Neural Processing Letters (2024) 56:92
https://doi.org/10.1007/s11063-024-11552-w

CEEMDAN-Based Hybrid Machine Learning Models for Time Series Forecasting Using MARS Algorithm and PSO-Optimization

Sandip Garai[1,3] · Ranjit Kumar Paul[2] · Md Yeasin[2] · A. K. Paul[2]

Accepted: 5 February 2024

Abstract
Accurate prediction of time series data is crucial for informed decision-making and economic development. However, predicting noisy time series data is a challenging task due to their irregularity and complex trends. In the past, several attempts have been made to model complex time series data using both stochastic and machine learning techniques. This study proposed a CEEMDAN-based hybrid machine learning algorithm combined with stochastic models to capture the volatility of weekly potato price in major markets of India. The smooth decomposed component is predicted using stochastic models, while the coarser components, selected using MARS, are fitted into two different machine learning algorithms. The final predictions for the original series are obtained using optimization techniques such as PSO. The performance of the proposed algorithm is measured using various metrics, and it is found that the optimization-based combination of models outperforms the individual counterparts. Overall, this study presents a promising approach to predict price series using a hybrid model combining stochastic and machine learning techniques, with feature selection and optimization techniques for improved performance.

Keywords CEEMDAN · Ensemble · Hybrid model · Machine learning · Optimization

✉ Ranjit Kumar Paul
ranjit.paul@icar.gov.in

Sandip Garai
sandipnicksandy@gmail.com

Md Yeasin
md.yeasin@icar.gov.in

A. K. Paul
amrit.paul@icar.gov.in

[1] The Graduate School, ICAR-Indian Agricultural Research Institute, New Delhi 110012, India

[2] ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110012, India

[3] ICAR-Indian Institute of Agricultural Biotechnology, Ranchi 834003, India

Published online: 06 March 2024

Abbreviations

ANN	Artificial neural network
ARIMA	Autoregressive integrated moving average
CEEMDAN	Complete ensemble empirical mode decomposition with adaptive noise
E_{LM}	Legates and McCabe index
GARCH	Generalized autoregressive conditional heteroscedasticity
IMF	Intrinsic mode function
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MARS	Multivariate adaptive regression splines
MASE	Mean absolute scaled error
ML	Machine learning
NSE	Nash–Sutcliffe efficiency
PSO	Particle swarm optimization
RMSE	Root mean squared error
RRMSE	Relative root mean squared error
SVR	Support vector regression
WI	Willmott's index

1 Introduction

A time series is often used to analyze and predict trends or patterns in various fields such as economics, finance, and engineering. ARIMA model is most used parametric time series model that combines AR and MA components with a differencing process to handle non-stationarity. GARCH is an extension of the ARIMA model that can capture the conditional variance of a time series. A vast literature is available for parametric time series model including their application; such as Paul et al. [1] applied ARIMA model with various important weather parameters as exogenous variable to forecast wheat yield data in the district of Kanpur in Uttar Pradesh. Paul [2] used a long memory time series model, Autoregressive fractionally integrated moving average (ARFIMA) model to forecast daily wholesale price of pigeon pea. Rakshit et al. [3] utilized various asymmetric GARCH-type of models to capture the volatility of weekly modal price of onion in Delhi, Lasalgaon and Bengaluru market and proved the outperformance of the Asymmetric power autoregressive heteroscedastic (APARCH) model over other alternatives.

As literature suggests, in many of the cases such as presence of noise, high skewness and heteroscedasticity, parametric time series models cannot capture the underlying data generating process and estimate the function efficiently. So, several non-parametric methods including machine learning (ML) algorithms are evolved. ML methods, such as ANNs and SVR, are efficient algorithms in time series analysis that can handle complex non-linear relationships and interactions between variables. ANNs can model complex patterns in time series data by learning from historical data and adapting to changing patterns [4], while SVR can identify a hyperplane that maximizes the margin between the data points and the hyperplane to capture non-linear patterns and make accurate predictions [5]. Application of ML techniques in time series forecasting may be found in many available literatures [6–10]. Gu et al. [11] developed a method for housing price forecasting using genetic algorithm and Support vector machine (SVM). Gu et al. [12] also developed a new SVR based forecasting

model for late blight of potato (BLIGHT-SVR). Thivakaran and Ramesh [13] proposed novel supervised and ANN algorithm-based demand and sales forecasting approach for big mart. Chen et al. [14–16] tried hybrid ML algorithm combining various models including SVR, PSO etc. in several areas, viz, for customers' credit assessment, prediction of traffic etc.

To improve the accuracy of time series forecasting, a combination of multiple methodologies has been explored, which can mitigate the risk of selecting an erroneous method and also provide the advantages of several methods [17]. Zhang [17] developed a hybrid ARIMA-ANN model which was used to predict water quality [18] and forecast electricity price [19]. The fusion of ARIMA and ANN was identified as an effective approach to enhance wind speed forecasting by refining the associations between wind speed and various meteorological variables [20]. As ARIMA method cannot model non-linearity in the time series data induced by high volatility, Rubio and Alba [21] introduced a hybrid ARIMA and SVR-based hybrid methodology (ARIMA–SVR) to forecast stocks of New York stock exchange (NYSE). Combination of multiple forecasting models is commonly implemented by assigning equal weightage to each model's output [22–24]. However, optimizing these weights is a rare practice in the literature.

Furthermore, noise present in existing datasets cannot be captured accurately by any parametric or non-parametric method, or combination of models. In such cases, decomposition methods are a valuable tool in time series modeling as they allow us to separate the different components of a time series, which can be analyzed and modeled independently to achieve more accurate predictions and forecasts [25–27]. Decomposition of the original dataset has been explored by many researchers [28, 29]. Babu and Reddy [30] have used MA filter for smoothening the dataset followed by an application of ARIMA on smoothed component and ANN in the remaining part. There are many decomposition methods available in the literature such as: EMD, EEMD, EWT, CEEMDAN etc. Among them CEEMDAN decomposition technique has been explored in this research work. CEEMDAN, a popular decomposition method, can extract useful features from the actual dataset that can improve prediction accuracy. By breaking down the time series into its underlying components, we can analyze and model each component separately, leading to more precise predictions. According to Torres et al. [31] CEEMDAN can resolve all of the flaws by providing a better spectral decomposition of the IMFs at lower computational time [32]. Li and Li [33] used an improved CEEMDAN and PSO-SVR-based algorithm for glucose detection. Garai and Paul [23] developed an ensemble model using CEEMDAN decomposition and Machine Intelligence (MI) models to forecast S&P 500 index. But they have not utilized any feature selection algorithm or any optimization technique to get final prediction. The combination model utilizing CEEMDAN for short-term load forecasting demonstrated superior performance compared to alternative models [34, 35].

Many feature extraction techniques have been evolved to get the latent signal contained in the data for further processing [36, 37]. Therefore, the extracted features only can be fitted into various models to eliminate noise, reduce the redundancy and computational complexity as well as improve the prediction accuracy [38–40]. Some of feature selection techniques used in literature are principal component analysis (PCA), decision trees (DT) [41], MARS etc. MARS is a popular feature selection technique that uses a combination of linear and nonlinear regression to identify the most relevant features. Kao et al. [42] applied MARS-based feature selection technique for variable selection in the domain of stock market forecasting. Adnan et al. [43] combined MARS and least square SVR technique for streamflow prediction. Bose et al. [44] implemented MARS and deep neural network in the area of stock price forecasting.

Optimization of weights to combine different models' predictions is an essential aspect of ensemble learning, which can significantly improve the accuracy and reliability of predictions [45]. Maximizing the predictive power, reducing the impact of errors, addressing model biases, improving robustness, and interpretability of the ensemble are some reasons why optimization of weights is important [37, 46–48]. PSO, an optimization algorithm, works by simulating the behavior of a swarm of particles, where each particle represents a potential solution to the optimization problem [49, 50]. Preprocessing coupled with feature selection and optimization had also been studied for performance improvement. Heidari et al. [51] pointed out the importance of highly accurate prediction models. By decomposing the original signal, the series can be denoised and features can be extracted at different frequencies. Despite these advantages, the use of variable selection techniques on CEEMDAN-decomposed series have not been explored.

Therefore, an attempt has been made to developed a CEEMDAN based hybrid machine learning model using MARS-based feature selection technique and PSO optimization. Based on the above discussion, this work proposed two different hybrid models (CARIGAAN and CARIGAS) for the efficient handling of noisy complex time series data. To evaluate the performance of the proposed model with existing stochastic and machine learning models, high volatile agricultural price series have been used.

2 Methodology

2.1 CEEMDAN Decomposition

CEEMDAN was fundamentally proposed by Wu and Huang as an extension to the original EEMD method [52, 53]. Later, Torres et al. [31] developed an altered version of CEEMDAN. They named it as EMD-NN as their method of selection for relevant IMFs was NN. CEEMDAN may be considered as a signal processing technique in time-series analysis, that can handle non-stationary as well as noisy series. A signal can be decomposed using CEEMDAN algorithms as follows.

Step 1 IMFs are generated using EMD (Eq. 1).
Step 2 Standard deviation (σ) of the IMFs generated in step 1 is calculated. Now, they are grouped into certain frequency bands.
Step 3 Each of the frequency band is added with a white noise. Their Standard deviations (SD) are determined from σ. Perform step 1 on these new IMFs which are less prone to noise.
Step 4 Step 2 and step 3 are repeated.

Repeat steps 2–4 until a stopping criterion is met, like desired noise level or maximum number of iterations whichever achieved first.

$$y(t) = \sum_{i=1}^{N} h_i(t) + r(t) \tag{1}$$

Here, original signal ($y(t)$) is decomposed by EMD into N number of IMFs (Eq. 1). The i^{th} IMF is $h_i(t)$ which is of zero mean and has a well-defined frequency: f_i and an amplitude: a_i; $r(t)$ is residual. The difference of number of extrema and zero-crossings must be at most one. The envelope formed by maxima and minima of the series must also be zero. Then step 3 is applied (Eqs. 2, 3, and 4).

$$h_i(t) = a_i(t) + cos(\theta_i(t)) \tag{2}$$

$$a_i(t) = A_i(t) + \varepsilon_i(t) \tag{3}$$

$$\theta_i(t) = \int \omega_i(s)ds + \varphi_i(t) \tag{4}$$

Amplitude envelope is represented as $A_i(t)$ and noise added to the signal at i^{th} frequency band is $\varepsilon_i(t)$ in Eq. 3. Instantaneous frequency and initial phase are represented as $\omega_i(s)$ and $\varphi_i(t)$ respectively in Eq. 4. Final IMFs obtained by CEEMDAN process are less noisy and can be used for further analysis.

2.2 ARIMA and GARCH Model

ARIMA [54] and GARCH [55] are most commonly named time series models in financial and economic forecasting. ARIMA captures autocorrelation of a series by using its lags and differences in the model. The ARIMA is mathematically represented as follows (Eq. 5).

$$y_t = c + \sum_{i=1}^{p} \varnothing_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{5}$$

An ARIMA representation of original time series (y_t) in Eq. 5 contains c, $\varnothing_i$, θ_j, and ε_t which are a constant, autoregressive and moving average coefficients, and an error term respectively. Number of autoregressive and moving average lags included in the model are p and q respectively. GARCH is a model that captures the time-varying volatility of a time series. However, a GARCH model can be represented as follows (Eqs. 6, and 7).

$$\varepsilon_t = \sigma_t z_t \tag{6}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{7}$$

A GARCH model is used to grab time-varying volatility of a time series. In Eq. 6, z_t is a standard normal variate. In Eq. 7, σ_t^2 is the conditional variance of the time series which has been modelled against lagged error and variance terms.

2.3 ANN

ANNs are ML algorithm that consists of layers of interconnected neurons, which process information [56–58]. A single node in an ANN can be mathematically presented as (Eq. 8)

$$n_j = f\left(\sum_i w_i x_i + b\right) \tag{8}$$

Output from j^{th} node or neuron is n_j; w_i is weight associated with i^{th} input, y_i; b is bias term and f represents activation function applied to the weighted sum of the inputs in Eq. 8. Sigmoid, Rectified linear unit (ReLU), softmax are some of the most popular activation functions. With these notions a feedforward NN is presented below (Eq. 9).

$$y = f(W_2 f(W_1 x) + b_1) + b_2 \tag{9}$$

Weight matrices, input vector, biases, and output vector are represented in Eq. 9 as W, x, b, and y respectively. Subscripts in this equation represent 1st and 2nd layers. Training of the

NN comprises of adjusting these weights and biases such that predicted output minimally differs from the actual input. However, one optimization technique for doing so is stochastic gradient descent. Time series analysis is one of its wide range of applications.

2.4 SVR

SVR is a type of regression algorithm that cultivates Support vector machines (SVM) to establish relationship between the predictors and response variables. It maximizes the margin between the predicted and actual values by finding a hyperplane ($f(x)$) which can be mathematically presented as below (Eq. 10)

$$f(x) = w^{'}x + b \tag{10}$$

In Eq. 10, w is the weight vector. A user-defined tolerance parameter ε is used to minimize difference between predicted and actual values. Some slack variables are used to allow the possibility of some observations lying out of bound of the margin in SVR optimization process which is certainly a quadratic programming problem. Whereas, penalty parameter will control the tradeoff width of the margin and amount of error tolerable which is defined by external user.

2.5 MARS Algorithm

MARS is a non-linear and non-parametric regression technique. This method was firstly introduced by [59]. A divide and conquer technique is used to train the model by partitioning the training dataset in separate regions. Through a fast and intensive search procedure and hinge functions, the knot points, i.e., the end points of the intervals of the input feature space are found. The important input variables are chosen one by one, as well as the relationship or interactions between them if any are also found for the better fit of the model. A general MARS model can be expressed by Eq. 11.

$$f(x) = a_0 + \sum_{m=1}^{M} a_m \prod_{k=1}^{K} \left[s_{k,m} \left(x(k, m) - t_{k,m} \right) \right] \tag{11}$$

a_0 is constant, a_m are the model coefficients, M is the number of basis functions, K is the number of splits, and $t_{k,m}$ depicts knot locations. $s_{k,m} = -1 or + 1$, indicates the left/right sense of the associated hinge functions respectively. $x(k, m)$ is the label of the independent variables.

It is a two-phase process to build and optimize a MARS model. Initially, all the variables available are allowed to freely enter into the model and interact with each other or are restricted to enter as additive components only. Then MARS finds the pairs of basic functions for the maximum reduction of sum of squares. In a pair, two functions are two different side of a mirrored hinge function. A hinge function can be defined by the knot/constant (t) and the variable (x) as: max(0, $x - t$) or (0, $t - x$). The basis functions are continuously chosen for the MARS model through the greedy search algorithm until the change in the residual error is too small to be considered or the maximum number of functions reached, i.e., the end of the process. This ends the first stage of MARS model building. Now, in the second stage, general cross validation (GCV) criterion (Eq. 12) is used to choose and keep the best of the

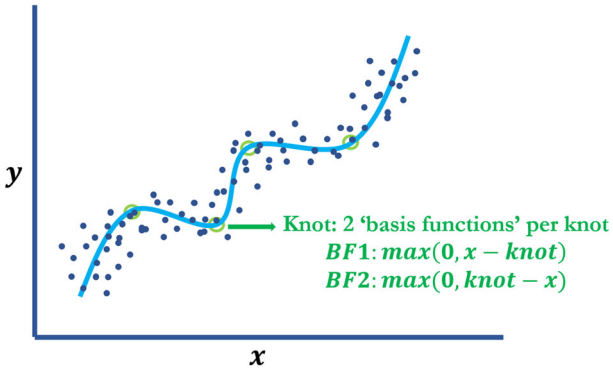

Fig. 1 MARS-based piecewise modeling

basis functions and delete the less important ones.

$$GCV(M) = \frac{1}{N}\sum_{i=1}^{N}\frac{[y_i - f_M(x_i)]^2}{\left[1 - \frac{C(M)}{N}\right]^2} \tag{12}$$

N and M are number of observations and the number of basis functions: $[f_M(x_i)]$; $C(M)$ is cost-penalty measure for model-complexity to penalize the complexity of the fitted model and for avoiding overfitting to make the model parsimonious.

MARS perform better than optimally pruned extreme learning machine (OP-ELM) and M5 model tree (M5Tree) [43]. MARS model (Fig. 1) is flexible in the sense that it fits piecewise linear equations at distinct intervals of the regressor variable space to approximate the non-linearity of the model. This means there will not be a fixed slope for the model and may change at the change of one interval to another when the 'knots' are crossed.

2.6 PSO Optimization

The PSO technique can be considered as a good solution for the engineering challenges of combining different forecast results to combine into one for the best performance of the sets of models. Convergence speed of the PSO technique is very fast and this technique can also solve multidimensional problem. The method was first proposed by Kenny [60] and also by Kennedy and Eberhart, [61]. PSO optimization can be obtained by iterative formulae [33] using Eq. 13 and Eq. 14.

$$\begin{aligned} V_{i,j}(k+1) = {} & \omega * V_{i,j}(k) + c_1 * rand(.) * \left(p_{best_{i,j}}(k) - X_{i,j}(k)\right) \\ & + c_2 * rand(.) * \left(g_{best_{i,j}}(k) - X_{i,j}(k)\right) \end{aligned} \tag{13}$$

$$X_{i,j}(k+1) = X_{i,j}(k) + V_{i,j}(k+1) \tag{14}$$

ith particle from the initial swarm (particle) of size $i = 1(1)N$ and dimension $j = 1(1)D$ is represented by $X_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,D})^T$. The velocity of each particle in the population is expressed as $V_i = (V_{i,1}, V_{i,2}, \ldots, V_{i,D})^T$. $rand(.)$ represents the random number between 0 and 1. The individual and global extreme values is expressed by the terms $p_{best_{i,j}}$ and $g_{best_{i,j}}$ respectively. Learning (acceleration) factors, i.e., c_1 and c_2 falls between

2 and 2.05 [62]. Weighting factor ω helps to decrease the velocity of the particles and control the swarm in this way and is expressed in Eq. 15.

$$\omega = \omega_{min} + \frac{(T_{max} - T)(\omega_{max} - \omega_{min})}{T_{max}} \tag{15}$$

In Eq. 15, T, and T_{max} indicates current and maximum iteration numbers.

To determine the best combining weights of the prediction values of different models MSE can be employed as a fitness function. The steps of PSO algorithm (Figs. 2 and 3) is provided below:

Step 1 Initializing the weights of prediction values.
Step 2 Set $T = 1$
Step 3 Set the parameter values of the PSO algorithm, and also mention the population size.

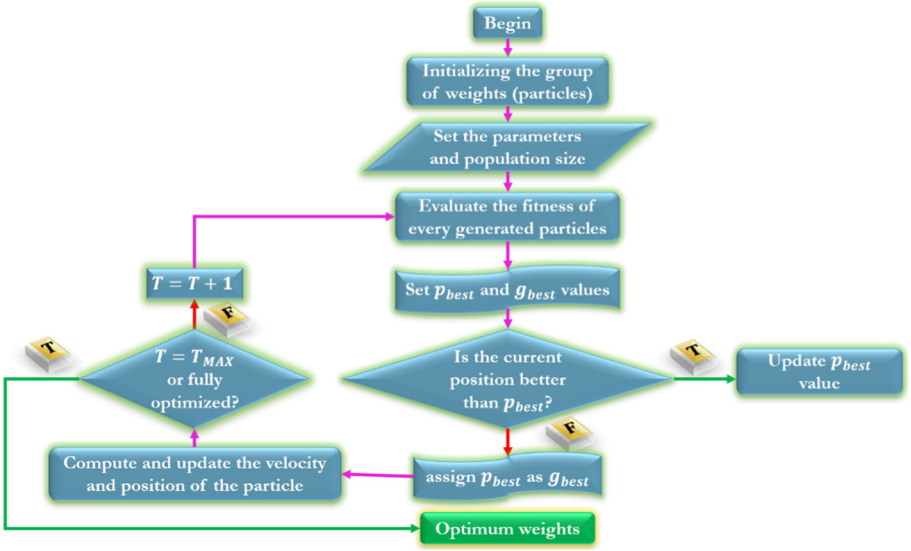

Fig. 2 PSO algorithm steps

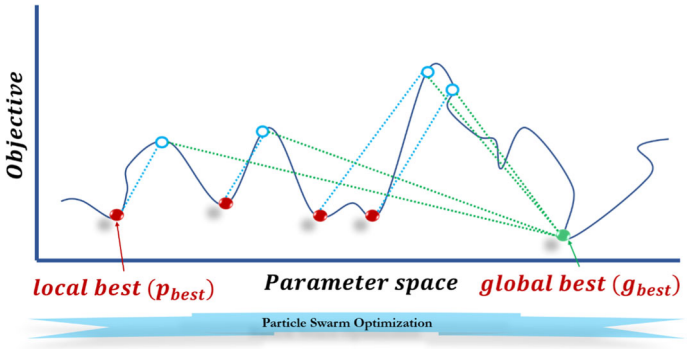

Fig. 3 PSO finding local and global best

Step 4 Produce particles through mentioning position (X_i) and velocity (V_i) vectors.
Step 5 Calculate the fitness of every generated particle.
Step 6 Update the individual extreme value ($p_{best_{i,j}}$) if the value of its fitness is better than previous extreme value.
Step 7 Update the global extreme value ($g_{best_{i,j}}$) accordingly if in step 6 individual extreme value is being updated. In the meantime, compute and update the velocity and position of the particle also.
Step 8 Repeat these steps by increasing the iteration number ($T = T + 1$) until $T = T_{max}$.
Step 9 End of optimization.

After completion of the optimization process, final weights of the prediction combination with minimum MSE will be obtained and also final forecast by then.

2.7 Formulation of Proposed Algorithm

In this section, we present a comprehensive account of the creation of a hybrid model based on the CEEMDAN algorithm. Additionally, we incorporate a feature selection technique using MARS, and combine it with stochastic models such as ARIMA and GARCH, as well as machine learning models including ANN and SVR. To further enhance the performance of the model, we optimize the weights of prediction combinations using PSO. A step-by-step procedure for constructing the algorithm is described below (Fig. 4).

i. Prepare the log return series and the lag series of actual and log-return series.
ii. Create training and testing series for both the actual and log-return series.
iii. Apply CEEMDAN to the log-return series and divide it into training and testing parts.
iv. Fit the smooth part of the training data obtained from CEEMDAN using an ARIMA/GARCH model and obtain forecasts for the testing data.
v. Check the residuals obtained from step iv for any patterns and store them if necessary, for step vi.

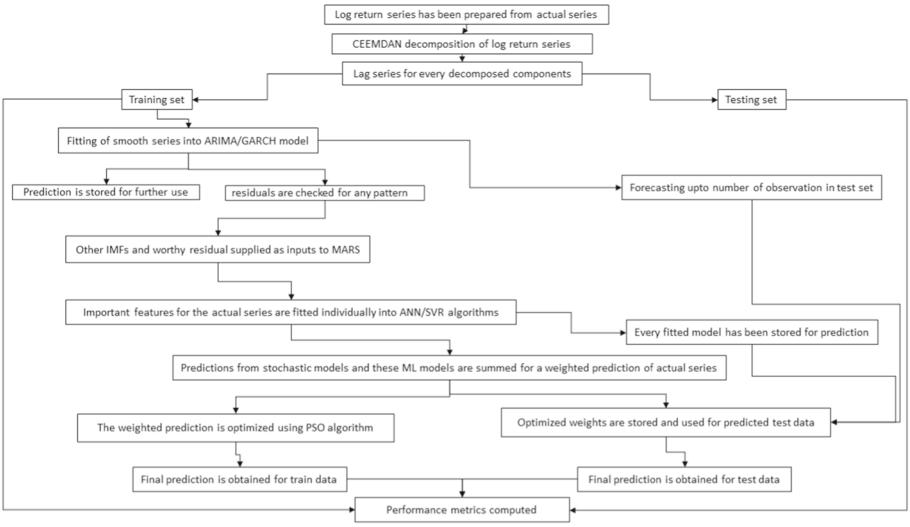

Fig. 4 Step-by-step procedure for the proposed methodology

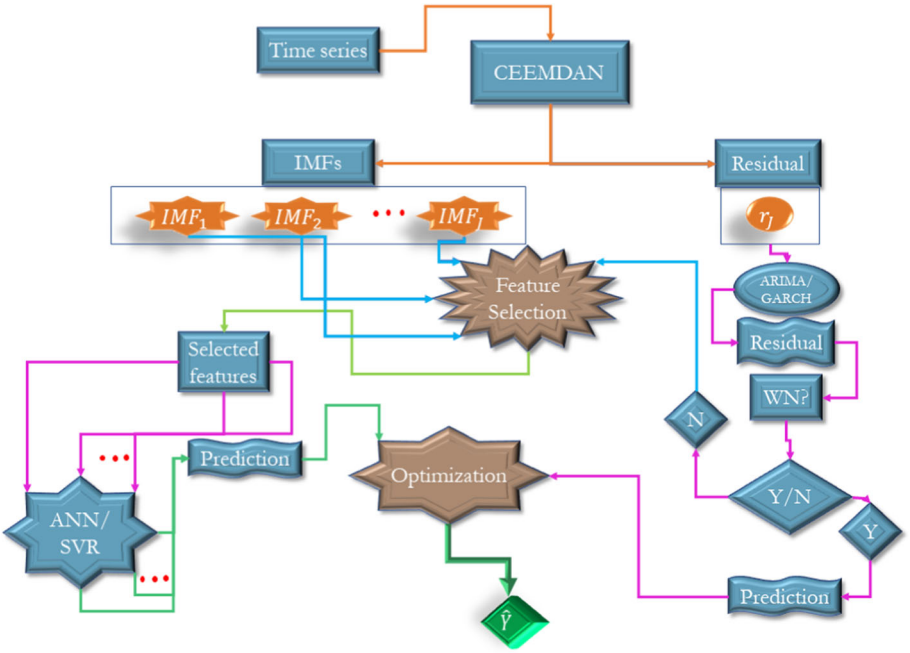

Fig. 5 Flowchart for CARIGAAN or CARIGAS hybrid models

vi. Use MARS algorithm to perform feature selection on the training IMFs (from step 3) and residuals (from step 5) and prepare lag series for the selected features.
vii. Utilize the selected features for prediction using ANN and SVR algorithms separately on the training sets.
viii. Store the fitted model for every selected feature and predict the training and testing data using the prepared data frames from step vi.
ix. Combine and optimize the predictions from the ANN and SVR models using the PSO algorithm.
x. Back-transform the predictions to obtain the predictions for the actual training and testing sets.

The performance of the algorithm is evaluated using various statistical measures such as RMSE, RRMSE, MAE, MAPE, MASE, E_{NS}, WI, and E_{LM}. The effectiveness of the proposed algorithm is compared to benchmark models, such as ARIMA/GARCH (ARIGA), ANN, and SVR. The proposed algorithm and the benchmark models are compared based on their prediction performance. The flowchart of the proposed algorithm is shown in Fig. 5.

3 Performance Measures

To confirm the significance of the performance of the newly developed algorithms and establish whether they are capable for providing relevant and accurate prediction, a number of statistical indicators are used, as a single metric of evaluation cannot determine the benefit and feebleness of a model.

3.1 RMSE

Root mean squared error (RMSE) is formulated in Eq. 16

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2} \tag{16}$$

The RMSE value is expected to be lesser to prove the efficiency of the newly developed model. When the prediction error is following normal distribution RMSE will be a more useful metric for the evaluation of the models. In other cases, the relative alternative measures of accuracy like RRMSE and MAPE will be more fruitful to evaluate model accuracy.

3.2 RRMSE

$$RRMSE = 100 * \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2}}{\overline{Y}} \tag{17}$$

In Eq. 17, representing RRMSE, $\overline{Y}$ is the mean of the actual observations used for prediction. $RRMSE < 10\%$ indicates the model's performance is excellent, in case of $10\% < RRMSE < 20\%$, the model is considered as good, fair if it provides $20\% < RRMSE < 30\%$ and it will be considered poor if the $RRMSE$ value given by the prediction of the model is more than 30% [63].

3.3 MAE

MAE is expressed with the given equation (Eq. 18).

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |e_t| \tag{18}$$

MAE takes the average of absolute errors of prediction and lesser is considered as better.

3.4 MAPE

MAPE is expressed in Eq. 19.

$$MAPE = \left(\frac{1}{N}\sum_{t=1}^{N} |e_t|/y_t\right) * 100\% \tag{19}$$

The more it is the less accurate prediction the model gives.

3.5 MASE

A modified version of MAPE is represented as MASE in Eq. 20.

$$MASE = \frac{N-1}{N}\frac{\sum_{t=1}^{N}|e_t|}{\sum_{t=2}^{N}|y_t - y_{t-1}|} \tag{20}$$

3.6 NSE (E_{NS})

$$E_{NS} = 1 - \left[\frac{\sum_{t=1}^{N} e_t^2}{\sum_{t=1}^{N} \left(Y_t - \overline{Y}\right)^2} \right]; -\infty \leq E_{NS} \leq 1 \tag{21}$$

When E_{NS} value (Eq. 21) achieved from a model's performance in prediction as or near 1, it is considered as an outstanding model in its domain [64].

3.7 WI

$$WI = 1 - \left[\frac{\sum_{t=1}^{N} e_t^2}{\sum_{t=1}^{N} \left(\left|Y_t - \overline{Y}\right| + \left|\widehat{Y}_t - \overline{Y}\right|\right)^2} \right]; 0 \leq WI \leq 1 \tag{22}$$

WI value (Eq. 22) of or close to 1 is considered as the best performing model [64].

3.8 E_{LM}

Legates and McCabe Jr [65] has updated the WI index as E_{LM}. This index (E_{LM}) represented in Eq. 23 provides greater accuracy than classical WI when relatively large values are predicted due to the squaring of error term [66–68].

$$E_{LM} = 1 - \left[\frac{\sum_{t=1}^{N} |e_t|}{\sum_{t=1}^{N} \left|Y_t - \overline{Y}\right|} \right]; -\infty \leq E_{LM} \leq 1 \tag{23}$$

$e_t = y_t - \widehat{y}_t$, is the error of prediction, y_t and $\widehat{y}_t$ are actual and predicted values of the original time series at t^{th} time respectively.

4 Results and Discussion

The daily wholesale prices of potato (in Rupees per quintal) from six major Indian markets have been collected from the Agricultural Marketing Information System (AGMARKNET) website (https://agmarknet.gov.in/) during period January 1, 2011 to December 31, 2022. The daily data sets have been transformed into weekly data before proceeding for analysis.

4.1 Characteristics of Data

Figure 6 and Table 1 display line plot and descriptive statistics of weekly price series of potatoes. Table 1 reveals that the minimum price was recorded at the Ahmedabad market, while the maximum price was observed in Kolkata. The lowest price was recorded in January 2012, and the highest price was observed in January 2021. Bengaluru has the highest average potato price, followed by Mumbai. The Agra market has the lowest average potato price.

Kolkata has the highest standard deviation (SD) of 606.485, while Mumbai has the lowest. Agra, Ahmedabad, and Mumbai have similar SD values. In contrast, the coefficient of variation (CV%) indicates that Agra, Ahmedabad, Delhi, and Kolkata markets are highly volatile. On the other hand, Bengaluru and Mumbai markets show lower volatility, with CV% values below 40%. All price series show positive skewness and leptokurtosis. The Shapiro–Wilk

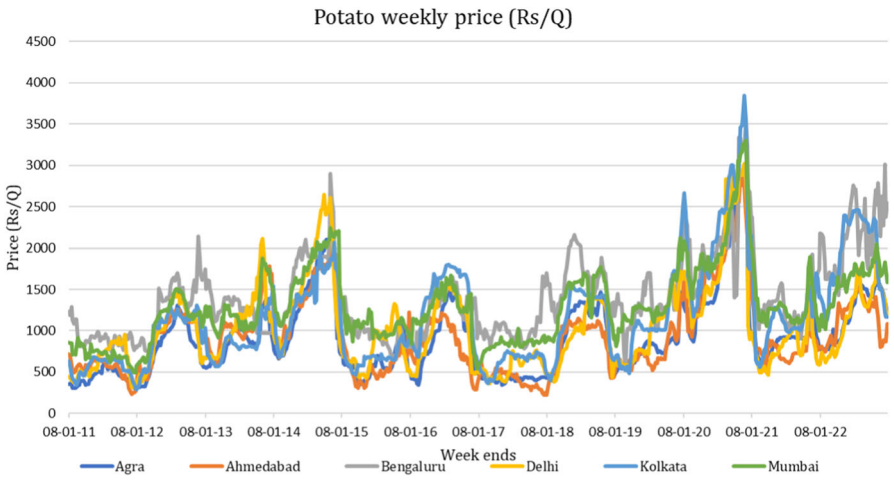

Fig. 6 Line chart of potato weekly price series

Table 1 Descriptive statistics of weekly potato wholesale price series of different markets

Statistics	Agra	Ahmedabad	Bengaluru	Delhi	Kolkata	Mumbai
Minimum	304.286	222.143	528.571	312.857	290	492.857
Maximum	2931.429	2885.714	3550	3015.714	3845.714	3300
Mean	916.866	929.765	1412.804	1032.581	1169.714	1302.557
SD	475.287	467.29	503.116	520.908	606.485	459.679
CV (%)	51.838	50.259	35.611	50.447	51.849	35.29
Skewness	1.291	1.17	1.109	1.3	1.168	1.1
Kurtosis	2.239	1.704	1.399	1.949	1.508	2.158
Shapiro–Wilk	0.9***	0.921***	0.923***	0.895***	0.909***	0.937***

***Indicates significance of values at 1% level

test has been conducted on each dataset, indicating non-normality for all data series. These results provide insights into the nature of potato price series in different Indian markets, which can be useful for developing model strategies.

In Table 2, the KPSS test indicates that all prices are stationary with truncation lag parameter 6. The non-linearity of all the price series has been checked using the Broock–Dechert–Scheinkman (BDS) test [69]. The null hypothesis of the BDS test is that the series is independently and identically distributed (i.i.d.). The BDS test statistics have been calculated for embedding dimensions of 2 and 3, and presented in Table 3. It is evident that test statistics for all the series are significant at 1% level of significance. Therefore, it can be concluded that the price series exhibit non-linear behaviour, and simple linear model may not be effective for modelling.

Table 2 Stationarity tests and lag order

Test	Parameters	Agra	Ahmedabad	Bengaluru	Delhi	Kolkata	Mumbai
KPSS	KPSS level	1.219	0.651	2.409	0.519	2.539	1.922
	Truncation Lag parameter	6	6	6	6	6	6
	p-value	0.01	0.018	0.01	0.037	0.01	0.01
	Stationary?	Yes***	Yes***	Yes***	Yes***	Yes***	Yes***

***Indicates significance of values at 1% level

4.2 Modelling

The potato price series from six different markets have been analysed using various models including ARIGA, ANN, SVR, and two proposed hybrid models—CARIGAAN and CARIGAS. The data has been split into a training set and a testing set with a 90:10 ratio, and the performance of each model has been evaluated using several performance metrics. The RMSE, RRMSE, MAE, MAPE and MASE are expected to be lower for an efficient model, while NSE, WI and LME coefficients approaching 1 are considered good.

Table 4 depicts the performance of implemented models for all the data series. The results indicate that for Agra market, CARIGAAN outperforms other models for predicting test data for all the accuracy measure expect WI. The WI value infer the superiority of CARIGAS model. In the case of the Ahmedabad market, the WI value of the CARIGAAN model is like that of CARIGAS. However, all other measurements witness the superiority of CARIGAAN model. Bengaluru market has the mixed interpretation about the CARIGAAN and CARIGAS model. CARIGAAN is found to be best model for Delhi and Kolkata markets whereas CARIGAS found proven its efficiency is Mumbai markets. Overall, the proposed hybrid models—CARIGAAN and CARIGAS—perform better than the other models in predicting potato prices in different markets. Furthermore, these models can capture the non-linear and volatile nature of the price series, which makes them suitable for real-world applications.

To examine potential of the proposed model in multi-step forecast, accuracy of 7 days, 10 days, 15 days and 30 days ahead forecast in terms of MAPE, RMSE and MAE have been computed and presented in Table 5. Regression lines of actual vs predicted value of proposed models have been depicted in Fig. 7.

A perusal of Table 5 indicates that the CARIGAAN and CARIGAS models have lower MAPE, RMSE and MAE values for all multi-step ahead forecast computed from all the data series. In Fig. 7, all regression line exhibited more that 90% R-square value. These results further supported the supremacy of our proposed models (CARIGAAN and CARIGAS).

5 Conclusions and Future Works

This paper proposes two different hybrid models for the efficient handling of volatility in agricultural price series. The proposed models outperform benchmark stochastic and machine learning models in this regard. The use of these hybrid models has significantly increased the efficiency of price series modeling over the benchmark models, with similar performance observed across different price volatilities. The proposed hybrid models are not limited to the application in agricultural data but can be used for other financial series such as stock markets,

Table 3 Test for linearity (BDS test)

Statistics	Embedding dimension	
	2	3
Agra		
eps[1]	178.859***	297.247***
eps[2]	77.355***	91.204***
eps[3]	55.311***	56.616***
eps[4]	45.314***	43.433***
Ahmedabad		
eps[1]	147.96***	242.028***
eps[2]	71.776***	84.787***
eps[3]	52.241***	53.58***
eps[4]	45.953***	44.319***
Bengaluru		
eps[1]	129.172***	204.284***
eps[2]	67.985***	79.802***
eps[3]	9.112***	50.716***
eps[4]	41.428***	40.458***
Delhi		
eps[1]	152.784***	243.662***
eps[2]	76.096***	89.242***
eps[3]	49.512***	50.416***
eps[4]	38.307 ***	36.506***
Kolkata		
eps[1]	157.869***	260.99***
eps[2]	76.866***	91.702***
eps[3]	55.384***	57.269***
eps[4]	45.438***	43.598***
Mumbai		
eps[1]	141.575 ***	235.642 ***
eps[2]	79.639***	94.913***
eps[3]	58.026***	59.930***
eps[4]	48.283***	46.736***

***Indicates significance of values at 1% level

weather, pollution data, etc. Future research will involve the application of the proposed models to simulated data with different volatility levels to determine their performance and improvements over other models. This will provide insight into the effectiveness of the proposed hybrid models in capturing volatility in a series and how their performance varies with different volatility levels.

It would be interesting to explore the use of other machine learning or deep learning models to determine the extent of further efficiency improvements that can be achieved. The proposed models can be further improved by using the Improved CEEMDAN (ICEEMDAN) method instead of CEEMDAN. Additionally, other feature selection and optimization techniques such

Table 4 Comparison of prediction performances of different models in test sets of all markets

Metrics	ARIGA	ANN	SVR	CARIGAAN	CARIGAS
Agra					
RMSE	68.059	62.181	58.984	**34.613**	36.663
RRMSE	5.747	5.251	4.981	**2.923**	3.096
MAE	48.562	44.228	42.237	**27.132**	27.26
MAPE	4.255	3.919	3.703	**2.416**	2.481
MASE	1.014	0.924	0.882	**0.567**	0.569
NSE	0.942	0.952	0.957	**0.985**	0.983
WI	0.986	0.988	0.989	**0.996**	**0.996**
LME	0.811	0.828	0.836	**0.895**	0.894
Ahmedabad					
RMSE	79.253	78.69	76.401	50.01	**48.414**
RRMSE	7.18	7.129	6.922	4.531	**4.386**
MAE	57.066	58.966	57.197	38.722	**37.048**
MAPE	5.432	5.717	5.502	3.839	**3.627**
MASE	1.001	1.034	1.003	0.679	**0.65**
NSE	0.9	0.901	0.907	0.96	**0.963**
WI	0.975	0.975	0.977	**0.99**	**0.99**
LME	0.756	0.748	0.756	0.835	**0.842**
Bengaluru					
RMSE	287.118	311.927	304.93	**210.872**	215.13
RRMSE	14.259	15.491	15.144	**10.472**	10.684
MAE	218.478	241.75	231.982	165.991	**154.117**
MAPE	11.156	12.244	11.827	8.594	**7.844**
MASE	0.989	1.095	1.05	0.752	**0.698**
NSE	0.588	0.514	0.535	**0.778**	0.769
WI	0.89	0.874	0.878	**0.944**	0.94
LME	0.417	0.355	0.381	0.557	**0.589**
Delhi					
RMSE	128.844	123.567	128.937	**74.125**	93.662
RRMSE	11.216	10.757	11.224	**6.453**	8.153
MAE	88.068	78.098	83.858	**54.003**	67.931
MAPE	7.771	6.833	7.384	**4.655**	6.154
MASE	1.007	0.893	0.959	**0.618**	0.777
NSE	0.878	0.888	0.878	**0.96**	0.936
WI	0.969	0.972	0.97	**0.99**	0.984
LME	0.732	0.762	0.745	**0.836**	0.793
Kolkata					
RMSE	132.147	120.421	116.154	**87.007**	96.836

Table 4 (continued)

Metrics	ARIGA	ANN	SVR	CARIGAAN	CARIGAS
RRMSE	7.163	6.528	6.296	**4.716**	5.249
MAE	91.617	86.916	80.259	**66.583**	71.707
MAPE	5.627	5.286	4.893	**4**	4.345
MASE	1.011	0.959	0.886	**0.735**	0.792
NSE	0.918	0.932	0.937	**0.964**	0.956
WI	0.979	0.983	0.984	**0.991**	0.989
LME	0.788	0.799	0.814	**0.846**	0.834
Mumbai					
RMSE	105.155	108.61	107.667	60.856	**56.958**
RRMSE	6.743	6.965	6.904	3.902	**3.652**
MAE	83.615	85.052	83.9	50.214	**45.443**
MAPE	5.529	5.601	5.465	3.261	**2.968**
MASE	0.984	1	0.987	0.591	**0.535**
NSE	0.833	0.822	0.825	0.944	**0.951**
WI	0.957	0.955	0.956	0.986	**0.988**
LME	0.611	0.605	0.61	0.767	**0.789**

Bold values are the best values; ARIGA represents ARIMA/GARCH model

Table 5 Accuracy measures of implemented models for multi-step forecasting

Datasets	Days	ARIGA	ANN	SVR	CARIGAAN	CARIGAS
MAPE						
Agra	7	5.75	6.28	6.04	4.79	4.33
	10	7.74	8.41	6.82	4.53	4.15
	15	6.83	7.14	5.97	3.94	3.76
	30	4.50	4.61	4.21	2.67	2.78
Ahmedabad	7	6.09	8.03	7.33	6.18	6.07
	10	5.14	6.00	5.45	5.04	5.03
	15	5.59	6.44	6.37	4.70	4.35
	30	4.82	4.99	4.88	3.33	3.06
Bengaluru	7	13.81	13.79	14.40	10.39	9.36
	10	16.90	16.87	17.12	16.09	14.30
	15	13.13	12.97	13.35	12.80	11.44
	30	10.15	10.81	11.02	8.42	7.79
Delhi	7	18.42	15.96	16.56	6.49	13.01
	10	17.71	12.86	15.59	5.65	12.65
	15	14.42	11.53	13.40	5.02	10.86
	30	8.02	7.13	7.49	4.48	6.14
Kolkata	7	8.51	8.01	6.88	3.36	4.64

Table 5 (continued)

Datasets	Days	ARIGA	ANN	SVR	CARIGAAN	CARIGAS
	10	9.21	9.36	8.10	4.90	5.20
	15	8.28	9.98	8.09	4.67	6.03
	30	5.30	6.19	5.02	4.07	4.77
Mumbai	7	9.34	10.47	9.65	2.98	2.85
	10	9.30	10.85	9.22	2.95	3.18
	15	9.50	11.05	9.79	3.20	3.61
	30	5.71	6.20	5.82	3.48	3.55
RMSE						
Agra	7	103.83	104.31	98.21	66.41	57.78
	10	117.82	117.83	104.89	61.33	55.61
	15	102.56	99.93	89.57	51.92	48.34
	30	70.42	69.35	65.66	37.21	37.46
Ahmedabad	7	94.75	105.26	104.43	69.69	58.24
	10	81.12	88.34	87.54	59.95	52.42
	15	76.09	81.58	83.19	53.99	46.93
	30	70.97	71.94	69.42	42.17	38.65
Bengaluru	7	271.62	254.97	293.31	189.76	182.44
	10	315.10	302.17	320.18	286.20	263.02
	15	280.04	271.86	285.23	254.30	229.17
	30	249.03	262.26	267.18	201.79	187.18
Delhi	7	253.40	231.53	245.88	99.13	183.61
	10	223.11	195.71	213.56	84.44	161.92
	15	186.82	166.99	180.04	71.32	136.42
	30	123.17	112.43	118.79	61.45	85.72
Kolkata	7	132.95	124.24	120.66	61.71	81.25
	10	175.54	174.22	164.92	98.14	117.07
	15	158.91	173.30	153.63	88.71	124.49
	30	120.46	130.49	114.49	86.33	105.78
Mumbai	7	182.13	217.39	187.77	54.06	53.14
	10	167.86	201.82	172.38	49.62	52.82
	15	150.78	182.00	159.18	47.87	57.50
	30	108.89	122.71	111.66	62.62	64.40
MAE						
Agra	7	70.29	76.82	73.40	57.21	50.38
	10	87.46	95.11	79.10	52.64	47.53
	15	71.70	75.94	63.77	42.44	39.57
	30	48.79	49.66	45.61	28.53	29.05
Ahmedabad	7	58.32	77.07	70.16	57.15	55.44
	10	48.64	57.33	51.97	46.25	45.71
	15	48.93	56.88	55.71	41.41	38.35
	30	50.44	51.03	50.41	33.27	31.00

Table 5 (continued)

Datasets	Days	ARIGA	ANN	SVR	CARIGAAN	CARIGAS
Bengaluru	7	198.06	196.80	206.13	146.28	134.05
	10	237.50	236.86	241.22	230.75	202.21
	15	197.13	193.74	201.09	196.88	173.15
	30	179.42	193.49	195.61	148.19	137.92
Delhi	7	222.98	188.27	208.40	73.86	149.39
	10	193.08	143.53	176.55	60.21	131.76
	15	146.48	116.16	138.04	48.61	104.00
	30	83.71	73.69	79.32	46.85	61.48
Kolkata	7	114.26	109.79	93.54	48.10	63.70
	10	132.19	135.46	116.44	74.62	76.40
	15	118.47	142.67	115.85	69.84	88.02
	30	86.06	98.46	79.75	66.72	78.35
Mumbai	7	151.56	165.19	155.39	47.01	43.58
	10	141.86	161.61	140.81	44.01	45.90
	15	130.14	148.53	132.95	42.81	47.27
	30	82.61	88.35	84.26	51.91	52.53

as principal component analysis, stepwise multiple linear regression, whale optimization, sparrow search algorithm, and farmland fertility algorithm can be used instead of MARS and PSO. A trial-and-error approach can be used to fit different decomposed components or groups of components into different models to determine any improvement in results.

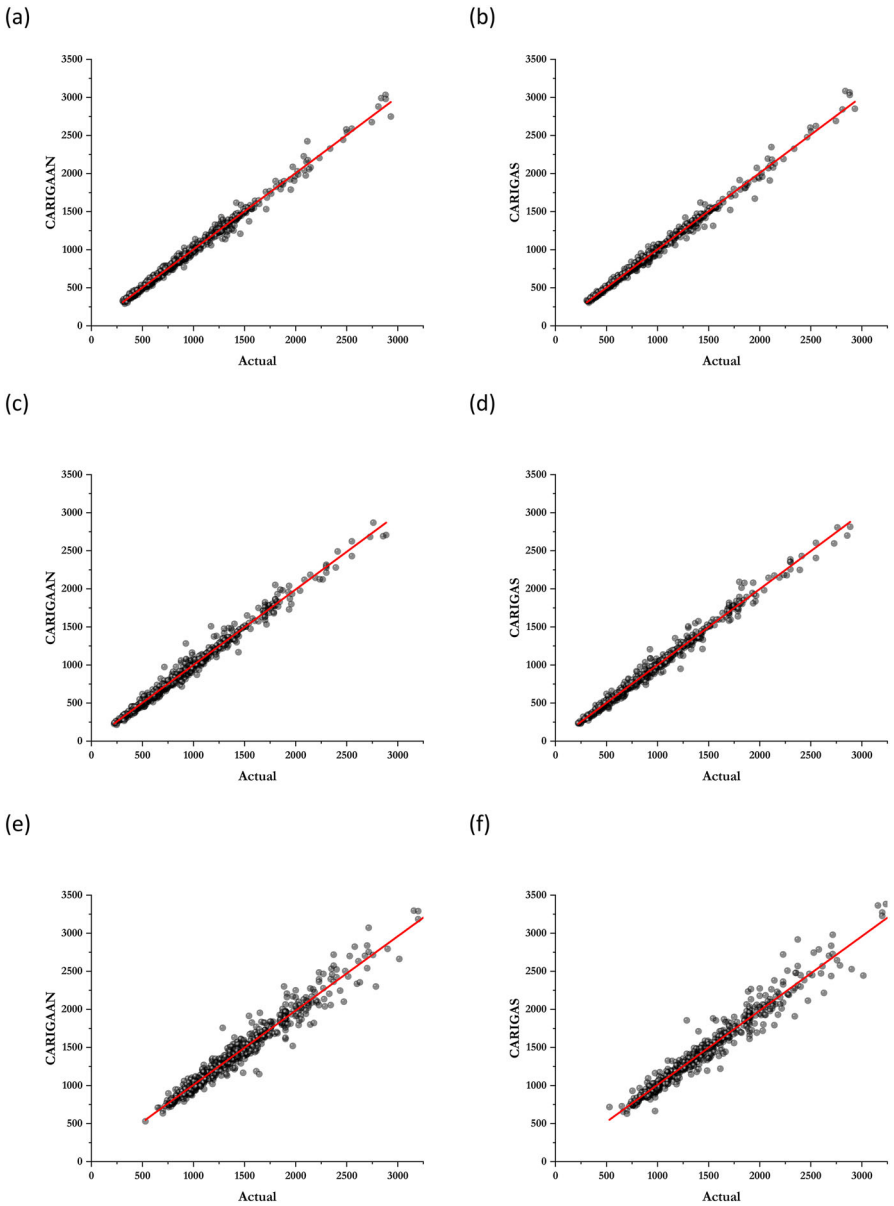

Fig. 7 Actual vs fitted plots with regression line of CARIGAAN and CARIGAS model for different data sets (Agra: a-b; Ahmedabad: c-d; Bengaluru: e–f; Delhi: g-h; Kolkata: i-j and Mumbai: k-l)

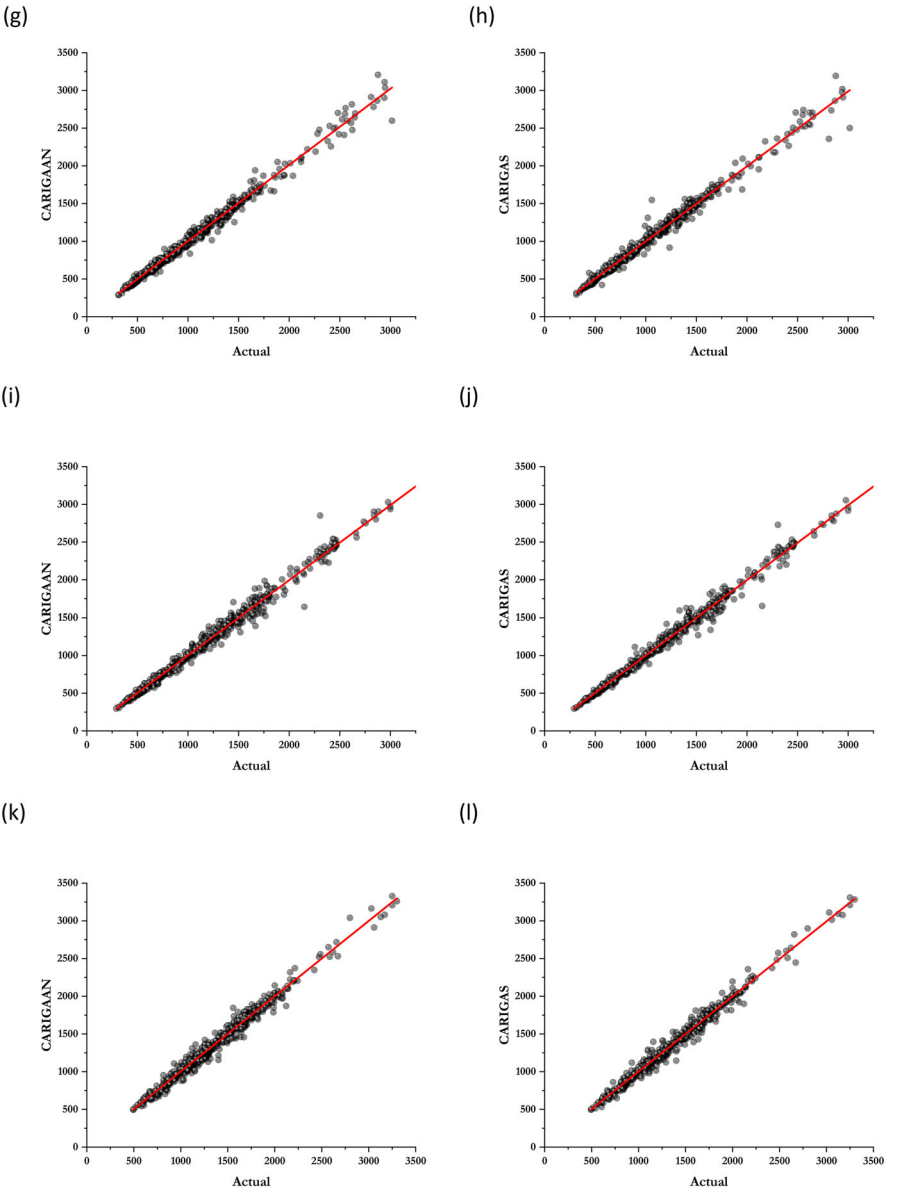

Fig. 7 continued

Acknowledgements The authors express their appreciation for the support of ICAR-IASRI.

Author Contributions: Conceptualization, RKP and SG; formal analysis, MY and SG; investigation, MY and RKP; resources, RKP, MY and AKP.; data curation, SG; writing—original draft preparation, SG; writing—review and editing, SG, MY, and RKP; visualization, SG; supervision, RKP and AKP; All authors have read the manuscript.

Funding: No funding was received for the work.

Declarations

Conflicts of interest The authors declare no competing interests.

References

1. Paul RK, Prajneshu GH (2013) Statistical modelling for forecasting of wheat yield based on weather variables. Indian J Agric Sci 83:180–183
2. Paul RK (2014) Forecasting wholesale price of pigeon pea using long memory time-series models. Agric Econ Res Rev 27:167–176
3. Rakshit D, Paul RK, Panwar S (2021) Asymmetric price volatility of onion in India. Indian J Agric Econ 76:245–260
4. Lee CM, Ko CN (2009) Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm. Neurocomputing 73:449–460. https://doi.org/10.1016/j.neucom.2009.07.005
5. Zhou C, Yin K, Cao Y, Ahmed B (2016) Application of time series analysis and PSO-SVM model in predicting the Bazimen landslide in the Three Gorges Reservoir, China. Eng Geol 204:108–120. https://doi.org/10.1016/j.enggeo.2016.02.009
6. Cherkassky V, Ma Y (2004) Practical selection of SVM parameters and noise estimation for SVM regression. Neural Netw 17:113–126. https://doi.org/10.1016/S0893-6080(03)00169-2
7. Bahrammirzaee A (2010) A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. Neural Comput Appl 19:1165–1195. https://doi.org/10.1007/s00521-010-0362-z
8. Paul RK, Sinha K (2016) Forecasting crop yield: a comparative assessment of ARIMAX and NARX model
9. Bonelli MG, Ferrini M, Manni A (2017) Artificial neural networks to evaluate organic and inorganic contamination in agricultural soils. Chemosphere 186:124–131. https://doi.org/10.1016/j.chemosphere.2017.07.116
10. Adeli H, Jiang X (2006) Dynamic fuzzy wavelet neural network model for structural system identification. J Struct Eng 132:102–111
11. Gu J, Zhu M, Jiang L (2011) Housing price forecasting based on genetic algorithm and support vector machine. Expert Syst Appl 38:3383–3386
12. Gu YH, Yoo SJ, Park CJ et al (2016) BLITE-SVR: New forecasting model for late blight on potato using support-vector regression. Comput Electron Agric 130:169–176
13. Thivakaran TK, Ramesh M (2022) Exploratory data analysis and sales forecasting of bigmart dataset using supervised and ANN algorithms. Meas Sensors 23:100388. https://doi.org/10.1016/j.measen.2022.100388
14. Chen K-H, Chen L-F, Su C-T (2014) A new particle swarm feature selection method for classification. J Intell Inf Syst 42:507–530
15. Chen YT, Sun EW, Lin YB (2020) Machine learning with parallel neural networks for analyzing and forecasting electricity demand. Comput Econ 56:569–597. https://doi.org/10.1007/s10614-019-09960-5
16. Chen W, Ma C, Ma L (2009) Mining the customer credit using hybrid support vector machine technique. Expert Syst Appl 36:7611–7616
17. Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175
18. Khashei M, Bijari M (2011) A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl Soft Comput 11:2664–2675
19. Adhikari R, Agrawal RK (2014) A combination of artificial neural network and random walk models for financial time series forecasting. Neural Comput Appl 24:1441–1449

20. Fogno Fotso HR, Aloyem Kazé CV, Djuidje Kenmoé G (2021) A novel hybrid model based on weather variables relationships improving applied for wind speed forecasting. Int J Energy Environ Eng 13:1–14
21. Rubio L, Alba K (2022) Forecasting selected Colombian shares using a hybrid ARIMA-SVR model. Mathematics. https://doi.org/10.3390/math10132181
22. Samuels JD, Sekkel RM (2017) Model confidence sets and forecast combination. Int J Forecast 33:48–60
23. Garai S, Paul RK (2023) Development of MCS based-ensemble models using CEEMDAN decomposition and machine intelligence. Intell Syst with Appl 18:200202
24. Hansen PR, Lunde A, Nason JM (2011) The model confidence set. Econometrica 79:453–497
25. Lindsay RW, Percival DB, Da R (1996) The discrete wavelet transform and the scale analysis of the surface properties of sea ice. IEEE Trans Geosci Remote Sens 34:771–787
26. Percival DB, Walden AT (2000) Wavelet methods for time series analysis. Cambridge University Press
27. Percival DB, Mofjeld HO (1997) Analysis of subtidal coastal sea level fluctuations using wavelets. J Am Stat Assoc 92:868–880
28. Paul RK, Garai S (2021) Performance comparison of wavelets-based machine learning technique for forecasting agricultural commodity prices. Soft Comput 25:12857–12873. https://doi.org/10.1007/s00500-021-06087-4
29. Paul RK, Garai S (2022) Wavelets based artificial neural network technique for forecasting agricultural prices. J Indian Soc Probab Stat 23:1–15. https://doi.org/10.1007/s41096-022-00128-3
30. Babu CN, Reddy BE (2014) A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. Appl Soft Comput 23:27–38
31. Torres ME, Colominas MA, Schlotthauer G, Flandrin P (2011) A complete ensemble empirical mode decomposition with adaptive noise. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 4144–4147
32. Karijadi I, Chou S-Y, Dewabharata A (2023) Wind power forecasting based on hybrid CEEMDAN-EWT deep learning method. Renew Energy 218:119357
33. Li X, Li C (2016) Improved CEEMDAN and PSO-SVR modeling for near-infrared noninvasive glucose detection. Comput Math Methods Med 2016:
34. Li K, Huang W, Hu G, Li J (2023) Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. Energy Build 279:112666
35. Ran P, Dong K, Liu X, Wang J (2023) Short-term load forecasting based on CEEMDAN and transformer. Electr Power Syst Res 214:108885. https://doi.org/10.1016/j.epsr.2022.108885
36. Samadi Bonab M, Ghaffari A, Soleimanian Gharehchopogh F, Alemi P (2020) A wrapper-based feature selection for improving performance of intrusion detection systems. Int J Commun Syst 33:1–26. https://doi.org/10.1002/dac.4434
37. Naseri TS, Gharehchopogh FS (2022) A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. J Netw Syst Manag 30:40. https://doi.org/10.1007/s10922-022-09653-9
38. Cook NR, Zee RYL, Ridker PM (2004) Tree and spline based association analysis of gene–gene interaction models for ischemic stroke. Stat Med 23:1439–1453
39. Lee T-S, Chiu C-C, Chou Y-C, Lu C-J (2006) Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Comput Stat Data Anal 50:1113–1130
40. Chang P-C, Fan C-Y (2008) A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting. IEEE Trans Syst Man, Cybern Part C (Appl Rev) 38:802–815
41. Tsai CF, Hsiao YC (2010) Combining multiple feature selection methods for stock prediction: union, intersection, and multi-intersection approaches. Decis Support Syst 50:258–269
42. Kao L-J, Chiu C-C, Lu C-J, Chang C-H (2013) A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting. Decis Support Syst 54:1228–1244
43. Adnan RM, Liang Z, Heddam S et al (2020) Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs. J Hydrol 586:124371
44. Bose A, Hsu C-H, Roy SS et al (2021) Forecasting stock price by hybrid model of cascading multivariate adaptive regression splines and deep neural network. Comput Electr Eng 95:107405
45. Mohammadzadeh H, Gharehchopogh FS (2021) A multi-agent system based for solving high-dimensional optimization problems: a case study on email spam detection. Int J Commun Syst 34:1–48. https://doi.org/10.1002/dac.4670
46. Ghafori S, Gharehchopogh FS (2022) Advances in spotted hyena optimizer: a comprehensive survey. Arch Comput Methods Eng 29:1569–1590. https://doi.org/10.1007/s11831-021-09624-4
47. Gharehchopogh FS (2022) Quantum-inspired metaheuristic algorithms: comprehensive survey and classification. Artif Intell Rev. https://doi.org/10.1007/s10462-022-10280-8

48. Gharehchopogh FS, Nadimi-Shahraki MH, Barshandeh S et al (2023) CQFFA: a chaotic quasi-oppositional farmland fertility algorithm for solving engineering optimization problems. J Bionic Eng 20:158–183. https://doi.org/10.1007/s42235-022-00255-4
49. Zhao L, Yang Y (2009) PSO-based single multiplicative neuron model for time series prediction. Expert Syst Appl 36:2805–2812. https://doi.org/10.1016/j.eswa.2008.01.061
50. Behnamian J, Fatemi Ghomi SMT (2010) Development of a PSO-SA hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. Expert Syst Appl 37:974–984. https://doi.org/10.1016/j.eswa.2009.05.079
51. Heidari AA, Akhoondzadeh M, Chen H (2022) A wavelet PM2.5 prediction system using optimized kernel extreme learning with Boruta-XGBoost feature selection. Mathematics. https://doi.org/10.3390/math10193566
52. Wu Z, Huang NE (2009) Ensemble empirical mode decomposition: a noise-assisted data analysis method. Adv Adapt Data Anal 1:1–41. https://doi.org/10.1142/S1793536909000047
53. Wu Z, Huang NE (2004) A study of the characteristics of white noise using the empirical mode decomposition method. Proc R Soc Lond Ser A Math Phys Eng Sci 460:1597–1611
54. Box GEP, Jenkins MG, Jenkins GM (1970) Time series analysis: forecasting and control. Holdan-Day, San Francisco
55. Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. J Econom 31:307–327
56. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
57. Werbos PJ (1988) Generalization of backpropagation with application to a recurrent gas market model. Neural Netw 1:339–356
58. Werbos P (1974) Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Comm Appl Math Harvard Univ Cambridge, MA
59. Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19:1–67
60. Kenny J (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks. pp 1942–8
61. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. pp 1942–1948
62. Alam MN (2016) Particle swarm optimization: algorithm and its codes in matlab. ResearchGate 8:10
63. Mohammadi K, Shamshirband S, Anisi MH et al (2015) Support vector regression based prediction of global solar radiation on a horizontal surface. Energy Convers Manag 91:433–441
64. Deo RC, Wen X, Qi F (2016) A wavelet-coupled support vector machine model for forecasting global incident solar radiation using limited meteorological dataset. Appl Energy 168:568–593
65. Legates DR, McCabe GJ Jr (1999) Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resour Res 35:233–241
66. Willmott CJ (1981) On the validation of models. Phys Geogr 2:184–194
67. Willmott CJ (1984) On the evaluation of model performance in physical geography. Spat Stat Model 443–460
68. Legates DR, McCabe GJ (2013) A refined index of model performance: a rejoinder. Int J Climatol 33:1053–1056
69. Broock WA, Scheinkman JA, Dechert WD, LeBaron B (1996) A test for independence based on the correlation dimension. Econom Rev 15:197–235