International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 02, February-2016

Parallel Implementation of Holt-Winters Algorithm for Big Data using MapReduce Programming Model

B. Arputhamary
Research Scholar,
Mother Teresa Women's University,
Kodaikanal.

Dr. L. Arockiam
Associate Professor,
St.Joseph's College,
Trichy

***Abstract* - Recent years have witnessed an enormous development in the area of cloud computing and big data, which brings up challenges in decision making process. As the size of the dataset becomes extremely big, the process of extracting useful information by analyzing these data has also become tedious. To overcome this problem of extracting information, parallel programming models can be used. Parallel Programming model achieves this by partitioning these huge data. MapReduce is the one of the parallel programming model which can be used with Hadoop Distributed File Sytems(HDFS), to partition the data in a more efficient and effective way. Once the data is partitioned, Holt-winters of time series algorithm is used with MapReduce programming model in order to improve the utilization of large volume of data and to reduce the time complexity of handling huge volume of data. In this paper, distributed implementation of Holt-winters algorithm is proposed with MapReduce computing model.**

Keywords: Big Data, Holt-winter, Hadoop, MapReduce, Prediction, Partitioning, Parallel Processing.

1. INTRODUCTION

The term "Big Data" is used for large data sets whose size is beyond the ability of commonly used software tools to capture, manage and process the data within a tolerable elapsed time. Big Data sizes are constantly increasing from a few dozen terabytes to many petabytes of data in a single data set. In 2010, Apache Hadoop defined big data as "Datasets which could not be captured, managed and processed by general computers within an acceptable scope. 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.

Today data are generated in an unprecedented 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. Due to its nature, big data has generated a number of challenges in the decision making process. Today organization's are struggling in capturing, storing and analyzing these high volume of data to increase the accuracy of decision making. 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. Using this environment, timely prediction with an increased accuracy can be achieved, which is the need of the hour. In this paper, issues related to big data are analyzed and the efficiency of the underutilized data is increased by utilizing it for prediction with parallel processing.

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]. Performance of data parallel computing such as MapReduce, 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].MapReduce is a programming model that runs on a large cluster of commodity machines and is highly scalable: a typical MapReduce computation processes many terabytes of data on thousands of machines [4]. Jeffrey Dean 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]. 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 [7]. Parallelizing time series analysis algorithms, where the training process requires a global ordering [8]. Analysis of prediction techniques in time series analysis and the adaptability in Big Data Environment [15].

III BACKGROUND

A. *Big data and predictive analytics*

Big Data is a new and emerging situation, where organizations are struggling in taking right decision at right time. Predictive models help people to take more right decisions, more quickly with less expense. They provide great support for human decisions, making them more efficient and effective. Sometimes, they can be used to automate the entire decision making process. Today data are generated in unprecedented manner and the nature of data has also undergone enormous changes. Data are generated with high volume, wide variety and high velocity. Predictive models take the historical information which already people have, and predict what will happen in future. The areas in which predictive models are generating significant value for organizations include online marketing, weather reporting, customer retention pricing optimization and fraud prevention. This paper aims at outlining the importance of prediction models in Big Data environment and the most important Holt Winter's prediction model.

B. *MapReduce and Hadoop*

Apache Hadoop is an opensource framework for distributed batch processing of big data[10]. MapReduce Programming model is used for parallel and distributed processing of large datasets on clusters [5]. There are two basic procedures in MapReduce: Map and Reduce. Typically, the input and output are both in the form of key/value pairs. In Fig.1, after the input data is partitioned into splits with appropriate size, Map procedure takes a series of key/value pairs, and generates processed key/value pairs, which are passed to a particular reducer by certain partition function. After data sorting and shuffling, the reduce procedure iterates through the values that are associated with specific key and produces zero or more output. The HadoopMapReduce provides a data processing model and an execution environment for MapReduce jobs for large scale data processing.

C. *Predictive Modelling and Techniques*

Generally three techniques are deployed for predictive modelling. These are traditional techniques, data adaptive techniques and model dependent techniques. Fig.1 depicts these models diagrammatically. Traditional approaches include linear regression and logistic regression. Data adaptive approaches find the most significant parameters which affect the prediction predominantly. Model dependent approaches use analytical methods to generate data; predictor functions etc. These include mathematical modelling, Linear Programming and Operations Research. Time series analysis is a data adaptive approach. The time series is a collection of observations of well-defined data items obtained through repeated measurements over time. For example, time seriesmeasuring the value of retail sales each month of the year comprises a time series. This is possible because sales revenue is well defined, and consistently measured at equally spaced intervals.

A time series is a collection of data recorded over a period of time—weekly, monthly, quarterly, or yearly. A time series—can be used by management to make current decisions and plans based on long-term forecasting. There are four components to a time series: the trend, the cyclical variation, the seasonal variation, and the irregular variation.

Secular Trend: The smooth long-term direction of a time series.

Cyclical Variation: The rise and fall of a time series over periods longer than one year.

Seasonal Variation: Patterns of change in a time series within a year. These patterns tend to repeat themselves each year.

D. *Holt Winters Prediction Model*

Time Series forecasting assumes that a time series is a combination of a pattern and some random error. The goal is to separate the pattern from the error by understanding the pattern's trend, its long term increase or decrease (ie.) level and its seasonality, the change caused by seasonal factors such as fluctuations in use and demand. Several methods of time series forecasting are available such as Moving Average Method, Linear Regression with Time, Exponential Smoothing. This paper concentrates on the Holt Winter's Exponential Smoothing techniques as time series that exhibit seasonality. The Holt Winter model uses a modified form of exponential smoothing. It applies three exponential formulae to the series. Firstly, the level/mean is smoothed to give a local average value for the series. Next, the trend and finally seasonal sub-series is smoothed separately to give a seasonal estimate for each of the seasons.

The exponential smoothing formulae applied to a series with a trend and constant seasonal component using Holt Winter's additive technique are[13],

$$a_t = \alpha\left((Y_t - s_{t-p}) + (1-\alpha)(a_{t-1} + b_{t-1})\right)$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$s_t = \gamma(Y_t - a_t) + (1-\gamma)s_{t-p}$$

where α, β and γ are the smoothing parameters.

a_t is the smoothed level at time t.

b_t is the change in the trend at time t.

s_t is the seasonal smooth at time t.

p is the number of seasons per year.

The initial values required for Holt Winter's algorithm are,

$$a_p = \frac{1}{p}(y_1 + y_2 + ... + y_p)$$

$$b_p = \frac{1}{p}\left[\frac{y_{p+1}-y_1}{p} + \frac{y_{p+2}-y_2}{p} + ... + \frac{y_{p+p}-y_p}{p}\right]$$

$$s_1 = Y_1 - a_p \; s_2 = Y_2 - a_p ...., \; s_p = Y_p - a_p$$

The Holt-Winters forecasts are then calculated using the latest estimates from the appropriate exponential smoothes that have been applied to the series.

$$y_{T+\tau} = a_T + b_T + s_T$$

Where a_T is the smoothed estimate of the level at time *T*, b_T is the smoothed estimate of the change in the trend value at time *T* and s_T is the smoothed estimate of the appropriate seasonal component at *T*.

The exponential smoothing formulae applied to a series using Holt-Winters Multiplicative models are:

$$a_t = \alpha \frac{Y_t}{s_{t-p}} + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$s_t = \gamma \frac{Y_t}{a_t} + (1-\gamma)s_{t-p}$$

The initial values for multiplicative model are:

$$s_1 = \frac{Y_1}{a_p} \;,\; s_2 = \frac{Y_2}{a_p} \;,\ldots\ldots\ldots\ldots\ldots\ldots., \; s_p = \frac{Y_p}{a_p}$$

IV PARALLEL PREDICTION ALGORITHM USING MRP (PPAMRP) MODEL

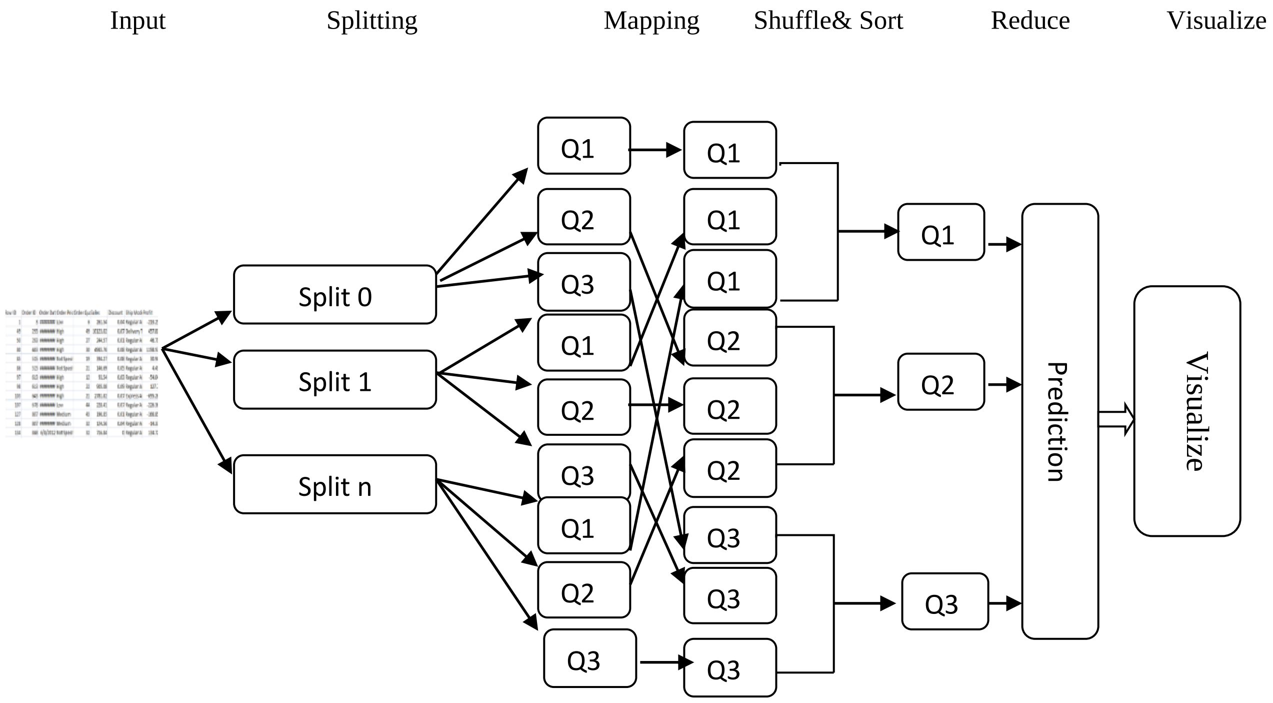


Figure 1.MapReduce Implementation of Holt Winters Algorithm

In the proposed model, large volume of online sales data are collected and split into n partitions and are stored in Hadoop Distributed File System[10]. In this work, parallel design of time series analysis and implementation of Holt Winters modeling is used with MapReduce Programming Model. The algorithms for fitting time series models are intrinsically sequential since any calculation for a specific time t depends on the result from the previous time step t - 1. Our solution parallelizes this computation by splitting the data into n chunks. The input data from each partitions are assigned to mapper and the mapper, split the input data into M splits using (key,value) pair. Apache hadoop, uses hash partitioning as default to achieve data partitioning. In this proposed work, range partitioning is used in order to split the data on quarterly basis. In fig.1, Q1, Q2 and Q3 are quarter1 (Jan to Mar), quarter2(Apr to Sep) and quarter3(Oct to Dec) respectively. Data are partitioned based on range keys. Q1, Q2 and Q3 from each split is shuffled and sorted. The reduced data set of Q1, Q2 and Q3 of all splits are assigned to the reducer with HoltWinter prediction model. The reducer predicts the sales on next coming years and forecast it in a quarterly fashion with the help of visualization process[14]-[16].

The following section describes the algorithmic steps of proposed work

Step1: Input data are partitioned into n splits. The actual form of the split may be specific to the location and form of the data. MapReduce allows the use of custom readers to split a collection of inputs into shards, based on specific format of files.

Step 2: Mapper reads the contents of the input split and produce the key/value pairs

//Key/value pairs are buffered

Step 3: Shuffling and sorting is performed at the reducer so all the keys arriving the same reducer is sorted.

Step 4: Result of the mapper is assigned to the reducer with same key and consolidated there.

Algorithm: Parallel Implementation of Holt Winters Algorithm using MapReduce

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

Output: *Forecasting values for next N years.*

```
if $T$ is NULL then
        return failure
 end if
 $T$ is splitted into $M_x$ data chunks
        $T \leftarrow \{M_1, M_2, ......, M_x\}$
  Parallel for each $M_x \in T$ do
 // Mapper
Map(String key, String value)
        //key: Order_date
        // records with corresponding key
    Parallel for each record $t_r$ in value do
        if(Month(Order_date)≥ 1 && ≤ 4)
            add the records into $Q_j$
        else if(Month(Order_date) ≥ 5 && ≤ 8)
                add the records into $Q_{j+1}$
        else
                add the records into $Q_{j+2}$
        end if
    end for
    //Reducer
Reducer(String key, Iterator values):
            //key : Order_date
            // values: a list of counts
// do shuffling and sorting in all the $Q_j$ partitions
// aggregate the data chunks from each mapper
  parallel for each $Q_j$ do
            Compute:
            //Holt-Winter's Model
            $a_t = \alpha((Y_t - s_{t-p}) + (1-\alpha)(a_{t-1} + b_{t-1}))$
            $b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$
            $s_t = \gamma(Y_t - a_t) + (1-\gamma)s_{t-p}$

            where α, β and γ lies between 0 to 1.
                $F_t \leftarrow a_t + b_t + s_t$
                return $F_t$
end for
```

$$RMSE(Y,\hat{Y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}$$

$$MAPE(Y,\hat{Y}) = \frac{1}{N}\sum_{1}^{N}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right|$$

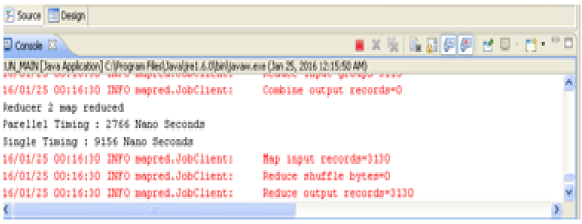
Where $Y = [Y_1, Y_2, ..., Y_N]$is the observed time series, $\hat{Y} = [\ \hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_N\ ]$ is the prediction results and N is the length of the time series.

V. EXPERIMENTAL RESULTS

The proposed algorithm has two notable phases such as Phase 1: Mapper which splits the data into chunks and Phase 2: Reducer which shuffles and sorts the data and performs predictions on the aggregated data. The first phase method makes it possible to parallelize the algorithm and the second phase greatly simplifies the implementation and improves the performance. Hadoop cluster deployed on ATOM, Dual and I3 processor machines. On each core, both Hadoop Distributed File System(HDFS) and MapReduce nodes are deployed. One node act as HDFS NameNode and MapReduce JobTracker(Master) and remaining nodes act as Datanode or MapReduce Task Tracker(Slave). The efficiency of the Holt-Winter's time series prediction algorithm is theoretically and empirically proved. In this work, the efficiency and the performance of parallel version of Holt-Winter's is tested in Big Data Environment. The eBay online sales dataset (eBay sales data from 2005 to 2015 which contains the monthly sales for various years) is taken for illustrating the proposed algorithm

A. *Time Complexity*

The performance of the proposed algorithm is compared on single and multi-node environments. Initially the comparison is made for 10000 records on dual core, I3and ATOM processors. The results are given in the following table 1. The performance of the proposed algorithm is tested with respect to the execution time in various processors.

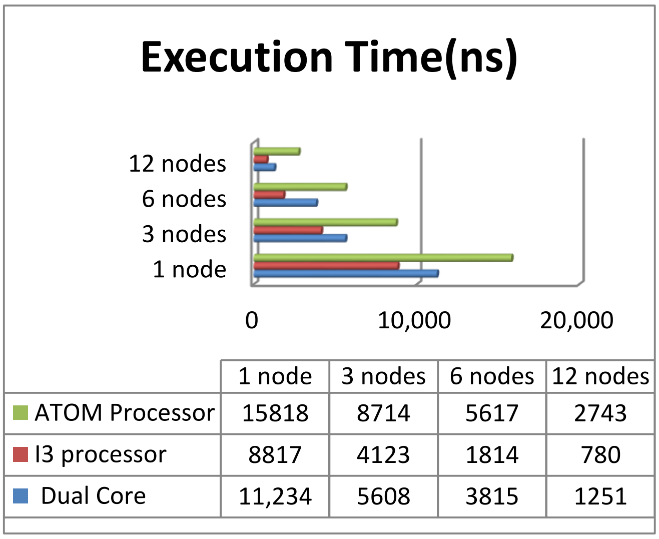


```
UN_MAIN [Java Application] C:\Program Files\Java\jre1.6.0\bin\javaw.exe (Jan 25, 2016 12:15:50 AM)
16/01/25 00:16:10 INFO mapred.JobClient:     Combine output records=0
Reducer 2 map reduced
Parellel Timing : 2766 Nano Seconds
Single Timing : 9156 Nano Seconds
16/01/25 00:16:10 INFO mapred.JobClient:     Map input records=3130
16/01/25 00:16:10 INFO mapred.JobClient:     Reduce shuffle bytes=0
16/01/25 00:16:10 INFO mapred.JobClient:     Reduce output records=3130
```

Figure 2: Execution time in nanoseconds(ns)

Table:1 Performance of the proposed algorithm on single and multi-node environment with different processors

cluster size	Dual Core	I3 processor	ATOM Processor
1 node	11,234	8817	15818
3 nodes	5608	4123	8714
6 nodes	3815	1814	5617
12 nodes	1251	780	2743



	1 node	3 nodes	6 nodes	12 nodes
🟩 ATOM Processor	15818	8714	5617	2743
🟥 I3 processor	8817	4123	1814	780
🟦 Dual Core	11,234	5608	3815	1251

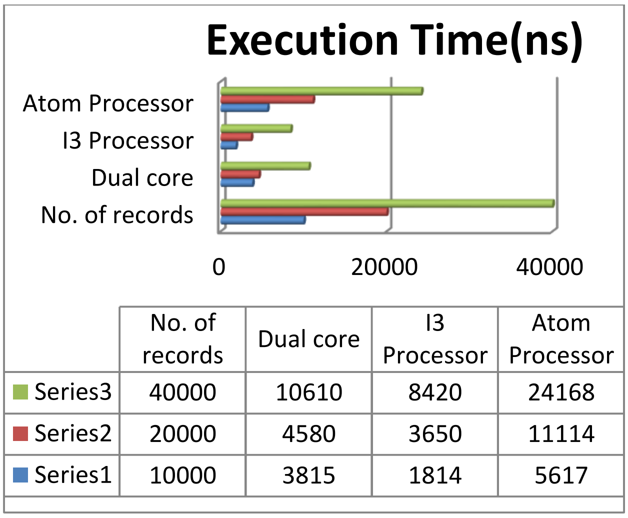
Figure 3: Performance analysis with respect to time

B. *Scalability*

The scalability of the proposed algorithm is analysed with respect to the size of the datasets. The results show that the performance of the proposed algorithm shows significance improvement in I3 processor.

Table : 2 Scalability testing with records of different sizes

No. of records	Dual core	I3 Processor	Atom Processor
10000	3815	1814	5617
20000	4580	3650	11114
40000	10610	8420	24168



	No. of records	Dual core	I3 Processor	Atom Processor
🟩 Series3	40000	10610	8420	24168
🟥 Series2	20000	4580	3650	11114
🟦 Series1	10000	3815	1814	5617

Figure 4: Performance analysis with scalability

The performance of the proposed algorithm is analysed with respect to time complexity and scalability and the results show that the proposed algorithm satisfied the expectation.

The Holt Winter's algorithm of time series prediction is implemented with parallel environment using MapReduce programming model with large scale data. The computational efficiency of the proposed algorithm is analysed in ATOM, Dual and I3 processors environment. Hadoop cluster with MapReduce programming model is used for the implementation of the proposed algorithm. The proposed algorithm performs better in parallel environment with respect to time and scalability. Parallelizing the time series data are difficult and in this approach parallelism of time series data are achieved and the results show significant improvements. From this, it is concluded that Holt Winter's algorithm is parallelized and in future this can be improved with respect to forecasting accuracy.

V CONCLUSION

In this paper, parallel prediction algorithm is proposed based on HoltWinter time series algorithm which showed significant runtime improvements compared to serial implementations. It is easy to see that the above two phases can be easily generalized to other algorithms also. The first phase can be applied for any algorithm that requires a custom or range partitioning of the data. The second phase can be used for improving the performance of any parallel algorithm. In future, another algorithm can be proposed with improvements in HoltWinter's algorithm.

VI REFERENCES

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

[2] Lisa Wu Raymon J. Barker, Martha A Kim, Kenneth A. Ross, "Hardware Partitioning for Big Data Analytics",2014 IEEE.

[3] KennSlagter, Ching-Hsien Hsu, Yeh-Ching Chung, Daqiang Zhang, " An improved partitioning mechanism for optimizing massive data analysis using MapReduce", Springer, 2013.

[4] Jeffrey Dean and Sanjay Ghemawat, " MapReduce: Simplified Data Processing on Large clusters", Google, Inc, 2004.

[5] Aditi Jain, ManjuKaushik, "Performance Optimization in Big Data Predictive Analytics", IJARCSSE, 2014.

[6] Ekaterina Gonina, AnithaKannan, John Shafer, MihaiBudiu, "Parallelizing large-scale data processing applications with data skew: a case study in product offer matching", Microsoft Research.

[7] Min Chen, Shiwen Mao, Yunhao Liu, "Big Data: A Survey", Springer, 2014.

[8] Lei Li, FarzadNoorian, Duncan J.M. Moss, Philip H.W. Leong, "Rolling Window Time Series Prediction Using MapReduce", 2006. Apache Hadoop, http://hadoop.apache.org/,Retrieved 2013.

[9] Sachchidanand Singh, Nirmala Singh, "Big Data Analytics", International Conference on Communication, Information and Computing Technology(ICCICT), Oct 19-20, 2012.

[10] Dilpreet Singh and Chandan K Reddy, " A survey on platforms of Big Data Analytics", Journal of Big Data, Springer Open Access, 2014.

[11] Dilpreet Singh and Chandan K Reddy, " A survey on platforms of Big Data Analytics", Journal of Big Data, Springer Open Access, 2014.

[12] Rashmi Ranjan Dhall and B.V.A.N.S.S. Prabhakar Rao, " Shrinking the Uncertainty In Online Sales Precdiction With Time Series Analysis", ICTACT journal on soft computing: special issue on distributed intelligent systems and applications, october 2014, volume: 05, issue: 01

[13] B. Arputhamary, L.Arockiam, R.Thamarai Selvi, "Analysis of Prediction Techniques in Time Series for Big Data Using R", International Conference on Engineering Technology and Science(ICETS'15), 2015.

[14] B. Arputhamary, L.Arockiam, "Parallel Prediction Model for Big Data using MapReduce Programming Model", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.82(2015), 2015.

[15] B. Arputhamary, L. Arockiam, "Improved Time Series Based Algorithm for Big Data using MapReduce Programming Model", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.85(2015), 2015.

B. Arputhamary working as Assistant Professor in the Department of Computer Applications, Bishop Heber College, Tiruchirappalli, TamilNadu, India. She has 10 years of experience in teaching and 4 years of experience in research. Her areas of interests are: Cloud Computing, Big Data Analytics, Data Mining and Computer Networks. At present she is pursuing Ph.D., in Computer Science in Mother Teresa Women's University, Kodaikanal, Tamil Nadu.

Dr. L. Arockiam is working as Associate Professor in the Department of Computer Science, St. Joseph's College, Tiruchirappalli, TamilNadu, India. He has 26 years of experience in teaching and 19 years of experience in research. He has published more than 213 research articles in the International & National Conferences and Journals. He has also presented 3 research articles in the Software Measurement European Forum in Rome, Indonesia and Malaysia respectively. He is also the Member of IEEE, Madras Section and lifetime senior member of ISRD, London. He has chaired many technical sessions and delivered invited talks in National and International Conferences. He has authored 4 books. His research interests are: Cloud Computing, Big Data, Cognitive Aspects in Programming, Data Mining and Mobile Networks. He has been awarded "Best Research Publications in Science" for 2009, 2010 & 2011 and ASDF Global "Best Academic Researcher" Award from ASDF, Pondicherry for the academic year 2012-13 and also the "Best Teacher in College" award for the year 2013 & 2014.