PATTERN DISCOVERY USING ENHANCED FUZZY APRIORI ALGORITHM (EFAA)

4.1 Introduction

4.2 Pattern Discovery
4.2.1 Path Analysis
4.2.2 Association Rules
4.2.3 Sequential Patterns
4.2.4 Clusters and Classification Rule

4.3 Association Rule Mining (ARM)
4.3.1 Definition
4.3.2 Key Concepts
4.3.3 Support
4.3.4 Confidence
4.3.5 Process
4.3.6 Apriori Algorithm
4.3.7 Drawbacks and Solutions

4.4 Proposed Enhanced Fuzzy Apriori Algorithm Methodology
4.4.1 Preprocessing of Web Log Data
4.4.2 Enhanced Fuzzy Apriori Algorithm (EFAA)
   4.4.2.1 Frequent Itemset And Rule Generation
   4.4.2.2 Fuzzy Intersection Pruning (FIP)

4.5 Experimental Results

4.6 Chapter Summary
Chapter - 4

PATTERN DISCOVERY USING
ENHANCED FUZZY APRIORI ALGORITHM (EFAA)

4.1 INTRODUCTION

Mining frequent traversal patterns is one of the most important techniques in web usage mining and forms the focus of this research work. Frequent Pattern (FP) mining has been applied extensively in the supermarket transaction data and the relational data. The Apriori algorithm (Xiao and Dunham, 2001) is a decisive algorithm, used to discover frequent patterns. It is based on the fact that the algorithm utilizes prior knowledge of frequent patterns properties, which means all nonempty subsets of a frequent pattern, must also be frequent. In Apriori algorithm, to discover the pages that are visited together even if they are indirectly connected, association rules are used. It also reveals the associations between groups of users with specific interest (Eirinaki and Vazirgiannis, 2003). This information can be used for restructuring websites by adding links between those pages which are visited together.

Association rules in web logs are discovered in (Liu, 2007; Mobasher 2006). FP-growth (Han et al. 2004) is another representative of frequent pattern mining algorithm. It espouses a pattern growth approach and divide-and-conquer strategy in search. Apriori-All and Generalized Sequential Pattern (GSP) algorithms are proposed to mine sequential patterns from spatiotemporal event datasets, which are planned with no reproduction and is uninterrupted (Huang et. al., 2008). Extraction of useful information from the set of generated association rules remains a difficult task, though a variety of measures and rule pruning methods have been applied to association rule mining of web usage data (Huang, 2007).

Apriori algorithm, using the Apriori principle, considerably reduces the size of candidate sets that suffers from two nontrivial costs. A huge number of candidate sets is generated initially and subsequently, the database is repeatedly scanned and the candidates are checked by pattern matching. Most association rule mining algorithms discover all the rules that contain frequent itemsets, which means items that coexists in the same session with the probability above a user specified minimum support threshold. But, the number of the discovered rules is often overwhelming, making it
difficult or even impossible for the data analyst to understand and utilize the rules (Shiying and Webb 2005).

To solve this problem, rule pruning methods and automatic support and confidence calculation method are introduced in this chapter. The goal is to study and overcome the problems of Apriori algorithm by proposing a fuzzy based rule pruning algorithm in which itemset pruning is used to reduce the number of candidate itemset in the web log data. Itemset are pruned using Fuzzy Intersection Pruning (FIP) to enhance pattern discovery results which reduces the number of candidate items in the frequent itemset mining. Fuzzy Automated Support Confidence Pruning (FASCP) technique arranges the rules in ascending order of support and confidence thresholds which is found from FAS for association rules generation.

The Enhanced Fuzzy Apriori Algorithm (EFAA) investigates how effective association rules can be when discovering potentially useful information in the web log database. Another goal of this study is to investigate what needs to be done with a set of association rules generated by the rule pruning algorithm, so that extracting useful knowledge from the web log data becomes a task worth carrying out.

The remainder of this chapter is summarized in the following manner. Initially, section 4.2 discusses and describes the information about the general pattern discovery, basic procedure of Apriori algorithm and finally the issue of same is discussed in detail. Section 4.3 introduces the association rule mining concepts and Apriori algorithm is discussed in detail. Section 4.4 introduces the proposed Enhanced Fuzzy Apriori Algorithm (EFAA) for pattern discovery of web user. Section 4.5 analyses the experimental results of the proposed method in comparison with existing methods by applying it on benchmark datasets. Finally, Section 4.6 presents the chapter summary.

4.2 PATTERN DISCOVERY

After the data preprocessing phase, the pattern discovery method is applied. Pattern discovery is the main phase in both web usage mining and data mining. Many algorithms have been proposed for capturing frequent user navigation patterns. In very huge datasets, finding desired patterns is quite challenging. The search space increases exponentially as the pattern length increases. Difficulty lies in interpreting the
discovered patterns and extracting useful knowledge from them. Various techniques derived from domains such as machine learning, statistics, pattern recognition, data mining etc are applied to the web logs in web usage mining. Some of the pattern discovery techniques used are association rules, statistical analysis, and clustering, sequential pattern analysis as illustrated in Figure 4.1.

**Figure 4.1 Pattern Discovery**

### 4.2.1 Path Analysis

In path analysis, the graph models are frequently used. In the representation of website using graph, each node in the tree represents a web page (html document), and the edges between the nodes represents the link between web pages. Here the nodes inside the same tree represent links between documents in a web site.

### 4.2.2 Association Rules

The goal is to predict the correlation of items where the occurrence of one set of items in a transaction implies the occurrence of other items. Association Rule Mining (ARM) is one of the major techniques of data mining and most commonly used for pattern discovery in unsupervised learning systems (Kotsiantis and Kanellopoulos, 2006; Romero et al. 2010; Mahafzah, Al-Badarneh and Zakaria, 2009). It serves as a
useful tool for finding correlations between items in large database. This method can be used to find the group of pages which are frequently accessed together with support exceeding a threshold. It is not necessary that the pages are connected directly. For example, Apriori algorithm can be used to find relation between users who access a faculty page of a college and those who access a syllabus download page.

4.2.3 Sequential Patterns

Sequential patterns discover the user’s navigation behavior. The sequence of items occurring in one transaction has a particular order between the items or the events. The same sequence of item may reoccur in the same order. For example, 30% of the users may undergo the link in this order “A=>B=>C=>D=>E” where A, B, C, D, E corresponds to each web page.

4.2.4 Clusters and Classification Rule

This process groups profile of items with similar characteristics. This ability enhances the discovery of relationships. For example, to discover the average age of customers who order a certain product, a company can use the classification of web access logs. The extracted information is used for developing new advertising strategies.

4.3 Association Rule Mining (ARM)

Association Rule mining (ARM) is a prominent data mining technique in research. The market basket analysis is a well known ARM application, where the relationship between objects in large databases is found using a rule based knowledge representation. Using some measures of interestingness, strong rules are identified from databases.

An association rule $X \Rightarrow Y$ states that, the occurrence of $X$ in any transaction in the database has a high probability of having $Y$. $X$ and $Y$ are the antecedent and consequent of the rule respectively. Support and confidence are used to measure the strength of the rule. The percentage of transactions with $X$ in the database having the consequent $Y$ is the confidence of the rule. The percentage of transactions in the database having both the antecedent and the consequent is the support of the rule.
4.3.1 Definition

ARM is defined as:
Let \( I = \{i_1, i_2, \ldots, i_n\} \) denote a set of \( n \) binary attributes called items.
Let \( D = \{t_1, t_2, \ldots, t_m\} \) denote a set of transactions called the database.
A unique transaction ID and contains a subset of the items in \( I \) is present in every transaction of \( D \). A rule is defined as an implication of the form:
\[ X \Rightarrow Y, \text{ where } X, Y \subseteq I. \]

4.3.2 Key Concepts

Constraints on various measures of significance and interestingness are used to select interesting rules from the set of all possible rules. Minimum thresholds on support and confidence are the well known constraints. Let \( X \) is an itemset, \( X \Rightarrow Y \) an association rule and \( T \), a set of transactions in a given database.

4.3.3 Support

Support is an indication of how frequently the itemset appears in the dataset. The support of \( X \) with respect to \( T \) is defined as the proportion of transactions \( t \) in the dataset which contains the itemset \( X \).

\[
\text{Supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \tag{4.1}
\]

4.3.4 Confidence

Confidence is an indication of how often the rule has been found to be true. Given a rule \( X \Rightarrow Y \), \( T \) a set of transactions, the confidence value is the proportion of the transactions that contains \( X \) which also contains \( Y \). Confidence is defined as:

\[
\text{Conf}(X \Rightarrow Y) = \frac{\text{Supp}(X \cup Y)}{\text{Supp}(X)} \tag{4.2}
\]

4.3.5 Process

Association rules are expected to simultaneously satisfy a minimum support and a minimum confidence that is user specified. The process consists of two steps.

- To find all frequent item sets in a database a minimum support threshold is applied.
- In order to form rules the frequent item sets are subjected to a minimum confidence constraint.
Criticality lies in finding the minimum support and confidence. Finding all frequent item sets in a database which involves searching of all item combinations in an itemset is a difficult task. The set of all possible itemsets is the power set over I and has size $2^n - 1$ (excluding the empty set which is not a valid itemset). Given the number of items $n$ in I, the size of the power set grows exponentially. Using the downward closure property of support (Agrawal, Imieliński and Swami, 1993; Tan et al. 2005) an efficient search is carried out (also called anti-monotonicity (Pei, 2011) which guarantees that for a frequent itemset, all its subsets are also frequent. To conclude, no infrequent itemset can be a subset of a frequent itemset.

4.3.6 Apriori Algorithm

The Apriori Algorithm is an influential algorithm for mining frequent item sets for boolean association rules. It is used for frequent itemset mining and ARM over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger itemsets as long as those itemsets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight the general trends in the database.

In Apriori algorithm, candidate generation i.e., frequent subsets are extended one item at a time, which is a bottom up approach. Groups of candidates generated are tested against the data. To count candidate itemsets efficiently Apriori uses Breadth First Search (BFS) and a hash tree structure. From itemsets of length $k-1$, it generates candidate itemsets of length $k$. The candidates are pruned for an infrequent sub pattern. Before each scan, in order to load up the candidate set with as many as possible, the algorithm generates large numbers of subsets. According to the downward closure lemma, the candidate set contains all frequent $k$-length itemsets. To determine frequent itemsets among the candidates the transaction database is scanned.

The Apriori algorithm is based on downward closure property, all the subsets of a frequent itemset are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. In the Apriori framework, candidates of $(k +1)$ itemsets are generated from known frequent $k$-itemsets by adding one or more possible frequent item. The mining begins at 1-itemset and the size of candidate itemsets increases by one at each level. In each level the Apriori algorithm has two major operations:
• Generating candidates of frequent \((k + 1)\) itemsets from known frequent \(k\)-itemsets.

• Counting support for numbers of candidate itemsets and comparing these support numbers with \(\text{minsup}\).

• For large databases, the support counting step is the performance hurdle of the \textit{Apriori} algorithm.

\textbf{4.3.7 Drawbacks and Solutions}

In the \textit{ARM} area, the priority in most of the research efforts went to improvise algorithmic performance (Ceglar, Roddick, 2006) and then to reduce the output set, by allowing the possibility to express constraints on the desired results. Over the past decade, a variety of algorithms that addresses these issues through the refinement of search strategies, pruning techniques and data structures have been developed. Most of the algorithms focus on the explicit discovery of all rules that satisfy minimal support and confidence constraints for a given dataset. Importance is given to specialized algorithms that attempt to improve processing time or facilitate user interpretation by reducing the result set size and by incorporating domain knowledge (Goethals, Nijssen, Zaki, 2005).

\textbf{Discovery of Too Many Rules:} The application of traditional association algorithms will be simple and efficient. However, an \textit{ARM} algorithm normally discovers a huge quantity of rules and does not guarantee that all the rules found are relevant. The support and confidence factors can be used for obtaining interesting rules which have values for these factors greater than a threshold value. Although these two parameters allow the pruning of many associations, another common constraint is to indicate the attributes that can or cannot be present in the antecedent or consequent of the discovered rules.

Another solution is to evaluate, and post prune the obtained rules in order to find the most interesting rules for a specific problem. Traditionally, the use of objective interestingness measures has been suggested (Tan, Kumar, 2000), such as support and confidence. Other measures such as Laplace, chi-square statistic, correlation coefficient, entropy gain, conviction, etc., are also used. These measures can also be used for ranking the obtained rules, facilitating the user to select the rules with higher measures of interestingness.

\textbf{Discovery of Poorly Understandable Rules:} Comprehensibility is an important factor in determining the quality of the extracted rules. Due to subjective nature of
Comprehensibility the quality of rule is often overlooked and cannot be measured independently of the person using the system (Huysmans, Baesens and Vanthienen, 2006), though the main motivation for rule extraction is to obtain a comprehensible description of the underlying model's hypothesis. In the assessment of comprehensibility prior experience and domain knowledge plays an important role. This contradiction with accuracy is evaluated independently of the users and considered as a property of the rule. To improve the comprehensibility of discovered rules many traditional techniques have been used. By constraining the number of items in the antecedent or consequent of the rules, the size of the rules can be reduced. To solve these problems, ARM task is improved by proposing Fuzzy Intersection Pruning (FIP) and Fuzzy Automated Support Confidence Pruning (FASCP) algorithms.

### 4.4 Proposed Enhanced Fuzzy Apriori Algorithm Methodology

In this work a frequent itemset generation called EFAA is proposed for solving the pattern discovery. An innovative method is used for sorting candidate set of items and counting their support, then to reduce size of the dataset significantly as execution progresses the development of effective rule pruning algorithm is utilized.

![Figure 4.2 Flow Diagram for the Proposed Methodology](image)
Mining task is improved by using the two pruning algorithms, which are Fuzzy Intersection Pruning (FIP) and Fuzzy Automated Support Confidence Pruning (FASCP). FIP technique is proposed for itemset pruning to enhance the accuracy of frequent itemset mining results. Steps carried out to increase the pattern discovery performance as illustrated in Figure 4.2 are described as follows.

4.4.1 Preprocessing Of Web Log Data

Web information is heterogeneous and semi structured or unstructured in nature. Due to this heterogeneity, preprocessing is performed to transform the raw click stream data into a set of user profiles. Preprocessing enables to translate the unprocessed data which is collected from the server log files into constructive data abstraction. Data preprocessing consists of three sub phases namely data identification, data cleaning and parsing as discussed detail in chapter 3.

**Parsing:** Web log access file on the server side contains web log information of a user who opens a dynamic session. These web log files include the list of items that a user agent has accessed. The log entry contains information about the corresponding SQL query, such as the Query content (Sql), Beginning time (StartTime), Finishing time (EndTime), Connection id (Spid), Request method (GET or POST), Success of return code and Number of bytes transmitted. In this context, the information to be extracted is defined as the click stream in a user dynamic session for a particular web server with user given query.

The parsing algorithm given below transforms a set of web log database files, WLD, expressed as, $WLD = \{WLD_1, \ldots WLD_i, \ldots WLD_l\}$. $|WLD|$: The number of web log database fields in each session,

$$WLD = \{IP_i, sql, starttime_i, endtime_i, spid, GET_i, CODE_i, BYTE_i, password\}.$$  

Each web log session contains $IP_i$, $PAGES_i$ sql that are navigated in that session, $WLS = (WLS_1, \ldots WLS_k, \ldots WLS_l), \forall i \leq |WLS|$, $PAGES_i = (URL_{i-1}, \ldots URL_{i-k})$.

Function EndTime returns the maximum time detected in the page set of corresponding session. Function Close_session removes corresponding session from Open_session set. Function Open_session adds corresponding session to open sessions set. When an URL is being considered the algorithm checks whether the given URL is in valid format. The algorithm discards the corresponding log entry if any incomplete or invalid format URL is found.
Improved Strategies for Session Identification and Frequent Pattern Generation in Web Usage Mining

### Parsing Algorithm

\[ \text{WLD}_i = \{ \text{WLD}_1, \ldots, \text{WLD}_n \} \]

| \text{WLD} | : The number of web log database fields in each session
\[ \Delta t : \text{time period} \]

\[ \text{WLD}_i = \{ \text{WLD}_1, \ldots, \text{WLD}_n \}, \forall i \leq |\text{WLD}| \]

Input : WLD, \Delta t, output: WLS, |WLS|

Function Log parser(\{WLS\}, WLS, \Delta t)

For each WL of WLD

If methods is ‘GET’ and URL is WEBPAGE // 1. If

If \( \exists \text{WLS}_k \in \text{open sessions} \) with \( \text{IP}_k = \text{IP}_l, \text{sql} \) then // 2. if

If \(( \text{time}_k - \text{endtime}_k ) < \Delta t \) then // 3. if

\[ \text{WLS}_k = ( \text{IP}_k, \text{PAGES}_k \cup \text{URL}_k ) \]

Else

Close_session(WLS_k)

Open_session(IP_l, URL_k)

End if // end of 3, if

Else

Open_session(IP_l, URL_k)

End if // end of 2, if

End if // end of 3, if

End for

### 4.4.2 Enhanced Fuzzy Apriori Algorithm (EFAA) algorithm

Enhanced Fuzzy Apriori Algorithm (EFAA) algorithm is proposed for effective pattern discovery of web log session by exploiting the frequent itemset count technique, in which an optimized minimum support value is calculated to reduce the number of infrequent itemsets.

An itemset pruning is performed by using Fuzzy Intersection Pruning (FIP) that finds the most frequent items in the web log session frequent itemset. To compute the measure value of a rule, the Fuzzy Automated Support (FAS) count value is used to get the optimal minimum support and generates frequent itemset. The work implements rule pruning methods to reduce the number of rules generated by AA.

The initial stage of work uses Fuzzy Automated Support (FAS) value as frequent itemset count technique. In this stage we reduce the length of itemset by calculating
optimal minimum support value FA_S. Let WLSI = \{wl_{si1},wl_{si2},...,wl_{sim}\} be a set of items. Let WLST be a set of transactions. The main aim of association rule mining is to find out rules that have support and confidence greater than a user-specified minsup (minimum support) and minconf (minimum confidence) (Zhang, 2003; Zhang et.al. 2008).

Let WLST is the web log session transaction database with n itemsets WLSI = \{wli_{1},wli_{2},...,wli_{m}\} and let S be the support of each item which is found from
\[
\mu_F : [M_{a},M_{b},M_{c},M_{d}] \rightarrow [0,1].
\]
Let M be the maximum number of items in WLST. The proposed method, first scans WLST to estimate S of each web log session item, from which the average support is calculated (Equation 4.3). This average value is used as initial min-supp which is then used during the optimal minimum support value estimation.

\[
FA_S = \frac{\sum_{i=1}^{n} S}{n}
\]

(4.3)

The concept of FAS value is to determine min-supp ‘S’ of ‘WLST’ within the interval \([M_{a}, M_{b}, M_{c}, M_{d}]\), using fuzzy membership function as a mapping \(\mu_F : [M_{a},M_{b},M_{c},M_{d}] \rightarrow [0,1]\). Here, \(M_{a}\) is the low minimum, \(M_{b}\) is the minimum, \(M_{c}\) is the maximum support and \(M_{d}\) is the high maximum support in WLSD. As, in many cases, this mapping is hidden, it is necessary to find a fuzzy membership value as optimal support value given in Equation (4). Here, let \(X = (x_1, x_2, ..., x_n)\) be the number of items in Web Log Session Data (WLSD) in \([M_{a},M_{b},M_{c},M_{d}]\) frequent and FAs be the new optimal support value. The minimum confidence is assumed as the user defined threshold.

\[
S = \mu_F(x,M_{a},M_{b},M_{c},M_{d}) = \begin{cases} 
0 & \text{if } x < M_{a} \\
\frac{x - M_{a}}{M_{b} - M_{a}} & \text{if } M_{a} = x = M_{b} \\
\frac{x - M_{c}}{M_{d} - M_{b}} & \text{if } M_{b} < x < M_{c} \\
\frac{M_{d} - x}{M_{d} - M_{c}} & \text{if } M_{c} = x = M_{d} \\
1 & \text{if } M_{d} < x
\end{cases}
\]

(4.4)

4.4.2.1 Frequent Itemset and Rule Generation

For calculated optimal support value FA_S, rules are generated in EFAA algorithm. Rule pruning algorithms are used to reduce the high number of association
rules generated by EFAA. Two pruning technique, namely, Fuzzy Automated Support Confidence Pruning (FASCP) and Fuzzy Intersection Pruning (FIP) are used in succession to improve pattern discovery results.

4.4.2.2 Fuzzy Intersection Pruning (FIP)

Any maximal web log session frequent itemset is also the maximal corresponding to a certain transaction in WLST, and merge all maximal web log session frequent itemset corresponding to every transaction into one set (which is denoted as WLSFI), and then delete all non frequent web log session maximal itemset in WLSFI, and the remaining set is the maximal frequent itemset.

FASCP starts by arranging the rules in ascending order of support ($FA_s$) and confidence thresholds. Rules quality is identified using the parameters like $C$, $FA_s$, $N$ represents the confidence, fuzzy automated support and the number of attributes in the left hand side of the rules such as $R_1$ and $R_2$. Using this FASCP algorithm, higher ranks are assigned by quality rules. After ranking process, the rules are sorted in descending order and the pruning process begins.

<table>
<thead>
<tr>
<th>Pseudocode of Fuzzy Intersection Pruning (FIP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let us consider $D_2$ to be a set of transactions in web log session, and the fuzzy minimal support threshold is $FA_s$.</td>
</tr>
<tr>
<td>Step 1. Consider a data-set WLST; if $</td>
</tr>
<tr>
<td>Step 2. Perform intersection operation of $WLST_1$ and $WLST_i$ ($1 &lt; i \leq n$)</td>
</tr>
<tr>
<td>2.1 Merge all intersections into a new data-set $FIPD_1$</td>
</tr>
<tr>
<td>2.2 if $</td>
</tr>
<tr>
<td>Step 3. Use the vertical data format of $D$ to find the intersection of $T_j$ and $T_i$ ($j = 2, 3, 4, \ldots, m &lt; n; j &lt; i \leq n$).</td>
</tr>
<tr>
<td>3.1 Merge all intersections into a new data-set $FIPD_2$, go to step 1 to perform another intersection pruning for $FIPD_2$.</td>
</tr>
<tr>
<td>3.2 if $</td>
</tr>
<tr>
<td>Step 4: End;</td>
</tr>
</tbody>
</table>

During pruning process, each rule should satisfy the property of generality. After ranking, the top-K rules are taken as optimal rules. The K value is estimated by using $k = \sqrt{\frac{n}{2}}$, where $n$ is the number of association rules. The result of FASCP algorithm is a set of rules arranged in descending order according to its rank.
A new Enhanced Fuzzy Apriori Algorithm (EFAA) is proposed for effective pattern discovery of web log session. The pseudo code of the EFAA, which uses the direct count technique, is given below. First the frequent itemset counters used for the pruning rules (line 1) is updated. The pseudo code for subroutine Global rules \((GR_k, WLSFI_i)\) handles the vector of Number of web log session items \((wlsm)\), frequent itemset count as \(C_{k[i]}\).

To simplify the frequent itemset, counter \(C_{k[i]}\) is incremented each time an item \(i\) is included in a web log frequent \(k\) itemset of the \(WLF_k\), with the satisfaction of Fuzzy Automated Support (FAS) value \((FA_s)\) calculation (at line 6). For each updated new dataset \(D_1\), FIP is performed (at step 7) and rules are pruned from the rule set by using FASCP (at step 8) and from this step generate all the 2 frequent itemset of the pruned web log data session transaction and the corresponding results are described at step 9.

**Pseudocode of Enhanced Fuzzy Apriori Algorithm (EFAA)**

**Input:**
- Number of web log session items \(-WLSI = \{wlsi_1, wlsi_2, ..., wlsi_m\}\), web log session frequent itemset \((WLSFI_i)\), size of the web log session items \(wlsm\), global rules \(GR_i\), rules set \(R = (r_1, r_n)\), Fuzzy Automated Minimum Support value \((FA_s)\), confidence \(C\), \(N\) represents the number of attributes.

**Output:**
Frequent itemset generation results
1. For each \(WLST\)
2. For all \(i \in [1, wlsm_k]\)
3. \(k \leftarrow 2\)
4. Global rules \((GR_i, WLSFI_i)\)
5. \(wlsm_k \leftarrow |WLSFI_i|\)
6. Compute Fuzzy Automated Minimum Support value calculation \((FA_s)\)
7. Generate new frequent itemset \((WLSFI_i)\)
   7.1 Consider a data-set \(WLST\) using the FIP method
      7.1.1 If \(|WLST| < FA_s\), terminate the processing for the current data-set
      7.1.2 Perform intersection operation of \(WLST_i\) and \(WLST_j\) \((1 < i \leq n)\)
         7.1.2.1 Merge all intersections into a new data-set \(FIPD_2\)
         7.1.2.2 If \(|FIPD_2| \geq FA_s\), then go to step 1 to perform another intersection
            pruning for \(FIPD_2\)
      7.1.3 Use the vertical data format of \(WLST\) to find the intersection of \(T_j\) and \(T_i\)
         \((j = 2, 3, 4, ..., m < n, j < i \leq n)\),
         7.1.3.1 Merge all intersections into a new data-set \(FIPD_3\), go to step 1 to perform another intersection
            pruning circle for \(FIPD_3\)
         7.1.3.2 If \(|FIPD_3| \leq FA_s\) stop finding intersections of \(T_j\) and \(T_i\), terminate
            the process for current data-set
   7.1.4 End
7. FASCP
   8.1 Arranging the rules in ascending order of support \((FA_s)\)
8.2 Higher $F_A \& C$ values top-K rules are taken as optimal rules
8.3 Top k rules values is determined by using $k = \sqrt{n}$
8.4 Find all applicable web log data session instances in $\text{WIST}$ that match $R_i$ condition
8.5 Then mark $GR_i$ correctly identifies in the frequent itemset $\text{WLSFI}_i$
8.6 Remove rules in the rule set $GR_i$ which is not covered by top K values
9 Generation of frequent itemset results
10 $k \leftarrow 3$ //go to step 4.

4.5 EXPERIMENTAL RESULTS

This section explains EFAA method for frequent itemset mining evaluated by comparing the performance of the method to the performance of the Apriori Algorithm (AA) and Modified Apriori Algorithm (MAA) methods. Proposed work explains rule pruning strategies to remove the irrelevant rules thus enhancing frequent itemset mining results. In this section, the performance of the pattern discovery is evaluated using different performance metrics. Each of the metrics results and their tabulated values on benchmark datasets are discussed in detail.

**Table 4.1 Comparison of Precision for Pattern Discovery Methods on Benchmark Datasets**

<table>
<thead>
<tr>
<th>Precision (%)</th>
<th>Methods</th>
<th>AOL Weblog</th>
<th>NASA Apache Weblog</th>
<th>EDGAR Log File</th>
<th>Apache Web Server log</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td></td>
<td>78.19</td>
<td>76.27</td>
<td>82.76</td>
<td>77.78</td>
</tr>
<tr>
<td>MAA</td>
<td></td>
<td>82.26</td>
<td>77.40</td>
<td>82.20</td>
<td>84.43</td>
</tr>
<tr>
<td>EFAA</td>
<td></td>
<td>86.39</td>
<td>86.97</td>
<td>88.53</td>
<td>88.85</td>
</tr>
</tbody>
</table>

Figure 4.3 shows that the proposed EFAA achieves 86.39 % precision which is 4.13 % and 8.2% higher when compared to MAA and AA methods respectively for AOL weblog dataset samples. Proposed EFAA achieves 86.97 % precision which is 9.57 % and 10.7% higher when compared to existing methods respectively for NASA Apache weblog dataset samples. In case of EFAA which achieves 88.53% precision and there is an approximate increase of 6.33 % and 5.77% precision when compared to existing methods respectively for EDGAR log dataset samples. The EFAA achieves 88.85% precision which is an increase of 4.42 % and 11.07% when compared to MAA and AA methods respectively for Apache web server log dataset samples (Refer Table 4.1).
Figure 4.3 Comparison of Precision for Pattern Discovery Methods under Study

Table 4.2 Comparison of Recall for Pattern Discovery Methods on Benchmark Datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>AOL Weblog</th>
<th>NASA Apache Weblog</th>
<th>EDGAR Log File</th>
<th>Apache Web Server log</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>77.67</td>
<td>75.00</td>
<td>80.00</td>
<td>81.67</td>
</tr>
<tr>
<td>MAA</td>
<td>85.00</td>
<td>82.33</td>
<td>84.67</td>
<td>89.67</td>
</tr>
<tr>
<td>EFAA</td>
<td>91.00</td>
<td>89.00</td>
<td>90.00</td>
<td>93.00</td>
</tr>
</tbody>
</table>
Figure 4.4 shows that the proposed EFAA achieves 91.00% recall which is 6% and 13.33% higher when compared to MAA and AA methods respectively for AOL weblog dataset samples. In case of EFAA which achieves 89% recall and there is an approximate increase of 6.67% and 14% recall when compared to existing methods for NASA Apache weblog dataset samples. Proposed EFAA achieves 90% recall which is 5.33% and 10% higher when compared to MAA and AA methods respectively, for EDGAR log dataset samples. The EFAA achieves 93% recall which is an increase of 3.33% and 11.33% recall when compared to the existing methods for Apache web server log dataset samples (Refer Table 4.2).

Table 4.3 Comparison of F-Measure for Pattern Discovery Methods on Benchmark Datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>AOL Weblog</th>
<th>NASA Apache Weblog</th>
<th>EDGAR Log File</th>
<th>Apache Web Server log</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>77.93</td>
<td>75.63</td>
<td>81.36</td>
<td>79.67</td>
</tr>
<tr>
<td>MAA</td>
<td>83.61</td>
<td>81.38</td>
<td>83.42</td>
<td>86.91</td>
</tr>
<tr>
<td>EFAA</td>
<td>88.64</td>
<td>87.97</td>
<td>89.26</td>
<td>90.88</td>
</tr>
</tbody>
</table>
Figure 4.5 shows that the proposed EFAA achieves 88.64% F-measure which is 5.03% and 10.71% higher when compared to MAA and AA methods respectively for AOL weblog dataset samples. Proposed EFAA achieves 87.97% F-measure which is 6.59% and 12.34% higher when compared to MAA and AA methods respectively for NASA Apache weblog dataset samples. In case of EFAA which achieves 89.26% F-measure, there is an approximate increase of 5.84 % and 7.9% F-measure when compared to the existing methods for EDGAR log dataset samples. The EFAA achieves 90.88% F-measure and there is an increase of 3.97% and 11.21% F-measure when compared to MAA and AA methods respectively for Apache web server log dataset samples. A higher F-measure value means a better overall performance (Refer Table 4.3).
Table 4.4 Comparison of Accuracy for Pattern Discovery Methods on Benchmark Datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>AOL Weblog</th>
<th>NASA Apache Weblog</th>
<th>EDGAR Log File</th>
<th>Apache Web Server log</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>73.60</td>
<td>71.00</td>
<td>78.00</td>
<td>75.00</td>
</tr>
<tr>
<td>MAA</td>
<td>80.00</td>
<td>77.40</td>
<td>79.80</td>
<td>83.80</td>
</tr>
<tr>
<td>EFAA</td>
<td>86.00</td>
<td>85.40</td>
<td>87.00</td>
<td>88.80</td>
</tr>
</tbody>
</table>

Figure 4.6 shows that the proposed EFAA achieves 86% accuracy which is 6% and 12.4% higher when compared to MAA and AA methods respectively for AOL weblog dataset samples. The EFAA achieves 85.4% accuracy which is an increase of 8% and 14.4% accuracy when compared to the existing methods for NASA Apache weblog dataset samples. Proposed EFAA achieves 87% accuracy which is 7.2% and 9% higher when compared to MAA and AA methods respectively for EDGAR log dataset samples. In case of EFAA which achieves 88.8% accuracy and there is an increase of 5% and 13.8% accuracy when compared to existing methods respectively for Apache web server log dataset samples (Refer Table 4.4).
Table 4.5 Comparison of Memory for Pattern Discovery Methods on Benchmark Datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>AOL Weblog</th>
<th>NASA Apache Weblog</th>
<th>EDGAR Log File</th>
<th>Apache Web Server log</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>36456</td>
<td>44286</td>
<td>45184</td>
<td>42513</td>
</tr>
<tr>
<td>MAA</td>
<td>34012</td>
<td>40581</td>
<td>42289</td>
<td>38954</td>
</tr>
<tr>
<td>EFAA</td>
<td>32653</td>
<td>36258</td>
<td>34526</td>
<td>33389</td>
</tr>
</tbody>
</table>

Figure 4.7 Comparison Of Memory For Pattern Discovery Methods Under Study

Figure 4.7 illustrates that the proposed EFAA consumes 32653 KB memory, which is 11.65% and 4.16% lesser when compared to MAA and AA methods respectively for AOL weblog dataset samples. In case of EFAA which consumes 36258 KB memory and there is a decrease of 22.14% and 11.92% memory consumption when compared to existing methods respectively for NASA Apache weblog dataset samples. Proposed EFAA consumes 34526 KB memory which is 30.86% and 22.48% higher
when compared to MAA and AA methods respectively for EDGAR log dataset samples. The EFAA consumes 33389 KB memory which is 27.32% and 16.66% lesser when compared to existing methods respectively for Apache web server log dataset samples (Refer Table 4.5).

A comparison of existing methods with EFAA reveals that there is significant improvement in performance in terms of precision, recall, F-measure, accuracy and memory.

4.6 CHAPTER SUMMARY

In this chapter, Enhanced Fuzzy Apriori Algorithm (EFAA) is proposed for solving the frequent itemset generation problem. Experimentation results shows that proposed EFAA produces significant improvement. Frequent itemset mining task is improved by using two techniques, which are Fuzzy Intersection Pruning (FIP) and Fuzzy Automated Support Confidence Pruning (FASCP). Here, FIP is applied for itemset pruning to enhance the accuracy of frequent itemset mining results. FASCP technique arranges the rules in ascending order of support and confidence thresholds which is found from FAS method for association rules generation. Frequent itemset are pruned using FIP to enhance pattern discovery results which reduces the candidate items in the frequent itemset mining. FASCP technique is used to reduce unimportant rules. The experimental results obtained reveals that proposed EFAA outperforms existing techniques in terms of precision, recall, F-measure, accuracy and memory.