. The advantages of using VMD over other techniques are that the modes are robust with respect to noise and have faster convergence with better accuracy. These characteristics of VMD make it highly suitable for addressing complex agricultural price data consisting of multi-frequency signals. The bandwidth of a mode is estimated using the following steps:

1. For each mode $c_j(t)$, the Hilbert transformation (HT) is used to obtain a one-sided frequency spectrum.
2. The sifting of the frequency spectrum of the mode is determined using the modulation properties.
3. The bandwidth of $c_j(t)$ is estimated finally using H^1 Gaussian smoothness of the demodulated signal *i.e.* the squared L^2-norm of the gradient.

The following constraints of the variation problem can be used to explain VMD:

$$\min_{\{\omega_j\},\{c_j\}} \left\{ \sum_j \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t} \right) * c_j(t) \right] e^{-i\omega_j t} \right\|_2^2 \right\}$$

such that $\sum_{j=1}^{n} c_j(t) = y(t)$; where $y(t)$ is the original price series, $\{c_j\} := \{c_1, c_2, \cdots c_n\}$ is the set of modes, $\{w_j\} := \{w_1, w_2, \cdots w_n\}$ is the of central frequencies, $\delta(t)$ is the impulse function, $\partial_t(.)$ is the partial derivative of time t, n is the number of modes, $*$ denotes convolution operation, $\|.\|$ denotes norm processing, and $i = \sqrt{-1}$. Lagrangian multipliers $\lambda(t)$ and quadratic penalty terms are used to transform a constraint problem into an unconstrained problem that is easy to solve:

$$\begin{aligned} L\left(\{c_j\}, \{\omega_j\}, \lambda\right) = \alpha \sum_j \left\| \partial_t \left[\left(\delta(t) + \frac{i}{\pi t} \right) c_j(t) \right] e^{-i\omega_j t} \right\|_2^2 \\ + \left\| y(t) - \sum_j c_j(t) \right\|_2^2 + \left[\lambda(t), y(t) - \sum_j c_j(t) \right] \end{aligned}$$

where α is said to be a balance parameter or penalty parameter of data fidelity constraint.

Furthermore, an iterative sequence called the alternate direction method of multipliers (ADMM) is applied to the above equation for updating c_j, ω_j and λ in two directions. The results are obtained as follows:

$$\hat{c}_j^{k+1}(\omega) = \frac{\hat{y}(\omega) - \sum_{l \neq j} \hat{c}_l(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha\left(\omega - \omega_j\right)^2}; \quad \omega_j^{k+1} = \frac{\int_0^\infty \omega \left|\hat{c}_j(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{c}_j(\omega)\right|^2 d\omega}$$

$$\text{and } \hat{\lambda}^{k+1}(\omega) \leftarrow \hat{\lambda}^k(\omega) + \tau \left[\hat{y}(\omega) - \sum_j \hat{c}_j^{k+1}(\omega)\right].$$

The stopping criterion for the iterations is $\sum_j \frac{\left\|\hat{c}_j^{k+1} - \hat{c}_j^k\right\|_2^2}{\left\|\hat{c}_j^k\right\|_2^2} < \epsilon$, where $\epsilon > 0, \hat{y}(\omega), \hat{c}_j(\omega), \hat{\lambda}(\omega)$ and $\hat{c}_j^{k+1}(\omega)$ are fourier transformations of $y(t), c_j(t), \lambda(t)$, and c_j^{k+1}(t), respectively, and k is the number of iterations.

2.2. Time-delay neural network (TDNN)

The artificial neural network (ANN) technique, which uses nonlinear units (neurons) to model any complex nonlinear time series, is being frequently utilized in many applications. There are three layers in a standard ANN architecture: input layer, where data is introduced to the network; hidden layer, where data is processed; and output layer, where the results of the given inputs are produced. There are two ways to model time series using neural networks: either using a recurrent neural network or creating short-term memory at the network's input layer, see Haykin (2009). TDNN is an example of the latter, which uses the temporal dimension of a univariate time series to develop a short-term memory, called heteroassociative memory, in its network. The usual TDNN is a feed-forward network with interconnected hidden and output neurons. A TDNN with a single hidden layer has the following generic expression, see Jha and Sinha (2014)

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

where $\hat{y}(t)$ is the predicted value, $y(t-i)$ is the i^{th} input (lag), $\alpha_j(j = 0, 1, 2, \ldots, q)$ and $\beta_{ij}(i = 0, 1, 2, \ldots, p; j = 1, 2, \ldots, q)$ are connection weights, p and q are the numbers of input and hidden nodes, respectively, f and g denote the activation functions at the hidden and output layer of the model.

2.3. VMD-TDNN model for agricultural price series

A nonstationary and nonlinear time series is decomposed into IMFs using VMD as a decomposition tool. Several models of TDNN are built for each IMF separately, varying the hyperparameters of the TDNN, and the best-fitted model is selected for each IMF to predict them separately, followed by ensemble prediction. Thus, a hybrid model, namely, VMD-TDNN, is proposed by integrating VMD and TDNN, and its details are displayed in

Figure 1. The procedure for this model can be separated into three parts:

1. **Data decomposition** – VMD decomposes agricultural price series $y(t)$ into n independent modes (IMFs), which are stationary and nonlinear. These IMFs have different oscillations of agricultural prices from high to low frequencies. These modes have a regular structure and stable fluctuation. Now, the patterns of each IMF can be captured more conveniently and accurately through TDNN.

2. **Individual prediction** – Each IMF is split into training and testing sets to ensure the generalization ability of the forecasting model. The TDNN model is used for modelling each of the IMFs as it is well suited for capturing nonlinear patterns.

3. **Ensemble prediction** - The final forecast of the original price series is obtained by adding the predicted values of all IMFs as:

$$\hat{y}(t) = \sum_{j=1}^{n} \hat{c}_j(t)$$

where, $\sum_{j=1}^{n} \hat{c}_j(t)$ represent the ensemble of predicted values of IMFs.

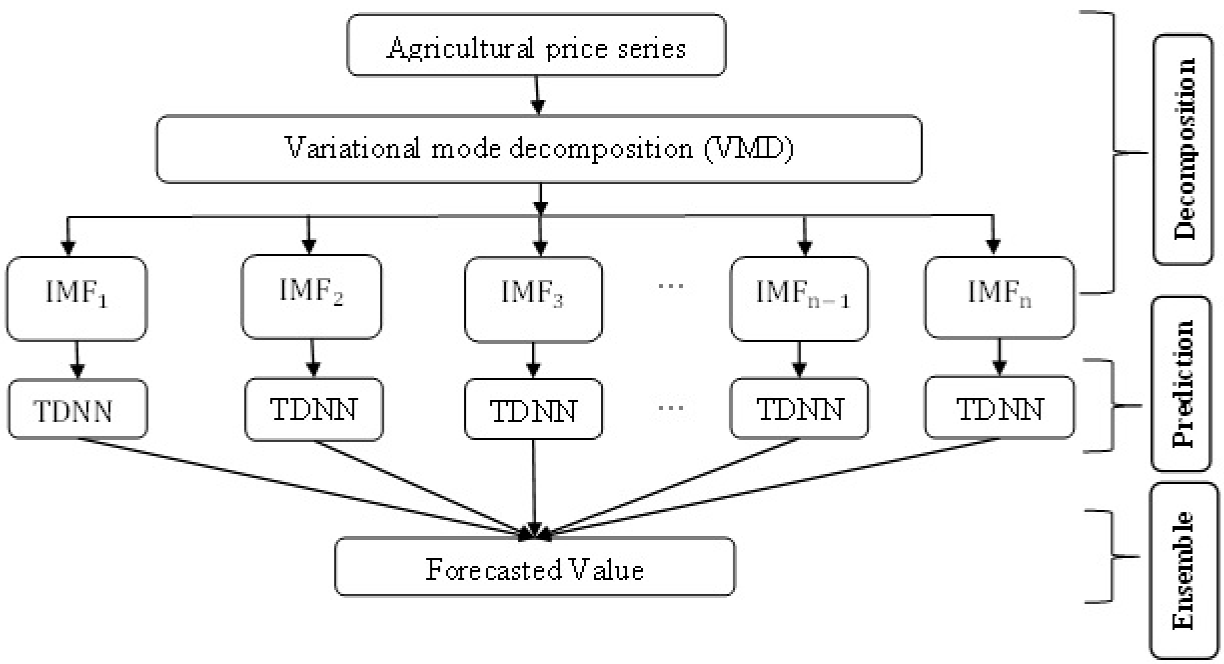

Figure 1: Flowchart of VMD-TDNN model for agricultural price forecasting

2.4. Forecasting evaluation criteria

Each prediction model employed in this paper is evaluated in terms of root mean squared error (RMSE), mean absolute percentage error (MAPE), directional prediction statistics (D_{stat}), and Diebold-Mariano (DM) test, since individual decision criteria is unable to capture errors completely, see Jaiswal *et al.* (2022). Moreover, for the ranking of each model, TOPSIS method, see Hwang and Yoon (1981), is employed, which ranks each model by giving weights after normalizing the decision matrix of all evaluation criteria and calculating the geometric distance between the different models. The following are the forecasting evaluation criteria for comparing the proposed model with other models:

1. **Root mean squared error (RMSE):**

$$RMSE=\sqrt{\frac{\sum_{t=1}^{h}(y(t)-\hat{y}(t))^{2}}{h}}$$

2. **Mean absolute percentage error (MAPE):**

$$MAPE=\frac{1}{h}\sum_{t=1}^{h}\left|\frac{y(t)-\hat{y}(t)}{y(t)}\right|$$

3. **Directional prediction statistics (D_{stat}):**

$$D_{\text{stat}}=\frac{1}{h}\sum_{t=1}^{h}a_{t}\times 100\%$$

where $y(t)$ and $\hat{y}(t)$ are the actual value and predicted value, respectively, h is the size of the testing set and $a_t=\begin{cases}1, & \text{if } [y(t+1)-y(t)][\hat{y}(t+1)-y(t)]\geq 0\\ 0, & \text{otherwise}\end{cases}$.

4. **Diebold-Mariano (DM) test:**

For a given time series $y(t)$, the Diebold-Mariano (DM) test statistics is defined as:

$$z_{DM}=\frac{\bar{d}}{\sqrt{\widehat{V}_{\bar{d}}}}$$

where h is the test size, $\{e_{tet}\}_{t=1}^{h}$ and $\{e_{ref}\}_{t=1}^{h}$ are error for test model and reference model respectively, g is the loss function, $\bar{d}=\frac{1}{h}\sum_{t=1}^{h}\left[g\left(e_{\text{tet}}\right)-g\left(e_{ref}\right)\right]$ is the sample mean, $\widehat{V}_{\bar{d}}=\frac{1}{h}\left[\gamma_{0}+2\sum_{j=1}^{l-1}\gamma_{j}\right]$ is the estimate of variance using l step forecasts and $\gamma_j=\operatorname{cov}\left(d_t,d_{t-j}\right)$ is the estimate of j^{th} autocovariance of $\left[g\left(e_{tet}\right)-g\left(e_{ref}\right)\right]$.

5. **TOPSIS:**

For a given decision matrix $\mathbf{X}=(x_{ij})$ and a weight vector $\boldsymbol{W}=[w_1,w_2,\cdots,w_e]$, rank of i^{th} model is defined as:

$$R_i=\frac{d_i^{-}}{d_i^{-}+d_i{}^{+}}$$

where x_{ij} denotes j^{th} evaluation criteria for i^{th} prediction model for $1 \leq i \leq m$ and $1 \leq j \leq e$, $d_i^+ = \sqrt{\sum_{j=1}^{e}\left(v_{ij} - v_j^+\right)^2}$ and $d_i^- = \sqrt{\sum_{j=1}^{e}\left(v_{ij} - v_j^-\right)^2}$ are measures of separation between the positive and negative ideal solutions, $v_{ij} = w_j * n_{ij}$ is weighted normalized decision matrix where $\sum_{j=1}^{e} w_j = 1, n_{ij} = \frac{x_{ij}}{\sqrt{\sum_i x_{ij}{}^2}}$ is the normalized value of x_{ij}, $v_j^+ = \begin{cases} \max v_{ij}, & \text{if } j \text{ is positive criterion} \\ \min v_{ij}, & \text{if } j \text{ is negative criterion} \end{cases}$ and $v_j^- = \begin{cases} \min v_{ij}, & \text{if } j \text{ is positive criterion} \\ \max v_{ij}, & \text{if } j \text{ is negative criterion} \end{cases}$ are extremely positive and extremely negative performance on each criterion.

3. Empirical results and discussion

Three different agricultural commodity price series are used in this section to empirically evaluate the proposed model's performance. In this study, all the model developments and their statistical analysis are done in R statistical software of version 4.1.2. The detailed R codes are given in the Appendix. In this section, data description, different decomposition techniques, and prediction results of the models are analysed for the price series.

3.1. Data description

This paper examines the efficiency of the proposed hybrid VMD-TDNN model using monthly international Maize, Palm oil, and Soybean oil price (dollar per metric tonne, $\$/MT$) data. Data are obtained from the "World Bank Commodity market" from January 1960 to December 2021(`https://www.worldbank.org/en/research/commodity-markets`). Each price series contains 744 observations divided into training and testing sets to ensure generalization capability. The training set carrying 732 data points is used to train the model, while the remaining 12 data points are used to test the effectiveness of the proposed model. Figure 2 shows time plots and the complex behaviour of each series, which is the characteristic of agricultural price data. Table 1 shows the basic descriptive statistics for each price series.

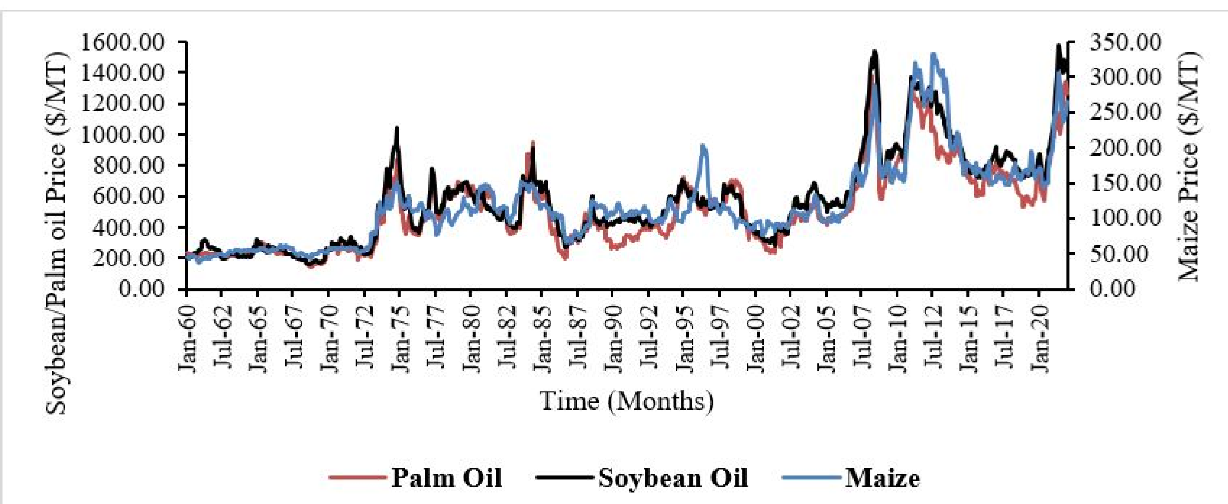

Figure 2: Time plots for monthly international Palm oil, Soybean oil and Maize price ($\$/MT$) series

Table 1: Descriptive statistics of the price ($/MT) series (from January 1960 to December 2021)

Statistics	Maize	Palm oil	Soybean oil
Mean	120.31	512.49	574.07
Maximum	333.05	1377.22	1574.67
Minimum	38.00	141.73	157.00
Standard deviation	59.65	257.75	294.08
Skewness	1.28	0.93	1.01
Kurtosis	1.80	3.58	0.82
Jarque-Bera	299.40	118.26	146.97

Since agricultural price series are complex and may exhibit nonstationarity and nonlinearity properties, it becomes necessary to test these properties, which can be helpful in skilful handling of data while fitting the model. The Augmented Dickey-Fuller (ADF) test, see Kumar *et al.* (2020), and Brock-Dechert-Scheinkman (BDS) test, see Choudhary *et al.* (2019), are used to check the stationarity and linearity characteristics, respectively. Table 2 shows the ADF test results that confirm the nonstationarity of each price series. Table 3 presents the BDS test results, which confirm the nonlinearity nature of each price series.

Table 2: Augmented Dickey-Fuller (ADF) test results

Price Series	ADF Test		Conclusion
	t-statistic	Probability	
Maize	−3.12	0.10	Nonstationary
Pam Oil	−3.14	0.09	Nonstationary
Soybean Oil	−3.01	0.15	Nonstationary

Table 3: Brock-Dechert-Scheinkman (BDS) test results

Price Series	Embedding dimension					Conclusion
	Epsilon	2		3		
		Statistics	Probability	Statistics	Probability	
Maize	0.5 σ	133.23	< 0.001	224.66	< 0.001	Nonlinear
	1.0 σ	67.16	< 0.001	77.15	< 0.001	
	1.5 σ	52.64	< 0.001	53.64	< 0.001	
	2.0 σ	41.41	< 0.001	39.72	< 0.001	
Palm oil	0.5 σ	230.22	< 0.001	399.29	< 0.001	Nonlinear
	1.0 σ	98.24	< 0.001	119.26	< 0.001	
	1.5 σ	64.78	< 0.001	67.36	< 0.001	
	2.0 σ	53.68	< 0.001	51.77	< 0.001	
Soybean oil	0.5 σ	193.52	< 0.001	335.36	< 0.001	Nonlinear
	1.0 σ	85.31	< 0.001	102.54	< 0.001	
	1.5 σ	60.46	< 0.001	62.45	< 0.001	
	2.0 σ	52.53	< 0.001	50.74	< 0.001	

Table 4: VMD parameters for different price series

Price series	α	τ	n	ϵ
Maize	2000	0	9	1×10^{-6}
Palm Oil	2000	0	9	1×10^{-7}
Soybean Oil	2000	1	9	1×10^{-6}

Table 5: Comparison of different decomposition algorithm in terms of θ

Price series	EMD	EEMD	CEEMDAN	VMD
Maize	0.1669	0.0472	0.0435	0.0214
Palm Oil	0.0420	0.0292	0.0190	0.0121
Soybean Oil	0.0608	0.0443	0.0279	0.0140

3.2. Decomposition of the agricultural price series

For the hybrid VMD-TDNN model, VMD is used to decompose the original agricultural price series into a set of IMFs. For decomposition, VMD requires four hyperparameters: (i) balancing parameter of the data-fidelity constrain(α), (ii) tolerance of convergence criterion (τ), (iii) number of modes (n), and (iv) time-step of the dual ascent (ϵ). The values of these hyperparameters are selected through experimentation in order to keep the energy evaluation parameter value (θ) as close to zero as possible to achieve superior decomposition outcomes and are presented in Table 4. While in the case of the number of modes, unlike EMD variants, a VMD technique provides as many modes as it is asked to produce, which significantly affects the accuracy of decomposition results. However, there is no practical or theoretical method to determine the optimum number of modes, see Dragomiretskiy and Zosso (2014); Lahmiri (2016). Therefore, in order to make all models comparable, the number of modes by VMD is chosen the same as that obtained by EMD and its variants. Accordingly, each price series is decomposed into nine different independent IMFs through VMD. Figure 3 shows the decomposed IMFs through VMD of the three price series from high frequency to low frequency. Here, high frequency shows the effect of short term fluctuations of the market, whereas low frequency represents any particularly significant event (like changes in policy, adverse effects of several biotic and abiotic factors, *etc.*) affecting the demand-supply equilibrium at that time. For instance, in our case, the two most significant events are observed in 2008 and 2011, which can be observed in Figure 3 in the form of spikes around 580^{th} and 620^{th} observations, respectively. Reasons behind both the events are the 2007-08's world food crisis and the production of biofuels, see Trostle (2011). For ethanol fuel production, usage of maize increased from 15% (2006) to 40% (2012) of total U.S. maize production. Moreover, the VMD based decomposed IMFs show more independent frequency distribution than the EMD variants, which can be empirically verified through the energy evaluation parameter (θ), see Zhu *et al.* (2016), defined as:

$$\theta = \frac{\left|\sqrt{\sum_{j=1}^{n} E_{j(t)}^2} - E_{y(t)}\right|}{E_{y(t)}}; \quad E_{y(t)} = \sqrt{\frac{\sum_{t=1}^{T} y^2(t)}{T}};$$

where $E_{y(t)}$ and $E_{j(t)}$ are the energy values of the original time series and j^{th}IMF, respectively. Here, θ is used as an evaluation parameter for orthogonality of IMFs such that θ closer to 0 indicates more orthogonality, whereas greater θ indicates the presence of elusive components among IMFs. Table 5 compares different decomposition methods in terms of θ for each price series, which shows that the value of θ in the case of VMD is the smallest. This motivates us to choose VMD over other techniques to construct the TDNN based hybrid model.

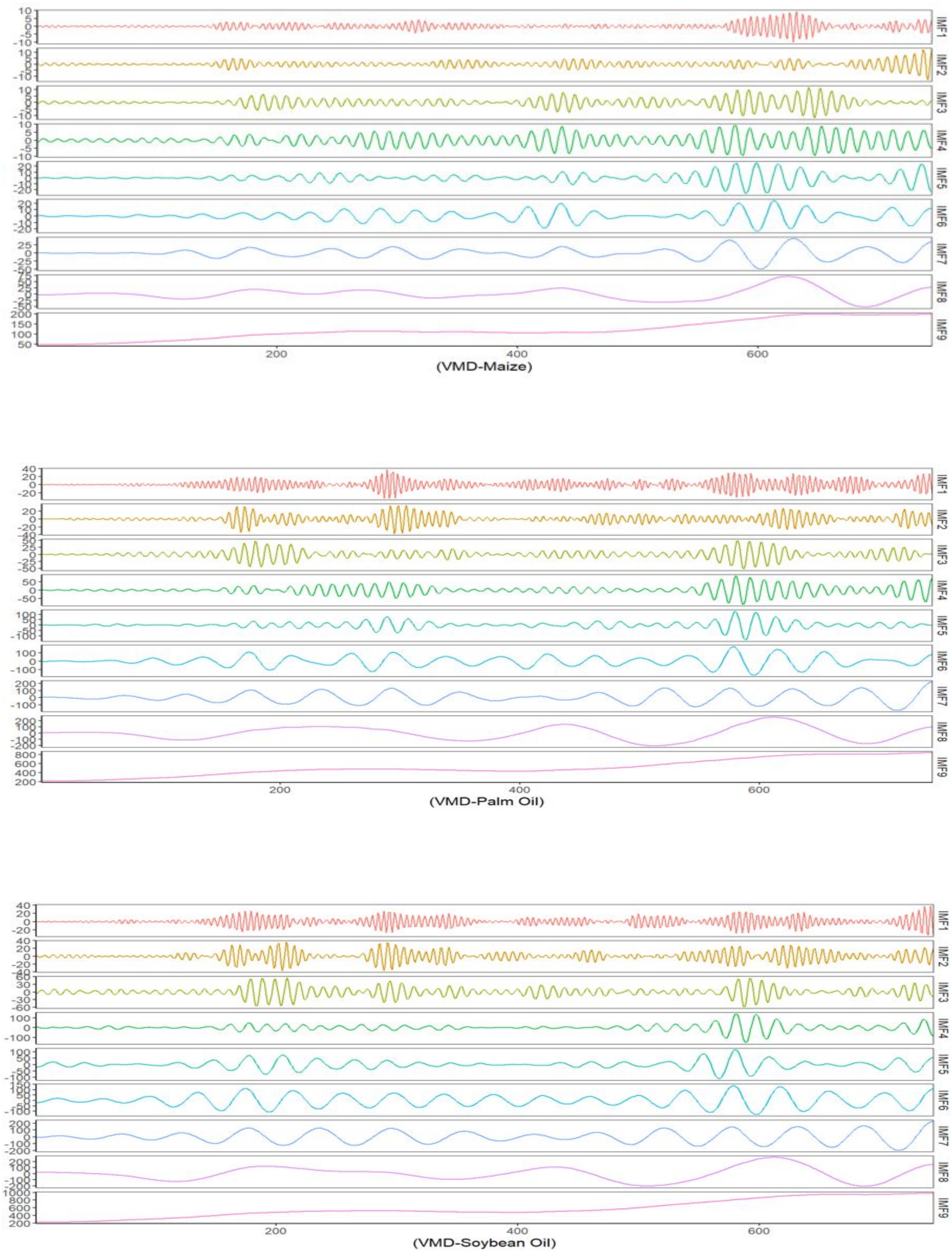

Figure 3: The decomposed IMFs for Maize, Palm oil and Soybean oil price series

3.3. Forecasting results and discussion

The datasets and code for machine learning-related forecasting studies are often not publicly available, making it impossible for the forecasting community to replicate and validate the stated performance. Thus, keeping in mind that one of the primary goals of our study is to make this work replicable by the entire research community for practical forecasting tasks, we develop two R software packages named eemdTDNN, see Choudhary *et al.* (2021) and vmdTDNN, see Choudhary *et al.* (2022), which are published in the comprehensive R archive network (CRAN). Here, the emdTDNN, EEMDTDNN, ceemdanTDNN and VMDTDNN are the functions of the above packages which are used to model and forecast each price series. The forecasting performance of the proposed VMD-TDNN model is compared with the existing individual model, *i.e.* TDNN and different hybrid models like EMD-TDNN, EEMD-TDNN, and CEEMDAN-TDNN for each price series. Figure 4 displays the plots of the predicted series by all models, along with the level series for each price series. The figure clearly shows that the VMD-TDNN model captures price movement patterns and directions better than conventional models. Moreover, the prediction ability of different models is tested in terms of different forecasting evaluation criteria. In this paper, RMSE, MAPE and directional prediction statistics (D_{Stat}) are employed to evaluate the performance of each model. Table 6 shows that all the hybrid models, including EMD-TDNN, EEMD-TDNN, CEEMDAN-TDNN and VMD-TDNN, outperform the single prediction model, *i.e.* TDNN, for each price series in terms of RMSE and MAPE.

It is mainly due to the "decomposition-ensemble principle" where decomposition techniques (EMD, EEMD, CEEMDAN, and VMD) reveal the hidden patterns of agricultural prices series and produce stationary and nonlinear modes which improve the forecasting ability of the TDNN. Among hybrid models, VMD-TDNN outperforms EMD-TDNN, EEMD-TDNN and CEEMDAN-TDNN in terms of both level and directional statistics since VMD is better than EMD variants, as discussed in section 3.2. With regards to D_{Stat} in particular, the results of the proposed VMD-TDNN model show better directional prediction than its competing models by showing 90% direction accuracy for maize series, almost 82% for palm oil, and 100% for soybean oil (Table 6). Though the different evaluation criteria used above show the superiority of the proposed model individually, there is ambiguity in choosing the best among other benchmark models as their results are not consistent. To get a better interpretation and a proper order of all models, we employ a novel technique called TOPSIS, which ranks all the models by combining their performances in both level and directional measures. Table 6 shows the ranks of each model obtained by the TOPSIS method.

Apart from these assessment criteria, the Diebold-Mariano (DM) test is also used to compare the predicting accuracy of various models statistically. Table 7 summarises the results of the DM test for each prediction model, and the following conclusions can be drawn. Firstly, the proposed VMD-TDNN model outperforms all existing models at a 5% significance level for each series. Secondly, all the hybrid models perform better than the TDNN model at the significance level of less than 1% for each series except for EMD-TDNN for palm oil which is significant at 4%.

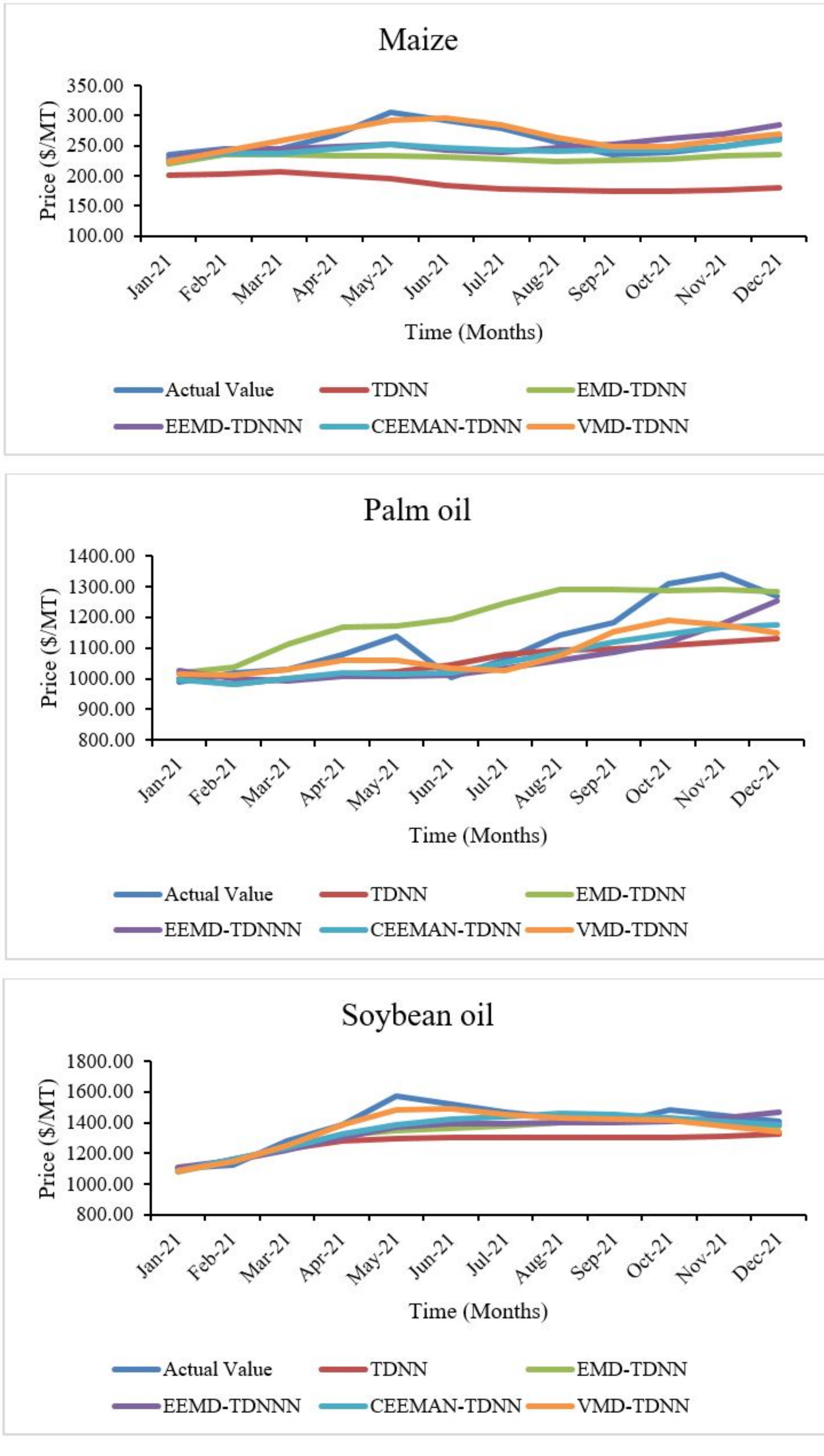

Figure 4: The predicted results of different models for Maize, Palm oil and Soybean oil price series

From the empirical analysis of various models using maize, palm oil, and soybean oil price series, it is clear that the proposed VMD-TDNN model significantly outperforms all other models regarding different forecasting evaluation criteria and thus can be considered as a competitive model for agricultural price forecasting. However, the VMD algorithm requires predetermination of the number of variational modes to be extracted contrary to the EMD and its variants. For EMD variants, the total number of modes is equal to $log_2 T$, where T is the total number of observations in the price series. Further, there is no theoretical or practical approach to determine the number (n) of extracted modes by VMD. For simplicity and to make all models comparable, the number of modes by VMD is chosen the same as that obtained by EMD and its variants, see Bisoi *et al.* (2019); Dragomiretskiy and Zosso (2014); Lahmiri (2016); Liu *et al.* (2018). Indeed, setting a higher number will further reduce the θ even if it is just a little but this will inevitably lead to higher computational burden and processing time during the decomposition process and the training of TDNN. In contrary, setting a lower number may lead to an inefficient representation and characterization of the original time series. The fact that with nine IMFs the VMD–TDNN achieved higher accuracy than others for all price series is very encouraging and promising in itself. However, a formal methodology should be developed in this regard in future works.

Table 6: Forecasting performance of different models for Maize, Palm oil and Soybean oil prices

Forecasting Models	Maize			Palm oil			Soybean oil			TOPSIS Rank
	MAPE	RMSE	Dstat	MAPE	RMSE	Dstat	MAPE	RMSE	Dstat	
TDNN	0.2725	76.29	45.45	0.0690	107.68	36.36	0.0854	143.64	72.73	5
EMD-TDNN	0.1075	36.09	54.54	0.0739	101.38	36.36	0.0477	91.02	90.91	4
EEMD-TDNN	0.0794	27.26	90.90	0.0613	92.57	36.36	0.0449	85.06	81.81	3
CEEMDAN-TDNN	0.0644	24.85	81.82	0.0575	88.86	81.81	0.0377	70.98	90.90	2
VMD-TDNN	0.0345	9.49	90.90	0.0478	76.90	81.81	0.0259	47.28	100.00	1

Table 7: Forecasting performance in terms of DM test of different models for Maize, Palm oil and Soybean oil prices for the one-year forecast horizon

Series	Tested Model	Benchmark Models			
		TDNN	EMD-TDNN	EEMD-TDNN	CEEMDAN-TDNN
Maize	**EMD-TDNN**	12.16(0.000)			
	EEMD-TDNN	13.96(0.000)	2.54(0.013)		
	CEEMDAN-TDNN	11.03(0.000)	5.44(0.000)	1.28(0.111)	
	VMD-TDNN	8.05(0.000)	3.12(0.004)	2.46(0.015)	3.67(0.001)
Palm Oil	**EMD-TDNN**	2.18(0.046)			
	EEMD-TDNN	2.83(0.002)	2.27(0.019)		
	CEEMDAN-TDNN	2.36(0.018)	2.31(0.018)	2.39(0.015)	
	VMD-TDNN	2.76(0.009)	2.81(0.005)	2.08(0.053)	2.56(0.012)
Soybean Oil	**EMD-TDNN**	5.19(0.000)			
	EEMD-TDNN	4.42(0.000)	1.76(0.042)		
	CEEMDAN-TDNN	4.87(0.000)	1.91(0.040)	1.39(0.095)	
	VMD-TDNN	4.66(0.000)	2.05(0.031)	1.90(0.041)	3.44(0.002)

4. Conclusions

Agricultural price series are highly vulnerable to several risks due to biotic and abiotic factors, which account for several characteristics, including nonlinearity and nonstationarity. This paper proposes a new hybrid VMD-TDNN model to improve the prediction accuracy of agricultural price data. The VMD algorithm decomposes a series into a set of subseries or modes for the proposed model. These obtained modes are forecasted separately using the TDNN model, and their forecasted values are aggregated to give a final forecast for a given price series data. VMD has many advantages over EMD based methods, including a better mathematical foundation, data adaptiveness capability, robustness to noise and faster convergence with better accuracy. For empirical evaluation, an extensive comparative analysis of the forecasting performance of the proposed VMD-TDNN model with the four different models is performed using three monthly international price series. The empirical results show that the VMD-TDNN outperforms the competing models in terms of different forecasting evaluation criteria like MAPE, RMSE and D_{stat}. In addition, to better understand the proper order, we utilise a unique technique called TOPSIS, which ranks all models by combining their performances of both level and directional metrics, and VMD-TDNN stands first among all. Further, the DM test result shows that the VMD-TDNN model significantly improves forecasting accuracy from other models. Overall, we can state that the proposed model provides a valuable decision support tool for every agricultural stakeholder who falls in the domain of agricultural price forecasting.

Acknowledgements

The first author is grateful to the University Grants Commission (UGC) for giving financial assistance and the PG School, ICAR-Indian Agricultural Research Institute, New Delhi, for providing the essential facilities for this research. We are also thankful to the reviewer for many insightful comments which have improved the quality of the paper. Further, the reviewer also suggested different packages in LaTeX for improving the readability of the manuscript.

Conflict of interest

There are no possible conflicts of interest that the authors should disclose that are relevant to the content of the work.

References

Bisoi, R., Dash, P. K., and Parida, A. K. (2019). Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis. *Applied Soft Computing*, **74**, 652–678.

Box, G., Jenkins, G., Reinsel, G., and Ljung, G. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley and Sons, 4 edition.

Choudhary, K., Jha, G. K., Kumar, R. R., and Jaiswal, R. (2021). eemdTDNN: EEMD and its variant based time-delay neural network model, https://cran.r-project.org/package=eemdTDNN.

Choudhary, K., Jha, G. K., Kumar, R. R., and Mishra, D. C. (2019). Agricultural commodity price analysis using ensemble empirical mode decomposition: A case study of daily potato price series. *Indian Journal of Agricultural Sciences*, **89**, 882–886.

Choudhary, K., Jha, G. K., Parsad, R., and Jaiswal, R. (2022). vmdTDNN :VMD based time-delay neural network model, https://cran.r-project.org/package=vmdTDNN.

Dragomiretskiy, K. and Zosso, D. (2014). Variational mode decomposition. *IEEE Transactions on Signal Processing*, **62**, 531–544.

Fang, Y., Guan, B., Wu, S., and Heravi, S. (2020). Optimal forecast combination based on ensemble empirical mode decomposition for agricultural commodity futures prices. *Journal of Forecasting*, **39**, 877–886.

FAO (2011). Price volatility in food and agricultural markets : policy responses. pages 1–68.

Hayat, A. and Bhatti, M. I. (2013). Masking of volatility by seasonal adjustment methods. *Economic Modelling*, **33**, 676–688.

Haykin, S. (2009). *Neural Networks and Learning Machines, Person Education*, volume 1-3. PHI Learning, INDIA, 3 edition.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Snin, H. H., Zheng, Q., Yen, N. C., Tung, C. C., and Liu, H. H. (1998). The empirical mode decomposition and the Hubert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **454**, 903–995.

Hwang, C. L. and Yoon, K. (1981). *In Multiple attribute decision making*, pages 58–191. Springer.

Jadhav, V., Chinnappa Reddy, B. V., and Gaddi, G. M. (2017). Application of arima model for forecasting agricultural prices. *Journal of Agricultural Science and Technology*, **19**, 981–992.

Jaiswal, R., Jha, G. K., Kumar, R. R., and Choudhary, K. (2022). Deep long short-term memory based model for agricultural price forecasting. *Neural Computing and Applications*, **34**, 4661–4676.

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.

Kumar, R. R., Jha, G. K., Choudhary, K., and Mishra, C. (2020). Spatial integration and price transmission among major potato markets in India. *Indian Journal of Agricultural Sciences*, **90**, 581–585.

Lahmiri, S. (2016). Intraday stock price forecasting based on variational mode decomposition. *Journal of Computational Science*, **12**, 23–27.

Liu, H., Mi, X., and Li, Y. (2018). Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. *Energy Conversion and Management*, **159**, 54–64.

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.

Qian, Z., Pei, Y., Zareipour, H., and Chen, N. (2019). A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. *Applied Energy*, **235**, 939–953.

Torres, M. E., Colominas, M. A., Schlotthauer, G., and Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4144–4147.

Trostle, R. (2011). Why another food commodity price spike? Technical report.

Wu, Z. and Huang, N. E. (2009). Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, **1**, 1–41.

Xu, X. and Zhang, Y. (2021). Corn cash price forecasting with neural networks. *Computers and Electronics in Agriculture*, **184**, 106–120.

Yu, L., Wang, S., and Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics*, **30**, 2623–2635.

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

Zhu, B., Shi, X., Chevallier, J., Wang, P., and Wei, Y. M. (2016). An Adaptive Multiscale Ensemble Learning Paradigm for Nonstationary and Nonlinear Energy Price Time Series Forecasting. *Journal of Forecasting*, **35**, 633–651.

Appendix

R codes used for empirical evaluation of the study

```
# To check and install the packages used in this analysis
ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[ ,"Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}
# usage
packages <- c("tseries", "moments", "Rlibeemd", "VMDecomp")
ipak(packages)
library(tseries)
library(moments)
library(Rlibeemd)
library(VMDecomp)


#Importing the actual price data specifying the location of the data file
data=read.csv(file.choose(), header=TRUE)
data
#plotting of the imported data
plot(ts(data))

#transforming the data into numeric vector (1-dimensional)
data=as.matrix(data)
data=as.vector(data)

#Basic descriptive of the data set
library(moments)
summary(data)
sd(data)
skewness(data)
kurtosis(data)
jarque.test(data)

#Stationarity and linearity test
adf.test(data) # Augmented Dickey-fuller test for testing stationarity
bds.test(data) # Brock-Dechert Shienkman test for testing nonlinearity

#Data Decomposition through EMD, EEMD, CEEMDAN, and VMD
library(Rlibeemd)

# Decomposition of price data using EMD technique
EMD=emd(data, num_imfs = 0, S_number = 4L, num_siftings = 50L)
```

```
# Decomposition of price data using EEMD technique
EEMD=eemd(ts(data), num_imfs = 0, ensemble_size = 250L, noise_strength = 0.2,
↪  S_number = 4L, num_siftings = 50L, rng_seed = 0L, threads = 0L)

# Decomposition of price data using CEEMDAN technique
CEEMDAN=ceemdan(ts(data), num_imfs = 0, ensemble_size = 250L, noise_strength =
↪  0.2, S_number = 4L, num_siftings = 50L, rng_seed = 0L, threads = 0L)

# Decomposition of price data using VMD technique
VMD=vmd(data, alpha = 2000, tau = 0, K = 9, DC = FALSE, init = 1, tol = 1e-06)

# Plotting of decomposed series

# PLotting of decomposed series extracted by EMD technique
plot(EMD,xlab="Time (Month)")

# PLotting of decomposed series extracted by EEMD technique
plot(EEMD,xlab="Time (Month)")

# PLotting of decomposed series extracted by CEEMDAN technique
plot(CEEMDAN,xlab="Time (Month)")


# VMdecomp package does not allow for auto-plot of all series,
# so we will extract all the decomposed series one by one done by VMD technique
# and then combine them in a two dimensional matrix and then plot them

#Extraction of all IMFS
AllIMF <- ts(VMD$u)

# VMD decompose price series in reverse order (From low to high frequency)
# in contrary to EMD variants.So IMF1 will be the last column, IMF2 will be
# the second last column,...

# Extraction of each IMF one by one
IMF1=ts(AllIMF[,9])
IMF2=ts(AllIMF[,8])
IMF3=ts(AllIMF[,7])
IMF4=ts(AllIMF[,6])
IMF5=ts(AllIMF[,5])
IMF6=ts(AllIMF[,4])
IMF7=ts(AllIMF[,3])
IMF8=ts(AllIMF[,2])
IMF9=ts(AllIMF[,1])

# Combining of all IMFs
VMD_IMFs <- cbind.data.frame(IMF1, IMF2, IMF3, IMF4, IMF5, IMF6, IMF7, IMF8,IMF9)
VMD_IMFs  <- ts(VMD_IMFs)

# Plotting of all IMFs of VMD together
```

```
plot(VMD_IMFs)

# Modelling and Forecasting results of EMDTDNN model
emd_tdnn=function(data, stepahead = 12, num.IMFs = emd_num_imfs(length(data)),
          s.num = 4L, num.sift = 50L)
{
  n.IMF <- num.IMFs # To find the total number of IMFs
  AllIMF <- emd(data, num_imfs = n.IMF, S_number = s.num, num_siftings =
  ↪  num.sift)
  data_trn <- ts(head(data, round(length(data) - stepahead))) # Extracting the
  ↪  training set
  data_test <- ts(tail(data, stepahead)) # Extracting the testing set
  IMF_trn <- AllIMF[-c(((length(data) - stepahead) + 1):length(data)),
  ]
  Fcast_AllIMF <- NULL
  # Applying For loop to model and forecast each decomposed series using TDNN
  ↪  model
  for (IMF in 1:ncol(IMF_trn)) {
    IndIMF <- NULL
    IndIMF <- IMF_trn[, IMF]
    EMDTDNNFit <- forecast::nnetar(as.ts(IndIMF))
    EMDTDNN_fcast = forecast::forecast(EMDTDNNFit, h = stepahead)
    EMDTDNN_fcast_Mean = EMDTDNN_fcast$mean
    Fcast_AllIMF <- cbind(Fcast_AllIMF, as.matrix(EMDTDNN_fcast_Mean))
  }
  # Combining all the forecasts to get final forecast using EMD-TDNN
  FinalEMDTDNN_fcast <- ts(rowSums(Fcast_AllIMF, na.rm = T))
  # Finding different evaluation criteria based on testing data set
  MAE_EMDTDNN = mean(abs(data_test - FinalEMDTDNN_fcast))
  MAPE_EMDTDNN = mean(abs(data_test - FinalEMDTDNN_fcast)/data_test)
  rmse_EMDTDNN = sqrt(mean((data_test - FinalEMDTDNN_fcast)^2))
  Plot_IMFs <- AllIMF
  AllIMF_plots <- plot(Plot_IMFs)
  return(list(TotalIMF = n.IMF, AllIMF = AllIMF, data_test = data_test,
              AllIMF_forecast = Fcast_AllIMF, FinalEMDTDNN_forecast =
              ↪  FinalEMDTDNN_fcast,
              MAE_EMDTDNN = MAE_EMDTDNN, MAPE_EMDTDNN = MAPE_EMDTDNN,
              rmse_EMDTDNN = rmse_EMDTDNN, AllIMF_plots = AllIMF_plots))
}
EMDTDNN=emd_tdnn(data, stepahead = 12, num.IMFs = emd_num_imfs(length(data)),
                s.num = 4L, num.sift = 50L)
EMDTDNN

# Forecasting result of EEMDTDNN model
EEMD_TDNN=function (data, stepahead = 12, num.IMFs = emd_num_imfs(length(data)),
          s.num = 4L, num.sift = 50L, ensem.size = 250L, noise.st = 0.2)
{
  n.IMF <- num.IMFs # To find the total number of IMFs
  AllIMF <- eemd(ts(data), num_imfs = n.IMF, ensemble_size = ensem.size,
```

```
                noise_strength = noise.st, S_number = s.num, num_siftings =
                 ↪  num.sift,
                rng_seed = 0L, threads = 0L)
  data_trn <- ts(head(data, round(length(data) - stepahead)))  # Extracting the
   ↪  training set
  data_test <- ts(tail(data, stepahead))  # Extracting the testing set
  IMF_trn <- AllIMF[-c(((length(data) - stepahead) + 1):length(data)),
  ]
  Fcast_AllIMF <- NULL
  # Applying For loop to model and forecast each decomposed series using TDNN
   ↪  model
  for (IMF in 1:ncol(IMF_trn)) {
    IndIMF <- NULL
    IndIMF <- IMF_trn[, IMF]
    EEMDTDNNFit <- forecast::nnetar(as.ts(IndIMF))
    EEMDTDNN_fcast = forecast::forecast(EEMDTDNNFit, h = stepahead)
    EEMDTDNN_fcast_Mean = EEMDTDNN_fcast$mean
    Fcast_AllIMF <- cbind(Fcast_AllIMF, as.matrix(EEMDTDNN_fcast_Mean))
  }
  # Combining all the forecasts to get final forecast using EMD-TDNN
  FinalEEMDTDNN_fcast <- ts(rowSums(Fcast_AllIMF, na.rm = T))
  # Finding different evaluation criteria based on testing data set
  MAE_EEMDTDNN = mean(abs(data_test - FinalEEMDTDNN_fcast))
  MAPE_EEMDTDNN = mean(abs(data_test - FinalEEMDTDNN_fcast)/data_test)
  rmse_EEMDTDNN = sqrt(mean((data_test - FinalEEMDTDNN_fcast)^2))
  Plot_IMFs <- AllIMF
  AllIMF_plots <- plot(Plot_IMFs)
  return(list(TotalIMF = n.IMF, data_test = data_test, AllIMF_forecast =
   ↪  Fcast_AllIMF,
              FinalEEMDTDNN_forecast = FinalEEMDTDNN_fcast, MAE_EEMDTDNN =
               ↪  MAE_EEMDTDNN,
              MAPE_EEMDTDNN = MAPE_EEMDTDNN, rmse_EEMDTDNN = rmse_EEMDTDNN,
              AllIMF_plots = AllIMF_plots))
}
EEMDTDNN=EEMD_TDNN(data, stepahead = 12, num.IMFs = emd_num_imfs(length(data)),
                  s.num = 4L, num.sift = 50L, ensem.size = 250L, noise.st = 0.2)
EEMDTDNN

#Forecasting Result of CEEMDANTDNN model
ceemdan_TDNN=function (data, stepahead = 12, num.IMFs =
↪  emd_num_imfs(length(data)),
          s.num = 4L, num.sift = 50L, ensem.size = 250L, noise.st = 0.2)
{
  n.IMF <- num.IMFs # To find the total number of IMFs
  AllIMF <- ceemdan(ts(data), num_imfs = n.IMF, ensemble_size = ensem.size,
                    noise_strength = noise.st, S_number = s.num, num_siftings =
                     ↪  num.sift,
                    rng_seed = 0L, threads = 0L)
  data_trn <- ts(head(data, round(length(data) - stepahead)))  # Extracting the
   ↪  training set
```

```
  data_test <- ts(tail(data, stepahead))  # Extracting the testing set
  IMF_trn <- AllIMF[-c(((length(data) - stepahead) + 1):length(data)),
  ]
  Fcast_AllIMF <- NULL
  # Applying For loop to model and forecast each decomposed series using TDNN
   ↪  model
  for (IMF in 1:ncol(IMF_trn)) {
    IndIMF <- NULL
    IndIMF <- IMF_trn[, IMF]
    CEEMDANTDNNFit <- forecast::nnetar(as.ts(IndIMF))
    CEEMDANTDNN_fcast = forecast::forecast(CEEMDANTDNNFit,
                                           h = stepahead)
    CEEMDANTDNN_fcast_Mean = CEEMDANTDNN_fcast$mean
    Fcast_AllIMF <- cbind(Fcast_AllIMF, as.matrix(CEEMDANTDNN_fcast_Mean))
  }
  # Combining all the forecasts to get final forecast using EMD-TDNN
  FinalCEEMDANTDNN_fcast <- ts(rowSums(Fcast_AllIMF, na.rm = T))
  # Finding different evaluation criteria based on testing data set
  MAE_CEEMDANTDNN = mean(abs(data_test - FinalCEEMDANTDNN_fcast))
  MAPE_CEEMDANTDNN = mean(abs(data_test - FinalCEEMDANTDNN_fcast)/data_test)
  rmse_CEEMDANTDNN = sqrt(mean((data_test - FinalCEEMDANTDNN_fcast)^2))
  Plot_IMFs <- AllIMF
  AllIMF_plots <- plot(Plot_IMFs)
  return(list(TotalIMF = n.IMF, data_test = data_test, AllIMF_forecast =
   ↪  Fcast_AllIMF,
              FinalCEEMDANTDNN_forecast = FinalCEEMDANTDNN_fcast, MAE_CEEMDANTDNN
               ↪  = MAE_CEEMDANTDNN,
              MAPE_CEEMDANTDNN = MAPE_CEEMDANTDNN, rmse_CEEMDANTDNN =
               ↪  rmse_CEEMDANTDNN,
              AllIMF_plots = AllIMF_plots))
}
CEEMDANTDNN=ceemdan_TDNN(data, stepahead = 12, num.IMFs =
 ↪  emd_num_imfs(length(data)),
                        s.num = 4L, num.sift = 50L, ensem.size = 250L, noise.st =
                         ↪  0.2)
CEEMDANTDNN

# Forecasting Result of VMDTDNN model
VMD_TDNN=function (data, stepahead = 12, nIMF = 9, alpha = 2000, tau = 0,
          D = FALSE)
{
  data <- ts(data)
  data <- as.vector(data)
  v <- vmd(data, alpha = 2000, tau = 0, K = nIMF, DC = D, init = 1,
           tol = 1e-06)
  AllIMF <- v$u
  data_trn <- ts(head(data, round(length(data) - stepahead))) # Extracting the
   ↪  training set
  data_test <- ts(tail(data, stepahead)) # Extracting the testing set
  IMF_trn <- AllIMF[-c(((length(data) - stepahead) + 1):length(data)),
```

```
  ]
  Fcast_AllIMF <- NULL
  # Applying For loop to model and forecast each decomposed series using TDNN
   ↪  model
  for (AllIMF in 1:(ncol(IMF_trn))) {
    IndIMF <- NULL
    IndIMF <- IMF_trn[, AllIMF]
    VMDTDNNFit <- forecast::nnetar(as.ts(IndIMF))
    VMDTDNN_fcast = forecast::forecast(VMDTDNNFit, h = stepahead)
    VMDTDNN_fcast_Mean = VMDTDNN_fcast$mean
    Fcast_AllIMF <- cbind(Fcast_AllIMF, as.matrix(VMDTDNN_fcast_Mean))
  }
  # Combining all the forecasts to get final forecast using EMD-TDNN
  FinalVMDTDNN_fcast <- ts(rowSums(Fcast_AllIMF, na.rm = T))
  # Finding different evaluation criteria based on testing data set
  MAE_VMDTDNN = mean(abs(data_test - FinalVMDTDNN_fcast))
  MAPE_VMDTDNN = mean(abs(data_test - FinalVMDTDNN_fcast)/data_test)
  RMSE_VMDTDNN = sqrt(mean((data_test - FinalVMDTDNN_fcast)^2))
  return(list(AllIMF = AllIMF, data_test = data_test, AllIMF_forecast =
   ↪  Fcast_AllIMF,
              FinalVMDTDNN_forecast = FinalVMDTDNN_fcast, MAE_VMDTDNN =
               ↪  MAE_VMDTDNN,
              MAPE_VMDTDNN = MAPE_VMDTDNN, RMSE_VMDTDNN = RMSE_VMDTDNN))
}
VMDTDNN=VMD_TDNN(data, stepahead = 12, nIMF = 9, alpha = 2000, tau = 0,
                D = FALSE)
VMDTDNN
```