International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 8
ISSN: 2454-4248
65 – 73

PAFHWKM: An Enhanced Parallel Approach to Forecast Time Series Data Using Holt-Winters and K-Means Algorithm

B. Arputhamary
Research Scholar,
Mother Teresa Women's University,
Kodaikanal
E-mail: arputhambaskaran@rediffmail.com,
Mobile: 9715179752

Dr. L Arockiam
Associate Professor,
St.Joseph's College,
.Tiruchirappalli.
E-mail: larockiam@yahoo.co.in
Mobile: 9443905333

***Abstract*—**Big Data is a recent research style which brings up challenges in decision making process. The size of the dataset turn intotremendously big, the process of extracting valuablefacts by analyzing these data also has become tedious. To solve this problem of information extraction with Big Data, parallel programming models can be used. Parallel Programming model achieves information extraction by partitioning the huge data into smaller chunks. MapReduce is one of the parallel programming models which works well with Hadoop Distributed File System(HDFS) that can be used to partition the data in a more efficient and effective way. In MapReduce, once the data is partitioned based on the <key, value> pair, it is ready for data analytics. Time Series data play an important role in Big Data Analytics where Time Series analysis can be performed with many machine learning algorithms as well as traditional algorithmic concepts such as regression, exponential smoothing, moving average, classification, clustering and model-based recommendation. For Big Data, these algorithms can be used with MapReduce programming model on Hadoop clusters by translating their data analytics logic to the MapReduce job which is to be run over Hadoop clusters. But Time Series data are sequential in nature so that the partitioning of Time Series data must be carefully done to retain its prediction accuracy.In this paper, a novel parallel approach to forecast Time Series data with Holt-Winters model (PAFHW) is proposed and the proposed approach PAFHW is enhanced by combining K-means clusteringfor forecasting the Time Series data in distributed environment.

Index Terms*—Big Data, Big Data Analytics, Time Series Models, Parallel Processing, Hadoop, MapReduce.*

******

I. INTRODUCTION

Big Data is used for large data sets whose size is beyond the cutting edge and cannot be processed with the ability of commonly used software tools. Also it is difficult to capture, manage and process the Big Data within a tolerable elapsed time[9]. Big Data sizes are constantly increasing from a few dozen terabytes to many petabytes.In 2010, Apache Hadoop defined Big Data as datasets which could not be captured, managed and processed by general computers within an acceptable scope [1]. Gartner defined Big Data with 3 V's model: Volume, Velocity and Variety. Volume describes the generation and collection of masses of data as well as the data scale which becomes increasingly big. Velocity is the timeliness of Big Data and Variety means various types of data which includes semi structured and unstructured data such as audio, video, web page and text [2]. Nowadays data are produced in an extraordinary manner. These data are generated through many sources such as web logs, social networks (Blogs, Comments and Likes), transactional data sources and sensor data. The data obtained through various sources are heterogeneous in nature[7]. Due to its nature, Big Data has emerged various challenges in the decision making process[20]. These days organizations are struggling in capturing, storing and analyzing these huge volume of data to increase the accuracy of decision making[10][18]. Storing these voluminous data does not pose much problem but the effective utilization of these stored data is another challenge focused today. The challenges like scalability, unstructured data accessibility, real time analytics, fault tolerance and many more are handled by traditional approaches which have proved to be less efficient. To increase the efficiency Massively Parallel Processing (MPP) databases are required[11][16]. By using this environment, timely prediction with an increased accuracy can be achieved, which is the need of the hour. In this paper, a novel parallel approach to forecast Time Series data with Holt-Winters model (PAFHW) is proposed and the proposed approach is enhanced by combining the PAFHW with K-means for forecasting the Time Series data in distributed environment.

II. RELATED WORKS

Big Data Analytics involves large scale computations that use huge volume of input data that is often in the order of terabytes or petabytes and are run in parallel with multiple data centers involving tens of thousands of machines [1]. The performance of data parallel computing such as MapReduce and DryadLINQ are highly dependent on its data partitions. A

key factor to make such computations efficient is to partition the data evenly across multiple data centers [2]. Range partition is one of the ways to partition the data that is needed whenever global ordering is required[3].Jeffrey Dean et al.,[4], have proposed MapReduce, which is a parallel programming model that runs on a large cluster of commodity machines and is highly scalable environment. A typical MapReduce computation processes many terabytes of data on thousands of machines. Aditi Jain et al.,[5] proposed data driven traffic flow forecasting system which is based on MapReduce framework for distributed system with Bayesian network approach. This approach mainly focused on the problem in distributed modeling for data storage and processing in traffic flow forecasting system. One of the drawbacks of MapReduce is that multiple jobs may be required for some complex algorithms, which limits load balancing efficiency[6]. The paper has proposed a systematic, time-series based scheme to perform prediction using the Hadoop framework and Holt-Winter prediction function in the R environment to show the sales forecast for forthcoming years [11]. In parallelizing Time Series analysis algorithms, the training process requires a global ordering [8]. Analysis of prediction techniques in Time Series analysis and the adaptability in Big Data Environment are discussed [12].

III. PAFHW ALGORITHM

A Parallel Approach to Forecast Time Series data using Holt-Winters (PAFHW)is proposed to simplify the process of forecasting the Time SeriesBig Data. Parallel algorithms play an important role in Big Data environment [19]. To achieve this, the traditional algorithms are converted into parallel algorithms to capture and manage huge amount of data [12]. In Time Series data, limited research has been undertaken with parallel processing. Because, the Time Series data are sequential in nature which means manipulation of current year require the availability of data in previous years[13-14].

Parallel algorithms can be proposed by partitioning the input data into small chunks and run the input chunks concurrently [17]. But, partitioning the input data in Time Series analysis may cause problems in prediction due to the dependency exists in Time Series data. Therefore, partitioning of Time Series data should by carefully done to retain the prediction accuracy. In this proposed algorithm, parallelism is achieved by partitioning the input data on the seasonal basis and the Time Series model is used for forecasting. And this approach is well suited for the data sets with seasonal components. From the empirical analysis, it is clear that Holt-Winters model is well suited for the data which has seasonal fluctuations [15]. For the study, Capital Bikesharing System(CBS) is considered and the PAFHW algorithm is tested for its prediction accuracy. The following algorithm 1 is used to forecast time series data with Holt-Winters model. The proposed algorithm follows MapReduce programming model which can be used with Big Data ecosystems such as Hadoop framework.

Algorithm 1: Parallel Approach to Forecast Time Series Data using Holt-Winters (PAFHW)

Input: *Training dataset T, Attribute Ak(is a Time Series data for N years)*

Output: *Forecasting values for next Nyears*

Step 1: *Start the process*

Step 2: *If* T *is NULL then*
Return failure
End if

Step 3: *Initialize a Hadoop Job SplitJob*

Step 4: *Set SplitTaskMapper as Mapper*

Step 5: *Adjust the block size of HDFS until the dataset T can be split into m subsets* $T_j = \{T_i\}_{i=1}^{N}$

Step 6: *Set SplitTaskReducer as the Reducer Class*
Input<key, Value>=<id, T_{id}*> where* X_{id}*is the set of List(T)*

Step 7: *// Prediction*
Call Holt-Winters

Step 8: *Visualization*

Step 9: *Stop the process*

The general steps in the proposed PAFHW are explained below.

Step 1: Start the process

Step 2: Read input from T. If T is NULL then return failure. In Figure 1 the input data are loaded.

Figure 1: Loading data

The data set used for this work is taken from a usage log of bikes being rented in a bike sharing system in Washington, USA. This is also known as Capital Bike Sharing (CBS) that are collected from different sources such as city bike website and weather API. The collected data is available in SQL format in MySQL Server. Hadoop framework is a cost effective and faster Big Data processing unit, which can be used to enhance the analysis process of Big Data. Also it is an open-source software framework for storing data and running applications on clusters of commodity hardware which provides massive storage for data and enormous processing power. MapReduce is a parallel programming model for writing applications that

 ISSN: 2454-4248
65 – 73

process large amounts of structured and unstructured data in parallel across a cluster of thousands of machines, in a reliable and a fault-tolerant manner. Similarly Hadoop Distributed File System (HDFS) provides reliable data storage and access across all the nodes in a Hadoop cluster. It links together the file systems on many local nodes to create a single file system. The dataset need to be formatted and uploaded to Hadoop Distributed File System (HDFS) and are further used by various nodes with Mappers and Reducers in Hadoop clusters.

Step 3: Split the input data T into multiple data chunks and each chunk is stored at Hadoop Distributed File System (HDFS) whose size ranges from 64 MB to 512 MB by default. Here n is the number of chunks, size of each chunk ranges from 64 MB to 512 MB. By default Hadoop follows hash partitioning to distribute the input data into HDFS. The process of distributing input data into HDFS is given in Fig.2.

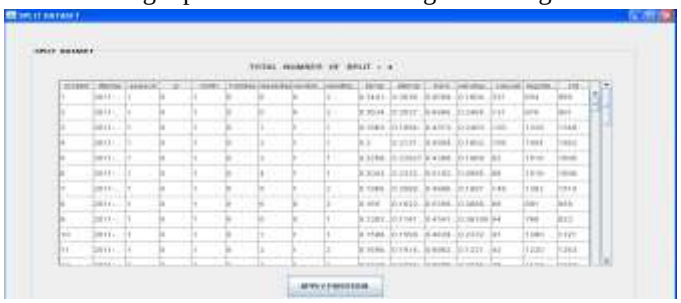

Figure 2: Data stored in HDFS

Data pre-processing allows transforming the data into a suitable format that can be used as input for our forecasting model. Several data pre-processing methods are available in data mining field which can be used to do this task. These include data cleaning, data integration, data transformation, data reduction, data modelling, path completion, user and session identification. In Hadoop environment the dataset are managed and pre-processed by Apache Hive. Hive provides a warehouse structure and SQL like access for data in Hadoop Distributed File System (HDFS). Now, the data which is required for the analysis is in required format and available in Hadoop Distributed File System (HDFS). After pre-processing the data analytics process will be initiated with this formatted data as the input.

Step 4: For each data chunk do the following steps repeatedly, until all the records are retrieved.

Step 4.1: The input data are split into data chunks and are assigned to the map tasks from the HDFS. The Map function takes an input pair and produces a set of intermediate key-value pair. In the proposed work, season attribute in Capital Bikesharing System (CBS) dataset is considered as a key. Then the MapReduce library groups all intermediate values associated with the same key season and passes them to the Reduce function. The process of mapper function is described in Algorithm 1(a).

Step 4.2: The Reduce function accepts an intermediate key and a set of values for that key and merges them into a subset. And the Reduce function iterates the set and produces key-value pair as output and rewrites the generated key-value pairs to the system. In Algorithm 1(b), the process of reducer function is explained.

Step 5: Once the data is getting available in the required format for data analytics, data analytics operations will be performed using algorithms. The data analytics operations are performed for discovering meaningful information from the data to take better decisions towards performance. It may either use descriptive or predictive analytics for the rented bikes evaluations. In this research, predictive analytics is considered to perform data analytics. In this work, the data in each split which are in HDFS is further divided on a seasonality basis with the map function Map(Range(Month(Date)),Value). Each split comprises of the data with its season value so that the entire data is split into 4 seasons where S_j comprises of data about season 1 and it is spring season. Similarly the data are grouped as season 2, season 3 and season 4 that represents summer, fall and winter respectively. Once the data is partitioned by the Mapper, intermediate data are brought forth which are shuffled and sorted. Now the input data that are partitioned by the mapper function are aggregated at the reducer side and are ready for seasonal forecasting. Model fitting is the essential step in forecasting. From the empirical study, Holt-Winters model is well suited for the Capital Bike Sharing (CBS) dataset. Therefore Holt-Winters algorithm can be applied to forecast the values for the forthcoming years. At this stage, Holt-Winters (HW) algorithm is used to forecast the demand of bike users for the forthcoming years. Since the data is split into small chunks on seasonal basis, Holt-Winters model performs well and gain its accuracy in forecasting level. For this R statistical package is used with Hadoop.

Step 6: Visualize the forecasting values. The final stage of the process consists of visualization of the results of data analytics. Visualization is an interactive way to represent the data insights. This can be done with various data visualization software. For visualization, the statistical package R is combined with Hadoop and the demands of the bikes for the forthcoming years are forecasted.

Step 7: Stop the process

The Mapper function in the PAFHW algorithm is used to split the input data into smaller chunks and which will ensure parallelism at data level. The Map() and Reduce() functions are the two important functions in MapReduce programming. The following algorithm 1(a) explains the working procedure of the Mapper function which applies range partitioning on the <key, value> pair and split the input data into four seasons.

Algorithm 1(a): Map (Range(String key), String value)

Input: $T_j = \{T_i\}_{i=1}^{N}$ *where* T_i *is the* i^{th} *instance with n attributes* $\{A_k\}_{k=1}^{n}$

Output : *<key, value> = < id,* T_i *>, where id is the label of output subset*

Step 1: Begin Mapper
Step 2: In the j^{th} *split task mapper*
Step 3: Parallel for each instance T_i *(i = 1,2,3, ……, N) do*
if(A_k *==1) then*
id ←1
else if(A_k *==2) then*
id ←2
else if(A_k *==3) then*
id ←3
else
id ←4
End for
Step 4: Output : <key,value>=<id, T_i*>*
Step 5: End Mapper

The general steps of the Mapper( ) function are given below:
Step 1: Start the Mapper
Step 2: For each mapper, read input record with <key, value>. In the proposed algorithm, key is the date and value is seasonal value. The seasonal values for the given dataset are 1 which represents Season 1 as spring, 2 as summer, 3 as fall and 4 as winter. The mapper will generate list of records with season 1, season 2, season 3 and season 4 separately for each Hadoop Distributed File System (HDFS).
Step 3: Take date as key and construct the list of values
Step 4: Return list of keys with corresponding set of values
Step 5: Stop the Mapper
The Reducer () function will get the output of the Mapper () function as input and aggregate the list of records that are having same key. The following algorithm 1(b) describes the working procedure of the Reducer () function.

Algorithm 1(b): Reducer (Range(String key), Iterator values)

Input: *<key, value> = <id,* T_{id}*> where* T_{id} *is the set of List(T)*
Output : *<key, value>= <id,* T_{id}*>*

Step 1: *Start Reducer*
Step 2: *Shuffling and sorting of* T_{id} *is performed where id is from 1 to 4*
Step 3: *List of records with similar seasons are aggregated at reducer side*
Step 4: *End Reducer*

The algorithmic steps of the algorithm 1(b) is given below:
Step 1: Start the process
Step 2: The records with same key values are shuffled and sorted.
Step 3: List of records with same key in all the mappers are collected and aggregated.
Step 4: End Reducer

Finally, the Holt-Winters algorithm is used to make prediction. There are various number of algorithms available for forecasting. Holt-Winters algorithm is well suited for the dataset which has seasonal fluctuations. The following algorithm 1(c) shows the procedure of the Holt-Winter (HW) algorithm.

Algorithm 1(c): Holt-Winters()

Input: *Set of all records at reducer of particular season*
Output: *Forecasting values for the next n years at a particular season*

Step 1: Start the process
Step 2: for each M_x *do*
Step 2.1: Calculate $a_t = \alpha((Y_t - s_{t-p}) + (1-\alpha)(a_{t-1} + b_{t-1}))$
Step 2.2: Calculate $b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$
Step 2.3: Calculate $s_t = \gamma(Y_t - a_t) + (1-\gamma)s_{t-p}$
Where α, β and γ lies between 0 and 1
Step 2.4: $F_t \leftarrow a_t + b_t + s_t$
Return F_t
End for
Step 3: Stop the process

The steps of the algorithm 1(c) is given,
Step 1: Start the process
Step 2: For each data chunk do the following steps.
Step 2.1: Calculate the level with N set of historical data.
Step 2.2: Calculate the trend
Step 2.3: Calculate the Season
Step 3: Find the forecasted value by adding the level, trend and Season.
Step 4: Stop the process

The following Table1 gives the error rates of HW and PAFHW with respect to Root Mean Square(RMSE), Mean Absolute Percentage Error (MAPE) and the Mean Absolute Square Error (MASE). From the experimental results RMSE, MAPE and MAPE rates of PAFHW is lesser than the sequential version of the HW. The proposed PAFHW follows MapReduce programming model that does not require global ordering which is present in the existing approaches.

 ISSN: 2454-4248
65 – 73

Table 1: Comparison on HW and PAFHW

Model	RMSE	MAPE	MASE
HW	809.1993	36.619595	4.4904944
PAFHW	685.65875	21.369595	2.4152944

And Figure 3 Gives the pictorial representation of the error rates of HW and PAFHW.

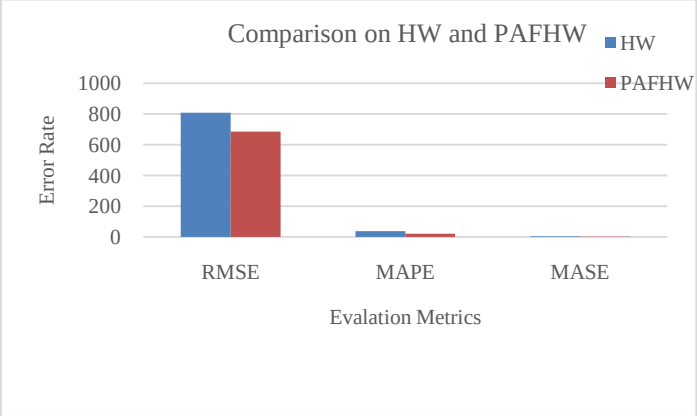

Figure 3: Comparison on HW and PAFHW

The following Figure 4 depicts the forecasting of PAFHW algorithm. The forecasting of next year data is done with the help of the previous year's data with same season value.

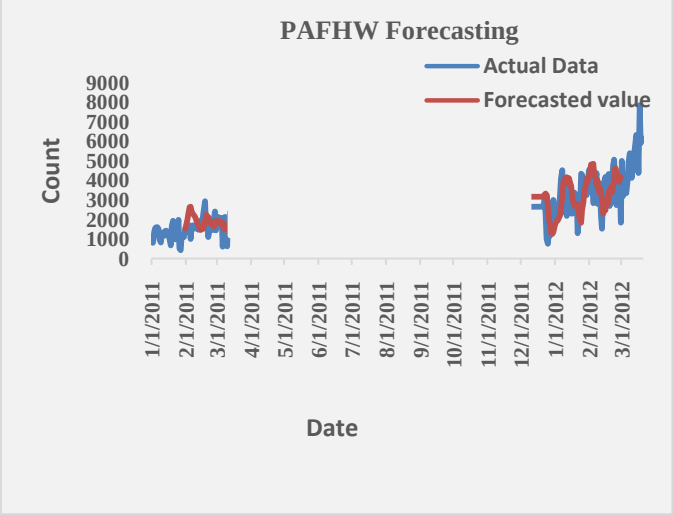

Figure 4: Forecasting of PAFHW

The PAFHW algorithm is implemented in Hadoop MapReduce environment in order to test the forecasting accuracy and the performance. The proposed algorithm proves its significance and which well suited for the Time SeriesBig Data with seasonal component. In this work, outlier deductions are not considered which can improve prediction accuracy if it is deducted and removed properly. Therefore, an enhanced algorithm is proposed by combining K-means clustering algorithm with PAFHW and discussed in the following section.

IV. PAFHWKM ALGORITHM

The Parallel Approach for Forecasting the Time Series data with Holt-Winters and the K-means (PAFHWKM) algorithm is the extended version of the PAFHW algorithm which is discussed in the previous section. The proposed PAFHWKM algorithm utilizes K-means clustering approach to improve the prediction accuracy. K-means clustering is a type of unsupervised learning, which is used when unlabeled dataset are used. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. Based on the features that are provided, the algorithm works sequentially to assign each data point to one of the K groups. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are:

1) The centroids of the K clusters, which can be used to label new data.

2) Labels for the training data (each data point is assigned to a single cluster)

Parallel Holt-Winters Algorithm (PAFHW) is offered to streamline the process of forecasting by parallelizing the Time SeriesBig Data. The proposed algorithm follows MapReduce programming model which can be used with Big Data echo systems such as Hadoop framework. The proposed algorithm retains its accuracy whereas the partitioning of data did not affect the prediction accuracy when compared with sequential prediction. But in the proposed algorithm, outliers present in the dataset are not considered which may result in less prediction accuracy. Hence, K-means algorithm is considered in this paper to remove the outliers and also to identify the large clusters.

K-means clustering is the famous algorithm to find clusters with centroid points. In this section, an improved algorithm is proposed with K-means clustering, which generates number of clusters. In Big Data world the size of the data is a primary problem and by using this proposed improved algorithm outliers can be removed and also the large clusters can be taken into consideration for prediction by leaving the small clusters with limited number of transactions.

Algorithm 2: Parallel Approach to Forecast Time Series Data using Holt-Winters and K-Means (PAFHWKM)

Input: *Training dataset T, Attribute Ak(is a Time Series data for N years)*

Output: *Forecasting values for next N years*

Step 1: *Start the process*

Step 2: *If* T *is NULL then*
Return failure
End if

Step 3: *Initialize a Hadoop Job SplitJob*

Step 4: *Set SplitTaskMapper as Mapper*

Step 5: *Adjust the block size of HDFS until the dataset T can be split into m subsets* $T_j = \{T_i\}_{i=1}^{N}$

Step 6: *Set SplitTaskReducer as the Reducer Class*
Input<key, Value>=<id, T_{id}> where $X_{id}$is the set of List(T)

Step 7: *//Mapper for clustering with centroid points*
Call Map (String key, String Value)

Step 8: *//Reducer for calculating the average of centroids*
Call Reducer (String key, Iterator values)

Step 9: *// Prediction*
Call Holt-Winters

Step 10: *Visualization*

Step 11: *Stop the process*

The general steps in the proposed Algorithm 2: PHWKMA are explained below:

Step 1: Start the process

Step 2: Read input from T. If T is NULL then return failure.

Step 3: Otherwise split the input data into multiple data chunks where x is from 1 to n. Each chunk is stored at Hadoop Distributed File System (HDFS) whose size ranges from 64 MB to 512 MB by default. Here n is the number of chunks, size of each chunk range from 64 MB to 512 MB

Step 4: For each data chunk do the following steps repeatedly, until all the records are retrieved.

Step 4.1: The mapper is invoked which will segregate the input records based on the <key, value> and output <list (keys), value> to the reducer. The process of mapper function is described in Algorithm 2(a).

Step 4.2: The data chunks generated by the mapper function are aggregated based on the key. Then, each reducer is assigned with list of keys with value. In Algorithm 2(b), the process of reducer function is explained.

Step 4.3: Here, the mapper reads the data and centroids from the disk. Then the instances are assigned to the clusters by the mappers.

Step 4.4: Once the mapper completes its operation, the reducer computes the new centroids by calculating average of data points present in the clusters. And the new centroids are written to the disk and are invoked by the mapper for the next iteration and this process is repeated.

Step 5: Then apply Holt-Winters algorithm for the large cluster to forecast the values for the forthcoming years.

Step 6: Visualize the forecasting values.

Step 7: Stop the process.

The mapper and reduce functions of step 4.1 and 4.2 in the PHWKMA algorithm is used to split the input data into smaller chunks , aggregate and are written into the disk which will ensure parallelism at data level. Another mapper and reduce process in step 4.3 and 4.4 are used to find the centroid points of each data chunk generated by the mapper and for generating clusters. The following algorithm 2(a) explains the working procedure of the Mapper and Reducer of K-means clustering to make clusters.

Algorithm 2(a): Map (String key, String Value)

Input: *Global variable centers, the offset key, the sample value Data points D, number of clusters K and centroids*

Output: *Centroids with associated Data points D*

Step 1: Begin Mapper
Step 2: Read the data and centroids from the disk and assigns them to clusters
Step 3: Construct the sample instance from value
Step 4: MIN_DIST = Double.MAX_VALUE;
Step 5: count = -1
Step 6: for i=0 to centers.length do
Distance = ComputeDist(instance, centers[i]);
If Distance < MIN_DIST
Then assign Distance to MIN_DIST
End for
Step 7: End Mapper

The general steps of the Algorithm 2(a) are given below:

Step 1: Begin Mapper

Step 2: The input data that is split and globally broadcast to the mappers are partitioned into smaller chunks with <season, value> pair and which is available in Hadoop Distributed File System (HDFS). For each map task, K-means construct a global variable centers which is an array of values about the centers of the cluster.

Step 3: Construct the sample instance from value

Step 4: Assign MIN_DIST as Double.MAX_VALUE.

Step 5: Assume and Initialize the Index variable as -1.

Step 6: Compute Distance and compare it with MIN_DIST. If Distance is less than the MIN_DIST, then assign Distance to MIN_DIST.

Step 7: Consider Index as key and construct values which are strings comprises of the values of different dimensions.

Step 8: Output is the number of clusters with <key, value> pair which is passed to the reducer. Here, the centroid point is generated for the temp variable. Therefore on each season, number of clusters are created based on the temp.

Step 9: End Mapper

The Reducer() function will get the output of the Mapper() function as input and generate the list of records that are having same key. In this work, on the mapper side clusters are generated based on the temperature and are transferred to the reducer. The reducer will aggregate the clusters based on the centroid points and will leave the outliers which are far from the centroid point. And finally the clusters of large sizes are considered for the data analytics while leaving the clusters of small sizes. Big Data has raised many challenges in decision-making process, the size of the data gets more attention than the other dimensions. Therefore the outliers can be ignored and

 ISSN: 2454-4248
65 – 73

also clusters with small sizes can be given least weightage for processing. The following section describes Algorithm 2(b), the working procedure of the Reducer () function.

Algorithm 2(b): Reducer (String key, Iterator values)

Input: *List of Keys with the value of Centroid points (Centroids with associated data points)*
Output : *<key, value> pair, where key is the Centroid point and values are the list of records*

Step 1: Begin Reducer
Step 2: Compute the new centroids by calculating the average data points in clusters
Step 3: Write the global centroids to the disk
Step 4: End Reducer

The algorithmic steps of the Reducer() function is given below:
Step 1: Begin Reducer
Step 2: Compute new centroids by calculating the average of data points in each cluster.
Step 3: New centroids are written to the disk which is used by the mapper for the next iteration until k clusters are generated.
Step 4: End Reducer
Finally, the Holt-Winters algorithm is used to make prediction. For prediction, only the large clusters are utilized which makes the decision-making process easy.

V. EXPERIMENTAL SETUP AND RESULTS

The proposed improved PAFHWKM algorithm is implemented and experimented in Hadoop framework. The data set used for this work is taken from log of bikes being rented in a bike sharing system in Washington, D.C., USA. This is also known as Capital Bike Sharing (CBS) that are collected from different sources such as city bike website and weather API. The collected data is available in SQL format in MySQL Server. Since it contains petabytes of data, these dataset are considered as Big Data.

Hadoop framework is a cost effective and faster Big Data processing unit, which can be used to enhance the analyzing process of such Big Data. Also it is an open-source software framework for storing data and running applications on clusters of commodity hardware which provides massive storage for data and enormous processing power. MapReduce is a parallel programming model for writing applications that process large amounts of structured and unstructured data in parallel across a cluster of thousands of machines, in a reliable and a fault-tolerant manner. Similarly Hadoop Distributed File System (HDFS) provides reliable data storage and access across all the nodes in a Hadoop cluster. It links together the file systems on many local nodes to create a single file system. The dataset need to be formatted and uploaded to Hadoop Distributed File System (HDFS) are further used by various nodes with Mappers and Reducers in Hadoop clusters. The collected dataset are uploaded to Hortonworks Data Platform (HDP) for analysis, using a tool SQOOP which is designed to transfer the data between Hadoop and relational database servers. Finally, R statistical package is integrated with Hadoop to perform analytics.

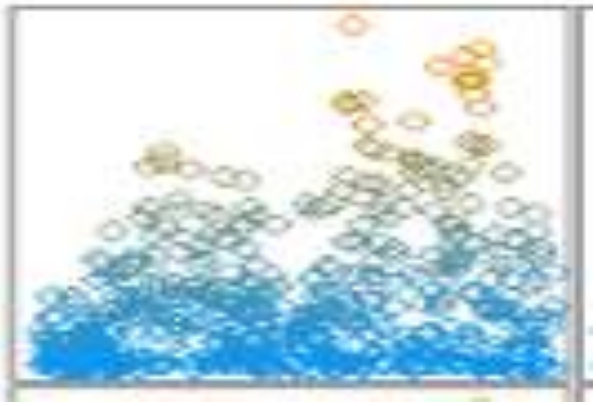

Figure 5: Cluster Generation

The total count of bike users for the different temperature for season 1 is taken and are plotted in the above Figure.5. In the improved algorithm, the large clusters are considered more for analytics. Similarly the small cluster is the one which has only limited demands so that which can be ignored or given less weightage for analytics.

The clustered output of the dataset taken is given in the Figure.5. It consists of 2 clusters which contain all the dataset and partitioned them into different clusters. Therefore the records which belongs to the large clusters of season 1 is considered and Holt-Winters(HW) model is applied. The actual value of the bike users' count is compared with the forecasted value. Table 2 gives the comparison of the PAFHW and PAFHWKM algorithm with respect to the forecasting accuracy. Forecasting accuracy is measured by using Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Square Error (MASE). The experimental results show that PAFHWKM perform better than PAFHW and retains its prediction accuracy. The following Figure 6 shows the forecasting of bike counts using PAFHWKM algorithm.

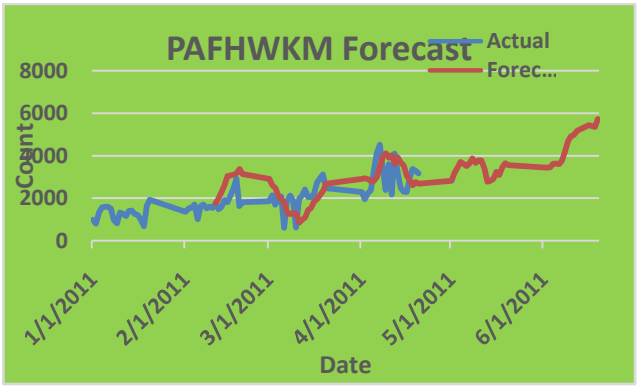

Figure 6: Forecasting of PAFHWKM

The following Table 2 gives the comparison on PAFHW and PAFHWKM forecasting with respect to RMSE, MAPE and MASE. The error rates of PAFHWKM algorithm is lesser than PAFHW. It is proved that, the enhanced PAFHWKM algorithm improved its forecasting accuracy by deducting and eliminating the outliers with K-means clustering and by finding the large clusters the size complexity of handling Big Data is reduced.

Table 2: Comparison on Forecasting Accuracy

Model	RMSE	MAPE	MASE
PAFHW	660.0385	21.369595	2.41529442
PAFHWKM	616.1719	19.952415	2.3332241

The following Figure 7 depicts the comparison on PAFHW and PAFHWKM.

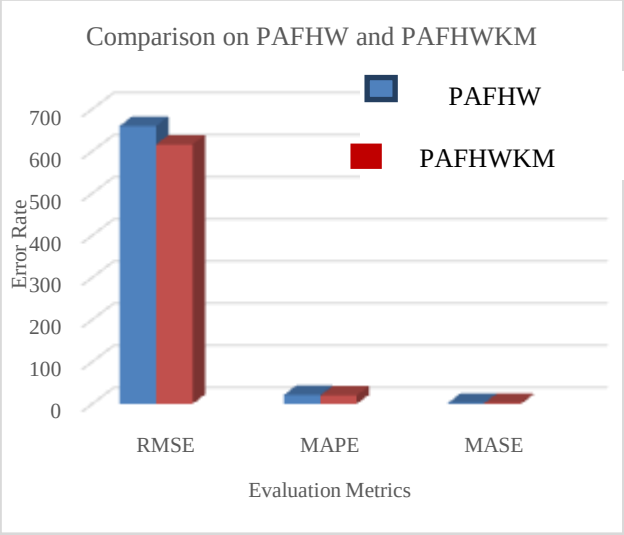

Figure 7: Comparison on PAFHW and PAFHWKM

VI. CONCLUSION

In this paper an enhanced parallel algorithm is proposed with K-means clustering using MapReduce programming model. The K-means clustering help to make clusters and to find the large clusters in the proposed work. In Big Data environment most of the dataset are underutilized which makes the decision making process tedious. In this work, the dataset are clustered and the dataset are utilized properly. In the proposed work, outlier deductions are considered which also improves the prediction accuracy. The proposed algorithms PAFHW and PAFHWKM are performed well for Big Data. The Holt-Winter model is performed well in forecasting the demand for the dataset with seasonal components.

REFERENCES

[1] Milan Vojnovi C, FeiXu, Jingren Zhou, "Sampling Based Range Partitioning Methods for Big Data Analytics", Microsoft Corporation,Mar 2012, pp 1-16.

[2] Lisa Wu Raymon J. Barker, Martha A Kim, Kenneth A. Ross, "Hardware Partitioning for Big Data Analytics" ISSN: 0272-1732, Volume 34, Issue 03,IEEE computer society, 2014, pp 109-119.

[3] KennSlagter, Ching-Hsien Hsu, Yeh-Ching Chung, Daqiang Zhang, " An improved partitioning mechanism for optimizing massive data analysis using MapReduce", Volume 66, Issue 01,Kluwer Academic Publishers Hingham, MA, USA Springer, 2013, pp 539-555.

[4] Jeffrey Dean and Sanjay Ghemawat, " MapReduce: Simplified Data Processing on Large clusters", Volume 51, Issue 01, ACM Digital Library, 2008, pp 107-113.

[5] Aditi Jain, ManjuKaushik, "Performance Optimization in Big Data Predictive Analytics", International Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE), ISSN: 2277 128X ,Volume 04, Issue 08, 2014, pp 126-129.

[6] Ekaterina Gonina, AnithaKannan, John Shafer, MihaiBudiu, "Parallelizing large-scale data processing applications with data skew: a case study in product offer matching", International Workshop on MapReduce and its Applications, 2011.

[7] Min Chen, Shiwen Mao, Yunhao Liu, "Big Data: A Survey", Volume 19, Issue 02, Mobile Networks and Applications, The Journal of Special Issues on Mobility of Systems, ISSN: 1383-469X (Print) 1572-8153 (Online), Springer, 2014, pp 171-209.

[8] Lei Li, FarzadNoorian, Duncan J.M. Moss, Philip H.W. Leong, "Rolling Window Time Series Prediction Using MapReduce", 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), ISBN: 978-1-4799-5879-5, 2006, pp 1-4.

[9] Sachchidanand Singh, Nirmala Singh, "Big Data Analytics", International Conference on Communication, Information and Computing Technology(ICCICT),ISBN : 978-1-4577-2078-9, 2012, pp 1-4.

[10] Dilpreet Singh and Chandan K Reddy, " A survey on platforms of Big Data Analytics", Journal of Big Data, Springer Open Access, Volume 02, Issue 08,2014, pp 1-20.

[11] RashmiRanjanDhall and B.V.A.N.S.S. Prabhakar Rao, " Shrinking the Uncertainty In Online Sales Precdiction With Time Series Analysis", Journal on Soft Computing(ICTACT), Volume 05, Issue 01, 2014, pp 869-874.

[12] B. Arputhamary, L.Arockiam, R.ThamaraiSelvi, "Analysis of Prediction Techniques in Time Series for Big Data Using R", International Conference on Engineering Technology and Science(ICETS'15), ISSN 0973-4562, Volume 10, Issue 09, 2015, pp 6712-6715.

[13] B. Arputhamary, L.Arockiam, "Parallel Prediction Model for Big Data using MapReduce Programming Model", International Journal of Applied Engineering Research, ISSN: 0973-4562, Volume 10, Issue 82, 2015, pp 1-6.

[14] B. Arputhamary, L. Arockiam, "Improved Time Series Based Algorithm for Big Data using MapReduceProgramming Model", International Journal of Applied Engineering Research, ISSN : 0973-4562, Volume 10, Issue 85, 2015, pp 1-6.

[15] Gueyoung Jung, Tridib Mukherjee, "Synchronous Parallel Processing of Big-Data Analytics Services to Optimize Performance in Federated Clouds", IEEE Fifth International

Conference on Cloud Computing, 2012, pp 811-818.

[16] Abouzeid A., Bajda-Pawlikowski K., Abadi D., Rasin A., and Silberschatz A., "HadoopDB in action: Building real world applications", In Proceedings of the 36th ACM SIGMOD International Conference on Management of Data, ISBN: 978-1-4503-0032-2, 2010, pp. 1-3.

[17] AnkitDarji and Dinesh Waghela, "Parallel Power Iteration Clustering for Big Data using MapReduce in Hadoop", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X, Volume 04, Issue 06, 2014, pp. 1357 – 1363.

[18] Chen Hsinchun, Roger H. L. Chiang and Veda C. Storey, "Business Intelligence and Analytics: From Big Data to Big Import", MIS Quarterly, ISSN: 0276-7783, Volume 36, Issue 04, 2012, pp. 1165-1188.

[19] GalitShmueli and Otto R. Koppius, "Predictive Analytics in Information Systems Research" MIS Quarterly, Volume 35, Issue 03, 2011, pp 553–572.

[20] Kanagalakshmi R, "Big Data: Performance Analysis of Vendor and Value Creation through Big Data Analytics", International Journal of Engineering Sciences and Research Technology (IJESRT), ISSN: 2277-9655, Volume 03, Issue 12, 2014, pp 429-434.