

Enhanced Load Balance to Predict Fast Data Stream using E-Tree MSI Method on Cloud

B. Balamurugan^{1*}, D. Kamalraj¹, S. Jegadeeswari¹ and M. Sugumaran²

¹Bharathiar University, Coimbatore – 641046, Tamil Nadu, India; balaanandmca@gmail.com, d. kamalrajmca@gmail.com, juga_sathya@yahoo.co.in

²Pondicherry Engineering College, Pillaichavadi – 605014, Puducherry, India; sugu@pec.edu

Abstract

Cloud computing implements virtualization processing of data service in the internet, where it delivers the conceptual, scalable platforms and applications as on data services. The important problem arises in Cloud infrastructure in storing a very large amount of data and processing on the computational load on the cloud. It is a big challenge to overcome computation complexity on cloud. An effectively predict the data streams process with load factors of ensemble model and data stream are implemented to overcome in cloud. Data stream processes on the cloud infrastructure runs with continuously varying load factors. In this work, we propose an architecture with a load balancing framework for cloud infrastructure by using the Ensemble Tree Metric Space Indexing (E-tree MSI) technique. We developed three techniques to construct our E-tree MSI technique: Fast Predictive Look-ahead Scheduling approach (FPLS) where the scheduling of Spatio-temporal data stream files takes place; Parallel Ensemble Tree Classification (PETC) which performs the process of classification operations on cloud data stream; and a Bilinear quadrilateral Mapping process which adds efficient implementation of cloud infrastructure. We have done an experimental evolution using CloudSim, from which it is achieved that the performance of load balancing factor is increased, the accuracy rate of classification is better and it reduced the execution time for mapping.

Keywords: Cloud Security, Data Stream, E-Tree, Load Balancing, Load Factor, Mapping and Metric Space

1. Introduction

In cloud infrastructure data are processed efficiently by running continuously with the varying load factors for data streaming. A cloud infrastructure presents a system to meet the fluctuations on the computational load¹. Cloud infrastructures meet the end-to-end latency objective and effectively predict the data streams of ensemble models^{2,3}. In data streaming, a different type of data stream supports the parallel processing for all the workflows. Many data stream processes are depend on work load balancing factors, and operator schedule on the secured cloud environment. Cloud computing uses the virtualized processing and storage resources in conjunction with modern technologies where it delivers different on data services⁴⁻⁶. In processing the data in cloud, the data are of secured billing on these data services is directly

tied to usage statistics⁷⁻⁹. Distributed cloud data stream processing engines often utilize the inflexible operator allocation strategies^{9,10}. The streaming strategy is analyzing the temporal relation between secured data in a cloud while in the streaming process. The research work is carried out to implement a secured and fast prediction of the data stream in cloud through load balancing by constructing an indexing structure^{4,11-13}. In cloud the security issues are general classified into trust, identity management, software isolation, data protection, availability reliability, ownership, data backup, data portability and conversion, multi platform support and intellectual property. The cryptography is the techniques that were implemented to protect data in cloud. This concept was developed to store the data in cloud in the encrypted data format which is based on block cipher, by using cryptography technique¹⁴.

*Author for correspondence

The Ensemble Tree Indexing Structure (E-Tree), as described in ¹⁵, arranges the entire base sorting for predicting the result with minimal time complexity. E-Tree straightly scans all the historic classifiers during prediction using the divide and conquer policy. E-tree assessment is not extended to the fast prediction of the results on balancing the load. Ensemble-tree indexing structure in ^{17,18} explains the prediction efficient problem on cloud data streams. Ensemble-tree indexing does not process the extend work with cloud data classification for balancing the load¹⁹.

In order to implement load balance statistics on the cloud data stream, the Ensemble Tree Metric Space Indexing (E-tree MSI) system is applied. The E-tree MSI system is assembled on the cloud infrastructure where application data streams are detached and distributed over various servers, even and also for the fast prediction of better performance^{16,17}. The E-tree with the statistical mechanism helps in handling the different another data stream in a flexible manner. The E-tree balances a high linear load organization with less processing computational complexity^{16,22,23}.

The Cooperative Provable Data Possession (CPDP) scheme in ⁴ present an effective method to select the clients and cloud storage service providers. CPDP provides particular purpose for data storage and association, whereas, large data streams affect the bilinear mapping operations. Multi-tier request deployments on Infrastructure-as-a-Service cloud as established in ^{9,20,21} increase a multiple linear regression model to predict application performance but the load balance statistics are not carried out on the cloud data stream.

Each data stream process is planned to start at an initial step and the progress is forwarded on balancing the load factors on the cloud data stream process^{16,24–26}. Metric Space Indexing in this system executes the operations on the cloud data stream to carry out the classification on each request of the cloud users. The classification of the data stream in the E-tree MSI technique reduces the overload factor and the completing time^{2,27}. The E-tree MSI method is able to dynamically pursue changes in the stable data stream modifications. Bilinear quadrilateral mapping in the E-tree MSI technique uses basics purpose to linearly predict the result beginning of the cloud data storage and managing streams.

Load balancing in cloud distributes the workload in dynamic ways and helps to utilize the resources optimally. The load balancing algorithms have process the following

factors, scalability, resource utilization, performance, response time, etc.^{10,28}. The load balancing method is applied practical across different data centers to ensure the network resources in different locations of the cloud. By using the load balancing techniques, resource utilization, availability of data, which improves performance cloud computing environment. It undertakes to decrease the cost of cloud implementation systems⁹.

This paper is of different sections, the first session about introduction and discussions on various existing work done on the data streaming and the data load balancing techniques for constructing our E-tree Metric space index (E-MSI) technique¹⁵. In the implementation and evaluation section, the discussion of the various results analyse are done by CloudSim. Finally, in the conclusion discuss some advantages and future scope of the work.

2. Proposed Framework E-Tree MSI Method for Load Balancing Factor

The new proposed framework for load balancing on cloud infrastructure and a formal definition of the Ensemble Tree Metric Space Indexing (E-tree MSI) technique. Here, the new proposed model comprised of three techniques for E-tree MSI method: the Fast Predictive Look-ahead Scheduling approach (FPLS) where the scheduling of Spatio-temporal data stream files takes place; Parallel Ensemble Tree Classification (PETC) that performs the process of classification operations on a cloud data stream; and the Bilinear Quadrilateral Mapping (BQM) process which supports efficient indexing in a cloud infrastructure, which implements the efficient construction of an effective load balancing query processing approach.

The proposed cloud computing technique is a significant measure to maintain and manage different client data streams. The E-tree MSI technique with a balanced load factor accomplish the process of client request to resources with minimum completing time and reduce the overload factor. The proposed cloud platform provides an effective indexing technique with a large distributed system. In the Figure 1, the scheduling, indexing, and mapping operation of the new framework architecture diagram of the E-tree MSI technique is shown and explained.

The scheduling method in the Ensemble Tree Metric Space Indexing technique is carried out using the Fast Predictive Look ahead Scheduling (FPLS) approach. The

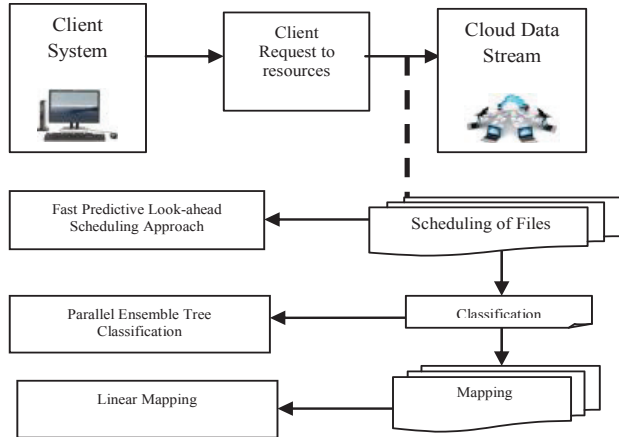


Figure 1. Architecture Diagram of the E-Tree MSI Technique

prediction is carried out faster manner by means of the Look-ahead Scheduling process. The cloud servers schedule the files obtained from multiple users, so that the prediction of results in an accurate manner, is performed with the balanced load. Subsequently, Parallel Ensemble Tree Classification (PETC) is applying to classify the processed files in a tree structure. The tree structure helps to easily perform the mapping process in the final step. The tree structure perceives the related perspectives and effectively predicts the various characteristic indexing in the Ensemble Tree Metric Space Indexing strategy. The mapping is carried out using the Bilinear Quadrilateral Mapping (BQM) process. Bilinear quadrilateral helps to map the client query which outcome in a direct form at both ends of the tree (i.e., left and right end of the tree structure) in a minimum time.

2.1 Fast Predictive Look-ahead Scheduling Approach

The Fast Predictive Look-ahead Scheduling methodology is successfully get to the cloud asset for different customer necessities. The query request is managed by initially scheduling the files. The files are properly scheduled with the balanced load factor in the E-tree MSI technique using the Predictive Look-ahead Scheduling approach. Let us assume a set of request being made by several clients in cloud at a time t , where the Look-ahead Scheduling of the Spatio-temporal data stream (i.e., files) is performed as,

$$RS_{look\ ahead} = \sum_i^n R_{il}(t) \quad (1)$$

From (1), the request scheduling ' R ' is formularized as the product of each client user request. The Request Scheduling using Look-ahead move toward in handled is an organized manner for ' i ' user at time ' t '. Initial, the efficient rule is take up for the time period ' t ' for scheduling the Spatio-temporal data stream on the cloud infrastructure. The scheduling divides the time interval as,

$$TI = t_0^i, t_1^{i+1}, t_2^{i+2}, \dots, t_n^{i+n} \quad (2)$$

The time interval ' TI ' denotes the time schedule for ' i ' users to perform the adaptive decision made on the cloud data stream. The fast prediction interval strategy set a variation of action to maximize the cumulative profit rate with a balanced load factor. In the Figure 2. the Fast Predictive Look-ahead Scheduling process is illustrated and explained.

The FPLS process in the E-Tree MSI strategy, where the present state regulations are aggregate with the succeeding principle to shape a compelling system with an effective technique with a balanced load factor. The diverse customers put their request for resources through a cloud system correspondence in the Figure 2. Utilizing FPLS, diverse customers Client1, Client2, ... , Client n put their request for resources such as CPU, memory et cetera. The cloud server by method for the FPLS process in the E-Tree MSI strategy utilizes the cradle status table to adjust the heap notwithstanding the Look-ahead Data Stream Address Table in the cloud system for proficient planning. With this, the Look-ahead based quick booking diminish the consistent completion time rate. The point of the FPLS in the E-Tree MSI techniques be of help to expand the utility of the cloud resources with a balanced load. Subsequently, various customer request is booked in the cloud in an effective way took after by which the procedure of classification is performed in the upcoming segment.

2.2 Parallel Ensemble Tree Classification (PETC)

Cloud process classification is essential to classify the process of the Spatio-temporal data stream. In the E-Tree MSI technique, Parallel Ensemble Tree Classification on the scheduled files recursively classifies to progress the efficiency rate on the cloud infrastructure. The classification of the Spatio-temporal data stream on the tree structure evidently defines the file characteristics. The non leaf in a tree defines the margins in the E-Tree MSI

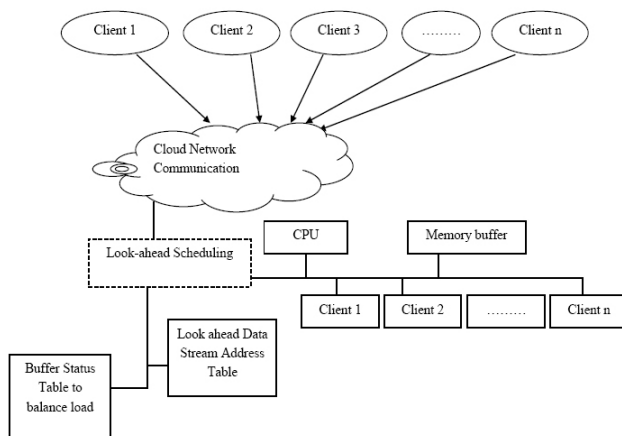
PETC($C(F_i)$)**Input:** Data Stream File $\{F_i = F_1, F_2, \dots, F_n\}$, IT , i **1: begin****2: for each F_i**

3: split the process

4: $F_i = \text{argmax}(F(\text{Process}_i)[\text{left } C(\text{Process}_i | t) + \text{Right } C(\text{Process}_i | t)])$ 5: **If** split criteria (process) **then**6: tree-left prediction = $(C(F_i))$ 7: **else**8: tree-right prediction = $(C(F_i))$ 9: **end if**

10: place the split process on the left and right sides of the tree

11: non-leaf for split branch specifies the margins

12: IT = splitted data stream13: **end for**14: **return** $(C(F_i))$ **15: end****Output:** Classification from cloud data stream files**Algorithm 1:** Parallel Ensemble Tree Classification**Figure 2.** Design flow of FPLS Process.

technique cloud data stream. The region boundaries facilitate to easily distinguish multiple attributes to decrease the overload factor.

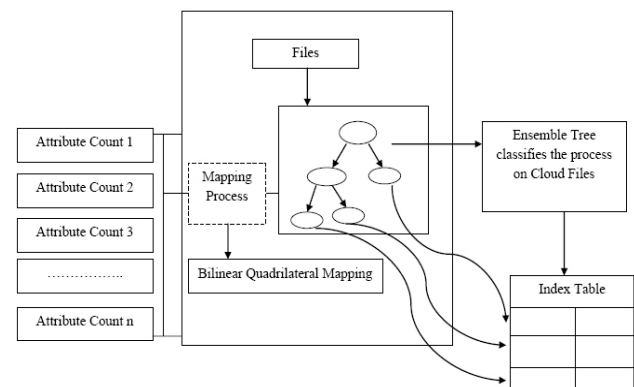
As per proposed method, in Parallel Ensemble Tree Classification the leaf node on left and right sides of tree involves fast prediction in parallel with different metric space as a simple steady function. Fast prediction ' F ' with a simple steady function is formularized as,

$$F = \sum_i^n \text{argmax}(F(\text{Process}_i)[\text{Left } C(\text{Process}_i | t) + \text{Right } C(\text{Process}_i | t)]) \quad (3)$$

File ' F ' on the cloud infrastructure categorize the process for ' i ' users on the left and right sides of the tree structure using the E-Tree MSI Technique. The Parallel Ensemble Learning Rule is take on for the classification of the Spatio-temporal data stream in the E-Tree MSI Technique.

The Spatio-temporal data stream classification process in our proposed work by means of the parallel ensemble tree procedure, which is clearly illustrated in the Figure 3. The Parallel Ensemble Learning Rule in the E-Tree MSI method helps to split the data stream on a cloud infrastructure in an equivalent fashion to reduce the overload factor. In the Figure 3, depicts the design of mapping process in the Ensemble Tree Metric Space Indexing technique using Bilinear Quadrilateral Mapping.

The Ensemble Tree classification (E-tree) uses the learning approach to obtain a better predictive performance of the Spatio-temporal data stream with high flexible minimal overload factor structure. In order to reduce the computational complexity the Spatio-temporal data stream is accumulated in the indexing table. The reduced overload factor of Cloud data stream classification is described as a step-by-step process. The Parallel Ensemble Tree Classification steps is described in the below algorithm. The algorithm classifies the scheduled data stream for easy processing with a minimal overload factor. The algorithm performs the classification process supported on the data stream file given as input. For every file, " F_i " on the cloud arrangement the procedure in a

**Figure 3.** Parallel ensemble classifications with mapping procedure.

productive way for “i” clients on the left and right sides of the tree. The non-leaf node on the tree concentrates on minimizing the overload factor. With this E- tree calculation parts the records processed for easy mapping from the cloud data storage frameworks.

2.3 Bilinear Quadrilateral Mapping

Once the spatio-temporal data streams are characterized in an effective way by utilizing Parallel Ensemble Tree Classification, all around composed mapping is performed to give the request for resource to various clients in the cloud. The last stage in the proposed work is the configuration of an all around composed mapping on the arranged cloud data stream documents ‘F’. To perform the very much sorted mapping system on the cloud base, the E-tree MSI procedure utilizes the bilinear quadrilateral mapping. The Bilinear Quadrilateral Mapping procedure is done in the E-Tree MSI strategy to performing the inquiry process on both sides of the outfit of the tree (i.e., left and right). The mapping capacity on the listed table procedure data contains the function as,

$$BQM = \left[1 + (\text{process}) + \frac{t}{2} \right] - \text{unmapped process} \quad (4)$$

The procedure of mapping with the previous time interval enhances the framework correctly. The bilinear method of mapping in equation 4, enhances the process proficiency by minimizing the execution time. Bilinear Quadrilateral Mapping investigates the request made by various cloud data stream files for mapping the file in the table is shows in Figure 3. Time-based mapping is a critical criteria in the E-tree MSI procedure to enhance the client inquiry handling on “F” data stream files. The consistent Spatio-temporal data are updated in the distributed storage framework, and the compelling E-tree MSI system is improve to enhance the load balance factor with a minimal time factor. The unmapped procedure is removed to enhance the client request result getting based distributed storage data stream. Bilinear quadrilateral transformation with extra scheduling and ordered process serves to effectively outcome the effective execution time. The E-tree MSI system with a bilinear map is a capacity that consolidates the components of the left and right tree branch to bring the outcome in all around composed manner. Subsequently, the tree-based order is completed to decrease the execution time variable.

3. Implementation and Evaluation

In this section, the implementation and performance evaluation of the Ensemble Tree Metric Space Indexing (E-tree MSI) method is discussed. The analysis compares Ensemble Models on Data Streams (E-Tree)¹⁶ and Multi-tier Application Component Deployments (MACD)²⁷ on IaaS clouds. The implementation on CloudSim Simulator with Java Platform. The Adult Data Set³ from the UCI repository files.

This data set contains 48842 instances with the mixture of continuous and discrete values. All the instances represent the population of specified socio-economic data. Each file has different attributes processes used for the processing the cloud data storage information. Initially, the populated data is planned in the cloud. At that point, the planned data is arranged taking into account the require giving out of the cloud client. here, individuals with same qualities are mapped together through comparable weightages. Various samples of data are taken for effective client handling. The examination is directed on the measurements, for example, the linear load factor component measure, the characterization correctness rate and the execution time for mapping.

4. Simulation Results

4.1 Performance Metrics

The performance assessment of the Ensemble Tree Metric Space Indexing (E-tree MSI) technique in the cloud computing platform is compared with the existing Ensemble Models on Data Streams (E-Tree)¹⁶ and Multi-tier Application Component Deployments (MACD)²⁷ in the cloud network by using CloudSim Simulation. The performance is evaluated according to the following metrics:

The Load Balancing factor in the E-tree MSI procedure measures the quantity of customer request hold effectively by the cloud server at a specific time interval. The Load Balancing factor measure in terms of percentage rate (%).

$$LLBF = \frac{(Client_R - Client_H) * t}{Client_R} \quad (5)$$

The linear load balance factor ‘LLBF’ is the ratio of the differentiation between the client requests ‘Client_R’ made

and handled ' $Client_H$ ' to the client requests made in cloud at time ' t '. The classification accuracy is the amount of spatio-temporal data stream properly classified on cloud. The higher the arrangement correctness is the more productive the strategy.

The classification accuracy in the E-tree MSI procedure measures the proportion of distinction between the expectation of actual classification and the prediction of the forecasted classification to the prediction of the actual classification and measured in terms of percentage (%).

$$CA = \frac{(Actual_p - Forecasted_p)}{Actual_p} * 100 \quad (6)$$

The execution time for mapping refers to the time taken to execute the Bilinear Quadrilateral Mapping (BQM) process using the index table.

$$ET = \text{Time} (IT[\text{Left C}(\text{Process}_i | t) + \text{Right C}(\text{Process}_i | t)] \quad (7)$$

The execution time in the E-tree MSI procedure is the time taken to perform the left tree and right tree arrangements stored in the index table. It is measured regarding milliseconds (ms).

4.2 Performance Comparison of Load Balance Factor

The execution measure of the Load Balance Factor and an examination is made with the current strategies the Ensemble Models on Data Streams (E-Tree)¹⁶ and Multi-level Application Component Deployments (MACD)²⁷ in the cloud system. Figure 4 demonstrates the consequence of the Load Balance Factor versus the number of client request. To better perceive the efficacy of the proposed E-tree MSI technique, substantial experimental results

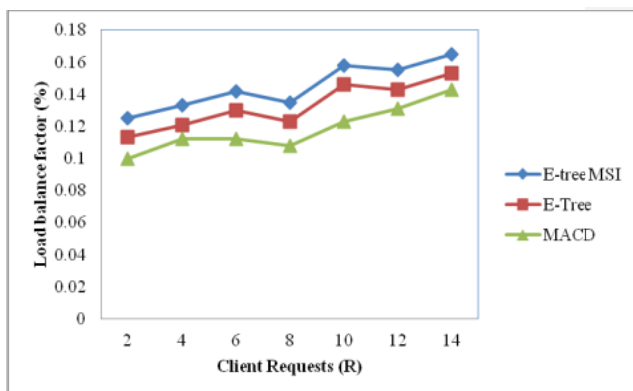


Figure 4. Measure of Load Balance Factor.

are illustrated in Figure 4 and are compared against the existing E-Tree and MACD, respectively.

Results are presented for different numbers of client requests with different resources in cloud network. The load balance factor in the cloud network for different resources sent at the speed of 20 ms is shown below. The results reported here confirm that with the increase in the number of client request being made, the load balance factor also increases. The process is repeated for 14 cloud users for conducting experiments.

4.3 Performance Comparison of Classification Accuracy Rate

In order to increase the classification accuracy rate with the dynamical client requests made in the cloud network, the prediction of the actual classification, and the prediction of the forecasted classification is considered. In the experimental setup, the data stream file ranges from 3 to 21 data. The results of 21 data stream files of packets of different age categories ranging from 20 to 25 are illustrated in Figure 5. The classification accuracy using the technique E-tree MSI provides comparable values than the state-of-the-art methods.

4.4 Performance Comparison of Execution Time for Mapping

In the Figure 6, which demonstrates the execution time for mapping using the E-Tree MSI method, E-Tree and MACD versus different cloud client's requests? The execution time for mapping returned over E-Tree and MACD increments progressively however not linear to differ cloud clients due to the request of cloud clients in various time periods.

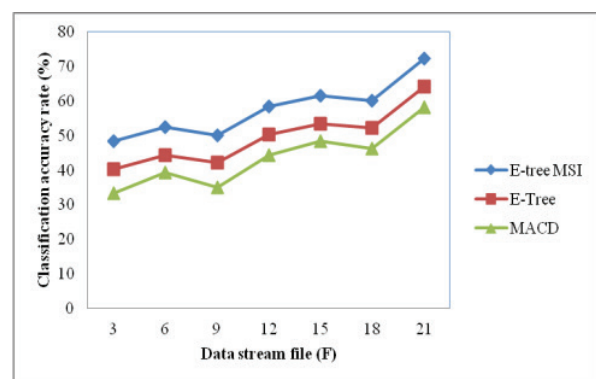


Figure 5. Measure of classification accuracy rate.

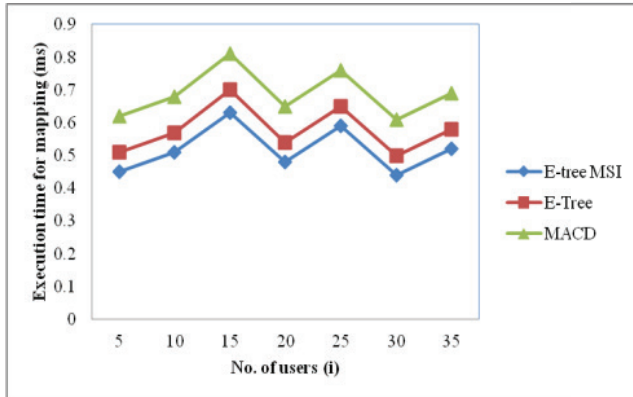


Figure 6. Measure of execution time for mapping.

5. Conclusion

In cloud infrastructure very large amount of data are processed and computation of these data over loaded. The Ensemble Tree Metric Space Indexing (E-tree MSI) methods builds a faster prediction and effective balancing of load factor on the cloud infrastructure in order to reduce the computation. MSI technique in the E-tree perform the classification procedure on the cloud data streams and the E-tree component is used for constructing with Fast Similarity Query Search indexing. Fast Similarity Query Search uses the inter-intra bin pruning technique and improves the grade of performance. An experimental evaluation is carried out on the factors of CPU excursion load rate, time complexity level, prediction time, indexing time, and such that the overall performance rate is increased.

6. Reference

- Schneider S, Andrade H, Gedik B, Biem A, Wu KL. Elastic scaling of data parallel operators in stream processing. *IEEE International Symposium on Parallel & Distributed Processing*. 2009; 2009 May 23-29. p. 1-12.
- Dautov R, Stannett M, Paraskakis I. On the role of stream reasoning in run-time monitoring and analysis in autonomous systems. *Proceedings of the 8th South East European Doctoral Student Conference*. Thessaloniki: Greece; 2013.
- Valle ED, Ceri S, Harmelen VF, Fensel D. It's a streaming world! Reasoning upon rapidly changing information. *Intell System IEEE*. 2009; 24(6):83-9.
- Carbunar B, Tripunitara MV. Payments for outsourced computations. *IEEE Transactions On Parallel And Distributed Systems*. 2012 Feb; 23(2):313-20.
- Koteeswaran S, Visu P, Kannan E. Enhancing JS- MR based data visualisation using YARN. *Indian Journal of Science and Technology*. 2015 Jun; 8(11):1-5. doi: 10.17485/ijst/2015/v8i11/71780.
- Sugumaran M, BalaMurugan B, Kamalraj D. An architecture for data security in cloud computing. *Proceedings of the International Conference on Information Technology and Applications*, IEEE Computer Society; 2013. p. 252-55.
- Monikandan S, Arockiam L. Confidentiality technique to enhance security of data in public cloud storage using data obfuscation. *Indian Journal of Science and Technology*. 2015 Sep; 8(24):88-97. doi: 10.17485/ijst/2015/v8i24/80032.
- Kalpana. V, Meena. V. Study on data storage correctness methods in mobile cloud computing. *Indian Journal of Science and Technology*. 2015 Mar; 8(6):495-500. doi: 10.17485/ijst/2015/v8i6/70094.
- Zhu Y, Hu H, Ahn GJ, Yu M. Cooperative provable data possession for integrity verification in multi-cloud storage. *IEEE Transactions On Parallel And Distributed Systems*. 2012 Dec; 23(12):2231-44.
- Xu G, Pang J, FuJaya X, Chintalapati B, Rao S. A load balancing model based on cloud partitioning for the Public Cloud. *IEEE Transactions on Cloud Computing*. 2013 Feb; 18(1):34-9.
- Gulisano V, Peris RJ, Martinez MP, Soriente C, Valduriez P. Streamcloud: an elastic and scalable data streaming system. *IEEE Transactions on Parallel and Distributed Systems*. 2012 Dec; 23(12):2351-65.
- Xu G, Pang J, FuJaya X, Chintalapati B, Rao S. A load balancing model based on cloud partitioning for the public cloud. *IEEE Transactions on Cloud Computing*. 2013 Feb; 18(1):34-9.
- Sheth A, Henson C, Sahoo SS. Semantic sensor web. *IEEE Internet Computing*. 2012 Jul-Aug; 12(4):78-83.
- Sugumaran M, BalaMurugan B, Kamalraj D. An architecture for data security in cloud computing. *World Congress on Computing and Communication Technologies (WCCCT)*; 2014 Feb-Mar 27-1. p. 252-55.
- Balamurugan B, Kamalraj D, Jegadeeswari S, Sugumaran M. A Framework to Data Stream with Balanced Load Factor using Ensemble-Tree in Cloud. *International Journal of Applied Engineering Research*. 2015; 10(72):204-7.
- Zhang P, Zhou C, Wang P, Gao BJ, Zhu X, Guo L. E-Tree: an efficient indexing structure for ensemble models on data streams. *IEEE Transaction on Knowledge and Data Engineering*. 2015 Feb; 27(2):461-74.
- Wang P, Gao J, Zhu X, Guo L. Enabling fast prediction for ensemble models on data streams. *International Conference on Knowledge Discovery and Data Mining, USA*; 2011. p. 177-85.
- Valle DE, Ceri S, Barbieri DF, Braga D, Campi. A first step towards stream reasoning. *Future Internet-FIS*, Springer: Berlin Heidelberg; 2009. p. 72-81.

19. Alzahrani AS, Qureshi MS. Privacy preserving optimized rules mining from decision tables and decision trees. *Indian Journal of Science and Technology*. 2012 Jun; 5(6):2831–34. doi: 10.17485 /ijst/2012/v5i6/30471.
20. Lloyd W, Pallickara S, David O, Lyonb J, Arabi M, Rojas K. Performance implications of multi-tier application deployments on Infrastructure-as-a-Service clouds: towards performance modeling. *Future Generation Computer Systems*. 2013 Jul; 29(5):1254–64.
21. Lanzasanto N, Komazec S, Toma I. Reasoning over real time data streams; 2012.
22. Thenmozhi M, Srimathi H. An analysis on the performance of tree and trie based dictionary implementations with different data usage models. *Indian Journal of Science and Technology*. 2015 Feb; 8(4):364–75. doi: 10.17485/ijst/2015/v8i1/59865.
23. Venkatesan P, Yamuna NR. Treatment response classification in randomized clinical trials: a decision tree approach. *Indian Journal of Science and Technology*. 2013 Jan; 6(1):3912–17. doi: 10.17485/ijst/2013/v6i1/30563.
24. Margaraa A, Urbania J, Harmelena FV, Bala H. Streaming the web: reasoning over dynamic data. *Journal of Web Semantics*. 2014; 25:1–6.
25. Reasoning over real time data streams [Internet]. [Cited 2012 Sep 25]. Available from: <http://www.envision-project.eu/wp-content/uploads/2012/11/D4.8-1.0.pdf>.
26. Gedik B, Schneider S, Hirzel M, Wu K. Elastic scaling for data stream processing. *IEEE Transactions Parallel Distributed Systems*; 2013. p.1–6.
27. Yiu ML, Assent I, Jensen CS, Kalnis P. Outsourced similarity search on metric data assets. *IEEE Transactions on Knowledge and Data Engineering*. 2012 Feb; 24(2):338–52.
28. Kaur SA, Kinger S. Analysis of load balancing techniques in cloud computing. *International Journal of Computers & Technology*. 2013 Mar–Apr; 4(2):737–41.