Frequent Subgraph Mining for Graph based Tamil Bibliographic Big Data Analytics

Elangovan.G, Dr.Kavya.G, Kalpana.A.V, Jagadish kumar.N, Anwar basha.H

Abstract: ***Data analysis can be done in more effectively, when they are represented in the form of graphs. Especially, Frequent Subgraph Mining (FSM) is an important technique for extracting similar patterns in the graphs. Normally, things have been assumed as the graph data we are taking will fit in main memory for their processing. But, as the data grow higher and higher, they will not fit in main memory, rather they need a special framework called MapReduce to put them in a distributed fashion and to process. Many Frequent Subgraph Mining (FSM) algorithms are changing their faces to adopt the MapReduce programming paradigm. Using FSM-H, analysis had been performed on various graph based data like molecular structures, viral patterns etc., Even DBLP i.e., the computer science bibliography data have also been analyzed using this pattern extraction technique. Whereas the details of Tamil Journals and Publications are kept hidden and not available widely to do research on them, for getting their insights to improve the number and quality of the journals; and to give some input for the authors interested to work on Tamil research. In this work, we collected the Tamil journals details from the available data sources and extracted essential patterns using Frequent Subgraph Mining Technique. Also, we presented a detailed statistical analytics on certain frequently happening Tamil Journals and Conferences.***

Keywords: ***Especially, Frequent Subgraph Mining, MapReduce programming paradigm, DBLP.***

I. INTRODUCTION

In recent years the evolution of Big Data Analytics has enormous impact on research and in application domains including information mining, computational data retrieval [1], ecological sciences, e-business [2], web mining, and interpersonal organization investigation [3]. In these areas, examining and mining of enormous information for separating novel experiences has turned into a standard assignment. Notwithstanding, customary strategies for information examination and mining are not intended to handle gigantic measure of information, so as of late numerous such techniques are re-outlined and re-actualized under a registering system that is better prepared to handle big data problems.

Among the late endeavors for building an appropriate registering stage for dissecting huge information, the MapReduce [4] structure of dispersed figuring has been the best. It embraces information driven methodology of circulated figuring with the philosophy of taking computation to data; other than it utilizes a conveyed document framework that is especially streamlined to enhance the IO execution while taking care of huge information. Another primary reason for this structure to pick up consideration of numerous admirers is the more elevated amount of deliberation that it gives, which keeps numerous framework level points of interest avoided the developers and permit them to focus more on the issue particular computational rationale.

MapReduce has turned into a well-known stage for breaking down vast systems as of late. Nonetheless, the larger part of such examinations are constrained to assessing worldwide measurements (for example, distance across) [5], ghostly investigation [6], or vertex-centrality examination [5]. There additionally exist some works that mine (and number) sub-structures from an expansive system. For occasion, Suri and Vassilvitskii [7] and Pagh and Tsourakakis [8] use MapReduce for tallying triangles, Afrati et al. [9] use MapReduce for specifying the occurrences of an inquiry diagram in a substantial system, and Bahmani et al. [10] mine densest subgraphs in huge graphs. Be that as it may, mining successive subgraphs from a diagram database has gotten the slightest consideration. Given the development of utilizations of Frequent Subgraph Mining (FSM) in different orders including interpersonal organizations, bioinformatics [11], cheminformatics [12], and semantic web [13], an adaptable strategy for incessant Subgraph Mining on MapReduce is of popularity.

Illuminating the assignment of frequent subgraph mining on MapReduce is trying for different reasons. Initial, a FSM technique processes the backing of an applicant subgraph design over the whole arrangement of information charts in a diagram dataset. In a disseminated stage, if the info charts are parceled over different specialist hubs, the neighborhood backing of a subgraph in the separate segment at a laborer hub is very little valuable for choosing whether the given subgraph is visit or not.

In this paper, we propose, FSM-H, an appropriated incessant subgraph mining strategy over MapReduce. Given the bibliography dataset of Tamil journals, and a base bolster edge, FSM-H creates a complete arrangement of frequent subgraphs. To guarantee culmination, it builds and holds all examples in a parcel that has a non-zero backing in the guide period of the mining, and afterward in the diminish stage, it chooses whether an example is continuous by collecting its

Revised Manuscript Received on November 12, 2019.

* Correspondence Author

Elangovan G, Assistant Professor, Department of Computer Science & Engineering, Velammal Institute of Technology, Chennai.

Dr Kavya G, Professor, Department of Electronics & Communication Engineering, S.A Engineering College, Chennai.

Kalpana A V, Assistant Professor, Department of Computer Science & Engineering, Velammal Institute of Technology, Chennai.

backing registered in all allotments from various processing hubs. To conquer the reliance among the conditions of a mining procedure, FSM-H keeps running in an iterative manner, where the yield from the reducers of cycle i-1 is utilized as a contribution for the mappers in the cycle i. The mappers of cycle i create applicant subgraphs of size i (number of edge). The reducers of emphasis i then locate the genuine continuous subgraphs (of size i) by totaling their neighborhood bolsters. They likewise compose the information in plate that is handled in resulting cycles.

II. RELATED WORKS

There exist numerous calculations for unraveling the in-memory form of frequent subgraph mining assignment; most outstanding among them are Gaston [18], gSpan [17] and FSG [16]. These strategies accept that the dataset is little and the mining task completes in a sensible measure of time utilizing an as a part of memory strategy. To consider the huge information situation, a couple of conventional database based chart mining calculations, for example, DB-Subdue [20], and DB-FSG [21] are likewise proposed.

MapReduce system has been utilized to mine successive examples where the exchanges in the info database are more straightforward combinatorial questions. In [22], the creators consider incessant subgraph mining on MapReduce, be that as it may, their methodology is wasteful because of different weaknesses. The most remarkable is that in their technique they don't embrace any system to abstain from producing copy designs. This cause an exponential increment in the measure of the competitor subgraph space; moreover, the yield set contains copy duplicate of the same chart designs that are difficult to bind together as the client needs to give a subgraph isomorphism routine to this alteration.

III. FREQUENT SUB-GRAPH MINING

Consider a set of connected undirected graphs of the areas of interest of various authors contributed in publishing Tamil related research papers. The number of edges in the graph represents the size of the graph. To decide the frequency of the patterns, a threshold value is fixed as the support value. While checking the support cardinality, it is to be checked that whether the support value getting is greater or equal to the minimum threshold support value. If the support value is greater than the support value, then the pattern is frequent. In such a way the frequent patterns are collected from graph size 1 to ‘n’, where n is the maximum number of edges in the graph having largest number of edges among all the graphs in the dataset.

IV. MAPREDUCE

MapReduce is a programming model that empowers conveyed calculation over monstrous information [4]. The model gives two unique capacities: Maper, Reducer. Map relates to mapping of relevant data and Reduce use to compare and combines certain processed intermediate entities in practical programming. In view of its part, a specialist hub in MapReduce is known as a mapper or a reducer. A mapper takes a gathering of (key,value) matches and applies the guide capacity on each of the sets to produce a discretionary number of (key, value) sets as middle of the road yield. The reducer totals all the qualities that have the same key in a sorted rundown, and applies the decrease capacity on that rundown. It likewise composes the yield to the yield document. The records (input and output) of MapReduce are overseen by a conveyed document framework.

V. CANDIDATE GENERATION

A frequent pattern c of size k is taken, this progression borders an incessant edge (which has a place with F1) with c to acquire an applicant design d of size k + 1. In the event that d contains an extra vertex then the additional edge is known as a forward edge, else it is known as a back edge; the last essentially interfaces two of the current vertices of c. Extra vertex of a forward edge is given a number id, which is the biggest whole number id taking after the ids of the current vertices of c; in this manner the vertex-id remains for the request in which the forward edges are appended while building a competitor design. In diagram mining wording, c is known as the guardian of d, and d is an offspring of c, and taking into account this guardian tyke relationship we can orchestrate the arrangement of applicant examples of a mining errand in a hopeful generation tree (see Fig. 1)

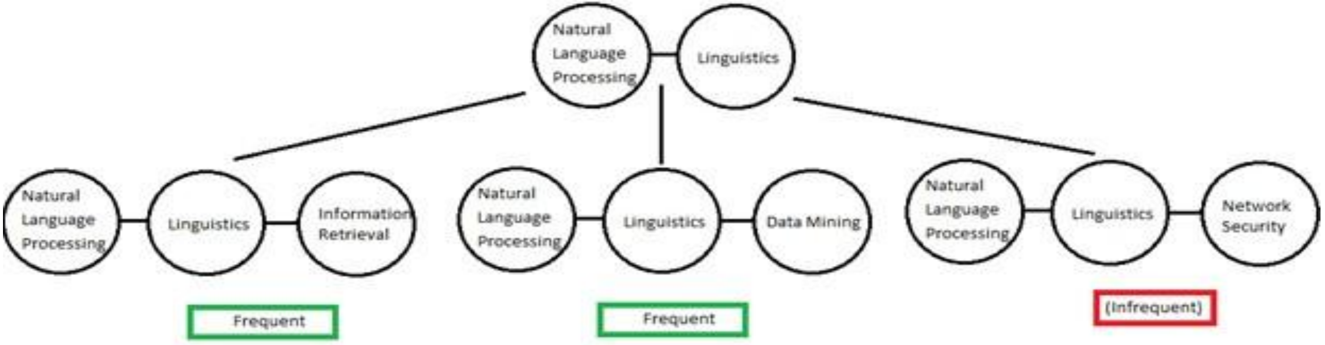

Fig. 1. Candidate generation subtree rooted under Natural Language Processing – Linguistics

Note that, if d has k + 1 edges, in view of the request how its edges have been appended, d could have a wide range of generating ways in a candidate generation tree; in any case, in all FSM calculations, stand out of the generation ways is viewed as legitimate, so that different duplicates of an applicant example are not created. With this confinement of candidate generation, the hopeful generation tree of a FSM task can be unambiguously characterized.

VII. ISOMORPHISM CHECKING

As we say in past section, a hopeful example can be produced from various generation paths, yet one and only such way is investigated among the candidate generation step and the remaining ways are recognized and in this way disregarded. To distinguish invalid candidate generation paths, a graph mining algorithm calculation needs to settle the diagram isomorphism task, as the copy duplicates of hopeful examples are isomorphic to each other. A surely understood strategy for distinguishing graph isomorphism is to utilize canonical coding plan, which serializes the edges of a diagram -utilizing a recommended arrange and produces a string such that all isomorphic diagrams will create the same string.

There are a wide range of accepted coding plans; min-dfs-code is one of those which is utilized as a part of [17]. As indicated by this plan, the generating way of an example in which the insertion request of the edges matches with the edge requesting in the min-dfs-code is consideredas the legitimate generation way, and the remaining generation ways are considered as copy and henceforth overlooked. FSM-H utilizes min-dfs-code based standard coding for isomorphism checking.

VI. EXPERIMENT RESULTS AND INFERENCES

Here we present the experimental results that explain the performance of FSM-H and its necessity for solving the frequent pattern mining problems. Real world Tamil Journal data sets have been taken as input from the online sources such as Research Articles on Tamil Language and Linguistics, Journal of Tamil Studies and International Forum for Information Technology in Tamil (INFIT). Available data have been cleaned and converted as graph dataset to feed in to over system for finding the frequent patterns and by the way other related statistical details have been obtained.

Source URL:
http://www.ulakaththamizh.org/JOTSissues.aspx
http://tamilelibrary.org/teli/tpapers2.html
http://home.infitt.org/ti-conference-papers/

A. *Research Articles on Tamil Language and Linguistics*

The dataset from Research Articles on Tamil Language and Linguistics has details of Tamil journals listed by various authors throughout the country.

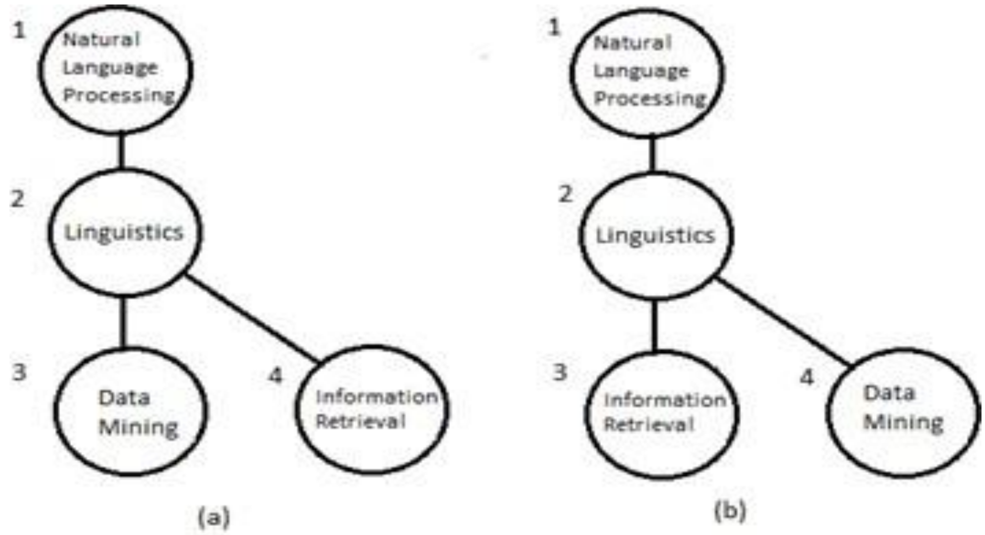

Fig. 2. Graph Isomorphism

In this forum, it gives the publication details such as the title, journal details and the number of journals published by a specific author. From the available details the various other details have been extracted and put it in a graph based data representation for finding various patterns.

In this forum, it gives the publication details such as the title, journal details and the number of journals published by a specific author. From the available details the various other details have been extracted and put it in a graph based data representation for finding various patterns.

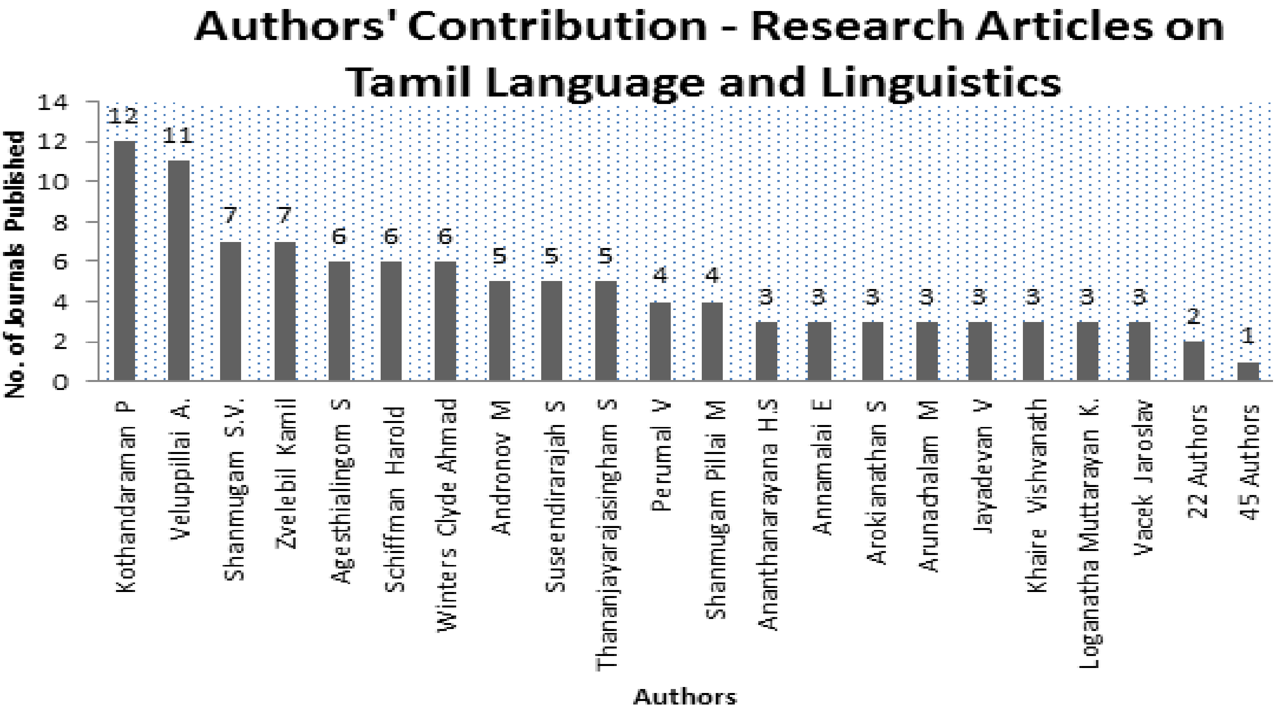

Fig 3. Authors' Contribution in various tamil conferences as per Tamilelibrary

In this forum, it gives the publication details such as the title, journal details and the number of journals published by a specific author. From the available details the various other details have been extracted and put it in a graph based data representation for finding various patterns.

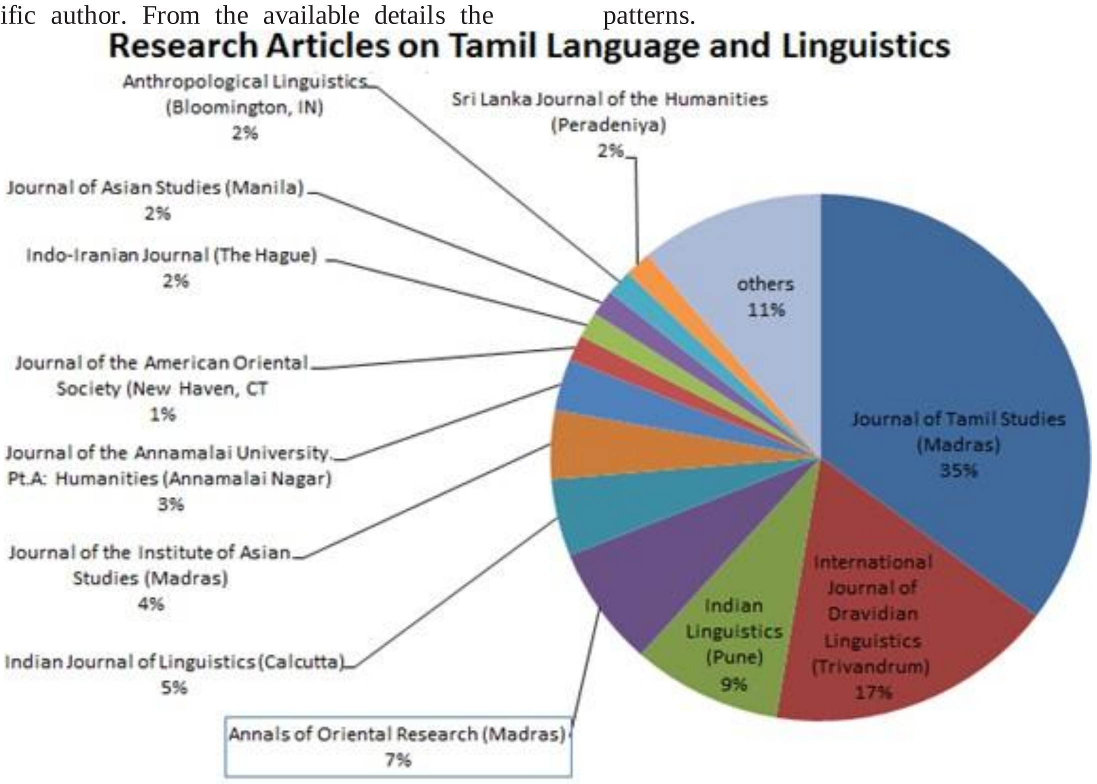

Fig. 4. Journalwise Distribution

Journal of Tamil Studies

The Journal of Tamil Studies considered as the oldest and an origination of publishing researching works on Tamil. International Institute of Tamil Studies possesses an authentic and widely gathered data on Tamil journals. Journal for Tamil studies have been started first for publishing some works on Tamil literature. Papers can be published in both Tamil and English in this Journal of Tamil Studies. Apart from the research papers, book can also be published in these journals. In year 1991 more number of books has been published when compared to all the other years. It can be inferred from the graph shown in Fig 5.

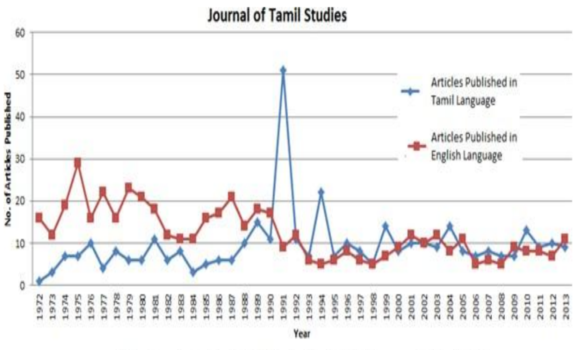

Fig 5 : Year wise Articles Published in Tamil and English languages for Tamil Studies

B. *International Forum for Information Technology in Tamil (INFITT)*

INFITT is a non-profit organization for Tamil Language. Its intention is to encourage people working in Tamil research. It use to conduct conferences regularly from the year 2000 in Tamil speaking countries like India, Singapore, Malaysia etc., Even it collaborates with the Tamil government organizations and universities of those countries to enhance the research on Tamil Literature and other related fields. INFITT's Tamil Internet Conference papers are available in its website in various formats. Data from them are cleaned and a graph representation of them is feed in our frequent subgraph mining system to bring the similar patterns in various dimensions of the data. Patterns have been recognized and a statistical data of year wise publications and country wise authors contribution have been inference and shown in Fig 6.

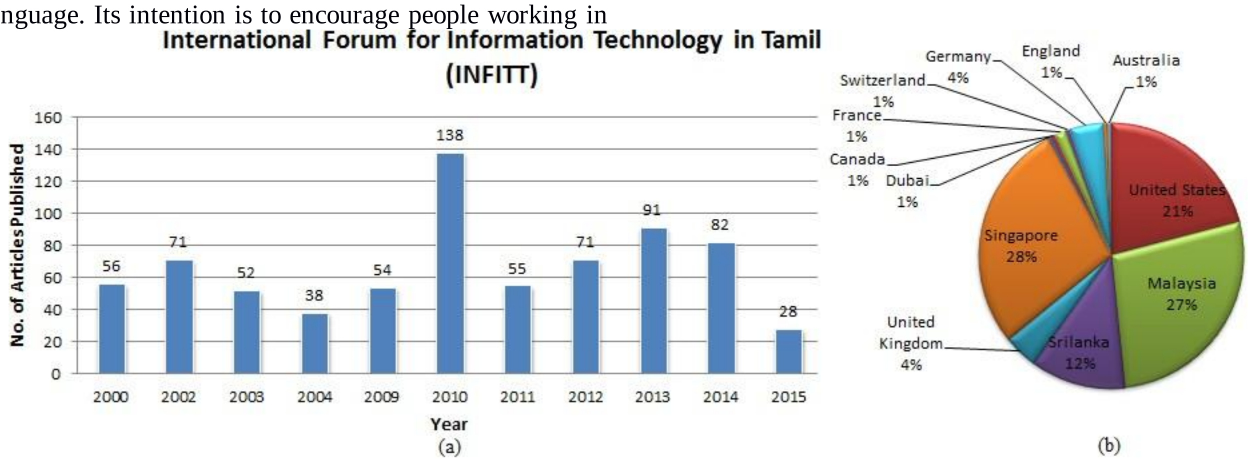

Fig 6 : Tamil Internet Conference of INFITT (a) Year-wise Publications (b) Other Countries Contribution on Publications apart from India

IX. CONCLUSION

In this paper we presented an extensive MapReduce technique for finding the similar patterns in data using Frequent Subgraph Mining. We have taken the Tamil bibliography datasets from the real world Tamil data source such as Journal of Tamil Studies, Research Articles on Tamil Language and Linguistics and International Forum for Information Technology in Tamil (INFITT) and their cleaned graph formatted data have been used to bring various frequent patterns in them. By extracting those similar patterns several related inferences have been noticed and a detailed statistical report has been presented. Further, Tamil data from other sources; globally from various worldwide libraries can be used as input to our system to do a wide range of big data analytics to infer much more details; so that they can be used in the growth of Tamil literature and other research works on Tamil Linguistics.

REFERENCES

[1] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds, “Cloud computing: A new business paradigm for biomedical information sharing,” J. Biomed. Inf., vol. 43, pp. 342–353, 2010.

[2] W. Lam, L. Liu, S. Prasad, A. Rajaraman, Z. Vacheri, and A. Doan, “Muppet: Mapreduce- style processing of fast data,” Very Large Data Bases Endow., vol. 5, pp. 1814–1825, 2012.

[3] G. Liu, M. Zhang, and F. Yan, “Large-scale social network analysis based on Mapreduce,” in Proc. Int. Conf. Comput. Aspects Soc. Netw., 2010, pp. 487–490.

[4] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” Commun. ACM, vol. 51, pp. 107–113, 2008.

[5] U. Kang, C. E. Tsourakakis, and C. Faloutsos, “Pegasus: A petascale graph mining system implementation and observations,” in Proc. 9th IEEE Int. Conf. Data Mining, 2009, pp. 229–238.

[6] U. Kang, B. Meeder, and C. Faloutsos, “Spectral analysis for billion-scale graphs: Discoveries and implementation,” in Proc. 15th Pacific-Asia Conf. Adv. Knowl. Discov. Data Mining, 2011, pp. 13–25.

[7] S. Suri and S. Vassilvitskii, “Counting triangles and the curse of the last reducer,” in Proc. 20th Int. Conf. World Wide Web, 2011, pp. 607–614.

[8] R. Pagh and C. E. Tsourakakis, “Colorful triangle counting and a mapreduce implementation,” Inf. Process. Lett., vol. 112, no. 7, pp. 277–281, 2012.

[9] F. Afrati, D. Fotakis, and J. Ullman, “Enumerating subgraph instances using map-reduce,” in Proc. IEEE 29th Int. Conf. Data Eng., Apr. 2013, pp. 62–73.

[10] B. Bahmani, R. Kumar, and S. Vassilvitskii, “Densest subgraph in streaming and mapreduce,” Proc. Very Large Data Bases Endow., vol. 5, no. 5, pp. 454–465, Jan. 2012.

[11] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, and J. Prins, “Mining protein family specific residue packing patterns from protein structure graphs,” in Proc. Int. Conf. Res. Comput. Mol. Biol., 2004, pp. 308–315.

[12] S. Kramer, L. Raedt, and C. Helma, “Molecular feature mining in HIV data,” in Proc. Int. Conf. Knowl. Discov. Data Mining, 2004, pp. 136–143.

[13] B. Berendt, “Using and learning semantics in frequent subgraph mining,” in Proc. Adv. Web Min. Web Usage Anal., 2006, pp. 18–38.

[14] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[15] A. Inokuchi, T. Washio, and H. Motoda, “An apriori-based algorithm for mining frequent substructures from graph data,” in Proc. 4th Eur. Conf. Principles Data Mining Knowl. Discov., 2000, pp. 13–23.

[16] M. Kuramochi and G. Karypis, “Frequent subgraph discovery,” in Proc. Int. Conf. Data Mining, 2001, pp. 313–320.

[17] X. Yan and J. Han, “gSpan: Graph-based substructure pattern mining,” in Proc. Int. Conf. Data Min., 2002, pp. 721–724.

[18] S. Nijssen, and J. Kok, “A quickstart in frequent structure mining can make a difference,” in Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2004, pp. 647–652.

[19] V. Chaoji, M. Hasan, S. Salem, and M. Zaki, “An integrated, generic approach to pattern mining: Data mining template library,” Data Min. Knowl. Discov. J., vol. 17, no. 3, pp. 457–495, 2008.

[20] S. Chakravarthy, R. Beera, and R. Balachandran, “Db-subdue : Database approach to graph mining,” in Proc. Adv. Knowl. Discov. Data Mining, 2004, pp. 341–350.

[21] S. Chakravarthy and S. Pradhan, “Db-FSG: An SQL-based approach for frequent subgraph mining,” in Proc. 19th Int. Conf. Database Expert Syst. Appl., 2008, pp. 684–692.

[22] S. Hill, B. Srichandan, and R. Sunderraman, “An iterative Mapreduce approach to frequent subgraph mining in biological datasets,” in Proc. ACM Conf. Bioinformat., Comput. Biol. Biomed., 2012, pp. 661–666.

AUTHORS PROFILE

G.Elangovan received B.E. Degree in Computer Science and Engineering from Anna University, Chennai. Then, he received M.E Degree in Computer Science and Engineering from Bannari Amman Institute of Technology, Anna University, Coimbatore

G. Kavya received her B.E. Degree in Electronics and Communication Engineering from Government College of Engineering, Salem under Madras University. Then, she completed her M.E in Electronics Engineering from Madras Institute of Technology, Anna University, Chennai and her Ph.D. in Electronics Engineering from Sathyaama University, Chennai, India 2015.

A.V. Kalpanais currently an Assistant Professor in the Department of Computer Science and Engineering at RMK Engineering College. She obtained her B.E. in Computer Science & Engineering in the year 2004 from University of Madras. She obtained her M.E. in Computer Science & Engineering in the year 2011 from Anna University. Her research interests include Wireless Networks, Mobile Computing and Wireless Sensor Networks.

N.Jagadish kumar B.E Degree in Computer Science and Engineering from Anna University,Chennai.Then he received M.E degree in Computer Science and Engineering from Madras Institute of Technology, Anna University,Chennai.

Anwar Basha H B.E Degree in Computer Science and Engineering from Anna University,Chennai.Then he received M.Tech degree in Computer Science and Engineering from Dr. MGR Educational and Research Institute University,Chennai.