Analysing Duplicate Data Detection in Hierarchical Structure with Different Aspects

Sahunthala S, Udhaya Kumar A, Latha Parthiban

***Abstract:** In a real world, huge amounts of data are processed on the internet with specific application format. Duplicate data becomes a problem when data is being processed in large volumes; it creates degradation in the performance of processing the query in hierarchical structure. In this paper, we present a survey for analyzing the data duplicate detection in the hierarchical structure with different aspects such as attributes, objects, index of file structure and diversity of the structure based on the number of computation. If the computation is increased the performance is decreased when the query is processed .We also propose a technique Binary Similarity Duplicate Detection(BSDD) to improve the performance for processing the query with detection of duplicate data in a hierarchical structure with reduction of number of computation. It produces good result than the existing techniques.*

Index Terms: Attribute Diversity structure, Duplicate detection, Index structure, Objects.

I. INTRODUCTION

Nowadays small portion of relation data is carried over the World Wide Web. XML data plays a vital role in the World Wide Web. A huge amount of data is travelling through the internet. The data is delivered in the form of XML structure and it supports its own markup language for the development of the application and provides portability to carry the data. XML has the technique to transfer and publish the data on the web with respect to the business, application, etc. In real world, data cleanup and duplicate detection in the application is the vital task to transfer the data among the business. The XML document is symbolized by the structure of the tree.The duplicate detection is identifying more entities in the data. The main application of duplicate detection and removal is data cleanup and data incorporation. This paper supports the user who is involved in the data processing of hierarchical data and XML data. In real world application, huge amount of data will be stored. The role of duplicate data, how it is detected and how the performance is improved is given in [10]..

IN THIS paper, section 2 describes the literature survey of the various techniques present to detect the duplicate data in hierarchical data, section 3 describes the problems identified in existing method and proposed methods to improve the detection of duplicate data in hierarchical data and section 4 describes the conclusion of this paper

II. LITERATURE REVIEW

We focus on the problem of detection of data duplication in hierarchical structure with different aspects. This section describes various techniques present to detect the duplicate data in hierarchical data with the aspects of attributes, objects, structure and indexing to process the query in hierarchical data. The structured data such as record duplicate detection is surveyed thoroughly in [1],[11].

A. XML Dup System

Duplicate data detection may be in a single relation, tree and graph. XML duplicate detection is implemented based on join operation of objects. In this method, the similarity measure considers both attribute contents and the descendants of each node.This XML dup system [2] is constructed based on the Bayesian network prototype. Hence the distance between the two objects is calculated by overlay method. If U and V are two XML tree then the overlay between the two trees is found by the nodes in the tree if, and only if the node have the same path from the root node. In this method, the threshold value is assumed based on the structure. Threshold is the weightage to ensure the given two documents are similar or not. Hence the threshold specifies the number of edges in the path from the root node. If the distance is closer to the threshold value then the two objects in the structure are duplicates. This model improves the efficiency of precision and recall. Hence, the two nodes of XML structures are duplicates based on the following assumptions) If the XML node values are duplicates and XML node's children nodes are duplicates ii) If the two XML tree's root nodes are duplicates then the XML trees are duplicates. Figure 1 shows the workflow of the XMLdup technique.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author

Sahunthala S*, Assistant Professor in Department of Information Technology at Anand Institute of Higher Technology, India.

Dr.Udhaya Kumar A, Professor in Department of Master of Computer Applications at the Hindustan Institute of Technology and Science India.

Dr.Latha Parthiban, Professor in Department of Computer Science at Pondicherry University India.

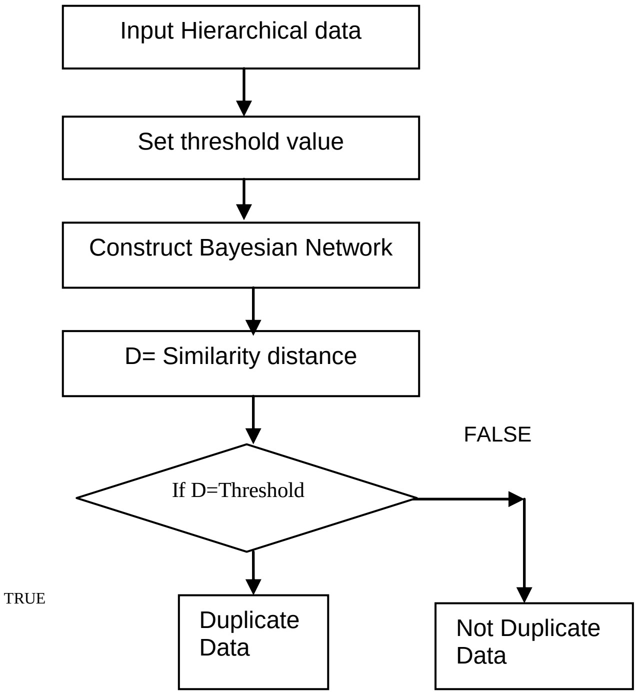

Fig 1: Workflow of XMLdup technique

The following probabilities are used to locate the duplicate nodes in two XML tree with the Bayesian network.

Prior prob: This value is assigned based on the similarity measures between two nodes. If the measure is impossible then default probability value is assigned to that node. That value is referred to as default constant

$$f_a = sim(U_i[x], U_i[x]) \quad (1)$$

where x is the attribute value of ith and j^th attribute in the structure. Hence *ned* (V_i, V_j) (normalized edit distance) is used in the similarity measure. If the similarity value is not measured then the default value f_a refer to (1) is assigned to the node.

Conditional prob: It defines the four types of probabilities. i) Cond1: Consider the XML node values are duplicates if attribute values are duplicates. If some attribute values are duplicates, it will be found based on the given probability value .ii) Cond2: If the probabilities of children nodes are duplicates then those nodes are duplicates in the structure. iii) Cond3: The nodes are duplicates based on both their values and the children nodes, find the conditional probability $P(A_i \mid A_1, A_2, ..., A_n)$ of the path from the root to the leaf node. iv) Final prob: This probability will find the duplicate data in hierarchical structure. The final probability F is given as

$$F = X - \frac{P(N1)+P(N2)+...+P(Nm)}{m} \quad (2)$$

Where X is the score value of the node, m is the non-leaf node value from the root node; N is the attribute value in the XML tree.

Limitations: Without machine learning approach the query is processed for a long duration due to the construction of Bayesian network.

B. Xml Join Technique

A pioneering approach suggests a join operation in XML database. Tree edit distance [2] is used in XML join algorithm. Comparison of more object representations is given by cosine method. It gives the high precision values in objects representation. Cosine method is used to find the similarity between two objects.

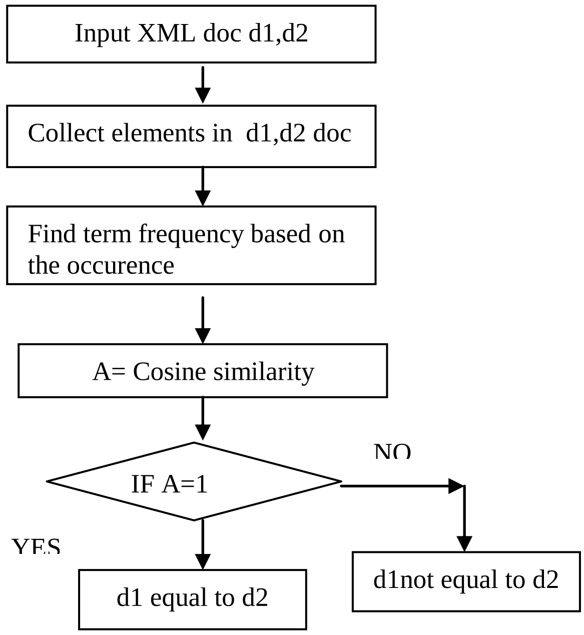

Fig 2 : Flow of XML join technique

Figure 2 shows how the Xml join operation is used to find the duplicate data in hierarchical structure. It achieves high precision. The comparison was done based on XML structure, textual content, and implicit semantics in the labels of XML structure. Cosine similarity is used to find the similarity between two data document in the data resource.It is denoted as

$$\text{Similarity (d1, d2)} = \sum X_i Y_j / \sum \sqrt{X_i^2}\sqrt{Y_j^2} \quad (3)$$

Where $(i = 1 \; to \; n, \; j = 1 \; to \; n)$ d1 and d2 are two XML documents, X and Y are XML tree structure, X_i and Y_j are the elements of X and Y respectively.

Limitations: This technique is used to compare the simple structure. When we try to process the complex structures the query is processed with slow manner.It degrades the performance.

C. Dogmatrix Framework

Dogmatrix Technology [3] uses keys to compare the objects in XML document. It compares the elements not only based on the direct values of the elements but also based on their parents, children and structures .Hence the duplicate objects are grouped together by using identification of the unique key. In this method the object is the group of elements to be compared for duplicate detection.

The unique key is generated for each object .Tool ToXGene is used to assign a unique ID to the data objects. Duplicate detection framework contains three parts. The first part is candidate definition.The second part is duplicate definition. The third part is duplicate detection. Candidate definition part gives the relevant objects to be compared. Duplicate definition identifies duplicate objects based on their description and a classifier between a pair of objects. Duplicate detection part specifies the approach to find the object identification.

This technique follows the following steps .i) First is Query formulation and execution of duplicate candidates from all possible candidates. ii) Second Description query formulation and execution and Generation of object description from the unstructured schema. iii) Third is Comparison Reduction for all duplicate pairs of object.

Limitations: When the heterogeneous structure is used it arises less performance in processing the query in the hierarchical structure.

D. SXNM (Sorted XML Neighborhood method)

This method [4] perform the comparison between objects in a hierarchical structure. Domain independent similarity measure is used to identify $sim(ob_i, ob_j)$ whether two objects are duplicates or not duplicates. The object description of the object is represented by OBDESC1, OBDESC2..., and OBDESCn. Hence the similarity between objects is obtained by aspects of comparable OBDESC tuples, similar OBDESC tuples, and Data relevance. The object is compared by ODTDIS as follows

$$(odt_i, odt_j) = \{1 \ \mathit{if} \ n_i \, \mathit{and} \, n_j \ \mathit{are\,not\,comparable}\} .$$

The above distance is derived from $obdes\,T_i = (v_i, n_i)\,\varepsilon\, ned\,(v_i, v_j)$.

This method contains two phases. One is key generation for each object another one is duplicate detection part. In this method, the following parts are considered to generate the keys. i) ***Key definition:*** It follows some rules to generate keys ii)***Key patterns:*** It contains parts of information to comprise the keys. The key generation contains three components of element identification, the key of each element and object description. In duplicate detection, the parameter contains object description and the relevance information. The relevance information are the usage of the XML element, threshold and descendants information: It contains the information if the descendants which is not used in duplicate detection. It finds the similarity measure between object description and the similarity of their descendants. If the keys are sorted properly then it produces a good result. If the keys are not sorted properly and contains error then it produces worst result in the duplicate detection.

Limitations: Not guaranteed that all objects are compared or not in the hierarchical structure. Automatically Candidate key is not generated to compare the objects.

E. Two Level Optimization Techniques

This technique [5] is used to detect the duplication of XML data which contains both object and attributes. Hence it provides the duplicate detection by reducing the comparison in a number of objects and attributes in XML data. The number of object comparison is reduced by a listwise optimization technique and the attribute comparison is reduced by pairwise optimization technique. Each pair of objects is compared using each object's attributes using a string similarity measure. Blocking strategy is used to avoid the computational complexity with the factor of threshold value which can be assigned by the user . This strategy uses the blocking key to compare the objects. Blocking key is assigned to each group . That group contains set of attributes or attributes segments. Each blocking strategy is combined with the object comparison.The attribute is selected automatically to avoid the user intervention with the database. Best parameters are selected automatically using blocking strategy.Optimization comparison is taken string edit distance to find the similarity between two strings. This technique uses a similarity measure of each object to find the duplicate detection. The similarity measure uses the method of the Jaccard coefficient or the cosine similarity. The pair wise optimization is based on the XMLDUP method. The problem of cosine method is discussed in section 2.2. It uses the probability method to find similar objects in the structure. It uses the Bayesian network to compare the objects. If the node contains the probability value as 1 then the nodes are duplicates. Otherwise, they are not duplicates. In bottom-up fashion, it finds the probability of each child nodes to find the duplicates. Two-level optimization technique produces a good result in terms of the parameters of recall and precision. This method improves the parameter of recall. It uses the unsupervised method to select the parameter.

Limitations: It takes more computation for comparison of objects. So the cost is high in computation of comparison of objects and degrades the performance.

F. Domain Independent Method

. In this technique [6] the elements are clustered using transitive closure approach. In this method, the elements are compared using string similarity method .The filter function is used to minimize the number of comparisons. The string similarity is found using string edit distance.XML structure contains collection of elements. We need to integrate more than one structure when it is reliable on the internet. If the structure is integrated it contains a huge amount of data. In that data duplicate elements are possible. It raises two issues. First, data occupies more memory space. Second, it takes more processing time. It concentrates the process in the XML document as element scope and structural diversity. This method is used to find the duplicate elements in the single XML document. Learning algorithms are used to improve the quality of similarity measure. We need to consider the structure and data to find the duplicates in the hierarchical document. Let U and U^1 are two nodes form document 1 and document2 respectively. The elements are duplicated by one of the following definitions i) If the parent elements of U and U^1 are equal and similar ii) U and U^1 have the same name iii) U and U^1 children nodes are having the same structure or contains the same data iv) Hence the string similarity measure is based on the Inverse document frequency (IDF)

$$IDF = \log \frac{N}{t} \qquad (4)$$

Where N is the number of XML documents, t is the number of occurrence of the element in all documents. Two string similarity measures S1 and S2 is given by

$$sim(S1, S2) = \frac{IDF(S1 \cap S2)}{IDF(S1 \cup S2) / IDF(S1 \cap S2)} \qquad (5)$$

Filtering method is used to find the comparison of elements effectively with reduction in the number of comparisons. Some Filtering methods are

Length Distance Filter

Let st_1 and st_2 are two strings. If those two strings are compared with respect to the length of the string $len(st_1)$, $len(st_2)$ it is true when the following inequality holds.

$$|len(st_1) - len(st_2)| \leq dt_e(st_1, st_2) \;|$$

Filtering using Triangle Inequality

In that method the comparison is based on the three strings. Let A,B,C be the three strings from the data to be compared.

$$dt_e(A,B) + dt_e(B,C) \geq dt_e(A,C) \geq |dt_e(A,B) -$$

Bag Distance Filter

Bag distance filter consider three string for duplicate data comparison, so that the similar data is grouped easily. It may reduce the number of comparison in the duplicate detection. This is based on the semantics of the string.

$D_{bg}(A,B) = \max(|A - B|, |B - C|)$ Where A, B and C are three strings.

Limitations: Difficulty arises in the process of integration of more than one document and utilization of memory is high.

G. Two Tier Index Structure

This method [7],[8] is used in wireless data transfer. This method is used to deliver the XML document based on the user request efficiently. The main strategy is to develop the index integration for all XML documents. The user can submit the queries using Xpath .First; the index can be placed when the query is broadcast. Then the corresponding XML document is retrieved from that index from the structure. The client hasn't any knowledge about the document so it continuously monitors the channel. In two tier index methods the index size is reduced and the access time of the document is also reduced. The index contains the nodes of the root node, leaf node and internal node. The node contains three blocks. The type of the index structure is given by the first block. The child information <entryin,ptrin> is carried by the second block. The document information <document,ptrin> is located in the third block. In the first tier, the collection of document information is there to access the document which has been requested by the client. The second tier index contains the resultant document.

The tuning time is calculated by TT=L_i+n.L_o+time to retrieve the required documents (n number of cycles to send the required documents). Hence, the query is passed through the XML document to retrieve the element. If the same Xpath is located in more document , the way of retrieving the query is reduced by using the index structure.If the index is same the query is processed in any one of the similar documents so that the query is processed very effective manner. In index structure the document part is compared to verify whether the documents are similar or not. The index is created based on the number of nodes in all the structure. If the node is present in more than one document, then the same index is used to utilize that node. Hence we need to find the nodes are same or not. The normalized edit distance is used to find the similarity between two nodes. The client sends the query to the channel. The server listens whether any query is requested by the client. If it is found then it searches the resultant of the query through the two-tier index structure. It reduces the time to retrieve the document. Hence the pruned compact index is used in processing.

Limitations: It occupies more memory space to compare the objects due to the creation of the document entry with the node.

H. Xedge Index algorithm

In Xedge Index technique the hierarchical structure similarity is measured by the level of the structure to be compared. In each level the element is identified with the unique number. In this method the unique and different terms find in each level of the hierarchical structure.The string edit distance method is used to find whether the terms are similar or not. This technique is described in detailed in [12].

Limitations: If level of the structure is increased then performance is less for processing the query.

III. PROBLEMS ANALYSED IN EXISTING TECHNIQUE AND PROPOSED TECHNIQUE

We present the proposed technique to detect the duplicates data in hierarchical structure with the aspects of attributes, objects, index structure with the different similarity measure. In this section we describe the problems in the existing techniques and proposed methods to improve the efficiency of finding duplicates data in the hierarchical data. In XML dup system the problem is not able to find all possible duplicates data when the structure is complex. When the query is processed it takes longer time to process the query from the hierarchical data. If threshold value is low we will not get the high precision and recall in the query processing even though if we are using a low pruning factor. We can propose the machine learning approach to construct the Bayesian network. If machine learning approach is used, the duplicate detection process is faster than the base Bayesian network. The Cosine technique gives less performance to find the similarity between two objects. Because it takes all elements from the set of document for comparison. If the document size is very large then the calculation of comparison time is also increased. Then automatically the performance of the query accessing process will be decreased. The proposed idea of the cosine method is enhancing the parameter of N gram value. This value quantifies the elements in the documents with Ngram into windowing size. Hence each window occupies the elements with Ngram size. In each comparison, if the candidate key is generated for comparison between objects in the hierarchical data, then the key is also generated even though if the same set of pair is to be compared.

This will not produce better result in heterogeneous structure. If the candidate key is found automatically then the number of comparison is reduced. Determining the candidates automatically is not explored in dog matrix framework.

Two difficulties are attained to use the methodology of Sorted XML Neighborhood method **t**o find out the duplicate detection in hierarchical data. One problem is finding similarity measure between objects in all group of objects and another problem is whether all objects are compared or not.

Hence the key is generated automatically by the tool ToXGene.If we can use efficient tool than the above tool then we can able to get good result for comparison of objects.

In the two level optimization technique there are two problems. First to decide the threshold value. Second using the similarity measure method of the Jaccard coefficient and cosine method, it arises high cost because of the complexity in computation of comparison.

It takes more time to find the duplicate detection in objects. String similarity measure's cost is expensive. A Bottleneck occurs when the comparison is in use. Some duplicate objects are missed for comparison.The user needs deep knowledge of the data to select the parameters for the blocking strategy. If we can try with E-dice method for similarity measure we can able to get good performance. The domain independent method takes more memory space when the string edit distance between two objects is found. Another problem of finding string edit distance is suitable for small sizes of hierarchical data comparision.We are not able to integrate all documents comparison. To integrate all documents using any distribution techniques such as naive, partial etc are used.

In two tier index structure the time to search the query is higher in the hierarchical structure. The structure occupies more space when it is integrated for hierarchical document. In this approach finding the similarity of documents are not explored. If it happens the process time is reduced. In all techniques the performance is compared with the parameters of Precision, recall. Xedge algorithm [11] uses the index level to compare the nodes on both the structure. The structure with the homogeneous can be used in this method. Hence the similarity parameter cosine or Jaccord distance is not suitable for heterogeneous structure. It will be produced by using the index as the Bayesian level. In the proposed method similar documents are grouped based on the value of the algorithm.

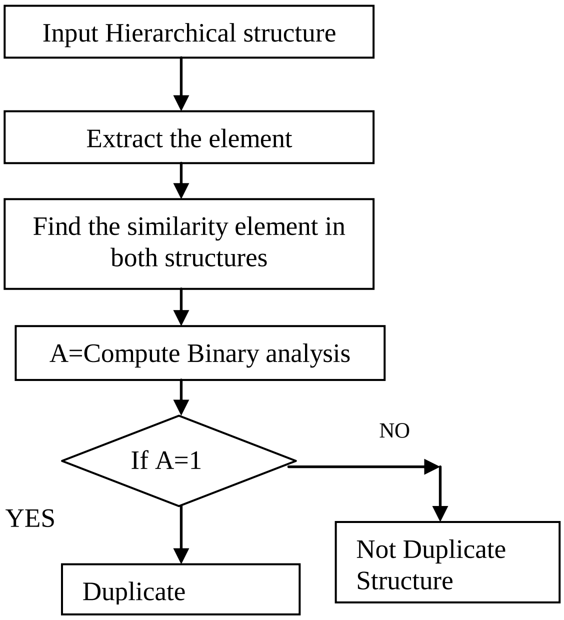
Input Hierarchical structure
Extract the element
Find the similarity element in both structures
A=Compute Binary analysis
If A=1
NO
YES
Duplicate
Not Duplicate Structure

Fig 3 : Proposed system model

Table 1 shows the Issues in various existing techniques and the method to be proposed to find the duplicate data in hierarchical structure to overcome the issues.

We present the proposed method to reduce the time to process the data in the hierarchical structure to analyze the duplicate data. Hence the method Binary Similarity Duplicate Detection is used to find sample documents

```
<book>
<title></title>
<author>
            <Firstname></Firstname>
            <Lastname></Lastname>
</author>
<publication></publication>
<year></year>
<ISBN>/ISBN>
```

Fig 4: Dcument1

```
                    <book>
                <title></title>
                   <author>
          <Firstname></Firstname>
           <Lastname></Lastname>
                   </author>
      <Publication></publication>
                <Year></year>
                <ISBN>/ISBN>
```

Fig 5: Dcument2

The proposed method work as follows the algorithm of Binary similarity Duplicate Detection is shown in Figure 6.

```
Algorithm: Binary Similarity Duplicate Detection
Input: XML documents
Variables: m is the no of documents, n is the number of
elements in mth document and p is the number of
elements in the m+1th document
Output: Binary value
Steps:
    1.  Give input as XML documents
    2.  Extract the element in all documents
    3.  Find the number of elements
    4.  Loop k=1 to m
    5.  Loop i=1 to n
    6.  Loop j= 1 to p
    7.  A=Perform similarity(i,j)
    8.  If A=0 , break
    9.  End loop, End loop, End loop
    10. If (A=1)
    11. Conclude similar structure
    12. Else
    13. Conclude not similar structure
```

Fig 6. Binary Similarity Duplicate Detection Algorithm

We analyze the different similarity measure approach with the following documents with respect to the number of computation. The computation may be the basic operation such as addition, multiplication, division etc. If the number of computation is reduced, it increases the speed of the process in the data structure.

In Figure 4, the elements are extracted as book,title,author(Firstname,Lastname).publication, year and ISBN.In Figure 5, the elements are extracted as in Figure 4.Then documnt1 (Figure 4) element in the each level is compared with the element which can be positioned on the document 2(Figure 5). If the result is same the binary calculation is performed. At the end of the comparison if the result is not affected we conclude both documents are similar.

S.No	Existing Method	Performance Issues	No of computation
1	Xmldup	Query is processed by long time.	n-1
2	Xml join Technique	Used to compare the simple structure.	3n+1
3	Dogmatix	Less performance in the heterogeneous structure.	n-1(N is the number of object description)
4	SXNM	Not guaranteed whether all objects are compared or not. Automatically Candidate key is not generated to compare the objects.	n-1
5	Two level optimization method	Cost is high for computation in comparison of objects.	2n-1
6	Domain Independent method	Problem in integration of more than one document and utilization of memory is high.	3n-1
7	Two tier index structure	Occupies more memory space when comparing the objects.	n-1
8	Xedge Index algorithm	If level of the structure is increased then less performance in processing the query.	4n+3
9	BSDD method	Element extraction may be complicated when huge amount of data is used.	1

Table 1 Issues in the various techniques to find the duplicate data in the hierarchical structure

Table 2 shows the number of computation with our sample structure given in Figure 4 and Figure 5.The method Binary Similarity Duplicate Detection is better than the existing techniques which can be discussed in section 2.

IV. CONCLUSIONS

In this paper, the various hierarchical data duplicate detection methods are analyzed with different aspects. The survey has analyzed some aspects such as attributes, objects, index of file structure, diversity structure. The data duplicate detection is identified in hierarchical data by various technologies. Duplicate detection is found via the attribute in the XML document by the methods of domain independent system and XMLdup system. The object duplicates are detected by Sorted neighborhood method and two level optimization technique. The file index size is pruned by using the method of two-tier index structure. Hence the survey is analyzed based on the number of computation to find the duplicate data in hierarchical structure. The proposed Binary Similarity Duplicate Detection reduces the number of computation in finding the similarity data in hierarchical structure.

REFERENCES

1. Ahmed K. Elmagarmid, Senior Member, IEEE,Panagiotis G. Ipeirotis, Member, IEEE Computer Society, andVassilios S. Verykios, Member, IEEE Computer Society, Duplicate Record Detection: A Survey, IEEE Transactions on Knowledge and Data Engineering.VOL. 19, NO. 1, JANUARY 2007
2. Luı´s Leita˜ o, Pa´ vel Calado, and Melanie Herschel, Efficient and Effective Duplicate Detection In Hierarchical Data. IEEE Transactions on Knowledge and Data Engineering. VOL. 25, NO. 5, MAY 2013 (1028-1041)
3. Melanie Weis , Felix Naumann "DogmatiX Tracks down Duplicates in XML "2005 ACM 1595930604/05/06.
4. Sven Puhlmann, Melanie Weis, and Felix Naumann, XML Duplicate Detection Using Sorted Neighborhoods, Y. Ioannidis et al. (Eds.): EDBT 2006, LNCS 3896, pp. 773–791, 2006._c Springer-Verlag Berlin Heidelberg 2006
5. Lus Leit~ao, P_avel Calado , Efficient XML Duplicate Detection Using an Adaptive Two-level Optimization**,** 2012 ACM 978-1-4503-1656-9.
6. D.V. Kalashnikov and S. Mehrotra, "Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph." ACM Trans. Database Systems, vol. 31, no. 2, pp. 716-767, 2006.
7. Weiwei Sun, Yongrui Qin, Jingjing Wu, Baihua Zheng, Zhuoyao Zhang,Ping Yu, Peng Liu, and Jian Zhang, Air Indexing for On-Demand XML Data Broadcast IEEE transactions on parallel and distributed systems, vol. 25, no. 6, june 2014(1371-1381)
8. Weiwei Sun#, Ping Yu#, Yongrui Qing#, Zhuoyao Zhang#, Baihua Zheng*, Two-Tier Air Indexing for On-Demand XML Data Broadcast**,** 2009 29th IEEE International Conference on Distributed Computing Systems.
9. Melanie Weis, Felix Naumann "Detecting Duplicate Objects in XML Documents" IQIS 2004 Maison de la Chimie, Paris, France _c 2004 ACM 1-58113-902-0/04/0006.
10. F. Naumann and M. Herschel, An Introduction to Duplicate Detection. Morgan and Claypool, 2010.
11. Nermeen Gamal Rezk, Amany Sarhan, and Alsayed Algergawy "Clustering of XML Documents Based on Structure and Aggregated Content", IEEE,2016, 978-1-5090-3267
12. Panagiotis Antonellis Christos Makris Nikos Tsirakis "XEdge: Clustering Homogeneous and Heterogeneous XML Documents Using Edge Summaries", VLDB,2017.

S.No	Technique	No of computation
1	Xmldup	7
2	Xml join Technique	25
3	Dogmatrix	7
4	SXNM	7
5	Two level optimization method	15
6	Domain Independent method	23
7	Two tier index structure	4
8	Xedge Index algorithm	35
9	BSDD method	1

AUTHORS PROFILE

Sahunthala S is currently working as Assistant Professor in Department of Information Technology at Anand Institute of Higher Technology, India.She completed B.E (Computer Science) in Bharathidhasan University.She in M.E in Arulmigu Kalasalingam College of Engineering and Technology in the discipline Network Engineering. She is interested in the research area of Data Mining, Big Data.

Dr.Udhaya Kumar A is currently Working as **a** Professor in Department of Master of Computer Applications at the Hindustan Institute of Technology and Science India. He completed his Ph.D in the area of Stochastic Optimization.He published more than 70 articles in the reputated journals and conferences.

Dr.Latha Parthiban is currently working as a Professor in Department of Computer Science at Pondicherry University India. She had completed her Ph.D in the area of Data Mining. She has 25 years of teaching experience in Engineering colleges which includes 3 years as professor in SSN college of Engineering. She is presently with Pondicherry University CC. She has 148 peer reviewed journal papers and 24 conference publications. Her research area are Data Mining and Image Processing.