International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-3, January 2020

Managing Multiversion Xml Documents with Compressed Delta

Vijay R. Sonawane, D. Rajeswara Rao

Abstract: ***Today web contains huge amount of information. Important part is to obtain business value from existing information. XML is useful to accomplish features of business functions. Hence to attain those promises it is important to retain those XML documents effectively. Content and structure of Dynamic XML documents changes frequently based on user behavior and produces multiple versions of it. Multiversion XML documents are having huge applicability which demands for their effective organization. Clustering is better solution to retain these documents. But due to dynamic nature of XML documents initially obtained cluster compositions becomes invalid in later stage and traditional methods are not effective to obtain most recent clustering solutions.***

In this paper we are proposing time efficient technique called Compressed Delta (CΔ) in response to obtain most recent clusters of multiversion XML documents. CΔ contains sufficient information to get essential document version with minimum operational cost for future references. Proposed CΔ is comprehensively assessed on several real-life datasets exhibiting extreme characteristics. Experimental results shows that proposed (CΔ) to obtain required XML document version outperform the related state-of-the-art approaches.

Keywords: XML, Multiversion, Compressed Delta, Homomorphic.

I. INTRODUCTION

Today electronic business (E-business) is an important term which brought greatest changes in the business sectors. In E-businessinvolving partiesuses common platform to exchange structured information. XML assures various business functions like content assimilation, intelligence and salvage. XML is platform that recognize portion of information using content and syntax. XML provides ability to define information using set of vocabularies. To achieve the promises of XML it is important to handle XML documents appropriately [1]. XML documents are static and dynamic in nature. Content of static document is stationary but the content of dynamic document changes with time and its ratio of change is depends on e-customer behavior. These changes are real time and customer specific, but infinite, surprising and create many versions of single documents called as Multiversion XML documents [2-3].

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author
Dr. Vijay R. Sonawane*, Associate Professor, MVPs K.B.T College of Engineering, Nashik Savitribai Phule Pune University, Pune (Maharashtra) India. E-mail; vijaysonawane11@gmail.com
Dr. D. Rajeswara Rao, Professor and Head, Department of CSE, V R Siddhartha Engineering College Vijayawada (Andhra Pradesh) India. E-mail; hodcse@vrsiddharth.ac.in

Multiversion document basically comprisesat least one element which specifies the genuinity of theversion.
Temporal element present in the document specifies the validity of that multiversion XML document and older version become invalid when new version appears. XML proposes organization of metadata to generate the vocabulary to exchange the information with connecting data sources via the internet [4]. Increased use of XML recommends the warehousing of XML documents for future use. Variations in the document content is the main cause to appear multiple versions of XML document.
Multiple version's of XML documents are useful in collaborative authoring, managing software versions, durability of web documents and warehousing knowledge from web. Multifaceted nature of XML documents for management and maintenance of information is demanding the multiversioning and suitable organization of XML documents [5]. Storage of each version of XML document may incur redundancy in XML documents collection. Therefore to store these multiple versions of documents need an effective technique so user will be able to get required document version in less effort.
To efficiently deal with this issue making group of similar XML documents is best solution, this grouping is called as clustering [6].Hence grouping similarXML documents in the form of clusters facilitates effective document management by providing improved querying, mining and data integration.
In spite of huge benefit of clustering an XML documents, it incur complexity in clustering process due to hierarchical nature of an XML documents. Hence traditional clustering methods are not useful to cluster an XML documents. In the distance based clustering method, distance signifies that when similarity between two XML documents is closest to each other that means their distance is less and be a part of single cluster.
Furthermore, clusters of static XML documents are always stable due to the nature of its content, so distances between clustered documents are always same. But due to the real-time changes based on customer behavior in multiversion XML documents influence the calculated distances amongst document versions, hence those documents may change their cluster composition and can become part of another cluster. So, to find most updated cluster compositions of multiple versions of an XML we are proposing a technique which makes use of "Compressed Delta" (CΔ). CΔ is standard XML document which stores the changes between successive XML documents versions from time T_1 to T_n.XML document version are compressed with Homomorphic compression (XGrind) which helps in reducing computing power of compressed delta.

CΔ can be directly revealed without applying decompression on it. CΔ holds adequate information to get essential document versions at any time period between T_1 to T_n when initial clustering solution becomes outdated with lowest operational cost.

II. MULTIVERSION XML DOCUMENTS

An example of multiversion XML document is shown in figure 1. Each consecutive version of document is achieved by using insert, update and delete operations on previous version of document. Distance between two documents indicates the total numbers of operations required to covert one document into other which is stored in the compressed delta (CΔ).

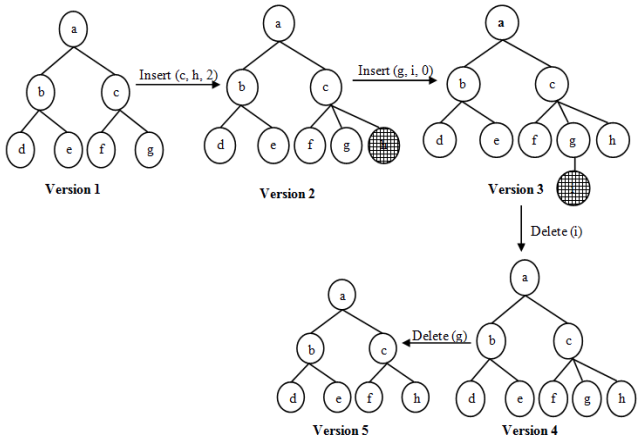

Figure1: An Example of Multiversion XML Document

Figure 2 shows example fragments of three real-time versions of online catalogue document obtained at three different time stamp T1 to T3. In real world these changes are infinite and depend on user behaviour. Document content affect partly or completely noted as degree of modification which is important in managing those documents efficiently.

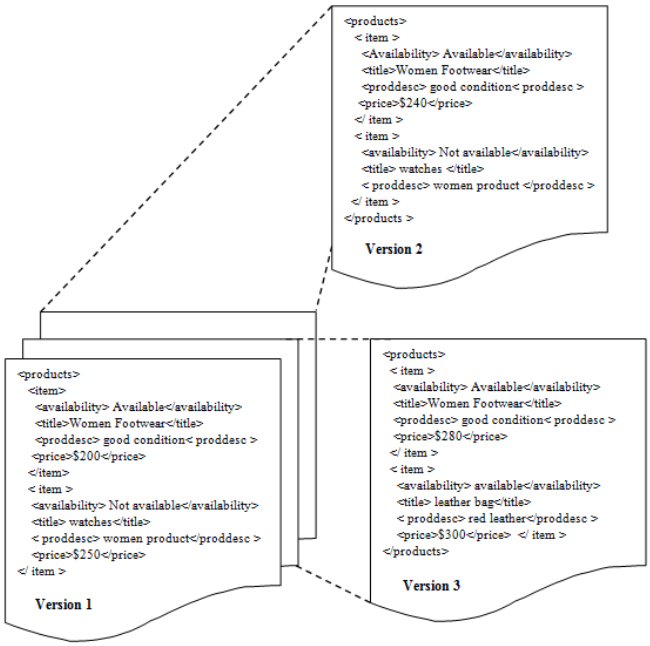

Figure 2: Fragment of XML Document Version

Delta (Δ) document denotes difference between two consecutive document versions. To reduce its storage space and computing power because of verbosity in XML documents we are using homomorphic compression tool (XGrind) to limit increasing size of XML document. Figure 3 shows the example fragment of XML document with its homomorphic compressed view. Homomorphic compression replaces all the elements by its precise values but also preserves document hierarchical structure in compressed format which is essential to calculate the similarity among XML document versions.

```
<!- student.xml -->
<STUDENT rollno = "88772255">
<NAME>Ramesh Wagh</NAME>
<YEAR>2015</YEAR>
<PROG>Master of
Engineering</PROG>
<DEPT name = "Computer Science">
</STUDENT>
```

```
<T0> A0 (88772255)
<T1> Ramesh Wagh</T1>
<T2> 2015</T2>
<T3> Master of Engineering </T3>
<T4> A1 enum(Computer Science) </T4>
</T0>
```

Figure 3: Homomorphic Compressed View

III. LITERATURE SURVEY

From available literature study it is observed that no existing technique performs time efficiently to obtain most recent cluster compositions of Multiversion XML documents using the concept of compressed delta (CΔ)[7].

XML version detection and management has tremendous application. To identify the versions, computation of similarities between different document versions and threshold values are vital. To develop similarity function content and structure content and structure similarity plays an important role.

Different schemes are proposed for discovering changes among multiple versions of XML documents using diff algorithm [8-11], Text [12], Structure [13-16] and Classification [17-18]. Comparison of different change detection methods requires investigation of different factors like object-referencing approach for change discovery, delta, relational, structured XML documents support, scalability, file volume, and illustration of unaffected parts [1]. Some change detection schemes clearly shows the user the changed part of the documents [19-21], whereas other may not [22-25].

Given methods to discover changes in XML documents should be scalable enough. The storage of in between full versions of XML documents improves the efficiency and space complexity as the required version can be created by using the suitable in-between complete version. Query processing becomes faster while a system stores the in-between complete versions because there is no need to rebuild the intermediate versions dynamically.

For clustering XML documents various methods are proposed in recent years. Scheme is proposed to obtain frequently changing structure (FCS) from consecutive versions of dynamic XML document and CDX method is proposed to cluster dynamic XML document with the help of FCS in [27]. With help of SemXClust framework XML document is divided into tree tuple set then use of XK-means algorithm and XFIHC algorithm is used to cluster XML document by using WordNet [28].

Weighted Element Tree Model (WETM) is projected in [29] for determining the structural similarity of XML documents. XML document is presented as ordered labelled trees then to find groups of structurally similar XML documents structural distance metrics is used in hierarchical clustering algorithms [30]. Mechanism for finding structural similarities among XML documents using graph-matching algorithms which permits adequate reduction of the essential computation costs is proposed in[31].

A various techniques for clustering sequence of heterogeneous XML documents projected by authors [32]. The theory of level structure is used to compute the similarity between arriving XML document and available clusters. Similarity is calculated at cluster level instead of pair-wise for single documents in the clusters.

IV. RECORDING CHANGES WITH COMPRESSED DELTA

Existing methods to record the changes between different versions of XML documents are discussed in literature survey. To deals with versioning issue completed delta approach has its limitation that needs to store either first or last XML document as well as all intermediate completed deltas. Figure 5 shows working of completed delta approach.

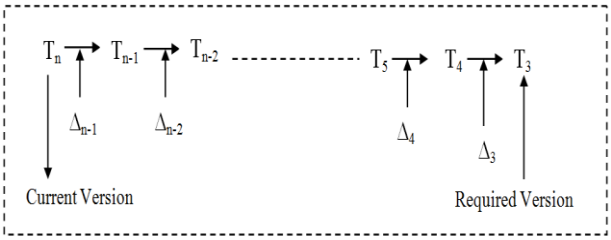

Figure 4: Completed Delta Approach

Storing all intermediate deltas incur recurring information when changes are frequent and to obtain required document versions it requires to run versioning algorithm number of times. To overcome these limitations we have proposed compressed delta (CΔ). It's simply a generic XML document which store total changes responsible for document transformation from time period T1 to Tn. Important to note here is to minimize storage space and computing power of delta document we are applying homographic compression on XML documents which maintains original hierarchical structure on XML document hence delta document can be directly discovered without decompressing XML documents. Temporal variable time stamp (*ST*) is maintained which comprises all the changes in time period T1 to Tn. *ST* records changed values all the times. *ST* has two components. Time component to records time value and delta component to record type of changes applied (insert, delete, update). CΔ is very effcient to record the changes between multiple document versions and to obtain necessary documents versions.

V. COMPRESSED DELTA (CΔ)

While recording changes responsible for transformation CΔ focus only on affected portion of the document than unaltered portion. To increases building time of compressed delta unique ID values are assigned to elements in original XML document and maximum id value is recorded, so next id value will be assigned on arrival of new element. Hence unique value will be maintained helpful for discovering the changes. Steps to be followed to build CΔ:

Step 1. Start with document Doc1 at time T1, for consecutive version Doci at Ti do next.
Step 2. Assign distinct ID value to each new inserted element in the Doci.
Step 3. Calculate the difference between current version Doci and new version Docj.
Step 4. For each altered elements in the version Dj, entry is recorded in CΔ (Ti and delta value)
Step 5. Doci is purged from warehouse and Docj is added in initial CΔ.
Step 6. Docj is preserved in warehouse till the arrival of new version and will be purged at the time of last run of algorithm.
Step 7. Finally CΔ will be store in warehouse.

Following rules are strictly followed while implementing CΔ:

Rule 1. If no children is affected, then parent remain unchanged →no entry recorded in CΔ.
Rule 2. If any children is affected, then parent will be changed → Entry will be recorded in CΔ for all child elements.
Rule 3. If any parent is deleted → delete entry is recorded in CΔ for all child elements.

To get CΔcompressed delta algorithm is repeatedly executed on arrival of new version in warehouse. At the end CΔ will contain preliminary XML version and all changes in time T1 to Tn. In CΔ instead of repetitive recording only affected element values are recorded. Figure 5 shows maintaining document versions with CΔ. Any document Doci at time Ti can be obtained directly by performing querying operation on CΔ.

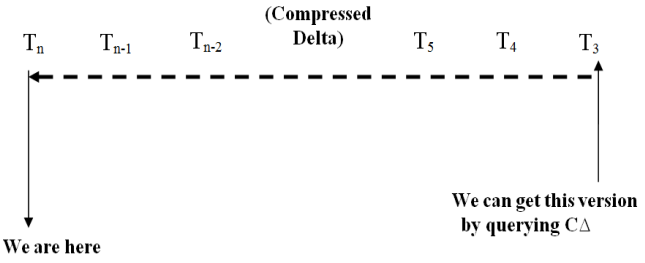

Figure 5: Versioning with Compressed Delta

Compressed delta algorithm (Cd_Algorithm):
To obtain document version with CΔ, document is examined from root element. If *ST*is observedwith any delta value, then version Doci will be as follows:

- **CΔ contain insert**– If element is composite element → all its children values are inserted in CΔ.
- **CΔcontain delete**– If parent element is deletedalong with children element→all deleted values with no children are inserted in CΔ.

- **CΔcontain modified** - If element is not composite element then its value can be used, but if it composite element then changes of each child elements are examined.

VI. RESULTS

Experimentation is performed on real-time databases collected from [33-36]. It is composition of homogeneous and heterogeneous XML documents.XML documents are preprocessed with SAX parser to extract structure and content. Table 1shows the details of the pre-processed XML documents from each dataset.

Table 1: Preprocessed Data sets

XML Document	Size (Kb)	Max-Depth	#Nodes	#attributes	#Complete paths
NASA	50	8	954	93	1428
University	100	5	3875	0	4536
SIGMOID	200	6	4468	1162	6082
DBLP	400	4	10393	1033	17716
E-Commerce1	800	3	15881	0	27785
E-Commerce2	1000	3	40962	0	27122

To assess the performance of compressed delta (CΔ) approach multiple versions of same XML document of size 50kB, 100kB, 200kB and 400kB are used from above datasets. Total 10 numbers of versions are considered of each XML document. Version are created using program developed in java.Performance of compressed delta approach is assessed with the consolidated delta proposed by Rusu et al. [20].From result charts shown in diagram 6 to diagram 8, it can be observed that how the CΔ performs better than consolidated delta for 10 consecutive versions of different sized XML documents after applying varying % of changes.

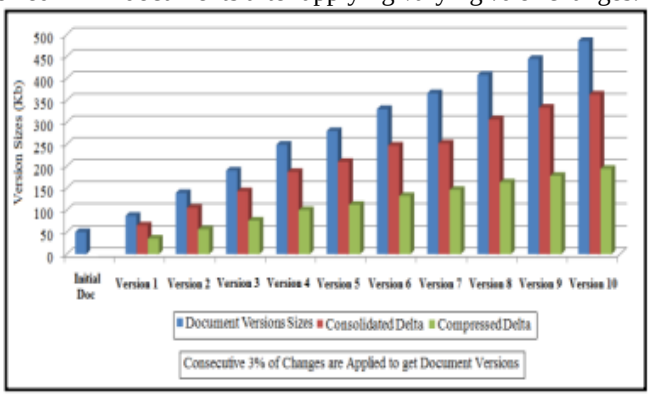

Figure 6: Result Chart 2

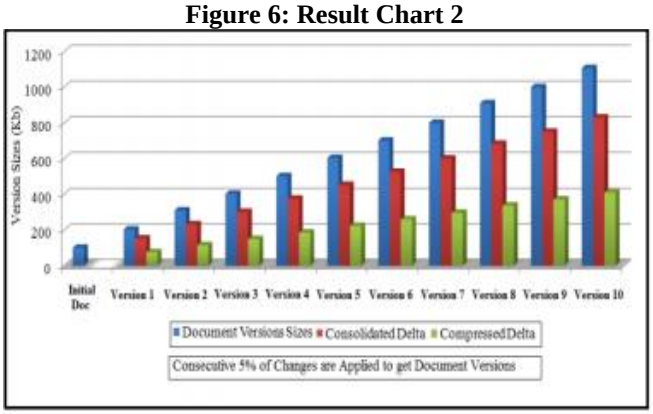

Figure 7: Result Chart 2

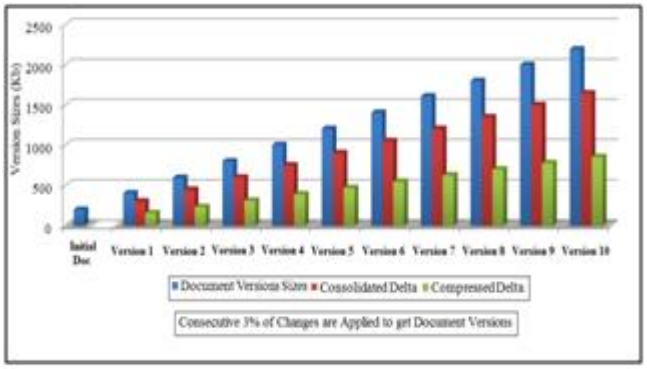

Figure 8: Result Chart 3

VII. CONCLUSION

XML acts as effective standard for storage and exchange of information. To obtain business values from XML documents effective management of these documents is really an important job. Techniques used to manage static XML documents are not useful to manage dynamic or multiversion XML documents. Moreover there is huge demand to manage those Multiversion documents properly due to their clever applicability in E-business. In response to this we have proposed time effcient compressed delta approach to obtain required document version with reduced operation cost. At the last run of an algorithm XML warehouseonly single XML document which contain sufficient information to get required document version. Experimental results for each unique real time datasets shows that proposed approach performs superior than prior consolidated delta approaches and retain only 25% data at the end in XML warehouse than size of all document versions.

REFERENCES

1. Sidra F., and Mansoor S., (2014) Temporal and multi-versioned XML documents: A survey, Information Processing and Management, Vol. 50, Issue 1, pp.113-131.
2. Rusu L.I., WennyRahayu., David Taniar., (2005) Intelligent Dynamic XML Documents Clustering, 22nd International Conference on Advanced Information Networking and Applications, pp. 449-456.
3. Dyreson C., and Grandi F., (2009) Temporal XML: Database Systems, pp. 3032-3035.
4. Pokorny J., (2002). XML data warehouse: Modelling and querying, Kluwer Academic Publishers, pp.67–80.
5. V.R.Sonawane, D.R.Rao., (2015) A Comparative Study: Change Detection and Querying Dynamic XML Documents, International Journal of Electrical and Computer, Vol.4 No. 5, pp.840-848.
6. Tran,T., Kutty,S., and Nayak, R., (2009) Utilizing the structure and content information for XML document clustering, Advances in Focused Retrieval, Lecture Notes in Computer Science, Springer Berlin / Heidelberg. Vol. 5631, pp.460–468.
7. Vijay Sonawane, D Rajeswaraa Rao, "An Optimistic Approach for Clustering Multi-version XML Documents Using Compressed Delta", International Journal of Electrical and Computer Engineering (IJECE), Vol. 5(6), pp. 1472-1479, 2015.
8. Al-Khalifa, S., Jagadish, H. V., Koudas, N., Patel, J. M., Srivastava, D., Wu, Y. "Structural joins: a primitive for efcient XML query patternmatching". Proceedings of the eighteenth international conference on data, engineering (pp. 141154), 2002.
9. Rusu, L. I., Rahayu,W., Taniar., "Storage techniques for multi-versioned XML documents", Proceedings of the thirteenth international conference on database systems for advance applications pp. 538545,2008.
10. Saccol, D. de B., Edelweiss, N., Galante, R. de M., Zaniolo, C., "XML version detection", Proceedings of the ACM symposium on document engineering pp. 7988, 2007.

11. Wang, Y., DeWitt, D. J., Cai, J. Y., "X-Diff: an effective change detection algorithms for XML documents", Proceedings of the nineteenth international conference on data, engineering pp. 519530, 2003.
12. Baeza-Yates, R., Ribeiro-Neto, B., "Modern information retrieval: The concepts and technology behind search" ACM Press/Addison-Wesley, 2011.
13. Flesca, S., Pugliese, A., "Fast detection of XML structural similarity", IEEE Transactions on Knowledge and Data Engineering,17(2), 160175,2005.
14. Gao,M., Chen, F., "Clustering XML Data Streams by Structure based on SlidingWindows and Exponential Histograms" Proceedings of the international conference on advances in databases, knowledge, and data applications pp. 224230, 2013.
15. Wan, X., Yang, J., "Using proportional transportation similarity with learned element semantics for XML document clustering", Proceedings of the fteenth international conference on world wide web pp. 961962, 2006.
16. Elham B. F., HasanK.,"Improving semantic clustering using with Ontology and rules", International Journal of Electrical and Computer Engineering (IJECE),Vol. 4, No. 1, pp. 7-15, Feb. 2014.
17. Pon, R. K., Crdenas, A. F., Buttler, D., Critchlow, T., "iScore: Measuring the interestingness of articles in a limited user environment" proceedings of the IEEE symposium on computational intelligence and data mining pp. 354361, 2007.
18. Wang, Y., Hodges, J., Tang, B., "Classication of web documents using a naive bayesian method" proceedings of the fteenth IEEE international conference on tools with, articial intelligence pp. 560564,2003.
19. Leonardi, E., Bhowmick, S. S., Madria, S., "Xandy: Detecting changes on large unordered XML documents using relational databases" Proceedings of the international conference on database systems for advanced applications pp. 711723 ,2005.
20. Rusu, L. I., Rahayu, W., Taniar, D., "Maintaining versions of dynamic XML documents" Proceedings of the sixth international conference on web information, systems engineering pp. 536543, 2005.
21. Wuwongse, V., Yoshikawa, M., Amagasa, T., "Temporal versioning of XML documents" Proceedings of the Seventh International conference on digital libraries: International collaboration and cross-fertilization pp. 419428, 2004.
22. Cavalieri, F. "EXup: an engine for the evolution of XML schemas and associated documents", Proceedings of the international conference on extending database technology pp. 110, 2010.
23. Cavalieri, F., Guerrini, G., Mesiti, M., Oliboni, B., "On the reduction of sequences of XML document and schema update operations" Proceedings of the IEEE twenty seventh international conference on data engineering workshops pp. 7786,2011.
24. Guerrini, G., Mesiti,M., "X-Evolution: A comprehensive approach for XML schema evolution" Proceedings of the international workshop on database and expert systems application pp. 251255, 2008.
25. Rosado, L. A.,Mrquez, A. P., Gil, J.M., "Managing branch versioning in versioned/temporal XML documents", Proceedings of the international symposium on XML, database pp. 107121, 2007.
26. Zholudev, V., Kohlhase, M., "TNTBase: A versioned storage for XML" Balisage: The Markup Conference, 2009.
27. Wei Li, XiongfeiLi,Regen Te., " Cluster Dynamic XML Documents based on Frequently Changing Structures " Advances in information Sciences and Service Sciences(AISS),Volume4, Number6., pp. 70-76 , April 2012.
28. A. Tagarelli.,S. Greco., "Semantic clustering of XML documents", ACM Transactions on Information Systems, vol. 28, no. 1, pp. 1-56, 2010.
29. C. Y. Wang, X. J. Yuan, H. Ning, et al., "Similarity Evaluation of XML Documents Based on Weighted Element Tree Model," in Advanced Data Mining and Applications, vol. 5678, R. Huang,et al., Eds., ed: Springer Berlin Heidelberg,pp. 680-687 , 2009.
30. T. Dalamagas, T. Cheng, K. J. Winkel, et al., "A methodology for clustering XML documents by structure",Information Systems, vol. 31, no. 3, pp. 187-228, 2006.
31. S. Flesca, G. Manco, E. Masciari, et al., "Fast detection of XML structural similarity", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 2, pp. 160-175, 2005.
32. Nayak, R., Xu, S., XCLS: A Fast and Effective Clustering Algorithm for Heterogeneous XML Documents, Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Singapore, LNCS 3918, 2006.
33. XML data repository, online at http:// www.cs.washington.edu / research /projects / xmltk / xmldata
34. http://panel.yepme.com/freefeed/ProductFeedhasoffer.xml
35. http://www.homeshop18.com/feeds/hs18products.xml
36. http://feeds.lenskart.com/overall_lk.xml

AUTHORS PROFILE

Dr. Vijay R. Sonawane Obtained his PhD in Computer Engineering in 2017. Currently he is working as Associate Professor in MVPS K.B.T.College of Engineering, Nashik. He published various papers in reputed journals and conferences. He is also working as reviewer to many journals. His is currently working on Data Visualization &Business Intelligence domain.

D. Rajeswara Rao he is a Professor and Head in department of CSE in V R Siddhartha Engineering College, Vijayawada.. He is He had published various research papers at National and International Journals/Conferences. His current area of research is Artificial Intelligence, Machine Learning, Data Mining and Warehousing, Data Base Systems, Soft Computing.