ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Real-Time Data Pre-processing Technique in Web Usage Mining

Seema Nimbalkar[1], Dr. Suhas Patil[2]

Assistant Professor, Dept. of Computer Science, Abeda Inamdar Senior College, Pune, India[1]

Professor, Dept. of Computer Engineering, Bharati Vidyapeeth University College of Engineering, Pune, India[2]

ABSTRACT: The application of data mining techniques to discover usage pattern from the web, which helps to better serve the needs of web based application is known as web usage mining. Web usage mining has become the main subject of intensive research, because of its extensive potential for personalized services, web-site improvement and usage characterization. Prior to applying any data mining algorithms to the data collected from server logs, there are several pre-processing tasks that must be performed. The successful application of data mining techniques to Web log data is highly dependent on the correct application of these pre-processing tasks. Data pre-processing is the most time consuming and computationally intensive step in the Web usage mining. The main problem in web usage mining is the massive quantity of Web usage data and its low quality; hence there is a need for further research to improve the performance and effectiveness of data pre-processing which is dependent on the quality of data in the log files. The traditional data pre-processing techniques applied on web log data are more time consuming and not effective. So there is need to apply real time data pre-processing techniques which require less time for processing of web log data. This paper discusses a technique of real-time data pre-processing consisting of data cleaning, semantic enrichment of web logs, user session and transaction identification to improve the performance and efficiency of web usage mining.

KEYWORDS: Real-time Data Pre-processing; Real-time Data Cleaning; Semantic; Web Usage Mining, Web Log.

I. INTRODUCTION

The World Wide Web continues to grow at an astounding rate in both the sheer volume of traffic and size & complexity of the web sites. As a result there is a challenging task for web master to understand the needs of users and keep their attention in the web-site. Analysing the user's navigational behaviour can help to improve the web-site design, performance and achieve personalization of the user. Web Usage Mining applies data mining procedures to analyse user access of the web sites. As with any KDD (Knowledge Discovery from Databases) process, Web Usage Mining consists of three main steps: Pre-processing, Knowledge Discovery & Analysis. The quantity of web usage data to be analysed and its low quality are the principal problems in Web Usage Mining. When applied to these data, the classic algorithms don't give proper results in terms of behaviour of the web-site's users. According to researchers two-thirds of data mining analysts consider that data cleaning and data preparation consume more than 80% of the total analysis time [2]. The quality of results from Data Pre-processing influences the result of pattern discovery and analysis [1]. Thus data pre-processing plays vital role in the whole web usage mining process and is the key of its quality.

A. DATA PRE-PROCESSING:

In web usage mining the pre-processing stage goal is to change the raw click stream information into a set of user profile. From a navigational view point, every profile seizes delimited sequence or set of page views that indicates a session of user. This sessionized data can be used as input for different algorithms of data mining or further abstracted and transformed. The web usage data pre-processing has several distinct challenges which leads to different heuristic techniques and algorithms for pre-processing tasks such as data cleaning and fusion, session and user identification, page view identification. The successful application of data mining techniques to web usage information is dependent highly on proper application of pre-processing tasks. This stage includes pre-processing of gathered data from numerous sources and transforming them into a form applicable for applying the operations of data mining. The data pre-processing purpose is to provide structural, integrated and reliable data source to pattern discovery. It comprises of 4 processes described below: Figure 1 shows the data pre-processing.

DOI: 10.15680/IJIRCCE.2016. 0409048

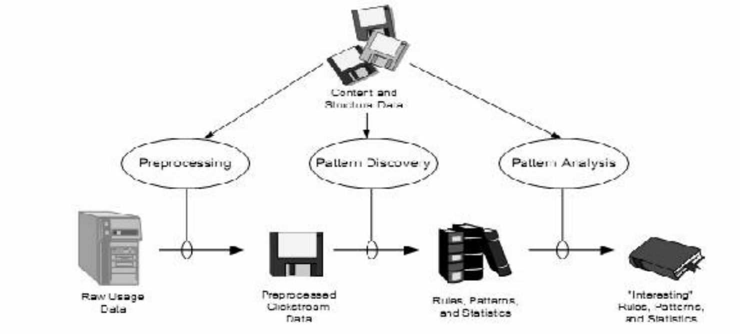

Content and Structure Data
Preprocessing
Pattern Discovery
Pattern Analysis
Raw Usage Data
Preprocessed Clickstream Data
Rules, Patterns, and Statistics
"Interesting" Rules, Patterns, and Statistics

Figure 1: Data Pre-processing

1.1.1 Data cleaning:

The web log file is the source of data to the process of data cleaning. Data cleaning is often web sites specific. The data cleaning purpose is to remove irrelevant record or items from the log file. The data cleaning task is to delete unrelated data to mining namely images in the format of JPEG, jpg and GIF, error response, JavaScript, Cascading Style Sheets and robot request. Data cleaning also involves the removals of references due to crawler information navigation.

1.1.2 User identification:

User identification is performed after data cleaning. The website visited by users is identified by the process of user identification. This is performed with the support of user agent and IP address. The web usage analysis does not require knowing about the identity of user. However it is essential to distinguish among different users. Since a user may visit a site more than once, the server logs register numerous sessions for every user. The user activity record phrase is used to identify the logged activities sequence of similar user. In the absence of authentication mechanisms, the most widespread approach to differentiate the distinct visitors is done by using client side cookies. It is not possible to use cookies for entire websites due to the concerns of privacy where the client side cookies are disabled by the users. Without client side cookies or user authentication is not possible to recognize distinct users accurately. Another approach can be the use of IP addresses for user identification. Users with different IP addresses can be treated as unique but it is not the perfect solution as ISPs assign rotating addresses to users.

1.1.3 Session identification:

The session is a time period between user's login and logout process. The user visits several pages during this time. Session is used to predict the page sequences and trace the activity of user. A user session can be defined as the set of pages visited by similar user within particular duration of one specific visit to a website. A user may have multiple or single sessions during time period. Once the user has been recognized, then every user click stream is divided into the logical clusters. The portioning method in the sessions is known as session reconstruction or sessionization. Session reconstruction can be categorized into two main approaches and they are navigation oriented approach and time oriented approach. The methods used for user identification to a certain extent, can be used for session identification.

1.1.4 Path completion:

The purpose of path completion is to identify user's travel pattern and also the missing pages in path where the user access must to be appended. It is possible to recognize several missed pages by cached versions and proxy servers of pages used by client. So the step of path completion is undertaken to recognize missing pages. The path set is incomplete accessed pages in a session of user and it is retrieved from each set of user session.

B. PATTERN DISCOVERY AND ANALYSIS:

In pattern discovery, the application of different data mining techniques process data like association, statistical analysis, pattern matching, clustering and so on. In pattern analysis, the patterns are discovered from web logs where uninteresting norms are filtered out. The analysis is performed using mechanism of knowledge query such as data cubes or SQL to perform the operations of online analytical processing. The pre-processed data is considered for knowledge extraction algorithm, application based on data mining algorithms, artificial intelligence, information theory and psychology. Most of the systems evolved for web usage mining process have mentioned several algorithms predicting maximal forward reference, large reference sequence to examine the user's traversal path. Various algorithms of mining like association rules, path mining, clustering, classification and sequential patterns are used for efficient process of web usage mining. It wholly relies on need of analyst to decide which techniques of mining to make use of. The last step in web usage mining process is to filter out uninteresting rules of patterns from the set found in pattern discovery phase. This can be used for changes in website, web personalization and/or system improvement. The similar techniques used for pattern analysis are online analytical processing techniques, visualization techniques, usability analysis, data and knowledge querying.

II. Related work

The following literature review will exclusively and intensively explain about the studies that dealt with data pre-processing techniques in web usage mining.

Data cleaning, User and Session Identification, Path completion, Transaction identification are the steps of Data pre-processing [1-10]. The web server logs are in CLF (Common Log Format) or ECLF (Extended Log Format) format recommended by World Wide Web consortium [2, 6]. Data cleaning involves removal of irrelevant and redundant items from the web server logs -

i) Accessorial resources - Removal of image, CSS and JavaScript file entries from the log.
ii) Web Robot (spiders) requests - Web robots are software tools that scan a web-site to extract its content. Normally a web robot identifies itself by using log file's user agent field [2]. There are heuristics suggested by various researches to look for web robots requests.
iii) Error requests - Error log entries are useless for mining and can be removed by checking the status of request.

User identification is the process to distinguish the log entries among different users [4]. For web-sites requiring user registration, the log file contains user login and this can be used for User Identification. Cooley et al. [3] have proposed some heuristics that can be used to identify unique users based on the IP address and user agent.

Session Identification is to divide the page access of each user into individual sessions [3]. Identification user sessions from the log file isn't simple due to existence proxy servers, dynamic addresses and user privacy issues. Bettina Berendt et al. [10] have proposed referrer-based and time-based heuristic methods to accomplish Session Identification. Yan Li et al. [5] have proposed an approach to combine the two heuristics to get more accurate results.

Path completion is the process to identify the missing pages in the users travel navigation path due to browser caching and proxy servers. Cooley et al. [3] have suggested a referrer-based method for Path completion. Use MFR (Maximum Forward References) algorithm to identify Travel-Path transactions. Reference Length algorithm can be used to identify auxiliary and content pages.

The issues and difficulties in data cleaning have not been discussed in detail by the authors in their study. The issue in the identification of user has been denoted as a significant issue since it is necessary to differentiate the IP address of each individual. The main issue author stated in this study is about 'personal information' login information that many users have been ignoring for accessing data (i.e. without registration) henceforth finding the user accessing relevant information or session is a tedious task which would take minutes to hours making the process difficult. Hence the authors had proposed DUI (Distinct User Identification) algorithm to retrieve user identification [18].

According to the authors, data cleaning under their proposed algorithm would require lesser time to process rather than the traditional algorithms. Difficulties in User and Session identification have been studied by the authors and they found that web log data is essential to identify the users and their information. KNN and PCA algorithm was proposed by the authors to differentiate data and to map filter information for faster and pattern discovery purposes. Authors had mentioned that the data pre-processing process consumes lot of time and hence altering algorithm would be an effective measure [19].

In this paper the authors have studied about the issues in data cleaning and the time required for it. According to them in order to remove the errors (data cleaning), basically the algorithm has to check for HTTP status code and the records found under the status codes of 200 or over 299 will be removed. The user identification along with the session identification has a different issue where the users sometimes access information without logging in with their registered information. The web log consists of data accessed for each session as per the web pages time-oriented or sometimes the structure-oriented limitations. The authors had studied about the time consumption for overall process of data pre-processing and they had found that: accessing raw web log, cleaning the data, finding the users or the unique users session are significant factors to be considered for time consumption and hence they proposed a better and efficient algorithm: Distinct User Identification [20].

The authors studied about data cleaning and they had stated that it is quite time consuming process where the data has to be cleaned especially if the data is in the form of pictures, videos, audios, and so on apart from text/ content. On contrary the authors studied about the user identification and session identification, however they didn't study in-depth about the issues in the processes. According to them web log mining for identifying users will consume longer time. Overall as per their view the data processing consumes more time since it has to overcome: data cleaning, user identification, session identification, path complement and transaction identification. To resolve this issue they formulated an algorithm where the traditional algorithm and a dynamic algorithm are collaborated into one to refine the session identification time-out issue and accessing information faster to identify the user through web log mining [21].

As per the authors' view the web log for GIF, CSS, JPEG in the URI field will consume more time for data cleaning where the algorithm has to examine from HTTP status codes. In the status code field, if found status error is under 200 or over 299, then the errors are removed through the structured algorithm. The time for identifying the user and for session identification varies according to the authors. The user identification could be done through session identification however the session identification has to found through examining the IP address, use of an operating system and the browser. Hence session identification consumes more time than user identification. Hence they proposed AxisLog Miner tool supported by Distinct User Identification algorithm to minimize the time consumption [22].

The authors mentioned that the user identification processing time and session identification filtering time had been reduced through their proposed algorithm (Modified Session Identification technique). According to them the log files and irrelevant entries has to be removed in the pre-processing stage [23].

III.PROPOSED SYSTEM

A. REAL-TIME DATA CLEANING:

The research work proposes a technique to perform the data cleaning of the web server logs in real-time. The technique involves logging only the essential log entries in the original access.log file and all the non-essential log entries like images, error requests, javascript, CSS etc. in the access_redundant.log file. It also does semantic enrichment of the logs. The technique has been implemented using the Apache directives and mod_perl handlers. Figure 2 shows real-time data cleaning.

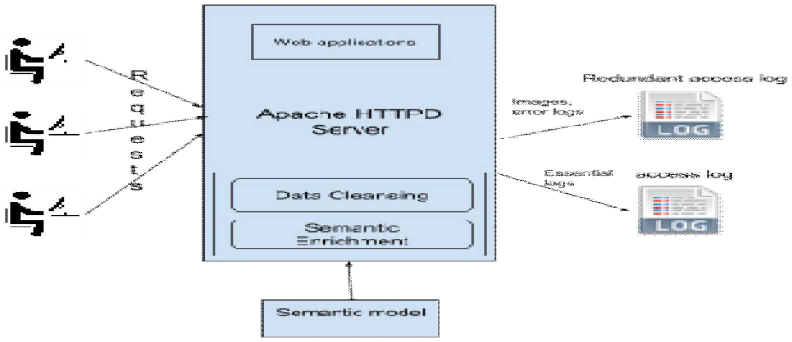


Figure 2: Real-Time Data Cleaning

The proposed system will perform the following functions -

1. The irrelevant or redundant web requests will be logged in a different log file named access_redundant.log file.
2. The essential Web requests for web usage mining will be logged into the web access log file.
3. Enrich the semantic information of the pages that are part of the relevant web requests in the acess.log file.

The Apache conditional logging directives are used to perform the real-time data cleaning process. All redundant or non-essential requests like multi-media request, internal Apache requests with error responses will be logged in the access_redundant.log file. The conditional logging directives used are - SetEnvIf, SetEnvIfNoCase and ResponseSetEnvIfPlus.

Algorithm:

Input: HTTP Requests and Responses
Output: access.log and access_redundant.log files
Step 1: Read the input http request data.
Step 2: If the request is for javascript, CSS, image, icon or any other multi-media file then log the request in the redundant access log file (access_redudant.log).
Step 3: If there is a client or server error response for the request e.g. 404 or 500 then log the request in the redundant access log file.
Step 4: If it is 'internal dummy connection' request then log the request in the redundant access log file.
Step 5: Lookup the request URI in the semantic model and get the semantic attributes for the request.
Step 6: Read the semantic attribute values from the http request and enrich the log entry for the request with the semantic information.
Step 7: The request log entry is also enriched with session id, user id and time required for processing the HTTP request.
Step 8: Repeat the above steps for each http request.

Modified Access Log format

The LogFormat for the access.log file is defined in the main server configuration file as below -

```
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referrer}i\" \"%{User-Agent}i\"\"%{semantics}e\" \"%{sid}C\"
\"%{userid}e\" \"%{ttime}e\""combined
```

i. Added “%{semantic}e” to LogFormat to include the semantic information of the request in the logs. The semantic information includes the user action namely – add in or remove from shopping cart, checkout and product Id.
ii. Added “%{sid}C” to the LogFormat to include the session Id from the 'sid' cookie in the logs.
iii. Added “%{userid}e” to the LogFormat to include the user Id in the logs.
iv. Added “%{ttime}e” to the LogFormat to include the time required to process the request.

New fields in Web log file	access.log	
	Field Name	**Purpose**
User Identification	userid	User Id attribute
Session Identification	sid	Session Id attribute
Product Identification	pID	Product Id attribute (semantic information)
Action	add_cart, remove_product, cart_remove, success	To identify action of the user (semantic information)
Time	ttime	Time required to process the request

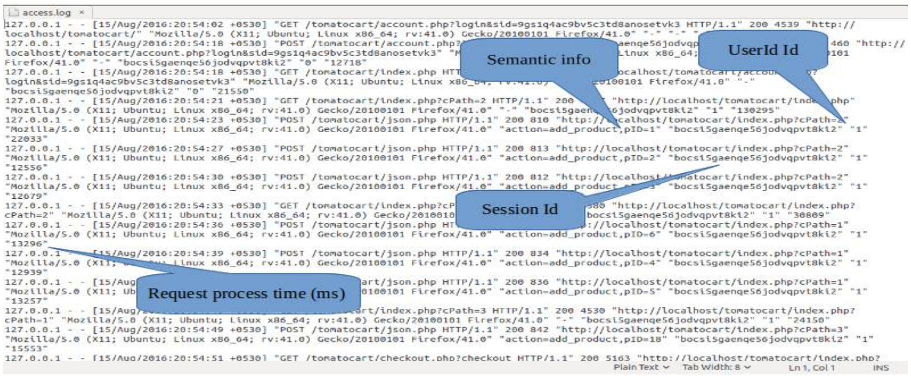
Table 1: New Web Log Fields

To validate the efficiency of proposed methodology, an experiment is conducted using the log files of Tomato Cart application. The different log files collected from Apache Web server were analysed. The results showed that our methodology reduced the Web access log file down to 60% of the initial size. The Table 2 shows comparison matrix of Real-Time Data Cleaning.

Comparison Factor	Algorithm	
	access.log	access_redudant.log
Multimedia Files	NA	Yes
HTTP status code	200	200, 400, 500 Series
HTTP Method	GET	GET, POST
Semantic info	Yes	NA
Percentage of Reduction	60%	NA

Table 2: Comparison Matrix of Real-Time Data Cleaning

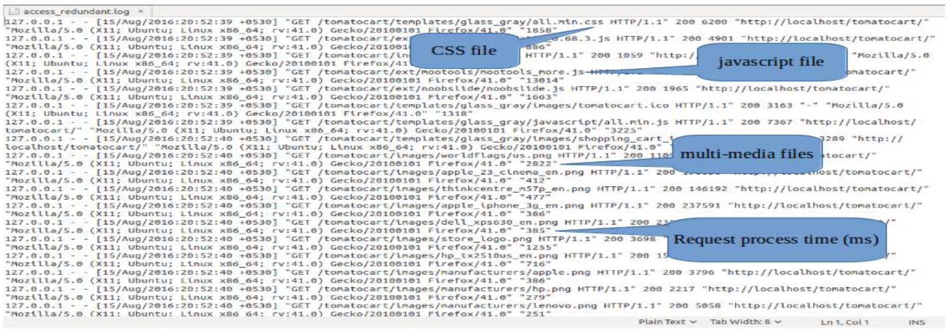
The Figure-3 shows the access.log file that was generated by the real-time data cleaning algorithm in the proposed system. The access.log file contains the essential log entries that are required for further data pre-processing and analysis. The log entries contain the userId, sessionId, semantic information of the associated request and the time required for processing the request.



```
127.0.0.1 - - [15/Aug/2016:20:54:02 +0530] "GET /tomatocart/account.php?login&sid=9gs1q4ac9bv5c3td8anosetvk3 HTTP/1.1" 200 4539 "http://
localhost/tomatocart/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "-" "-" "
127.0.0.1 - - [15/Aug/2016:20:54:18 +0530] "POST /tomatocart/account.php?                    aenqe56jodvqp              460 "http://
localhost/tomatocart/account.php?login&sid=9gs1q4ac9bv5c3td8anosetvk3" "M                    Linux x86_64;              101
Firefox/41.0" "-" "bocsi5gaenqe56jodvqpvt8ki2" "0" "12718"
127.0.0.1 - - [15/Aug/2016:20:54:18 +0530] "GET /tomatocart/index.php HTT                    ocalhost/tomatocart/accou      ?
login&sid=9gs1q4ac9bv5c3td8anosetvk3" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0)      0100101 Firefox/41.0" "-"
"bocsi5gaenqe56jodvqpvt8ki2" "0" "21550"
127.0.0.1 - - [15/Aug/2016:20:54:21 +0530] "GET /tomatocart/index.php?cPath=2 HTTP/1.1" 200       "http://localhost/tomatocart/inde  php"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "-" "bocsi5gaen   6jodvqpvt8ki2" "1" "130295"
127.0.0.1 - - [15/Aug/2016:20:54:23 +0530] "POST /tomatocart/json.php HTTP/1.1" 200 810 "http://  calhost/tomatocart/index.php?cPath=2"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product,pID=1" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"22033"
127.0.0.1 - - [15/Aug/2016:20:54:27 +0530] "POST /tomatocart/json.php HTTP/1.1" 200 813 "http://localhost/tomatocart/index.php?cPath=2"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product,pID=2" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"12556"
127.0.0.1 - - [15/Aug/2016:20:54:30 +0530] "POST /tomatocart/json.php HTTP/1.1" 200 812 "http://localhost/tomatocart/index.php?cPath=2"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product      "bocsi5gaenqe56jodvqpvt8ki2" "1"
"12679"
127.0.0.1 - - [15/Aug/2016:20:54:33 +0530] "GET /tomatocart/index.php?cP              580 "http://localhost/tomatocart/index.php?
cPath=2" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/2010010              bocsi5gaenqe56jodvqpvt8ki2" "1" "30809"
127.0.0.1 - - [15/Aug/2016:20:54:36 +0530] "POST /tomatocart/json.php HT              ://localhost/tomatocart/index.php?cPath=1"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product,pID=6" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"13296"
127.0.0.1     15/Aug/2016:20:54:39 +0530] "POST /tomatocart/json.php HTTP/1.1" 200 834 "http://localhost/tomatocart/index.php?cPath=1"
"Mozilla/5.0 (X11;        Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product,pID=4" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"12939"
127.0.0.1 - - [15/Aug                                       art/json.php HTTP/1.1" 200 836 "http://localhost/tomatocart/index.php?cPath=1"
"Mozilla/5.0 (X11; Ub                                       0100101 Firefox/41.0" "action=add_product,pID=5" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"13257"
127.0.0.1 - - [15/Aug/                                      art/index.php?cPath=3 HTTP/1.1" 200 4530 "http://localhost/tomatocart/index.php?
cPath=1" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "-" "bocsi5gaenqe56jodvqpvt8ki2" "1" "24150"
127.0.0.1 - - [15/Aug/2016:20:54:49 +0530] "POST /tomatocart/json.php HTTP/1.1" 200 842 "http://localhost/tomatocart/index.php?cPath=3"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "action=add_product,pID=18" "bocsi5gaenqe56jodvqpvt8ki2" "1"
"15553"
127.0.0.1 - - [15/Aug/2016:20:54:51 +0530] "GET /tomatocart/checkout.php?checkout HTTP/1.1" 200 5163 "http://localhost/tomatocart/index.php?
```

Figure 3: access.log file after Data Cleaning

The Figure-4 shows the access_redundant.log file that was generated by the real-time data cleaning algorithm in the proposed system. This log file contains all the non-essential log entries namely of multi-media requests, CSS, javascript files and internal dummy requests. These log entries are not required for further data pre-processing.



```
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/templates/glass_gray/all.min.css HTTP/1.1" 200 6200 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "1858"
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/ex                  .68.3.js HTTP/1.1" 200 4901 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/201001                  886"
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/in                  HTTP/1.1" 200 1059 "http:/                     "Mozilla/5.0
(X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/ext/mootools/mootools_more.js                                  matocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "13014"
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/ext/noobslide/noobslide.js HTTP/1.1" 200 1965 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "1663"
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/templates/glass_gray/images/tomatocart.ico HTTP/1.1" 200 3163 "-" "Mozilla/5.0
(X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "1318"
127.0.0.1 - - [15/Aug/2016:20:52:39 +0530] "GET /tomatocart/templates/glass_gray/javascript/all.min.js HTTP/1.1" 200 7367 "http://localhost/
tomatocart/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "3225"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/templates/glass_gray/images/shopping_cart_1                  3289 "http://
localhost/tomatocart/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/worldflags/us.png HTTP/1.1" 200 110                  tocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "2822"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/apple_23_cinema_en.png HTTP/1.1" 200              host/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "412"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/thinkcentre_m57p_en.png HTTP/1.1" 200 146192 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "477"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/apple_iphone_3g_en.png HTTP/1.1" 200 237591 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "386"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/dell_xps630_en.png HTTP/1.1" 200 23                        /"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "385"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/store_logo.png HTTP/1.1" 200 3698
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "1255"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/hp_tx2510us_en.png HTTP/1.1" 200 15                        /"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "716"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/manufacturers/apple.png HTTP/1.1" 200 3796 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "386"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/manufacturers/hp.png HTTP/1.1" 200 2217 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "279"
127.0.0.1 - - [15/Aug/2016:20:52:40 +0530] "GET /tomatocart/images/manufacturers/lenovo.png HTTP/1.1" 200 5058 "http://localhost/tomatocart/"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:41.0) Gecko/20100101 Firefox/41.0" "251"
```

Figure 4: access_redundant.log file after Data Cleaning

Real-Time Data Cleaning Time Evaluation:

The real-time data cleaning time evaluation is done to compute the time required to execute the Apache conditional logging directives to direct the log entries to the respective log files. There is no direct way to compute the time required for this activity, so in the proposed research work this time was computed based on the time required to handle the http requests with and without the real-time data cleaning technique.

The TomatoCart ecommerce application was used to perform user browsing activities of multiple users and sessions without the real-time data cleaning solution (Traditional Data Cleaning) and then with real-time data cleaning solution. Multiple sets of log files of varied sizes namely - log files of around 800, 1500, 2500, 3500 and 4500 entries were generated for both with and without real-time data cleaning technique. The time is recorded in the log entries using the "%{ttime}e"field added in the LogFormat directive in the Apache configuration file.

The following formula is used to compute the time required for real-time data cleaning.

$$\textit{Real-time Data cleaning time} = \sum_{r\in R} n\left(\frac{1}{n}\sum_{i=1}^{n} r_i - \frac{1}{m}\sum_{j=1}^{m} r_j\right)$$

where,

R = set of application resources requested/logged in the log files

r = application resource requested

r_i = request processing time (with real-time data cleaning) for the application resource from the log files (acess.log and access_redundant.log)

r_j = request processing time (without real-time data cleaning) for the application resource from the log file (access_full.log)

n = no of requests/log entries in the log files (with real-time data cleaning) for the resource r

m = no of requests/log entries in the log file (without real-time data cleaning)

B. TRADITIONAL DATA CLEANING:

Algorithm:

Input - access_full.log file

Output - access.log and access_redundant.log files

```
i.    Create an access_redundant.log file to collect all the redundant log entries and a access.log file to collect all
      the essential entries.
ii.   Read each log entry of the access.log file.
iii.  Parse the entry and get the request URL.
iv.   If the request URL is a javascript, CSS, error request or any multi-media then copy the log entry to write in
      the access_redundant.log file.
v.    If the log entry is for Apache 'Internal Dummy Connection' then copy the log entry to write in the
      access_redundant.log file.
vi.   If the log entry has an error response (404) then copy the log entry to write in the access_redundant.log file.
vii.  If the request URL is anything other than the above then copy the log entry to write in the access.log file.
viii. Repeat from Step ii until the end of the log file.
```

Traditional Data Cleaning Time Evaluation:

The time required for traditional data cleaning is calculated for each of the log files of varied sizes by recording the start and end time of the data cleaning process. Traditional data cleaning approach is applied on the access_full.log and the execution times are calculated for all the log file samples.

```
public static void main(String[] args) {
        long starttime = System.currentTimeMillis();
        ApacheLogFileCleaner logFileCleaner = new ApacheLogFileCleaner("/var/log/apache2/access_full.log");
        logFileCleaner.processLogFile();
        long endtime = System.currentTimeMillis();
        System.out.println("Offline Data cleaning total time = "  +  (endtime - starttime) +  " millisecs");
}
```

3.3 Real-time Data Pre-processing:

The proposed system will comprise three steps - User and Session Identification, Transaction Identification and Recommendation generation. The system will take the access.log file from the Data cleaning phase as input and perform the Data pre-processing tasks. The User Identification task will read the user id from the log entry to group the log entries by user ids. The Session identification task will then group the log entries for a user based on the user session. The Transaction Identification task will group the log entries into user order transactions by using the semantic information in the logs. The Recommendation generation task will then take the user order transactions as input and create or update the dataset.csv file with all the products purchased by the users. The dataset.csv file is then processed further by the recommendation generation task to generate recommendations for each user using the Apache Mahout Recommender tool. Figure 5 shows real-time data pre-processing.

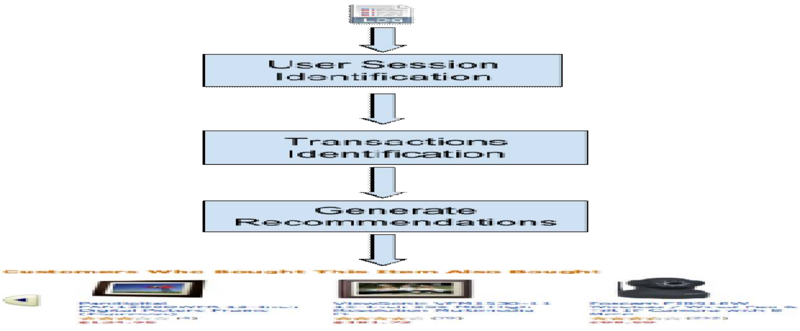

User Session Identification
Transactions Identification
Generate Recommendations
Customers Who Bought This Item Also Bought

Figure 5: Real-Time Data Pre-processing

Algorithm:

Input: Web Log (Web Server access.log) after data cleaning.
Output: Real-time user based recommendations generation.

```
Step 1:  Check if the access log file has changed since its last process time.
Step 2:  If the file has changed, then skip the old log entries and parse the newly added entries.
Step 3:  Read the log entries that have semantic information of the user actions and ignore other entries.
Step 4:  For each log entry read the userId, sessionId and semantic information of the user actions.
Step 5:  Collect the parsed log entries.
Step 6:  Record the position of the last processed log entry with successful user transaction.
Step 7:  Record the current time after processing of the log file.
Step 8:  Sessionization - Based on the session id from the log entries, group the log entries by their session id.
Step 9:  For each of the user sessions, group the log entries by user transactions using the semantic information.
Step 10: For each of the user order transactions identify the products purchased by a user.
Step 11: Prepare the CSV dataset file with userId and purchased productId
Step 12: The recommendation generations task read the CSV dataset file to and generate user based recommendations
         for each of the users using TanimotoCoefficientSimilarity method.
Step 13: Repeat above steps every 5 minutes.
```

Real-time Data pre-processing time is evaluated based on this formula -

$$\boxed{\%\ \textit{Time required for Real-time Data Pre-processing} = (TDC+TUT) * 100 / (TDC+TUT+TRG)}$$

Where,
TDC = Time required for real-time Data Cleaning.
TUT = Time required for User-Session and Transaction Identification.
TRG = Time required for real-time Recommendation Generation.

IV.EXPERIMENTAL RESULTS

A. TIME COMPARISON BETWEEN REAL-TIME AND TRADITIONAL DATA CLEANING:

Log files of sizes varying from around 800 to 4500 for the TomatoCart application were collected. One set of the log files (access.log and access_redundant.log) was generated with real-time data cleaning solution and the other set of the log files (access_full.log) was generated without the real-time data cleaning solution. Figure 6 shows time comparison of data cleaning process. It can be observed that there is reduction of around 12% with the real-time data cleaning as compared to the traditional data cleaning. The data cleaning time may vary from system to system depending on the Apache server version, type of operating system and machine configuration.

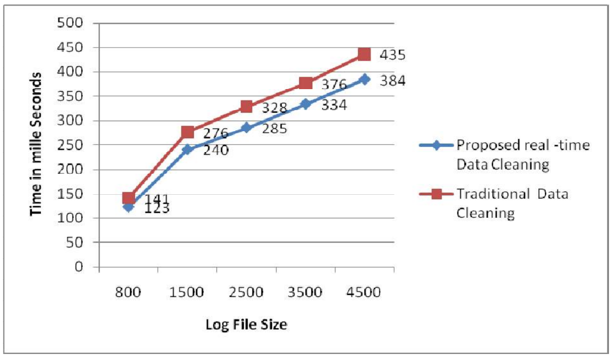


Figure 6: Time Comparison of Data Cleaning Process

B. RESULT OF REAL-TIME DATA PRE-PROCESSING:

The percentage of time for real-time data pre-processing is evaluated for each of the log files by using the real-time data pre-processing formula. The average percentage of time for real-time data pre-processing is 62% of the total web usage mining time. Figure 7 shows result of real-time data pre-processing.

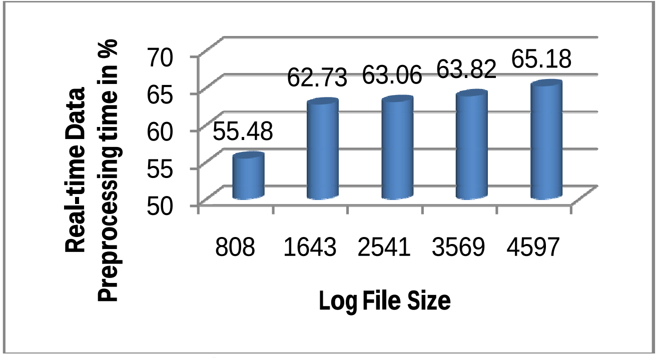
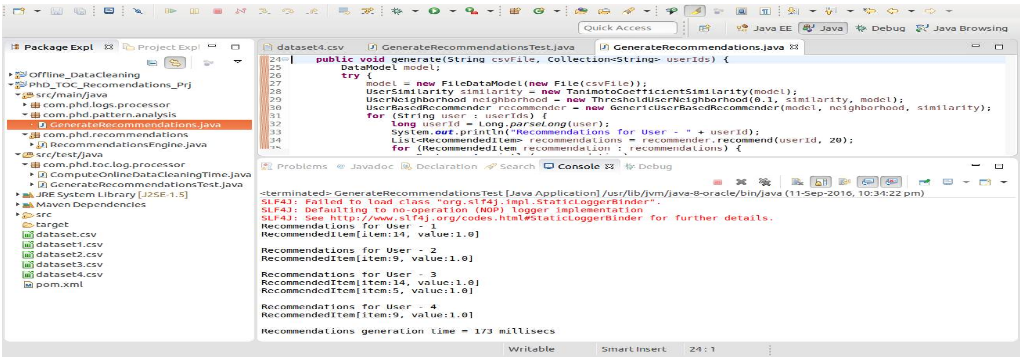


Figure 7: Real-time Data Pre-processing time in %

C. RECOMMENDATIONS GENERATION:

The result of data pre-processing has an effect on the result of recommendation generation phase. In proposed work, real time user based recommendations recommend items by finding similar purchasing behavior of users. This is often harder to scale because of the dynamic nature of users. In proposed research, TanimotoCoefficientSimilarity measure is used to find out the similarity between various users. TanimotoCoefficientSimilarity is based on Tanimoto coefficient, or extended Jaccard coefficient. Tanimoto coefficient is the ratio of the size of the intersection to the size of the union of their preferred items. This is used when user don’t provide preference values. This would be helpful to compute similarity as long as at least preference information is available as a boolean. Figure 8 shows real-time recommendation generation.

DOI: 10.15680/IJIRCCE.2016. 0409048



```
    public void generate(String csvFile, Collection<String> userIds) {
        DataModel model;
        try {
            model = new FileDataModel(new File(csvFile));
            UserSimilarity similarity = new TanimotoCoefficientSimilarity(model);
            UserNeighborhood neighborhood = new ThresholdUserNeighborhood(0.1, similarity, model);
            UserBasedRecommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);
            for (String user : userIds) {
                long userId = Long.parseLong(user);
                System.out.println("Recommendations for User - " + userId);
                List<RecommendedItem> recommendations = recommender.recommend(userId, 20);
                for (RecommendedItem recommendation : recommendations) {
```

```
<terminated> GenerateRecommendationsTest [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (11-Sep-2016, 10:34:22 pm)
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Recommendations for User - 1
RecommendedItem[item:14, value:1.0]

Recommendations for User - 2
RecommendedItem[item:9, value:1.0]

Recommendations for User - 3
RecommendedItem[item:14, value:1.0]
RecommendedItem[item:5, value:1.0]

Recommendations for User - 4
RecommendedItem[item:9, value:1.0]

Recommendations generation time = 173 millisecs
```

Figure 8: Real-time User Based Recommendations Generation

V. CONCLUSION AND FUTURE WORK

In the present work, we propose algorithms for real time data cleaning and data pre-processing. The algorithms are tested with log files of TomatoCart application. In addition, we propose a new structure of web log file to enhance the performance of data pre-processing. The efficiency of proposed data cleaning algorithm is evaluated based on time required for data cleaning process and size of the log file. With the proposed data cleaning algorithm, the size of the Web server log file is reduced by 60% and cleaning time is reduced by 12% in comparison to the traditional data cleaning process. Thus the proposed real-time data cleaning algorithm improves the web log structure, reduces the size of web log file and requires less time for cleaning.

The performance evaluation of real time data pre-processing is measured in terms of time. The data cleaning process is a pre step of data pre-processing technique .The proposed real time data cleaning algorithm, reduces substantial amount of time which affects the result of data pre-processing phase. So the overall time required for data pre-processing technique is reduced i.e. 62%.

Scope for further work:

The proposed system performs well and gives promising results of data pre-processing, still there is considerable scope for further research.

1. Proposed system can be modified to solve the path completion problem in web log pre-processing.
2. Transactions Identification can also be done for other navigation behaviour of users like products added to cart but there was no checkout.
3. User Identification is possible through the combination of IP addresses and other information such as user agents and referrers.
4. User based recommendation can also be improved using product rating as a preference value
5. There is a scope to generate item based recommendations in real-time.

REFERENCES

1. Li Chaofeng, “Research and Development of DataPreprocessing in Web Usage Mining.”, International Conference on Management Science and Engineering, 2006.
2. Doru Tanasa and Brigitte Trousse, “Advanced Data Preprocessing for Intersites Web Usage Mining”, Published by the IEEE Computer Society, MARCH/APRIL 2004.
3. Robert Cooley, Bamshed Mobasher & Jaideep Srivastava, “Data Preparation for Mining World Wide Web Browsing Patterns”.Myra Spiliopoulou, Bamshed Mobasher, Bettina Berendt, Miki Nakagawa,“A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis.” INFORMS Journal on Computing, Vol. 15, Issue 2, April 2003.
4. Yan Li, Bo-qin Feng, Yan Li, “The Construction of Transactions for Web Usage Mining”, Published by IEEE Computer Society, 2009.
5. K. R. Suneetha, Dr. R. Krishnamoorthi, “Identifying User Behavior by Analyzing Web Serve Access Log file”, International Journal of Computer Science and Network Security, Vol .9, No.4, April 2009.
6. Ms. Dipa Dixit, Mr.Jayant Gadge, “Automatic Recommendation for Online Users Using Web Usage Mining”, International Journal of Managing Information Technology (IJMIT) , Vol.2, No.3, August 2010.
7. Sathiyamoorthi, Dr. V. Murali Bhaskaran, “Data Preparation techniques for Web Usage Mining in World Wide Web – An Approach”, Internal Journal of Recent Trends in Engineering, Vol.2, No.4, Nov 2009.

8. Yan LI, Boqin Feng, Qinjiao MAO , "Research on Path Completion Technique in Web Usage Mining", International Symposium on Computer Science and Computational Technology, 2008
9. Bettina Berendt ,"Semantic Web Usage Mining – Overview and Case studies", Humboldt University, Germany.W3C Common Log Format. http://www.w3.org/Daemon/User/Config/Logging.html W3C Extended Log Format. http://www.w3.org/TR/WD-logfile.html
10. Liu Jian, Wang Yan-Qing, "Web Log Data Mining Based on Association Rule", Eighth International Conference on Fuzzy Systems and Knowledg Discovery (FSKD), pp: 1855-1859, 2011.
11. Izwan Nizal Mohd, Shaharanee, Fedja Hadzic, Tharam S. Dillon, "Interestingness measures for association rules based on statistical validity", Knowledge-Based Systems, pp: 386–392, 2011.
12. P. Nithya, Dr. P. Sumathi "Novel Pre-Processing Technique for Web Log Mining by Removing Global Noise and Web Robots", IEEE Conference Publications, 2012.
13. Chintan R. Varnagar, Nirali N. Madhak, Trupti M. Kodinariya, Jayesh N. Rathod "Web Usage Mining: A Review on Process, Methods and Techniques" , IEEE Conference Publications, pp: 40-46, 2013.
14. Johannes K. Chiang, Rui-Han Yang, "Multidimensional Data Mining for Discover Association Rules in Various Granularities", IEEE Conference Publications, pp: 1-6, 2013.
15. Raiyani.S.A, Jain.S and Raiyani.A.G., "Advanced Preprocessing using Distinct User Identification in web log usage data", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 1, Issue 6, pp: 418-422, ,August 2012.
16. Vishwakarma.A and Singh.K.N., "A Survey on Web Log Mining Pattern Discovery", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , pp: 7022-7031, 2014
17. Raiyani.S.A, Jain.S and Raiyani.A.G., "Advanced Pre-processing using Distinct User Identification in web log usage data", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 6, pp: 418-422, August 2012.
18. Patel.P and Parmar.M., "A Review on User Session Identification through Web Server Log", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1), pp: 146-148 , 2014.
19. Raiyani.A.G. and Pandya.S.S., "Discovering User Identification Mining Technique For Preprocessed Web Log Data", Journal Of Information, Knowledge And Research In Computer Engineering, Vol 2(2), pp: 477-482 , 2014.
20. Chitraa.V and Thanamani.A.S., "A Novel Technique for Sessions Identification in Web Usage Mining Pre- Processing", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, pp: 23-27, November 2011.

BIOGRAPHY

Seema Ashok Nimbalkar is a Research Assistant in the Computer Science Department, Bharati Vidyapeeth University Yashwantrao Mohite College of Arts, Science and Commerce, Pune, India. She holds a Master's degree of Computer Science from Abeda Inamdar Senior College, Pune, India. Her research interests are Web Mining, Data Mining and Big Data.

Dr. Suhas Patil is a Professor in the Computer Engineering Department, Bharati Vidyapeeth University College of Engineering, Pune, India. He holds a Ph.D in Computer Science and has more than 24 years of teaching experience. He has published more than 160 papers in National and International conferences and journals. His research interests are Fuzzy logic and Linux Operating System.