Pre Processing of Web Logs – An Improved Approach For E-Commerce Websites

Jothi Venkateswaran C. [1], Sudhamathy G. [2]

[1] Department of Computer Science, Presidency College (Autonomous), Chennai – 600 005, India
[2] Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women University, Coimbatore – 641 043, India
[1] cjpcmcahead@gmail.com
[2] sudhamathi10@hotmail.com

Abstract—In this paper an improved approach for pre processing of web logs data has been proposed and evaluated so that it can be applied for web logs of e-commerce web sites. The resultant web log data after these pre processing steps can be used for further pattern discovery and analysis that helps to provide useful prediction to enhance e-commerce. Ideally, the input for the Web Usage Mining process is a user session file that gives an exact account of who accessed the web site, what pages were requested and in what order, and how long each page was viewed. A user session is the set of the page accesses that occur during a single visit to a web site by a web user. However, the information contained in a raw web server log does not reliably represent a user session file before data pre processing. Hence, data pre processing plays an important role in web usage mining applications. The data preparation process is often the most time consuming and computationally intensive step in the web usage mining process. The scope of this work is to enhance existing pre processing techniques for user and session identification that makes the web log data ready to use. This research work proposes a time-oriented and web ontology based user session identification algorithm which is found to be effective than the existing pre-processing approaches considering the run time, memory usage and processing complexity factors.

Keyword-Web Usage Mining, Web Logs, Pre Processing, E-Commerce

I. INTRODUCTION

The most significant phase in web usage mining is pre-processing as the data obtained from web that are collected through various sources are not ready for applying the web mining techniques such as association rule mining, clustering and classification. The pre-processing phase in web usage mining takes about 80% of the time and this phase involves complex computations. The success of this step is the base for pattern discovery and pattern analysis phases. A survey reveals that the growth of web sites is multiplying every day and the number of web sites double every year. As a result the numbers of users accessing the web sites have also tremendously increased over the years and so the web server log entries are also increasing accordingly. The data for web usage mining can be the web server logs, proxy server logs, browser logs, and user profiles. But for this work only the web server logs are used and hence the proposed pre-processing techniques have been applied only to the web server logs. This research deals with proposing and evaluating effective pre-processing techniques that can be applied on web server logs to make it relevant to be used for E-Commerce web sites. The contributions of this work involves discussion of heuristics that can be applied to clean the web server logs, identify the different users, identifying the user sessions and path completion and evaluating their efficiency using sample web log files.

II. PRE-PROCESSING

Web Server logs are plain text files in American Standard Code for Information Interchange (ASCII) format. These files contain the requests made to the web server when a web user browses the web site and it is recorded in the chronological order. Currently there are three formats existing to record the web server logs. They are the World Wide Web Consortium (W3C) Extended Log File format, Microsoft Internet Information Services (IIS) Log File format and National Centre for Supercomputing Applications (NCSA) Common Log File format. Of these file formats the log files used for this study follow the W3C Extended Log File format. Pre-processing is a process of converting the raw web server logs into a formatted user session database which is ready for applying web usage mining techniques. Pre-processing of web logs is done in four steps via, data cleaning, user identification, session identification, path completion and transaction identification. Data cleaning is a process of removing irrelevant records from the web server logs that does not make any sense for web usage mining. These are the requests with the image files, audio files, video files, unsuccessful requests and reference records due to spider navigations on the web site.

The principal aim of data cleaning is to reduce the count of records used for further processing. There are many difficulties involved in cleaning the raw server logs to eliminate outliers and irrelevant items [19]. User Identification is the process of identifying users by using the IP address, user agent and referrer fields of web log entries. Session Identification is the process of identifying the sequence of pages accessed by a user in his single visit to the web site. Session identification can be time based or web site topology based. Path completion is the process of filling in the missing pages in the sequence of pages that belongs to a user session. Transactions are constructed from the user sessions by considering a subset of the user session where the pages of the transaction have a semantic relation.

III. RELATED WORK

The focus of this section is to study, compare and contrast the available pre-processing techniques. Some of the works [1], [2], [3] proposed an intelligent algorithm of data pre-processing in which the users and sessions were identified and this algorithm is called as "USIA". An implementation of data pre-processing system for Web usage mining and the details of algorithm for path completion have been presented [4]. After user session identification, the missing pages in user access paths have been appended by using the referrer-based method which is an effective solution to the problems introduced by using proxy servers and local caching.

User identification is an important issue that deals with how exactly the users have to be distinguished. Three different methods for identifying the web users have been presented [5] and two of them are the most commonly used methods in web log mining systems, that is the users are identified only with their IP address. This can provide an acceptable result for short time periods or when the expected results from the data mining task do not need more precise information about the unique web users. The third one is an approach that uses a complex cookie-based method to identify web users. Furthermore in this work they also take steps towards identifying the individuals behind the impersonal web users.

In many other works [6] some heuristics are used for better identification of the users and these methods can be grouped into two classes, namely the pro-active methods and the re-active methods. Proactive strategies differentiates the users before or during the page request while reactive strategies relates users with the web log entries after the web log is written. Proactive strategies can be simple user authentication with forms, using cookies or using dynamic web pages that are associated with the browser invoking them. Reactive strategies work with the recorded log files only, and the different users will be distinguished by their navigational patterns, download timing sequence or some other heuristics based on some assumption regarding their behaviour.

The research carried out by Jaideep [7] had discussed the difficulty in the identification of users and sessions from web server logs. A study to assess heuristics for session identification from web log data has been presented [6]. In one of the research carried out [8], data cleaning was performed by removing the erroneous requests and image requests. A session was defined as a sequence of requests from the same IP address with a time limit of 30 minutes between consecutive requests. After session identification, sessions with a single request were removed.

The data pre-processing techniques detailed in [7], [9] have been experimented with the web server log of the library of South-Central University for Nationalities by Li [10]. As part of data pre-processing, data conversion was performed on the CTI data set [11], assigning a numerical value to each URL and bitmap algorithm was used to group a set of attributes into one attribute. One of the works [12] has outlined the various data pre-processing activities and has presented algorithms for data cleaning and data reduction. Edmond [13], have presented an efficient multidimensional data model for aggregating user access sessions for Web usage analysis.

IV. PROPOSED PRE PROCESSING TECHNIQUES

The proposed pre-processing technique involves the steps of data collection and storage, data cleaning, user identification, session identification and path completion. At the end of these steps, the resultant web log data becomes usable for applying web usage mining techniques in e-commerce web sites.

Data Collection and Storage

A web server log file is the one that stores information of an application server activity in the chronological order of web page requests by the users. This log file contains the Hyper Text Transfer Protocol (HTTP) requests made from clients to the web server of a web site. There are several formats available for the web server log files, but the most used is the W3C Extended Log File Format [14].

It can be seen that in the log files there are several entries with the same date and time. This is because the precision is limited to seconds, and therefore it is not possible to discriminate between requests made with a smaller time gap. In general, a *Timestamp structure* is of the following type: YYYY-MM-DD hh:mm:ss, with l as digit for milliseconds. In the logs, there are only the date with structure YYYY-MM-DD and the time with structure hh:mm:ss. But if the log entries are observed they are chronologically ordered, the chronological chain of events can be obtained, even with small time gaps. Hence this problem of having same date and time can be overcome by adding an incremental value for the milliseconds field in the timestamp [15].

The *user agent string may offer a great amount of information useful* for optimizing and personalizing Web sites. It is a valuable source that gives hints about browsers, operating systems used by users, and even analyzes the activity of crawlers. The user agent provides all these kinds of information, but there are no standards that define it. Indeed, it is a string of characters, which may even be empty, and which requires a specific parser to extract the data contained in it. A *user agent parser* [1] should therefore extract: information like the browser, the operating system and their respective versions, when a standard definition is found; identify a crawler with its name, or its owner, when a non-standard sequence of characters is found. Once the data from the various web log files have been collected they are loaded into database tables that have mapping fields that correspond to the fields in the web server log files [20]. This database is then used for further steps of data cleaning, user identification, session identification and path completion.

TABLE I
Fields in Web Log Files of W3C Extended Log File Format

Field Name	Field Description
Date	Date, in the form of yyyy-mm-dd
Time	Time, in the form of hh:mm:ss
s-ip	The IP of the server
cs-method	The requested action. Usually GET for common users
cs-uri-stem	The URI-Stem of the request
cs-uri-query	The URI-Query, where requested
s-port	The port of the server for the transaction
cs-username	The username for identification of the user
c-ip	The IP address of the client
cs(User-Agent)	The User-Agent of the Client
cs(Referer)	The site where the link followed by the user was located
sc-status	HTTP status of the request, i.e. the response of the server
sc-bytes	Bytes of data downloaded when the page request made
time-taken	Processing time of server in seconds to download the web page

Data Cleaning

Cleaning the raw server logs has many challenges in identifying the unwanted entries that does not make any sense to the web usage analysis. It is a *tough task* to perform analysis with the *huge amount of records* in the web server log files and hence to reduce the number of records to be processed further, an initial cleaning is required. The user requests normally include all type of necessary and unnecessary records like the *image files, video files, audio files* and others. These are not actually the user interested web pages; rather it is just the documents embedded in the web page. So it is not necessary to include in identifying the user interested web pages. This cleaning process helps in discarding unnecessary evaluation and also helps in fast identification of user interested patterns.

The web log files will also consist of failed requests from the users which can be identified by their failed status codes. These records also do not lead to any useful information and hence should be removed from the huge set of records. This cleaning process will further reduce the evaluation time for determining the user interested patterns. The Method field in the web log records can take the values *GET, POST* or *HEAD*. Of these the requests from the common users whose interest is targeted takes the value GET alone in the Method filed. Hence, to obtain more accurate user information it is required to *ignore the records with values POST and HEAD.*

TABLE II
Sample Records of the Raw Web Server Logs

S. No.	Sample Web Log Records
1.	2013-08-13 11:07:34 119.56.113.64 - W3SVC1460 IIS306 69.49.241.120 80 GET /Last.gif - 200 0 873 456 0 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.info/guest.html
2.	2013-08-13 11:07:35 119.56.113.73 - W3SVC1460 IIS306 69.49.241.120 80 GET /Last.gif - 200 0 28455 469 109 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.org/noplist.php
3.	2013-08-13 11:07:35 119.56.113.73 - W3SVC1460 IIS306 69.49.241.120 80 GET /prx1.php - 200 0 5688 426 16 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.org/noplist.php
4.	2013-08-13 11:07:36 119.56.113.73 - W3SVC1460 IIS306 69.49.241.120 80 GET /prx1.php - 200 0 57196 472 31 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.org/noplist.php
5.	2013-08-13 11:07:36 119.56.113.73 - W3SVC1460 IIS306 69.49.241.120 80 GET /prx1.php - 200 0 2662 457 16 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.org/noplist.php
6.	2013-08-13 11:07:37 119.56.113.73 - W3SVC1460 IIS306 69.49.241.120 80 GET /prx1.php - 200 0 10765 482 16 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=c3606a943d1c5f8bff7a76c40eb264cd;+msno=46;+mname=suresh http://eretailstore.org/logguest.php
7.	2013-08-13 11:07:37 119.56.113.75 - W3SVC1460 IIS306 69.49.241.120 80 GET /style.css - 200 0 26585 470 47 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=75f7991a2bbad133378a2ce26f3d2710 http://eretailstore.org/logguest.php
8.	2013-08-13 11:07:37 119.56.113.75 - W3SVC1460 IIS306 69.49.241.120 80 GET /style.css - 200 0 33044 480 16 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID =75f7991a2bbad133378a2ce26f3d2710 http://eretailstore.org/logguest.php
9.	2013-08-13 11:07:37 119.56.113.75 - W3SVC1460 IIS306 69.49.241.120 80 GET /style.css - 200 0 2243 479 0 HTTP/1.1 Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.11+(KHTML,+like+Gecko)+Chrome/23.0.1271.97+Safari/537.11 PHPSESSID=75f7991a2bbad133378a2ce26f3d2710 http://eretailstore.org/newguest.php

Apart from this the records produced by the *web robot or spider or crawler* have also to be removed. *Web robot is a software tool* that *periodically scans a web site* to extract its contents. Web robots automatically follow all the hyperlinks from a web page. Search engines such as Google, periodically use web robots to gather all the pages from a web site in order to update their search indexes [22]. Eliminating web robot generated log entries not only simplifies the mining task that will follow, but it also removes uninteresting sessions from the log file. The number of requests from one web robot may be equal to the number of the web site's URIs [18]. If the web site does not attract many visitors, the number of requests coming from all the web robots that have visited the site might exceed that of human generated requests. The requests of a web robot are out of the analysis scope, as the analysts are interested in discovering knowledge about users' behaviour.

Algorithm for Data Cleaning

1. Identify web log records with filename extensions that includes *.gif, *.js, *.jpg, *.jpeg, *.png and *.css in the URI-Stem field and remove those records from the web logs database.
2. Identify records with failed HTTP status code by examining the sc-status field of every record in the web access logs that has status code greater than 299 and status code less than 200 and remove those records from the web logs database.
3. Identify records with value "GET" in the cs-method field and only retain those in the web logs database. That is delete the records with the value "POST" or "HEAD" in the cs-method field as they do not represent the request from common users.
4. Identify records with text "robot" or "spider" or crawler" in the cs(User-Agent) field and remove those records from the web logs database.

Thus, once these four steps are executed the data cleaning step gets over and the cleaned web log database is ready for user identification and session identification.

```
Procedure Cleaning
Input: Unclean Web Log Database UR
Output: Cleaned Web Log Database R
Begin
{
    For each record r UR do
    {
        If r.URI-Stem <> "*.gif" AND r.URI-Stem <> "*.js"
        AND r.URI-Stem <> "*.jpg" AND r.URI-Stem <> "*.jpeg"
        AND r.URI-Stem <> "*.png" AND r.URI-Stem <> "*.css"
        AND r.sc-status >= 200 AND r.sc-status <= 299
        AND r.cs-method = "GET"
        AND r.cs(User-Agent) <> "*robot*"
        AND r.cs(User-Agent) <> "*spider*"
        AND r.cs(User-Agent) <> "*crawler*"
        {
            Add r to R;
        }
        End If;
    }
    End For; // for each record in UR
}
End; // procedure Cleaning
```

User Identification

The attention here is on deriving the information on users and user sessions from the analysis of the Hyper Text Transfer Protocol (HTTP) requests made by clients, grouped in sessions. A heuristic that identifies a single user with the pair IP address and user-agent is identified, and it permits only a fixed gap of time between two consecutive requests. Different people using the same proxy result in requests done by the same client IP address, despite of the real identity of the clients. The introduction of the user-agent permits to differentiate more clearly the source of requests. The *relationship between users and web log records is one to many*.

User Identification Assumptions

1. Each user has a unique IP address while browsing the website. The same IP address can be assigned to other users after the user finishes browsing.

2. The couple of client IP address and user-agent are considered for single user identification as different users can come from the same proxy.
3. The user may stay in an inactive state for a finite time (say *30 minutes*) after which it is assumed that the user left the website.

Given a list of web log records R = <r_1 … r_k>, where k is the total number of records in the web log database and k > 0. For each record r_i in R, r_i is defined as <date, time, c-ip, s-ip, user-agent, cs-uri-stem, cs-uri-query, status-id>, which is based on the key fields of the designed database. A user u has been represented by the set of fields <c-ip, user-agent, date, start_time, end_time, {p_s, …, p_e}>, where c_ip is the user's ip address, user-agent consists of the browser and OS details, date is the date when the user accessed the web site, start_time is the time when the user accessed the first page and end_time is the time when the user accessed the last page in that date.

```
Procedure User_Identification
Inputs: A list of n web log records R and maximum idle time β.
Outputs: A Set of users U consisting of l user records
Begin
{
        Let r1 be the first record in R;
        unew.c-ip = r1.c-ip;
        unew.user-agent = r1.user-agent;
        unew.date = r1.date;
        unew.start_time = r1.time;
        unew.end_time = r1.time;
        unew.url = r1.cs-uri-stem;
        For each record r in R do // Start from second record
        {
                If [(r1.c-ip = r.c-ip) and (r1.user-agent = r.user-agent) and
                        (r1.date = r.date) and (r1.time + β >= r.time)]
                {
                        unew.end_time = r.time;
                        unew.url = unew.url, r.cs-uri-stem;
                                //Append r.cs-uri-stem to unew.url
                        r1 = r;
                }
                Else if [(r1.c-ip = r.c-ip) and (r1.user-agent = r.user-agent)
                        And (r1.date = r.date) and (r1.time + β < r.time)]
                        Or [(r1.c-ip <> r.c-ip)
                        Or (r1.user-agent <> r.user-agent)]
                {
                        Add unew to U;
                        unew.c-ip = r.c-ip;
                        unew.user-agent = r.user-agent;
                        unew.date = r.date;
                        unew.start_time = r.time;
                        unew.end_time = r.time;
                        unew.url = r.cs-uri-stem;
                        r1 = r;
                }
                End If;
        }
        End For; // loop continues until the last record in R is processed.
}
End; // procedure User_Identification
```

The task of user identification is to find all users U = <u_1, … u_l> from the web records R such that if two consecutive requests from the same c-ip and user-agent has time gap of more than that of the maximum user's idle time β then they are considered as two separate users. Empirical observations showed that there are high chances that different users access the web site when an amount of time greater than thirty minutes passes between two requests from the same client [16].

The algorithm shown above (*Procedure User_Identification*) takes the first record from the cleaned web log database and initializes the first user with its data. Then it loops over the remaining records of the web log database starting from the second record until all the records in the web log database are processed [21]. In this loop two consecutive records are compared. If their ip, user-agent, date are same and the time difference does not exceed β then the url of the current record is appended with the url list of the current user and the current user's end time is updated with the time of the current record. If the ip address of two consecutive records are different or if their user-agent fields are different or if the time difference between them exceeds β, then the current user is added to the user's list and a new user is initialized with the data of the current record.

User Session Identification

A session has been defined as the stream of mouse clicks whereby a user is trying to perform a specific task [17]. In this research, different task specific browsing behaviours are compared. For example, assume two users, A and B, performed the following tasks. User A searched for classes and checked his grade; whereas, user B paid his tuition fees and searched for classes. The user identification process identifies users A and B as two separate users with totally different behaviours. However, if each user has been divided into different sessions, user A will have two sessions: searching for classes and checking for grades; user B will also have two sessions: searching for classes and paying tuition. This shows that users A and B are partially similar in searching for classes rather than being totally different if using only user identification process. Hence different sessions in a single user visit have been identified using the website ontology. It is assumed that the website ontology is already available through methods of retrieving website ontology. The website ontology is defined as a triple W = (P, L, F) where P is a finite list of website pages, denoted as P = <p_1 … p_k>, where k is the number of pages in the website. L is the group of links for a web application. Each link l = <p_s, p_d> has been defined by two pages, the source page p_s where a link starts from, and the destination page p_d where links ends. F is a list of website functionalities, which is defined as F = <f_0 … f_{n-1}>, where each f in F, f = <p_s … p_e>.

```
    Procedure Session_Identification
    Inputs: A Set of users U consisting of l user records
    Outputs: A Set of user sessions S consisting of k user session records
    Begin
    {
        For each user u in U do
        {
            For each page p in user u do
            {
                If (p in B)
                {
                    Split u at p location;
                    Snew = the first part of u;
                    u = the remaining part of u;
                }
                End If;
                Add Snew to S;
            }
            End For; // for each page in user
            Add u to S;
        }
        End For; // for each user
```

```
}
End; // procedure Session_Identification
```

Each web functionality f consists of at least two pages, a start page and an end page. There can be zero or more pages between the start and end pages. The list of consecutive pages in each f has link between them. That is for example f = <p_7…p_1> has the set of pages, f = < p_7, p_3, p_8, p_{11}, p_2, p_4, p_1> where p_7 is the start page for the functionality f and p_1 is the end page for the functionality f. Then there exists link between the pages p_7 and p_3, p_3 and p_8, p_8 and p_{11}, p_{11} and p_2, p_2 and p_4 and p_4 and p_1. In this case the set of links {l_1, l_2, l_3, l_4, l_5, l_6} is a subset of the set of all links L.

Where l_1= <p_7, p_3>,
l_2 = <p_3, p_8>,
l_3 = <p_8, p_{11}>,
l_4 = <p_{11}, p_2>,
l_5 = <p_2, p_4>,
l_6 = <p_4, p_1>.

The session identification algorithm will divide the users identified by the previous algorithm into different sessions using the website functionalities. From the website functionalities, it is possible to identify pages that are considered as the breaking points for the session, such as the sign-in or the sign-out pages. The algorithm above (***Procedure** Session_Identification*) shows the ontology-based session identification methodology, where B is the set of breaking pages or the set of end pages in each functionality f (Set of all p_e). The algorithm splits each user into one or more sessions and returns a final list of sessions S.

Path Completion

Due to the problem of local caching and the usage of the browser's back button in web pages some of the important page accesses may not be recorded in the web log file. Hence, to find such missing pages, the path completion step is required. If a current page request made by a user in a session is not directly linked to the last page request by the same user in the same session, then it is required to apply the path completion step and fill in the missing pages of the user session [23].

```
Procedure Path_Completion
Inputs: A set of user sessions S1 with Incomplete Paths
Outputs: A set of user sessions S2 with Completed Paths
Begin
{
    For each user session s in S1 do
    {
        For each set of consecutive pages p and q in user session s do
        {
            If p and q are not linked by any l in L
            {
                If p and q are not members of any f in F
                {
                    Split s into two user sessions;
                    // one of this user session contains p
                    // and the other contains q
                    Add these two user sessions into S2;
                }
                Else
                {
                    Complete the path between p and q;
                    // fill in missing pages using members of f
                }
                End If;
            }
```

```
                End If;
            }
            End For;
            Add the path completed user session to S2;
        }
        End For;
    }
    End; // procedure Path_Completion
```

Here in this *Procedure Path_completion*, the path completion is done using the set of links L and the set of functionalities F of the web site pages. If two consecutive pages accesses in a user session are not in the set of links, then find if these two pages are members of any functionality f. If so, fill the missing links between the pages using the set of pages in that functionality f. If the two consecutive pages are not members of any functionality f then they both belong to separate functionalities and hence should be further split into different sessions.

At the end of this step the result is the user session database with completed paths and this forms the base for further processing or applying of other web usage mining techniques such as clustering and association rule mining.

V. Experimental Results and Discussions

The proposed web logs pre-processing techniques were evaluated in this section and the results of the different stages are discussed. The method is evaluated using the web log files of an e-commerce web site www.eretailstore.org for the periods July 2013 to December 2013. The techniques were experimented using the software developed using *C# .Net in Visual Studio 2010* environment with *SQL Server 2008* as the database. The experiments were conducted using the *2.5 GHz Intel Core Pentium CPU with 3061 MB of main memory*.

TABLE III
Analysis of Web Log Data on Different Stages of Pre-Processing

S. No.	**Web Log Data (in Months)**	**Size of Log File (in MB)**	**No. of Raw Records**	**No. of Records (After Cleaning)**	**No. of Users**	**No. of User Sessions (Path Incomplete)**	**No. of User Sessions (Path Completed)**
1.	Jul 2013	3.08	9461	1455	121	302	421
2.	Aug 2013	3.36	10322	1613	115	322	389
3.	Sep 2013	3.11	9554	1447	104	229	267
4.	Oct 2013	3.31	10168	1614	132	316	387
5.	Nov 2013	3.34	10260	1531	117	315	349
6.	Dec 2013	3.04	9339	1460	98	225	267
	Total	**19.24**	**59104**	**9120**	**687**	**1709**	**2080**

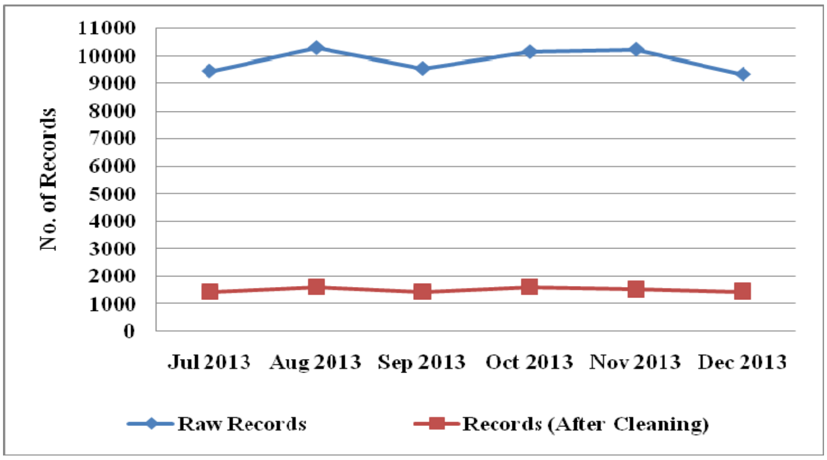


Fig. 1. Comparison of Record Counts Before and After Data Cleaning Step

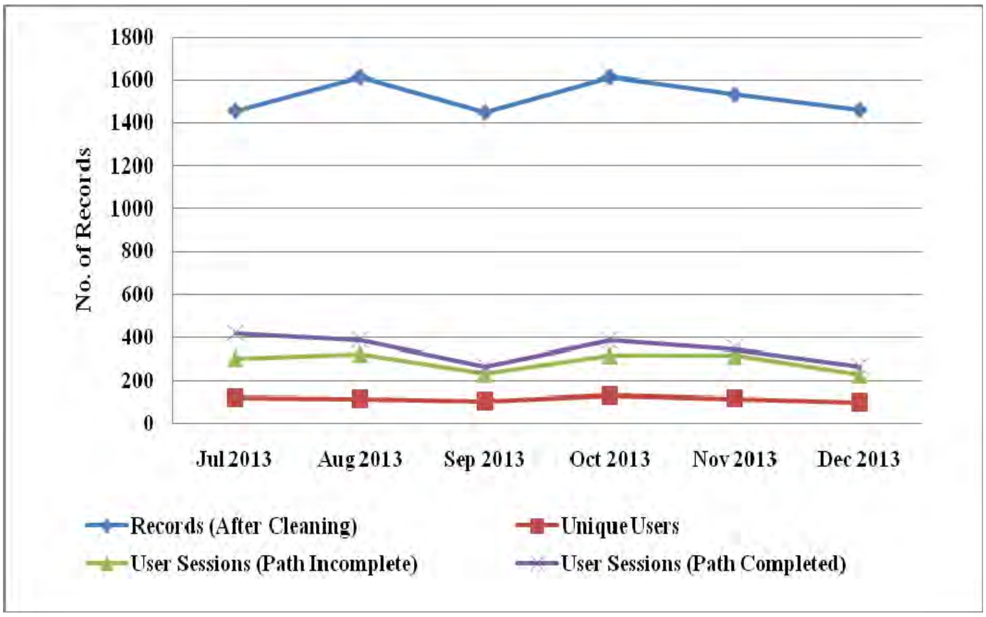
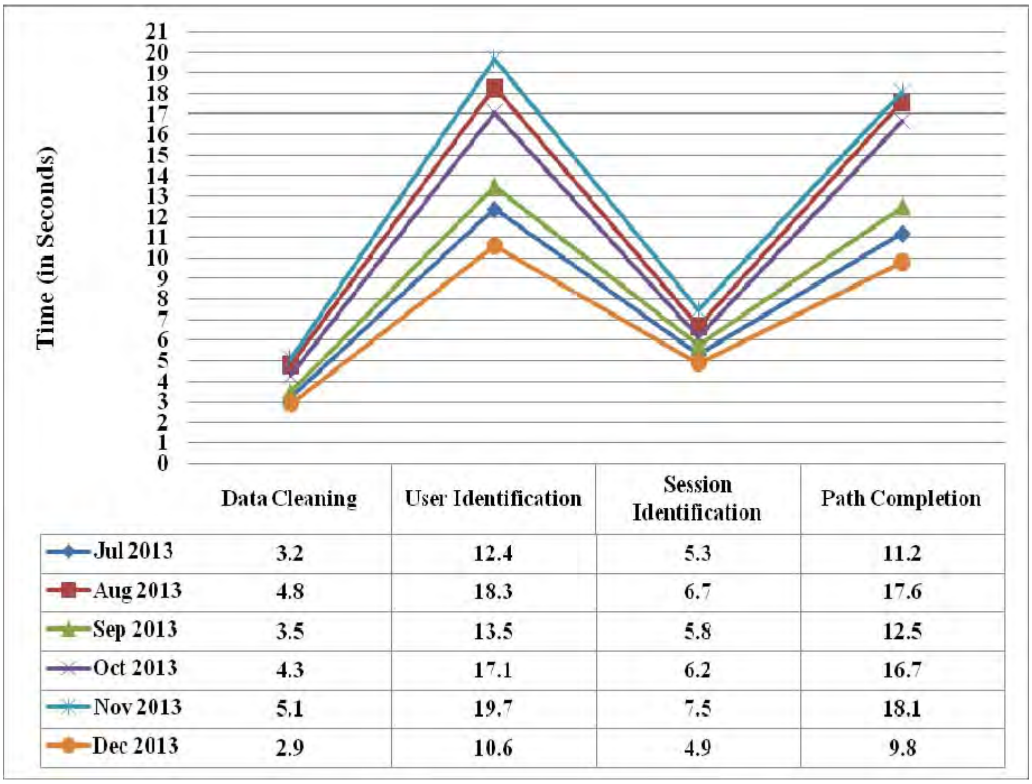


Fig. 2. Comparison of Record Counts After Every Step of Pre-Processing



	Data Cleaning	User Identification	Session Identification	Path Completion
Jul 2013	3.2	12.4	5.3	11.2
Aug 2013	4.8	18.3	6.7	17.6
Sep 2013	3.5	13.5	5.8	12.5
Oct 2013	4.3	17.1	6.2	16.7
Nov 2013	5.1	19.7	7.5	18.1
Dec 2013	2.9	10.6	4.9	9.8

Fig. 3. Comparison of Time Taken by Various Steps of Pre-Processing

The table III shows the analysis of the six months of web log data when processed through the various pre-processing steps / procedures. The graph representations of the comparison of record counts of the six months web logs after the various steps of pre-processing have been presented in figures 1 and 2. The times taken by the various pre-processing procedures are compared with respect to the six months web logs in the figure 3.

VI. CONCLUSION

This paper have proposed the effective pre-processing algorithms for making the raw web logs ready for application of web usage mining techniques such as clustering and association rule mining which are applied in E-Commerce. The techniques discussed apply both time-oriented heuristics and web site semantics which produces effective results than the existing approaches. With the given set of web log files used as the sample for studying the procedures that were introduced in this work, it has been proved that the method reduces the size of the web log data drastically and it takes fewer seconds to execute the steps of pre-processing. Also, the quality of the session database that is available as a result of applying all the steps reflects the real user session and their preferred set of requests on the web site. The significance of the proposed pre-processing procedures is evident from the results.

REFERENCES

[1] Robert, C., Bamshed, M., & Jaideep, S. (1997). “Web Mining: Information and Pattern Discovery on the World Wide Web”, In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI ‘97), (pp. 558-567).

[2] Robert, C., Bamshed, M., & Jaideep, S. (1999). “Data Preparation for Mining World Wide Web Browsing Patterns”, Journal of Knowledge and Information Systems, (pp. 5–32).

[3] Zhang, H., & Liang, W. (2004). “An Intelligent Algorithm of Data Pre-processing in Web Usage Mining”, In Proceedings of the 5th World Congress on Intelligent Control and Automation.

[4] Yan, Li., Bo-Qin, F., & Qin-Jiao, M., (2008). “Research on path completion technique in web usage mining”, In Proceedings of the International Symposium on Computer Science and Computational Technology, IEEE Xplore, Shanghai, (pp. 554-559).

[5] Renáta, I., & Sándor, J. (2007). “Analysis of Web User Identification Methods”, World Academy of Science, Engineering and Technology, 34, (pp. 34-59).

[6] Spilipoulou, M., Mobasher, B., & Berendt, B. (2003). “A framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis”, INFORMS Journal on Computing Spring, 15(2), (pp. 171-190).

[7] Jaideep, S., Robert, C., Mukund, D., Pang-Ning, T. (2000). “Web Usage Mining : Discovery and Applications of Usage Patterns from Web Data”, ACM SIGKDD Explorations, 1(2), (pp. 12-23).

[8] Jose, B., & Mark, L. (2008). “Mining users’ web navigation patterns and predicting their next step”, NATO Secur. Sci. Ser. D-Inform. Commun. Secur., (pp. 45-55).

[9] Bamshad, M. (2004). “Web Usage Mining and Personalization”, Practical Handbook of Internet Computing. Ed. Editor, CRC Press M.P. Singh., (pp. 1-37).

[10] Li, C. (2009). “Research on Web Session Clustering”, Journal of Software, (pp. 460-468).

[11] Norwati, M., & Mehrdad, J. (2009). “Expectation maximization clustering algorithm for user modeling in web usage mining systems”, Eur. J. Sci. Res., (pp. 467-476).

[12] Navin Kumar, T., Solanki, A.K., & Sanjay, T. (2010). “An Algorithmic Approach To Data Preprocessing in Web Usage Mining”, International Journal of Information Technology and Knowledge Management, (pp. 279-283).

[13] Edmond, H.W., Michael, K.N., & Joshua, Z.H. (2004). “A Data warehousing and Data Mining Framework for Web Usage Management”, Communications in Information and Systems, (pp. 301-324).

[14] Hallam-Baker, P., & Behlendorf, B. (1996). “Extended Log File Format, W3C Working Draft”, WDlogfile-960323, URL: http://www.w3.org/TR/WD-logfile.html.

[15] Agosti, M., & Di Nunzio, G.M. (2007). “Web Log Mining: A Study of User Sessions”, In Proceedings of 10th DELOS Thematic Workshop on Personalized Access, Profile Management, and Context Awareness in Digital Libraries, PersDL, Corfu, Greece, (pp. 70–74).

[16] Zaiane, O.R., Xin, M., & Han, J. (1998). “Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs”, In Proceedings of Advances in Digital Libraries Conference, (ADL’98), Santa Barbara, CA, (pp. 19–29).

[17] Jansen, B.J., Spink, A., Blakely, C., & Koshman, S. (2007). “Defining a Session on Web Search Engines”, Journal of the American Society for Information Science and Technology, 58(6), (pp. 862-871).

[18] Huaqiang, Z., Hongxia, G., & Han, X. (2010). “Research on Improving method of Preprocessing in web log mining”, IEEE.

[19] Theint Theint, A. (2011). “Web log cleaning for mining of web usage patterns”, IEEE.

[20] Sudheer Reddy, K., Partha Saradhi Varma, G., & Ramesh Babu, I. (2012). “Preprocessing the web server logs an illustrative approach for effective usage mining”, ACM.

[21] Arvind Kumar, D., & Sunita, S. (2013). “A new approach for user identification in web usage mining Preprocessing”, IOSR-JCE.

[22] Nithya, P., Sumathi, P. (2012). “Novel Pre-Processing Technique for web log mining by removing global noise and web robots.” IEEE.

[23] Wasvand, C., Devale, P.R., Ravindra, M. (2014). "Survey on Data Preprocessing Method of Web Usage Mining", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), (pp. 3521-3524).