Expert Systems With Applications 116 (2019) 108–120

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Time series decomposition and predictive analytics using MapReduce framework

Mininath Bendre[a,*], Ramchandra Manthalkar[b]

[a] Department of Computer Engineering, Pravara Rural Engineering College, Loni, Ahmednagar 413736, India
[b] Department of Electronics and Telecommunication Engineering, Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded 431606, India

ARTICLE INFO

Article history:
Received 13 December 2017
Revised 7 September 2018
Accepted 8 September 2018
Available online 8 September 2018

Keywords:
Big data
Time series analysis
Neural network
Predictive weather analysis and forecasting
MapReduce

ABSTRACT

The recent development in precision agriculture, a large amount of data are generated by site-specific weather stations which will demand a platform for the processing and predictive weather analytics. The sophisticated methodology to solve large amount of data handling problem and process data in a small time is important. In this study, future conditions are predicted from weather stations large data by proposing the predictive approaches based on time series and neural network using MapReduce programming model. We have proposed predictive analytics approaches including the modules, i.e., analysis and decomposition, classification, and prediction. The time series based decomposition approach is proposed to decompose and find out the trend, regular and sophisticated components. The linear components are handled by time series MapReduce based Autoregressive Integrated Moving Average (M-ARIMA) model and nonlinear components are handled by M-K-Nearest Neighbors (M-KNN) model. In addition, the MapReduce-based Hybrid Model (M-HM) was proposed which will use the advantages of time series and neural network to increase prediction accuracy. The study verifies the effectiveness of proposed model over the regular and randomness component of the data. The performance measures and statistical test are performed to validate and check data consistency. In addition, excellent speed-up, scale-up, and size-up were tested by changing the size of data set. However, when the data size increases, the average execution time is reduced by using the MapReduce-based approach over the multiple-node workers.

1. Introduction

In precision agriculture, the huge amount of weather data can be accumulated in a less time by using sensing and Information & Communications Technology (ICT) based equipment. Such huge data generated using ICT components in the application of agriculture can be utilized for precision farming management and decision support using various kinds of strategies applied over off-site or on-site farming. Weather forecasting plays the main role in the farm management, such as rainfall predictions, temperature change, and humidity analysis, identifying frost conditions, deciding cropping patterns, early harvesting etc. Currently, weather data have diverseness in different aspects such as volume, velocity and variety etc. The new generation computing system having faster processing unit with the real-time operations capability will help to analyze such data. The algorithm modeling over big data platform can be solve data analytics problem in a small time. To forecast weather conditions, of many different approaches processed including numerical weather prediction (Miyoshi, Kondo, & Terasaki, 2015), Artificial Neural Network (ANN) (Zhang, Patuwo, & Hu, 1998), hybrid approach using ANN and statistics (Ganguly, 2002), fuzzy time series (Chen & Hwang, 2000), linear time series model (Mathew, Sreekumar, Khandelwal, Kaul, & Kumar, 2016), Kalman filtering and prediction algorithm (Sharma & Mahalanabis, 1974). The literature focuses on methods which give predictive decisions by self-generated input and output. Therefore, to find out the stochastic behavior of the input data and accuracy in forecasting is a more complicated task. When data size increases, the real-time data handling capability, efficiency, and performance of the system for these methods decreases. To achieve better performance of the model and prediction accuracy, the data must be pre-processed and updated regularly. The uncertainty and a variation in the patterns of data will give inaccurate future predictive decisions. Therefore, proper data analytics methods and approaches are important. The ANN and decision tree models depend on learning and training process to increase accuracy and mini-

* Corresponding author.
E-mail addresses: bendremr@pravaraengg.org.in (M. Bendre), rrmanthalkar@sggs.ac.in (R. Manthalkar).

https://doi.org/10.1016/j.eswa.2018.09.017
0957-4174/

mize squared error. If data is huge, then process requires large time to execute over a single-node system and it creates a burden on it. Therefore, large storage and computational processing resources for better data analysis and management are required. This problem, however, can be solved by using MapReduce-based parallel and distributed platform.

The parallel and distributed platforms (Li, Lu, & Meng, 2015; Meng, Dou, Zhang, & Chen, 2014; Sakr et al., 2015; Shang et al., 2013; Xing et al., 2015) can be utilized to solve predictive weather analytics problem. To get insights from such huge data the sophisticated data analytics methods are need to be proposed. Recently, parallel and distributed processing platforms, methods and techniques are used in many applications for the purpose of data management and analytics. For the site-specific agricultural management suitable data processing platform is required with the capability of handling sensor-generated data. The data acquisition and storage management through Internet of Things (IoT) and ICT components on parallel distributed systems is valuable in the precision agriculture management.

To analyze and predict homogeneous temperatures from different locations by applying clustering techniques such as global K-means clustering algorithm and fuzzy C-means clustering algorithm are possible (Bharath, Srinivas, & Basu, 2015). Whenever data size increases, the data analysis, behavior analysis and accurate prediction are the main challenges. If any changing temperature frost event occurs in future which will may cause drought condition. Therefore, our contribution focuses temperature data analysis, decomposition, and forecasting using proposed approach. The main contributions of this paper are summarized as follows.

1. We propose a method of distributed modeling under the MapReduce framework to avoid time consumption in weather forecasting application. In this study, we use parallel and distributed computing clusters to meet the storage and processing power required for the execution. We have integrated the MapReduce framework with proposed time-series, neural network and hybrid model to forecast decisions from a large data set. The proposed approach is able to handle variety of data in huge amount with less time which will satisfies properties of big data such as variety, volume and velocity.
2. A new decomposition approach is proposed on big data platform to improve the capability of a large amount of data processing and check the behavior of the signal. The approach uses time series based method to find the trend, regular and stochastic components present in the data which will be further utilized for the accurate predictions and weather stations profile analysis.
3. The MapReduce based Autoregressive Integrated Moving Average (M-ARIMA) model on a big data platform is presented to forecast temperature and decompose components, by analyzing Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) information of whole data.
4. The robust MapReduce based K-Nearest Neighbors (M-KNN) and novel MapReduce based Hybrid Model (M-HM) are proposed to increase the forecasting accuracy and handle both linear-nonlinear components easily.
5. The prediction performance of the proposed approaches is investigated with data collected from 39 weather stations of Maharashtra during 100 years. The empirical results stated the proposed approach are better in terms of accuracy, speed-up and size up.

The paper is organized as follows: Section 2 describes the concept of data sets, big data system architecture, MapReduce approach and the proposed forecasting models. The results and discussion based on proposed model are demonstrated in Section 3. Finally, the conclusion drawn based on results is given in Section 4.

2. Methodology and proposed strategy

2.1. Data sets and study area

We have used large weather data generated by site-specific stations located in various district areas of Maharashtra state. The referred data is downloaded from Indian Water Portal authorized by Indian meteorological Department (IMD). The dataset contain 39 weather stations 100 years old historical minimum, average and maximum temperature, humidity and more variables data. These stations are selected randomly on basis of agro climatic zones of Maharashtra state. Maharashtra state is divided into nine agro climatic zones on the basis of rainfall, temperature, soil types, cropping pattern etc. In this study, 39 weather stations monthly average maximum temperature is used for the purpose of analytics which is decomposed and predicted using proposed approach. To evaluate and test the performance of the proposed model, three different stations monthly maximum temperature data is used. The data is big by the number of data-points and to process huge number of data-points requires more time. These stations are located at district in the area having a variety of agricultural productions and different land use characteristics in the Maharashtra state. The low level of precipitation, short cool season and long warm season are among the climatic features of this state. According to the average of long-term measured data, the monthly average air temperature varies from 17.7°C to 34.4°C, and the yearly average is 27°C. The monthly average relative humidity lies between 60% and 69% of the annual average of 64%. Fig. 1 shows the various weather stations located at the different area of the Maharashtra. Mainly, we have utilized to analyze, decompose, forecast, and comparison purpose the data of three weather stations such as Ahmednagar, Nanded, and Amravati. Further the results of these stations are checked, tested and validated using predictive models.

If noise is present in the data which will creates big problem for the data processing. To removal of noise and pre-processing of collected data is important. Noisy data is data with additional garbage information which is not useful. While, collecting and storing large amount data with noise is an additional challenge for big data platform. Needs more memory space and processing unit to analyze such data. Therefore we have used the averaging method to replace garbage or unavailable data before analyzing. In this study, we have defined boundary conditions for temperature (−100 to 100 °C), humidity (1 to 100%) and rainfall (0 to 1000 mm). Those variables values exceeds the conditions are replaced with averaging method.

In this study, the data from the year 1901–1992 are used as an input to the proposed model. Also, the data from the year 1993–2002 are used to test the difference between actual and predicted data by evaluating error values. From the available parameters were considered as inputs to predict monthly maximum temperature. The data sets consist of various parameters, we are analyzing measured temperature in degree Celsius (°C).

2.2. Big data system architecture and platform

We proposed modeling in the big data architecture to analyze and forecast big weather data taken from different weather stations. The process of data acquisition initiated to store data in the datastore. To the application design purpose, the big data platform with the capability of storing and processing of data are to be designed. Fig. 2 shows the big data platform, in addition the data loading and pre-processing module used in this framework. The platform uses parallel processing power using workers on community cluster. The platform is mainly divided into the four components, i.e. data load, MapReduce framework, workers, and output or analytics. Initially, the data is loaded into datastore object, then

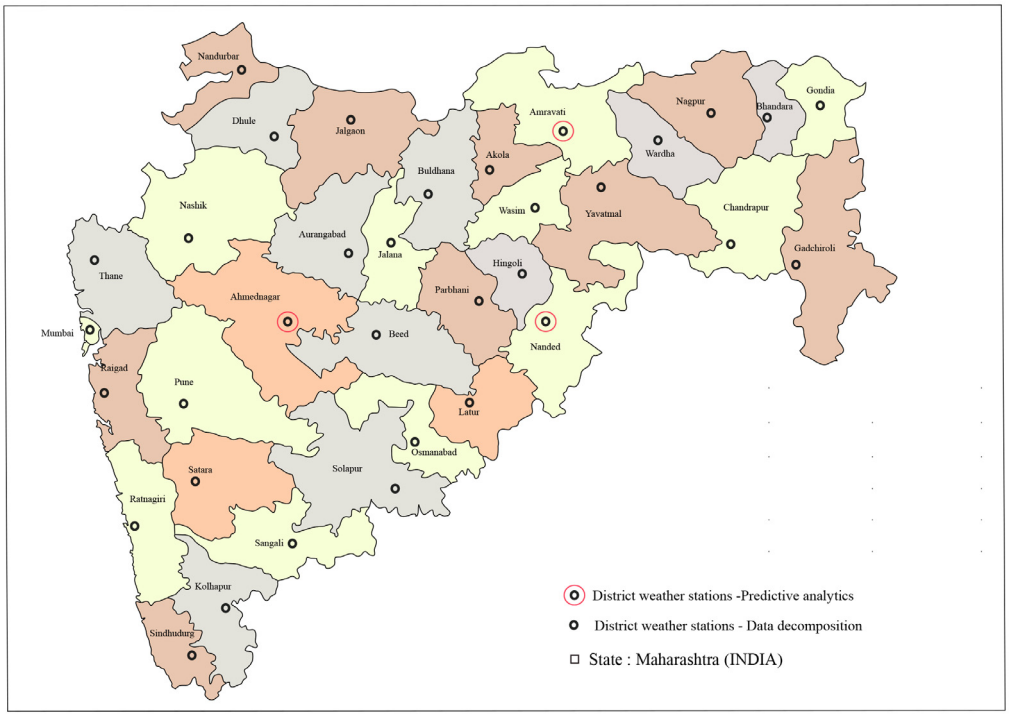

Fig. 1. Available weather stations data of Maharashtra state.

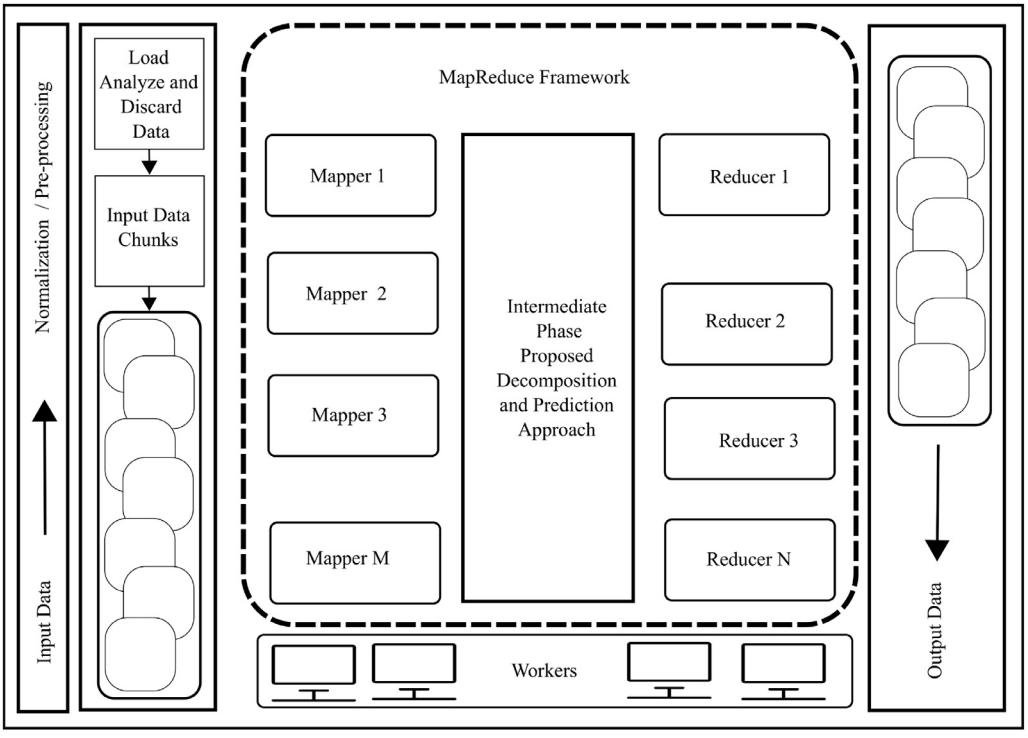

Fig. 2. Big data system architecture.

it is read in the mapping variables by loading and partitioning into different chunks for easily process over the MapReduce programming model. The chunks of data are initially pre-processed using pre-processing method before applying to the MapReduce phases. The data is pre-processed using averaging method. Secondly, map and reduce phases of the framework are used to perform parallel processing using data analytics methods. The data analytics methods are going from the map and reduce phases of the framework. Thirdly, workers or workstations plays the main role of big data processing using internal communication between server and clients. Finally, the output is collected by combining the output of all reduce phases into a single file.

In the application of agricultural management, the big data platform having parallel and distributed processing of huge data plays the main role regarding execution time, large storage, and faster computing management. Parallel architecture can be used

for the simultaneous execution of the different processes to avoid the time consumed and also used for the distribution of large data to avoid storage and processing problems. To process mean, average, minimum and maximum values of temperature using a MapReduce based methodology is possible in less time. The parallel execution of processes is used in the proposed work to reduce execution time and MapReduce architecture for the data distribution and processing. In this study, we have used parallel processing feature of the Matlab Production Server big data platform over MapReduce programming model. The group of workers (cluster) has been configured for the large weather data processing and management. The cluster is configured by varying number of workers (two, four and eight are used in this study) as increasing the size of input data. Also, the default and customized configuration are validated and applied at the time of execution.

2.3. MapReduce approach

The MapReduce is the programming technique to analyze large amounts of data by dividing into memory fit-able chunks. The Fig. 3 shows the working of MapReduce model with different phases. The MapReduce model presented in the figure is divided into input, mapping, reducing and combining phase. The chunk of data is applied to map phase, each mapper reads input chunk and produces intermediate output. In the Fig. 3, term ($< k1, < v1, v2, v3 > >$) represents a key (k1) with associated values (v1,v2,v3) which are further processed by worker. The dataset have many parameters, these values noted for unique time period. Key is the time based component and values are the signals noted at that time. MapReduce model automatically distributes these chunks to the workers present in the cluster. Similarly, reducer function reads intermediate data and produces the final output. In this study, different chunks (chunk 1, chunk 2, ... , chunk M) are the inputs to the mappers (mapper 1, mapper 2, ... , mapper N) respectively. The mapper function works on the individual chunk and adds one or more $\{key - value\}$ pairs in an intermediate object. Before calling reduce function, the MapReduce framework is responsible for grouping the values into the intermediate object. Also, reducers (reducer 1, reducer 2,... , reducer N) and output splits (outkey 1, outkey 2,... , outkey N) are shown in the Fig. 3. All the unique keys associated with values are called in the reduce function to perform the final analysis. Separate mapper and reducer functions are designed to perform data decomposition and prediction of decomposed data. The linear mapper and reducer functions are used to handle strongly linear components and distance function is used to evaluate Euclidean distance for the KNN. Similarly these MapReduce functionality is used for the prediction of the all decomposed components used in this study. Finally each reducer combines decomposed component separately.

In this paper, we have proposed MapReduce based decomposition and prediction approaches. We have designed algorithms which will fit in the MapReduce paradigm to give better results. To analyze and decompose data into different components the decomposition mapper were proposed. Also separately M-KNN and M-HM mappers are developed to process and predict data accurately.

2.4. Data analysis and model selection

The selection of the model is based on the behavior of data. Also, the decomposed trend, periodicity, regularity and stochastic behavior of components present in the data used to train model correctly. The ACF and PACF plots of data at n-lag help to identify stationary or nonstationary and basic behavior of the data. However, the ACF and PACF analysis of 39 weather stations temperature data clearly utilized for the selection of the decomposition method, seasonal filter and use of Moving Average (MA), Autoregressive (AR), ARIMA, KNN and HM model parameters. The decomposed components may contain periodic, seasonal, stochastic, trended and noise patterns which are categorized as per structure. Data chosen from three stations are used for the comparative analysis of stochastic and regular behavior and evaluation of high-temperature frost conditions forecasting evaluation for future analysis. The selected data having more seasonality and periodicity which will need a decomposition with a proper seasonal filter. The Fig. 4 shows the ACF and PACF of the three different locations weather station data as a case study. The ACF of these plots shows linear decay (nonstationarity) in the data. The Fig. 4 (a)–(e) utilized to analyze seasonal pattern present in the monthly average data of three different station. It clearly shows the present periodic behavior of data for each station mostly similar, which will force to design a model applicable to all stations. Therefore, to accurately understand the various components of data, we have plotted ACF and PACF up to the 20 lags. Before going to predict and analyze monthly data we have analyzed daily values of the Ahmednagar station. Which will clearly show the after first differentiation data are in the stationary state to easily predict by applying MA and AR models. The data with weekly, monthly, seasonally or yearly presence of periodicity or seasonality need more differentiation's. If after 1st ordered difference the data is stationary then no need to analyze further ACF and PACF. The tool for determining stationarity is the ACF. If a time series is nonstationary, the ACF will not die out quickly. If the autocorrelation does not die out even for large lags then autocorrelation for the first difference is computed by, $\nabla x_t = x_t - x_{t-1}$. Also, the ACF and PACF analysis shows if seasonality is present, then it removed by backward difference e.g. $\nabla\nabla^{12}x_t$, $\nabla\nabla^{30}x_t$ etc., in our case for monthly averages 11, 12, 13 are the lags repeating at the same interval.

We conclude from the ACF and PACF plot, the MA, AR models can be used for the forecasting. This analysis will help to decide non-seasonal autoregressive term (p), non-seasonal difference (d) and non-seasonal moving average (q) in the time series analysis and model selection. But the strong component is seasonality which is present in the repeated form in the time series. Therefore, we further proceed to decompose time series and find patterns in data for forecasting. The decomposition of different components is possible, but accurate forecasting is challenging task. To forecast nonlinear component the time series approach has less accuracy. The time series approaches easily applied over the linear, nonlinear or seasonal components. Therefore, to forecast nonlinear component and to handle linear components we propose a hybrid model which uses the M-HM algorithm described in later section (see Section 2.5.3).

2.5. Proposed decomposition and predictive approach

2.5.1. Decomposition approach

In this subsection, the decomposition approach under a MapReduce framework is described. The MapReduce based decomposition flow is represented in the Algorithm 1. In the algorithm, mapper function and reducer function separately defined by the pseudocode each decomposed components are added in the intermediate code and finally, these are combined in the reduce phase. The underlay of the approach is described as follows.

The denoted weather data for the station as $W = \{X_1, X_2, X_3, \ldots, X_P\}$ where X_P representing the Pth number of variable. Mainly, in this study, three stations data is decomposed and related temperature is forecast. So, all parameters present in the database are passed to the decomposition method. For e.g. monthly average maximum temperature represented as vector x_t or $T = \{x_1, x_2, x_3, \ldots, x_n\}$ with x_n representing the temperature in the nth time slot of single time series. Where n is the total number of values

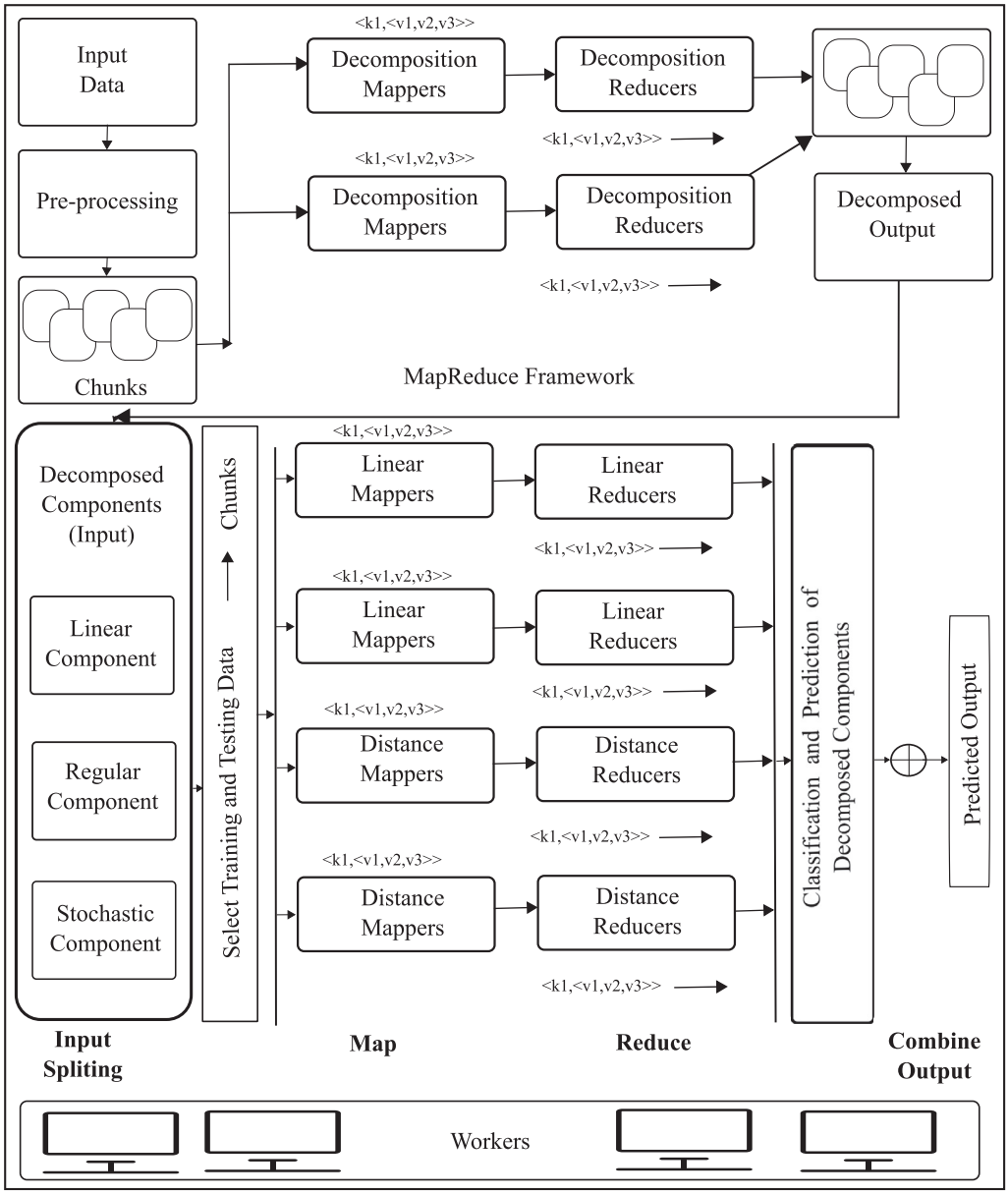

Fig. 3. MapReduce approach and its data flow.

present in the time series. Then, we define x_t is the combination of u_t and y_t,

$$x_t = u_t + y_t + e_t,\ t = 1, 2, 3, \ldots, n, \tag{1}$$

where the u_t is the seasonal or regular component and y_t is the stochastic component. Also in the time series decomposition approach the x_t represented with following components,

$$x_t = t_t + p_t + s_t,\ t = 1, 2, 3, \ldots, n, \tag{2}$$

where t_t shows a general trend, p_t is the periodic component and s_t represents the stochastic component of the series. In this paper, defined seasonal component as the sum of t_t and p_t , and stochastic component as s_t,

$$u_t = t_t + p_t,\ t = 1, 2, 3, \ldots, n, y_t = s_t, t = 1, 2, 3, \ldots, n, \tag{3}$$

In this study, we have evaluated trend by moving average and detrend methods. We are trying to find out strongly linear component present in the data. These methods accurately find the strong linear component present in the data which will be used to evaluate periodic and seasonal patterns present in the data.

To decompose data, the applied moving average filter to find out the trend of data is as follows,

$$\hat{m}_t = (x_{t-q} + x_{t-q+1} + \ldots + x_{t+q} + x_{t+q+1})/d,\ q < t \leq n - q, \tag{4}$$

where d is the time window, in this case d is 12 (number of months in the year for monthly average parameter) and $q = d/2$ which will used to compute the average w_k and periodic pattern p_k as follows,

$$\hat{p}_k = \begin{cases} w_k - d^{-1} \sum_{i=1}^{d} w_i, & k = 1, \ldots, d, \\ \hat{p}_{k-d}, & k > d, \end{cases} \tag{5}$$

The remaining data is defined by removing obtained periodic pattern,

$$d_t = x_t - p_t,\ t = 1, 2, 3, \ldots, n, \tag{6}$$

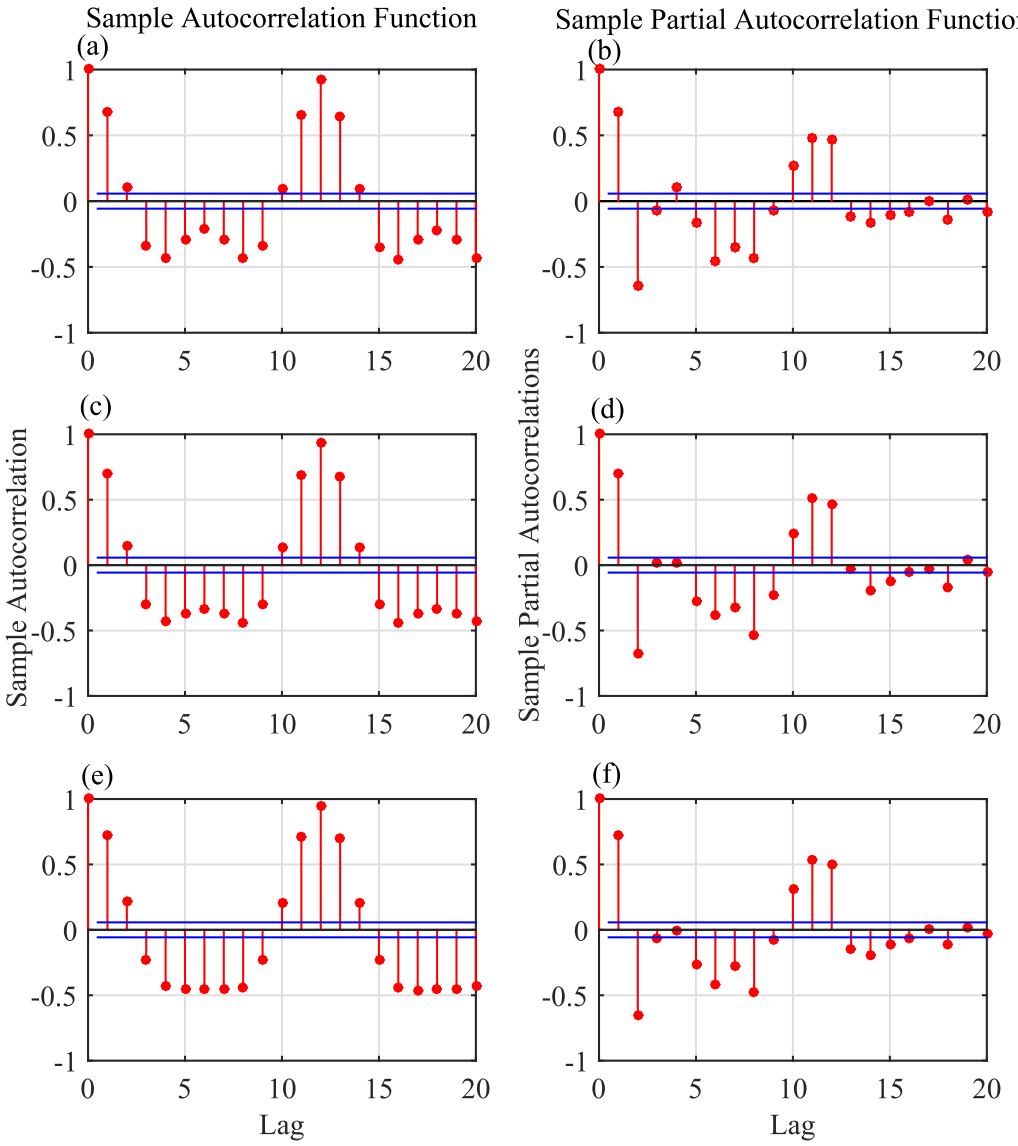

Fig. 4. The acf and pacf analysis. (a, b) the acf and pacf analysis plot of Ahmednagar station, (c, d) the acf and pacf analysis plot of Nanded station and (e, f) the acf and pacf analysis plot of Amravati station.

Afterwords, the moving average filter is applied on d_t to estimate general trend in the data and it is computed as follows,

$$t_t = (2q+1)^{-1} \sum_{i=-q}^{+q} x_{t-j} \tag{7}$$

where, q is depend on time window d and u_t, y_t are estimated from x_t, t_t and p_t. The Fig. 5 shows the evaluation and decomposition of all stations maximum temperature data as seasonal, periodic, trend and stochastic components using MapReduce based decompose method. The Fig. 6 shows the separate decomposition of the chosen three weather stations maximum temperature for the case study. Decomposition components are separately used over M-ARIMA, M-KNN and the hybrid approach M-HM to forecast future patterns.

2.5.2. M-ARIMA approach

The data have a seasonal component such as monthly, season-wise, and yearly. In this study, seasonal ARIMA model is used with general form of ARIMA $(p, d, q) \times (P, D, Q)_m$, where m is the number of periods per season, (P, D, Q) is the seasonal parts of the model with P is the seasonal autoregressive term, D is the seasonal differences and Q is the seasonal moving average term and p, d, q are corresponding non-seasonal parts. The working of M-ARIMA is presented in the Algorithm 2. The ARIMA uses Box-Jenkins time series approach to evaluate the moving average and regressive coefficients of the model. In this approach, mappers and reducers are defined to evaluate the predicted value.

2.5.3. M-KNN and M-HM approach

This section explains the hybrid approach based on ARIMA and KNN algorithm. The linear problems are greatly handled over the ARIMA model but the nonlinear samples gives poor performance. The KNN act as a powerful algorithm to handle nonlinear components but the strongly linear component have poor results. Therefore, to handle stationary and nonstationary time series and taking advantages of the both the models, here hybrid linear-nonlinear model is proposed based on ARIMA and KNN. The input data or decomposed components structure vary compared to each other. The hybrid approach act as good predictor based on structure of input data. The proposed hybrid model based on ARIMA and KNN is shown in Algorithms 1–3. In this model decomposed compo-

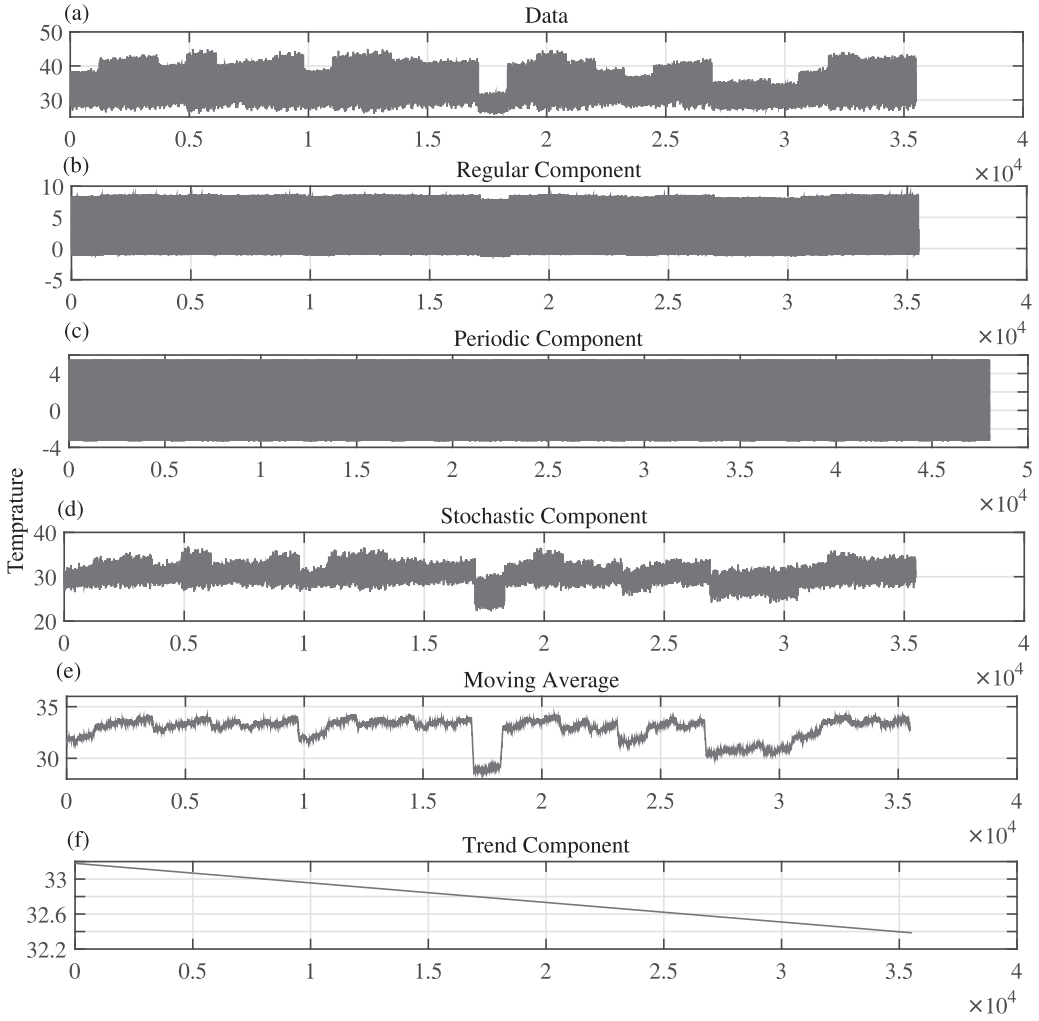

Fig. 5. Decomposed all stations data plotted linearly. (a) 39 weather stations maximum temperature data, (b) decomposed regular component, (c) decomposed periodic component, (d) decomposed stochastic component, (e) moving average and (f) linear trend component.

nents are treated as separately to analyze and forecast. The main key concepts used for the working of the hybrid model is as follows:

1. *Trend:* If the trend is strongly linear, ARIMA model is used to handled it by changing the AR coefficient. Also, it will handle by M-HM by training linear non-labeled component as labeled. Initially each component (data value for e.g. temperature at *n*th time) is act as a non-labeled component. When new non-labeled component come in the list which is matched with exiting stored label component. If it matches then particular class is assigned, if not then store it as new labeled component. Once all components are labeled then M-HM predictor easily predict forecast values using newly labeled data set.
2. *Periodic pattern:* The periodicity totally depends on the time series which repeated at a particular time interval. To forecast a periodic component M-ARIMA model easily utilized by using MA components or M-KNN. In M-KNN one periodic class is labeled initially which is act as main labeled class for the non-labeled components. Once one group of class is labeled then all other non-labeled components will labeled easily.
3. *Seasonal component:* The seasonal components are also treated using AR, MA, and seasonal filter in M-ARIMA and in M-HM it will train for the forecasting.
4. *Stochastic component:* To forecast stochastic component it is difficult to the ARIMA model or having less accuracy. The M-HM approach easily forecasts this component.
5. *Stationary and nonstationary component:* Nonstationary components are also handled by ARIMA by making first it stationary. The M-HM directly handles it for the forecasting.
6. *Nonlinear component:* The nonlinear components are better handled by M-HM than M-ARIMA model. In M-ARIMA model these components are treated with a MA and seasonal filter.

The underlay of the M-HM is M-KNN and both are described as follows. The hybrid labeled M-HM approach based on the KNN algorithm is discussed. The model is re-trained several times based on incoming parameters. As per the structure of data the model is changing. Basically, KNN is the widely used for the non-parametric bases on closest training sample algorithm. The KNN calculates the distance between each testing sample (T_{test}) and all the training samples (T_{train}). After successful calculation of the nearest neighbor list, the testing sample is classified based on the majority class. The M-HM uses the goodness of KNN and ARIMA to forecast linear and nonlinear components. In M-HM the distance matrix and neighbor list are evaluated similarly to the KNN. The detailed M-HM is shown in the Algorithm 3. In the MapReduce based KNN labeled and non-labeled components are sorted separately over the distributed platform by separate worker. Which will decrease the total time required for the classification of data.

In this approach, T represents linear or nonlinear input time series. The whole time series is distributed into training (T_{train}) and testing (T_{test}) samples. Initially, approximately 90% data from the input is used for training and 10% for the testing. The k is the num-

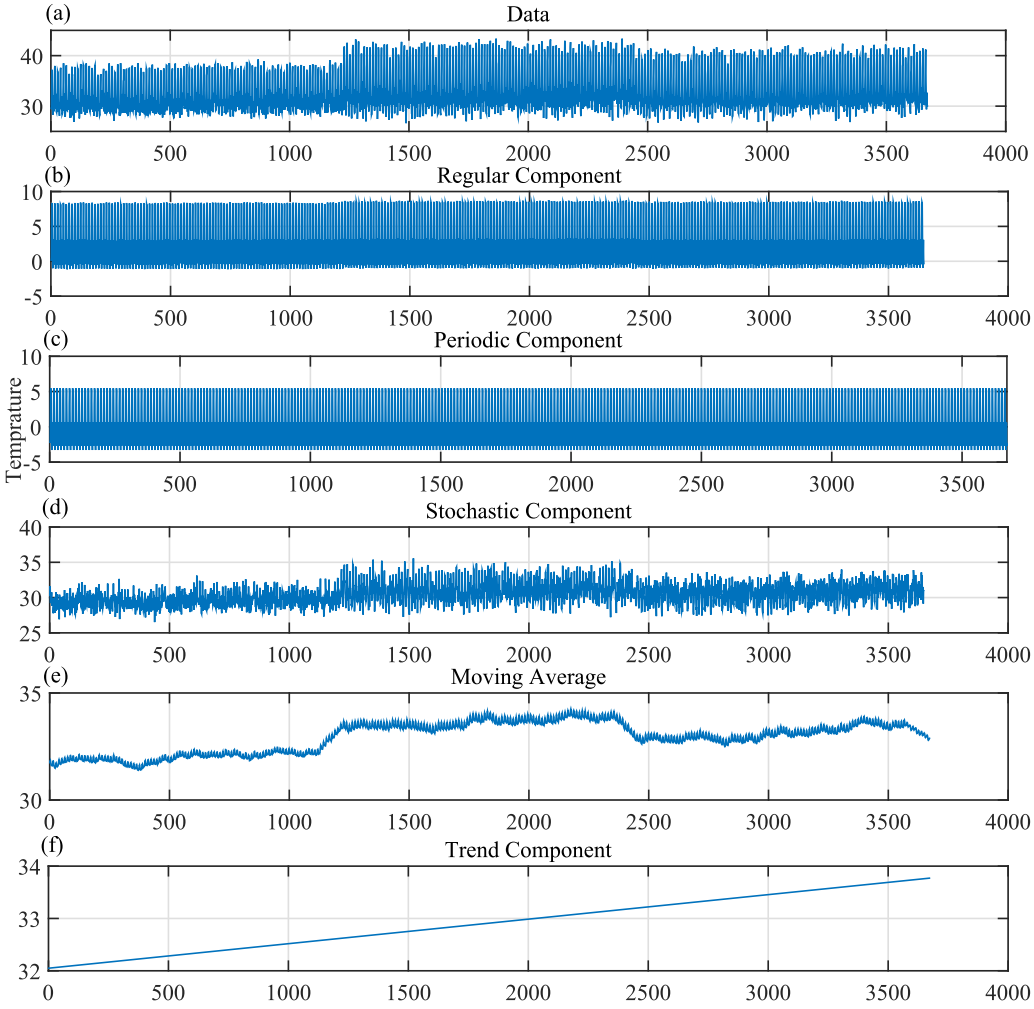

Fig. 6. Decomposed selected three stations data plotted linearly. (a) 3 weather stations maximum temperature data, (b) decomposed regular component, (c) decomposed periodic component, (d) decomposed stochastic component, (e) moving average and (f) linear trend component.

ber of nearest neighbors which is used to assign object to the class. When k =1 the single nearest training class is assign to the object which is treated as estimated value. The time series is distributed as (x, y) training sample and (x', y') as testing sample. The basic Euclidean distance (d) function is used evaluate similarity as,

$$d_i(x, y) = \sum_{j=1}^{n} \sqrt{(x_j - y_j)^2} \tag{8}$$

Where n is the total number of attributes, $d_i(x, y)$ represents current data point to the ith data point in historical data set. The x_j and y_j are the jth attribute in the current and historical data vector respectively. Therefore, y' is the predicted value which will evaluated using the majority class. The y' is evaluated as,

$$y' = arg\ max \sum_{(x_i,\ y_i)\ \in\ T_{train}} C(c = c_{y_i}) \tag{9}$$

where, C is the function to check condition, c is the class label and c_{y_i} is the class label for neatest neighbor object. A common prediction function is defined which can be used to predict both linear or nonlinear components.

In M-HM, each time the linear data value present in the sample vector is trained with new or existing class in the testing sample. If the arrival sample is not present in the testing data then the index is calculated depending on the index it will add in the training data set and re-trains of the algorithm is invoked. Due to this methodology, the unlabeled parameters are labeled and utilized for the forecasting of the new prediction values. Each time the training set is updated. How this algorithm work for the different components discussed as follows:

1. The strongly linear component is treated as unlabeled class samples and after verification of each data point, a new class is assigned. The linear property (positive or negative trend) of data is in increasing or decreasing order which will be used by M-HM to make new class each time. The labeled new classes are used as training for the future prediction of the data.
2. The seasonal and periodic components are easily handled, for that the ARIMA seasonal filter will help to forecast it.
3. The nonlinear components are defined in the fixed range and labeled each component without repetitions. If the element of testing set is out of range then it is labeled as new class and the algorithm re-trains the training set.

3. Results and discussion

3.1. Performance evaluation

In this study, the performance of the algorithm and system are tested using the standard measures. Accuracy of the algorithm is tested using the Mean Absolute Deviation (MAD), Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percent Error (MAPE) coefficients and the values are shown in the Table 2. Also, the performance of the

Algorithm 1 MapReduce based decomposition approach.

Map Class
Input: (*key* ← `name of the input`; *value* ← `value of the input`)
Output: (*key*, *value*)
Map (*key*, *value*)
$t_s \leftarrow$ `time series signal to decompose`, $n \leftarrow$ `lenght of` t_s, $d \leftarrow$ `time window`
$trend \leftarrow \hat{m}_t = (x_{t-q} + x_{t-q+1} + \ldots + x_{t+q} + x_{t+q+1})/d$, $q < t \leq n-q$, # $\hat{m}_t$ moving average trend and $q = d/2$ (Eq.4)
$\hat{p}_k = \begin{cases} w_k - d^{-1}\sum_{i=1}^{d} w_i, & k = 1, \ldots, d, \\ \hat{p}_{k-d}, & k > d, \end{cases}$ # periodic pattern (Eq.5)
`ramaining data` $\leftarrow t_s$ `- periodic pattern`
$r_t \leftarrow m_t + p_t$ # r_t regular component or seasonal component
$s_t \leftarrow t_s - r_t$ # s_t stochastic component
Intermediate: ('*key*', *value*) # add multiple values such as m_t, p_t, r_t and s_t
Reduce Class
Input: (*key* ← `name of the mapped data`; *value* ← `list of all map data`)
Output: `Key of the mapped data into row and column`
Reducer (*key*, *value*)
while values.hasNext() **do**
data = getnext(value)
Output: (*key*, *value*) # return the final output

Algorithm 2 M-ARIMA forecasting algorithm.

Map Class
Input: (*key* ← `name of the input`; *value* ← `value of the input`)
Output: (*key*, *value*)
Map (*key*, *value*)
`Initialize p, d and q values from the ACF and PACF analysis`
seasonality ← 12 # seasonality monthly average input values of the year
`ARIMA(p,d,q)`
`forecast(mdl, data)`
Intermediate: (*key*, *value*)
Reduce Class
Input: (*key* ← `name of the mapped data`; *value* ← `list of all map data`)
Output: `Key of the mapped data into row and column`
Reducer (*key*, *value*)
while values.hasNext() **do**
data = getnext(value)
Output: (*key*, *value*) # return the final output

Algorithm 3 M-HM model with M-KNN forecasting algorithm.

Map Class
Input: (*key* ← `name of the input`; *value* ← `value of the input`)
Output: (*key*, *value*)
Map (*key*, *value*) # Distance mapper
$T \leftarrow$ `time series`
stringlen ← `length of time series`
$k \leftarrow$ `number of nearest neighbors`# where, $k = 3$ used in this study
$T_{test} \leftarrow$ `testing sample`
$T_{train} \leftarrow$ `training sample`
if $T_{train} \neq T_{test} > \max(T_{train})$ **then**
  $T_{test} = T_{train}$ # labeling unlabeled objects
for `each testing sample` $T_{test} = (x', y')$ **do**
  `Compute` $d(x', x)$, `the distance between` T_{test} (Eq.8)
  `and each sample,` $(x, y) \epsilon\, T_{train}$
  `Select` $T_{train_{T_{test}}} \subseteq T_{train}$,
  `the set of k closest training samples to` T_{test}
Intermediate: (*key*, *value*)
Reduce Class
Input: (*key* ← `name of the mapped data`; *value* ← `list of all map data`)
Output: `Key of the mapped data into row and column`
Reducer (*key*, *value*) # Distance reducer
while values.hasNext() **do**
data = getnext(value)
Output: (*key*, *value*) # return the output
Map Class
Input: (*key* ← `name of the input`; *value* ← `value of the input`)
Output: (*key*, *value*)
Map (*key*, *value*) # Prediction mapper
if *neighbor(j, i)* == *k* **then**
  $pred(j, i) \leftarrow T_{test}(k)$ (Eq.9)
  goto *loop*.
  close;
Intermediate: ('*key*', *value*) # add multiple values such as m_t, p_t, r_t and s_t
Reduce Class
Input: (*key* ← `name of the mapped data`; *value* ← `list of all map data`)
Output: `Key of the mapped data into row and column`
Reducer (*key*, *value*) # Prediction reducer
while values.hasNext() **do**
data = getnext(value)
Output: (*key*, *value*) # return the final output

system is tested by evaluating the time required for the execution and large data handling capability. We have evaluated the performance of the proposed approach by the setup, metrics, accuracy, speed-up, scale-up and size-up.

1. *Setup:* Technically to test the big data, the parallel system cluster, 1 master machine and 4/8 working nodes with Intel i7 HD graphics 4600 3.40GH (12 Core) and 8.00 GB RAM are used. The experiments are performed on Ubuntu 16.04 LTS OS with Matlab R2016b and JDK.
2. *Metrics:* The following are the metrics were used to validate the effectiveness of the proposed algorithm in comparison with existing. The measures of effectiveness are,

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|X_t - \hat{X}_t|}{X_t}\right) \times 100\% \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\bar{X}_t - Y_t)^2} \tag{11}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|X_t - \hat{X}_t| \tag{12}$$

$$ME = \max_{t=1,\ldots,n}|X_t - \hat{X}_t| \tag{13}$$

3. *Accuracy:* The accuracy of the proposed approach is compared with prediction methods M-ARIMA, M-KNN using Margin of Error (MOE) metrics (see Table 1).
4. *Speed-up, scale-up and size-up:* To validate the speed-up property of approach, we have experimented on different size of cluster and machine configurations. We have holding 8 groups

Table 1
Performance measures.

Method	M-ARIMA				M-KNN				M-HM			
Parameter	MAD	MSE	RMSE	MAPE	MAD	MSE	RMSE	MAPE	MAD	MSE	RMSE	MAPE
Station-1 Data	0.8376	1.0864	1.0423	2.6097	0.01913	0.0010	0.0329	0.0001	0.0123	0.0005	0.0239	0.0364
Regular Component	0.0743	0.0096	0.0980	10.1247	0.0088	0.0009	0.0300	0.0234	0.0066	0.0007	0.0268	0.8001
Stochastic Component	0.8955	1.3853	1.1770	2.9767	0.0326	0.0243	0.1566	0.0001	0.0186	0.0099	0.0995	0.0569
Station-2 Data	1.0633	1.8610	1.3641	3.1503	0.0224	0.0011	0.0343	0.0001	0.0151	0.0005	0.2425	0.0442
Regular Component	0.0867	0.0137	0.1173	4.8002	0.0064	0.0001	0.0105	0.0080	0.0039	0.0001	0.0071	−0.2765
Stochastic Component	1.0700	1.9830	1.4082	3.4336	0.0131	0.0015	0.0390	0.0000	0.0089	0.0011	0.0336	0.0296
Station-3 Data	0.9027	1.3086	1.1439	2.7078	0.0234	0.0010	0.0322	0.0000	0.0153	0.0005	0.0227	0.0441
Regular Component	0.0802	0.0112	0.1060	9.8425	0.0069	0.0001	0.0120	0.0434	0.0036	0.0000	0.0072	−0.2307
Stochastic Component	0.9615	1.6187	1.2523	3.1132	0.0099	0.0002	0.0166	−0.0002	0.0055	0.0001	0.0116	0.0176

Table 2
The performance of system (execution time in seconds).

Workers	0.5 MB	1 MB	2 MB	5 MB	10 MB	20 MB	50 MB	100 MB
01	06.23	12.23	15.78	32.78	74.27	458.30	957.20	1301.00
04	05.96	08.35	11.54	22.70	118.42	383.67	526.20	847.90
08	02.05	06.95	09.22	19.67	22.83	41.47	183.46	223.00

Table 3
Statistical analysis of actual and predicted data by M-ARIMA, M-KNN and M-HM.

Station	Approach	t-value	Kurtosis	Skewness
Station-1	M-ARIMA	−0.62738	{−0.66871, −0.80469}	{0.705639, 0.731290}
	M-KNN	0.001106	{−0.66871, −0.68943}	{0.705639, 0.699391}
	M-HM	0.000553	{−0.66871, −0.68581}	{0.705639, 0.699926}
Station-2	M-ARIMA	−0.50658	{−0.56077, −1.19388}	{0.729063, 0.579152}
	M-KNN	−0.05501	{−0.56077, −0.56774}	{0.729063, 0.731338}
	M-HM	−0.00022	{−0.56077, −0.55871}	{0.729063, 0.726787}
Station-3	M-ARIMA	−0.43797	{−0.56687, −1.01969}	{0.770925, 0.625338}
	M-KNN	−0.01316	{−0.56687, −0.58163}	{0.770925, 0.764657}
	M-HM	−0.00986	{−0.56687, −0.55638}	{0.770925, 0.776288}

of data sets (0.5 MB, 1 MB, 2 MB, 5 MB, 10 MB, 20 MB, 50 MB and 100 MB) for the experiment purpose. Approximately, 1 MB of input data set have 12*100000 data points which are decomposed, evaluated and predicted by the proposed approaches. To check the scale-up property, clustered network is setup by varying nodes.

In Addition, the noticed different results are presented in the Tables 1 and 2. Mainly, for the testing and validation purpose performance measures and statistical tests are performed. The data points were tested firstly on the monthly maximum temperature of the 100 years which is used as training and 10 years (1993–2002) data as validation and testing purpose. The error values are separately noted and the performance measures such as MAD, MSE, RMSE & MAPE in the Table 1. The Table 2 shows the absolute and squared error values between the actual and predicted temperature using all three models. The statistical measures to check the performance of prediction models are presented in Table 3. The minimum squared error values are noted by the M-HM model which will show the accuracy in the prediction. The separately decomposed components prediction accuracy and comparison with each other are presented in the Table 2. From these results, we conclude that the prediction of the regular or seasonal component has more accuracy than stochastic component.

3.2. Prediction of regular component

In order to forecast regular component, we utilized proposed M-ARIMA, M-KNN, and M-HM models. The data have a regular component varying monthly, therefore seasonal ARIMA model with the general form of ARIMA is used to forecast. Also, using M-KNN and M-HM the regular component is forecast and compared with M-ARIMA. To evaluate the performance of the model, three different stations monthly maximum temperature data is used. Initially, the *p, d, q* parameters from the analysis of the ACF and PACF are analyzed. Then time series approach gives directions to the selection of the model after differences. The forecast results are categorized by two approaches. Firstly, direct use of input data to the model by estimating coefficients such as *p, d, q* by analyzing ACF and PACF. Secondly, each component present in the data such as general, stochastic or periodic are separately forecast using the all three models. In both approaches the same periodicity of the month, seasonal or year is applied.

The Fig. 6 shows the decomposed evaluation of three stations maximum temperature data as seasonal, periodic, trend and stochastic components using MapReduce decomposition method. The predicted different components for the same stations are separately depicted in the Fig. 7. The actual data and decomposed trend with predicted data and trend are shown. We noted the M-KNN and M-HM have more prediction accuracy than the M-ARIMA model. The linear trend component easily handled by the AR model or M-HM model. The M-KNN model not able to handle such component. Similarly, Fig. 7 shows the forecasting of the regular component present in the data which have more seasonality and the stochastic component prediction. The accurate prediction of the stochastic component is more important to analyze any future stochastic event occurrence. We noted M-ARIMA have low accuracy to forecast stochastic component which will accurately predict by M-KNN and M-HM. From this result analysis, the ARIMA approach and KNN approach can well be suitable for the forecasting weather parameters using Box-Jenkins approach and neural network for any weather station located in the different locations.

The Cumulative Distribution Function (CDF) of relative error is presented in the plot. Observing the results, we know that 100% of

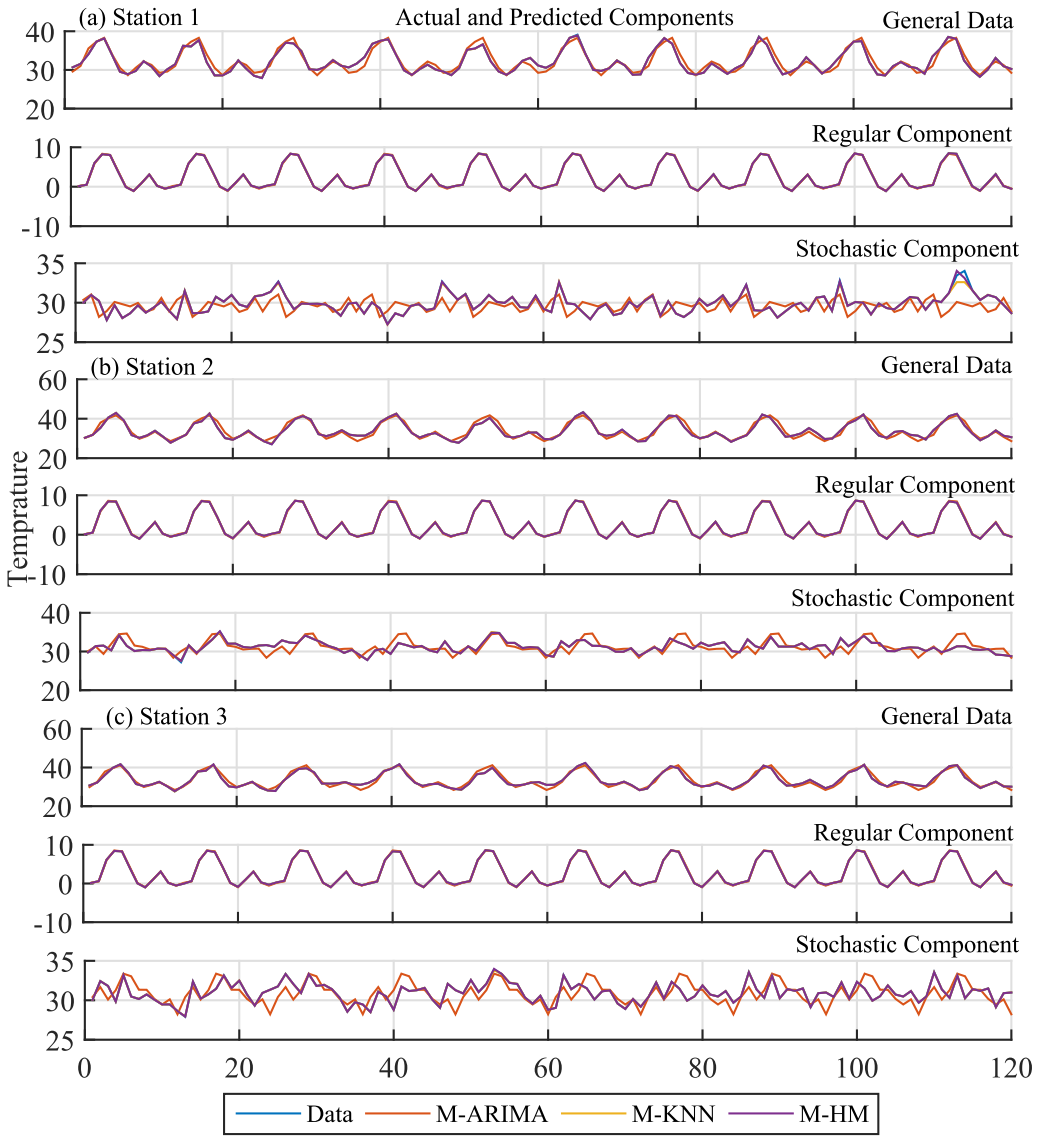

Fig. 7. Predicted 3 stations regular, stochastic and general components data using M-ARIMA, M-KNN and M-HM. (a) station-1 (Ahmednagar), (b) station-2 (Nanded) and (c) station-3 (Amravati).

the absolute value of the relative error is less than 15% (0.15 error value) in M-ARIMA, approximately 0.1% (0.00004 error value) in M-KNN and 0.1% (0.00006 error value) in the M-HM which suggest predicting results are accurate. The Fig. 8 (a)–(c) shows the CDF of the relative error between actual and predicted temperature using proposed approaches for three stations.

3.3. Prediction of stochastic component

To analyze stochastic components, the cross correlation between actual and predicted data is plotted (see Fig. 8 (d)–(f). The three different stations decomposed stochastic components are tested by cross correlation. There may possible variation in the stochastic data which will affect the monthly data. It may be possible to check stochastic changes that will happen on the particular day/month and affects the general situation by taken cross correlation of three stations data separately. We are checking if there any high cross-correlation which is a measure of similarity of two series as a function of the displacement of one relative to the other, in our case, it will check the relation between actual and predicted data values of all types of components. In our study, for the testing purpose, we have chosen 10 years (1993 - 2002) actual, regular and stochastic data. Cross correlation between these selected years was plotted for three stations and which are shown in Fig. 8 (d)–(f). Fig. 9 shows cross correlation boxplot using M-ARIMA, M-KNN and M-HM for actual, decomposed and predicted component.

Mainly, the predictive analytics of results are presented using the proposed MapReduce based forecasting model. In this work, three different configuration computing systems were used to handle and process data. The need of computing workers depends on the data size used and the processing algorithms. We have tested the performance of models on Intel i7 core with HD graphics 4600 processor by variation in the workers. The 1, 4, 8 set of workers are utilized to check the speed-up and big data handling capabilities over the parallel MapReduce based big data platform. Also, noted that decomposition and predictive approaches for data processing and prediction gives better results and minimizes the time required for the processing.

Table 1 illustrates the comparison of MAD, MSE, RMSE and MAPE values for the proposed model. From the Table 2, we conclude that the models are used in this study are able to give better predictions on temperature.

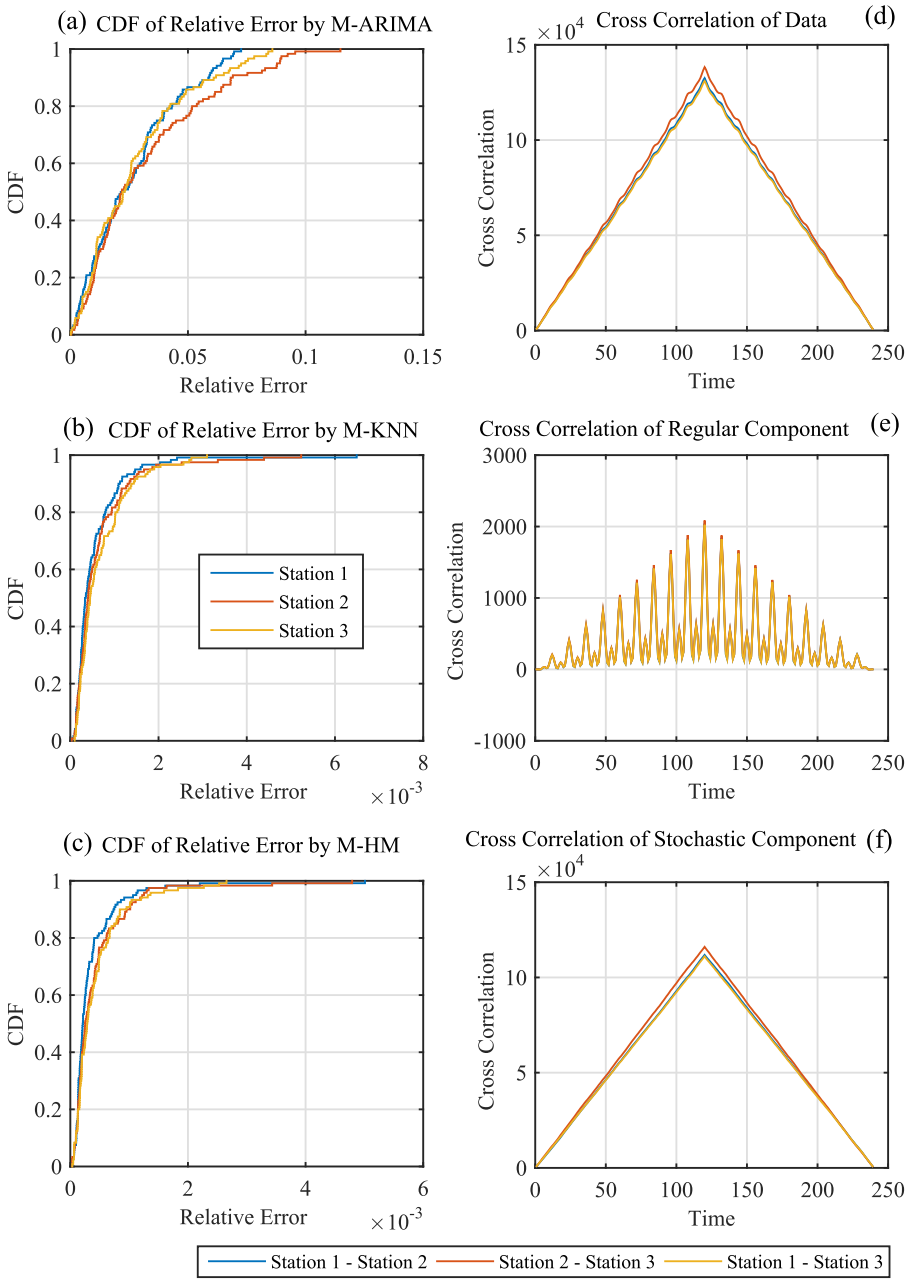

Fig. 8. CDF of relative error between actual and predicted data for station-1, station-2 and station-3 using proposed approaches. (a) M-ARIMA, (b) M-KNN, (c) M-HM, the cross correlation between three stations for the input (d) actual data, (e) regular component and (f) stochastic component.

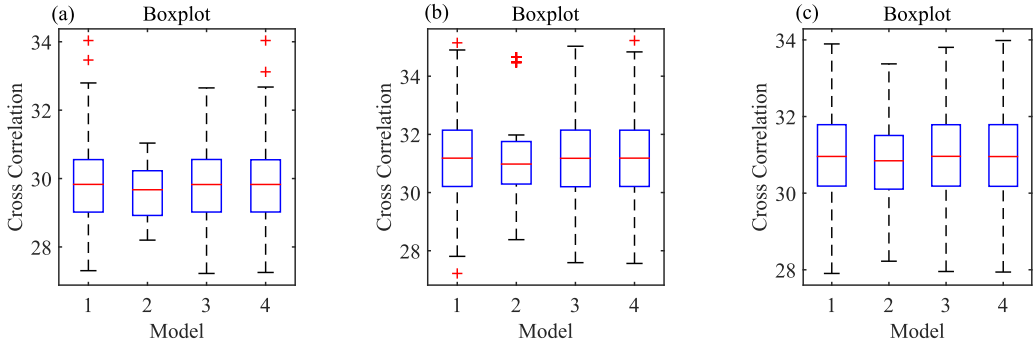

Fig. 9. Boxplot using M-ARIMA, M-KNN and M-HM for actual, decomposed and predicted component.

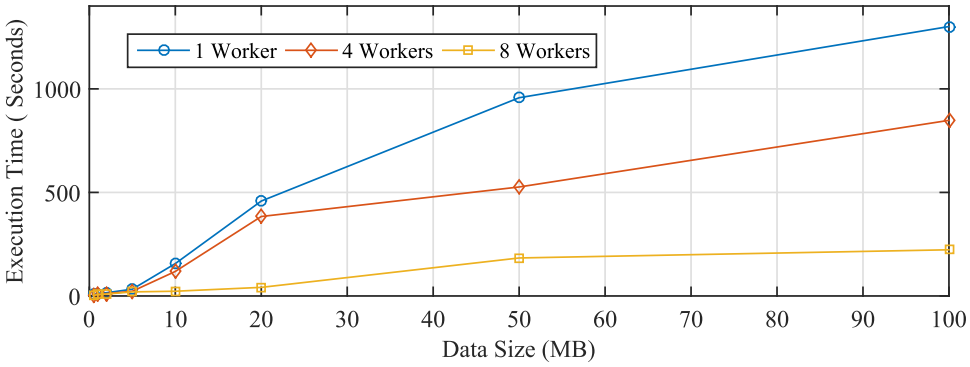

Fig. 10. Performance of decomposed approach over 1, 4, and 8 workers.

The best MSE, RMSE and MAPE achieved for the model are shown in the table. The Table 2 shows the calculated execution time by using the MapReduce based approach on single-node and multiple-node system by executing and distributed tasks parallel. We noted the performance of the system with parallel execution is approximately 90% greater than the single-node system. We analyzed the results and performance on system having a configuration (CPU-Intel i7 Core 3.40GH, Memory-8 GB, GPU-Intel(R) HD Graphics 4600) with different workers as shown in Table 2. The examined results are evaluated by changing the size of input data to the system and noted the best performance of the model.

In terms of time required for the execution and prediction accuracy, whenever number of nodes are proportional to the data size. To test the data size used in this study, all proposed approach are tested over MapReduce framework to handle a large amount of data and processes it quickly. The approach M-HM has a more accuracy as compared M-ARIMA and M-KNN algorithms utilized in this study. Whenever data size increases, performance of proposed approaches are better with respect to execution time. We experienced, the M-HM model takes 30% time for the execution as compared to M-KNN and 15% as compared to the M-ARIMA approach with same number of working nodes.

The *t*-test statistics were used to check the performance the proposed approaches (see Table 3). The squared values are noted which are minimal for the M-HM model as compared to other. The *t*-test: two sample test for an unequal variance is applied on actual and predicted temperature of the 10-year data. The t-value for the selected stations satisfies the condition of null hypothesis acceptance i.e $(+1.96 < t - value > -1.96)$ for all the station predictions. Statistical measures, kurtosis and skewness of actual and predicted values of temperature are used to check symmetry in data and tailed distribution of data. Lastly, the performance of the model tested using the different size of data sets having large number of data points. After applying this data set the performance is analyzed and compared on single-node and multiple-node system. The Fig. 10 shows the performance comparison of the algorithm with different size of data input on different systems. We noted that the MapReduce based algorithm produces better performance than simple model. The MapReduce based M-HM successfully predict the results and handle a large amount of data.

4. Conclusions and future work

Fast and accurate weather analytics is an important need for todays and tomorrows agriculture. The available huge data in precision agriculture is handled by suitable big data platform which is successfully integrated. In addition, the data is carefully preprocessed and decomposed for the weather analytics. To predict fast results, the power of systems and efficiency of the algorithm are to be considered. The parallel and distributed processing task achieved by using the MapReduece framework over big data platform using the parallel systems. The challenges such as the volume of data, velocity, and predictive analysis were addressed in this study. The proposed M-ARIMA, M-KNN, and M-HM models are successfully predicted the decomposed, regular and stochastic components. The predicted results are used to identify forecasting behavior of the weather stations. We conclude that the same models can be used to decompose, forecast and behavior analysis of the weather stations situated in the state. The proposed MapReduce based approaches are capable to give predictive analytics on data generated by weather stations and also able to handle increasing data. Using the predictions, analytics scientist can suggest how to avoid yield loss from disaster conditions within time using huge data. Also, noticed that time required for the execution using MapReduce approach is decreased half of the other comparative models executed on the same system. In future work various climatic variables can be used for the predictive analytics in the agriculture. In the end, predicted results would suggest various decisions to farmers for deciding the crop pattern and water management in the future.

References

Bharath, R., Srinivas, V. V., & Basu, B. (2015). Delineation of homogeneous temperature regions: A two-stage clustering approach. *International Journal of Climatology, 36*(1), 165–187.

Chen, S.-M., & Hwang, J.-R. (2000). Temperature prediction using fuzzy time series. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 30*(2), 263–275.

Ganguly, A. R. (2002). A hybrid approach to improving rainfall forecasts. *Computing in Science & Engineering, 4*(4), 14–21.

Li, H., Lu, K., & Meng, S. (2015). Bigprovision: A provisioning framework for big data analytics. *Network, IEEE, 29*(5), 50–56.

Mathew, A., Sreekumar, S., Khandelwal, S., Kaul, N., & Kumar, R. (2016). Prediction of land-surface temperatures of jaipur city using linear time series model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9*(8), 3546–3552.

Meng, S., Dou, W., Zhang, X., & Chen, J. (2014). Kasr: A keyword-aware service recommendation method on mapreduce for big data applications. *IEEE Transactions on Parallel and Distributed Systems, 25*(12), 3221–3231.

Miyoshi, T., Kondo, K., & Terasaki, K. (2015). Big ensemble data assimilation in numerical weather prediction. *Computer, 48*(11), 15–21.

Sakr, S., Bajaber, F., Barnawi, A., Altalhi, A., Elshawi, R., & Batarfi, O. (2015). Big data processing systems: State-of-the-art and open challenges. In *2015 international conference on cloud computing (ICCC)* (pp. 1–8).

Shang, W., Jiang, Z. M., Hemmati, H., Adams, B., Hassan, A. E., & Martin, P. (2013). Assisting developers of big data analytics applications when deploying on hadoop clouds. In *Proceedings of the 2013 international conference on software engineering* (pp. 402–411). IEEE Press.

Sharma, K. L. S., & Mahalanabis, A. K. (1974). Modeling and prediction of the daily maximum temperature. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-4*(2), 219–221.

Xing, E. P., Ho, Q., Dai, W., Kim, J. K., Wei, J., Lee, S., et al. (2015). Petuum: A new platform for distributed machine learning on big data. *Big Data, IEEE Transactions on, 1*(2), 49–67.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35–62.