International Journal of Technology (2018) 1: 192-200
ISSN 2086-9614

AN APPROXIMATION METHOD OF REGRESSION ANALYSIS IN CONCURRENT BIG DATA STREAM

Chanintorn Jittawiriyanukoon[1*], Vilasinee Srisarkun[2]

[1]*Graduate School of eLearning, Assumption University, 88 Moo 8, Bang Na Trad Km 26, Bang Sao Thong District, Samut Prakan 10540, Thailand*
[2]*Martin de Tours School of Management and Economics, Assumption University, 88 Moo 8, Bang Na Trad Km 26, Bang Sao Thong District, Samut Prakan 10540, Thailand*

(Received: February 2017 / Revised: September 2017 / Accepted: January 2018)

ABSTRACT

Time series big data dynamically changes the size, and, unfortunately, it may be difficult to curate the enormous amount of data due to the processing capacity and storage size. This big data allows researcher to iterate on the model millions of times over. To execute a regression on several billion rows of data on a distributed network, the resource capacity regarding large volumes of data and its distributed environment must be considered. Algorithms must be real-time based data awareness. Moreover, analyzing big data sources requires the data to be pre-processed rather than immediately collected and analyzed. This pre-processing approach for the big data sources helps minimize the amount of collected data by extracting insights. It analyzes big data quicker and is cost-effective for storage space. Hence, in this research, an approximation method for analyzing regression problems in a big data stream with parallelism is proposed. The partitioning method for huge data stream helps reduce the computing time and required space, and the speed-up can improve the processing time. The performance evaluation of concurrent regression model is first executed by massive online analysis (MOA) simulation. Then, to validate the approximation method, the results performed by our proposed method are compared to those results collected from the simulation. The comparisons show evenly between the two methods.

Keywords: Approximation method; Big data curation; MOA; Parallel processing; Regression analysis

1. INTRODUCTION

Machine learning in regression analysis refers to an infinite set of tackles to manipulate data. These tackles can be categorized as manual or automation. Generally, manual machine learning regards the development of a statistical model for forecasting outputs based on input parameters, and the problems of this occurrence reflect medical related technology, business, finance, and market analysis. With an automated machine learning mode, there are more input parameters but no inductive outputs, while analyzers can comprehend nature and behavior from such descriptive data. For example, Figure 1 displays wage dataset versus age for a group of males in the United States. The figure shows that wage increases against age until 60 years old and then decreases thereafter, and the bold blue line in the figure provides the mean wage value versus age. Given an employee's age, the wage can be predicted using this curve trend.

*Corresponding author's email: chanintornjtt@au.edu, Tel: +66-2723-2948, Fax: +66-2723-2959
Permalink/DOI: https://doi.org/10.14716/ijtech.v9i1.1509

However, age cannot provide an accurate approximation of an individual's wage. To provide this accurate estimation, an in-depth regression analysis needs to be performed (Fox & Weisberg, 2011). An anomalous detection for the given set of big data can be also applied, as discussed by Hodge (2014).

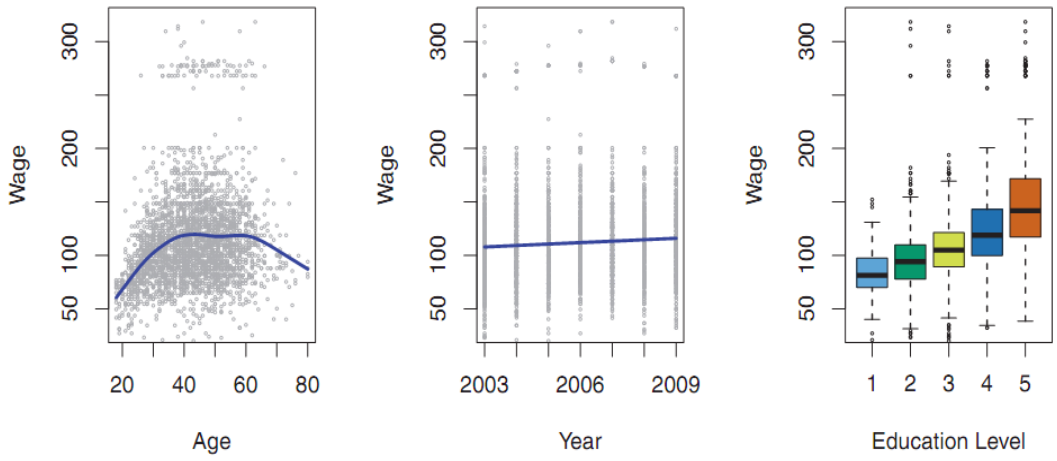

Figure 1 Income data (wage in hundred USD unit), which contains information from a survey for males who came from the Atlantic region in the United States (James et al., 2013)

Now, massive data streams (Hu & Kaabouch, 2014) are created from traffic generating environments such as social networks, Internet of Things, and smart communication traffic. Free format (unstructured) data travels continually at a fast speed and a varying rate. In contrast, big data generates specific formats (structured) that includes volume, velocity, and variety. Opting a processing location and curating enormous records is a challenge, as it would produce a predictive answer for velocity, volume, and data range. Volume represents the quantity of input data, velocity represents the rate of data arrival, and variety represents the imitation of information categories (unstructured or structured information). The shaped information certainly reflects the database and the documents from users, while unorganized statistics include email, tweets, images, movies, and video files. According to the variety element of the unorganized data, troubles regarding curation, repository management, and prediction of data increase. Big data can be illustrated by finding a concurrency (parallelism) to suit available processing units to fasten and meet real-time processing requirements, especially while avoiding the bottleneck situation.

Hu and Kaabouch have defined big data as a collection of data that cannot be easily controlled by traditional database processing methods due to the gigantic size of the data (Hu & Kaabouch, 2014). Such data includes the mixture between multimedia and traditional data (e.g., text, documents). Thus, the format of this data unstructured. Sunghae et al. (2015) divided big data technologies into four categories: curation, infrastructure, analysis, and predictive support processing. In the infrastructure category, all data needs to be collected (storing and retrieving). This is far beyond the scope of traditional technology of database and data structure. In the curation, one of most important processes in big data is an analytical approach that extracts high value data (insight) from big data sources. One of the problems is applying an appropriate algorithm to simultaneously analyze big data sources. This step consumes time and processing power to analyze all sources at once. Specifically, the execution experiences limited computation. To ease the computing burden for big data curation, big data can be divided into subsets of smaller data to smooth the computational load (Sunghae et al., 2015). However, they did not refer to the concurrent regression analytical model, and both partitioning and merging

times were neglected. In the proposed research in this study, the approximation method was also applied to perform concurrent regression analysis of big data sources. To validate the approximation method, the simulation method is employed and then the results are compared to the approximated calculations. Another approximation approach for analyzing big data is discussed by Heinis (2014).

This research focuses on evaluating the concurrent regression analysis of big data using MOA simulation (Bifet et al., 2010). Both dividing and integrating time were considered for concurrency execution. First, the speed-up has been emphasized to improve processing capacity. Second, an approximation method has been performed, and these calculation outputs are compared to results from the simulation to validate the method. Lastly, the approximation method must be simpler and faster than simulation, and the percentage of errors is presented after comparison. Approximation is an option to help shorten the execution if the accuracy of the result is not significant.

2. CONCURRENT REGRESSION ANALYSIS

2.1. Regression Model of Data Stream

Regression model is the common method used to curate data. It is a statistical model that is practical, well explained, and hypothetically comprehensive. Big data that is extracted into the regression model can be used to help solve complex problems. The insights from regression model are understandable and best-fit models. Regression model was chosen as a first choice for problem-solving in any practical cases because of its simplicity, providence, and adaptability. Then, multiple regression is an addition of simple regression model (linear regression). The multiple regression model is used to calculate the value of two or more dependent variables, and they sometimes are called the outcome, output, or target variables. In addition, multiple regression is an advanced analytical tool and is powerful, as a prediction model for a variety of outcomes can be developed. For example, multiple regression allows the model fit (variance of the model) to be defined, as well as the relative correlation of each prediction to the total variance to be defined. Once how these variables relate to a dependent variable is identified, then information about independent variables can be considered and accurate estimations can be concluded. In general, the multiple regression model of Y on X_1, X_2, …, X_k can be specified by Equation 1.

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \ldots\ldots\ldots\ldots\ldots\ldots + b_k X_k + \sigma \quad (1)$$

where X_1, X_2, …, X_k represent independent variables and Y represents the dependent variable. In addition, b_0 represents the cut-off and b_1, b_2, b_3,…, b_k represent regression coefficients or the slopes in the simple regression model. σ represents an error of the estimation.

In the concurrent regression model, the partitioning technique is applied by cautiously regarding the inherit concurrency of given big dataset. The purpose of all partitioning is to obtain a representative smaller piece of sub tasks from the big data. In this research, simple and identical sub tasks for partitioning big data are assumed, and the partitioning time that depends on concurrency is measured and considered. Figure 2 shows the partitioning method for concurrent big data.

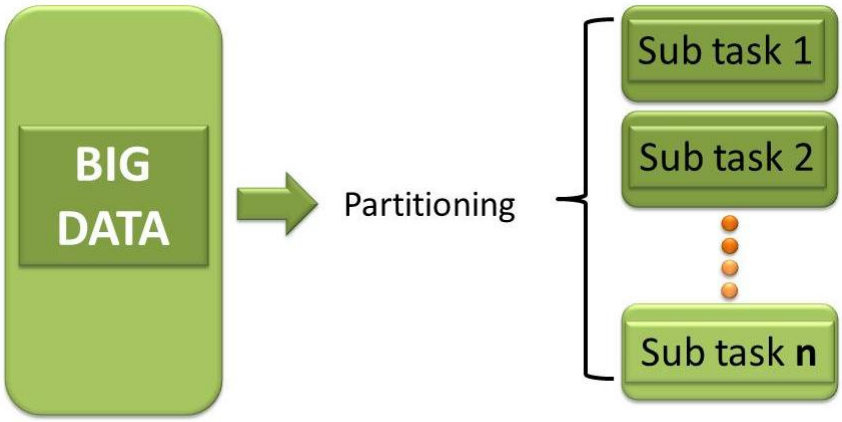

Figure 2 Partitioning of Big Data

To merge the results from sub task *1* to sub task *N*, as shown in Figure 3, the average value for residual time (RT) is calculated as follows.

$$RT = \max (M_1, M_2, \ldots, M_n) + p + \alpha \quad (2)$$

where p represents an average partitioning time, M_n represents a processing time of each sub task n, and α represents the reassembling time after the completion of all the processing of sub tasks.

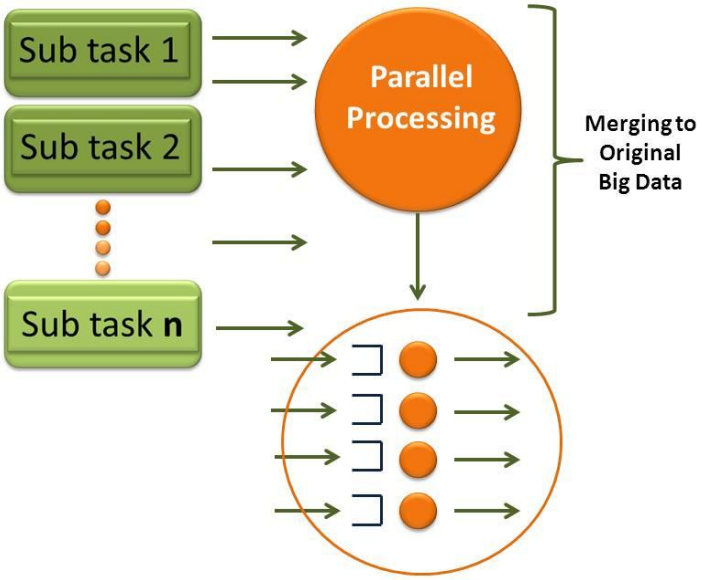

Figure 3 Integration of Parallel Processing Results

2.2. Accuracy of Fit (AoF)

To determine the performance measurement of machine learning on big data, how well predicted values match the monitored data must be evaluated. The root mean squared error (RMSE) denotes the standard deviation of the alterations between forecast values and monitored values, and this alteration is called squared error, as the computation is executed over the sample dataset used for prediction, which is also called estimation error. Namely, whether the predictions for an observed data is close to the true value needs to be quantified. In the regression analysis, the common metric is the RMSE, which can be simply calculated by the

following:

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

The RMSE of predicted values $\hat{y}_i$ for the ith observation of a regression's variable y_i is calculated for n altered forecast as the square root of the mean of the squares of the error. The RMSE is a small value if the forecast values are close to the true values, while it is a large value if the predicted and true values significantly differ. The RMSE in Equation 1 was calculated using the training data that was employed to fit the experimental model, and to be more specific, this is denoted as the training RMSE. In practice, how well the experiment performs on the training data is trivial. The accuracy of fit (AoF) for obtainable predictions is important metric as the method is applied to unknown test datasets (James et al., 2013).

3. APPROXIMATION METHOD

In general, no one likes to wait but the queue involves a life. Reduction of the waiting time in queue requires faster services for customers, which may lead to increased investment costs. If fast service is not provided, waiting time in the queue can become too long. To decide whether to invest, the cost effectiveness must be studied, which is the balance between service cost versus efficiency. Thus, specific models and methodologies are needed to analyze situations. The queuing model is the study of waiting time, scheduler, and processing speed, and the applications of the queuing model include manufacturing, transportation and supply chain systems, computer systems, and information systems. Queuing models are suitable for the design of these systems regarding control, traffic management, and processing capacities. For example, trainers can be cost-effectively managed for a fitness center. Regarding allocating an extra trainer to the center, which he or she must be supervised by an existing trainer, the performance is compared to how much the center can lower the waiting time for VIP customers (to earn their loyalty) if the extra trainer is hired (involving extra cost).

This section introduces the analytical model used in this research. First, it is assumed that the big data is composed of n independent regression datasets. Each sub dataset is called the sub task, and the network consists of several servers such as partitioning, merging, and parallel processing devices (Tsai et al., 2016). Big data enters the partition-server with a service time. In this partition-server, big data was equally split into n sub tasks. The sub tasks spawned by the big data is called sibling, and all siblings were subsequently processed by parallel servers in the network system, as depicted in Figure 4. It is assumed that siblings are all independent. When each sibling finishes the processing in the network, it must leave for the merge-server. Unless the processing in another sibling is finished, the ended sibling must wait for their completion in the buffer. As all sibling completion was attained, they were merged into the original big data. At this time, the synchronization of sibling started, and the process was constantly repeated. The duration from when big data enters a partition-server until the synchronization is met is called a cycle time (partitioning time and RT described in Equation 2). If a big data (with identifier n) is split in a partition-server into M_n sub tasks, then it is defined that the sub tasks have the concurrency M_n. In general, M_n changes cyclically. The concurrency for each cycle is denoted by a vector $C_n = \{M_{n1}, M_{n2}, M_{n3}, \ldots\}$ called the concurrency vector. An approximation method presented in (Jittawiriyanukoon, 2014) was employed to measure the performance metrics such as processing time.

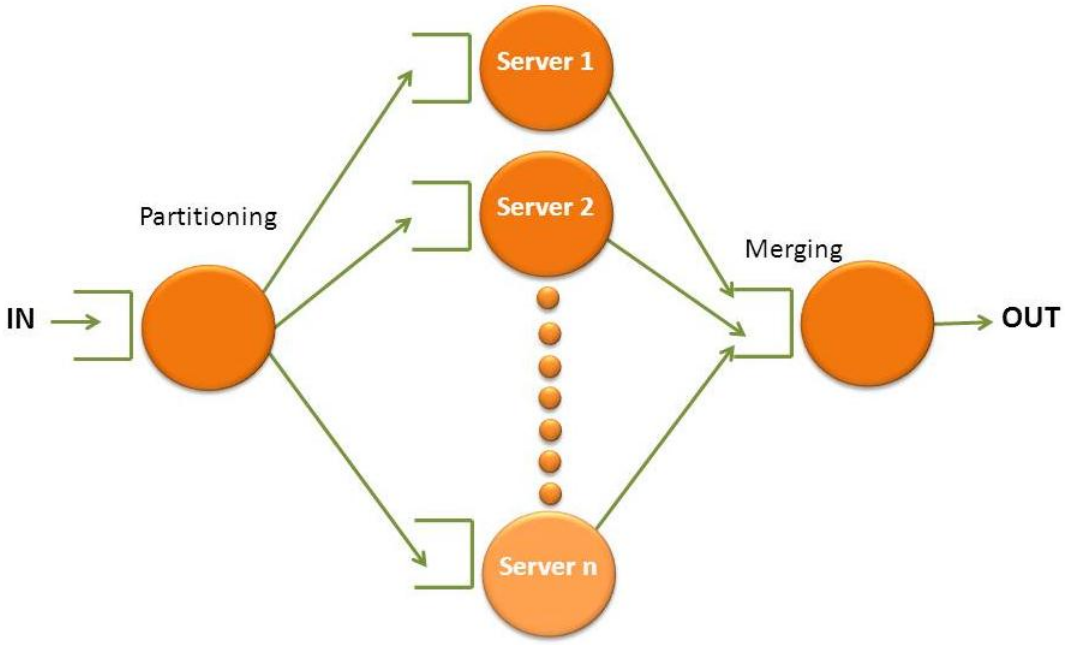

Figure 4 Parallel Processing System

The *M/M/4* network system shown in Figure 4 was implemented using MOA (Bifet et al., 2010). MOA can be also applied to other applications in the area of data mining (Srimani & Patil, 2016). Parallel simulation but for cloud computing (Khan & Ahirwar, 2011) was introduced by Malik et al. (2010). Each big dataset fluctuates with an inter-arrival time (λ) with Poisson distribution, and the service time on each server was exponentially distributed (mean = μ). Big data partition produced an average time *p*, and it is assumed that the merging is executed at an average time α. The average processing time was computed as the approximated performance measurement and compared to the results from the MOA simulation.

4. RESULTS AND ANALYSIS

The MOA simulation configuration contains these parameters, *Performance Evaluator*: Window Regression, *Instance Limit*: 100,000,000, and *Evaluation Task*: Prequential Regression. In addition, the synthetic datasets with different size (31, 54, 432, and 436 MB) are used with the MOA simulation. Table 1 lists the results from simulation such as mean absolute error (MAE), RMSE, simulation time (ST), and memory space consumption (MSC). The simulation has been run on a Samsung Windows 8 with Intel® Core ™ i5 6200U CPU, 2.3 GHz Processor, and 8 GB DDR3 RAM. The datasets have been chosen so they are distinctive in size, attributes, and instances, and the simulation execution time and required memory are independent from the size of the datasets.

Table 1 Simulation results

	Dataset			
	1	2	3	4
MAE	0.47	0.15	0.35	0.05
RMSE	0.49	0.77	0.53	0.09
ST (sec)	17.86	11.01	97.41	151.94
MSC (MB)	0.08	0.47	16.45	0.8

Datasets 1, 2, 3, and 4 were executed on a single server, and each dataset were partitioned into four sub tasks to be independently processed on four parallel processors. The latter execution include both partitioning time and merging time based on the calculations. Partitioning is based on splitter software introduced by G.D.G. Software SARL (G.D.G. Software SARL, 2016), and the residual time (RT) can be approximately computed by Equation 2. The comparisons between simulation and approximation results executed on one and four processing units are shown in Table 2, and the approximation results are close to those results from the simulation.

Table 2 Results comparison for one and four processing units

Residual Time (sec)		Dataset			
		1	2	3	4
M/M/1	SIM	18.06	10.97	99	151
	APPR	17.94	10.95	98.60	151.60
M/M/4	SIM	8.27	8.75	26.37	50.08
	APPR	8.50	8.70	34.50	49.75

The residual time after splitting into 2, 4, 6, 8, and 10 sub tasks were investigated by the processing power of 2, 4, 6, 8, and 10 units. The largest dataset 4 was chosen for the test, and the comparison between simulation and approximation results executed on different processing units is presented in Table 3. The approximation results also fit those results from the simulation.

Table 3 Results comparison for dataset #4

Dataset 4 RT (sec)	Processing units				
	2	4	6	8	10
SIM	82.16	50.08	41.2	35.04	31.1
APPR	81.5	49.75	39.16	34.87	30

The speed-up metric for processing efficiency was further analyzed for price-performance to consider cost-effectiveness. The largest dataset 4 is the choice for performance evaluation, and the comparison results are displayed in Table 4. The table shows that the simulation and approximation results are comparable.

Table 4 Speed-up results for dataset #4

Dataset 4	Speed-up				
	2 PUs	4 PUs	6 PUs	8 PUs	10 PUs
SIM	1.83	3.01	3.66	4.3	4.85
APPR	1.85	3.03	3.85	4.33	5.03

Processing efficiency was considered using speed-up metric to analyze. The ideal or theoretical speed-up was achieved by the top plotted line in the graph shown in Figure 5, and the simulation results draw an adjacent line to the approximation line.

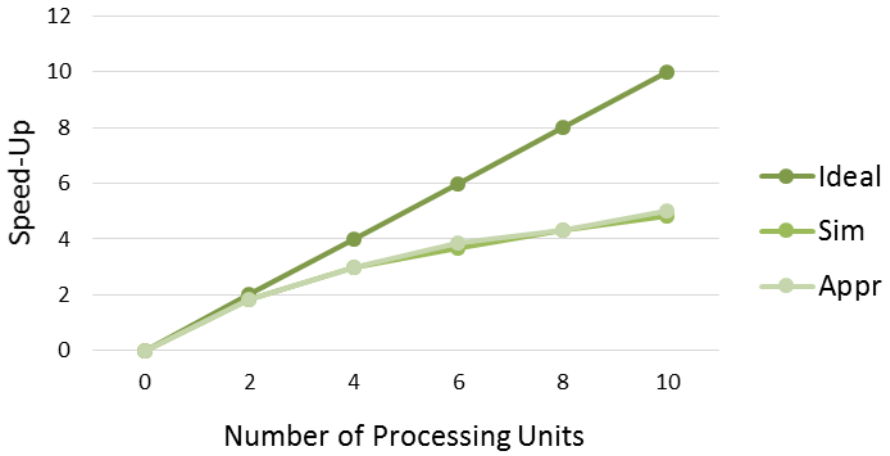

Figure 5 Comparison of speed-up data for actual simulation, approximation, and ideal situations

5. CONCLUSION

This paper proposes an approach to handle the computing cost in big data curation. The main contribution of this research is to partition huge dataset into sub tasks and then execute siblings on parallel processing systems. Both splitting and reassembling time are seriously considered in our calculation. The approximation technique based upon queuing theory has been introduced, and to validate these approximation results, simulation results were compared to the approximation results, which the approximation results were comparable to the simulation results. In addition, the cost-effective analysis was emphasized to measure the speed-up factor. The research contributes to how parallel computation can accommodate for the massive data cost. To manipulate big data analytics on business intelligence, the capability to execute a big number of concurrent queries is a critical problem for an application such as Google Big Query. Regarding to new benchmark recently released by Google's cloud, a new mechanism can simplify data preparation for both machine learning and deep learning. For instance, the cloud-serverless model allows as large as 25 concurrent queries on datasets. Hence, future research will study on 30 concurrent queries on applications such as Google's cloud to determine whether or not they can handle high concurrency. Another investigation may include cost-effectiveness analysis for the case of multiple processing units. The approximation method will then be applied to reduce the complexity in simulation execution in future research.

6. REFERENCES

Bifet, A., Kirkby, R., Holmes, G., Pfahringer, B., 2010. MOA: Massive Online Analysis. *Journal of Machine Learning Research 11*, pp. 1601–1604

Fox, J., Weisberg, S., 2011. *An R Companion to Applied Regression*. California: SAGE Publications

G.D.G. Software SARL., 2016. Software SARL. Available online at http://www.gdgsoft.com, Accessed on February 15, 2017

Heinis, T., 2014. Data Analysis: Approximation Aids Handling of Big Data. *Nature*, pp. 198–198

Hodge, V.J., 2014. *Outlier Detection in Big Data*. IGI Global, pp. 1762–1771

Hu, W., Kaabouch, N., 2014. *Big Data Management, Technologies, and Applications*. Information Science Reference, IGI Global

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. *An Introduction to Statistical Learning with Applications in R*. New York: Springer

Jittawiriyanukoon, C., 2014. Performance Evaluation of Reliable Data Scheduling for Erlang

Multimedia in Cloud Computing. *IEEE Proceedings of the Ninth International Conference on Digital Information Management (ICDIM 2014)*, pp. 39–44

Khan, A., Ahirwar, K.K., 2011. Mobile Cloud Computing as a Future of Mobile Multimedia Database. *International Journal of Computer Science and Communication*, Volume 2(1), pp. 219–231

Malik, A.W., Park, A.J., Fujimoto, R.M., 2010. An Optimistic Parallel Simulation Protocol for Cloud Computing Environments. *SCS M&S Magazine*, Volume IV, pp. 1–9

Srimani, P.K., Patil, M.M., 2016. Mining Data Streams with Concept Drift in Massive Online Analysis Frame Work. *WSEAS Transaction on Computers*, Volume 15, pp. 133–142

Sunghae, J., Seung-Joo, L., Jea-Bok, R., 2015. A Divided Regression Analysis for Big Data. *International Journal of Software Engineering and Its Application*, Volume 9(5), pp. 21–32

Tsai, C.-F., Lin, W.-C., Ke, S.-W., 2016. Big Data Mining with Parallel Computing: A Comparison of Distributed and Map Reduce Methodologies. *Journal of Systems and Software*, Volume 122, pp. 83–92