

Data driven algorithm selection to predict agriculture commodities price

Girish Hegde¹, Vishwanath R. Hulipalled¹, Jay B. Simha²

¹School of Computing and Information Technology, REVA University, Bangalore, India

²Abiba Systems, CTO, and RACE Labs, REVA University, Bangalore, India

Article Info

Article history:

Received Oct 6, 2022

Revised Dec 19, 2022

Accepted Dec 21, 2022

Keywords:

Agriculture

Ensemble model

Machine learning

Price forecasting

Seasonal autoregressive

integrated moving average

ABSTRACT

Price prediction and forecasting are common in the agriculture sector. The previous research shows that the advancement in prediction and forecasting algorithms will help farmers to get a better return for their produce. The selection of the best fitting algorithm for the given data set and the commodity is crucial. The historical experimental results show that the performance of the algorithms varies with the input data. Our main objective was to develop a model in which the best-performing prediction algorithm gets selected for the given data set. For the experiment, we have used seasonal autoregressive integrated moving average (SARIMA) stack ensemble and gradient boosting algorithms for the commodities Tomato and Potato with monthly and weekly average prices. The experimental results show that no algorithm is consistent with the given commodities and price data. Using the proposed model for the monthly forecasting and Tomato, stack ensemble is a better choice for Karnataka and Madhya Pradesh states with 59% and 61% accuracy. For Potatoes with the monthly price for Karnataka and Maharashtra, the stack ensemble model gave 60% and 85% accuracy. For weekly prediction, the accuracy of gradient boosting is better compared to other models.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Girish Hegde

School of Computing and Information Technology, REVA University

Rukmini Knowledge Park, Kattigenahalli, Srinivasa Nagar, Yelahanka, Bangalore-560064, India

Email: girishhegde37@gmail.com

1. INTRODUCTION

Agriculture is the backbone of most of the countries in this world. In the fastest-growing or developing countries such as India, the majority of the workforce is involved in agriculture. In India, approximately two-thirds of the population depends on agriculture. The contribution to Indian gross domestic product (GDP) from the agriculture sector is about 28% [1], and the contribution to exports is about 15%. However, still, farmers are not getting the expected return. For sustainable food security, it is important to increase the farmer's income or return on investment. Farmers should get the proper price for their produce.

Recent advancements in technology and communication are promising for farmers to get a better return for their produce. Artificial intelligence (AI), machine learning (ML), internet of things (IoT), and natural language processing (NLP) technology can be applied in smart farming to increase the farmers' income [2]. Prediction and forecasting of the price will help farmers to make informed decisions. By knowing the forecasted price, farmers can decide about harvesting, selling, and storing the commodities for a certain time. From the earlier research, we can see that many of the researchers used different prediction algorithms for different agriculture commodities. Each algorithm's accuracy is different for the same commodity or for different commodities.

The AI and machine learning techniques are used in various applications such as weather forecasting [3], prediction of earthquakes [4], cyber security [5], prediction of crimes [6], smart irrigation [7], stock price prediction [8], [9], electricity consumption forecasting [10], [11] and many more. The advancement in the development of natural language processing systems [12], [13] and speech recognition systems helped a lot to increase the farmer's income and agricultural commodities production. For predicting the price of the commodities different methods are used, time series algorithms, regression techniques and machine learning algorithms. Each method performs well for a specific set of data. It is difficult to generalize the model or select the generic algorithm, which performs well for different datasets and commodities.

Over the past few decades, research has been carried out to predict and forecast agricultural commodities prices. The researchers used different algorithms and different parameters to get better performance. Still, the research gap is to select the best performing algorithm for the input data and the parameters dynamically. Some algorithms are good for long-term forecasting, and few are good for short-term forecasting [14]. In this paper, the authors used autoregressive integrated moving average (ARIMA), back propagation network (BP network), and recurrent neural network (RNN). With these algorithms, they tried to forecast the monthly, weekly, and daily average prices for cucumbers. The results show that RNN gave higher accuracy than ARIMA and BP networks. The accuracy of ARIMA is increased for long-term forecasting.

When the price variation is having a high influence on seasonality, then it is good to use the seasonal ARIMA model [15]. In this research paper, we can see that the authors used the seasonal ARIMA (SARIMA) model to forecast the monthly price of tomatoes for major tomato-producing states of India. Even with the same algorithm used, the forecasting performance is different for different states. The author got a 28.8% error for Madhya Pradesh and a 47.7% error for Andhra Pradesh. This indicates that even with the same commodity, the performance may vary with input data.

Wang *et al.* [16], provided the recent trend and advancements in agricultural product price forecasting methods. The authors mentioned the advantages and disadvantages of traditional and intelligent forecasting methods. The authors also highlighted that performance of the hybrid model is better than that of the single model. Many factors affect the price variation of commodities, and it is important to consider the same during forecasting. The most popular seasonal time series model used for forecasting is SARIMA [17]. However, the performance of the algorithm is poor if the data is too nonlinear.

An ensemble model of learning is one of the advanced machine learning techniques used for price prediction and forecasting [18]. With the ensemble model, the prediction accuracy will improve. The authors performed an extensive comparative study with different ensemble techniques such as bagging, boosting, blending, and stacking. In many cases, it is good to consider a group of prediction models than the single one to get better performance [19]. According to Wolpert [20], stacking prediction algorithms will give better performance than using a single prediction algorithm.

The advancement of the recent development in artificial neural networks (ANN) and other advanced algorithms like long short term memory (LSTM), convolutional neural networks (CNN) [21], back propagation neural network (BPNN) [22], [23], and support vector machine (SVM) [24], are very promising to increase the prediction accuracy. The intelligent algorithms will give better performance with nonlinear data also, but one of the problems with the neural network model is the slow convergence speed. If there is a good price predictor, then farmers can decide which crop to grow in advance [25]. Depending on the forecast price and the government announced minimum support price, farmers can decide which crop to grow and when to sell. The author's used a decision tree supervised algorithm to predict the price of different commodities.

The agriculture sector still lacks technological advancement. In India, farmers are not using the latest available techniques [26]. With the help of machine learning techniques, farmers can get benefitted from the best price for their crops. Using ANNs, it is possible to predict weather conditions, soil characteristics, market demand and supply, and future prices. Predicting the price of commodities is an important problem in agriculture [27]. The accuracy and generality are the important aspects while predicting the price. In this paper, the authors used the datasets like the total area of the crop planted, and harvested. They used different machine learning algorithms like logistic regression, XGBoost, and neural network and mentioned that XGBoost gave them better accuracy.

Farmers play an important role in a country's agricultural development [28]. The crop price fluctuation will affect the country's GDP. Before planting the crop, if farmers can get information about price variations, then they can make an informed decision to reduce the loss. The authors used the wholesale price index (WPI) and previous rainfall data to predict the price using a decision tree supervised machine learning algorithm. From the historical research, we can see that the researchers used different algorithms for a variety of agriculture commodity price predictions. Table 1 shows a few of them.

From Table 1, we can see that, 3 different forecasting algorithms were used for tomato, 2 different algorithms for potato and 3 different algorithms for onion. Even though the commodity is the same, for the different sets of data, the performance of the algorithm is different. Reddy [15] predicted the monthly average price of tomato for 11 years from January 2006 to December 2016 and got the highest accuracy of 82% for the state Madhya Pradesh with SARIMA. In this study, we got 59% accuracy with the 10 years of data from January 2011 to December 2020 for the same commodity and SARIMA model.

From the related work or historical research, we can see that for the same commodity different prediction algorithms are used and the performance of the same algorithms varies with the data used. It is difficult to choose the best performing algorithm for the given data set. Since the same algorithm gives different accuracy for different data sets and time periods. The major research gap is, that no model selects the best-fit prediction algorithm for the input price data dynamically. To address this issue, in this study we proposed a flexible and data-driven algorithm selection model, which selects the best performing algorithm dynamically. With this user no need to run the individual prediction algorithms for their price data set.

The main significance of this study aims to find a best-fitting algorithm for predicting the future prices of commodities near the real value (price), based on the input data dynamically. In this study, we proposed an algorithm which selects the forecasting model for the given commodity and price data. The algorithm is flexible such that any new algorithm can be added to the set and that will be used for dynamic algorithm selection.

Table 1. The algorithms used for price forecasting

Commodity	Authors	Prediction Algorithms
Tomato	Zhang <i>et al.</i> [29]	Wavelet neural network
	Reddy [15]	SARIMA
	Adanacioglu and Yercan [30]	SARIMA
	Boateng <i>et al.</i> [31]	ARIMA
	Ansari and Ahmed [32]	ARIMA
Tea	Assis <i>et al.</i> [33]	ARIMA, GARCH, mixed ARIMA/GARCH
Coco Bean	Darekar and Reddy [34]	ARIMA
Pigeon pea	Darekar and Reddy [35]	ARIMA
Cotton	Xiong <i>et al.</i> [36]	VECM-MSVR
	Zhemmin <i>et al.</i> [37]	Dynamic chaotic neural network
Potato	Dipankar <i>et al.</i> [38]	ARFIMA-FIGARCH
Onion	Areef [39]	ARIMA
	Nalini <i>et al.</i> [40]	ARIMA, LASSO
Corn	Wang <i>et al.</i> [41]	SSA ELM
	Shahhosseini <i>et al.</i> [42]	Stacked LASSO
Cucumber	Xiong <i>et al.</i> [43]	Hybrid STL, ELM
	Weng <i>et al.</i> [14]	ARIMA, back propagation neural network, RNN

Note: generalized autoregressive conditional heteroskedasticity (GARCH), vector error correction model and multi-output support vector regression (VECM-MSVR), autoregressive fractionally integrated moving average and fractionally integrated generalized autoregressive conditional heteroscedastic (ARFIMA-FIGARCH), least absolute shrinkage and selection operator (LASSO), seasonal trend decomposition based on loess (STL), and extreme learning machines (ELM).

2. METHOD AND PROPOSED ALGORITHM

2.1. Dynamic algorithm selection

In the past few decades, different kinds of algorithms are used to predict and forecast agriculture commodity prices. Historical evidence or research shows that users should not rely on a single algorithm for different commodities. The selection of an algorithm for the given commodity is not an easy task. When a new commodity is given, it is difficult to select the optimal algorithm for forecasting. Experts can compare the performance of the different algorithms and select the best fitting algorithm. However, training different models each time and finding the optimal algorithm is a tedious job.

It is important to select the forecasting algorithm based on the given input data and the forecasting duration like monthly, and weekly. In this study, we propose a runtime forecasting algorithm selection model based on the commodity price data and forecasting duration. Figure 1 shows the flow of the proposed dynamic algorithm selection model.

Price data-In the first phase, we will get the historical price data for the commodities. We have gathered the average price of tomato, potato, and onion for monthly and weekly forecasting. For monthly forecasting, we have collected 10 years of historical average price from January 2011 to December 2020 and for weekly forecasting, we have collected 5 years of average price from January 2016 to December 2020 from the agriculture marketing (AGMARKNET) portal. This data is regularly updated and owned by the Government of India.

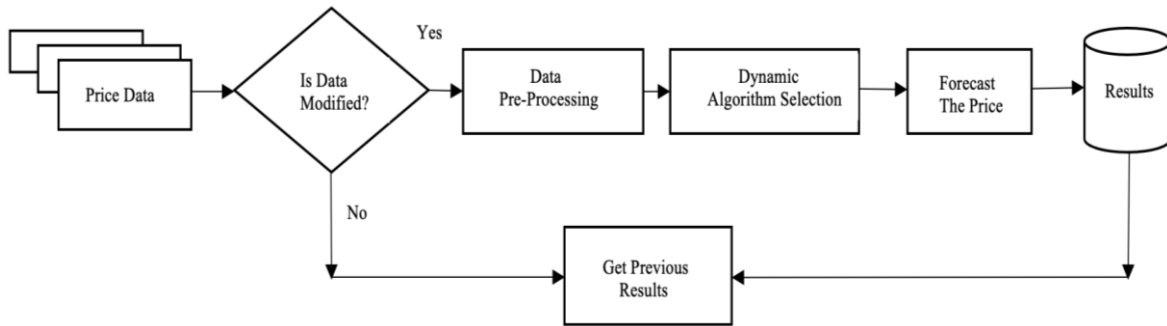


Figure 1. Dynamic algorithm selection process

Is data modified-Training and fitting different machine learning models is a time-consuming process. Most of the time, it is not required to train and finding a suitable model for forecasting is not required. For example, in the case of monthly average price data, the data will change only once a month. Similarly, with weekly price data, the data is constant over a week or changes only once a week. If the data is not changing frequently, then we can take advantage of transfer learning or fetch the previously forecasted result. In a real-world scenario, there may be multiple queries related to price from the farmers and it is required to run the model for each query. If the queries are related to the same commodity, like tomato, and period, monthly, and no change in the input price data, then it is not required to train the different models and select the optimum model to predict and forecast the price. In this phase, we are checking whether the input data is modified and if the answer is not, then directly take the previous result. For the first time, always the flow goes to pre-processing, training the different models, and selecting the optimum model for forecasting the price.

Data pre-processing-in the data processing or the pre-processing stage, we verified if there is any missing data, and if any, we must fill the same with the processed value. We have used linear interpolation logic to fill in the missing data. This logic works as (1).

$$price(t) = \frac{t-t_1}{t_0-t_1} * price(t_0) + \frac{t-t_0}{t_1-t_0} * price(t_1) \quad (1)$$

where, t is the date on which price data is missing, t_0 is the immediate previous date and t_1 is the immediate after date, price (t_0) and price (t_1) is the known price before and after the missing price data on the date.

Dynamic algorithm selection-once the data gets processed, the proposed dynamic algorithm logic will run. In this stage, different machine learning algorithms get trained and predict the price for the given data. Later comparing the performance of the algorithms, selecting the best-fitted forecasting algorithm with better performance or less error.

Forecast the price-this is the final stage of the model or the algorithm. In the previous stages with the pre-preprocessing and the dynamic algorithm selection, the best performing forecasting algorithm decided. In the final stage, the price forecast will happen for the given commodity and data using the selected algorithm. This gives the forecast price near to the real value. For example, stacking used for tomato price forecasting for Karnataka, and the results get stored.

2.2. Dynamic algorithm selection

The algorithm 1 gives the details of dynamic forecasting algorithm selection. The proposed algorithm takes the price data set id as the input parameter and generates the forecasted price as the output or result. The price data of different commodities can be stored in a shared location and assigned a unique identifier. For example, the monthly tomato price for Karnataka has id ktmid1, for Maharashtra, and it is mtmid1. Depending on the requirement, like predicting the monthly price or weekly price, pass the corresponding dataset id to the algorithm as input. Similarly, the prediction algorithms are stored in an array, in this case, ARIMA, SARIMA, stacking and gradient boosting are stored in an array of algorithms. The error is initialized with 100 (maximum percentage).

Let D contains the price data set for different commodities.

$$D = \{pd_1, pd_2, \dots, pd_n\} \quad (2)$$

where, pd_1, pd_2, \dots, pd_n are the data set ids for the commodities.

Algorithm 1. Dynamic forecasting algorithm selection

```

function select Forecast Algorithm (datasetId)
{
    input: datasetId // Each dataset assigned with unique Id
    output: forecasted price
    var algorithms [] = {ARIMA, SARIMA, Stacking, Gradient Boosting}
    var error = 100
    var final = 0
    dataset = readData(datasetId) // Read the data /tmp/tomato_weekly.csv
    var updated = isUpdated (datasetId) // Check whether data got modified
    if (! updated)
    {
        var result = readResult(datasetId)
        if(result)
            return result
    }
    for i=0; i< len(algorithms); i++
    {
        var prederror = runPrediction (dataset, algorithms[i])
        if (error > prederror)
        {
            error = prederror
            final = i
        }
    }
    var fprice = forecast (dataset, algorithms[final], period)
    store (datasetId, fprice)
    return fprice
}
// This is the actual prediction algorithm, like SARIMA to be used to predict the price.
// Just the steps given, not the actual implementation of the algorithm
function runPrediction (priceData, predAlgorithm)
{
    size = int(len(priceData) * 0.8)
    // Partition the dataset in to training set and testing set
    train = priceData [0: size]
    test = priceData [size:]
    // Train the algorithm with training data set
    fit1 = predAlgorithm(train).fit()
    // Predict the price using testing data set
    var predict_price = fit1.predict(len(test))
    // Calculate accuracy
    error = mean (abs (test - predict_price)/abs(test))* 100
    return error
}
// Price forecast using finally selected algorithm
function forecast (dataset, algorithm, period)
{
    fprice = forecast the price using dynamically selected algorithm
    return fprice
}

```

Let A is the set containing n number of prediction algorithms. In our study, it is ARIMA, SARIMA, stacking and gradient boosting.

$$A = \{a_1, a_2, \dots, a_n\} \quad (3)$$

When the dynamic algorithm selection model invoked with dataset id, based on the identifier, find whether the data got modified considering the updated date and the last result date. If the data is not modified, then no need to execute the whole model and train the different predictions. This will save the training of different prediction algorithms as well as the execution time of algorithms and finds the best fit. Instead of that just retrieve the result of the previous run for the given data. For example, if ten queries come in a day to get the forecasted monthly price for tomatoes, it is not required to run this algorithm ten times. Only once execute the dynamic algorithm selection model, train the given set of prediction algorithms, find the best fit for the given data set and store the forecasted price. For the next nine queries, the same forecasted result can be used. This greatly reduces the time required to train the prediction algorithms and forecast the price. In our study, if the data set given is the monthly price of potatoes for Maharashtra, then, in the first run the data set is used to train SARIMA, stacking and gradient boosting algorithms and the final prediction is carried out with gradient boosting and predicted values are stored. In the subsequent execution, if the price data is not modified, the model will fetch the stored predicted.

Let pdi be the dataset used for forecasting then, select the prediction algorithms from set A and find the prediction accuracy for each algorithm.

$$accuracy(ac_j) = predict(pdi, A_j) \quad (4)$$

where, $j=1$ to n and $0 < i < n$

In the first run, the model will invoke individual prediction algorithms with the given price data. The price data set is divided into 80% for the training set and 20% for the data considered for the testing set. Each prediction algorithm is trained separately and predicts the price and finds the accuracy. Later, compare the error value or accuracy of the individual algorithms and the model selects the prediction algorithm with better accuracy and final forecasting happens.

Let ac_1, ac_2, \dots, ac_n are the prediction accuracy of the different algorithms for the given data set pdi , then,

$$ac = highest(ac_i) \quad (5)$$

$$al = A[i] \quad (6)$$

where, ac is highest accuracy. al is final prediction algorithm selected from set of algorithms A . $i=1$ to n . Finally forecast the price using the dynamically selected algorithm with the highest accuracy.

$$fprice = forecast(pdi, Ai) \quad (7)$$

The above proposed data-driven algorithm selection logic is very flexible. At any point in time, new algorithms can be added, and still, the final selection is based on prediction accuracy. The performance of the algorithm is not static, if the data gets changed and the error parameter MAPE gets changed, then the best-performed algorithm will get selected for forecasting. This proposed algorithm is flexible and independent of data, forecasting period, and type of commodity. Since, with the first run, the model considers all the given prediction algorithm in the list and predict the price and compare the accuracy, the execution complexity or time complexity is $O(n)$, where n is the number of prediction algorithms considered. The final forecasting happens with the best-fitted algorithm and the complexity is $O(1)$.

3. RESULTS AND DISCUSSION

3.1. Data set

The monthly average price data for tomatoes and potatoes contains 10 years of historical average price from January 2011 to December 2020. We have collected weekly average price data for tomatoes and potatoes, for 5 years, from January 2016 to December 2020. Both the monthly and weekly price data are for Karnataka, Maharashtra and Madhya Pradesh, the major producing states in India. With the collected price data 80% of the price data was used as a training set and 20% of the data was considered as a testing set as given in Table 2.

In this study, our goal is to show why dynamic algorithm selection is important based on the input price data, and we have proposed an algorithm which selects a forecasting algorithm based on the price data. For this purpose, we have considered the traditional forecasting method seasonal ARIMA (SARIMA), and two advanced techniques, stacking and gradient boosting.

Table 2. Training set and testing data used by monthly and weekly prediction

Temporal scale	Training Data	Testing Data
Monthly	January 2011 to December 2018	January 2019 to December 2020
Weekly	January 2016 to December 2019	January 2020 to December 2020

3.2. Seasonal ARIMA (SARIMA)

This is an extension of the ARIMA model with the additional seasonal component. With SARIMA, the effect of seasonality will be considered [44]. The SARIMA model can be represented [45] as,

$$\begin{array}{ccc} \text{ARIMA} & (p, d, q) & (P, D, Q) m \\ & \uparrow & \uparrow \\ & \text{Non-seasonal part} & \text{Seasonal part} \end{array}$$

where, trend components: p is autoregression order, d is differencing component, and q is moving average value. Seasonal components: P is AR component with seasonality, D is differencing component with seasonality, Q is MA component with seasonality, and m is the time steps for a single seasonal period.

We can represent the SARIMA model mathematically as (8).

$$\Phi_p(B^s)\phi(B)\nabla_s^D\nabla^D X_t = \theta_Q(B^s)\theta(B)w_t \tag{8}$$

For example, SARIMA (1,0,4) (2,0,2)12 can be expressed as (9).

$$(1-\phi_1B)(1-\Phi_1B_{12}-\Phi_1B_{12}-\Phi_2B_{24})x_t = \theta_0 + (1-\theta_1B-\theta_2B_2-\dots-\theta_4B_4(1-\theta_1B_{12}-\theta_2B_{24}))w_t \tag{9}$$

$$\begin{aligned} x_t = & \theta_0 + \phi_1x_t - 1 + \Phi_1x_t - 12 - \phi_1\Phi_1x_t - 13 + \\ & \Phi_2x_t - 24 - \phi_1\Phi_2x_t - 25 + w_t - \theta_1w_t - 1 - \dots - \theta_4w_t - 4 - \theta_1w_t - 12 + \\ & \theta_1\theta_1w_t - 13 + \dots + \theta_4\theta_1w_t - 16 - \\ & \theta_2w_t - 24 + \theta_1\theta_2w_t - 15 + \dots + \theta_4\theta_2w_t - 28 \end{aligned} \tag{10}$$

In the case of SARIMA, the non-seasonal part is the same as ARIMA. The SARIMA model also expects stationary data. Like ARIMA we can make the data stationary by differencing and choosing the value for d . Along with the non-seasonal component, we must consider the seasonal trend and perform the seasonal differencing and set the value for D . The trend components p , q and P , Q are to be chosen by observing the autocorrelation function (ACF) and partial ACF (PACF) graphs such that the value of Apollo intermediate chart (AIC) is minimized.

3.3. Stacking

Sometimes the traditional models may not perform well and need to consider the group of algorithms or ensemble techniques we need to use for prediction. In the case of stacking prediction will happen on two levels. In level 0 multiple weak learners are used called base learners. The level 1 model is called a meta learner. Multiple algorithms fit with the same training dataset at level 0. The result of the base learner is used to train the meta learner for final prediction. Figure 2 shows the simple stacking model.

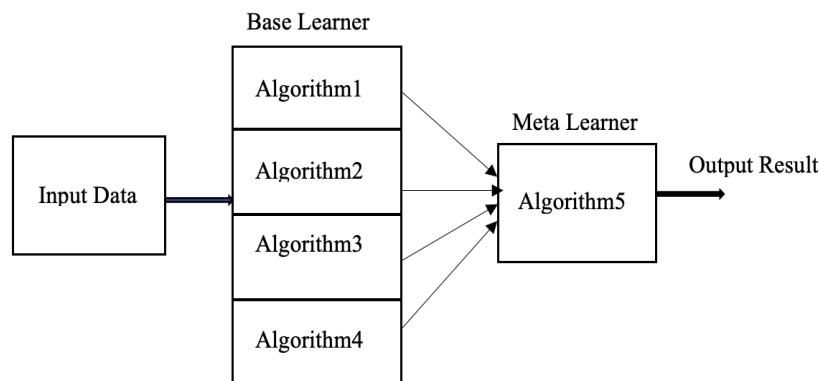


Figure 2. Stacking model

With the target Y and input X , we can represent the linear ensemble prediction as (11).

$$b(g) = w_1 * g_1 + w_2 * g_2 + \dots + w_M * g_M \tag{11}$$

where g_1 g_M are predictions learned from M machine learning algorithms and w_1 to w_M are model weights. In regression models, the model weights can be found by (12).

$$\min \sum_{i=1}^N (y_i - (w_1 * g_{1i} + w_2 * g_{2i} + \dots + w_M * g_{Mi}))^2 \tag{12}$$

In this paper for our study, we have used k-nearest neighbors (KNN), decision tree (DT) and support vector regression (SVR) algorithms as base learners and trained the models with training price data set. At

level 1 we have used the linear regression model as a meta learner. The linear regression model is trained with the results of base learner algorithms and the final prediction result is generated.

3.4. Gradient boosting

The gradient boosting algorithm is one of the powerful techniques used for prediction. The idea is originated from Leo Breiman and subsequently improved by Friedman [46]. In the case of gradient boosting, many models are getting trained gradually and sequentially. In gradient boosting, each predictor corrects its predecessor's error. This algorithm will reduce bias errors.

We can generate a diverse set of models by using many different machine learning algorithms at various hyper parameter settings. The gradient boosting method is based on combining diverse decision tree models. The boosting algorithm sequentially adjusted the residuals. That means at each level the prediction will happen and the residuals calculated as observed value-predicted. This residual is used for the next level decision tree and the process will continue till getting the improved model. Figure 3 shows how gradient boosting works. For our study, we have used the scikit learn gradient boosting regressor model.

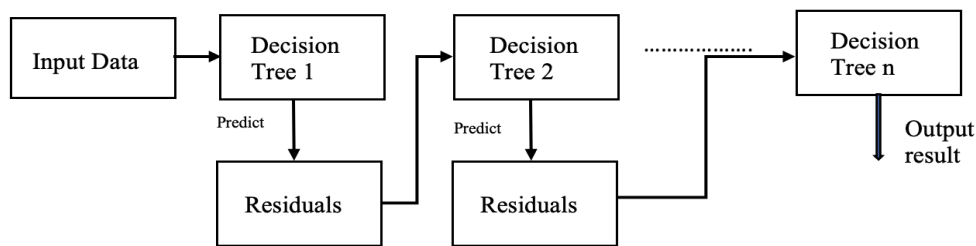


Figure 3. Gradient boosting model

3.5. Model performance evaluation

To measure the model performance, we have used a statistical parameter, mean absolute percentage error (MAPE), mean absolute error (MAE), mean standard error (MSE), root mean square error (RMSE) and coefficient of determination metrics (R^2). The MAPE is one of the commonly used statistical parameters to measure the performance or accuracy of forecasting algorithms. The value of MAPE is near zero means the accuracy of the algorithm is high.

$$MAPE = \frac{100}{n} \sum_{i=1}^n |(A_t - F_t)/A_t| \quad (13)$$

where, n is number of elements in the data set, A_t is actual value, and F_t is predicted value.

The MAE indicates the average of the residuals in the given dataset. This represents the average of the absolute difference between the actual and predicted values in the dataset.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (14)$$

where, n is number of elements in the data set, y_i is actual value and \hat{y}_i is predicted value.

The RMSE gives the standard deviation value. The RMSE is nothing but the square root of MSE. This helps us to evaluate the usefulness and accuracy of the prediction model.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

where, n is number of elements in the data set, y_i is actual value, and \hat{y}_i is predicted value.

The R-squared error represents the fraction of variance of actual value instead of residuals. The value of this error indicates the quality of the regression or prediction model. This also represents the goodness of fit for the regression model.

$$R2 = 1 - (\sum (y_i - \hat{y}_i)^2 / \sum (y_i - \bar{y}_i)^2) \quad (16)$$

where, \bar{y}_i is mean value of actual data, y_i is actual value, and \hat{y}_i is predicted value.

3.6. Prediction results

Using the proposed model, we got the following results with the given historical price for different commodities and states. Table 3 gives the details of the error values we got with the different prediction algorithms for tomato and Figure 4 shows the graphical representation of the performance values. For the monthly price data, the value of MAPE is comparable for all the used algorithms. But for the weekly data, the performance of stacking and gradient boosting is better than SARIMA.

Table 4 provides details about the performance of different algorithms for potatoes and Figure 5 is a graphical representation of performance values. With experimental results, we can see that with the stacking algorithm, we got 85% accuracy for Maharashtra and monthly price data. For the weekly price data gradient boosting gave 85% accuracy for Karnataka and Maharashtra. This clearly shows that a single algorithm will not provide the same performance for the different datasets.

The experimental results clearly show that no single algorithm consistently performed well on any of the commodity and prediction frequencies. For Karnataka and tomato, the performance of SARIMA and stacking is almost the same with a 41% error on monthly data. However, for the weekly data, gradient boosting is better than other algorithms with a 12% of error. If we consider the same commodity tomato with monthly and weekly data for Maharashtra, gradient boosting is the algorithm with less error.

Table 3. Price prediction performance for tomato with different algorithms

Algorithm	State	MAE		R ²		RMSE		MAPE	
		Monthly	Weekly	Monthly	Weekly	Monthly	Weekly	Monthly	Weekly
SARIMA	Karnataka	573.906	566.433	0.468	0.234	729.153	679.506	41.877	50.12
	Maharashtra	608.538	694.066	0.551	0.06	716.756	778.225	41.607	62.087
	Madhya Pradesh	767.835	797.998	0.305	0.667	1004.07	1087.3	41.089	40.397
Stacking	Karnataka	507.838	191.565	0.079	0.836	625.17	247.667	41.326	15.06
	Maharashtra	555.867	299.127	0.386	0.729	677.452	393.461	37.927	22.061
	Madhya Pradesh	645.669	272.059	0.657	0.831	851.249	346.215	39.919	17.672
Gradient Boosting	Karnataka	608	141.929	0.482	0.89	732.569	202.605	50.381	12.661
	Maharashtra	544.401	302.182	0.339	0.685	665.929	424.501	35.481	21.961
	Madhya Pradesh	750.786	263.58	0.888	0.816	908.633	0.816	57.306	16.662

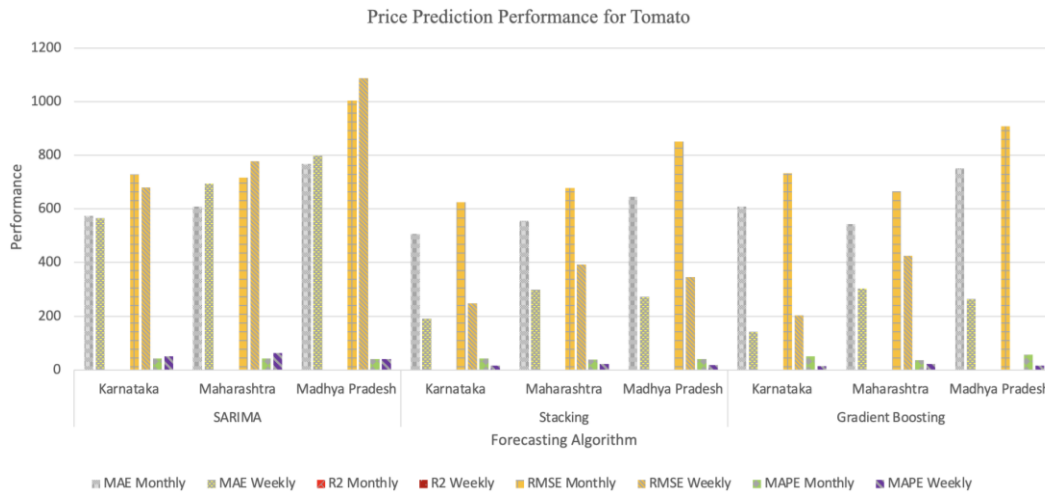


Figure 4. Price prediction performance values for tomato

Table 4. Price prediction performance for potato with different algorithms

Algorithm	State	MAE		R ²		RMSE		MAPE	
		Monthly	Weekly	Monthly	Weekly	Monthly	Weekly	Monthly	Weekly
SARIMA	Karnataka	1255.64	700.169	0.598	0.908	1890.54	864.046	50.917	29.081
	Maharashtra	759.169	1093.55	0.587	0.506	969.283	1239.74	38.434	44.286
	Madhya Pradesh	701.355	974.116	0.18	0.453	953.341	1155.71	45.463	50.809
Stacking	Karnataka	738.779	425.014	0.351	0.374	1204.83	593.975	40.591	16.82
	Maharashtra	276.29	514.634	0.626	0.573	368.494	732.468	15.7	20.697
	Madhya Pradesh	369.097	600.754	0.135	0.744	600.513	821.268	20.83	28.252
Gradient Boosting	Karnataka	822.895	375.193	0.156	0.149	1373.62	543.068	44.968	15.133
	Maharashtra	312.63	414.072	0.481	0.159	434.199	628.783	16.701	15.518
	Madhya Pradesh	339.887	554.386	0.169	0.569	588.504	779.107	18.359	25.393

If we consider potatoes with monthly data for Maharashtra, we can see that stacking gave better performance, with a 28% error, instead of gradient boosting, with a 36% error, which performed better for tomatoes with a 35% MAPE. However, in the case of potatoes with monthly data, for Madhya Pradesh, gradient boosting is the choice with 18% MAPE. Table 5 gives the details of forecasting algorithm selection for the different commodities with the above proposed dynamic algorithm, for the given monthly and weekly data set. Our experimental results show that stacking is a good option for monthly price forecasting and gradient boosting is good for weekly price forecasting.

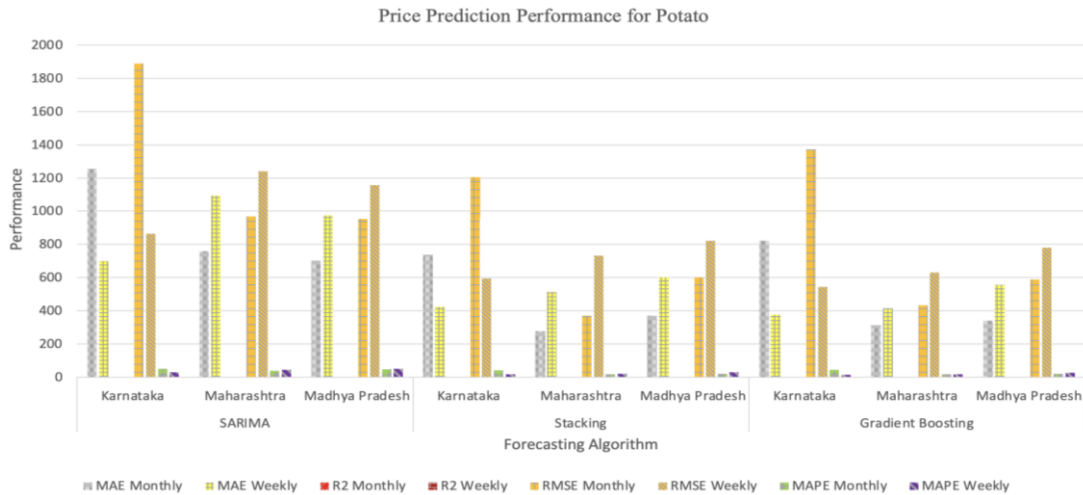


Figure 5. Price prediction performance values for potato

Table 5. Forecasting algorithm selected using dynamic algorithm selection

Commodity	Algorithm	Karnataka		Maharashtra		Madhya Pradesh	
		Monthly	Weekly	Monthly	Weekly	Monthly	Weekly
Tomato	Stacking	✓				✓	
	Gradient Boosting		✓	✓	✓		✓
Potato	Stacking	✓		✓			
	Gradient Boosting		✓		✓	✓	✓

4. CONCLUSION

In this study, we proposed a novel dynamic algorithm selection logic that will help to select the prediction algorithm based on prediction accuracy for the given data set. From the experimental results, we found that users cannot select the algorithm based on the type of commodity like a tomato. For this study, we have collected monthly and weekly price data for tomatoes and potatoes. Our experimental results show that for long-term forecasting, the performance of the stack ensemble model is good. For the short-term, gradient boosting gives better accuracy. For Karnataka, we got 59% and 60% prediction accuracy for the monthly price of tomatoes and potatoes respectively with stacking.

Similarly, for the weekly price we got 88% and 85% accuracy for tomatoes and potatoes respectively with the gradient boosting model. For Maharashtra, we got 85% prediction accuracy for the monthly price of potatoes with stacking. For tomatoes, it is 65% accuracy with gradient boosting. Similarly, for the weekly price we got 79% and 85% accuracy for tomatoes and potatoes respectively with the gradient boosting model. These results clearly show the requirement of a data-driven prediction algorithm.

The advantages of the proposed model are i) dynamically selecting the best-fit algorithm based on the prediction accuracy and the given data, ii) no need to execute the individual algorithm and find the best-fit prediction model, and iii) unless input data got changed or new prediction model added, it is not required to run the prediction algorithm. For further enhancements we are trying to i) use the caching of results; ii) improve the prediction accuracy using other factors like weather, production, the arrival of commodities, inflation; and iii) predict and forecast the price using advanced techniques like ANN, LSTM, back propagation network, and RNN. Since the price data of the commodities will change over time and for forecasting frequency, the dynamic prediction algorithm selection logic is very useful and save the prediction and forecasting time.

ACKNOWLEDGEMENTS

The authors want to thank the REVA University and ServiceNow management for their support of this research activity.




REFERENCES

- [1] MRIN, "AGMARKNET," Research and Information Network. <https://agmarknet.gov.in/> (accessed Sep. 27, 2022).
- [2] G. Hegde, V. R. Hulipalled, and J. B. Simha, "Price prediction of agriculture commodities using machine learning and NLP," in *2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Dec. 2021, pp. 1–6, doi: 10.1109/ICSTCEE54422.2021.9708582.
- [3] M. Fathi, M. Haghi Kashani, S. M. Jameii, and E. Mahdipour, "Big data analytics in weather forecasting: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 1247–1275, Mar. 2022, doi: 10.1007/s11831-021-09616-4.
- [4] A. Peresan, V. G. Kossobokov, and G. F. Panza, "Operational earthquake forecast/prediction," *Rendiconti Lincei*, vol. 23, no. 2, pp. 131–138, Jun. 2012, doi: 10.1007/s12210-012-0171-7.
- [5] M. Jones, G. Kotsalis, and J. S. Shamma, "Cyber-attack forecast modeling and complexity reduction using a game-theoretic framework," in *Control of Cyber-Physical Systems*, Springer International Publishing, 2013, pp. 65–84, doi: 10.1007/978-3-319-01159-2_4.
- [6] S. Sivaranjani, S. Sivakumari, and M. Aasha, "Crime prediction and forecasting in Tamilnadu using clustering approaches," in *2016 International Conference on Emerging Technological Trends (ICETT)*, Oct. 2016, pp. 1–6, doi: 10.1109/ICETT.2016.7873764.
- [7] H. R. Maier and G. C. Dandy, "Application of artificial neural networks to forecasting of surface water quality variables: issues, applications and challenges," in *Water Science and Technology Library*, Springer Netherlands, 2000, pp. 287–309, doi: 10.1007/978-94-015-9341-0_15.
- [8] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing and Applications*, vol. 32, no. 6, pp. 1609–1628, Mar. 2020, doi: 10.1007/s00521-019-04212-x.
- [9] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, May 2021, doi: 10.1007/s00521-020-05532-z.
- [10] P. Shine, M. D. Murphy, J. Upton, and T. Scully, "Machine-learning algorithms for predicting on-farm direct water and electricity consumption on pasture based dairy farms," *Computers and Electronics in Agriculture*, vol. 150, pp. 74–87, Jul. 2018, doi: 10.1016/j.compag.2018.03.023.
- [11] M. C. Pegalajar, L. G. B. Ruiz, M. P. Cuéllar, and R. Rueda, "Analysis and enhanced prediction of the Spanish electricity network through big data and machine learning techniques," *International Journal of Approximate Reasoning*, vol. 133, pp. 48–59, Jun. 2021, doi: 10.1016/j.ijar.2021.03.002.
- [12] R.-Z. Wang, Z.-H. Ling, and Y. Hu, "Knowledge base question answering with attentive pooling for question representation," *IEEE Access*, vol. 7, pp. 46773–46784, 2019, doi: 10.1109/ACCESS.2019.2909826.
- [13] Y. Sun and T. Xia, "A hybrid network model for Tibetan question answering," *IEEE Access*, vol. 7, pp. 52769–52777, 2019, doi: 10.1109/ACCESS.2019.2911320.
- [14] Y. Weng, X. Wang, J. Hua, H. Wang, M. Kang, and F.-Y. Wang, "Forecasting horticultural products price using ARIMA model and neural network based on a large-scale data set collected by web crawler," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 547–553, Jun. 2019, doi: 10.1109/TCSS.2019.2914499.
- [15] A. A. Reddy, "Price forecasting of tomatoes," *International Journal of Vegetable Science*, vol. 25, no. 2, pp. 176–184, Mar. 2019, doi: 10.1080/19315260.2018.1495674.
- [16] L. Wang, J. Feng, X. Sui, X. Chu, and W. Mu, "Agricultural product price forecasting methods: research advances and trend," *British Food Journal*, vol. 122, no. 7, pp. 2121–2138, Jun. 2020, doi: 10.1108/BFJ-09-2019-0683.
- [17] A. Hayat and M. I. Bhatti, "Masking of volatility by seasonal adjustment methods," *Economic Modelling*, vol. 33, pp. 676–688, Jul. 2013, doi: 10.1016/j.econmod.2013.05.016.
- [18] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock-market prediction," *Journal of Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00299-5.
- [19] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.
- [20] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.
- [21] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, pp. 1–10, Nov. 2020, doi: 10.1155/2020/6622927.
- [22] Y. Yu, H. Zhou, and J. Fu, "Research on agricultural product price forecasting model based on improved BP neural network," *Journal of Ambient Intelligence and Humanized Computing*, Aug. 2018, doi: 10.1007/s12652-018-1008-8.
- [23] M. Yi, W. Xie, and L. Mo, "Short-term electricity price forecasting based on BP neural network optimized by SAPSO," *Energies*, vol. 14, no. 20, Oct. 2021, doi: 10.3390/en14206514.
- [24] B. Wang *et al.*, "Research on hybrid model of garlic short-term price forecasting based on big data," *Computers, Materials & Continua*, vol. 57, no. 2, pp. 283–296, 2018, doi: 10.32604/cmc.2018.03791.
- [25] G. S. Kakaraparthi and B. V. A. N. S. P. Rao, "Crop price prediction using machine learning," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, no. 6, pp. 3477–3481, 2021.
- [26] M. Rakhra *et al.*, "WITHDRAWN: crop price prediction using random forest and decision tree regression: a review," *Materials Today: Proceedings*, Apr. 2021, doi: 10.1016/j.matpr.2021.03.261.
- [27] P. Samuel, B. Sahithi, T. Saheli, D. Ramanika, and N. A. Kumar, "Crop price prediction system using machine learning algorithms," *Quest Journals Journal of Software Engineering and Simulation*, 2020.
- [28] R. Dhanapal, A. AjanRaj, S. Balavinayagapragathish, and J. Balaji, "Crop price prediction using supervised machine learning algorithms," *Journal of Physics: Conference Series*, vol. 1916, no. 1, May 2021, doi: 10.1088/1742-6596/1916/1/012042.
- [29] J. H. Zhang, F. T. Kong, J. Z. Wu, M. S. Zhu, K. Xu, and J. J. Liu, "Tomato prices time series prediction model based on wavelet neural network," *Applied Mechanics and Materials*, vol. 644–650, pp. 2636–2640, Sep. 2014, doi: 10.4028/www.scientific.net/AMM.644-650.2636.
- [30] H. Adanacioglu and M. Yercan, "An analysis of tomato prices at wholesale level in Turkey: an application of SARIMA model," *Custos e@ gronegocio on line*, vol. 8, no. 4, pp. 52–75, 2012.




- [31] F. Boateng, J. Amoah-Mensah, M. Anokye, L. Osei, and P. Dzebre, "Modeling of tomato prices in Ashanti region, Ghana, using seasonal autoregressive integrated moving average model," *British Journal of Mathematics and Computer Science*, vol. 20, no. 2, pp. 1–13, Jan. 2017, doi: 10.9734/BJMCS/2017/30535.
- [32] M. I. Ansari and S. M. Ahmed, "Time series analysis of tea prices: an application of ARIMA modelling and cointegration analysis," *The Indian Economic Journal*, vol. 48, no. 3, pp. 49–54, Mar. 2001, doi: 10.1177/0019466220010305.
- [33] K. Assis, A. Amran, and Y. Remali, "Forecasting cocoa bean prices using univariate time series models," *Researchers World*, vol. 1, no. 1, 2010.
- [34] A. Darekar and A. A. Reddy, "Price forecasting of pulses: case of pigeon pea," *Journal of Food Legumes*, vol. 30, no. 3, pp. 42–46, 2017.
- [35] A. Darekar and A. A. Reddy, "Cotton price forecasting in major producing states," *Economic Affairs*, vol. 62, no. 3, 2017, doi: 10.5958/0976-4666.2017.00047.X.
- [36] T. Xiong, C. Li, Y. Bao, Z. Hu, and L. Zhang, "A combination method for interval forecasting of agricultural commodity futures prices," *Knowledge-Based Systems*, vol. 77, pp. 92–102, Mar. 2015, doi: 10.1016/j.knsys.2015.01.002.
- [37] L. Zheming, X. Shiwei, and C. Ligu, "Prediction study based on dynamic chaotic neural network: Taking potato timeseries prices as an example," *Systems Engineering-Theory and Practice*, vol. 35, no. 8, pp. 2083–2091, 2015.
- [38] M. Dipankar, R. K. Paul, and A. K. Paul, "Statistical modelling for forecasting volatility in potato prices using ARFIMA-FIGARCH model," *Indian Journal of Agricultural Sciences*, vol. 88, no. 2, pp. 268–272, 2018.
- [39] M. Areef, "Price behaviour and forecasting of onion prices in Kurnool Market, Andhra Pradesh State," *Economic Affairs*, vol. 65, no. 1, Mar. 2020, doi: 10.30954/0424-2513.1.2020.6.
- [40] R. Nalini, K. Sountharya, R. V. Priya, and R. V. Punitha, "Onion price prediction based on artificial intelligence," *International Research Journal of Multidisciplinary Technovation*, pp. 11–20, Jul. 2020, doi: 10.34256/irjmt2043.
- [41] J. Wang, C. Qi, and M. F. Li, "Prediction of commodity prices based on SSA-ELM," *System Engineering Theory and Practice*, vol. 37, no. 8, pp. 2004–2014, 2017.
- [42] M. Shahhosseini, G. Hu, and S. V. Archontoulis, "Forecasting corn yield with machine learning ensembles," *Frontiers in Plant Science*, vol. 11, Jul. 2020, doi: 10.3389/fpls.2020.01120.
- [43] T. Xiong, C. Li, and Y. Bao, "Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China," *Neurocomputing*, vol. 275, pp. 2831–2844, Jan. 2018, doi: 10.1016/j.neucom.2017.11.053.
- [44] Chang, "Seasonal autoregressive integrated moving average model for precipitation time series," *Journal of Mathematics and Statistics*, vol. 8, no. 4, pp. 500–505, Apr. 2012, doi: 10.3844/jmsp.2012.500.505.
- [45] R. J. Hyndman and G. Athanasopoulos, "Seasonal ARIMA models," in *Forecasting: Principles and Practice*, Australia, 2018.
- [46] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, Oct. 2001, doi: 10.1214/aos/1013203451.

BIOGRAPHIES OF AUTHORS






Girish Hegde    is a Staff Software Engineer at ServiceNow and a research scholar in the School of Computing and IT, REVA University, Bangalore, Karnataka, India. He Completed B.E. in Computer Science and Engineering and M.E. in Computer Science and Engineering. His area of Interest includes AI and Machine Learning, Cloud Computing, and Data Analytics. He can be contacted at email: girishhegde37@gmail.com.



Vishwanath R. Hulipalled    is a Professor in the School of Computing and IT, REVA University, Bangalore, Karnataka, India. He completed BE, ME and Ph.D. in Computer Science and Engineering. His area of Interest includes Machine Learning, Natural Language Processing, Data Analytics and Time Series Mining. He has more than 24 years of academic experience and research. He authored more than 50 research articles in reputed journals and conference proceedings. He can be contacted at email: vishwanth.rh@reva.edu.in.



Jay B. Simha    is the CTO of ABIBA Systems and Chief Mentor at RACE Labs, REVA University. He completed his BE (Mech), M.Tech (Mech), and M.Phil (CS) and Ph.D. (AI). His area of interest includes fuzzy logic, soft computing, machine learning, deep learning, and applications. He has more than 20 years of industrial experience and 4 years of academic experience. He has authored/co-authored more than 50 journal/conference publications. He can be contacted at: jay.b.simha@reva.edu.in, jay.b.simha@abibasystems.com.