A Novel Semantically-Time-Referrer based Approach of Web Usage Mining for Improved Sessionization in Pre-Processing of Web Log

Navjot Kaur
Department of Computer Engineering
Punjabi University
Patiala, Punjab, India

Dr. Himanshu Aggarwal
Department of Computer Engineering
Punjabi University
Patiala, Punjab, India

***Abstract*—Web usage mining(WUM) , also known as Web Log Mining is the application of Data Mining techniques, which are applied on large volume of data to extract useful and interesting user behaviour patterns from web logs, in order to improve web based applications. This paper aims to improve the data discovery by mining the usage data from log files. In this paper the work is done in three phases. First and second phase0 which are data cleaning and user identification respectively are completed using traditional methods. The third phase, session identification is done using three different methods. The main focus of this paper is on sessionization of log file which is a critical step for extracting usage patterns. The proposed referrer-time and Semantically-time-referrer methods overcome the limitations of traditional methods. The main advantage of pre-processing model presented in this paper over other methods is that it can process text or excel log file of any format. The experiments are performed on three different log files which indicate that the proposed semantically-time-referrer based heuristic approach achieves better results than the traditional time and Referrer-time based methods. The proposed methods are not complex to use. Web log file is collected from different servers and contains the public information of visitors. In addition, this paper also discusses different types of web log formats.**

Keywords*—Web Usage Mining; User Identification; Session Identification; Semantics; Data Cleaning; Time Heuristics; Referrer Heuristics

I. INTRODUCTION

Web mining [1] is the application of data mining techniques used to extract interesting, useful patterns and hidden information from the Web documents and Web activities. Web mining simply refers to the discovery of information from Web data that includes web pages, media objects on the Web, Web links, Web log data, and other data generated by the usage of Web data. Web mining is quite similar to the mining of valuable minerals from the earth. Web mining is further divided into three types of mining; namely web content mining, web structural mining, and web usage mining [14]. Web Content Mining is the process of extracting useful information from the contents of Web documents. Content data corresponds to the collection of facts that a Web page is designed to convey to the users. It may consist of text, images, audio, video, or structured records such as lists and Tables. Web Structure Mining is the structure of a typical Web graph consisting of Web pages as nodes, and hyperlinks as edges connecting two related pages. Thus it can be regarded as the process of discovering structural information of the Web entities. Web Usage Mining discovers the user navigation patterns from web log data in order to understand user behaviour and better serve the needs of web users and web-based applications. Web mining is necessary because, the data or the web pages on the www are increasing exponentially day by day. It is a highly challenging task to extract useful data from web. An effective approach to find the interesting or useful data quickly, efficiently and accurately from www is web mining.

II. WEB USAGE MINING

First, Web usage mining (WUM) also known as Web Log Mining is the application of data mining techniques applied on large volume of data to extract relevant, useful and interesting patterns from Web data, specifically from web logs, in order to improve web based applications [12]. Web Usage Mining is often regarded as a part of the Business Intelligence in an organization rather than the technical aspect. This is used for deciding business strategies through the efficient use of Web Applications. It is also crucial for the Customer Relationship Management (CRM) as it can ensure customer satisfaction as far as the interaction between the customer and the organization is concerned. The major problem with Web Mining in general and Web Usage Mining in particular is the nature of the data they deal with. With the updation of Internet in this millennium, the Web Data has become huge and a lot of transactions and usages are taking place in micro seconds. The data is not completely structured. It is in a semi-structured format so it needs a lot of pre-processing and parsing before the actual extraction of the required information.

Web Usage mining consists of four phases-data collection, pre-processing of log data, pattern discovery and pattern analysis. In first Phase the data is collected from Web Log files. There are three types of data sources of log files namely Web server, Web Proxy Server and Client Browser [17]. In Second Phase which is Pre-processing, elimination of irrelevant information from original log file is done to make it ready for Sessionization and user identification process. The main purpose of pre-processing is to improve the quality and accuracy of data. Next and Third Phase of Web usage mining is Pattern Discovery which means discovering patterns from

pre-processed data using various data mining techniques[14,17] like Association, Clustering , Statistical Analysis and so on. In the last phase of WUM, Pattern analysis is done using Knowledge Query Mechanism such as SQL or data cubes to perform OLAP operations [7].

After the completion of these four phases, the user can find the required usage patterns and use this information for the specific needs in a variety of ways such as improvement of the Web application, to fight with terrorism and identifying criminal activities, to increase profitability for web based applications, identifying the visitor's behaviour, customer attraction, customer retention, etc.

III. Related Work

Web usage mining is an important area of research. A lot of research has been conducted by many researchers. Cooley et al defined the term web usage mining for the first time and the aim was to predict the user's preferences and behaviour [2]. The pre-processing [1] of web logs is usually complex and time consuming task. It consists of three main phases: Data cleaning, User Identification, Session identification. Hence the related work of each phase of pre-processing is discussed in sequence under this section.

A. Data Cleaning

Every phase of pre-processing is important in its own right. Without the data cleaning phase the user and session identification is useless. This phase removes all the data tracked in web logs that is irrelevant, useless or noisy and not needed for further web usage mining phases [4, 5, 6,7,8]. Generally the following steps are followed to clean the web log file.

- Removal of local & global noise
- Removal of multimedia requests
- Removal of failed and corrupted requests
- Removal of requests originated by web robots
- Removal of requests with access method other than GET method

Removal of multimedia requests depends upon the purpose of the web usage mining [9].When the purpose is to support web caching or pre-fetching, the analyst should not remove the log entries referring to images and multimedia files. In such cases just the suffix like ".jpg or .jpeg or .mvi" is removed from log file and the whole record is kept for analysis. Web robots also known as web crawlers or web spiders, are the software tools that automatically download complete websites by following every hyperlink on every page within the site. Search engine such as google periodically use robots or spiders to grab all the pages from a website to update their search indexes[4,9,10].Three different techniques are used to identify web robots request[9,10,11]:

- Remove all records which contain robots.txt in URL field
- List of user agents known as robots can be used [9].
- Calculate the browsing speed and delete all requests whose browsing speed is greater than threshold value.

B. User Identification

User identification is to identify who accesses the website and which pages are accessed [11, 12]. A user is defined as the principal using a client to interactively retrieve and render resources or resource manifestations [13]. Due to the existence of local caches, proxy servers and corporate firewalls, user identification becomes a highly challenging task. Proxy caching cause a single IP addresses to be associated with different users. So only the IP address is not sufficient to identify a user. This problem is partially solved by the use of cookies, by URL rewriting or by requiring user to login when entering the websites [14]. Cookies help to identify different users but due to limited information and some browsers not supporting cookies and some browsers allow the users to disable cookies support, it becomes ineffective to reveal different users [14].Common methods used to identify different users are IP and User agent, cookies or direct authentication [11]. It becomes easy to identify different users if the user is a registered user for that website. But in case of unregistered users, the IP and agent field is used to uniquely identify the users [14].

C. Session Identification

A user session is a sequence of activities performed by the same user within a particular visit. Users are identified in user identification phase. This phase identifies the number of sessions by each user.[4,12]. To identify sessions from log data is again a complex task, because the server log files always contain limited information. Sessionization process helps to find out more potential and meaningful information like a user's preference and even his intention. There are four traditional methods used to identify different sessions, they are time based [4, 12], referrer based [4,12],and semantic based[18]. Time-based heuristics consider temporal boundaries such as maximum session length or maximum time allowable for each page view [18, 19].There are several methods to calculate temporal boundaries. Most commonly used timeout is 30 min (maximum) for session length and 10 min (maximum) time for page view. Catledge and Pitkow have calculated the maximum page view time as 9.3 min [21]. Similarly cooley et al. 2000 has calculated 25.5 min as maximum session length for the mining of log file [2]. There are limitations of time heuristic method because the sessions are divided, based only on time and not segregated according to the navigation patterns. Most of the time the same session is divided into more than one session or either multiple sessions are counted as a single session.

Navigation-based heuristics overcome this limitation to some extent but it also has several limitations. In this method if the requested URL is not directly accessible from previous URL then the current request is assigned to a new session. But for this technique the topological graph structure of particular website has to be maintained which is complex task [18].

Referrer based heuristics is another method used for sessionization which is completely based on referrer field. In

this method, if referrer of the current request is same as the URL of the previous request then the current request is counted in the same session, otherwise new session is created[16,18]. But this method also has a limitation i.e. most of the times the referrer field of log file contain hyphen.

Yongyo Jiang has proposed a new approach called the time-referrer based method [16], which is the combination of time and referrer based methods. In their algorithm if the referrer field is not empty or it does not contain any commercial search engine or it is not the first page of website then T_{pq} which is the time between current and previous request is compared. This method looks for the most recent page p whose request is identical to current referrer. It is found the T_{pq} is calculated between p & q and it is compared with T*N, where N is number of logs between p and q and T is threshold. This algorithm improves the performance of referrer based heuristic only if the log file is free from the blank referrer field or entries contain hyphen, otherwise it will give poor performance.

Jason J. Jung has implemented semantic outlier detection from online web request streams and their sessionization[18]. Semantic labelling of all the webpages of the log file has been done. The registered URLs are labelled directly using web directory but for unregistered websites the semantic labelling is done using link-analysis-based indirect labelling. The author has discussed the limitation of this method.

- The multi attributes of a website
- The relationship between categories: subordination and redundancy.

So the implementation or use of this method is again a tough task. It is time consuming and due to its limitations mentioned above it does not give accurate results.

Fang Yuankang and Huang Zhiqiu proposed a new method for sessionization in which they uses page access time threshold to identify sessions [15]. After identifying specific users, a great deal of frame pages were filtered, the relatively reasonable access time threshold for each page was made up according to contents of each page and all web structure and user's session sets were identified by this threshold. It improves the authenticity and efficiency of session identification at some level, but before implementing this method, we have to construct the threshold ($\delta=\alpha_{t(1+\beta)}$)of page access time according to the importance of the each page. Where β is the influence factor of page R_{LCR} to access time threshold δ with its formula $\beta=1-\exp(-\text{sqrt}(\text{sqrt}(R_{LCR})))$. Linking content ratio (R_{LCR}) is calculated using formula: $R_{LCR=}(L1+L0)/S_{DS.}$ L1 refers to amount of link-in pages and L0 means the amount of link-out pages.

Log file cleaning is a very important and major process of web usage mining. It requires a lot of time and effort. In conclusion of literature review, existing methods do not use very effective data cleaning methods that completely overlook the characteristics of web server log files. Most of the methods of cleaning log file are based on status code, HTTP Method, multimedia clicks, robots requests. But it does not automatically cleans the other text files, automatic requests made by advertisements or software's for their updation, for error recovery by web application when you are online but not active especially at night time. These requests also get recorded in log files. For better analysis of finding user navigation pattern, one should clean file completely and properly. Further after cleaning the user and session identification is mostly done by Time or navigation based heuristics where the use of a lot of empirical values are made but without detailed discussion. This has many problems like in time heuristics only time is checked and the total time of session is taken as 30 min. If the request is made even a few seconds after 30 min. Window, it will not be considered in same session even if otherwise it belongs to the same session. Also, not much work has been done regarding semantic user and session identification.

In this paper we will be comparing three methods of sessionization, out of which 'time sessionization' is the traditional method. Second method is the 'referrer-time' which is modified concept of time and referrer. The Third proposed method is 'semantically-time-referrer method' in which we are using the concept of semantics, time and referrer in combination.

We have first cleaned our log file using traditional cleaning method with some minor modifications. Then, from cleaned log file we have identified the unique users using IP and agent field and in the last step, we have found the session activity for each user using three different methods which are time, referrer-time and semantically-time-referrer methods. Two new methods have been proposed for sessionization which is giving better results than existing or traditional methods. Comparison of these three sessionization methods has also been done.

IV. Location of Log Files

Data is collected form server in form of log file (or files), which is automatically created and maintained by a server. Log file consist of list of activities performed by the visitors on web pages. There are three types of servers which act as the sources of log files – Web Server Logs, Web Proxy Server and Client side logs. [14].

A. Web Server Log files

Web server Log file are most accurate but these files contain personal information and do not record cached pages visited.

B. Web Proxy Log Files

Proxy servers accepts HTTP request from user, gives them to the web server and then result passed by the web server is returned to the user [8]. Web proxy server's construction is difficult. Web proxy servers are used for various purposes like to share internet connection on LAN, to hide the IP address of client, to implement internet access control and the most importantly, to speed up the internet surfing due to proxy's cache.

C. Client Side Log files

Client side log files can reside in client's browser window itself. For this special software is downloaded by the users to their browser window [14].

Web server log file contain entries of users in terms of plain text, who access that website or web pages. Each entry of

which contain the public information of visitor like IP address, remote user, date, time , zone, method, URL, status code, number of bytes transferred, operating system used etc. This type of data can be merged into a single file or separated into distinct logs, like Access log, Agent Log, Referrer log, Error Log. These files are not accessible to every internet user but the administrator.

V. LOG FORMATS

The server access log records all requests processed by the server. The location and content of access log are controlled by the CustomLog directives. The LogFormat directives can be used to simplify the selection of the contents of the logs. The format of access log is highly configurable. Some of the formats are discussed below:

A. Combined Log Format

The configuration of combined access log looks like shown in Figure 1. This defines the common nickname and associates it with a particular log format string [17].

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i
\" \"%{User-agent}i\"" combined

CustomLog log/access_log combined
```

Fig. 1. Shows the Configuration of Common Log Format

```
117.96.61.194 - - [26/Aug/2014:06:03:30 +
0530] "GET /misc/drupal.css HTTP/1.1" 200
9315 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.1 (KHTML, like Gecko)
Chrome/21.0.1180.83 Safari/537.1"
```

Fig. 2. Sample Combined Log Format from our Colg Log File

Figure 2 reflects the information of first entry of our log file of an Education institute which is in combined log format as follows:

- 117.96.61.194: It is the Remote IP address or domain name which is 32 bit host address defined by the internet Protocol.
- - : This is remote user. Usually the name of remote user is omitted and replaced by hyphen (“-“).
- - : Login of remote user. Like the name of remote user, Login of remote user is also usually omitted and replaced with hypen (“-“).
- [26/Aug/2014:06:03:41 +0530]: It contains date, time and Zone. First is Date in
- DD/MM/YYYY] format, Second time which is in HH: MM:SS format. And last is zone.
- “GET/misc/drupal.css HTTP/1.1” : It contain Method, URL relative to domain and Protocol. GET or POST or HEAD is Method. “?q=policy.html” is the URL and HTTP/1.1 is a protocol with version 1.1.
- 200: This field is for Status code and 200 code is for success. If code is <200 and >299 it is considered as error or failure of request
- 9315: This field shows the content-length of the document transferred in bytes.
- - : It is the field of referrer. It person directly access the site then this field contain hypen (“-“).Otherwise the URL of referrer.
- “Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.83 Safari/537.1” Almost all browsers start with Mozilla Browser type, Netscape Navigator with version 5, OS: “WindowNT 6.1”, “WOW64” means, a 32-bit Windows is running on a 64-bit processor, “AppleWebKit/537.1” is unknown fragment, “KHTML” is a free HTML layout engine developed by the KDE project, “like Gecko” is not a Geckeo browser, but behaves like a Gecko Browser. Gecko is the open source browser engine designed to support open Internet standards and is used in several browsers like Firefox, SeaMonkey and other, “Chrome/21.0.1180.83”: was a Beta Channel Update for Windows or Table Channel Update for Windows. Safari/537.1:unknown fragment

B. Common Log Format

This type of format looks like as shown in Figure 3 below. Common Log format log File does not contain last two fields of combined log format which are referrer and agent fields [17].

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common

CustomLog logs/access_log common
```

Fig. 3. Shows the Configuration of Common log format

Figure 4 reflects the information of first entry of our NASA log file, which is in the common log format .It contain the information of IP address ,remote user, login of remote user, date time and zone URL ,status code and number of bytes transferred during of user access.

```
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400]
"GET /history/apollo/ HTTP/1.0" 200 6245
```

Fig. 4. Sample Common Log Format from our NASA Log File

C. Multiple Access Logs

In this format multiple log file can be created by specifying multiple CustomLog directives, where as in common and combined access log, only one log file can be created. Example shown in Figure 5 creates three access log files [17].The real power of multiple log files come from the ability to create log files in different formats. As an example three files have been created in above Figure, as well as a CLF transfer log, the server could log the referrer information and the user agent of each client. This example also shows that it is not necessary to

write a nickname with the LogFormat directive. Log format can be specified in the customLog directive.

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
CustomLog logs/referer log "%{Referer}i -> %U"
CustomLog logs/agent_log "%{User-agent}i"
```

Fig. 5. Shows the Configuration of Multiple Logs

D. Conditional Logs

Conditional log is very powerful and flexible. In this log, the user can include or exclude certain entries from access log based on some characteristics of the client request. For this process SetEnvIf variable is used. As an example shown in Figure 6, user sets the condition that log entries from IP address 127\.0\.0\.1, robots.txt requests need not to be considered in the log. In our work, the third log file is from online guitar selling website which is a conditional log file, it does not have the failure status code entries, robots.txt requests.

```
SetEnvIf Remote_Addr "127\.0\.0\.1" dontlog
SetEnvIf Request_URI "^/robots\.txt$" dontlog
CustomLog logs/access_log common env=!dontlog
```

Fig. 6. Shows the Example of Conditional Logs

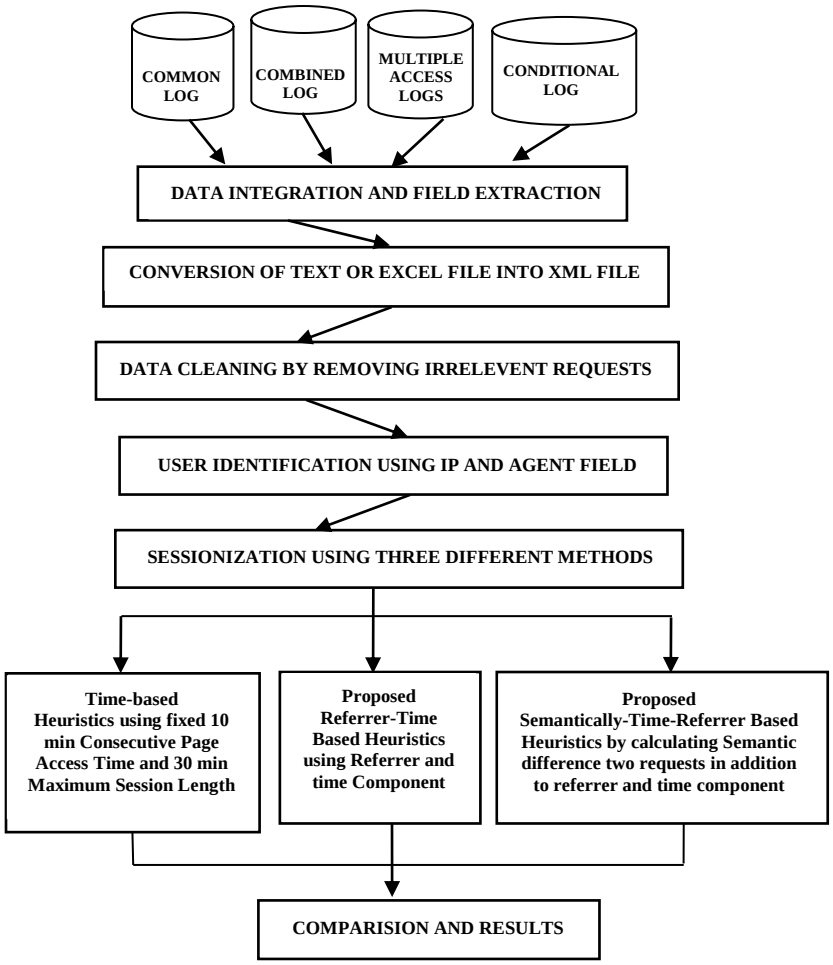

Fig. 7. Proposed Model for Pre-processing of Web Log

VI. Proposed Methodology

The proposed model for pre-processing of web log data is described in Figure 7 and the algorithm for the same is as follows.

Algorithm 1. **Consider the following steps.**

Step 1. Log files are collected from the web servers, then data integration and field extraction is done.

Step 2. Log file is converted to XML format

Step 3. Log cleaning by removing irrelevant data

Step 4. Identification of unique users from the cleaned log file.

Step 5. Session Identification for each of the above identified unique users using three different techniques.

- Time based
- Proposed Referrer-Time based

- Proposed Semantically-Time-Referrer based

Step 6. Comparison of above three session identification techniques.

Step 7. Results

The first task is to collect the log files from the sources and then data integration and field extraction is done Next step is to convert the log file from text or excel file into XML file. Third step is to apply the traditional log cleaning method. In step four unique users are identified from the cleaned log file. In the next step, for each uniquely identified user we have applied three different methods to identify the number of sessions for each unique user. A user session is a sequence of activities performed by the same user with in a particular visit. This paper has implemented three methods of sessionization namely time-based, referrer-time based and semantically-time-referrer based. Time-referrer and Semantically-Time-referrer based methods are proposed methods whereas time based is the traditional method. In the end the comparison of above three methods is done to find the best sessionization method out of these three.

A. Phase-I: Data Cleaning

Initially merging of the log files from various web and application servers is done in the data fusion or integration phase. Log files contain various fields which need to be separate out for pre-processing [5]. Field extraction is a process of separating fields from log file. We have used two methods of field extraction. First is to directly read the text file, extract the field and convert the file into XML file for further processing. But this file should either be in combined or common log format. In the second methodology, an excel log file of any format can be read. Our model will extract the necessary fields and convert the file into XML file for further processing.

Data cleaning phase is the most powerful phase, because it has great impact on the results of web usage mining. All results of Web-usage Mining (WUM) depend on this phase if the data cleaning, user and session identification of log is not done properly, than results will not be of any use. Data cleaning is required because most of the log files contain noisy, ambiguous or irrelevant data which may affect the result of Log Mining process. The raw data should be cleaned to eliminate irrelevant information form original log file and to make the web log file easy for session and user identification process. The main purpose of pre-processing is to improve the quality and accuracy of data. The main steps of pre-processing phase are as follows:

- ✓ Collect the data in form of log files from the servers
- ✓ Clean the web logs by removing the redundant, noisy or irrelevant information

Initially the log files are collected from the servers and data integration and field extraction is done. The Proposed pre-processing model can extract fields from text and excel file of any format and insert them into XML file in tabular form. In our work we have taken three log files of different formats from web servers namely “colg”, “guitar” and “NASA”.

The colg log file has been collected from an educational institute and it is in a combined log format. Second log file is a NASA log file having common log format, containing requests to NASA Kennedy Space Centre WWW server in Florida. Third and last log file is guitar log file. It is a conditional log file and has been collected from online guitar selling website. This paper has shown the results of each phase of pre-processing for these three log files. The details of log files before data cleaning are shown in Table 1.

TABLE I. DETAILS OF LOG FILES BEFORE DATA CLEANING

Features	Educational Institutional Log File	Online Guitar Selling Log File	NASA File
File name	Colg	Guitar	NASA
Size in KB	559KB	391KB	285KB
Time period	5 days	79 days	1day
No of entries in log file	7956	4589	4045
Format of log File	In Combined Log Format	Conditional Log Format	In Common log Format
Type of file	Txt file	Excel file	Txt File
Data Transferred in KB	137042300	79001375	93136545

This phase of pre-processing removes the irrelevant, noisy, unnecessary and redundant log entries. Data cleaning algorithm removes the failure requests, robots requests and requests from method other than GET. Although we have not removed the multimedia and text files requests but removed their tags or extension and have kept them in the same file for further analysis. Requests in log file which shows nearly zero transferred bytes have also been removed during the cleaning process.

TABLE II. DETAILS OF LOG FILES AFTER DATA CLEANING

Features	Colg log file		Guitar web site		NASA Website	
	Count	%age	Count	%age	Count	%age
Multimedia clicks	4518	56.78	133	0.28	2359	58.31
Text File clicks	1005	12.63	Nil	0	78	0.19
Robots.txt clicks	369	0.47	Nil	0	Nil	0
Error clicks	1280	16.08	Nil	0	442	10.92
Other than GET Method	137	0.17	Nil	0	02	0
Size of Cleaned file in KB	471	84.25	367	93.61	252	88.42
Entries in cleaned file	6522	77.46	4589	100	3601	89.02

As we have already discussed, if the log file is not in Common log format or combined log format, still our model can read the log file providedt that the file should be in excel format. Table 2 shows the results data cleaning algorithm after removing the noisy data or irrelevant data.

TABLE III. SHOWS THE DETAILS OF DIFFERENT FILE REQUESTS

Text Files Requests				Multimedia Requests			
	Colg	Guitar	NASA		Colg	Guitar	NASA
.doc	181	0	0	.png	1441	0	0
.xls	12	0	0	.jpg	367	77	184
.txt	369	0	78	.jpeg	0	0	12
.pdf	440	0	0	.gif	771	0	2048
.xml	3	0	0	.wav	0	0	26
Method Of Request				.mp3	0	0	89
GET	7819	4589	4043	.exe	0	56	0
POST	55	0	1	.css	1939	0	0
HEAD	24	0	1				
PROPFIND	40	0	0				
CONNECT	1	0	0				
OPTIONS	17	0	0				

In Table 2, the number of failure requests, robots requests, multimedia requests, and text file requests and failure requests are shown for each log file selected for analysis. The number of requests other than GET method are also shown. The main point to note is the number of entries for each cleaning step is calculated from the main unclean log file whereas if you find it step by step during cleaning process, the results will be different. Detailed number of multimedia log files, text files and method requests except GET are shown in Table 3.

The column chart in Figure 8 shows the change in each log file after cleaning. Change in guitar log file is minimal as compared to colg and NASA log file, because the guitar log file is a conditional log file. This log file does not contain any robots.txt file or failure status code requests. Only GET method requests are there. Therefore the data needs to clean just for multimedia and text file requests.

B. Phase-II: User Identification

User identification phase of pre-processing identifies individual user by using their IP address. User's identification is to identify who accesses the website and more precisely which pages are accessed. Traditional method of user identification has been used by addition of time constraint. IP address and agent field are used to find unique users in existing method. But we have also the time constraint i.e if IP and agent are same even after long time our algorithm will create new user. The threshold value is determined from the log file by calculating the average accessing time of all unique users.

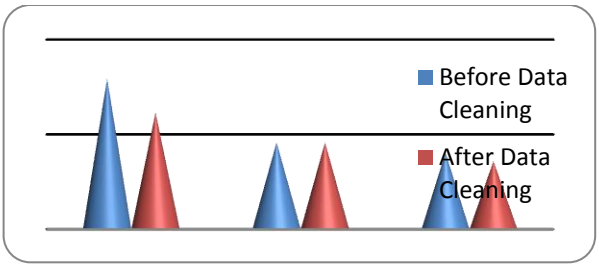

Fig. 8. Column Chart Shows Before and After Cleaning Change

An exemplary explanation of user identification method with IP and agent field and time constraint is shown in Table 4. If the IP address of next request is same then the user agent is checked and if both are same then time constraint is checked. New user is created when any of the given three conditions get are not satisfied. In this example we have taken an hour as time constraint. Whenever the difference between the time of the first request of that current user and the current processed request in log file is greater than an hour, the new user will be created. In Table 4 the user U2 and U4 are created in third and seventh row respectively because in third row, IP is same but the User agent is different and in seventh row, IP and agent both are different. In sixth row, the user U3 has been created, because the time difference between the first request of U2 in third row and the sixth row was greater than an hour, where as the IP and agent field was same with the fifth row.

TABLE IV. SHOWS THE WORKING OF USER SESSION IDENTIFICATION METHOD

	IP	Time	User agent	User Identification
	1.2.3.4	11:12:13	P	U1
	1.2.3.4	11:16:13	P	U1
	1.2.3.4	11:16:13	Q	U2
	1.2.3.4	11:17:18	Q	U2
	1.2.3.4	11:17:18	Q	U2
	1.2.3.4	12:17:18	Q	U3
	1.2.3.5	12:17:18	P	U4

NASA log file is a common log file; therefore unique users are identified on the basis of IP and time because it does not contain any information in agent field. The number of unique users identified using the above discussed concept for each input log file is shown in Table 6.

C. Phase-III: Session Identification

Correct identification of sessions is an important step in pre-processing data from web logs. A user session can be defined as a set of pages visited by the same user within the duration of one particular visit to a web-site or; a session of a particular user means how much time that user is connected to a particular website. A user may have a single or multiple sessions during a period. Once a user is identified, the click stream of each user is portioned into logical sessions. The method of portioning into sessions is called as sessionization. In this paper three methods of sessionization have been implemented and the results are compared in the end. These three methods of sessionization arenamely time-oriented,

Referrer-Time and Semantically-Time-Referrer are discussed below.

1) Time Oriented Session Identification: Time oriented session identification method is a traditional method which considers temporal boundaries such as a maximum session length which is normally taken as 30 minutes or the maximum time allowable for each page view, which is normally taken as 10 minutes. There are several other ways to calculate the time asdiscussed in related work. The exemplary explanation of time heuristic is shown in Table 4. According to this method, whenever the stay time (10min) or complete session time (30min) gets greater than the threshold value, it has created new session. In third and sixth row the time gap becomes more then 10 minutes, due to this new session S2 and S3 are created, whereas if we consider the navigation path, the session should remain same. Therefore, this is the biggest limitation of this method as most of the time one session is divided in to more than one session or many sessions comes into one session. Table 5 shows the results of Time-heuristic sessionization for three different log files which has been taken as input. Results are discussed in the section of results and discussion.

2) Proposed Referrer-Time Oriented Session Identification: Traditional referrer based method creates sessions using only the referrer field of log file. In this method, immediate previous access is checked to create the session if in the current request the referrer field contain the requested URL of the previous request. In our proposed method, we have combined the time based and the referrer based methods and some additional checks to get better results.

Algorithm 1: Referrer-Time Oriented Sessionization

Input: $U= (U_1, U_2, U_3 \ldots U_n\}, U_1= \{R_1, R_2, \ldots\ldots R_m\}$
$S=0$
$LOOP_1$ *from* $U1 \text{---} U_n$
  $LOOP_2$ *for every Request of User* U_i
   IF (referrer! = " - ")
    IF :the referrer field contain the Requested URL of previous entry then CONTINUE;
    ELSE IF: referrer is same with the referrer of previous request then CONTINUE;
    ELSE IF: referrer contains any search engine than create new session, S++;
    ELSE IF: the time interval between current request and previous request is <= threshold page access time then CONTINUE ;
    ELSE IF: the time interval between first request of this session and current request ($TR_m - TR_1$) *is <=threshold session time then CONTINUE;*
    ELSE : create new session S++;
  ELSE
    IF: the time interval between current request and previous request is <= threshold page access time then CONTINUE ;
    ELSE IF: the time interval between first request of this session and current request ($TR_m - TR_1$) *is <=threshold session time then CONTINUE;*
    ELSE : create new session S++;
  END $LOOP_2$
END $LOOP_1$
Output: $U= \{U_1= (s_1, s_2 \ldots s_p), U2= (s_1, s_2 \ldots s_q), \ldots\ldots. U_n= (s_1, s_2 \ldots s_r)\}$

The Table 5 shows the example of working of the above referrer-time based algorithm. All the conditions are covered in this example and it has created three sessions. Till fifth row the session is S1 but at time 10:50 no condition of referrer matches and even time exceeds. So the new session has created. In sixth row or entry neither referrer field matches with the URL field of fifth row, nor the URL equal to URL of previous request, nor the referrer field of sixth row matches the referrer field of fifth row . At tenth row the S3 session has been created because the referrer contain the search engine. Last at S4 session, no condition matches and the time becomes greater than 10 minutes, so session has incremented.

TABLE V. EXEMPLARY TREATMENT OF ALL THREE SESSIONIZATION METHODS

Time	URL	Referrer	Time	Time-Referrer	Semantically-Time-Referrer
10:24	/	-	S1	S1	S1
10:25	/admissions/home	/	S1	S1	S1
10:36	/admissions/home/10+2	/admissions/home	S2	S1	S1
10:39	/admissions/home/clarification	/admissions/home	S2	S1	S1
10:39	/admissions/home/prospectus	/	S2	S1	S1
10:50	/admissions/home/handbooks	/admissions/home	S3	S2	S1
10:53	/admissions/home/prospectus	/admissions/home/handbooks	S3	S2	S1
10:53	/	/admissions/home/prospectus	S3	S2	S1
10:57	/admissions/home	/	S3	S2	S1
11:06	/	Google	S4	S3	S2
11:17	/gurmatgyanonline/	-	S5	S4	S3

3) Proposed Semantically-Time-Referrer Sessionization: Due to some limitations of time and referrer based methods, we have proposed a new method, which will find the sessions based on semantic, time and referrer check. Before creating a new session at any level, first it will check all the mentioned conditions. If none of conditions gets satisfied, only then it will create a new session. It will not create a new session by just checking one of semantic, time and referrer condition. For semantic check, it will calculate the semantic difference between two URL's using the following equations [18].

$$d_{path(url1,url2)} = \frac{\max\{\ln(url1), \ln(url2)\} - I(url1, url2)}{\max\{\ln(url1), \ln(url2)\}} \quad \ldots\ldots\ldots(1)$$

Ln(url) is the function that returns the length of the URL, I(url) is the index value of first character where the mismatch of string starts, max function will tell you which of the URL have maximum length. The whole result is stored in variable

d_{path} (url1, url2). For example if the url1="punjabiuniversity.ac.in/admissions/home" and url2= "punjabiuniversity/sports/home". The length of string or url1 is 38 and url2 is 34 and index value where the string change from the starting is at 25. So the result will be (38-25)/38=0.34. We will compare the result with our threshold value, which is assigned according to the URL in the log file.

Algorithm 2: Semantically-Time-Referrer Sessionization

Input: $U= (U_1, U_2, U_3 \ldots U_n\}, U_1= \{R_1, R_2, \ldots\ldots R_m\}$
$S=0$
$LOOP_1$ *from* $U_1 \text{---} U_n$
  $LOOP_2$ *for every Request of User* U_i
   IF (referrer! = " - ")
    IF :the referrer field contain the Requested URL of previous entry then CONTINUE;
    ELSE IF: referrer is same with the referrer of previous request then CONTINUE;
    ELSE IF: referrer contains any search engine than create new session, S++;
    ELSE IF: semantic difference between current referrer and previous referrer is<=threshold value then CONTINUE
    ELSE IF: semantic difference between current referrer and previous URL is<=threshold value then CONTINUE
    ELSE IF: semantic difference between current URL and previous URL is<=threshold value then CONTINUE
    ELSE IF: the time interval between current request and previous request is <= threshold page access time then CONTINUE ;
    ELSE IF: the time interval between first request of this session and current request (TR_m -TR_1) *is <=threshold session time then CONTINUE;*
    ELSE : create new session S++;
  ELSE
    IF: semantic difference between current referrer and previous URL is<=threshold value then CONTINUE
    ELSE IF: the time interval between current request and previous request is <= threshold page access time then CONTINUE ;
    ELSE IF: the time interval between first request of this session and current request (TR_m -TR_1) *is <=threshold session time then CONTINUE;*
    ELSE : create new session S++;
  END $LOOP_2$
END $LOOP_1$
Output: $U= \{U_1= (s_1, s_2 \ldots s_p), U2= (s_1, s_2 \ldots s_q), \ldots\ldots U_n= (s_1, s_2 \ldots s_r)\}$

The example of our proposed semantically-time-referrer heuristic method is shown in Table 5. In the beginning it will check the three conditions of referrer. First, if the referrer field of current entry matches with the URL field of the previous entry and second, if the referrer of current entry is same as the referrer of previous entry and third if the referrer field contain any search engine. Next there are three semantic checks, which are applied in sequence to find the semantic difference between current and previous URLs, referrers and current referrer and previous URL. If at any position the condition gets satisfied, the algorithm will move to next entry of log. Otherwise it will move on checking the conditions and if no condition matches, in the end it will check the time constraint. New session gets created only if all the condition of referrer, semantic and time is failed. Table 5 shows that till the 9th row the conditions are satisfied, the session remains S1 but at 10^{th} row it has been changed to S2, because the third condition of referrer check fails, it contain search engine in referrer field. Similarly at eleventh row the referrer is blank, so the only semantic check is between current URL i.e (/gurmatgyanonline/) and previous URL (/). It has given the negative result, so the session is incremented from S2 to S3.

TABLE VI. DETAILS OF USER AND SESSION IDENTIFICATIONS FOR LOG FILES

	Colg	Guitar	NASA
Total Log Entries	7956	4589	4045
Entries in Cleaned log	6522 (77.46 %)	4589 (100%)	3601 (89.02)
Time Period of Log	5days	79days	1day
Average Access Time of Web Pages	5.05 sec	8.91 sec	1.03 sec
Unique URLs in Log	2555	139	702
Unique IPs	835	2555	414
Users Identification	1076	2650	450
Time Heuristic Sessionization	1780	3213	645
Proposed Time-Referrer Sessionization	1587 (89.15)%	3070 (95.54)%	645 (100)%
Proposed Semantically-Time-Referrer Sessionization	1376 (86.70)%	2902 (90.32)%	480 (74.41)%

This paper has implemented the above proposed algorithm and shows the results of three log file taken for analysis in Table 6 discussed in next section. Similarly the line chart in Figure 9 shows the changes at every phase of pre-processing for every log file.

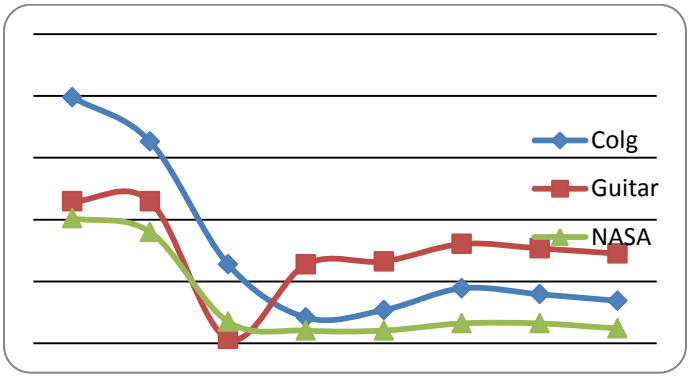

Fig. 9. Shows the effects of every phase of pre-processing

TABLE VII. SHOWS COMPARISON OF EXISTING AND PROPOSED METHODS WITH TRUE SESSIONS

Methods	Colg		Guitar		NASA	
	Obtained Sessions	%age	Obtained Sessions	%age	Obtained Sessions	%age
Time Based	15	68.18%	16	72.72 %	17	77.27%
Proposed Time-Referrer Based	16	72.72%	18	81.81 %	17	77.27%
Proposed Semantically-Time-Referrer	18	81.81%	19	86.36 %	19	86.36%

VII. RESULTS AND DISCUSSION

Implementation of all the techniques has been done in #c using .net software. Three log files of different format have been used for testing namely colg, guitar and NASA. Table 1 shows details of these log files before cleaning. It shows the size, time period, number of entries, format of log file, type of log file and total data transferred in KB. Data cleaning is performed in the first phase of pre-processing, irrelevant or noisy data is removed from these three log files shown in Table 2. Table shows the number of entries for robots, failure status code, multimedia, text files, and other than GET method requests. Detailed number of requests by multimedia, text files and method requests is shown in Table 3. Figure 8 shows the change in log file before and after cleaning in form of bar chart. Table 4 shows the unique number of users called as user identification using IP and agent field for three different log files. Number of unique IPs and unique URLs of log file shows the access rate, no of different pages accessed by the users during particular time period. It also shows the comparison of the three different sessionization methods namely Time, Referrer-time and Semantically-time-referrer based heuristics on three different log files. In time heuristic sessionization we have taken the standard threshold value 10 min for consecutive page access time and 30 min for maximum session time. For time-referrer, the results mainly depends upon the referrer field. Otherwise in case there is no referrer, the algorithm will behave like time-heuristic. In our proposed algorithm named time-referrer-semantical sessionization the sessions depends upon all three factors namely time, referrer and semantics. The new session is created only when all three conditions are failed. Time-oriented heuristics estimate denser sessionization than two other methods. The referrer-time and semantically-time-referrer sessionization methods decreased the number of sessions to 89.15% and 86.70% respectively for colg log file, 95.54% and 90.32% respectively for guitar log file and zero percent and 74.41% respectively for NASA log file. Tested log files are large in size and contain huge data. So it is difficult to find the accuracy of algorithms on whole data. Due to this we have taken small data to test the performance of proposed algorithms. The small testing data contain 20 true sessions which are counted manually. Every true session contain more than two entries. Table 7 shows the performance of existing and proposed algorithms on 22 true sessions for all three log files. Results clearly show that using the Semantically-time-referred method, the accuracy substantially increases to 81.81%, 86.36% and 86.36% for colg, guitar and NASA log file respectively. The results of any log file also depends upon the size of website, usage of website by customers, time period of log file and number of different IP addresses. Line chart in Figure 9 shows the changes in every log file during each phase of pre-processing.

VIII. CONTRIBUTIONS

- This paper proposed the two sessionization methods out of which semantically-time referrer outperforms the other approaches.
- The pre-processing model presented in this paper can process text or excel log file irrespective of the format.
- Semantics concept has been used to deduce meaningful results.
- Semantically-time-referrer method deals with the empty referrer requests.
- Semantically-time-referrer method has reduced the complexity of the existing methods and increased their efficiency.

Hence the proposed model is computationally simple and easy to deploy.

IX. CONCLUSION

Web Log files records the activity information whenever a web user submits a request to a Web Server. This paper presents the implementation results of each phase of pre-processing as concluded from our research i.e. data cleaning, user identification and session identification from raw log data. This paper proposed two session identification methods including referrer-time based and semantically-time-referrer based methods. In addition to the traditional data cleaning algorithm, our algorithm is also cleaning the text file and requests in which transferred bytes are nearly zero. In comparison to the traditional time and referrer based heuristic, the referrer-time based heuristic improves the performance from two aspects: First, by not only comparing the referrer of current request with URL of previous request but also comparing the referrer field of current and previous requests to form an actual session. Second, a time component adds a dynamic time frame. If the referrer conditions are not satisfied, rather than directly breaking the session it will check the time limit which avoids the generation of an unreasonably long session. In comparison to traditional method, our referrer-time proposed method, the novelty of semantically-time-referrer based heuristic is that by introducing the time referrer and semantics concepts, it not only improves the authenticity but also improves the efficiency of session identification from two aspects: First is even if the request has empty referrer field, it will still calculate the semantic difference between the requested URL and the previous URL. Second, time component adds a dynamic time frame. That is, if all the conditions are not satisfied, then it will check the time constraint which avoids the generation of unreasonably long sessions. Using semantically-time-referrer method 1376, 2902

and 480 sessions have been identified for three input log files (colg, guitar and NASA respectively). Comparing the results of the experiment for 22 actual true sessions of the given data showed that using Semantically-time-referred method, the accuracy level increased to 81.81%,86.36% and 86.36% for colg, guitar and NASA log file respectively. In Future this work can be extended to extract user patterns by applying web mining techniques on identified sessions by semantically-time-referrer method.

ACKNOWLEDGMENT

The authors are greatly thankful to the department of Computer Engineering, Punjabi University, Patiala, for providing excellent lab facilities that make this work possible.

REFERENCES

[1] Marathe Dagadu Mitharam, "Preprocessing in Web Usage mining",International Journal of Scientific & Engineering Research, ISSN 2229-5518, vol. 3, Issue 2, February 2012.

[2] Cooley, R., "Web Usage Mining: Discovery and Application of Interesting Patterns from Web data", http://citeseer.nj.nec.com/426030.html. 2010

[3] The int Aye: Web Log Cleaning for Mining of Web Usage Patterns, IEEE, 2011.

[4] G.Castellano, A.M.Fanelli, M.A.Trsello, "Log data preparation for mining Web usage patterns" IADIS International Conference Applied Computing, pp. 371-378, 2007.

[5] Priyanka Patil and Vjwala Patil, "Preprocessing of webserver Log File for Web mining" World Journal of science and technology, vol. 2, pp. 14-18, 2012

[6] Er. Romil V Patel, Dheeraj Kumar Singh, "Pattern Classification based on Web Usage Mining using Neural Network Techniques"International journal of computer applications(0975-8887), vol. 77, No.21 , June 2013

[7] Murti Punjani, Mr. Vinit Kumar Gupta "A survey on data preprocessing in web usage mining" IOSR Journal as computer engineering, e-ISSN:2278-0661,P-ISSN:2278-8727, vol. 9, Issue 4, pp. 76-79, 2013

[8] Hongzhou Sha, Tingwen Liu,Peng Qin,Yong Sun,Qingyun Liu, "EPLogCleaner: Improving Data Quality of Enterprise Proxy Logs for Efficient Web Usage Mining" Procedia computer Science 17, pp. 812-818, 2013

[9] Doru Tanasa ,Brigitte Trousse, "Advanced Data Preprocessing for Intersites Web Usage Mining" Enhancing information,IEEE Intelligent System , pp 59-65, 2004.

[10] Muskan, Kanwal Garg, "An Efficient Algorithm for Data Cleaning of Web Logs with Spider Navigation Removal"International Journal of Computer Application(2250-1797) vol. 6, No.3, May-June 2016.

[11] Sheetal A.Raiyani,Shailendra Jain, "Efficient Preprocessing technique using Web Log Mining"International Journal of Advancement in Research & Technology, vol. 1,Issue 6, November 2012, ISSN2278-7763.

[12] Aswin G. Raiyani, Sheetal S. Pandya, "Discovering User Identification Mining Technique for Preprocessing Log Data", ISSN: 0975 – 6760, vol 2, Issue 2, pp. 477-482, Nov 12 to Oct 13.

[13] Li Chaofeng "Research and Development of Data Preprocessing in Web Usage Mining" International Journal of Computer application 2011.

[14] F.M.Facca, P.L.Lanzi,"Mining Interesting Knowledge from WebLogs: A Survey" Data and Knowledge Enggineering , pp 225-241, 2005.

[15] Fang Yuankang, Huang Zhiqiu, " A Session Identification Algorithm Based on Frame page and Pagethreshold",IEEE 2010.

[16] Yongyao Jiang , Yun Li, Chaowei Yang, Edward M. Armstrong , Thomas Huang and David Moroni, "Reconstructing Sessions from Data Discovery and Access Logs to Build a Semantic Knowledge Base for Improving Data Discovery" ISPRS International Journal of Geo-Information 2016

[17] L.K.Joshila Grace,V. Maheswari, Dhinaharan Nagamalai, " Analysis of Web Logs and Web User in Web Mining", International Journal of Network Security & its applications, vol.3, No.1, January 2011.

[18] Jason J. Jung, "Semantic Preprocessing of Web Request Streams for Web Usage Mining", Journal of Universal Computer Science, vol.11, no.8, pp. 1383-1396, 2005.

[19] S.Kaviarasa, K.Hemapriya, K.Gopinath,"Semantic Web Usage Mining Techniques for Predicting User's Navigation Requests"International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, Issue 5, May 2015.

[20] Sharma, N.;Makhika, P.Web Usage Mining: A Novel Approach for Web User session Construction", Glob.J.Comput. Sci. Technol, vol. 15,Issue 3, pp. 23-27, 2015.

[21] Lara D. Carledge, James E. Pitkow, "Characterizing browsing strategies on the world wide web", Computer Networks and ISDN Systems,vol. 27,Issue 6, pp.1065-1073, April 1995.