# PXML-MINER: A PROJECTION-BASED INTERESTING XML RULE MINING TECHNIQUE

***D Sasikala*** [*] ***and K Premalatha***

*Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Erode, Tamil Nadu, India*
**E-mail:* sasikala0979@gmail.com

## *ABSTRACT*

*In recent times, the mining of association rules from XML databases has received attention because of its wide applicability and flexibility. Many mining methods have been proposed. Because of the inherent flexibility of the structures and the semantics of the documents, however, these methods are challenging to use. In order to accomplish the mining, an XML document must first be converted into a relational dataset, and an index table with node encoding is created to extract transactions and interesting items. In this paper, we propose a new method to mine association rules from XML documents using a new type of node encoding scheme that employs a Unique Identifier (UID) to extract the important items. The node scheme modified with UID encoding speeds up the mining process. A significance measure is used to identify the important rules found in the XML database. Finally, the mining procedure calculates the confidence that the identified rules are indeed meaningful. Experiments are conducted using XML databases available in the XML data repository. The results illustrate that the proposed method is efficient in terms of computation time and memory usage.*

**Keywords:** XML mining, XML documents, Semantics, Node encoding, Rule mining

## 1 INTRODUCTION

XML is widely used as the *de facto* standard for data exchange on the Internet. As more and more data are stored and represented in XML format, research efforts to mine these data are increasing (Zhao et al., 2005). With the continuous growth of XML data sources, the ability to extract knowledge from them for decision support becomes increasingly important and desirable (Nayak, 2005; Mohammadzadeh et al., 2006). For example, detecting structural and content affinities among XML data can help conceive techniques for indexing such data, thus narrowing the search space and improving the design of query plans (Tagarelli & Greco, 2006). XML data exist in two forms: XML documents and XML schemas. An *XML schema* defines structure while an *XML document* defines data (Abiteboul et al., 1999). XML documents are instances of an XML schema that includes acceptable elements, attributes, the number of element occurrences, and other constraints. There are many languages to describe the structure and content of these documents, such as Document Type Definition(DTD) and XML Schema Definition (XSD); According to Lee & Chu (2000) and Nayak (2008), XSD has more capabilities than DTD. An XML document includes tags and data, with tags describing names of elements contain concepts such as text data. In addition to document tags, structure tags also show the relationships among elements (Nayak & Richi 2008; Alishahi et al., 2010).

In recent years, the Internet has become a repository for huge amounts of data, and the quantity of XML data shared over the World Wide Web has increased drastically. A large majority of the XML data is data-centric, but text-centric XML document collections are now becoming more and more frequent. As a consequence, it has become necessary to provide a means to manage these collections. This can be done by automatically organizing very large collections into smaller sub-collections using document-clustering techniques. Unfortunately most of the research on structured document processing is still focused on data-centric XML (Guillaume & Murtaugh, 2000; Yi & Sundaresan, 2000; Doucet & Ahonen-Myka, 2002). The processing and management of XML documents have already become popular research issues (Abiteboul et al., 2000), with the main problem in this area being the need to optimally index these documents for storage and retrieval purposes.

There have been many search techniques developed that essentially rely on a set of weighted keywords in the search query to determine the proximity of the query and a document in the feature space. Searching XML documents, however, departs from the conventional information retrieval strategy in the sense that XML documents have nested XML elements and semantics of data values indicated by tags. As a result, the notion of keyword proximity used in the research is too simple to be effective in XML searches. Many algorithms have been proposed for providing an effective mining algorithm for XML databases. In this paper, a projection -

based mining algorithm is proposed for XML data mining that effectively mines the rules in less time than other methods with the use of a very small amount of memory.

Projection-based mining is characterized by finding the most precise rules that are relevant to the problem scenario. Projection uses less time for the rule mining process and less memory usage. The algorithm works by calculating different length values, i.e., the intra-relationship between each object in an XML document. The UID-based system is used for parsing the XML document, which is also the foundation for the proposed approach for XML data mining. We used this projection-based mining on representative datasets taken from the XML data repository. The performances have been evaluated, and we found that the proposed projection-based mining improves the quality of XML data mining.

The main contributions of the proposed projection based mining method are:

- Effective and small index table characterized by UIDs,
- XML parsing based on UIDs, and
- Projection based mining.

The rest of the paper is organized as follows. Section 2 reviews related works about XML data mining. Section 3 defines the supporting algorithm for the proposed approach. Section 4 discusses details of the proposed approach with mathematical models. Section 5 lists the advantages of our proposed approach, and Section 6 gives the results of our experiments and a discussion of the proposed approach.

## 2 REVIEW OF RELATED WORKS

A sample of the research available in XML document mining literature concerning the efficient storing, indexing, and searching of XML data follows.

Porkodi et al. (2009) have presented an improved framework for mining association rules from XML data using XQUERY and NET based implementations of the *a priori* algorithm. Shahriar and Liu (2011) improved the association rule mining technique with semantic constraints in XML. These semantic constraints were expressed through the use of proximity properties of items in an XML document that conformed to a schema definition. The proposed association rule mining with semantic constraints was used for mining both content and structures in XML.

Li et al. (2007) gave a definition of transaction and item in the XML context and then built a transactional database based on an index table. Based on those definitions and the index table, the relationship between a transaction and an item could be easily identified. A highly adaptive mining technique was also described. This technique retrieves frequent item sets based on a user-given tag. It also generates all frequent item sets and rules. The effectiveness of this technique was proven on real-life data. Feng and Dillon (2005) presented a template model to help users specify interesting XML-enabled associations to be mined. Techniques for template-guided mining of association rules from large XML datasets were also described. A set of experiments on both synthetic and real-life data demonstrated the effectiveness of this technique.

Kliegr et al. (2010) proposed the General Unary Hypotheses Automaton (GUHA) Association Rules (AR) Model, an XML schema-based formalism for representing the set up and results of AR mining tasks. In contrast to the item-based representation of the Predictive Model Markup Language (PMML) 4.0 Association Model, this model expresses the association rules as a couple of general Boolean attributes related by a condition based on one or more arbitrary interest measures. This makes the GUHA AR Model suitable for AR mining algorithms other than *a priori*, such as those mining negative ARs. In addition, they presented important research results on special logical calculi formulas that correspond to such association rules. The GUHA AR Model was intended as a replacement for the PMML Association Model. It is tightly linked to the Background Knowledge Exchange Format (BKEF), an XML schema proposed for representation of data-mining related domain knowledge and to the AR Data Mining Ontology ARON.

The increasing number of very large XML datasets available to casual users is a challenging problem for the data community, calling for appropriate support to efficiently gather knowledge from XML data. Data mining is a widely applied tool used to extract frequent correlations of values from both structured and semi-structured datasets. This is the appropriate field for knowledge elicitation. Mirjana Mazuran et al. (2009) described an approach to extract tree-based association rules from XML documents. Such rules provide approximate,

intentional information on both the structure and the content of XML documents and can be stored in XML format to be queried later on. Their prototype system demonstrated the effectiveness of their approach.

Shaharanee et al. (2010) proposed a strategy that combines data mining and statistical measurement techniques to discard non-significant patterns. They considered the "prions" database, which describes the protein instances stored for the Human Prions Protein. A unified framework was applied to this dataset to demonstrate its effectiveness in assessing the interestingness of discovered XML patterns by statistical means. When the dataset was classified/predicted, the approach discarded the non-significant XML patterns without the cost of a reduction in the accuracy of the pattern set as a whole.

# 3 ASSOCIATION RULE MINING

Association rule mining is a process of discovering interesting relations among attributes in large databases. One use of association rule mining is to find interesting and useful patterns in a transaction database. This type of database contains transactions that consist of a set of items and a transaction identifier. The problem of association rule mining is defined as follows. Let $I = \{i_1, i_2, i_3 \ldots i_n\}$ be a set of n binary attributes called *items*. Let $D = \{t_1, t_2, t_3 \ldots t_m\}$ be a set of transactions called the *database*. Each transaction in D has a unique transaction ID and contains a subset of the items in I.

Association rules are implications of the form $X \rightarrow Y$ where $X, Y \subseteq I$, and $X \cap Y = \phi$ (X and Y are two disjoint subsets of all available items). X is called the antecedent, and Y is called the consequent. Association rules have to satisfy constraints on measures of significance and interestingness. Association rule generation is usually split into two separate steps. The first step needs more attention while the second step is straight forward.

1. Minimum support (see below) is applied to find all *frequent itemsets* in a database.
2. These frequent itemsets and the minimum confidence constraint are used to form rules.

## 3.1 Measures of association rule mining

The strength of an association rule is quantified by a significance measure. The following are the basic measures of association rule mining.

**Support and Confidence**
The rule has support s in D if s% of the transactions in D contain both X and Y. The interestingness issue refers to finding rules that are interesting and useful to users. A rule has confidence c if c% of the transactions in D that contain X also contain Y. A rule is said to hold on a dataset D if the confidence of the rule is greater than a user-specified threshold. This can be represented as

$$\text{Support } (X \rightarrow Y) = P(X \cup Y) \quad (1)$$

$$\text{Confidence } (X \rightarrow Y) = P(Y/X) \quad (2)$$

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time.

# 4 PROPOSED APPROACH FOR MINING BASED ON THE PROJECTION-BASED METHOD

Mining of association rules from an XML database has received significant attention due to its wide applicability. Our proposed method attempts to mine the data from an XML database using a projection-based mining algorithm. This method is proposed because there are many drawbacks to the existing approaches. For example, the node encoding scheme for an XML document is complicated by including the immediate predecessor of each node in the successor node. The node encoding schemes used are based on numbering the tags, which affects the differentiation of the same nodes present in different levels of different documents. Another problem is the long period of time needed for full execution of the process. Our proposed approach is detailed in the following section.

## 4.1 Tag encoding using a Unique Identifier (UID)

The first step of the proposed approach is to encode the XML document. The encoding method here assigns a unique identifier (UID) for each node. Every closed node with a distinct value in the XML document is assigned an UID. Consider the following example.

**Example 1**. XML document 1

<*book*>
**UID:1** <*thriller*> *Sherlock Homes* </*thriller*>
**UID:2** <*drama*>*Tempest* </*drama*>
</*book*>

In the above example, tags <thriller> and <drama> possess values *Sherlock Homes* and *Tempest* and an UID is assigned to them. If we have additional documents in the database with the same tag, every occurrence of the tags are assigned the same UID as shown in Example 2.

**Example 2.** XML document 1

<*book*>
**UID:1** <*thriller*> *Sherlock Homes* </*thriller*>
**UID:2** <*drama*>*Tempest* </*drama*>
</*book*>

XML document 2

<*book*>
**UID:1** <*thriller*> *Sherlock Homes* </*thriller*>
**UID:3** <*thriller*>*star wars*</*thriller*>
**UID:2** <*drama*>*Tempest* </*drama*>
</*book*>

The above example illustrates how the UIDs are assigned in the proposed approach. The next step of the process is creation of the index table according to the UID occurences.

## 4.2 Index table creation

The index table is a compilation of the XML tags encoded in the XML documents. The proposed approach makes use of the UID provided for each tag for the construction of the index table. Because each UID specifies a unique tag, each UID is present only once. As in the previous approaches, for the same value there will not be two indexes. Therefore, the UIDs are not duplicated.

$$index\ table = (DocID, UID) \tag{3}$$

The two fields of the index table are *DocID* and *UID*. *DocID* represents an identifier for each XML document in the dataset whereas *UID* represents the unique identifier values for the XML tags present in each document. The advantage of the index table is that it only stores assigned tags and not the string defined by the node. Thus less memory is used in comparison to other index tables, such as those defined in Li et al. (2007).

XML Document 1

<*pc*>
**UID:1** <*utility*>*hard disk*</ *utility*>
**UID:2** <*hardware*>*ram*</ *hardware* >
**UID:3** <*software*> *MS office* <*software*>
</*pc*>

XML Document 2

<*pc*>
**UID:2** <*hardware*>*ram*</ *hardware* >
**UID:4** <*RAM*>*1GB*<*RAM*>
**UID:3** <*software*> *MS office* <*software*>
</*pc*>

XML Document 3

<*pc*>
**UID:4** <*RAM*>*1GB*<*RAM*>
</*pc*>

We give an example (Table 1) of how an index table is constructed using the following three XML dummy documents. The index table generation is the main processing step in the mining process. The detailed process of mining is explained in the following section.

**Table 1.** Index table

| DocID | UID |
|---|---|
| **1** | UID:1<br>UID:2<br>UID:3 |
| **2** | UID:2<br>UID:3<br>UID:4 |
| **3** | UID:4 |

## 4.3 Projection-based mining method

The proposed method uses a projection-based approach for mining the data, based on the support value of each UID (i.e., a UID above the support value threshold) in the index table.

***Step 1***. Calculate the support values of all unique elements in the index table based on the occurrence of each number of the UID value in each document (see equation (1)). From Table 1, we get:

**Table 2.** Support values of all unique elements

| UID | Support |
|---|---|
| **UID:1** | 0.33 |
| **UID:2** | 0.66 |
| **UID:3** | 0.66 |
| **UID:4** | 0.66 |

***Step 2***. Set a threshold for the minimum support measure in order to filter out the lowest frequency UIDs. In our example, we set the threshold as 0.5. Thus the selected values from Table 2 can be listed as:

**Table 3**. Support values above threshold

| UID | Support |
|---|---|
| **UID:2** | 0.66 |
| **UID:3** | 0.66 |
| **UID:4** | 0.66 |

***Step 3***. Apply the projection-based method to process the data with frequent UIDs. Every UID is analyzed to obtain the one-length pattern, i. e., how often a frequent UID occurs in a document with every other frequent UID.

**Table 4.** Projection for one-length patterns

| UID | DocID | Projection |
|---|---|---|
| **UID:2** | 1 | UID:3 |
| | 2 | UID:3 |
| | | UID:4 |
| | 3 | - |
| **UID: 3** | 1 | UID:2 |
| | 2 | UID:2 |
| | | UID:4 |
| | 3 | - |
| **UID:4** | 1 | - |
| | 2 | UID:2 |
| | | UID:3 |
| | 3 | - |

Table 4 gives the projection values of the most frequent UIDs after removal of duplicates. We list the one-length pattern in Table 5.

**Table 5**. Support values for all one length patterns.

| One-length patterns | Support |
|---|---|
| **UID:2→ UID:3** | 0.66 |
| **UID:2→ UID:4** | 0.66 |
| **UID:3→ UID:4** | 0.33 |

Again select a threshold to identify the most frequent one-length patterns, which we set as 0.5.

***Step 4***. Apply the projection-based mining method again to find the two-length patterns.

**Table 6.** Projection values of two-length patterns

| UIDs | DocID | Projection |
|---|---|---|
| **UID:2→ UID:3** | 1 | - |
| | 2 | UID:4 |
| | 3 | - |
| **UID: 2→ UID:4** | 1 | - |
| | 2 | - |
| | 3 | - |

Thus we get the second length as:

**Table 7.** Two length patterns

| Two length patterns | Support |
|---|---|
| **UID:2→ UID:3→UID:4** | 0.33 |

***Step 5*.** Continue to apply the projection-based method again to get the next length pattern until no additional length patterns are found, as shown in Table 8.

**Table 8.** Projection values of three-length patterns

| UIDs | DocID | Projection |
|---|---|---|
| **UID:2→ UID:3→UID:4** | 1 | - |
| | 2 | - |
| | 3 | - |

As no other UIDs are related to identify the next pattern, the algorithm is stopped.

***Step 6.*** If no more patterns are discovered, the projection-based methods ends, and the mined patterns are listed.

**Table 9**. Mined UIDs

| One Length | Two Length | Three Length |
|---|---|---|
| **UID:2** | **UID:2→ UID:3** | **UID:2→ UID:3→UID:4** |
| **UID:3** | **UID:2→ UID:4** | |
| **UID:4** | **UID:3→ UID:4** | |

## 4.4 Confidence

Confidence can be interpreted as an estimate of the probability X. The confidence of the method is calculated using the following equation:

$$confidence = \frac{Support(X \cup Y)}{Support(X)} \tag{4}$$

For the generated two length and three length patterns the confidence values are calculated. The rules are generated based on the user given threshold. i.e., the rules above the given confidence have been chosen.

**Table 10.** Confidence

| Mined Rules | Confidence |
|---|---|
| **UID:2→ UID:3** | **1** |
| **UID:2→ UID:4** | **1** |
| **UID:3→ UID:4** | **0.5** |
| **UID:2→ UID:3→UID:4** | **0.5** |

The example above uses the projection-based mining algorithm to mine XML documents. This method provides a unique ID for an item in particular rather than having it as a number representation of all tags. Consider the first rule from Table 10, UID:2→ UID:3. The confidence of this rule can be calculated from Table 9. Initially we have to calculate the support of UID:2 and UID:3 together in Table 9. Then we calculate the support of UID:2 alone. The support of UID:2 and UID:3 together is 1, and their frequency is 1. The support of UID:2 is also 1 as it exists only once in the table. Thus,

$$confidence(UID:2 \rightarrow UID:3) = \frac{Support(UID:2 \cup UID:3)}{Support(UID:2)} = \frac{1}{1} = 1$$

This method helps in improving the process of quickly identifying frequent patterns. It clearly explores the maximum length of different UIDs, which helps to identify the relationship between the objects. Thus, the system is more efficient in making decisions on association of objects.

# 5 RESULTS AND DISCUSSION

In this section we present results from using our proposed projection-based rule mining technique as applied to three separate datasets. The algorithms are implemented using the Java language with JDK 1.6 on a system with an Intel core i5 processor with 4GB RAM and a 500GB hard disk.

## 5.1 Dataset description

The test datasets were extracted from an XML data repository at the University of Washington Department of Computer Science and Engineering. The WSU and Reed datasets represent course details offered by Washington State University and Reed College, respectively. The third dataset contains a collection of functionally annotated protein sequences.

**Washington State University (WSU) dataset**: The dataset WSU is compiled from information about the courses offered by Washington State University, USA. The dataset includes the course details, time, and location. The WSU dataset contains 74557 elements, and its maximum depth is 4. The WSU dataset was selected because it is considered to be a standard and benchmark dataset for the analysis of XML data mining methods.

**Reed dataset**: The Reed dataset is derived from details of courses offered by Reed College, Oregon, USA. The Reed dataset contains a total of 10546 elements with a depth of 4.The Reed dataset is also a benchmark dataset for analyzing XML data mining methods.

**Protein sequence dataset**: The protein sequence dataset is an integrated collection of functionally annotated protein sequences. These sequences are stored as an XML dataset. The protein sequence dataset is one of the most commonly used datasets for testing XML data mining methods. This dataset contains details of different proteins in sequences, stored in an XML format. It contains a total of 21305818 elements and 1290647 attributes. The average depth of the XML is 5.15, and it possesses a maximum depth of 7. The structure of the dataset is dissimilar to the other 2 mentioned above and was very helpful in evaluating the proposed projection based-mining method because it includes attributes whereas the previous two do not.

**Table 11.** Datasets used

| S. No | Name of the Dataset | No. of elements | Maximum Depth | Average Depth | Space |
|---|---|---|---|---|---|
| 1 | Washington State University | 74557 | 4 | 3.15787 | 1 MB |
| 2 | Reed | 10546 | 4 | 3.19979 | 277 KB |
| 3 | Protein Sequence | 21305818 | 7 | 5.15 | 683 MB |

## 5.2 Performance analysis

In this section, we evaluate the performance of the proposed projection-based mining algorithm in terms of the number of rules generated, the amount of time required, and the amount of space used for processing the datasets under a given support value. The WSU and Reed datasets were mined first, and the performance of the proposed approach was analyzed from the evaluation results. The graphs plotted below represent the performance of the proposed mining algorithm in comparison with that of the Li et al. (2007) index-table method.

The performance of the proposed approach is evaluated according to the number of rules generated, and the time and space required. The number of rules obtained from the dataset set is calculated in order to identify the effectiveness of the proposed approach. A method is said to be effective if it generates an optimal number of rules and the generated rules are meaningful and associated with each other. The meaningfulness of each rule is estimated by checking the association of each element with which the rule is generated. The number of rules generated is used to assess effectiveness, and time and space are used to assess the efficiency of the proposed approach.

### 5.2.1 The WSU dataset

Figures 1, 2, and 3 represent the performance of the proposed UID-XML mining approach compared to the index table-mining approach proposed by Li et al. (2007) with respect to the WSU dataset. The data mining was conducted by setting the support threshold to 30% confidence for both methods. The optimal confidence values that provided the best results for both algorithms were 60% and 80%. The evaluation process considered the number of rules generated, computation time, and space used. The number of rules obtained by our proposed method was 850 and that for the index-table approach was 1021. Although our method obtained fewer rules than Li's, our results are more specific. Our approach emphasizes the UID of every object and finds the closest association of the UID to other UIDs while Li's merely selects most of the indexed IDs. The computation time analysis (Figure 2) indicates that our proposed approach consumes less time for the same set of data. In Figure 3 for a confidence value of 80% the results converge, which shows that the methods use equal space because the number of rules generated for this confidence value is about the same for both methods.

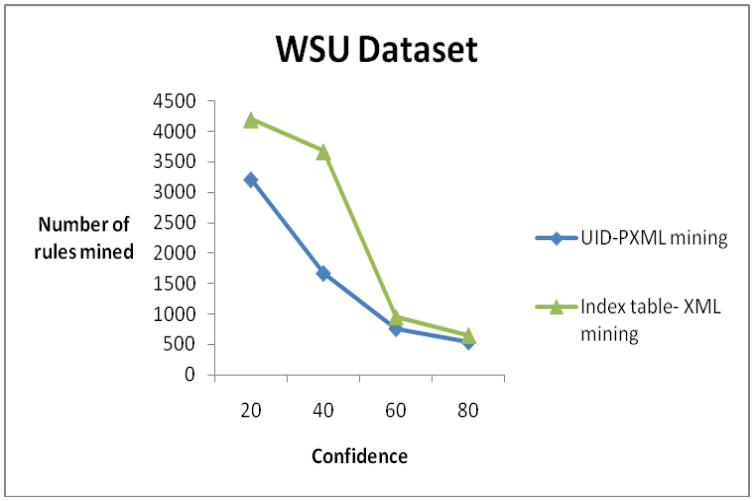


**Figure 1.** Number of rules generated for support=30%; confidence expressed as %.

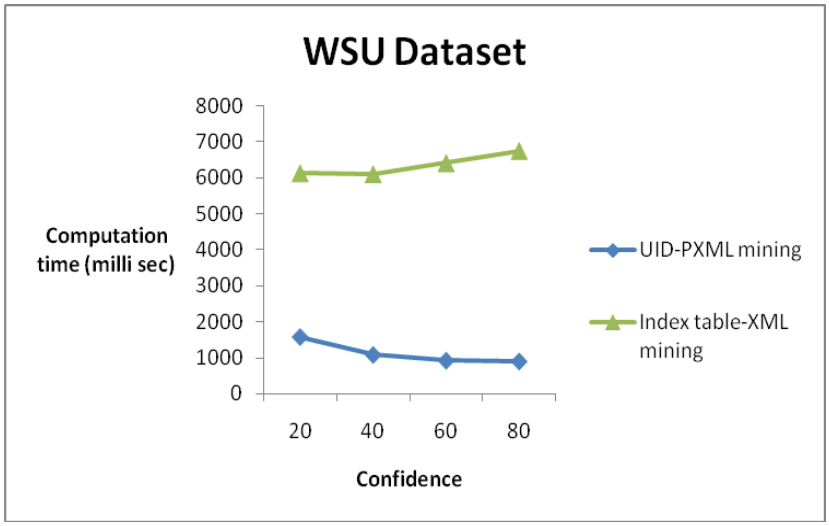


**Figure 2.** Computation time for support=30%; confidence expressed as %.

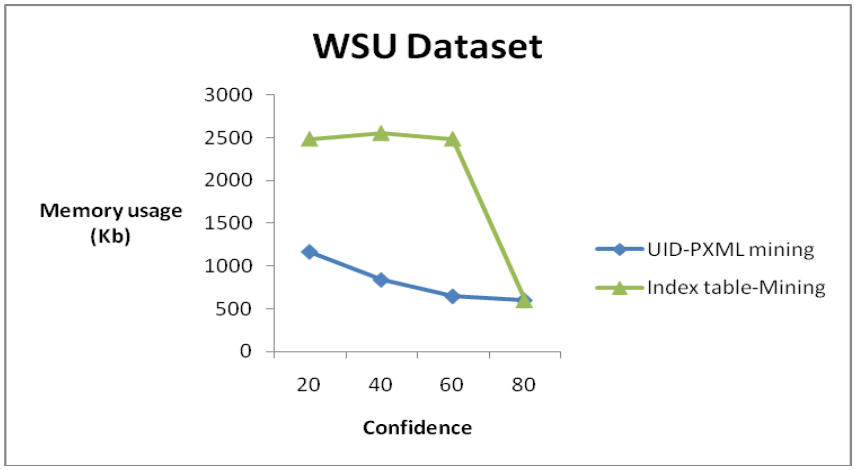


**Figure 3.** Memory usage for support=30%; confidence expressed as %.

### 5.2.2 The Reed dataset

To mine XML rules from the Reed dataset, we fixed the support at 30% and varied the confidence from 20 to 80%. Then we found the number of rules and identified computation time and memory usage. Figures 4, 5, and 6 show the comparison between the index-table approach and the proposed approach for the Reed dataset. Similar to the findings using the WSU dataset, the proposed approach provided the optimal number of rules more efficiently than the existing approach. The space and time taken for the proposed method are less when compared to the existing method. As the Reed dataset is comparatively small, the number of rules mined converged.

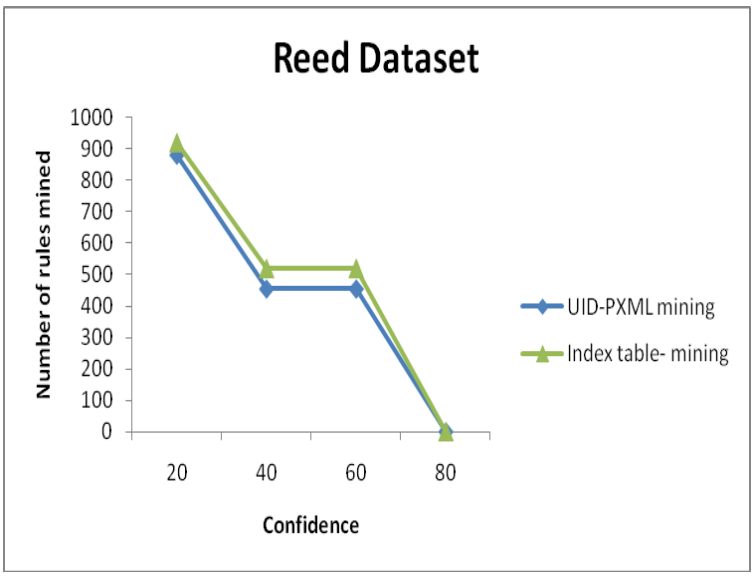


**Figure 4.** Number of rules generated for support=30% ; confidence expressed as %.

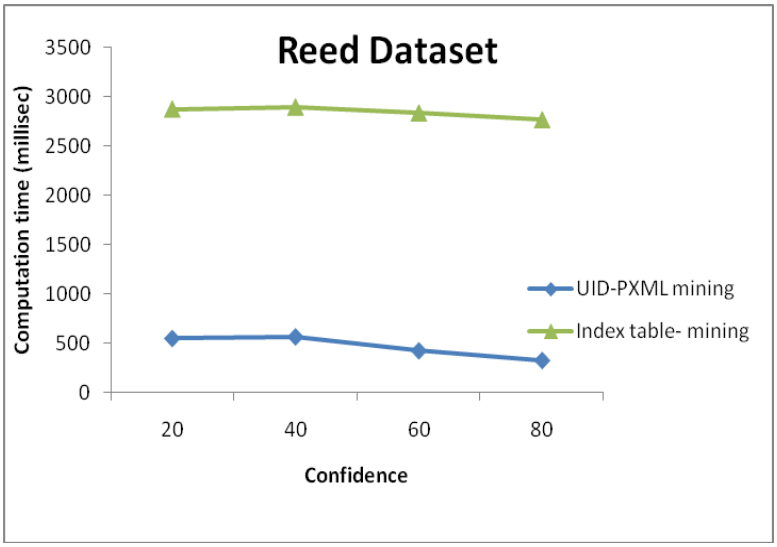


**Figure 5.** Computation time for support=30%; confidence expressed as %.

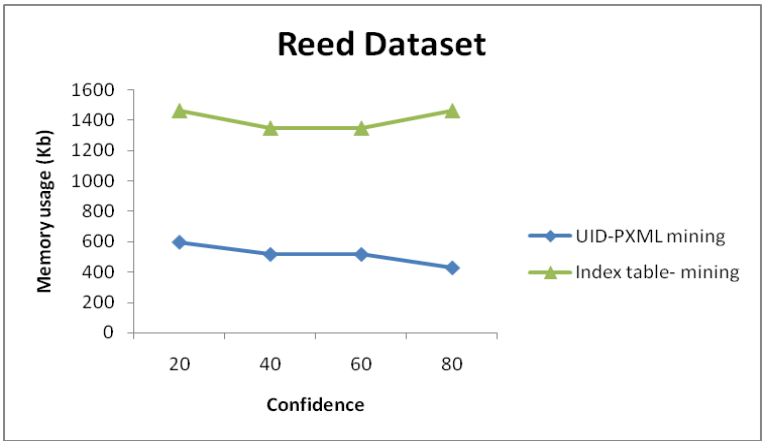


**Figure 6.** Memory usage for support=30%; confidence expressed as %.

### 5.2.3 Protein sequence dataset

Using our proposed approach, a section of the protein sequence dataset was processed, including maximum depth elements. The proposed approach was executed, and the responses were recorded. The responses were stored in terms of memory, execution time, and number of rules generated. This method was not compared with the Li et al. method because the later method does not support mining attributes. Figures 7, 8, and 9 represent the responses of the proposed approach when applied to the protein sequence dataset. The experiment was conducted by varying the confidence value from 20% to 40% to 50% and 60%. The support was kept constant at 30%. An analysis of the figures shows that the proposed approach performs well under confidence value 40% and support value 30%. For the given data of the protein sequence dataset, the proposed approach has optimized a number of rules, obtaining 2768 at confidence 40% and 1858 at confidence 60%.

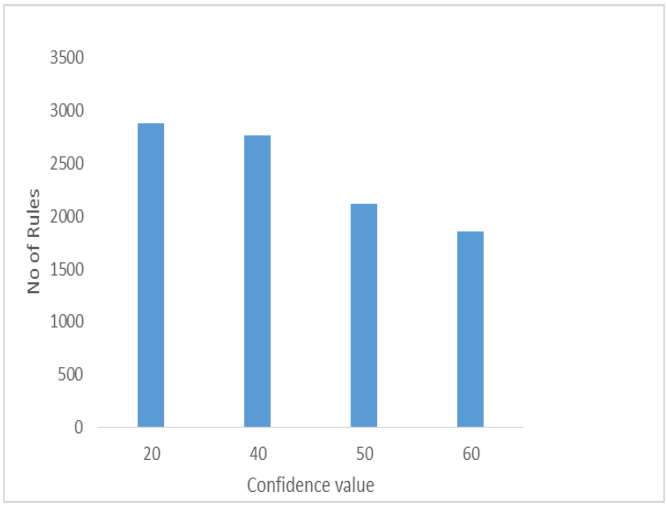


**Figure 7.** Number of rules obtained at support =30%; confidence expressed as %.

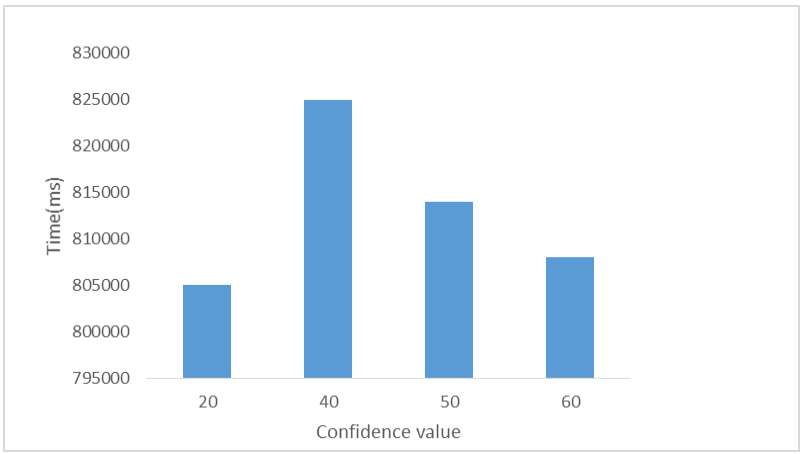


**Figure 8.** Time for execution at support =30%; confidence expressed as %.


**Figure 9.** Memory usage at support 30%; confidence expressed as %.

## 6 CONCLUSIONS

Our proposed projection-based approach to mine the data from XML databases is characterized by finding more precise rules that are relevant to the problem scenario. The algorithm works by calculating different length values, i.e., the relationship between each object. A unique ID based system is used for parsing an XML document, which also provides the foundation for the proposed approach for XML mining. The proposed projection-based mining was used on three datasets, and the results were compared with mining the same datasets using the existing index-table approach from Li et al. (2007) The experimental results show that the

proposed approach generated more precise optimized rules as compared to the index-based method. Although the proposed approach obtained 850 rules while the existing approach obtained 1021 rules at confidence value 60%, the optimal confidence value, the precision of the 850 rules is greater than that of the larger number. This is true because the proposed approach emphasizes the UID of every object and finds the closest association of each UID to other UIDs. The Li approach just selects most of the idexed IDs. The results from the analysis also showed that the proposed approach needed less computation time and less memory usage than the existing algorithm. Even though our proposed approach converges with Li's approach, it takes less time and space for the computation, i. e., to generate the equivalent number of rules the proposed approach takes less time and space. Thus we can state that the projection-based mining algorithm performs well.

## 7 REFERENCES

Abiteboul, S., Buneman, P., & Suciu, D. (1999) *Data on the Web: From Relations to Semi-Structured Data and XM,*. Morgan Kaufmann Publishers: San Francisco, CA, USA.

Abiteboul, S., Buneman, P., & Suciu, D. (2000) *Data on the Web*, Morgan Kaufmann.

Ali Mohammadzadeh, R., Rahgozar, M., & Zarnani, A. (2006) A New Model for Discovering XML Association Rules from XML Documents. *World Academy of Science, Engineering and Technology 21*, pp 160-165.

Alishahi, M., Naghibzadeh, M., & Aski, B. S. (2010) Tag Name Structure-based Clustering of XML Documents. *International Journal of Computer and Electrical Engineering 2*(1), pp 119- 126.

Doucet, A. & Ahonen-Myka, H. (2002) Naive clustering of a large XML documentollection. *Proceedings of the 1st INEX*, Germany.

Feng, L. & Dillon, T. (2005) Mining Interesting XML-Enabled Association Rules with Templates *Lecture Notes in Computer Science 3377*, pp 66-88.

Kliegr, T. & Rauch, J. (2010) An XML Format for Association Rule Models Based on the GUHA Method. *Lecture Notes in Computer Science 6403*, pp 273-288.

Lee, D. & Chu, W. (2000) Comparative analysis of six XML schema languages. *SIGMOD Record 9*(3), pp 76–87.

Li, X.-Y., Yuan, J.-S., & Kong, Y.-H. (2007) Mining Association Rules from XML Data with Index Table. *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, Hong Kong, pp 19-22 and pp 3905-3910.

.

Mazuran, M., Quintarelli, E., & Tanca, L. (2009) Mining tree-based association rules from XML documents. *Proceedings of SEBD*, pp 109-116.

Murtaugh, G. (2000) Clustering of XML Documents. *Computer Physics Communication 127*, pp 215-227.

Nayak, R. (2005) Discovering Knowledge from XML Documents. In Wong, J. (Ed.) *Encyclopedia of Data Warehousing and Mining, Idea Group Publications,* pp 1-1382.

Nayak, R. (2008) XML Data Mining: Process and Applications. In. Song, M. &Wu, Y. F. (Eds.) *Handbook of Research on Text and Web Mining Technologies*. Idea Group Inc: USA.

Nayak, R. (2008) Fast and effective clustering of XML data using structural information. *Knowledge Information System 14*(2), pp 197-215.

Porkodi, R., Bhuvaneswari, V., Rajesh, R., & Amudha, T. (2009) An Improved Association Rule Mining Technique for Xml Data Using Xquery and Apriori Algorithm. *Proceedings of the IEEE International Advance Computing Conference*, pp 1510 - 1514.

Shaharanee, I., N., M., Hadzic, F., & Dillon, T. (2010) A Statistical Interestingness Measures for XML Based Association Rules. *Lecture Notes in Computer Science 6230*, pp 194-205.

Shahriar, M. S. & Liu, J. (2011) On mining association rules with semantic constraints in XML. *Sixth International Conference on Digital Information Management (ICDIM).*

Tagarelli, A. & Greco, S. (2006) Toward Semantic XML Clustering. *Proceedings of the Siam Conference on Data Mining,* Maryland, USA, pp 188-199.

University of Washington Department of Computer Science and Engineering. Retrieved from the World Wide Web, February 11, 2014: http://www.cs.washington.edu/research/xmldatasets/www/repository.html#pir

Yi, J. & Sundaresan, N. (2000) A classifier for semi-structured documents. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 340-344.

Zhao, Q., Chen, L., Bhowmick, S., & Madria, S. (2005) XML Structural Delta Mining: Issues and Challenges. *Data & Knowledge Engineering 59* (3), pp 627 – 651.

(Article history: Received 25 March 2013, Accepted 17 December 2013, Available online 3 April 2014)