AN EFFICIENT STRATEGY OF PREPROCESSING FOR OBTAINING KNOWLEDGE FROM WEB USAGE DATA

Manjula S[1], Rashmi M.J[2] and Varsha.D[3]

[1]Research Scholar, Dept. of Studies in CS, Solapur University, Solapur (India)
[2,3] Student 4th sem MSc Computer Science, Davangere University, Davangere (India)

ABSTRACT

The World Wide Web (WWW) is a collection of huge amount of Web Usage Data. The process of extracting the relevant data from Web Usage Data is known as Web usage mining. This data must be assembled into a consistent and comprehensive view, in order to be used for further steps of the Web Usage Mining. However, often most of this data are not of much interest to most of the users. Due to this abundance, it became essential for finding ways in extracting relevant data from this ocean of data, hence several researches have been done and researchers proposed an significant and unifying area of research is known as Web mining. As most in data mining technique the data preprocessing involves the removing of irrelevant and inconsistent data, but proper data cannot be achieved without implementing proper preprocess techniques. In this paper we are mainly focusing on the complete preprocessing techniques, such as- data fusion, data cleaning, user identification, session identification, data formatting and summarization. These are the activities used to improve the quality of the data by reducing the quantity of data. This methodology will reduce the size of the data from 75% to 85% from its original data size in Web Usage Mining.

Keywords- Data Preprocessing, Data Formatting and Summarization, Session Identification, User Identification, Web Log Data, Web Usage Mining.

I. INTRODUCTION

In the today's world the use of the Web or Internet is increasing enormously with the huge amount of the data. Details available in the web are related to all most all the fields. The key thing is the user must extract only the details which are necessary for his related work. The process of extracting the details or information from the web is known as *"web mining"*. This web mining is the combination of both data mining and World Wide Web. The web mining is categorized into three areas, such as *web content mining, web structure mining and web usage mining*. The web usage mining is relatively independent while compare with the web content mining and web structure mining, but it's not isolated, this web usage mining technique helps in preprocessing of user data, discover the user's pattern and try to predict the user's behavior figure 1 represents the different steps of Web Usage Mining. In this paper we are mainly focusing on one of the steps of the web usage mining, that is

preprocessing technique. The important aim of the preprocessing technique is to remove inconsistent data, redundant data, noisy data where all these data will be present in the web server log files along with useful information, by applying preprocessing techniques we can reducethe size of the web server log file from 75% to 85% of the original size of the web log file, finally the output of preprocessing that is interesting patterns can be transformed to a relational database models.

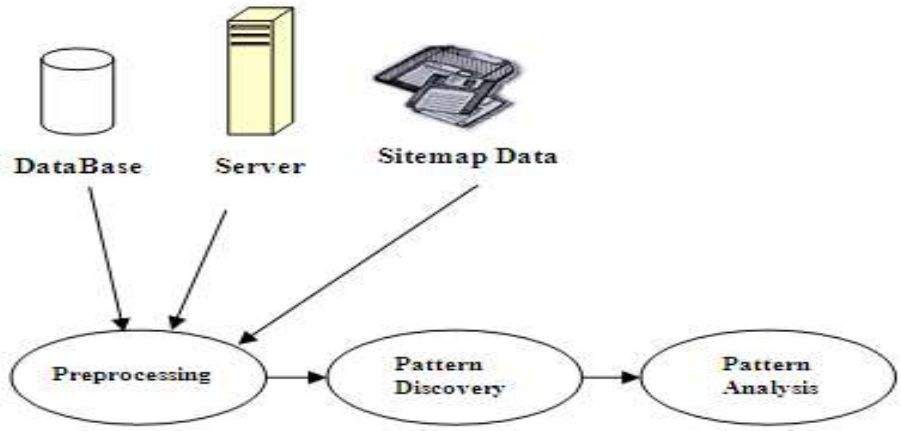


Figure1: General Web Usage Mining Process

II. LITERATURE SURVEY

The data preprocessing is a set of operations that process the available sources of information and lead to the creation of an ad-hoc formatted dataset to be used for knowledge discovery through the application of mining techniques [8]. The main purpose of preprocessing is to produce result that can be used to improve and optimize the content of a site [24].This mining technique will reduces the quantity of data by increasing the quality of data, this will reduce nearly 80% of the original size of the log data[18]. The major steps involved in the preprocessing are data merging, data cleaning, user identification, session identification and data formatting and summarization. The data merging is the technique that combines web log data from multiple sources to retrieve additional information with respect to identification of users and sessions [15].Several papers have different explanation which gives the depth knowledge on this data merging techniques. The data cleaning refers to the removing of irrelevant information which is useless for mining purposes from the HTTP server log file[19]. User identification, is to identify the visitors who access web site and which pages are accessed, it is necessary to distinguish among different users, since user may visit a site more than once [14]. The goal of session identification is to divide the each uses who visit the web site more than once into individual sessions, this can be achieved by using the technique of timeout to break a user's click-steam into session [6]. After the identification of session obtained preprocessed data need to be stored in the relational database module which can be used to properly format the session or transaction for the type of data mining to be accomplished[16].The Data summarization concerns with computation of aggregated variables at different abstraction level[4]. In this paper we are mainly focusing on different treads of preprocessing by providing appropriate algorithm for implementing preprocessing strategy for obtaining knowledge from Web Usage Data.

III. PREPROCESSING

The data preprocessing technique is the set of operations that extract the information from the available source of information which can be used for further steps of Web Usage Mining. The available Web Usage Data is usually consists of irrelevant data, inconsistent data, noise, etc. For example, the user requests for graphical page content and requests for any other file which might be included into a web page or even navigation sessions performed by robots or web spiders which need to be removed, hence this preprocessing strategy have been proposed to extract the required data from the Web Usage Data. Data preprocessing is predominantly significant phase in Web Usage Mining due to characteristics of web data and its association to other related data collected from the multiple source. The data provided by the data source can be used to construct a data model, the web server is the richest source of data, web serverwill collect large amountof data from web sites and these data is stored in the web log files.The web log files are act as input for this preprocessing. These log files are available in web servers.These log files contains the multi-user details in the log file format. The two main log file formats are Common Log Format (CLF) and Extended Common Log Format (ECLF).A typical line in CLF is shown in figure 2.Recently W3C [W3C log] has represented an improved format for Web server log known as Extended Common Log Format (ECLF), this ECLF log file consists of two more fields then the CLF, the referrer (the URL client was visiting before requesting the URL) and user agent (the software that claims to be using) fields. A typical line in ECLF is shown figure3. Both the CLF and ECLF consist of following fields: *IP address:* This represents the address of the client's host name. *Rfc:* It is the remote login name of the user but most of the time it's not available because systems are usually identified by IP address, if remote login name is available then it can be used (here it's not available hence minus sign has been mentioned). *Authorized User:* This is available only when the WWW documents or pages are password protective or if any web site has restricted for the public use then the person who wish to access this sites then the authorization name and password will be provided for those visitors. *Date and time:* This provides the details of the site visitors on what date & time visitor access this sites. *Request:* The details of the website which browser has requested from the website. *Status:* The request status code will be provided so that it will identify whether the requested page or document is ok to access or its under development, this can be achieved by representing code numberin the log file format (Status codes uses in the log files are: 200 OK, 201 Created, 202 Accepted, 400 Bad request, 404 Not found, 403 Forbidden, 204 No Content, 401 Unauthorized, 304 Not modified, 301 Moved Permanently etc…,). *Byte:* Requested web page size in the bytes format. *Referrer:* This represents which is the URL of the Web page containing hyperlink that will display the current document. *User agent:* This will help to find the browser and operating systems which is used for this request to be displayed.These web log files will be the input for the preprocessing strategy; Preprocessing involves Data fusion, Data cleaning, User identification, Session identification, Data Formatting and Summarization.

The goal of preprocessing is to transform the raw data contained in database of web log file into a transaction log file. The data preprocessing presents a number of unique challenges which lead to a variety of algorithm and heuristic techniques for preprocessing tasks. Figure 4represents the different steps of preprocessing.

<ip_addr><base_url><date>
<method><file><protocol>
<code><bytes>

Figure 2: Common Log Format (CLF)

<ip_addr><base_url><date><method>
<file><protocol><code><bytes>
<referrer>

Figure 3: Extended Common Log Format (ECLF)

3.1 Data fusion

Data fusion refers to the use of techniques that combines data from multiple sources to retrieve additional information with respect to identification of user and session, than if they are retrieved from a single data source.

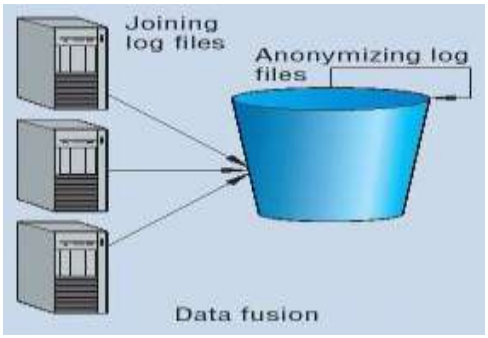


Figure5: Data fusion representation

The different log files are put together and forms a single log file. There is a slight improvisation technique. In this process as it is not only a combining of log files, at the same time it will sort all the entries with respect to time stamp, so it reduces the time and resources to apply and implement separate steps for the sorting of this huge of fusion data of different log files. Figure 5 represents the data fusion method. The fusion and sorting problem is formulated as "The set of log files are provided that is considered as $L_f = \{L_1, L_2 ... L_n\}$ are combined into single log file L". The data variable required to store the values or an array [12]. The data fusion technique can be implemented with the help of algorithm given in figure 6.

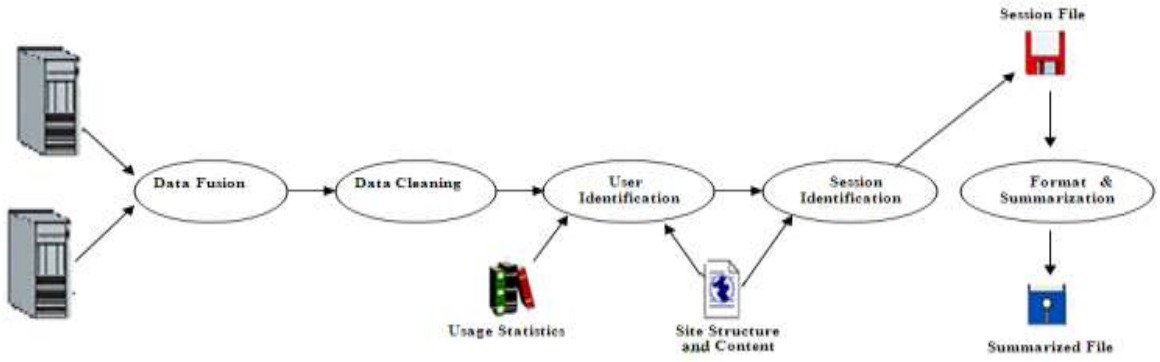


Figure 4: General steps for Preprocessing

Algorithm for Data fusion with different web server log at Internet Service Provider (ISP)

```
//Input: different log file generated by proxy servers.
//Output: single file containing combined output in timestamp order.
For (each log file input) do
        Read log file one by one; do until end of the file reached.
        Read record one by one from current log and put into output file.
        Increment the pointer to point the next record in current log file.
    Increment the pointer in output file to store the next record.
End of for loop.
Sort the output log file entries in ascending order by access time.
Return output.
```

Figure 6: Data fusion algorithm

3.2 Data cleaning

The process of filtering or removing irrelevant item, noise, inconsistent data which is stored in the log files that may not be useful for further web usage mining process, is known as data cleaning. The object of this is to obtain only usage data file request that the user did not explicitly request can be eliminated, this technique also removes the spider navigations requests of web robots. The log entry files such as *.gif, *.jpeg, *.?, *.GIF, *.JPEG, etc. can be removed [19]. Invalid requests from the proxy log files that refer to either internal server error with code and server sides, errors are also removed from the proxy server log files. This technique can be implemented by using the algorithm of data cleaning represented in figure 7. Consider an example for data cleaning, the table I containing all the details of the log file with different extension, by using data cleaning algorithm we need to remove all the irrelevant data from this log file. If the status code of the requested page is greater than 299 or less than 200 then those records need to be deleted, or if the requested method is not in {GET, POST} status then those records need to be removed from the log files. The outcome obtained from this is presented in table II.

Algorithm for Data Cleaning.

```
//Input: Raw log file generated by the proxy server.
//Output: Preprocessed log file with relevant entries consists of tuple (timestamp, IP, URL).
//Constraint: Log file stored in text file must transform into database for further processing.
Read record in database for each record in database.
Read files URL field in proxy server log; the requested objects is the URL field.
If requested URL field contains or end with substring {*.jpeg, *.gif, *.css, *.?}
        Then remove the records.
Else if response code is >299 or < 200
        Then remove records.
Else if requested method not in {GET, POST} status
        Then remove records.
Save record in output.
End of if.
```

Figure 7: Algorithm for Data Cleaning

No	Object Type	Unique Users	Requests	Bytes In	% of Total Bytes In
1	*.gif	1	46	89.00 KB	0.50%
2	*.js	1	37	753.95 KB	4.40%
3	*.aspx	1	34	397.05 KB	2.30%
4	*.png	1	31	137.67 KB	0.80%
5	*.jpg	1	20	224.72 KB	1.30%
6	Unknown	1	15	15.60 KB	0.10%
7	*.ashx	1	15	104.79 KB	0.60%
8	*.axd	1	13	274.81 KB	1.60%
9	*.css	1	8	71.78 KB	0.40%
10	*.dll	1	7	26.41 KB	0.20%
11	*.asp	1	4	1.26 KB	0.00%
12	*.html	1	3	2.17 KB	0.00%
13	*.htm	1	2	69.87 KB	0.40%
14	*.pli	1	2	24.92 KB	0.10%

Table I: Web Log File with different extension before implementing data cleaning algorithm.

No	Object Type	Unique Users	Requests	Bytes In	% of Total Bytes In
2	*.js	1	37	753.95 KB	4.40%
3	*.aspx	1	34	397.05 KB	2.30%
4	*.png	1	31	137.67 KB	0.80%
5	*.jpg	1	20	224.72 KB	1.30%
6	Unknown	1	15	15.60 KB	0.10%
7	*.ashx	1	15	104.79 KB	0.60%
8	*.axd	1	13	274.81 KB	1.60%
10	*.dll	1	7	26.41 KB	0.20%
11	*.asp	1	4	1.26 KB	0.00%
12	*.html	1	3	2.17 KB	0.00%
13	*.htm	1	2	69.87 KB	0.40 %
14	*.pli	1	2	24.92 KB	0.10%

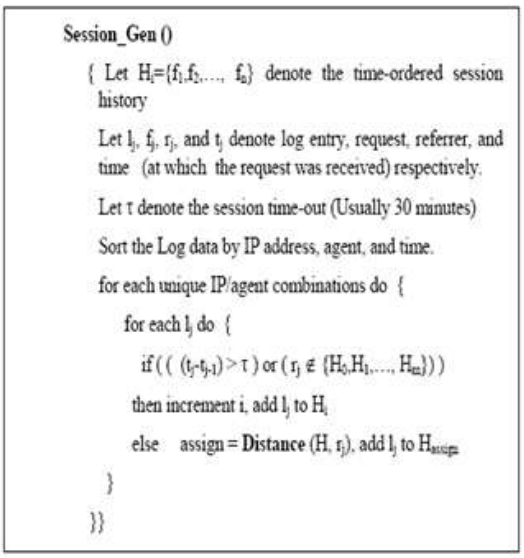
Table II: Web Log File with different extension after implementing data cleaning algorithm.

3.3 User identification

The main task of user identification is to identify the user who access web site and which pages are accessed. This deals with associating page reference with different users; this reduces network traffic and improves performance. In most of the cases web log file provides only the computer IP address and user agent (if web log is using ECLF), then users are identified with the IP address. Some web site requires user registration for the user identification purpose, but due to privacy reason many users prefer not to browse those sites which require registration and logins [19]. Hence in most of the cases user is identified by the IP address of the computer, but different users can use the same IP address for browsing purpose, in this case we can distinguish the user by the software tool or by the operating system used by the user.

3.4 Session identification

The goal of session identification is to divide thepage accesses of user into individual sessions; at present the methods to identify user sessions include timeout mechanism and maximal forward reference. A user session is a directed list of page accesses performed by an individual user during the visit in a web site [4]. For the identification of user session the timeout has been fixed, if the time between page requests exceeds a giving time limit then it is assumed that the user is starting new session. A user may have a single (or multiple) session(s) during a period of time. Thus the session identification problem is formulated as "Given the Web log file *Log*, capture the web users navigation trends, typically expressed in the form of Web users sessions". In this paper we have provided the "timeout threshold" to define the user's sessions. Figure 8 represents Session identification algorithm.



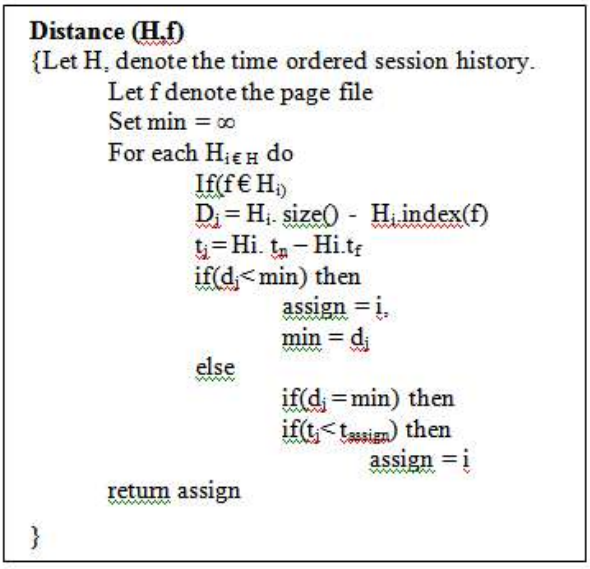


Figure 8: Algorithm for Session Identification Figure 9: Algorithm for Distance function

The session identification algorithm checks to see the session time-out or if the referring file is not present in any of the open session histories. If so, a new session is opened. Since the Log has already been stored by IP address/Agent, all of the open session histories are potential candidates for the page file access being processed. 'Session_Gen' algorithm calls the 'Distance' function finds the history that most recently accessed f. The 'Distance' function shown in figure 9 given a list of histories and page file f. Finally, if the times are equal, by default, the access is assigned to the history with the lower index. A random assignment could be used to break ties. The index of the history that f should be assigned to is returned by the 'Distance' function.

3.5 Data Formatting and summarization:

It is the terminal step of preprocessing technique,the preprocessed log entries are represented using the final preparation module. This can be used to properly format the obtained preprocessed session or transaction file for the different instance of the web usage mining techniques.The data generalization method is applied at the request level and aggregated for visits and user sessions to completely fill in the database [20]. The data summarization concerns with the computation of aggregated variables at different abstraction levels; these aggregated variables are later used in the data mining step. Table III represents the sample database in session table. This formatted database table is very important for further steps like clustering, pattern discovery, by using this table one can easily come to know about user's session details with IP address and URL accessed by this we can generate the matrix representation for the clustering purpose.

Session Id	IP Address	Date &Time	URL Accessed
1	120.33.med.umich.edu	1995-08-03 22:42:31	/history/apollo/apollo-13/apollo-13.html
1	120.33.med.umich.edu	1995-08-03 22:43:03	/facilities/lc39a.html
1	120.33.med.umich.edu	1995-08-03 22:43:42	/facilities/mlp.html
1	120.33.med.umich.edu	1995-08-03 22:45:01	/facilities/tour.html
2	128.101.144.178	1995-08-03 23:15:10	/shuttle/missions/sts-69/mission-sts-69.html
3	128.102.143.201	1995-08-04 03:15:22	/shuttle/missions/sts-64/mission-sts-64.html
4	128.102.143.212	1995-08-04 03:11:51	/shuttle/missions/sts-69/mission-sts-69.html
4	128.102.143.212	1995-08-04 03:13:15	/facilities/vab.html
4	128.102.143.212	1995-08-04 03:14:28	/shuttle/missions/sts-71/mission-sts-71.html
4	128.102.143.212	1995-08-04 03:15:11	/shuttle/missions/missions.html
4	128.102.143.212	1995-08-04 03:15:18	/shuttle/missions/sts-70/mission-sts-70.html
4	128.102.143.212	1995-08-04 03:16:35	/shuttle/missions/sts-72/mission-sts-72.html
4	128.102.143.212	1995-08-04 03:17:10	/shuttle/missions/sts-72/sts-72-info.html
4	128.102.143.212	1995-08-04 03:17:24	/ksc.html
5	128.102.202.133	1995-08-03 23:13:02	/shuttle/missions/missions.html
5	128.102.202.133	1995-08-03 23:15:30	/shuttle/missions/missions.html
5	128.102.202.133	1995-08-03 23:15:51	/shuttle/missions/missions.html
5	128.102.202.133	1995-08-03 23:16:02	/shuttle/missions/missions.html
6	128.102.236.36	1995-08-03 23:16:45	/history/apollo/apollo-13/apollo-13.html
7	128.102.86.216	1995-08-04 02:23:24	/shuttle/missions/sts-70/mission-sts-70.html

Table III: Sample database in session

IV. IMPLEMENTATION AND RESULTS

Preprocessing methodology has been implemented on the web log files which is downloaded from the NASA website, this is the input for our experiment, these log files which consist of all the irrelevant data, inconsistent data, web spiders, robotic movements which always invoke automatically at the time of browsing hence preprocessing techniques is implemented on these log files which reduces the size of the log file from 75% to 85% of its original size of web log file. We have used JAVA for the implementation of preprocessing methodology. The main part of the implementation is file control which is represented in figure 10; this code segment represents the access control to perform the preprocessing strategy. Before implementing preprocessing, first we need to check whether the requested file name is present in the log files, this code helps

to check the requested file exists or not, if it exists then it will allow to read the file, otherwise it will reflects the error message. Figure 11 represents code segment for data fusion, it will check the size of the log file and it will add the items (if multiple log files are available) for further steps of the preprocessing. After merging the log file data cleaning method will be implemented, here it checks the status of the log file, if file contains the inconsistent data then it will be removed and cleaned data will be stored in the array. Figure 12 represents code segment for data cleaning. Figure 13 represents session identification code segment, in this "TimeOut" is fixed, based on the timeout session will be generated. Figure 14 represents the snapshot of the results which we obtained after the implementation. Table IV represents the result after the implementation of preprocessing methodology; here the size of the log file is reduced from 75% to 85% of its original size. The preprocessed data which is obtained from this implementation can be further used for the pattern discovery purpose in Web Usage Mining.

```
Public static boolean checkFileName(String fName)
{File fileName= new File(fName);
 If(fileName.exists())
 {          if(fileName.isFile())
            {          if(fileName.canRead())
                                    return true;
                       else
                       JOptionPane.showMessageDialog(null,"WUM:Error!");
            }else
            JOptionPane.showMessageDialog(null,"WUM:Error!Specify location");
 }else
 JOptionPane.showMessageDialog(null,"WUM:Error!Specify location");
 return(false);
}
```

Figure 10: JAVA code for File Control

```
Public int addItem(String item)
{          int size;
           If(items.size()=0 || !tems.contains((String)item))
           {          size=itme.size();
                      ItemRef newItemRef = new ItemRef();
                      newItemRef = new ItemRef();
                      newItemRef.setItemID(size+"");
                      newItemRef.setItemName(item);
                      itemsRef.add(newItemRef);
                      items.add((String)item);
           }else
           Size = item.indexOf((String)item);
           return size;
`
```

Figure 11: JAVA code for Data Fusion

```
Public LinkedList removeInfrequentItemsets(LinkedList itemsFreq.int minSup)
{       ListIterator itItemsFreq = itemsFreq.listIterator(0);
        ItemSet newItemsetElements;
        LinkedList indexesToDelete = new LinkedList();
        While(itItemsFreq.hasNext())
        {       newItemsetElements = (ItemSet)itItemsFreq.next();
                If(newItemsetElements.getSupport() < minSup)
        IndexToDelete.add((newInteger(itemsFreq.indexOf(newItem        setElements))))
        )for(int count = indexes To Delete.size()-1; count >=0; count--)
        {       itemsFreq.remove(((Integer)indexes ToDelete.get(count).intValue());
        }return itemsFreq;
}
```

Figure 12: JAVA code for Data Cleaning

```
Public int accessLogProcessor(JTextArea pResultArea)
{int k=statValues.accessData.size();
 While(k>0)
 {      temp = (AccessLogData)statValues.accessData.get(k-1);
        /* fist check whether the IP address is same or not
         * if yes, check whether the user agent and referer is matched.
         * if yes, add this to the session
         * else look for other access records in queue within the session TimeOut*/
  If(temp.ip.equals(result[0]) && (!temp.identifySession(x, session TimeOut))
&&(temp.compareAgentReferer(result[11], result[10])))
                        { temp.addURL(result[6]); ////
                            /* replace the previous datetime*/
                          Temp.setDateTime(x);
                          Flag = true;
                          Break;
                        } k--;
        } /* end of while*/
   }
```

Figure 13: JAVA code for Session Identification

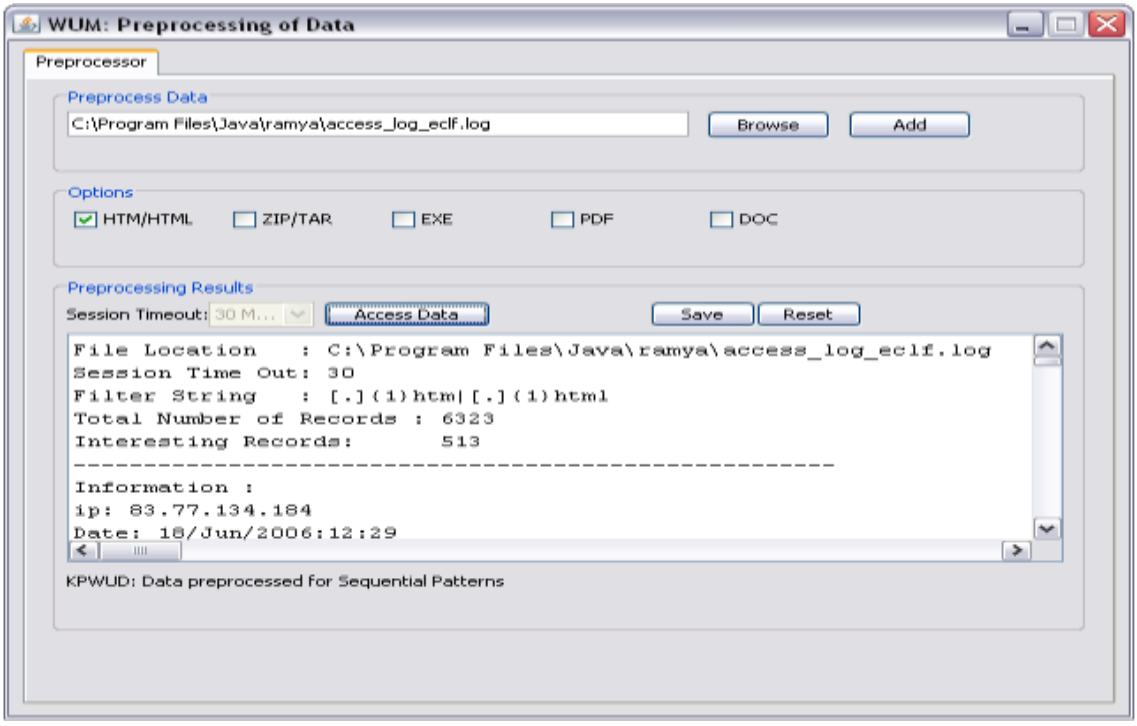
V. CONCLUSIONS

The data collection from different log file will contain all the details about the pages visited by the user during his interaction with the web, along with this log files also contain all the inconsistent, noisy, irrelevant data, hence it is necessary that these log files need to be preprocessed in the précised manner. For the implementation purpose we have downloaded NASA web log file. The preprocessing methodology, such as - data fusion, data cleaning, user identification, session identification, data formatting and summarization steps are implemented on that to obtain the interesting patterns, Table IV represents the result after the preprocessing of data. By this we have proved that this implementation technique reduces the size of the log file from 75% to 85% of its original size, hence it is very efficient strategy of preprocessing for obtaining knowledge from Web Usage Data.

VI. FUTURE WORK

The future work involves getting the solution for the various issues which usually raises at the time of data collection tasks, data transformation tasks and user identification. This also involves various data transformation tasks that are likely to influence the quality of the preprocessed data resulting from the different strategy of preprocessed, this preprocessed data can be used for the pattern discovery which can be further used for various web usage application which as site improvements, business intelligence and recommendation.

Figure 14: Snapshot showing Interesting web patterns after preprocessing.



Log File Format	Total number of records	Interesting records	No. of Session Identified	Original log file size	Preprocessed log file size
CLF	5166	2224	1218	419 KB	109 KB
ECLF	6323	513	358	1668 KB	55 KB

Table IV: Results after preprocessing obtained from NASA Log File

REFERENCES

[1]. Configuration files of W3C http. http:/www.w3.org/Daemou/User/Config/ (1995).

[2].W3C Extended Log File Format, http:/WWW.W3.org/TR/WD-logfile.html (1996).

[3]. Yan Wang: Web Mining and Knowledge Discovery of Usage Patterns, CS 748T Project (Part I). Feb-2000.

[4]. Ramya C, Dr. Shreedhara K. S and Kavitha G: Preprocessing: A Prerequisite for Discovering Patterns in Web Usage Mining, 2011.

[5]. J. Srivastava, R. Cooley, M. Deshpande, P-N. Tan: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, SIGKDD Explorations, Jan-2000.

[6]. Hui Yu, Zhongim Lu: Analysis of Web Usage Mining.

[7]. Pronalazenje Skrivenog Znanja, Bojan Furla: Data Preprocessing.

[8]. Mathias Gery, Hatem Haddad: Evaluation of Web Usage Mining Approaches for User's Next Request Prediction, Nov-2003.

[9]. Haugua Dai, Bamshad Mobasher: Integrating Semantic Knowledge with Web Usage Mining for Personalization.

[10].Li Chaofeng: Research and Development of Data Preprocessing in Web Usage Mining.

[11]. Aniket Dash: Web Usage Mining: An Implementation.

[12]. Mr. Sanjay Bapu Thakare, Prof. Sangram.Z.Gawali: A Effective and Complete Preprocessing for Web Usage Mining. Vol. 02, No. 03, 2010, 848-851.

[13]. V. Sathiyamoorthi, Dr. V. Murali Bhaskaran: Data Preparation Techniques for Web Usage Mining in World Wide Web- An Approach. Vol.2, No. 4, Nov-2009.

[14]. V.V.R Maheswara Rao and Dr. V. Valli Kumari: An Enhanced Preprocessing Research Framework For Web Log Data Using A Learning Algorithm.DOI:10:5121/csif2011.1101.

[15]. C.P Sumathi, R.Padmaja Valli, and T. Santhanam: An Overview of Preprocessing of Web Log Files for Web Usage Mining. ISSN 1992-8645, E-ISSN 1817-3195.

[16]. R. Cooley, B. Mobasher and J. Srivastava: Data Preparation for Mining World Wide Web Browsing Patterns HER-9554517.

[17]. V. Chitraa, Dr. Antony Selvadoss Thanamani, A Novel Technique for Session Identification in Web Usage Mining Preprocessing. Vol. 34, No.09, Nov—2011.

[18]. Navin Kumar Tyage, Ak. Solonki and Manoj Wadhwa: Analysis of Server Log by Web Usage Mining for Website Improvement. Vol. 07, Issue4, No. 08, July-2010.

[19]. V. Chitraa, Dr. Antony Selvdoss Dawamani: A Survey on Preprocessing Method for Web Usage Data Vol. 07, No.03, 2010.

[20]. Zhiqiang Zheng, Balaji Padmanabhan, Steven O. Kimbrough: On the Existence and Significance of Data Preprocessing Biases in Web Usage Mining. Vol. 15, No. 02, Spring 2003.

[21]. V. Sathiyamoorthi and Dr. Murali Bhaskaran: Data Preprocessing Techniques for Pre-Fetching and Caching of Web Data through Proxy Server. Vol. 11, No. 11, Nov-2011.

[22]. Lukas Cenovsky: web Usage Mining on is.muni.c2 Master's Thesis.

[23]. Suneetha K.R, Dr. R. Krishnamoorthi: Data Preprocessing and Easy Access Retrieval of Data through Data ware House. Oct-2009, ISBN: 978-988-17012-6-8.

[24]Mohd Helmy Abd Wahab, Mohd Nozali Haji Mohd, Hafizul Fahri Hanafi, Mohamad Farshan, Mohamad Mohsin: Data Preprocessing of Web Server Logs for Generalized Association Rules Mining Algorithm-2005.

[25]. Jiawei Han and Micheline Kamber: Data Mining Concepts and Techniques. ISBN-81-8147-049-4