An Efficient Algorithm for Data Cleaning of Log File using File Extensions

Surbhi Anand
Department of Computer Science & Engineering
Thapar University, Patiala-147004 (India)

Rinkle Rani Aggarwal
Department of Computer Science & Engineering
Thapar University, Patiala-147004 (India)

ABSTRACT
World Wide Web is a monolithic repository of web pages that provides the Internet users with heaps of information. With the growth in number and complexity of Websites, the size of web has become massively large. Web Usage Mining is a division of web mining that involves application of mining techniques to web server logs in order to extract the behavior of users. A Web Usage Mining process comprises of three phases: data preprocessing, patterns discovery and pattern analysis. Data preprocessing tasks are carried out former to the application of mining algorithms. Preprocessing enables to translate the unprocessed data which is composed from server log files into constructive data abstraction. The appropriate analysis of a web server log proves to be beneficiary to manage the websites efficiently from the administrative and users' prospective. Preprocessing results also strongly influences the later phases of Web Usage Mining. This makes the preprocessing of server log files a significant step in Web Usage Mining. This paper emphasizes on the Web Usage Mining process and makes an exploration in the field of data cleaning.

General Terms
Web Mining, Web Usage Mining.

Keywords
World Wide Web, Preprocessing, Web Usage Mining, Data Cleaning, Web Server logs.

1. INTRODUCTION
With the brisk growth of the World Wide Web, the web has become an imperative medium of information dissemination. Therefore, the information available on the Web has become a vital source of information for the users of the internet. Due to these reasons there is an increase in the number and size of websites available on internet and makes the World Wide Web remarkably gigantic. The growth in the mass of data present on web has engrossed the attention of the scholars and researchers towards the application of data mining techniques on the data available on the web in order to extract useful information.

When data mining techniques are applied to Web data, it is referred to as Web mining. In 1996 it's Etzioni was first to coin the term web mining [1]. According to him Web mining is the use of data mining techniques to automatically extract information from World Wide Web documents and web services.

The aim of Web Mining is to discover constructive information from the hyperlink structure of web pages, webpage content, and web usage data. Though web mining takes its foundations from the data mining techniques, but it does not solely depends on data mining. The reason is that data mining is applied on structured data while web mining is applied on web data which is heterogeneous and unstructured or semi structured in nature.

Depending on the principal types of data used in the mining, web mining can be classified into three areas: Web content mining, Web structure mining and Web usage mining. Web Content Mining extracts useful information or knowledge from the contents of Web pages. Web structure mining discovers useful knowledge from hyperlinks that portrays the structure of the Web. Web Usage Mining refers to the discovery of user access patterns from Web server's logs, which keeps record of every click made by each user. In general, Web Usage Mining consists of three processes: data preprocessing, patterns discovery and patterns analysis.

The objective of this paper is to provide a review of web usage mining and an approach for performing data cleaning. This paper is organized as follows: Section 2 gives an overview of web usage mining. In Section 3 steps in preprocessing phase are described. In section 4 algorithms for separating the web log entries and data cleaning are described. Section 5 and section 6 consists of experimental results and result analysis.

2. WEB USAGE MINING
Web usage mining is the application of data mining techniques on large web log repositories in order to extract useful knowledge about user's behavioral patterns. The primary data source in case of web usage mining is a web server log (or web access log). A Web server log is a textual file, independent of server platform, in which a Web server enters a record whenever a user requests for a resource. Analyzing the web access logs of different web sites can help understand the user behavior and the web structure, thereby improving the design of this colossal collection of resources [2]. A sample log entry is shown below in fig 1.

213.135.131.79 - - [15/May/2002:19:21:49 -0400] "GET /features.htm HTTP/1.1" 200 9955

Fig 1: A Sample Log Entry

The process of Web Usage Mining consists of three main steps (see Figure 2) [3]: Data Preprocessing, Pattern Discovery and Pattern Analysis.

a) Data Preprocessing: In this phase, a series of processing tasks are applied on web log file such as data cleaning, user identification, session identification, path completion and transaction identification.

b) Pattern Discovery: In this phase, techniques from several research areas, such as data mining, machine learning, statistics, and pattern recognition are examined

to be applied on data obtained after preprocessing in order to generate indentify meaningful patterns.

c) Pattern Analysis: In this phase, uninteresting patterns are removed from the patterns identified during pattern discovery phase. There are two most common approaches for the pattern analysis: SQL query mechanism and constructing multi-dimensional data cube to perform OLAP operations [4].

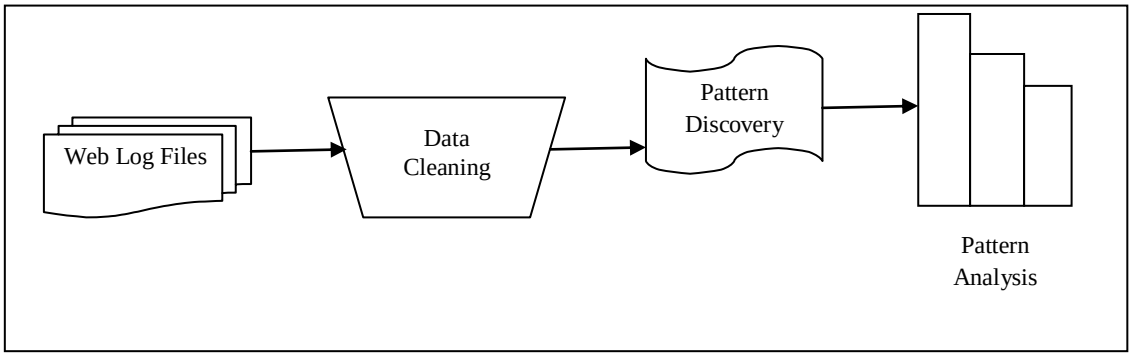


Fig 2: Phases of Web Usage Mining

3. DATA PREPROCESSING

The information that can be accessed through web is heterogeneous and semi structured or unstructured in nature. Due this heterogeneity a web log file may consists of some undesirable log entries, whose presence does not matters from the web usage mining point of view. This makes the preprocessing of log file an important precondition for discovering the knowledgeable patterns. The goal of preprocessing is to transform the raw click stream data into a set of user profiles [5]. Data preprocessing presents a number of unique challenges which led to a variety of algorithms and heuristic techniques for preprocessing tasks such as merging and cleaning, user and session identification etc [6].

Data preprocessing is one of the most complex phase of the Web Usage Mining process.

Data preprocessing consists of four sub-phases (see figure 3):- data cleaning, user identification, session identification, path completion.

3.1 Data Cleaning

With the important entries, a web log file may consist of certain undesirable rather useless data which has nothing to do with the mining procedure. Therefore, it is imperative to remove those irrelevant entries from the log file. There are three kinds of irrelevant or redundant data needed to clean: accessorial resources embedded in HTML file, robots' requests and error requests [7].

Accessorial resources are nothing but graphics, scripts or some videos, that may be present in an html page which may have no relationship with the content of the html page on which they are embedded. Rather they may be part of some advertisement. When a user requests for a particular webpage, these may also get downloaded along with the HTML file and forms several log entries. As discussed earlier the objective of Web Usage Mining is to capture the user's behavior; therefore these entries for graphics, images and scripts are useless. Due to this reason the removal of these irrelevant items seems to be essential. Web robots are software tools that are used to automatically extract contents of a website by following all the hyperlinks from a Web page. Search engines such as Google periodically use spiders to grab all the pages from a Web site to update their search indexes [8].These are also not important from the mining perspective and hence must be removed.

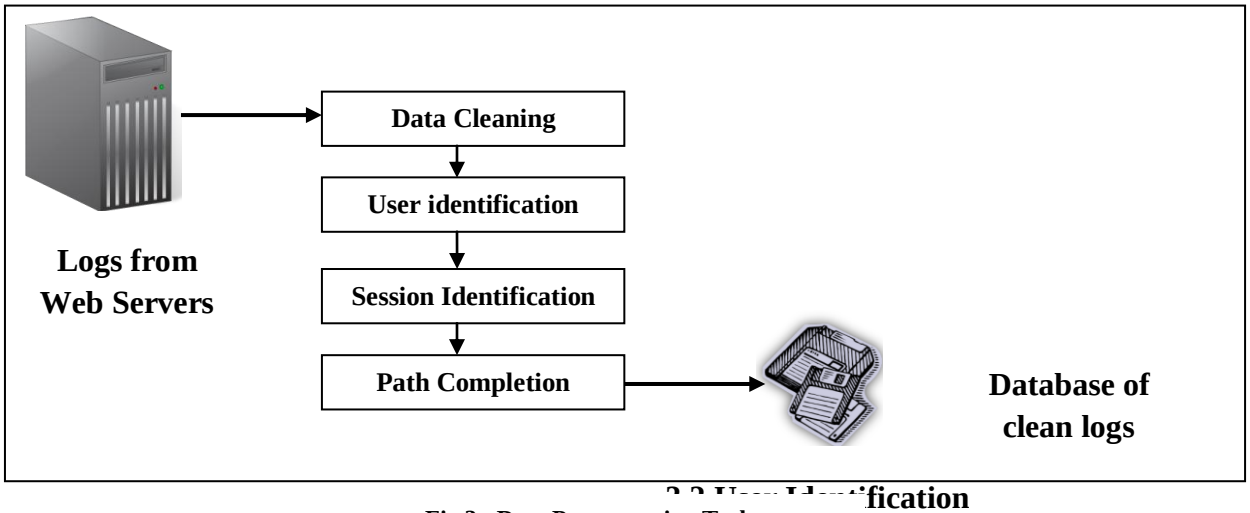


Fig 3: Data Preprocessing Tasks

3.2 User Identification

...t users are identified, who contact web server, requesting for some resource on the web. Different

methods are suggested for user identification. The simplest one is to assign different user id to different IP address. During user identification, problem due to caching may occur. Caching problem can be rectified by assigning a short expiration time to HTML pages enforcing the browser to retrieve every page from the server [9].

3.3 Session Identification

After the identification of each user, individual user's sessions are made. The simplest method for identifying the session uses a timeout mechanism. The significance of timeout method is that if the time between page requests exceeds a certain limit, signifies user is beginning a new session.

3.4 Path Completion

There are chances of missing pages after constructing transactions due to proxy servers and caching problems [10][11]. In such a condition it becomes necessitate identifying the user's access path, and adding the missing paths.

4. PROPOSED WORK

4.1 Data Field Extraction

A server log file consists of various data fields that should be separated before applying cleaning procedure. This process of separating out different data fields from single server log entry is identified as data field extraction.

A server uses different characters such as a comma or a space character which works as separators. The algorithm proposed below for data field extraction uses the space character as a separator to separate the fields of the log file.

A portion of the log file used for the experimentation has been shown in the following figure (see figure 3).

```
129.173.67.107 - - [23/Feb/2004:14:22:01 -
0500] "GET
/~ai04/_derived/sponsors.htm_cmp_glacier1
10_vbtn.gif HTTP/1.1" 304 0

29.173.67.107 - - [23/Feb/2004:14:22:05 -
0500] "GET /~ai04/submit/ HTTP/1.1" 304 0

129.173.67.107 - - [23/Feb/2004:14:22:05 -
0500] "GET /incoming/cyberstyle.css
HTTP/1.1" 404 2231

137.207.216.174 - - [23/Feb/2004:14:22:10 -
0500] "GET
/~janyst/chat/chatAppletXML.php?id=20
HTTP/1.1" 200 34
```

Fig 4: Sample Log File

The implementation of the algorithm is done in Java programming language. It makes use of some of Java's inbuilt classes and methods. The data field extraction is done using methods of String class. It is assumed that space character is acting as the separator. The log file is read character by character up to the end and then by using the methods of String Tokenizer class the data fields are broken into tokens and save din an array.

Table 1. Algorithm for Data Field Extraction

Algorithm 1: Field Extraction

Input: Log File

1. Initialize token := null /* token is variable that will contain the items read from the log file */
2. Initialize retTokArr := null /* retTokArr is an array that will contain the items read from the log file after separation */
3. Find location of the file to be read
4. Open file for reading.
5. token := readFile() /* Read items from the log file character by character in the form of string */
6. while(token != null) /* Run a while loop until all the items are not read from the file */

   retTokArr := token.split(" ")/* Use space as a delimiter to separate out the fields */
7. Close the file

Output: Separated log fields

4.2 Data Storage

The second algorithm (see Table 2) describes the storage of field extracted from the log file using the first algorithm. Before data storage we need to create the table named as log table in which each entry from the original log file is stored.

The sample SQL query for creating the log table with the column names and data types is shown in Fig 5.

```
CREATE TABLE LOGTABLE
(
        IPADDRESS VARCHAR2 (50),
        HOSTNAME VARCHAR2 (50),
        USER_NAME VARCHAR2 (50),
        TIME_STAMP VARCHAR2 (50),
        OFFSET VARCHAR2 (50),
        METHOD VARCHAR2 (50),
        PATH VARCHAR2 (250),
        PROTOCOL VARCHAR2 (50),
        STATUS VARCHAR2 (50),
        BYTES VARCHAR2 (50)
)
```

Fig 5: SQL Query for table creation

Table 2. Algorithm for Data Storage

Algorithm 2: Data Storage

Input: Results of Algorithm 1

Output: Log table

1. Open a database connection and create a statement object.
2. Create a table to store the log data.
3. Add field names to the log table.
4. Insert the items to the appropriate field column into the log table.
5. Close the database.

4.3 Data Cleaning

The third algorithm (see Table 3) shows the data cleaning. This algorithm retains only those data entries in the log file whose status code is 200, method is GET and file type is except from gif, jpg and css.

Table 3. Algorithm for Data Cleaning

Algorithm 1: Data Cleaning

Input: Log table

1. Declare filename, method, ip_address, file_extension, hostname, username, timestamp, offset, protocol, bytes, status_code.
2. Open a database connection.
3. Create an object of PreparedStatement to read each record in log table.
4. For each record read from the log table
   a. Read status_code /* the status as extracted from the database.*/
   b. Read method /* method as extracted from the database.
   c. If(status_code =200 and method = GET){
      i. Read ip_address, hostname, username, timestamp, offset, protocol, bytes, and path.
      ii. Extract file_extension from path.
      iii. If file_extension != {.gif, .jpg,.css}
           Insert data entries into summarized logtable.
      iv. Else
           Remove data entries.
5. Close connection
6. End

Output: Summarized log table

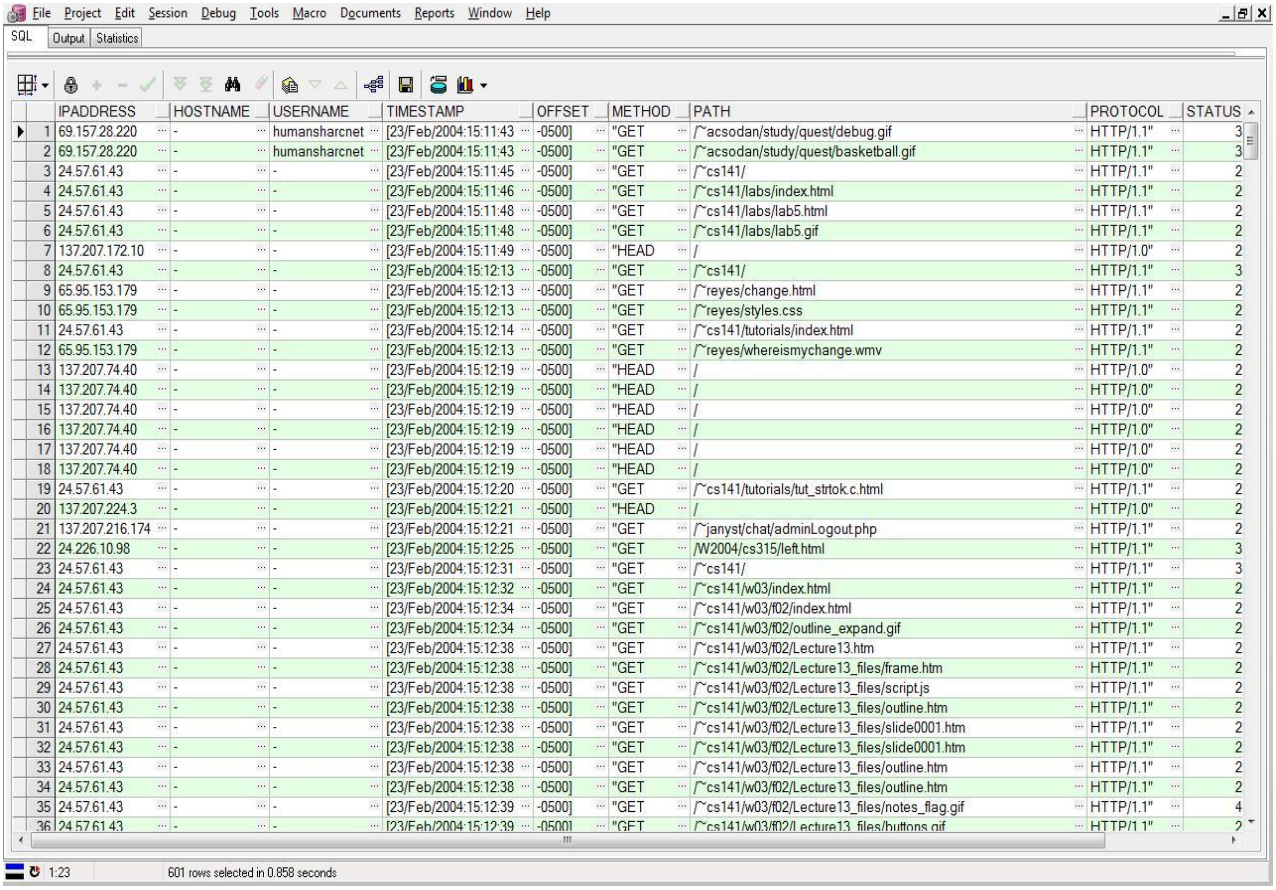
5. EXPERIMENTAL RESULTS

The log file used for the work was of size 60 KB and consists of 601 entries. But after data cleaning the size of the file is reduced to 15.6 KB.

After data cleaning only 150 entries are left in the log. The results are shown through the snapshots as shown in figure 5 and figure 6.

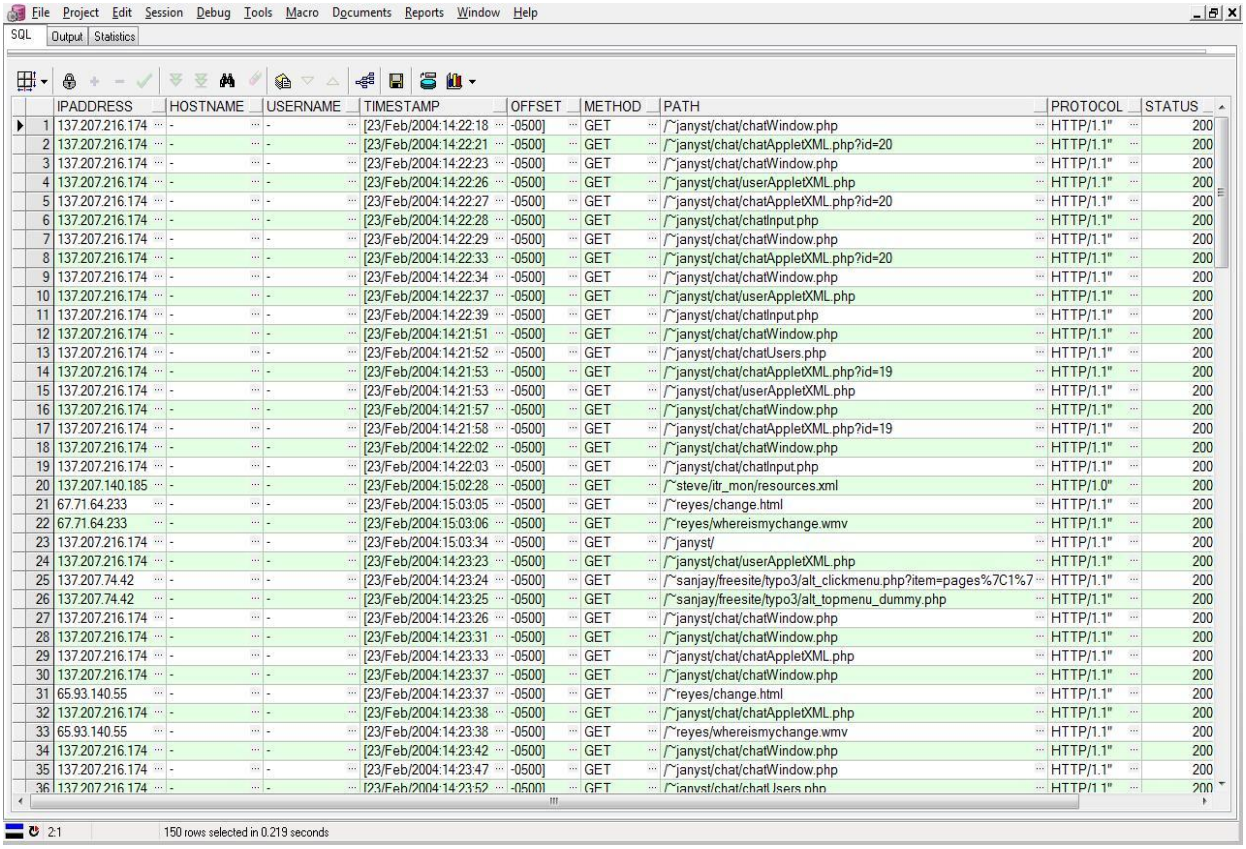
Figure 5 shows the log table formed by separating the data fields of a log file. In this figure we can see that we entries for gif, jpg and css files but as discussed in previous sections, these are unnecessary from the Web Usage Mining perspective. Therefore, it is essential to remove those entries. The log table consists of 601 entries.

Figure 6 shows the resulting summarized log table obtained after data cleaning. In this we can see that we are left with 150 entries only. Only these 150 entries are useful for Web Usage Mining.



	IPADDRESS	HOSTNAME	USERNAME	TIMESTAMP	OFFSET	METHOD	PATH	PROTOCOL	STATUS
1	69.157.28.220	-	humansharcnet	[23/Feb/2004:15:11:43	-0500]	"GET	/~acsodan/study/quest/debug.gif	HTTP/1.1"	3
2	69.157.28.220	-	humansharcnet	[23/Feb/2004:15:11:43	-0500]	"GET	/~acsodan/study/quest/basketball.gif	HTTP/1.1"	3
3	24.57.61.43	-	-	[23/Feb/2004:15:11:45	-0500]	"GET	/~cs141/	HTTP/1.1"	2
4	24.57.61.43	-	-	[23/Feb/2004:15:11:46	-0500]	"GET	/~cs141/labs/index.html	HTTP/1.1"	2
5	24.57.61.43	-	-	[23/Feb/2004:15:11:48	-0500]	"GET	/~cs141/labs/lab5.html	HTTP/1.1"	2
6	24.57.61.43	-	-	[23/Feb/2004:15:11:48	-0500]	"GET	/~cs141/labs/lab5.gif	HTTP/1.1"	2
7	137.207.172.10	-	-	[23/Feb/2004:15:11:49	-0500]	"HEAD	/	HTTP/1.0"	2
8	24.57.61.43	-	-	[23/Feb/2004:15:12:13	-0500]	"GET	/~cs141/	HTTP/1.1"	3
9	65.95.153.179	-	-	[23/Feb/2004:15:12:13	-0500]	"GET	/~reyes/change.html	HTTP/1.1"	2
10	65.95.153.179	-	-	[23/Feb/2004:15:12:13	-0500]	"GET	/~reyes/styles.css	HTTP/1.1"	2
11	24.57.61.43	-	-	[23/Feb/2004:15:12:14	-0500]	"GET	/~cs141/tutorials/index.html	HTTP/1.1"	2
12	65.95.153.179	-	-	[23/Feb/2004:15:12:13	-0500]	"GET	/~reyes/whereismychange.wmv	HTTP/1.1"	2
13	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
14	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
15	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
16	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
17	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
18	137.207.74.40	-	-	[23/Feb/2004:15:12:19	-0500]	"HEAD	/	HTTP/1.0"	2
19	24.57.61.43	-	-	[23/Feb/2004:15:12:20	-0500]	"GET	/~cs141/tutorials/tut_strtok.c.html	HTTP/1.1"	2
20	137.207.224.3	-	-	[23/Feb/2004:15:12:21	-0500]	"HEAD	/	HTTP/1.0"	2
21	137.207.216.174	-	-	[23/Feb/2004:15:12:21	-0500]	"GET	/~janyst/chat/adminLogout.php	HTTP/1.1"	2
22	24.226.10.98	-	-	[23/Feb/2004:15:12:25	-0500]	"GET	/W2004/cs315/left.html	HTTP/1.1"	3
23	24.57.61.43	-	-	[23/Feb/2004:15:12:31	-0500]	"GET	/~cs141/	HTTP/1.1"	3
24	24.57.61.43	-	-	[23/Feb/2004:15:12:32	-0500]	"GET	/~cs141/w03/index.html	HTTP/1.1"	2
25	24.57.61.43	-	-	[23/Feb/2004:15:12:34	-0500]	"GET	/~cs141/w03/f02/index.html	HTTP/1.1"	2
26	24.57.61.43	-	-	[23/Feb/2004:15:12:34	-0500]	"GET	/~cs141/w03/f02/outline_expand.gif	HTTP/1.1"	2
27	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13.htm	HTTP/1.1"	2
28	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/frame.htm	HTTP/1.1"	2
29	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/script.js	HTTP/1.1"	2
30	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/outline.htm	HTTP/1.1"	2
31	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/slide0001.htm	HTTP/1.1"	2
32	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/slide0001.htm	HTTP/1.1"	2
33	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/outline.htm	HTTP/1.1"	2
34	24.57.61.43	-	-	[23/Feb/2004:15:12:38	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/outline.htm	HTTP/1.1"	2
35	24.57.61.43	-	-	[23/Feb/2004:15:12:39	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/notes_flag.gif	HTTP/1.1"	4
36	24.57.61.43	-	-	[23/Feb/2004:15:12:39	-0500]	"GET	/~cs141/w03/f02/Lecture13_files/buttons.gif	HTTP/1.1"	2

Fig 5: Result after extraction of data fields (Log table)



	IPADDRESS	HOSTNAME	USERNAME	TIMESTAMP	OFFSET	METHOD	PATH	PROTOCOL	STATUS
1	137.207.216.174	-	-	[23/Feb/2004:14:22:18	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
2	137.207.216.174	-	-	[23/Feb/2004:14:22:21	-0500]	GET	/~janyst/chat/chatAppletXML.php?id=20	HTTP/1.1"	200
3	137.207.216.174	-	-	[23/Feb/2004:14:22:23	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
4	137.207.216.174	-	-	[23/Feb/2004:14:22:26	-0500]	GET	/~janyst/chat/userAppletXML.php	HTTP/1.1"	200
5	137.207.216.174	-	-	[23/Feb/2004:14:22:27	-0500]	GET	/~janyst/chat/chatAppletXML.php?id=20	HTTP/1.1"	200
6	137.207.216.174	-	-	[23/Feb/2004:14:22:28	-0500]	GET	/~janyst/chat/chatInput.php	HTTP/1.1"	200
7	137.207.216.174	-	-	[23/Feb/2004:14:22:29	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
8	137.207.216.174	-	-	[23/Feb/2004:14:22:33	-0500]	GET	/~janyst/chat/chatAppletXML.php?id=20	HTTP/1.1"	200
9	137.207.216.174	-	-	[23/Feb/2004:14:22:34	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
10	137.207.216.174	-	-	[23/Feb/2004:14:22:37	-0500]	GET	/~janyst/chat/userAppletXML.php	HTTP/1.1"	200
11	137.207.216.174	-	-	[23/Feb/2004:14:22:39	-0500]	GET	/~janyst/chat/chatInput.php	HTTP/1.1"	200
12	137.207.216.174	-	-	[23/Feb/2004:14:21:51	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
13	137.207.216.174	-	-	[23/Feb/2004:14:21:52	-0500]	GET	/~janyst/chat/chatUsers.php	HTTP/1.1"	200
14	137.207.216.174	-	-	[23/Feb/2004:14:21:53	-0500]	GET	/~janyst/chat/chatAppletXML.php?id=19	HTTP/1.1"	200
15	137.207.216.174	-	-	[23/Feb/2004:14:21:53	-0500]	GET	/~janyst/chat/userAppletXML.php	HTTP/1.1"	200
16	137.207.216.174	-	-	[23/Feb/2004:14:21:57	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
17	137.207.216.174	-	-	[23/Feb/2004:14:21:58	-0500]	GET	/~janyst/chat/chatAppletXML.php?id=19	HTTP/1.1"	200
18	137.207.216.174	-	-	[23/Feb/2004:14:22:02	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
19	137.207.216.174	-	-	[23/Feb/2004:14:22:03	-0500]	GET	/~janyst/chat/chatInput.php	HTTP/1.1"	200
20	137.207.140.185	-	-	[23/Feb/2004:15:02:28	-0500]	GET	/~steve/itr_mon/resources.xml	HTTP/1.0"	200
21	67.71.64.233	-	-	[23/Feb/2004:15:03:05	-0500]	GET	/~reyes/change.html	HTTP/1.1"	200
22	67.71.64.233	-	-	[23/Feb/2004:15:03:06	-0500]	GET	/~reyes/whereismychange.wmv	HTTP/1.1"	200
23	137.207.216.174	-	-	[23/Feb/2004:15:03:34	-0500]	GET	/~janyst/	HTTP/1.1"	200
24	137.207.216.174	-	-	[23/Feb/2004:14:23:23	-0500]	GET	/~janyst/chat/userAppletXML.php	HTTP/1.1"	200
25	137.207.74.42	-	-	[23/Feb/2004:14:23:24	-0500]	GET	/~sanjay/freesite/typo3/alt_clickmenu.php?item=pages%7C1%7	HTTP/1.1"	200
26	137.207.74.42	-	-	[23/Feb/2004:14:23:25	-0500]	GET	/~sanjay/freesite/typo3/alt_topmenu_dummy.php	HTTP/1.1"	200
27	137.207.216.174	-	-	[23/Feb/2004:14:23:26	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
28	137.207.216.174	-	-	[23/Feb/2004:14:23:31	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
29	137.207.216.174	-	-	[23/Feb/2004:14:23:33	-0500]	GET	/~janyst/chat/chatAppletXML.php	HTTP/1.1"	200
30	137.207.216.174	-	-	[23/Feb/2004:14:23:37	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
31	65.93.140.55	-	-	[23/Feb/2004:14:23:37	-0500]	GET	/~reyes/change.html	HTTP/1.1"	200
32	137.207.216.174	-	-	[23/Feb/2004:14:23:38	-0500]	GET	/~janyst/chat/chatAppletXML.php	HTTP/1.1"	200
33	65.93.140.55	-	-	[23/Feb/2004:14:23:38	-0500]	GET	/~reyes/whereismychange.wmv	HTTP/1.1"	200
34	137.207.216.174	-	-	[23/Feb/2004:14:23:42	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
35	137.207.216.174	-	-	[23/Feb/2004:14:23:47	-0500]	GET	/~janyst/chat/chatWindow.php	HTTP/1.1"	200
36	137.207.216.174	-	-	[23/Feb/2004:14:23:52	-0500]	GET	/~janyst/chat/chatUsers.php	HTTP/1.1"	200

Fig 6: Result after Data Cleaning (Summarized log table)

6. RESULT ANALYSIS

The result shows that the proposed methodology reduces the size of the log file considerably by removing the unnecessary and irrelevant entries from the file. Earlier there were 601 entries in the log file, but after cleaning only 150 entries have been left. The original size of the log file before cleaning was 60 KB and after cleaning the size reduces to 15.6 KB. This is shown in the table 4.

Table 4. Comparison in Size Before and After Cleaning

	Size (KB)	**No. of Records**
Before cleaning	60	601
After cleaning	15.6	150

The change in size and number of records for the log file is graphically represented by means of a bar chart in figure 7.

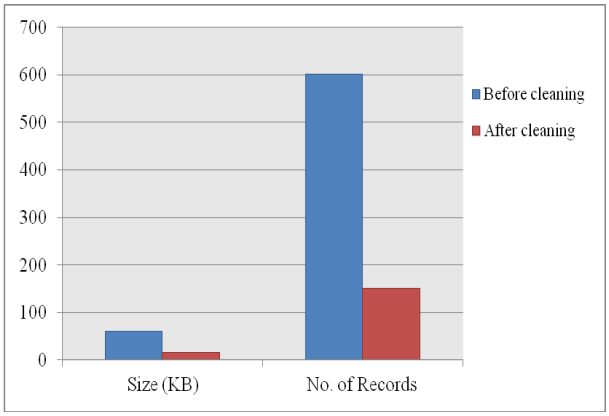


Fig 7: Bar Chart Showing Change in Size and Number of Records

The graph makes it clear that there is a severe change in both the size and number of records after data cleaning. Therefore, from this observation if calculate the percentage amount of decrease in the size of log file, it gives a reduction of 74% (see table 5) which is quite a major value.

Table 5. Results of Data Cleaning

Web Server Log File	**Result**
Original Size	60 KB
Reduced Size	15.6KB
Percentage in Reduction	74.00

7. CONCLUSION

Web data preprocessing is an important research direction of in the field of Web Mining. Web log files are the best source to predict user's behavior. Along with the useful information the raw log files also contains entries for unnecessary details like image access, failed entries etc. which are of no use from the perspective of the Web Usage Mining. Therefore, it becomes necessary to get rid of this irrelevant information. In this paper, the stages of data pre-processing have been explained. Algorithms for performing the data cleaning technique on server log have also been discussed. The proposed algorithm was successfully tested on the log files for data cleaning. The results which were obtained after the analysis were satisfactory and contained valuable information about the log files. The discussed approach showed a quite salient reduction in the number of records and in the log files size and hence increases the quality of the available data.

8. REFERNCES

[1] Etzioni, O., 1996. The World Wide Web: quagmire or gold mining?, Appears in Communications of the ACM, 1-6.

[2] Han, Q., Gao, X., and Wu, W., 2008. Study on Web Mining Algorithm Based on Usage Mining. In the Proceedings of 9th International Conference on Computer-Aided Industrial Design and Conceptual Design, 1121-1124.

[3] Aye, T.T., 2011. Web Log Cleaning for Mining of Web Usage Patterns. In the proceedings of 3rd International Conference on Computer Research and Developments, 490-494.

[4] Srivastava J. and Cooley, R., 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, ACM SIGKDD Explorations, 1(2), 12-23.

[5] Dong, D., 2009. Exploration on Web Usage Mining and its Application. International Workshop on Intelligent Systems and Applications.

[6] Raju G.T. and Sathyanarayana, P.S., 2008. "Knowledge discovery from Web Usage Data: Complete Preprocessing Methodology", International Journal of Computer Science and Network Security, 8(1), 179-186.

[7] Chaofeng, L., 2006. Research and Development of Data Preprocessing in Web Usage Mining. In the Proceedings of International Conference on Management Science and Engineering, 1311-1315.

[8] Tanasa D. and Trousse, B., 2004. Advanced data preprocessing for intersites Web usage mining. IEEE Intelligent Systems, 19(2), 59-65.

[9] Shahabi, C., Zarkessh, A.M., Abidi, J, and Shah, V., 1997. Knowledge discovery from users Web page navigation. In Seventh International Workshop on Workshop on Research Issues in Data Engineering, 20-29.

[10] Li, Y., Feng, B. and Mao, Q., 2008.Research on Path Completion Technique in Web Usage Mining International Symposium on Computer Science and Computational Technology ISCSCT '08, 1, 554 - 559.

[11] Li, Y. and Feng, B., 2009. The Construction of Transactions for Web Usage Mining. In the Proceedings of International Conference on Computational Intelligence and Natural Computing CINC'09, 1, 121 - 124.