"EFFECTIVE DATA PRE-PROCESSING TECHNIQUE IN WEB USAGE MINING"

A THESIS SUBMITTED TO

BHARATI VIDYAPEETH DEEMED UNIVERSITY, PUNE

FOR AWARD OF DEGREE OF

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

UNDER THE FACULTY OF SCIENCE

SUBMITTED BY

SEEMA ASHOK NIMBALKAR

UNDER THE GUIDANCE OF

PROF. DR. SUHAS PATIL

DEPARTMENT OF COMPUTER SCIENCE

BHARATI VIDYAPEETH DEEMED UNIVERSITY

YASHWANTRAO MOHITE COLLEGE OF ARTS, SCIENCE AND COMMERCE, PUNE

MARCH 2017

CERTIFICATION OF THE PRINCIPAL

This is to certify that the work incorporated in the thesis entitled **"EFFECTIVE DATA PRE-PROCESSING TECHNIQUE IN WEB USAGE MINING"** for the degree of **'Doctor of Philosophy'** in the subject of **Computer Science** under the faculty of **Science** has been carried out by **Seema Ashok Nimbalkar** in the Department of **Computer Science** at **Bharati Vidyapeeth Deemed University, Yashwantrao Mohite College, Pune** during the period from August 2010 to July 2016 under the guidance of **Dr. Suhas Patil**

Dr. K. D. Jadhav
Principal
Bharati Vidyapeeth Deemed University's
Yashwantrao Mohite College, Pune

Place: Pune
Date: 31/03/2017

CERTIFICATION OF THE GUIDE

This is to certify that the work incorporated in the thesis entitled **"EFFECTIVE DATA PRE-PROCESSING TECHNIQUE IN WEB USAGE MINING"** Submitted by **Seema Ashok Nimbalkar** for the degree of 'Doctor of Philosophy' in the subject of **Computer Science** under the faculty of **Science** has been carried out in the Department of **Computer Science**, **Bharati Vidyapeeth Deemed University, Yashwantrao Mohite College, Pune** during the period from August 2010 to July 2016, under my direct supervision/ guidance.

Prof. Dr. Suhas Patil

Research Guide

Place: Pune

Date: 27/3/17

DECLARATION BY THE CANDIDATE

I hereby declare that the thesis entitled **"EFFECTIVE DATA PRE-PROCESSING TECHNIQUE IN WEB USAGE MINING"** submitted by me to the Bharati Vidyapeeth University, Pune for the degree of **Doctor of Philosophy (Ph.D.)** in **Computer Science** under the **Faculty of Science** is original piece of work carried out by me under the supervision of **Dr. Suhas Patil.** I further declare that it has not been submitted to this or any other university or Institution for the award of any degree or Diploma.

I also confirm that all the material which I have borrowed from other sources and incorporated in this thesis is duly acknowledged. If any material is not duly acknowledged and found incorporated in this thesis, it is entirely my responsibility. I am fully aware of the implications of any such act which might have been committed by me advertently or inadvertently.

Seema Ashok Nimbalkar

Place: Pune

Date: 27/03/2017

ACKNOWLEDGEMENT

*I wish to express my sincere gratitude to **Dr. Suhas Patil**, Research guide Bharati Vidyapeeth Deemed University, Pune for his continuous support, encouragement and valuable guidance during my research work. I benefited a lot from his constructive suggestion, dedication and efforts to accomplish timely completion of my research. I shall always remain indebted to him.*

My heartfelt thanks to Dr. K.D. Jadhav, Principal, Yashwantrao Mohite College and Prof. S.S. Shukla, Head, Dept. of Computer science, for extending necessary facilities to carry forward my research.

I also thank Prof. Dr. M. G. Bodhankar, Prof. Dr. S. R. Patil for their continuous support and encouragement to complete my work.

My heartfelt thanks to my colleague Prof. Veena Gandhi, Prof. Salauddin Sajjan and all fellow research scholars for their active participation in the technical discussion and providing a lively atmosphere during the course of research.

Lastly and most importantly, words cannot express my deepest gratitude to my beloved parents, my in-laws, my husband, my son and family members, for their love, support, patience and being source of inspiration during the course of work.

Seema Ashok Nimbalkar

DEDICATED

TO

MY

PARENTS

Table of Contents

List of Figures

List of Tables

ABSTRACT

Keywords: *Real-time data cleaning, Real-time data pre-processing, Traditional data cleaning, web log, web usage mining.*

Retrieving knowledge from World Wide Web is a tedious task because of the growth in the availability of information resources on it. Web Usage Mining is the area of Web Mining which deals with the extraction of interesting knowledge from logging information produced by web servers. Source data mainly consist of the (textual) logs that are collected when users access web servers and might be represented in standard format. Web browsing behaviour of users is captured by Web usage data through web site. User activities are stored in web logs. Due to more usage, the files in log are increasing at higher rate in size. Web usage mining consists of three phases, namely pre-processing, pattern discovery, and pattern analysis. First perform the pre-processing for finding access pattern because, raw data which is collected from the web server is incomplete. The Pre-processing plays an important role in efficient mining process as Log data is normally noisy and not distinct. So there is need to reduce the quantity of data being analyzed and to enhance its quality. The present work has proposed algorithms for real time data cleaning and data pre-processing. In addition, a new structure of web log file has been proposed 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. 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 than traditional data 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. 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 behaviour 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. Item based recommendations are generated offline by using Apriori algorithm. Association rules are evaluated on the metric of support, confidence and lift.

CHAPTER -1

INTRODUCTION

Chapter 1
INTRODUCTION

This chapter presents the introduction of the terms used in the thesis title. It gives the basic research background and concepts related to effective data pre-processing technique in Web Usage mining.

1.1 Background

With the continuing progress and upsurge of e-commerce, internet services, the information (click on stream and consumer data) gathered through web based organizations from their day-to-day operations has reached enormous proportions. Inspecting such information will facilitate these organizations to assess the life time value of shoppers like; apply cross-advertising and marketing methods across merchandise and services, measure the effectiveness of promotional campaigns and optimize the practicality of web based applications.

This sort of research includes the automated discovery of important patterns and relationships from an outsized miscellany of fundamentally semi-organized information, usually kept in web and applications server access logs.

The utilization of data mining techniques to find usage pattern from the web, which better serves the requirements of internet based applications is known as web usage mining.

Web usage mining has become the main subject of intensive research, as its extensive potential for personalized services, web-site improvement and usage characterization. When data is collected from server logs several pre-processing task ought to be carried out prior to application of data mining algorithms.

The fruitful use of data mining approaches to web usage data is greatly dependent on the right application of these pre-processing tasks. The most long-drawn and computationally meticulous step in the web usage mining is data pre-processing. The main problem in web usage mining is the massive amount 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 in web usage mining. The effectiveness and efficiency of the data pre-processing step is solely dependent on the quality of data in the log files. In the 21st century World Wide Web proceeds to grow in leaps and bounds in volume, traffic, size and complexity of the web sites. The need of the users has become a major concern and challenging task for a web master who must keep their attention in the web site. Analyzing the user's navigational behaviour can help to improve the web-site design, performance and achieve personalization of the user.

Web Usage Mining comprises of data mining procedures to analyze user access of the web sites. According to KDD (Knowledge Discovery from Databases) process, Web Usage Mining comprises of three stages: Pre-processing, Knowledge Discovery and Analysis. Due to massive size of web log data, it is exceptionally hard to analyse user's usage behaviour patterns. The traditional data pre-processing techniques applied on web log data are more time consuming and not effective. Data cleaning and Data preparation consume 80% of the total analysis time of web usage mining .Very less attention is given to improve the steps of data pre-processing which is often conducted offline on web log data. So there is need to apply real time data pre-processing techniques which require less time for processing of web log data. The result of pattern discovery and analysis is dependent on the quality of results from data pre-processing. Therefore data pre-processing plays crucial role in the entire web usage mining procedure and is the key of its quality.

1.2 Web Mining

Web mining is one of the data mining techniques which help to retrieve knowledge from web information including web documents, services and hyperlinks between the websites. The similar taxonomy of web mining refers to three major lines of research namely structure mining, usage mining and content mining (Losarwar and Joshi, 2012).

1.2.1 Web Content Mining

The web content mining (Eirinaki and Vazirgiannis, 2003) is the method to discover useful data from the web page content. Basically the web content comprises of many kinds of data such as image, textual, video, audio, hyperlinks as well as metadata. Content data is gathering of information from a web page. It can offer interesting and useful patterns about user behaviour and user needs. Web content mining is also referred as text mining which is the second step in web data mining. Content mining is the mining and scanning process which comprises audio, text, structured records, unstructured data, semi structured data and multimedia data. Web content mining is a technique that helps to discover information from web resources and document categorization from the web pages.

1.2.2 Web Structure Mining

Web structure mining is the method that helps to extract knowledge from WWW organization and it relates between referents and references in the web. The web structure mining is the method of using graph theory to examine the connection structure and nodes of a website. The typical web graph structure consists of web pages such as hyperlinks, nodes and edges that are linking web pages (Mahanta, 2008). The structure information or data is discoverable by the act of providing web structure schema through database techniques for web pages. This process can be completed by acquiring the use of spiders through scanning websites, extracting home page, then relating data through reference links to bring forth particular page comprising desired data. The principle purpose for structure mining is to extract undefined relationships between the web pages. The structure data mining is utilized for a business to relate the data of its own website to improve cluster and navigation data into site maps. The structure data mining permits its users to access desired data through content mining.

1.2.3 Web Usage Mining

Web usage mining refers to the automated discovery and analysis of patterns in click stream and related knowledge gathered or produced as an outcome of user interactions with web assets on one or more websites. The objective is to record, model, and analyse the behavioural patterns and profiles of user interacting with web portal. The observed patterns are sometimes described as group of pages, objects or resources which are usually accessed by users with similar interests. Other sources of information like the website content or structure, as well as semantic information from web site ontology, (for example, product catalogue or concept hierarchies), might also be utilized in pre-processing or to reinforce user transaction information. (Bamshad Mobasher).

The web usage mining is also called as web log mining and it is one of the data mining strategies on massive repositories of web log to find knowledge about behavioural patterns of client and statistics of website access which can be utilized for various tasks of website design (Chitra et al, 2010). Web usage mining is the strategy for fetching valuable information from server logs which contains history of client browsing. Web usage mining is the method of predicting what clients are viewing on web. Some users might be viewing at only textual information whereas some others perhaps concerned in multimedia data. The largest web portal in the globe like MSN, Yahoo, etc requires several insights from behaviour of their web visit users. Without the usage report of web usage mining, it will be difficult to frame their monetization efforts. In general, usage mining has direct influence on the businesses. Srivastava et al (2000) has mentioned that Web usage mining is the data mining process applied to find interesting patterns from web usage data so as to better serve and recognize the needs of the customers. Usage information captures the origin or identity of web clients together with their behaviour of website browsing. Web usage mining can be categorized based on the type of usage information considered such as:

i. Data of Web Server

User logs are collected by web server and it usually incorporates reference page, access time and internet protocol address.

ii. Data of Application Server

Application servers for example, Tomcat, JBOSS and WebLogic have essential characteristics to enhance ecommerce applications to be constructed on top of them with

small efforts. The major characteristic is the capability to track different types of business events and log them in the application server logs.

iii. Application level data

In an application, new types of events can be defined and logging can be enabled for them to produce these events histories. Several end applications needs an integration of more than one technique used in the above classifications.

1.2.3.1 Structure of Web Log File

The web log file resides in three locations namely web proxy server, web server logs and client browser (Lokeshkumar et al, 2014). Web log file offers much exact and complete data usage to web server but the log file do not record visited cached pages. The log file located in web server mentions the activity of the client who accesses the website through browser. The proxy server is said to be an intermediate server between the web server and client. Therefore if the web server acquires a client request through proxy server then the entries to log file would be the proxy server data and not of actual user. These web proxy servers manage a separate file of log for collecting the user information. The client browser log files are a type of log file that can be made to locate in browser window of client. Despite the fact that the log file is present in browser window of the client the entries to log file is done only by web server.

Types of web server logs

There are four types of web server logs namely transfer log or access log, agent log, referrer log and error log (Grace, 2011).

i. Access log

The server access log or transfer log records entire requests processed by server. Access or transfer log offers several server data such as internet protocol, address of client, the time, data access and requested web resource. Examining transfer log enhances web administrator to design usage patterns such as number of visitors, most visited page in website, and number of accesses during particular days and hours of the week.

ii. Error log

Error log file is used to record the error on the websites particularly when user clicks on specific link and the browser does not show the specific website or page and returns the

HTTP 404 error message. In error log the first entry is for time and date of message, the second entry is for listing the error severity level being reported. The level directive is used for managing the kinds of errors that are sent to error log by limiting the level of severity. The third entry is client's internet protocol address that produced the error message.

iii. Agent log file

Agent log file is used to register the data about browser of user, OS and browser version. Several users browsing history is useful for designer and they make changes in website accordingly. Also, this can be used to find out the most popular browsers and operating systems among the users.

iv. Referrer log file

Referrer log is used to record the entry of referrer when the user visited specific website by using the link of user's page. Google has implemented an algorithm for page rank to allot the weight to referrer sites. For instance, as someone jumps from www.yahoo.com to any website by clicking the hyperlink, referrer log file of that web server will report a referrer entry that a user originated from www.yahoo.com.

1.2.3.2 Approach of Web usage Mining

i. Data Collection

The first step in web usage mining process is the data collection (Domenech and Lorenzo, 2007). It comprises of collecting relevant web information. The source of data can be gathered at client side, server side, proxy side or it can be acquired from the organization database which comprises consolidated web or business data.

ii. Server Side

Server side gathers requests of client that are stored in server as web logs. Server side can gather huge amount of data in their log files (Chen and Liu, 2006). These logs mostly contain fundamental data e.g.name and internet address of the remote machine, date and time of the request, the request line exactly as it came from the client etc.

iii. Proxy Side

The proxy side is the data gathered from intermediate server between web servers and browsers (Tao et al, 2008). Proxy caching is used to decrease the web page loading time

faced by users as well as load of network traffic at the client and server sides**.**

iv. Client Side

The client side is beneficial than server side since it avoids both the session identification and caching issues. Browsers are designed to record the behaviours of browsing. The remote agents like Java applets are utilized to assemble client browsing (Bayir et al, 2008).

1.2.4 General process of Web Usage mining

The web usage mining generally includes the following steps: data collection, data pre-processing, knowledge discovery and pattern analysis. Figure 1.1 shows the phases of web usage mining.

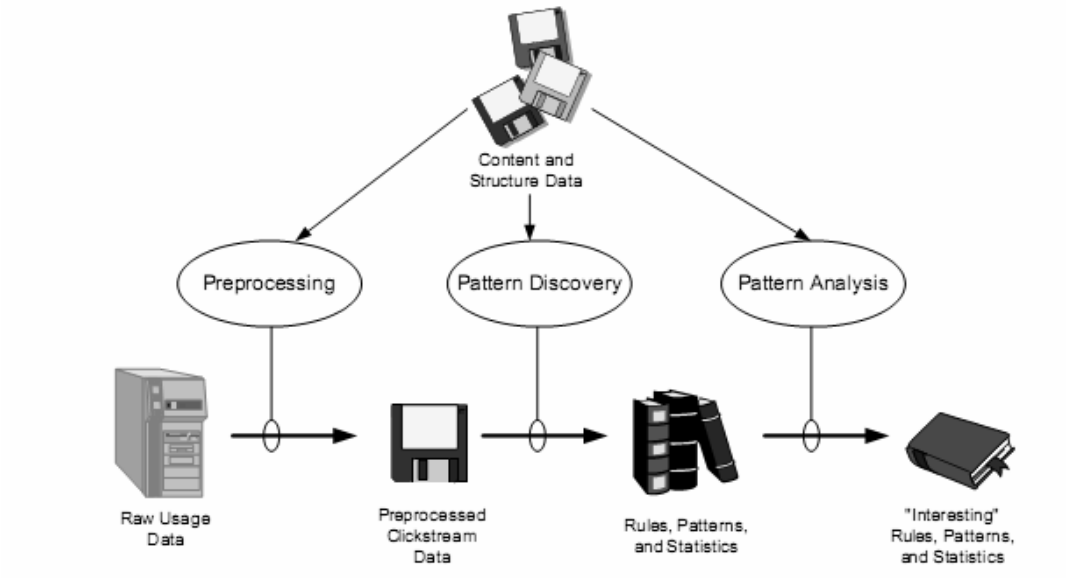

Figure 1.1 Phases of Web Usage Mining

1.2.4.1 Data Pre-processing

A vital step in any knowledge mining utility is that, the formation of an appropriate target data set to which data mining and statistical algorithms can be applied. This can be notably vital in web usage mining owing to the characteristics of click stream information and its relationship to other related information gathered from numerous sources and various channels. The data preparation is essentially the most time ingesting and computationally intensive step in the web use mining. This process is crucial to the successful extraction of usage patterns from the data. A lot of the research and practice in usage data preparation has been emphasized on pre-processing and integrating these

information sources for various analyses.

In web usage mining the goal of pre-processing stage 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 information can be utilized as input for various 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 utilization of data mining methods to web usage information is greatly dependent on proper use of pre-processing tasks. This stage includes pre-processing of gathered data from various sources and transforming them into a type relevant for applying the operations of data mining. The purpose of data pre-processing is to provide structural, integrated and reliable information source to pattern discovery (Dangi and Sangwan, 2013). It comprises of four processes described below.

i. Data cleaning

Data cleaning is often site-specific, and the web log file is the source of data to the process of data cleaning. The data cleaning purpose is to remove irrelevant record or items from the log file. The data cleaning task is to delete unrelated data for mining namely images in the format of JPEG, PNG and GIF, error response, JavaScript, cascading style sheets and robot requests. Data cleaning also involves the removals of references due to crawler information navigation.

ii. User identification

User identification is performed after data cleaning. The users who have visited the website are 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 knowledge about the identity of user. However it's essential to differentiate amongst various users, considering the fact that a user may visit a website more than once, the server logs register various sessions for every user. The user activity record phase is utilized to identify the sequence of logged actions of similar users. Without user login mechanisms, the most popular approach to distinguish 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 it is not possible to recognize distinct users accurately. A further procedure can be the use of IP addresses for client identification. Client with distinct IP addresses can also be treated as unique but it's not the ideal solution as ISPs (Internet service provider) assign rotating addresses to clients.

iii. 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 client side session can be defined as the set of pages visited by a specific user within specific duration of one particular visit to a web portal. A client may have multiple or single sessions in the course of a time interval. Once the user has been recognized, then every user click stream is divided into the logical clusters. The partitioning of this data into 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 strategies utilized for client identification to a specified level, can also be utilized for session identification**.**

iv. 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 be appended (Chandrama et al, 2014). 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.

1.2.4.2 Pattern Discovery and Analysis

In pattern discovery, the application of different data mining techniques like association, statistical analysis, pattern matching, clustering and so on (Srivastave et al, 2000) process the data to discover the useful patterns. 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 and 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 depends on need of analyst to decide which techniques of mining to make use of. The final step in web usage mining process is to filter out uninteresting rules of patterns from the set found in pattern discovery stage.

This can be used for changes in website, web personalization and/or system improvement. The similar methods used for pattern analysis are on-line analytical processing techniques, visualization techniques, usability analysis, data and knowledge querying.

1.3 Organisation of the Thesis

Chapter 1 is the introduction chapter that gives the primary research background and concepts related to the research.

Chapter 2 presents the survey of literature and analyses several works done on data pre-processing which aim to enhance the performance of data pre-processing technique. Data pre-processing, pattern discovery and pattern analysis techniques are reviewed.

Chapter 3 focuses on the objectives of our research and the problem statement is introduced. A brief description about the proposed data pre-processing technique is mentioned.

The **Chapter 4** describes about the research methodology used for real-time data cleaning process. Main objective of research is to significantly reduce the size of the Web server access log file, reduce the time required for data cleaning process and increase the quality of the data in the web server logs. The efficiency of proposed real time data cleaning algorithm is validated by conducting trials on different sizes of web log files. The performance of traditional data cleaning process and the proposed online data cleaning process is compared.

The **Chapter 5** presents the extensive experiments conducted in order to assess the performance and effectiveness of real-time data pre-processing. The proposed algorithm for real time data pre-processing in web usage mining is divided into three main steps:

User and Session Identification, Transaction Identification and Recommendation Generation.

The **Chapter 6** discusses the results of experiment using the log files of TomatoCart ecommerce website. The results shows that our methodology reduced the web access log file down by 60% of the initial size. As per our observation around 12% less time is required than traditional data cleaning process. The average percentage of time required for data pre-processing is 62%.

Chapter 7 Summarizes the thesis results, which shows the result of data cleaning, data pre-processing and recommendation generation. It also covers the future research directions.

CHAPTER -2

LITERATURE REVIEW

Chapter 2

LITERATURE REVIEW

This chapter presents reviews of the related research work on data pre-processing, pattern discovery and analysis. The different stages of web usage mining are discussed in this chapter.

2.1 Introduction

The literature review is the section of a study where the researchers would explain elaborately about the existing studies/ researches on the relevant topic and also explain the similarities and differences between the studies and the current study. The research topic is ‘Effective Data Pre-processing Technique in Web Usage Mining’ which would analyze the studies about: web log characteristics, data pre-processing: data cleaning, user identification, session identification, transaction identification, issues in data pre-processing, pattern discovery and analysis. Finally a summarization of this section would be presented for crisp understanding of future research and similarly the difficulties in data pre-processing would also be discussed for better understanding.

2.2 Web Log Characteristics

The authors Grace et al., (2011) had studied about the characteristics or features of the web log and the uses in the web mining. According to them, the log files in the web mining consists of varied information, such as: time stamp, access request, user name, IP address, result status, referred URL, user agent and the bytes transferred. Basically the web logs are stored and maintained in the main web servers. When the data has to be accessed for varied processes for analysis (for instance: data pre-processing) it can be done through accessing the web servers. The features of content in the web logs are: user name, visiting path, path traversed, time stamp, page last visited, success rate, user agent, URL and request type. Similarly the log file is found at three sources, a) web servers, b) web proxy servers and c) client browsers. There are four types of log files namely: 1) transfer/access log, 2) agent log, 3) error log and 4) referrer log. The latter logs (error and referrer) are normally found in the "extended" log file format whereas the former logs (transfer and agent) are basic and standard. Thus the characteristics of the web log had been structured by the authors for better understanding along with web mining and data pre-processing techniques.

Dhawan and Goel (2013) studied the web usage mining with usage patterns from web logs. Web mining is one of the applications of data mining techniques to web data. It can be file of log data, web document content and hyperlink structure. Web usage mining encompasses discovering knowledge from web log files. In fact, the web log files have represented the user activities accessing a web site and hence it has provided the vital information about the user access patterns. Access log, error log, agent log file and referrer log are some of the types of server log files associated with the web server logs.

Talakokkula (2015) has surveyed about the web usage mining applications and tools used for extracting useful information from various web logs. In addition, the web log mining has been considered as the process of extracting interesting patterns from the web access logs. Server logs, client logs and network logs are the various web logs associated with the web usage mining. In fact, the various web logs used in the web usage mining are maintained by server, web browsers and other networks.

2.3 Review of Literature on Data Pre-Processing

Losarwar and Joshi (2012) mentioned about the data pre-processing which is called as the data mining ground work and the time spent for this project of data mining is 80% in the real world of data mining. The information gathered from the World Wide Web is called the web mining which involves patterns discovery. The determination of any organisations value of the product can be carried out by the data pre-processing and the specific customers rating, promotional products campaigns, products strategies across the marketing network, etc can be analysed. The steps of the data pre-processing are data cleaning, session identification and user identification.

C. E. Dinucă (2012) mentioned association could be a data mining technique to extract useful knowledge from web usage data. Java programming language is used for identification of association of web pages from sessions. The author co-jointly expressed that fruitful analysis is affirmed on accurate information and high-quality of data. Preparation of information required time somewhere around 60% and 90% from data analysis and contributes to a success rate of 75-90% to the whole process of extracting knowledge.

Shaily Langhnoja, Mehul Barot, Darshak Mehta (2012) proposed algorithm for data cleaning, user and session identification. Data cleaning algorithm used for cleansing web log file. After data cleaning of web log file records reduced 411 from 1217 records. User and Session identification algorithm marked every record in the database with respective client. Session identified groups that later can be utilized for further course of action of web usage mining process. The resulted group of records can be embedded into database and later results of which can be highly useful like aggregate number of clients, aggregate number of sessions, difference between aggregate number of records before pre-processing and post-pre-processing, etc. used for analysis.

S. Prince Mary and E. Baburaj (2013) delivered the steps of pre-processing involving data cleaning, user identification, session identification and path completion. Once pre-processing is carried out, patterns are discovered using techniques like statistical analysis, association and clustering. The exposed patterns are analysed for different applications

like web personalization, site improvement, site modification, business intelligence, and so on. Web data mining ground work is done at pre-processing phase. Any real time data mining project usually spends 80% of the time on data pre-processing step.

Mitali Srivastava, Rakhi Garg, P.K.Mishra (2014) discussed data pre-processing techniques in detail with their advantage and disadvantage. They expressed pre-processing of log file is an important step and it takes almost 80% time of whole web usage mining process.

V. Vidya and Priya S. Kalaivani (2015) proposed clustering of web log data to identify user access pattern. In the first phase pre-processing is done. After data cleaning, 25% records are left. In second phase K-means and farthest first clustering techniques is used. By clustering the web pages they can easily identify the user interest and the access pattern.

In 2015-2016 Vijayashri Losarwar and Dr. Madhuri Joshi proposed complete pre-processing technique to pre-process the web log for extraction of user patterns .The authors worked on data cleaning algorithm, removes near about 50-60% irrelevant records from web log and filter algorithm discards the disinterested attributes from log file .User and Sessions are identified. This paper also mentioned to carry out real world data mining project 80 % time is spent on data pre-processing.

Abraham and Puthiyidam (2016) expressed the data pre-processing used in many remotes sites and complex scenarios that rely on integration of the time in collection of large amount of databases. The various steps in data pre-processing are normalisation, transformation, selection, cleaning and feature extraction. The processing time is in much higher amount in the preparation of data. The operation of the data mining is potentially helpful for identifying patterns from the existing data.

Makwana and Rathod, (2014) explained the data presents in the websites log files used to analyze users behaviour. The data pre-processing is the technique for discovering various usage patterns from log files with different tasks like extraction, user identification, path

completion and transaction identification. The pattern discovery techniques used for deriving the data, uses (utilizes) association rule, clustering, and classification.

Tomar and Agarwal (2014) stated about the data mining steps like pre-processing and post processing techniques in a survey. The voluminous data of the information gained from the websites are explored and processed in knowledge discovery phase. The processing may be difficult if it contains irrelevant data in log files, unrelated features, and several other inconsistencies. The prediction model of data pre-processing feature algorithm are clustering, text categorisation and rule induction. The characteristics of the information of datasets are data set size, multi-dimensional supporting, revealing pattern ability, clusters and background noise amount. The major steps in the visualisation of the pre-processing undergo major changes with the time requirements, detailed description of different data, dimensional reduction and cleaning approaches.

Pre-processing in web usage mining has been studied by Mitharam in 2012. Pre-processing section depends upon web log files. Web usage mining process seems to be incomplete without using the step of pre-processing. It also depends on the server log file. Data fusion, cleaning and user identification are significantly associated with the data pre-processing. In this way, the data pre-processing is a milestone of web usage mining has been studied by Kherwa and Nigam (2015). The raw log data was pre-processed to attain the reliable session for efficient mining. The main advantage of data pre-processing is the capability to improve quality of data and it helps to enhance the mining accuracy successfully. Extraction of fields has been considered as the first phase of data pre-processing.

2.3.1 Literature Review on Data Cleaning

Koh (2006) clearly states the improvement of the data quality can be done by the data cleaning with various high evolutionary warehouses database. The clean data is the data which are cleaned with adequate amount of steps in cleaning and quality checking. The cleaning of data undergoes steps like process of data mining, data artefacts complexity with dirty data which is a complicated analysis. The applications of the clean data being demanded by data warehouse, customer matching, and information system integration.

The correlation of the clean data approaches are observed with the relationship with semantic entities, attributed value beyond the individual and additive information extraction. The various data cleaning approaches are as follows: duplicate detection methods, efficiency driven methodology, accuracy driven method, and outlier detection method.

Ganti and Sarma (2013), explained about data cleaning with various activities in the warehouse of business and decisions in the business supporting reports. The high quality of data can be maintained in any organisation with the process of data cleaning. Even a small error in the process of data cleaning can ruin the reports of the data, so there are high challenges with critical analysis to have a clean data platform.

Krishnan, et al (2016) determined about the data cleaning in a reliable interaction with frequency iteration process. The novel challenges of the technical, analytical and organisational tools in the designing and implementing the iterative process of data cleaning belongs to the community of data mining. The three main schemes of the data cleaning are data cleaning iterative nature, data cleaning correctness with lack of evaluation and querying of data.

Hongzhou Sha, Qingyun Liu (2013) proposed the algorithm named EPLogCleaner that can filter out lot of unrelated items based on the common prefix of their URLs. The authors reviewed EPLogCleaner with a real network traffic trace captured from one enterprise proxy. The experimental outcome showed that EPLogCleaner enhances the data quality of logs by filtering more than 30% URL requests compared with traditional data cleaning techniques**.**

Hellerstein (2008) used sort based algorithm, one pass algorithm, median to other robust estimator’s algorithms for data cleaning. He has also used visualization techniques and data detection techniques as the data cleaning types. However, the statistical matching techniques of data cleaning types are used by Erhard Rahm and Hong Hai Do.
K. Sudheer Reddy, G. Partha Saradhi Varma, and M. Kantha Reddy (2014) proposed data cleaning technique for web log pre-processing, by removing unwanted click streams from

the log file. They reduced the original file size by 50-55%. Heuristics approach is used for User Identification and Session identification. Session identification is done by using time frame between the page requests usually 25.5 or 20 minutes.

According to Sagar and Nimavat (2015) unnecessary and irrelevant fields from the raw log files have been removed like gif and jpg because these file types are not requested by the users. In addition to these, the data cleaning techniques have been used to remove the irrelevant data from the log and it also has the tendency to follow all the hyperlinks from web pages appropriately.

Web usage mining framework for data cleaning and IP address identification has been studied by Verma and Kesswani (n.d). Data clean-up has been referred as the first stage of pre-processing. Data cleaning process has the capability to remove the data inconsistencies. According to the field extraction step, the records get extracted in MS Excel and it is then stored in SQL server to perform the user identification and data cleaning. After data cleaning, records get reduced from 500 records to 441. Pre-processing is the most vital stage as it requires 80% effort of whole web usage mining process.

In 2016, Muskan and Dr. Kanwal Garg emphasized on data cleaning algorithm. The algorithm removes accessorial entries like multimedia files, status code other than 200 and entries made by spider or bots. Initially, log table had 1545 entries after applying proposed algorithm, entries remained 462. The size of the log file decreased up to 71%.

2.3.2 Literature Review on User Identification

According to Singh and Badhe (2014), the user identification in web mining is the user centric approach for the best process of identifying the users with their data in any website logs. There are vast amount of techniques used in the web mining process. The three categories of the web mining are the content mining, usage mining and structure mining of the web. The methods used for user identification process are: IP address, user generated informative data, cookies collected from the users. The user can be identified by using the data with specific categories like specific person, name, working status, age

group, hobbies, nature of person and temporal strategies.

Singh and Badhe (2014) has surveyed on web usage mining for user identification. User identification has been considered as one of the steps of pre-processing techniques. Individual user identification referred as one of the important step in web usage mining. Various methods have been implemented to follow the user identification. Assigning different user id to different IP address is one of the simplest methods in the pre-processing techniques. Some of the methods of user identification are: IP address, user registered data and cookies.

User identification, classification and recommendation in web usage mining is an approach for personalized web mining has been studied by Priyanga and Naveen (2015). Once the method of data cleaning in the log file has completed, the user identification through IP address and cookies has been carried out. Authors have also revealed the pseudo code to identify the user with the usage of IP address. IP address and user agent are considered as the heuristics of user identification process.

Jiadi and Hai (2016) have used the heuristic rule based algorithm for user identification. Rejena and Malika (2016) used unique user identification algorithm and dynamic hashing technique for user identification. User identification algorithm, distinct user identification algorithm, algorithm based on pattern using clustering and classification to focused on separating potential users from others, decision tree classification using C4.5 algorithm to identify interested users for user identification are mentioned in the literature survey by Singh and Badhe (2014) and they also expressed the problem of user identification in web usage mining.

2.3.3 Literature Review on Session Identification

Dinuca and Ciobanu (2011) explored about the data pre-processing step called the session identification which is used for identifying the session of each users. The sessions are usage of web sites at different times by individual users. The session should be correctly identified to maintain the user's history of web site access. The three categories for perfect analysis of the session are the determination of average time on single page,

visiting of single web sites, duration of each visits and data used for session identification in the web log.

Halfaker, et al (2015) stated the common strategies used to develop web analytics and behaviour analysis of the user. In this work they demonstrate a strong regularity in the temporal rhythms across various web domains in the online activities like page viewing, video gaming and searching web pages. The session identification is navigational oriented heuristics and as its time dependent it is called the time oriented heuristics.

Patel and Parmar (2014) researched about the user identification through the websites, log repositories and servers sites. The pattern of the users can be analysed by the knowledge sources obtained to acquire the behaviour of the users. The modern process of user identification is based only on the single piece of information called the time of the users logging and suing the web servers. The method is used for acquiring the cluster of sessions used by the users with timing analysis. The timing analysis is the time per each session, statistical results of page and initial time degree of each page can be observed.

Kamat, Bakal and Nashipudi (2013) studied about the improved data preparation technique in web usage mining. Sessions is considered as time between the sign in and sign out. It additionally finds the click stream sequence to trace the user effectively. Session identification algorithm has taken input as user list. In this way, the session identification has done using AHL (Access History List) by considering link analysis. Authors have concluded that the system is more complex but the sequences of the user session has generated with less time and greater precision effectively.

Nigrel et al (2015) studied about the web log pre-processing for web usage mining. The proposed system of the authors has identified the users using two-way hash structure. Backward referencing has taken much lesser time and hence it has identified the session with higher precision. Creation of new session is based on: 1. If there is a new user, there is a new session. 2: In one user session, if the refer page is null, there is a new session.
3: If the time between page requests exceeds a limit of 30 minutes (default timeout for session).

Padala et al (2013) has used the Data cleaning algorithm, hierarchical session identification algorithm and user identification algorithm for session identification. Data pre-processing technique and pattern discovery technique are also used in their study.

2.3.4 Literature Review on Transaction Identification

Cleve, et al (n.d) expressed about the transaction of data in the data mining process which are used to analyse the customer's data transactions. This transaction identification is used to clearly get the needed results of the transactions of the users or customers logging the web servers. The phases of the data mining process in characteristic transaction identifications are business understanding, data understanding, data preparation, modelling, evaluation and deployment. Data preparation of the transactions of the users can be used with pre-processing which includes data selection, normalisation and transformation. Various algorithms are used for clustering the data acquired from the transaction of the users like artificial neural networks approach and clustering algorithms.

Joslyn, et al (n.d) reports about the potentiality of the graphs used in the web mining process with exploration strategies of the transaction data from the users. The advancement in science and technology in the web mining now make it ease with use of graphs, models, directed graphs, hyper graphs, to represent the identification of transactions made by the users. The bit coin block chain is the representation of the transaction identification. The potentiality of the data identification objectives with the help of the bit coin block chain is attempted with graph analysis and mathematical structuring. The mathematical structuring representation of the block chain has three main regulations of graph called transaction graph, transaction hyper graph and transaction bipartite multi graph. The report of the algorithms used in the mathematical representation can be further analysed with the visualisation process of user's identification and statistical methods. The transaction identification can be used with computerised methodology with structural patterns and contents from the users.

Tamrakar and Ghosh (2014) studied about the identification of frequent navigation pattern using web usage mining. The identification of transaction has varied from case to

case depending on the technique of web usage mining. Usage pre-processing, content pre-processing are some of the pre-processing phase used for knowledge discovery through some of the data mining techniques such as association rules, sequential patterns and clustering. Data cleaning, user session, path completion and transaction identification are some of the steps of pre-processing phases handles the system effectively. A-priority algorithm has designed for finding association rules.

Patel et al (2015) has surveyed web mining approaches. Aprior-Tid algorithm has associated number of entries which may be smaller when compared to number of transactions in the database. Further the algorithm also scans transaction database and it also has the tendency to reduce the cost of system and increases the data mining efficiency. The transaction data has transformed into a representation in which it incorporates the complex semantic entities successively. The interested transactions have identified from the clean log using the process of transaction identification, user registration data and documentations.

Vellingiri and Pandian (2011) has focused on providing techniques for better data cleaning and transaction identification from the web log. They mentioned aim of transactions identification is to create relevant clusters of references for each user. Transaction identification is done by means of merge or divides techniques. To discover the user's travel pattern and user's interests, two types of transactions are outlined i.e. travel path transactions and content only transactions. The content only transactions are content pages which are utilized as a part of mining to find user's interest and cluster users visiting the same web site. There are three approaches available to identify transactions; they are identification via reference length, identification through Maximal forward reference and identification by way of Time Window.

2.4 Literature Review on Pattern Discovery

Aggarwal and Bhatia (2015) researched about the methodology called the pattern discovery used for tailoring the various websites and their contents for the users gain information. The information is found in large collections in the web which can be navigated to customer's services by sending appropriate data that the users might want to

discover. The pattern mining is responsible to the additional information acquired from the websites according to the requirements of data for the users. The three categories of the web mining are web usage mining web structure mining and web content mining. All the categories of the mining are personalised and recommended by the performances of the users.

B.Santhosh Kumar, K.V.Rukmani (2010) focused on web usage mining and specifically on finding the web usage patterns of websites from the server logs. The comparison of memory utilization and time utilization is compared using Apriori algorithm and Frequent Pattern Growth algorithm. The fundamental disadvantage of Apriori algorithm is that the candidate set generation is expensive, particularly if a huge number of patterns and/or long patterns exist. The fundamental disadvantage of FP-growth algorithm is the absences of good candidate generation method.

Saxena and Shukla (2010) explained clearly about the web log data and their pattern discovery in the significant interval of time in the process of sequence mining. This pattern discovery can be analysed by the significant use of algorithms with the traditional pattern detection approaches. Pattern discovery can be discovered by the Significant Interval Discovery algorithm (SID) which has the following intervals: unit significant interval, disjoint significant interval and overlapping significant interval.

Zhong, et al (2012) proposed effective techniques in the text mining with the effective discovery of patterns in the text documents. The data mining approaches of data has attracted the attention of the digital world which is rapid in its growth and can be benefitted by the knowledge and data gained in the business field. The effective pattern discovery has two main fundamental issues called the misinterpretation of pattern and low frequency. Patterns can be effectively studied with the text mining by the variety of algorithms like FP tree, SPADE (Sequential Pattern Discovery using Equivalence classes), Aprori algorithms.

Kayuturk, et al (2005) expresses about the discrete attributes of the pattern discovery with compression and clustering. The data analysis is threatened with challenges when the data

sets have high frequency in various applications. The two key technologies of analysing the pattern discovery datasets are compression and sampling method. While comparing with the data analysis and pattern discovery the reduction in data are the main scope of latent structures discovery and matrix decomposition. Instead of considering the expensive traditional algorithms the patterns can be presented by the technique called the PROXIMUS which is accuracy oriented. The main properties of the PROXIMUS are interpretation in patterns, interesting discovery, highly capable in compact performance, large datasets scalability, and efficient at runtime for the data pattern discovery.

According to Charjan and Pund, (2013) the algorithms for the pattern discover in text mining are used in various pattern mining techniques which are used in search of interesting patterns. The patterns can be determined by the use of the effective, interesting and relevant information being deployed from the process of pattern innovation and improvising pattern evolution. The process of knowledge development model, like data selection, data processing, data transacting and pattern discovery are always being extracted from the documents of text. So the evolution of the discovered patterns is correlated with the knowledge patterns which are interesting and associated with mining of rule. The knowledge discovery has the ability to find the worthiness of information in pattern discovery of the uses text documents.

Sundari, et al (2014) stated about the techniques for pattern discovery by using web mining process with great advancement in the revolutionary technology. The information retrieval can be made easy with the use of pattern discovery. The pattern discovery can be included with the knowledge gain of web log files and recognized the web based useful information for the user interests. Pre-processing of data with the knowledge extraction is ensure to process the data loading, accuracy checking ,gaining data transformation and structuring data ,which is based on the data mining algorithms and analysis.

Bhaiyalal Birla and Sachin Patel (2014)**,** incorporates pattern mining algorithms for assessment and implementation of frequent pattern analysis from the web data using Apriori Algorithm. To enhance the efficiency they proposed a new modified Apriori algorithm and implemented it. Lastly they made a comparison of the outcomes of the

Apriori and new modified Apriori algorithm. As per their observation the modified Apriori algorithm takes lesser time than Apriori algorithm for search patterns and building data models.

B. Uma Maheswari and Dr. P.Sumathi (2015), compared two standard web usage mining algorithms namely Apriori algorithm and Frequent Pattern algorithm. Particularly, they focused on discovering the web usage patterns of websites from the server log files.

Mr. Ashish Vitthalrao Galphade and Mr. Dhiraj Bhise (2016), analyzed the association rule based algorithms specifically Apriori algorithm, which address the needs of different web service providers and different viewers, clients, business analysts, and so forth. It enhances the techniques of Web Usage Mining by first finding the log files of individual users at one place. This collective knowledge can be utilized to outline business techniques to boom sales.

2.5 Literature Review on Pattern Analysis

Campagni, et al (n.d) studied about the methodology of the data mining in the data base of students with the pattern analysis. The algorithms used for the pattern analysis is SPAM (Sequential Pattern Mining) to determining the students examinations database with corresponding courses for identification of frequent patterns. The clustering techniques used in the pattern analysis for refining the large collection of student's data. The traditional data analysis is not efficient for identifying the hidden patterns and the rapid increase in the technological development in storage of data. Mining helps in understanding the pattern analysis. The techniques used in the pattern analysis of data mining are association, clustering and classification rules. In the sequential pattern analysis the sequence of elements are used in collections. The length of the sequence can be called by k events and sequences are named as k sequence.

Steinbach, et al (2007) determined the measurement of objectives in the pattern analysis with challenges in data mining and data analysis with the huge data sets. The wide range of datasets being deployed in the business, industries and science field are associated with the tools of data analysis in the data mining area with collection of unique sets of data attributes and variables relationships in the binary data. The associated analyses of the

patterns are the key concept in measurements of any desired datasets which gives interesting approaches. The properties of the quality pattern analysis are inversion property, scaling property and scaling invariance property.

Cheng, et al (n.d) mentioned the classification of the frequent and discriminative pattern analysis in sporadic studies dealing with the graphs, text documents and relational data with their classifications. Analysis of the pattern mining with the frequent data analysis are divided into various types namely, sequences, graphs and item sets. The applications of the frequent pattern mining are clustering, indexing and associated rule mining. The pattern mining frequency requires very less threshold support .The algorithms are used in an efficient way to derive the classification of frequent sets and discriminative pattern analysis. The scalability of the pattern analysis based classification give better results with algorithms enumerating the combinational methodology. These classification based on the frequent pattern analysis are classified into three steps as follows, feature generation, feature selection and model learning.

An effective approach for pattern discovery in web usage mining has been studied by Deshpande, Bagwan and Deshmukh (2014). Pattern analysis motivation has filtered out uninteresting patterns in the previous phase. The techniques of visualization helped the application domain and it also helps to analyse the discovered patterns appropriately. Authors pointed that the pattern analysis has considered as the last stage of web usage mining process (Rathi and Raipurkar, 2016). The tools of pattern analysis have helped to transform the information into knowledge.

2.6 Literature Review on Issues in Data Pre-processing

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

The issues/ 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 the author stated in this study is about ‘personal information’ login details 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 and making the process difficult. Hence the authors had proposed DUI (Distinct User Identification) algorithm to retrieve user identification. Authors had mentioned that the data pre-processing is a time consuming and hence it poses a huge issues (Raiyani and Jain, 2012).

According to the authors, data cleaning under their proposed algorithm would be more effective 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 user's information. KNN (K-nearest neighbour) and PCA (Principal Component Analysis) algorithm was proposed by the authors to differentiate data and to map filter information for faster 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 (Vishwakarma and Singh, 2014).

In this paper the authors have studied about the issues in data cleaning process. 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; hence identifying a user with the IP address will be difficult task. 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 session users, are the factors that are to be considered for each sub-process (Raiyani, Jain and Raiyani, 2012).

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. On contrary the authors studied about the user identification and session identification. 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 completion and transaction identification. Hence 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 to identify the user through web log mining (Patel and Parmar, 2014).

Hanane Ezzikouri, Mohammed Erritali and Mohamed Oukessou (2015), mentioned web data pre-processing required most time, due to the lack of structuring and the large amount of noise present in the raw data. The first stage of web usage mining process, which is obviously Pre-processing, occupies about 60% to 80% of the time involved in the whole process. Pre-treatment of Web log files is to clean and organize the data contained in these files to prepare them for future analysis. Data Cleaning shown that significant reduction (up to 70%) of the initial number of requests and offers richer structured logs for the next step of data mining.

Khushbu Patel, Anurag Punde, Kavita Namdev, Rudra Gupta and Mohit Vyas (2015) worked on survey of web mining. The aim of the paper is to provide past and current techniques in Web Mining. They discussed about research work done by different researchers and important research issues related to it .Moreover, they expressed Web cleaning is the important process and it becomes difficult when it comes to heterogeneous data. The accuracy of data needs to be concentrated. According to researchers 70% of the time is spent on data pre-processing.

As per the author’s 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 session varies according to the authors. The user identification could be done through session identification however the session identification has to be 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 used Distinct User

Identification algorithm to improve the overall designing and performance of upcoming access of pre-processing results. (Raiyani and Pandya, 2012).

The authors mentioned data cleaning, user identification and session identification techniques. In the data cleaning phase, unnecessary records including graphics files, robots are removed. The records resulted after cleaning phase is 1476 from 9464 records. After the data cleaning process is performed, users are identified by using IP address and User-Agent fields. The next process is the identification of sessions which is derived by forming the user behaviour in matrix format. (Chitraa and Thanamani, 2011).

2.7 Summary

This section has evaluated the determinants and factors that revolve around data pre-processing in web usage mining. Through literature review it can be observed that the web usage mining depends on data pre-processing technique. The different attributes and factors involved in web logs are (Grace et: 2011) (Rao and Kumari: 2011); data cleaning processes in data pre-processing, algorithms and techniques utilized in user, session and transaction identification of data pre-processing, pattern discovery and its attributes were analysed in depth, to know more about data pre-processing of web usages mining.

Basically there are numerous studies in the pool of literature that has explored about the data preprocessing and web usage mining. Howsoever the research gap that exists in between the data preprocessing in web usage mining has been studied by very few authors/ researchers (such as: Rao and Kumari: 2011, Herring: 2009, etc). Hence the current study has focused upon bridging the gap between the existing studies on data pre-processing techniques in web usage mining.

CHAPTER -3

RESEARCH OBJECTIVES AND APPROACH

Chapter 3

RESEARCH OBJECTIVES AND APPROACH

This chapter presents the Problem Statement; Objectives of the Research and Proposed System are introduced in this chapter.

3.1 Problem Statement

With the continual development and rise of e-commerce web services and Web-based information systems, the magnitude of click stream and user data gathered by Web-based organizations have reached in vast proportions. Analyzing such data is main objective of these organizations to discover knowledge about user behaviour pattern and web site usage statistics that can be used for various website design tasks. This data is generated automatically by web servers and collected in server access logs. So there is need to reduce the quantity of data being analyzed and to enhance its quality.

The main aim of this research is to develop an effective real time data pre-processing technique that will help in better web usage mining for generating user based and item based recommendations efficiently.

3.2 Objectives of research

The main objectives of the thesis can be stated as:

i. To identify the problems associated with existing techniques of data pre-processing phase.
ii. To design and developed a real time data pre-processing technique by developing efficient algorithms for data cleaning and data pre-processing.
iii. To modify the structure of web log file.
iv. To improve the performance of data pre-processing phase.

v. To generate real time user based recommendation and generate offline item based recommendation using association rules.

3.3 Difficulties in Data Pre-processing

Data Pre-processing in web usage mining consists of Data cleaning, User Identification, Session Identification, Path completion and transaction identification. Data cleaning involves removing requests for irrelevant resources. As in case of web-sites like Yahoo, Amazon the size of log files can reach to hundreds of GB per hour. Web log file of these web sites may contain irrelevant request like images, errors pages, Java script, CSS requests etc. Therefore, data cleaning of such files is a complex process. Although existing data cleaning techniques can reduce the size of the log file up to 50%. , but these techniques are more time consuming and not effective for massive size of web log files. User and Session Identification are complicated processes due to existence of proxy server, browser cache, security and privacy issues.

Researchers have suggested various heuristics which still don't give accurate results. According to various researchers, over 80% of time required to carry out any web usage mining project is spent on data pre-processing. Result of data pre-processing affects the quality of pattern discovery and analysis. So there is a need to improve the structure of web log file, reduce the size and time required for data pre-processing techniques.

3.4 Proposed System

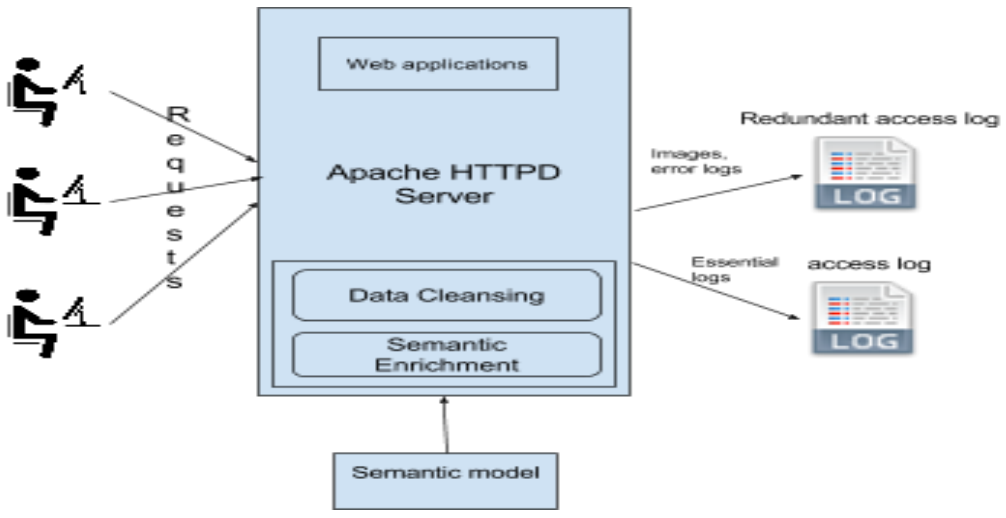

Figure 3.1 Real-Time Data Cleaning

The proposed real time data cleaning algorithm uses the Apache HTTP Server which results in generation of access.log file with semantically enriched essential log entries and access_redundant.log with non-essential log entries like images, javascript, error and CSS request. Figure 3.1 shows real-time data cleaning approach.

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 & ResponseSetEnvIfPlus.

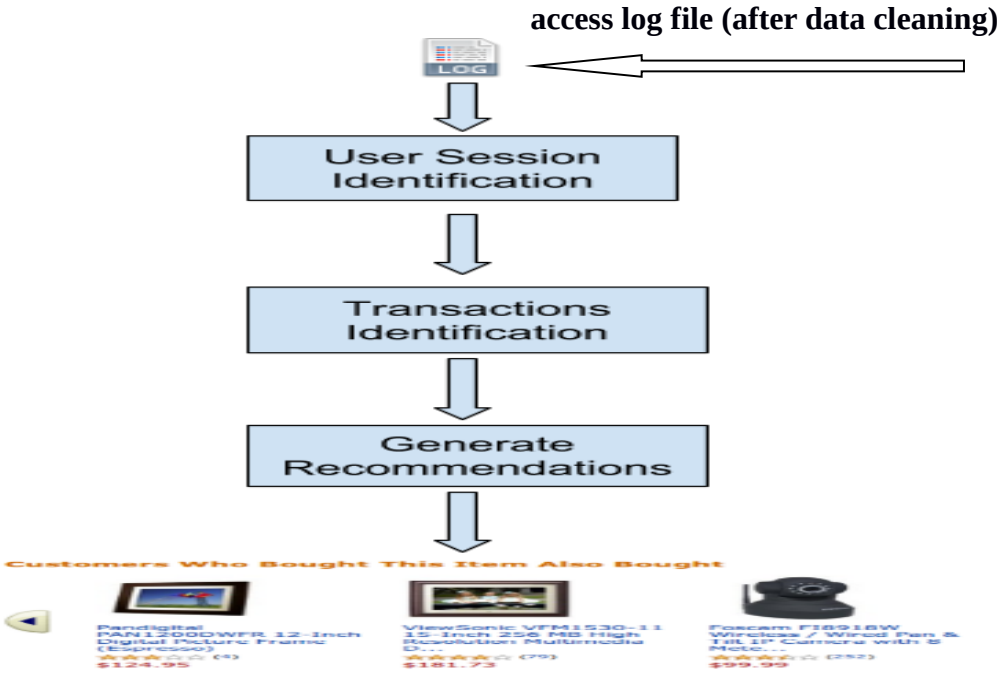

Figure 3.2 Real-Time Data Pre-processing

The proposed data pre-processing algorithm incorporate User Session Identification, Transaction Identification and Recommendation generation process. The User Session Identification process identifies unique user sessions. This process is followed by Transaction Identification which group the user actions namely: add_product, remove_product and order success by using the information of User Session Identification process. The Recommendation generation process generates real time user based recommendation using apache mahout and item based recommendations are generated offline using Apriori algorithm.

3.5 Contributions of Thesis

This thesis has contributed for real time data cleaning and data pre-processing system development. Some contributions are:

1. Modified the structure of existing Combined Log file format will be discussed in Chapter 4.
2. Perform real-time data cleaning process on the Web Server log files to significantly reduce the size of the log files will be discussed in Chapter 4.
3. Proposed and developed algorithm for real time Data cleaning and Data Pre-processing will be discussed in Chapter 4 and 5.
4. Real-time User Based recommendations are generated using Apache Mahout and Item Based recommendations are generated offline using Apriori algorithm will be discussed in Chapter 5
5. Chapter 6 and 7 shows the observation, results and conclusion of the proposed work.

CHAPTER -4

REAL-TIME DATA CLEANING

Chapter 4

REAL-TIME DATA CLEANING

This chapter describes the research methodology used for data cleaning process. Main objective of research is to significantly reduce the size of the Web server access log file, reduce the time required for data cleaning process and increase the quality of the data in the web server logs. The performance of traditional data cleaning process and the proposed real-time data cleaning process is compared.

4.1 Apache HTTPD Server

4.1.1 Introduction

Apache HTTPD server is used for proposed research work as it is the world widely used web server. The Apache Web Server operates on variety of operating systems including UNIX, Linux and Windows. Apache HTTPD Server is freeware and open-source software.

4.1.2 Architecture

The Apache architecture is based on a modular approach which allows easy extension with optional functionality, third party add-ons, and custom modifications. In order to achieve this modular approach, the Apache Server is divided into two main components:

Apache core - Implements the basic functionality of the server and includes a platform-dependent layer (the APR - Apache Portable Runtime).

Apache Modules - Implement/override/extend the functionality of the Apache web server and implement the different phases of handling http request.The figure 4.1 shows Apache web server architecture.

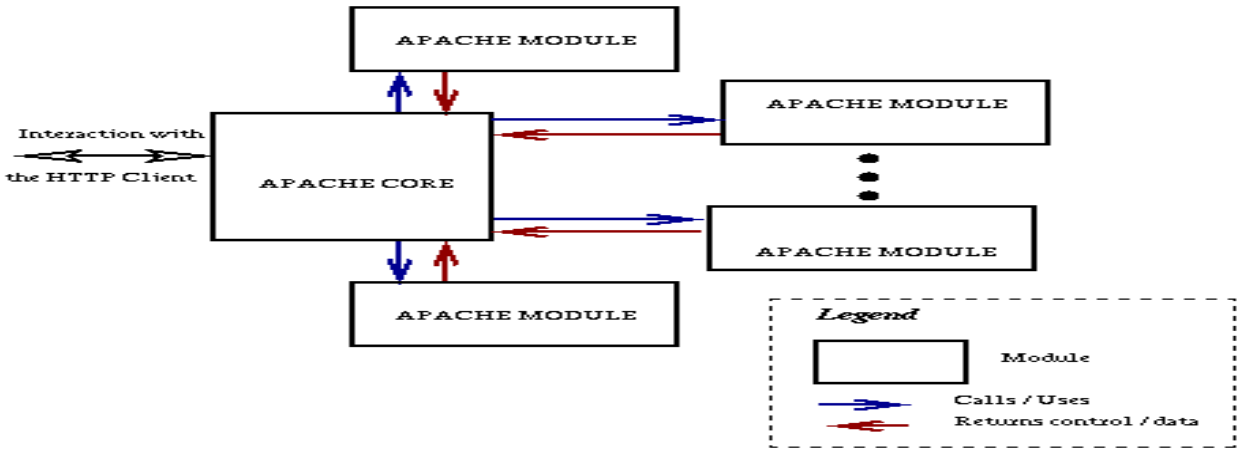

Figure 4.1 Apache Architecture

4.1.3 Installation

In this proposed work, Apache Server 2.2.22 is used. The Apache installation directory structure (Linux) is as below -

/etc/apache2

apache2.conf - Main server configuration file.

mods-enabled - Symbolic links to the Apache modules that are enabled.

mods-available - Pre-compiled modules installed as part of Apache2.

Sites-available/000-default.conf - Apache2 host configuration file.

/var/log/apache2

access.log - Log file that records all the requests processed by the server.

error.log - Log file to record any errors that are encountered in processing requests.

4.1.4 Apache Configuration Directives

i. SetEnvIf - This directive sets the environment variables based on attributes of the request.

Syntax - SetEnvIf attribute regex [!]env-variable[=value] [[!]env-variable[=value]]

Module -mod_setenvif

ii. SetEnvIfNoCase - This directive sets the environment variables based on attributes of the request without respect to case.

Syntax - SetEnvIfNoCase attribute regex [!]env-variable[=value] [[!]env-variable[=value]]

Module - mod_setenvif

iii. ResponseSetEnvIfPlus - This directive is applied at the time of HTTP response processing and against the HTTP response header. It also allows matching against the HTTP response status using the RESPONSE_STATUS attribute.

Syntax - ResponseSetEnvIfPlus<attribute><regex> [!]<env-variable>[=<value>]

Module -mod_setenvifplus

4.1.5 Apache Logging

i. access.log

The location and contents of the access.log file are controlled by the CustomLog Apache directive. The LogFormat directive defines the information that will be included in the logs.

ii. error.log

The location and contents of the error.log file are controlled by the ErrorLog directive.

Apache server records any diagnostic information and errors encountered in request processing in this log file.

iii. Existing Log Configurations

CustomLog ${APACHE_LOG_DIR}/access.log

LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined

%h is the remote host (i.e. the client IP).

%l is the identity of the user determined by identd (not usually used since not reliable).

%u is the user name determined by HTTP authentication.

%t is the time the request was received.

%r is the request line from the client that contains request method, request URL and HTTP protocol version.

%>s is the status code sent from the server to the client (200, 404 etc.).

%O is the size of the response to the client (in bytes).

%{Referrer}i is the page that linked to this URL.

%{User-agent}i is the browser identification string.

iv. Log Variables

%{VARNAME}C – Contents of the cookie VARNAME in the request is written to the log file.

%{VARNAME}e – Contents of environment variable VARNAME is written to the log file.

v. Conditional Logging

This is used to exclude some entries or choose from the defined log formats for the access.log based on the characteristics of the request. This is achieved with the help of environment variables. The environment variable can be set with 'SetEnvIf' or 'SetEnvIfNoCase' directives. Then the env clause of the CustomLog and/or LogFormat directive is used to include or exclude requests where the environment variable is set.

4.1.6 Apache Modules and Handlers

Modules are pieces of code which can be used to provide or extend functionality of the Apache HTTP Server. Modules can either be statically or dynamically included with the core. The module mod_so provides the functionality to add modules dynamically. The Apache server core handles the most common aspects of an HTTP conversation namely listening for a request, parsing the incoming request line and headers, composing the outgoing HTTP response message. Modules can define custom handlers that can hook

into the request processing phases.

i. Apache Perl Module

mod_perl is a Apache module that provides a persistent Perl interpreter embedded in the Apache web server. mod_perl provides various handlers that can be hooked into the http request processing phases. The figure 4.2 shows Apache Perl Module.

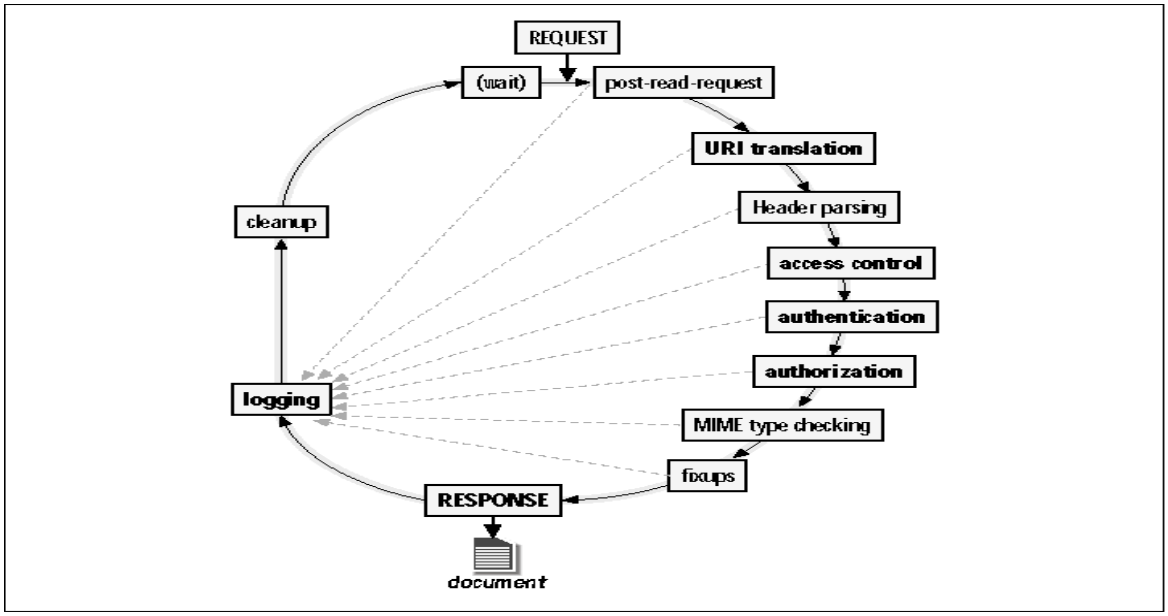

Figure 4.2 Apache Perl Module

mod_perl provides handlers for each of the 12 lifecycle phases. The proposed research uses two handlers namely -

- PerlPostReadRequestHandler (PerlInitHandler) - It is the first handler to be invoked when serving a request. Executes after the request has been read and is used for any processing. This handler is used in the research work to enrich the semantic information of the request in the log entries. Since this is the first handler to be executed, it is also used to record the start time of the request handling in this research.
- PerlLogHandler - This handler is always executed no matter how the previous phases have ended up. In this research, the handler is used to record the end time of the request handling.

4.2 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, javasript, 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. The following figure 4.3 shows Real-Time Data Cleaning.

4.2.1 System Design

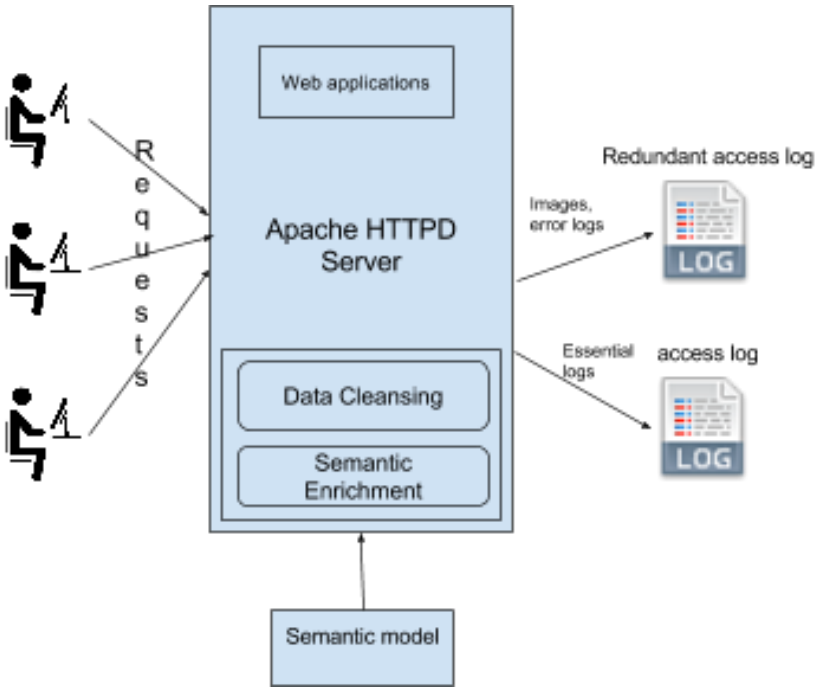

Figure 4.3 Real-Time Data Cleaning

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.

4.2.2 Implementation

4.2.2.1 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.

4.2.2.2 Host Configuration File

i. Set the environment variable to 'redundant' for http request with error response (4xx and 5xx) using the ResponseSetEnvIfPlus directive.

ResponseSetEnvIfPlus RESPONSE_STATUS ([4-5][0-9]+[0-9]+) redundant

ii. The Semantic Enrichment handler implemented to get the semantic information of the http request and set it in the request environment. The handler is registered as a PerlInitHandler in the request lifecycle phase.

PerlInitHandlerWebLogSystem::SemanticEnrichmentHandler

iii. Set the environment variable to 'redundant' for internal dummy connection log entries of internal requests within the server using the SetEnvIf directive.

SetEnvIfRemote_Addr "::1" redundant

iv. Set the environment variable to 'redundant' for Http requests for various image types using the SetEnvIfNoCase directive.

SetEnvIfNoCaseRequest_URI "\.gif$" redundant
SetEnvIfNoCaseRequest_URI "\.jpg$" redundant
SetEnvIfNoCaseRequest_URI "\.png$" redundant
SetEnvIfNoCaseRequest_URI "\.bmp$" redundant
SetEnvIfNoCaseRequest_URI "\.js$" redundant
SetEnvIfNoCaseRequest_URI "\.css$" redundant
SetEnvIfNoCaseRequest_URI "\.ico$" redundant

v. No change in the error.log configuration
ErrorLog ${APACHE_LOG_DIR}/error.log

vi. Log all the essential log entries to the original access.log file using conditional logging i.e.check for environment variable value is not redundant.
CustomLog ${APACHE_LOG_DIR}/access.log combined env=!redundant

vii. Log all the non-essential log entries to the access_redundant.log file using conditional logging .i.e. check for environment variable value is redundant.
CustomLog ${APACHE_LOG_DIR}/access_redundant.log combined_redundantenv=redundant

viii. Log format for access_redundant.log file
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"\"%{ttime}e\"" combined_redundant

4.2.2.3 Semantic Data Model

mysql> select * from semantic_data_model;

request_url	semantics
/tomatocart/checkout.php	action,pID
/tomatocart/json.php	action,pID

+-------------------------+------------+

The semantic_data_model table holds the list of semantic attributes for the http request URLs in the application (tomatocart in this case).

In case of tomatocart application the 'action' attribute will be:

add_product – For product added to the shopping cart.

remove_product – For product removed from the shopping cart.

success – For checkout and confirm order and the 'pID' attribute will be for the productId added/removed from the cart.

4.2.2.4 Semantic Enrichment Perl Handler

```
#file:WebLogSystem/SemanticEnrichmentHandler.pm
packageWebLogSystem::SemanticEnrichmentHandler;
use strict;
use warnings;
use Apache2::RequestRec ();
use Apache2::Request ();
use APR::Request ();
use Apache2::Cookie ();
use Apache2::Const -compile =>qw(OK);
use DBI;

sub handler {
 my $r = shift;
my ($seconds, $microseconds) = gettimeofday;
my $epoc = ($seconds * 1000 * 1000) + $microseconds;
$r->subprocess_env(ttime => $epoc);
 my $dbh = DBI->connect('dbi:mysql:mysemanticDB','root','root')
          or die "Connection Error: $DBI::errstr\n";
 my $sql = "select request_url, semantics from semantic_data_model";
 my $sth = $dbh->prepare($sql);
 $sth->execute or die "SQL Error: $DBI::errstr\n";
 my %semanticsMap;
 my @row;
 while (@row = $sth->fetchrow_array) {
```

```
                $semanticsMap{$row[0]} = $row[1];
}
my $requestURI = $r->uri;
my $semanticKeys = $semanticsMap{$requestURI};
my @semanticKeysArr = split(',', $semanticKeys);
my $req = APR::Request::Apache2->handle($r);
my @reqParams = $req->param;
my %params;
for my $param (@reqParams) {
        my $paramValue = $req->param($param);
                $params{$param} = $paramValue;
}
my $semantics;
my @keys = keys %params;
my $action;
foreach my $key (@semanticKeysArr) {
        if ($key eq 'action') {
                my $param = $params{$key};
                if ($param) {

                } elsif (scalar(@keys) >= 1) {
                        $param = $keys[0];
                }
                if ($parameq 'add_product' || $parameq 'cart_remove' || $parameq
                        'remove_product' || $parameq 'success') {
                                $action = $param;
                $semantics = "action=$action";
                }
                delete $params{$key};
        }
}
        if ($action && ($action eq 'add_product' || $action eq 'cart_remove' || $action eq
                                'remove_product')){
        foreach my $key (@semanticKeysArr) {
                        if ($key eq 'pID' || $key eq 'pQty') {
```

```
                        my $param = $params{$key};
                                if ($param) {
                        $semantics = "$semantics,$key=$param";
                                } elsif (scalar(@keys) >= 1) {
                        $semantics = "$semantics,$key=$keys[0]";
                                }
                                delete $params{$key};
                        }
                }
 }
  $r->subprocess_env(semantics => $semantics);
 my $cookiesJar = Apache2::Cookie::Jar->new($r);
 my $sidCookie = $cookiesJar->cookies("sid");
 my $userId;
 if ($sidCookie) {
                my $sessionId = $sidCookie->value();
                my $toc_dbh = DBI->connect('dbi:mysql:tomatocart','root','root')
                or die "Connection Error: $DBI::errstr\n";
        my $toc_sql = "select customer_id from toc_whos_online where session_id =
                                        '$sessionId'";
        my $toc_sth = $toc_dbh->prepare($toc_sql);
        $toc_sth->execute or die "SQL Error: $DBI::errstr\n";
        my @toc_row;
        if (@toc_row = $toc_sth->fetchrow_array) {
                        $userId = $toc_row[0];
         }
 }
 $r->subprocess_env(userid => $userId);
 return Apache2::Const::OK;
}
1;
```

The implementation of the above handler can be described as below:

- Connect to the mysemanticDB database and read the list of semantic attributes from the semantic_data_model table for the request URI.

- Read all the request parameters from the http request and stores them as key-values in the hash map.
- For each semantic attribute
    - Get the value of the semantic attribute from the request parameters.
    - Add the attribute name-value pair to the semantic info variable.
- Set the semantic information in the environment variable 'semantics'.
- Read the 'sid' session cookie from the request.
- Connect to the 'tomatocart' application database.
- Get the userId by querying the application database using the session id.
- Set the userId in the 'userid' environment variable.

4.2.2.5 TomatoCart Application

In the proposed work TomatoCart e-commerce shopping cart website used for demonstration purpose. This is a free and open-source website installed on the Apache Web server.

i. Logged-in with a user 'tom@example.com' to the application and add products to the shopping cart. Figure 4.4 shows Tomato Cart Application Add Products.

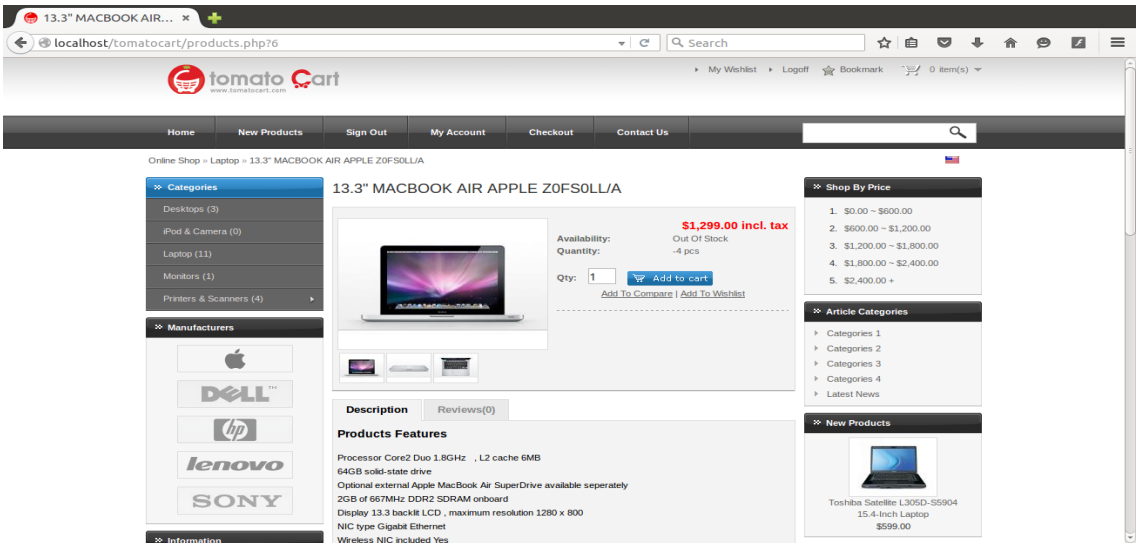

Figure 4.4 Tomato Cart Application Add Products

ii. Then check-out the products from the cart and complete the order. Figure 4.5 shows Tomato Cart Application Check-Out Products

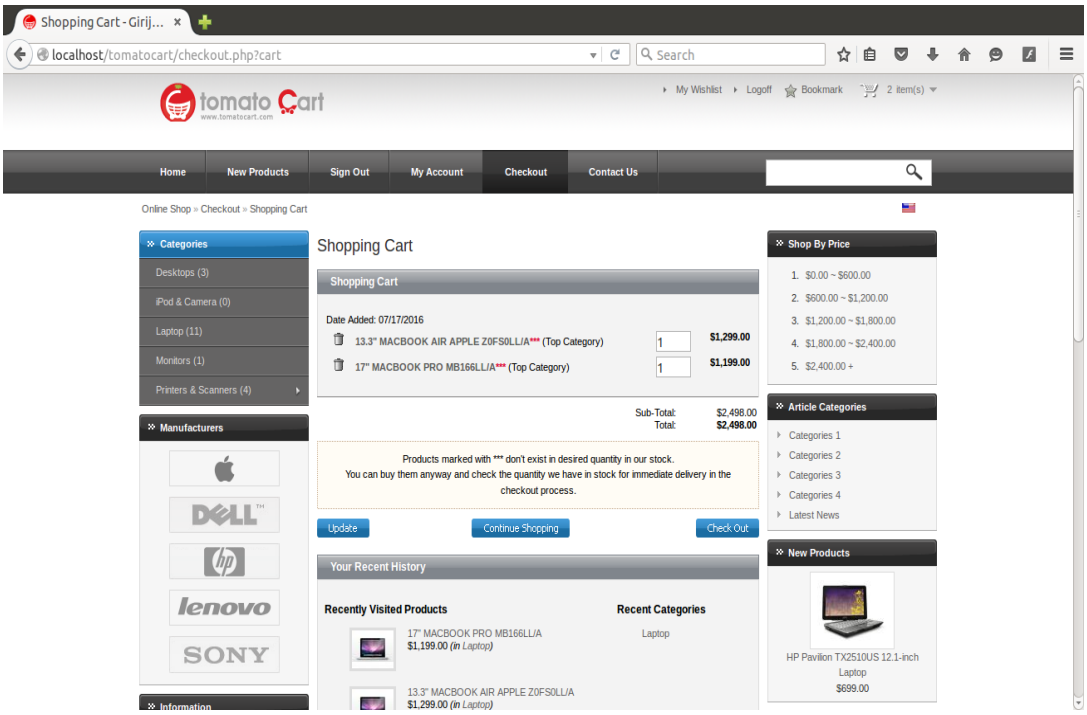

Figure 4.5 Tomato Cart Application Check-Out Products

iii. Successfully completed the order.Figure 4.6 shows Tomato Cart Application Order Complete process successfully.

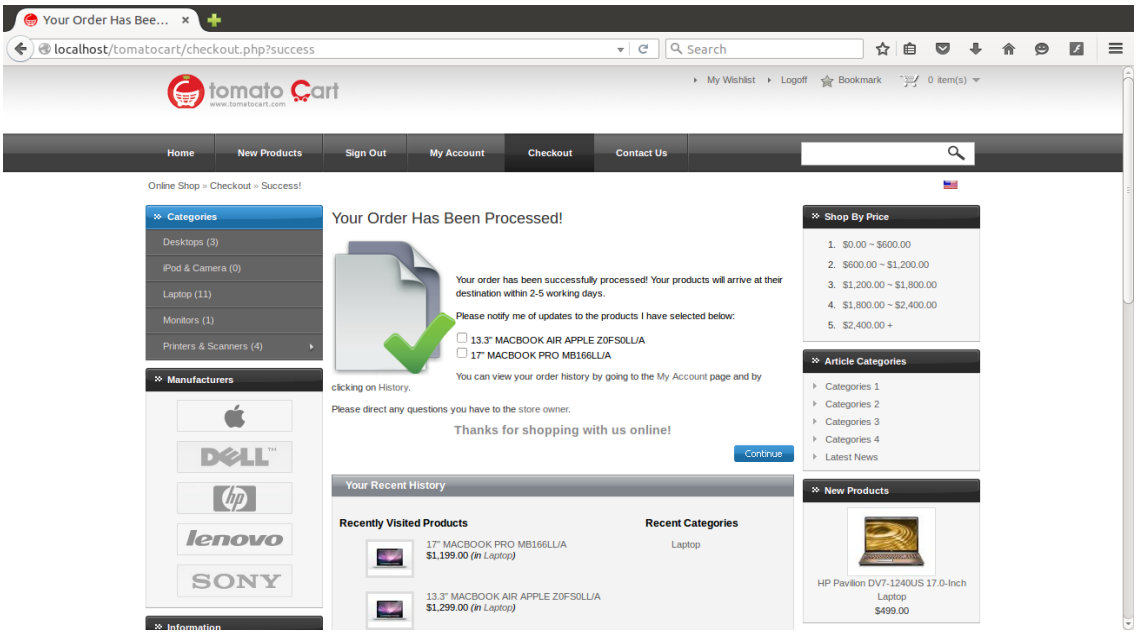

Figure 4.6 Tomato Cart Application Order Complete process successfully

For each of the requests corresponding to the user actions for add product and confirm order, the perl handler will enrich the semantic information namely the action and product id in the logs. The session id and user id, processing time are also added to the

log entries as shown in the figure 4.7 of the access.log file.

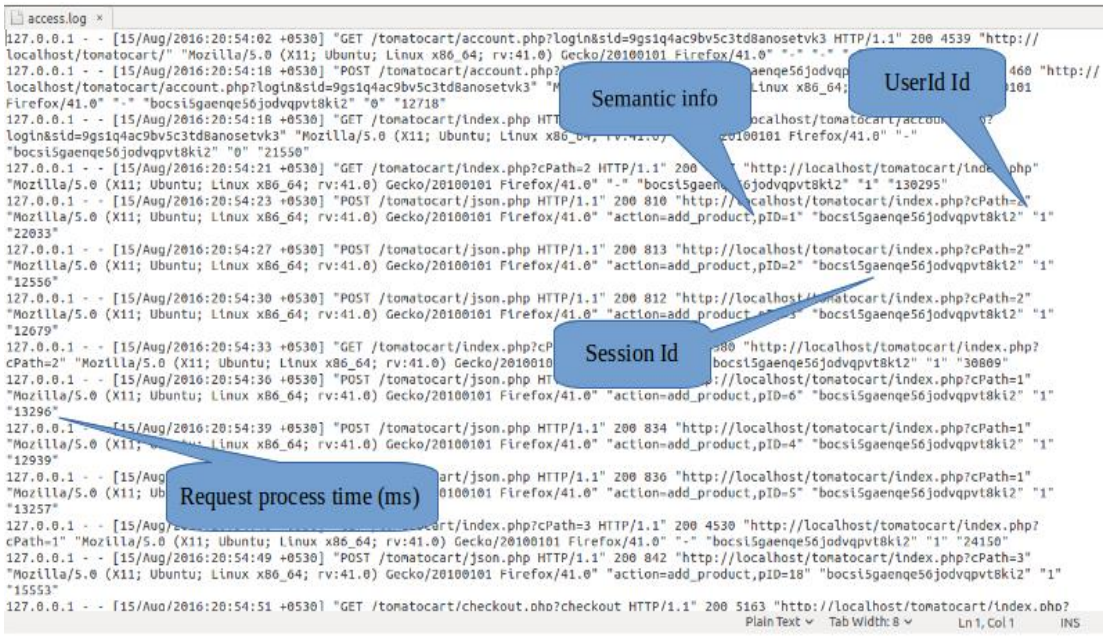

Figure 4.7 access.log

All the other non-essential requests like images, javascripts, css, error response and internal dummy connections are logged in the access_redundant.log file as shown in the figure 4.8

Figure 4.8 access_redudant.log

4.3 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 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 (traditional data cleaning).

To compute the request processing times without real-time data cleaning solution, the below mod_perl handlers are configured in the Apache host configuration file 'default.conf'. The StartTimeHandler is registered as PerlInitHandler to record the start time of the request handling, while the EndTimeRecordHandler is registered as PerlLogHandler to compute the time (in micro seconds) required for the request processing. The time is recorded in the log entries using the "%{ttime}e"field added in the LogFormat directive in the Apache configuration file.

```
#file:WebLogSystem/StartTimeHandler.pm
#--------------------------
packageWebLogSystem::StartTimeRecordHandler;
use strict;
use warnings;
use Apache2::RequestRec ();
use Apache2::Request ();
use APR::Request ();
use Time::HiResqw(gettimeofday);

sub handler {
my $r = shift;
my ($seconds, $microseconds) = gettimeofday;
my $epoc = ($seconds * 1000 * 1000) + $microseconds;
```

```
$r->subprocess_env(ttime => $epoc);
}
1;

#file: WebLogSystem/EndTimeRecordHandler.pm
#--------------------------
packageWebLogSystem::CleanupHandler;
use strict;
use warnings;
use Apache2::RequestRec ();
use Apache2::Request ();
use APR::Request ();
use Time::HiResqw(gettimeofday);

sub handler {
my $r = shift;
my $ttime = $r->subprocess_env('ttime');
my ($seconds, $microseconds) = gettimeofday;
my $epoc = ($seconds * 1000 * 1000) + $microseconds;
$ttime = $epoc - $ttime;
$r->subprocess_env(ttime => $ttime);
}
1;
```

The start time of the request processing is computed using the SemanticEnrichmentHandler (described in section 4.2.2.4). The EndTimeRecordHandler which is registered as PerlLogHandler is used to compute the time required for processing the request and record that time (in micro seconds) for the log entry in the access.log or the access_redundant.log file.

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

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

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

$\mathbf{r_j}$= 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)

```
public static void computeOnlineDatacleaningTime() {
Map<String, Integer>onlineURLsCountMap = newHashMap<String, Integer>();
Map<String, Integer>offlineURLsCountMap = newHashMap<String, Integer>();

Map<String, Integer>onlineURLsTimeMap = newHashMap<String, Integer>();
Map<String, Integer>offlineURLsTimeMap = newHashMap<String, Integer>();

Map<String, Integer>onlineURLsAvgTimeMap = newHashMap<String, Integer>();
Map<String, Integer>offlineURLsAvgTimeMap = newHashMap<String, Integer>();

Map<String, Integer>onlineURLsFinalTimeMap = newHashMap<String, Integer>();

processLogFile(ONLINE_ACCESS_LOG, onlineURLsCountMap,
                onlineURLsTimeMap, PATTERN);
processLogFile(ONLINE_ACCESS_REDUNDANT_LOG, onlineURLsCountMap,
                        onlineURLsTimeMap, PATTERN2);
processLogFile(OFFLINE_ACCESS_LOG, offlineURLsCountMap,
                onlineURLsTimeMap, PATTERN2);

Set<String>onlineURLsKeys = onlineURLsCountMap.keySet();
for (String onlineURLsKey : onlineURLsKeys) {
        onlineURLsAvgTimeMap.put(onlineURLsKey,
        (onlineURLsTimeMap.get(onlineURLsKey)/onlineURLsCountMap.get(onlineURLsKey
        )));
}
```

```
Set<String>offlineURLsKeys = offlineURLsCountMap.keySet();
for (String offlineURLsKey : offlineURLsKeys) {
        offlineURLsAvgTimeMap.put(offlineURLsKey,
        (offlineURLsTimeMap.get(offlineURLsKey)/offlineURLsCountMap.get(offlineURLsKe
        y)));
}
for (String onlineURLsKey : onlineURLsKeys) {
        if (offlineURLsAvgTimeMap.containsKey(onlineURLsKey)) {
                onlineURLsFinalTimeMap.put(onlineURLsKey,
                        ((onlineURLsAvgTimeMap.get(onlineURLsKey) -
                                offlineURLsAvgTimeMap.get(onlineURLsKey)) *
                                        onlineURLsCountMap.get(onlineURLsKey)));
        }
}

        inttotalCleaningTime = 0;
        Set<String>onlineFinalURLsKeys = onlineURLsFinalTimeMap.keySet();
        for (String onlineURLsKey : onlineFinalURLsKeys) {
                totalCleaningTime = totalCleaningTime +
                                        onlineURLsFinalTimeMap.get(onlineURLsKey);
        }
        System.out.println("Real-time Data cleaning total time = "+totalCleaningTime/1000 + "
        millisecs");
}
publicstaticvoidprocessLogFile(String logFile, Map<String, Integer>     URLsCountMap,
Map<String, Integer>URLsTimeMap,    Pattern PATTERN) {
        FileReaderfileReader = null;
        BufferedReaderbuffReader = null;
        try {
                fileReader = newFileReader(logFile);
                buffReader = newBufferedReader(fileReader);
                String logEntry = null;
                        while ((logEntry = buffReader.readLine()) != null) {
                                ApacheAccessLogapacheAccessLog =
                                        parseLogEntry(logEntry, PATTERN);
                                if (apacheAccessLog != null) {
                                        String endpoint = apacheAccessLog.getEndpoint();
                                                intindex = endpoint.indexOf('?');
```

```
                                        if (index> -1) {
                                                endpoint =
                                                endpoint.substring(0, index);
                                        }
                                    if (!URLsCountMap.containsKey(endpoint)) {
                                            URLsCountMap.put(endpoint, 0);
                                    }
                                    intcount = URLsCountMap.get(endpoint);
                                    count++;
                                    URLsCountMap.put(endpoint, count);

                                    if (!URLsTimeMap.containsKey(endpoint)) {
                                            URLsTimeMap.put(endpoint, 0);
                                    }
                                    intttime = URLsTimeMap.get(endpoint);
                                    ttime = ttime + apacheAccessLog.getTtime();
                                    URLsTimeMap.put(endpoint, ttime);
                            }
                        }
                    } catch (Exception e) {
                        e.printStackTrace();
                    } finally {
                        buffReader.close();
                        fileReader.close();
                        }
                    }
            }
}
```

4.4 Traditional Data Cleaning

The traditional process of data cleaning is removal of outliers or irrelevant data, carried out offline on web log file. Analysing the massive amounts of records in server logs is a complex activity. In traditional data cleaning process records of multi-media resources like GIF, JPEG, CSS and so forth can be removed from the web log file and moved to the redundant log file. The HTTP status code is then considered for further cleaning. The records with status code over 299 or below 200 are removed from the web access log. Figure 4.9 shows traditional data cleaning.

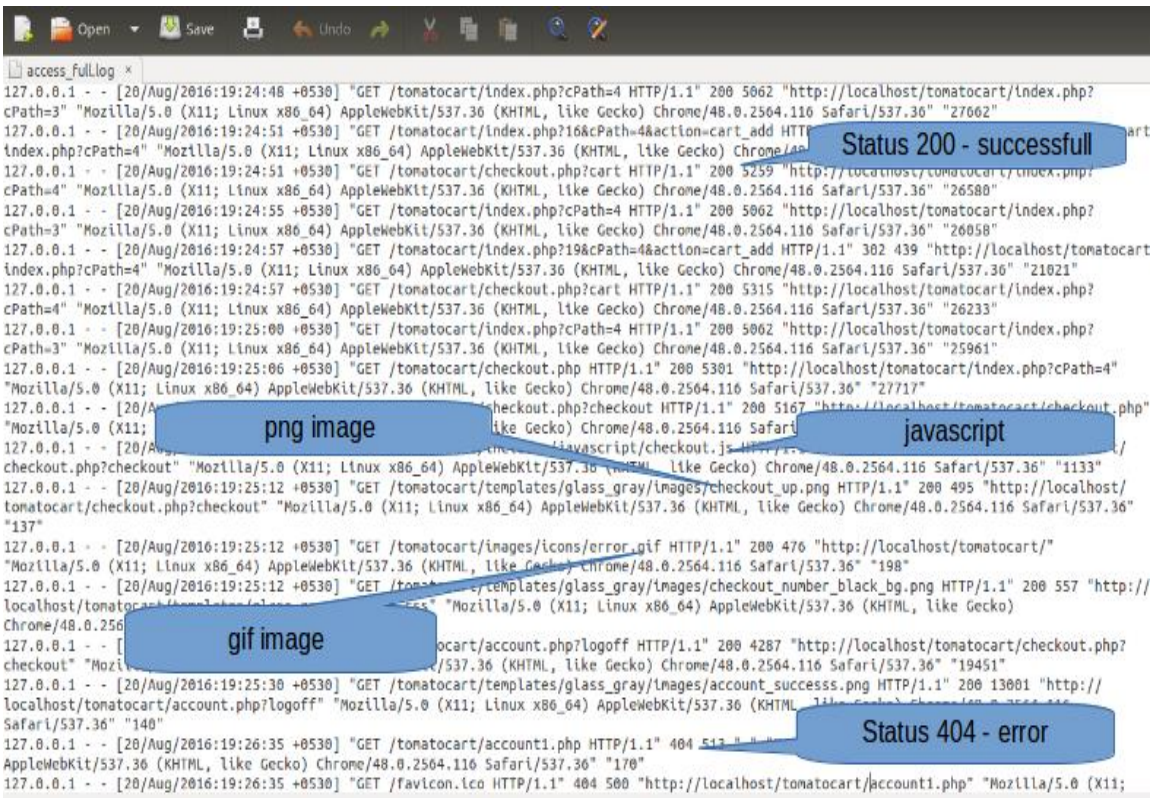

Figure 4.9 access_full.log

4.4.1 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.

4.4.2 Implementation

```
privatestaticfinal String LOG_ENTRY_PATTERN = "^(\\S+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(\\S+) (\\S+) (\\S+)\" (\\d{3}) (\\d+) \"([\\S]+)\" \"([\\S\\s]+)\" \"(\\d+)\"";

publicvoidprocessLogFile() {
        FileReaderfileReader = null;
        BufferedReaderbuffReader = null;
        FileWriterfileWriter = null;
        BufferedWriterbuffWriter = null;
        FileWriterfileWriter2 = null;
        BufferedWriterbuffWriter2 = null;
        try {
                fileReader = newFileReader(accessFullLogFile);
                buffReader = newBufferedReader(fileReader);
                fileWriter = newFileWriter(accessLogFile);
                buffWriter = newBufferedWriter(fileWriter);
                fileWriter2 = newFileWriter(accessRedundantFile);
                buffWriter2 = newBufferedWriter(fileWriter2);
                String logEntry = null;
                while ((logEntry = buffReader.readLine()) != null) {
                        ApacheAccessLogapacheAccessLog =
                                        parseLogEntry(logEntry);
                        if (apacheAccessLog != null) {
                        if(apacheAccessLog.getEndpoint().endsWith(".gif")
                                ||apacheAccessLog.getEndpoint().endsWith(".jpg")
                                || apacheAccessLog.getEndpoint().endsWith(".png")
                                || apacheAccessLog.getEndpoint().endsWith(".bmp")
                                || apacheAccessLog.getEndpoint().endsWith(".js")
                                || apacheAccessLog.getEndpoint().endsWith(".css")
                                || apacheAccessLog.getEndpoint().endsWith(".ico")
                                || apacheAccessLog.getEndpoint().endsWith(".GIF")
                                || apacheAccessLog.getEndpoint().endsWith(".JPG")
                                || apacheAccessLog.getEndpoint().endsWith(".PNG")
                                || apacheAccessLog.getEndpoint().endsWith(".BMP")
                                || apacheAccessLog.getEndpoint().endsWith(".JS")
                                || apacheAccessLog.getEndpoint().endsWith(".CSS")
                                || apacheAccessLog.getEndpoint().endsWith(".ICO")
                                || apacheAccessLog.getResponseCode() == 404
```

```
                    || apacheAccessLog.getIpAddress().equals("::1")) {
                    buffWriter2.write(logEntry);
                    buffWriter2.write(System.lineSeparator());
                    buffWriter2.flush();
                } else {
                    buffWriter.write(logEntry);
                    buffWriter.write(System.lineSeparator());
                    buffWriter.flush();
                }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            buffReader.close();
            fileReader.close();
            buffWriter.close();
            fileWriter.close();
            buffWriter2.close();
            fileWriter2.close();
        }
}

public ApacheAccessLogparseLogEntry(String logline) {
        Matcher m = PATTERN.matcher(logline);
        if (m.find()) {
            returnnewApacheAccessLog(m.group(1), m.group(2), m.group(3), m.group(4),
            m.group(5), m.group(6), m.group(7), m.group(8), m.group(9), m.group(10),
            m.group(11), null, null, m.group(12));
        }
        return null;
}
```

4.4.3 Traditional Data Cleaning Time Evaluation

```
public static void main(String[] args) {
        longstarttime = System.currentTimeMillis();
        ApacheLogFileCleanerlogFileCleaner =
        newApacheLogFileCleaner("/var/log/apache2/access_full.log");
```

```
        logFileCleaner.processLogFile();
        longendtime= System.currentTimeMillis();
        System.out.println("Offline Data cleaning total time = " + (endtime - starttime) + "
        millisecs");
}
```

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 as mentioned in the above code.

4.5 Time Comparison between Real-time Data Cleaning and Traditional Data Cleaning

The table 4.1 shows the comparison matrix of traditional data cleaning and real-time data cleaning. It can be observed that real-time data cleaning requires around 12% less time than traditional data cleaning.

No. of Records in Log File (approx)	Traditional Data Cleaning (milli-secs)	Proposed Real-Time Data Cleaning (milli-secs)	% Reduction in Time
800	141	123	12.75
1500	276	240	13.04
2500	328	285	13.1
3500	376	334	11.3
4500	435	384	11.72

Table 4.1 Comparison Matrix for Data Cleaning Process

CHAPTER -5

REAL-TIME DATA PRE-PROCESSING

AND

RECOMMENDATION GENERATION

Chapter 5

REAL-TIME DATA PRE-PROCESSING AND RECOMMENDATION GENERATION

This chapter, describes proposed methodology of real-time data pre-processing. The methodology is applied on the TomatoCart ecommerce application with Web access logs as input and generates the real-time user based recommendations and offline item based recommendations.

5.1 Application for Experimental Evaluation

The Java based application is developed for real-time data pre-processing in proposed research work. Apache Mahout Recommender tool is used for generating the real-time user based recommendations and offline item based recommendations are generated using TANAGRA data mining tool.

5.1.1 System Design

The proposed system in the research will comprise of 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.

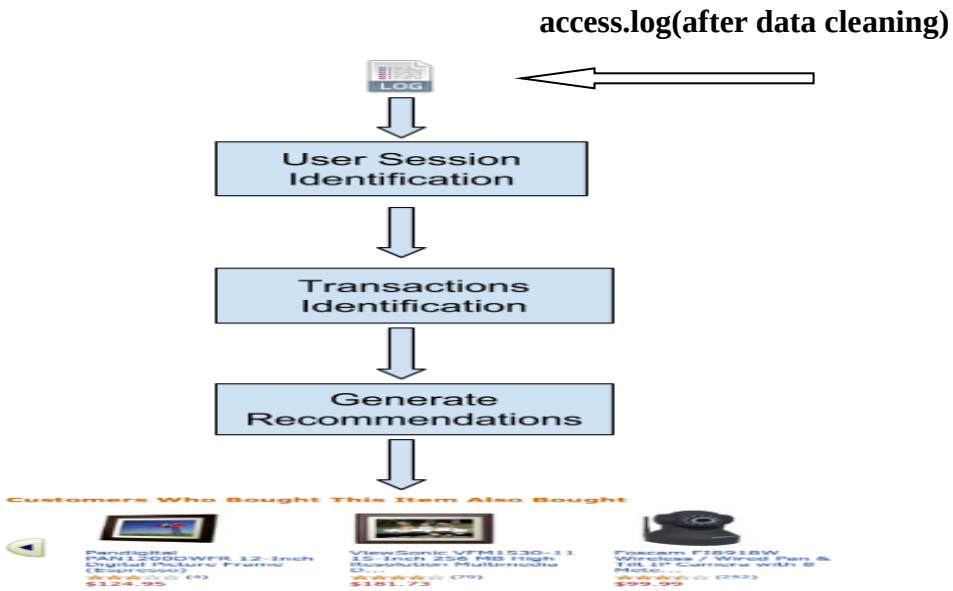

Figure 5.1 Real-Time Data Pre-processing

5.1.2 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: Input the CSV dataset file to recommendation tool and generate user based recommendations for each of the users using TanimotoCoefficientSimilarity method.

Step 13: Repeat above steps every 5 minutes.

5.2 Implementation

The ApacheAccessLogProcessor method will be executed after every 5 minutes to process the log file for any new entries like; identify user sessions, purchase transactions and generate user recommendations. The ApacheAccessLogProcessor will take the access log file, the position of last processed log entry and last processed time as input.

```
ApacheAccessLogProcessor apacheAccessLogProcessor
                                        = new ApacheAccessLogProcessor(logFile,
lastProcessedLogEntry, lastProcessedTime);
lastProcessedLogEntry = apacheAccessLogProcessor.processLogFile();
lastProcessedTime = System.currentTimeMillis();
apacheAccessLogProcessor.sessionizeLogEntries();
apacheAccessLogProcessor.transactionizeLogEntries();

CSVFileProcessorcsvFileProcessor = new CSVFileProcessor("dataset.csv");
csvFileProcessor.updateCSVDataSet(apacheAccessLogProcessor.getTransactionizedLogEn
tries());

GenerateRecommendationsgenerateRecommendations = new GenerateRecommendations();
generateRecommendations.generate("dataset.csv", csvFileProcessor.getAllUsers());
```

5.2.1 User Session Identification

The processLogFile method from ApacheAccessLogProcessor will proceed with processing of the log file only if it finds that the log file has changed. It will read the log file and will parse each of the newly added log entries using a pattern matcher and below

regular expression. The regular expression is used to parse the log entry that is in Combined Log Format in addition. The regular expression only matches with those log entries that have the semantic information .i.e. those entries that are part of user order transaction. All the ApacheAccessLog objects will be added to the semanticLogEntries collection. At the end it will record the position of last ProcessedLogEntry with a successful transaction and the last processed time.

```
String LOG_ENTRY_PATTERN = "^(\\S+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(\\S+)
(\\S+) (\\S+)\" (\\d{3}) (\\d+) \"([\\S]+)\" \"([\\S\\s]+)\" \"(action=[\\S]+)\" \"([\\S]+)\"
\"(\\d+)\"";

public long processLogFile() {
        longlastProcessedLogEntry = this.lastProcessedLogEntry;
        if (hasLogFileChanged()) {
                FileReaderfileReader = null;
                BufferedReaderbuffReader = null;
                try {
                        fileReader = newFileReader(logFile);
                        buffReader = newBufferedReader(fileReader);
                        String logEntry = null;
                        longlogEntryCounter = 0;
                        while ((logEntry = buffReader.readLine()) != null) {
                                logEntryCounter++;
                                if (logEntryCounter>lastProcessedLogEntry) {
                                        ApacheAccessLogapacheAccessLog =

        parseLogEntry(logEntry);
                                        if (apacheAccessLog != null) {
                                                semanticLogEntries.add(apacheAccessLog);
                                        if(isSuccessTransactionLogEntry(apacheAccessLog)) {
                                        lastProcessedLogEntry = logEntryCounter;
                                }
                                        }
                                }
                        }
                logger.log(Level.INFO, "Last processed log entry - " +
        lastProcessedLogEntry);
                logger.log(Level.INFO, "Last processed log time - " + new
```

```
                Date(lastProcessedTime));
                } catch (Exceptione) {
                        e.printStackTrace();
                }
        }
        returnlastProcessedLogEntry;
}
```

The sessionizeLogEntries method will group all the ApacheAccessLog objects from the semanticLogEntries collection by the user session ids. The sessionizedLogEntriesRegistry Map will contain the list of the ApacheAccessLog objects for each session. Thus the sessionization is done for the log entries. The sessionizedLogEntriesRegistry Map will be input to the next step of Transaction identification.

```
public void sessionizeLogEntries() {
        for (ApacheAccessLogapacheAccessLog : semanticLogEntries) {
                List<ApacheAccessLog>sessionizedLogEntries =
                        sessionizedLogEntriesRegistry.get(apacheAccessLog.getSession());
                if (sessionizedLogEntries == null) {
                        sessionizedLogEntries = newArrayList<ApacheAccessLog>();
                sessionizedLogEntriesRegistry.put(apacheAccessLog.getSession(),
                sessionizedLogEntries);
                }
                sessionizedLogEntries.add(apacheAccessLog);
        }
}
```

5.2.2 Transaction Identification

In this method for each session, successful user order transactions will be identified and the ApacheAccessLog entries will be grouped by transactions. The transactions will be identified using the semantic information 'action=success'.

A TransactionizedUserLogEntries object will be created to hold the ApacheAccessLog entries for a transaction. The transactionizedLogEntriesRegistry will hold the list of transactions for each session.

```
private void transactionizeLogEntries(String session) {
        List<ApacheAccessLog>sessionizedLogEntries =
        sessionizedLogEntriesRegistry.get(session);

        List<TransactionizedUserLogEntries>sessionTransactions =
                                        transactionizedLogEntriesRegistry.get(session);
        if (sessionTransactions == null) {
                sessionTransactions = new
                                                ArrayList<TransactionizedUserLogEntries>();
                transactionizedLogEntriesRegistry.put(session, sessionTransactions);
        }

        boolean isNewTransaction = true;
        TransactionizedUserLogEntriescurrTransactionLogEntries = null;
        for (ApacheAccessLogsessionizedLogEntry : sessionizedLogEntries) {
                if (isNewTransaction) {
                                        currTransactionLogEntries = new
                                TransactionizedUserLogEntries();
                        currTransactionLogEntries.setSessionId(session);
        currTransactionLogEntries.setUserId(sessionizedLogEntry.getUserID());
                                isNewTransaction = false;
                }
                currTransactionLogEntries.addLogEntry(sessionizedLogEntry);

                booleanisTransactionClosed =
                        sessionizedLogEntry.getAction().equals("action=success");
                if (isTransactionClosed) {
                        sessionTransactions.add(currTransactionLogEntries);
                        isNewTransaction = true;
                }
        }
}
```

5.2.3 Recommendations Generation

The proposed work generates user based recommendations as well as item based recommendations. Apache Mahout is used to implement the real-time user recommendations. Mahout is a collection of scalable machine learning algorithms. It

supports many algorithms such as collaborative filtering, clustering, classification and so on.

Offline Item Based Recommendations are generated using Apriori algorithm of free TANAGRA data mining software. Tanagra is widely used by academic and research purposes. Data mining methods like exploratory data analysis, statistical learning and machine learning offered by TANAGRA software

5.2.3.1 Real-Time User Based Recommendations

Mahout has a good implementation of many similarity algorithms. This allows developers to design and implement collaborative filtering recommender systems by using similarity algorithms and identifying similar neighbourhoods for different users, or to find out similarities between items.

The real-time user based recommendations are generated by using following three measures:

i. Collaborative Filtering:

Collaborative recommendation follows the concept that many users might have similar shopping interests. In other words, if some users have already purchased some similar items in the past they may also have interest in same items in the future. This means that if users A and B have purchased the same movies in the past, and appear to have similar purchase records, then if user A buys a movie that user B has not yet seen, the recommender system will suggest that movie to user B. Because of the indirect co-operation between the users, it is called Collaborative filtering.

This type of recommendation is widely used on many ecommerce websites. One of the advantages of this system is that no additional information about the item is required to provide the recommendation. In other words, the system does not need to have information about the item itself. For instance it is not necessary for the recommender system to know the name, genre or the director of the movie. The recommender system only needs to know the unique identifier of the item.

Collaborative filtering algorithms utilize the similarity between data like preferences of users, neighbourhoods and items to be able to recommend desirable items from a large

number of options.

ii. TanimotoCoefficientSimilarity:

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 as boolean type is available. Recommendations are more accurate when preferences are not considered. Recommender with Tanimoto Coefficient Similarity is good choice for recommending research papers (Ashish kumar B.,et. al.;2012).The following formula is used to compute TanimotoCoefficientSimilarity.

$$Jaccard(X,Y) = \frac{X \cap Y}{X \cup Y}$$

Tanimoto/Jaccard Coefficient

iii. Neighbourhood-based Recommendation:

Recommender systems, which are based on nearest-neighbours, automate the process of prediction, in the way that one is dependent on the opinion of people who have similar or identical opinions in order to evaluate the value of an item based on his or her preferences.

Implementation of Real-Time User Based Recommendations:

A list of all the TransactionizedUserLogEntries objects will be passed to this method. In this method for each user order transaction, a UserOrderDetails object will be created that will hold the userId and the products purchased by the user in that transaction. Based on each of the UserOrderDetails objects, the userId and purchased productId records will be added to the CSV file.

```
public void updateCSVDataSet(List<TransactionizedUserLogEntries>
                                        transactionizedUserLogEntries) {
    BufferedWriterbuffWriter = null;
    FileWriterfWriter = null;
    try {
        List<UserOrderDetails>userOrderDetailsList =
        getUserOrderDetails(transactionizedUserLogEntries);
        fWriter = newFileWriter(this.csvFile, true);
```

```
                buffWriter = newBufferedWriter(fWriter);
                booleanfirstEntry = isEmptyFile();
                for (UserOrderDetailsuserOrderDetails : userOrderDetailsList) {
                        for (String product : userOrderDetails.getProducts()) {
                                if (firstEntry) {
                                        firstEntry = false;
                                } else {
                                        buffWriter.write(System.lineSeparator());
                                }
                                buffWriter.write(userOrderDetails.getUserId() + "," +
                        product );
                        }
                }
                buffWriter.flush();
        } catch (Exception e) {
                        e.printStackTrace();
        }
}
```

The getUserOrderDetails method will create the UserOrderDetails objects from the user transactions using the semantic information in the log entries. The UserOrderDetails will check for the 'action=add_product' in the semantic information and add those products to the UserOrderDetails object for the user transaction. It will also check for 'action=remove_product' or 'action=cart_remove' and remove the products from the UserOrderDetails object. Thus each UserOrderDetails object will hold the user id and the products purchased by the user in a transaction.

```
public List<UserOrderDetails>
                                getUserOrderDetails(List<TransactionizedUserLogEntries>
                                        transactionUserLogEntries) {
        List<UserOrderDetails>userOrderDetailsList = new

        ArrayList<UserOrderDetails>();
        for (TransactionizedUserLogEntriestransactionizedUserLogEntries :
                                                        transactionUserLogEntries) {
                UserOrderDetailsuserOrderDetails = newUserOrderDetails();
        userOrderDetails.setUserId(transactionizedUserLogEntries.getUserId());
        userOrderDetails.setSessionId(transactionizedUserLogEntries.getSessionId());
```

```
                for (ApacheAccessLogapacheAccessLog :
                                transactionizedUserLogEntries.getLogEntries()) {
                        String action = apacheAccessLog.getAction();
                        if (action.startsWith("action=add_product")) {
                                String productId = StringUtils.substringAfter(action,
                        ",pID=");
                                userOrderDetails.addProduct(productId);
                        } elseif (action.startsWith("action=cart_remove") ||
                        action.startsWith("action=remove_product")) {
                                String productId =                              `
                                        StringUtils.substringAfter(action, ",pID=");
                                userOrderDetails.removeProduct(productId);
                        }
                }
                userOrderDetailsList.add(userOrderDetails);
        }
        returnuserOrderDetailsList;
}
```

The generate method takes the CSV dataset file and the userIds,productIds as input and generates the user recommendations for each of thoseusers. The method uses Apache Mahout user-based recommender to generate the product recommendations for the users. The dataset from the CSV file is loaded using the FileDataModel Mahout class. The TanimotoCoefficientSimilaritymethod is used to identify users with similar behaviour for the products and generate the product recommendations for each user.

```
public void generate(String csvFile, Collection<String>userIds) {
        DataModelmodel;
        try {
                model = newFileDataModel(new File(csvFile));
                UserSimilaritysimilarity = new TanimotoCoefficientSimilarity(model);
                UserNeighborhoodneighborhood = new ThresholdUserNeighborhood(0.1,
                similarity, model);
                UserBasedRecommenderrecommender = new
                        GenericUserBasedRecommender(model, neighborhood, similarity);
                for (String user : userIds) {
                        longuserId = Long.parseLong(user);
                        System.out.println("Recommendations for User - " + userId);
                        List<RecommendedItem>recommendations =
```

```
                    recommender.recommend(userId, 20);
                for (RecommendedItemrecommendation : recommendations) {
                    System.out.println(recommendation);
                }
                System.out.println();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
}
```

5.2.3.2 Offline Item Based Recommendations

Numerous algorithms can be applied to mine association rules from the data available; a standout amongst the most utilized and famous is the Apriori algorithm put forth and described by Agrawal and Srikant in the year 1994. Given the minimum support and confidence levels, this algorithm can swiftly give back rules from a data set through the discovery of large item set.

For example, if one finds that 70% of the user accessing company/products/laptop.html also accessed company/products/printer.html, yet just 30% of those who accessed company/products also accessed company/products/printer.html, then it is likely that some information in laptop.html leads users to access printer.html.

This relationship may recommend that this data ought to be moved to a better stage to increase access to printer.html. This additionally helps in making business strategy that people who wish to buy laptop; they are likewise inclined to purchase a printer. Discovery of such rules for organizations engaged in electronic commerce can help in the development of effective sales strategies. Aside from being exploited for business applications, the associations can likewise be utilized for Web recommendation, personalization or improving the system's performance through predicting and pre fetching of Web data.

There are three approaches to measure association:

1: Support, is measured by the proportion of transactions wherein an item-set appears.

2: Confidence, This says how likely object B is purchased when item A is purchased, expressed as {A -> B}. This is measured by the proportion of transactions with item A,

in which item B also appears.

3: Lift, This says how likely item B is purchased when item A is purchased, while at the same time controlling for how popular item B is. The lift value 1 implies that there is no relationship between items. A lift value more than 1 signifies that item B is likely to be purchased if item A is purchased, while a value lower than 1 implies that item B is unlikely to be purchased if item A is purchased.

Figure 5.2, 5.3, 5.4 shows the result of Apriori algorithm with support 30% and confidence 75%.

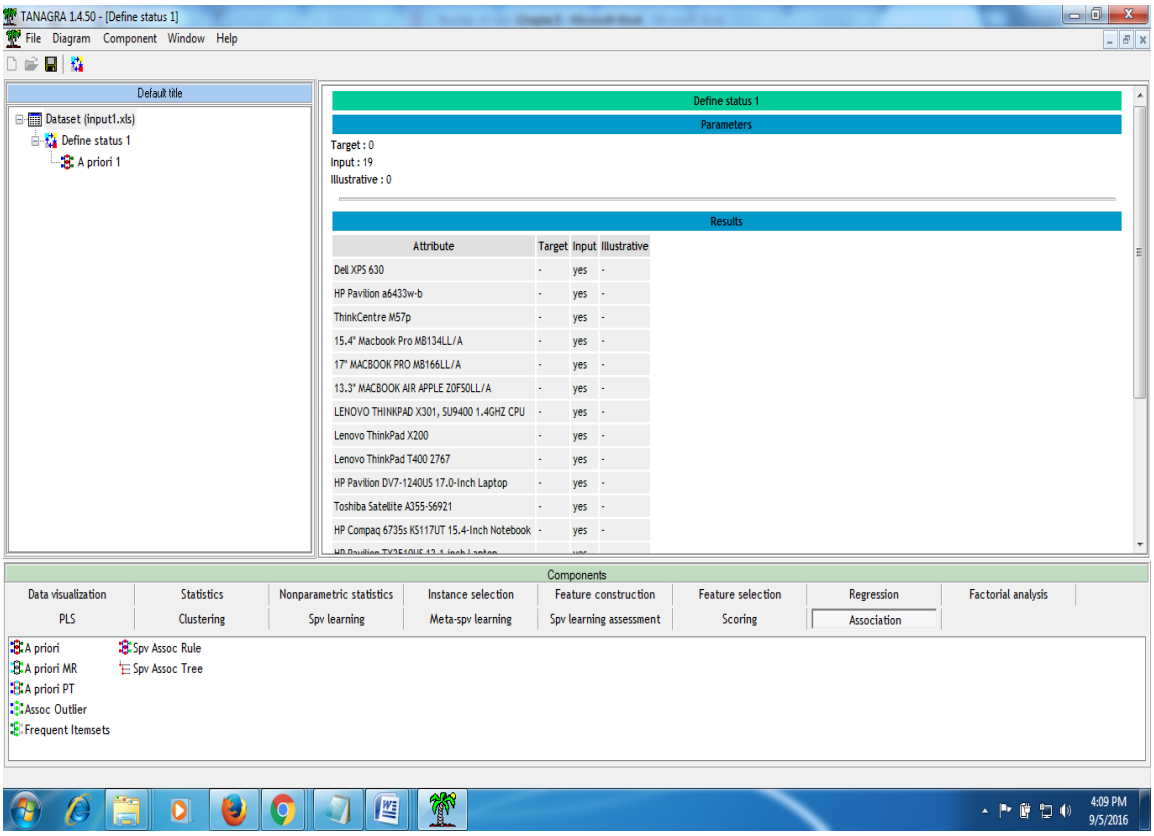

Figure 5.2 Tanagra explorer for input visualization

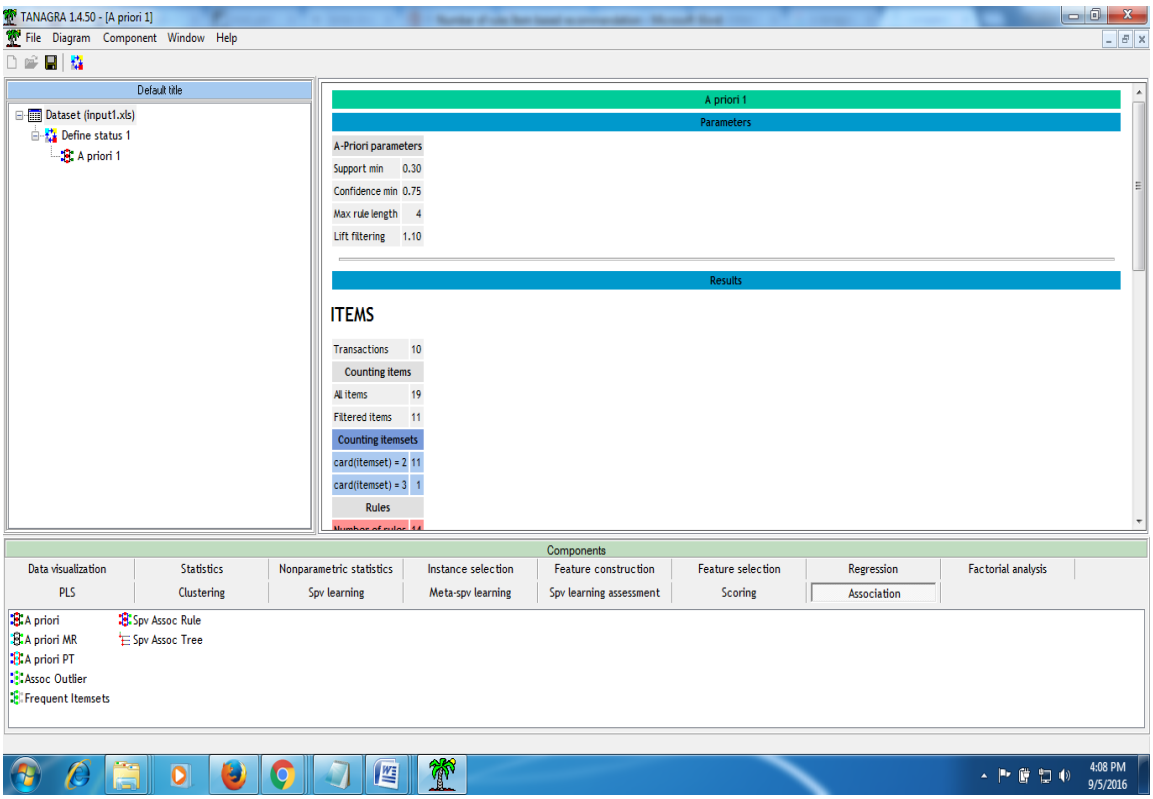

Figure 5.3 Tanagra explorer for result visualization

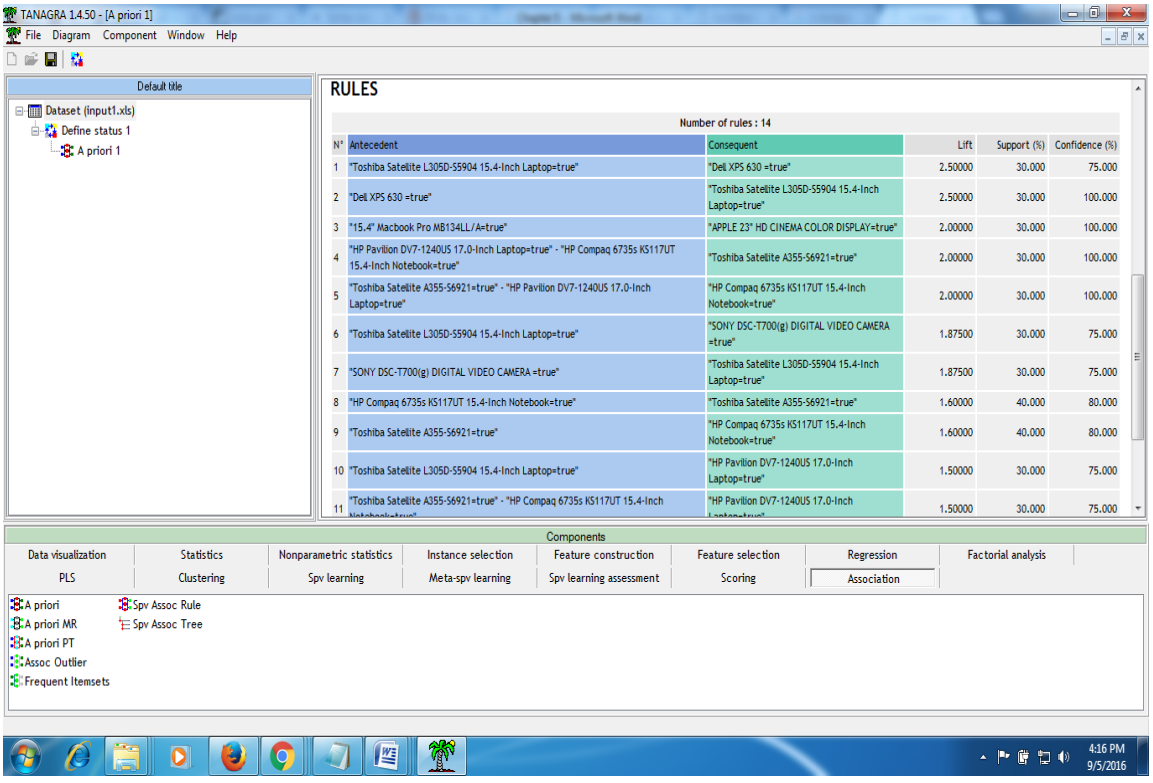

Figure 5.4 Tanagra explorer for rules visualization

5.3 Time Comparison of Real-time Data Pre-processing

The experiment is made with the web server logs to validate the effectiveness and efficiency of our above mentioned methodology. The performance of the real-time data pre-processing is evaluated by using following formula. Table 5.1 shows time comparison of real-time data cleaning process.

% Real-Time required for

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.

Log File Records (access.log + access_redundant.log)	Proposed real-Time Data Cleaning (TDC) (milli-secs)	Proposed Real-time User-Session and Transaction Identification (TUT) (milli-secs)	Proposed Real-Time User BasedRecommendation Generation (TRG) (milli-secs)	% Time for Data Pre-processing
808	123	155	223	55.48
1643	240	186	253	62.73
2541	285	234	304	63.06
3569	334	271	343	63.82
4597	384	292	361	65.18

Table 5.1 Time Comparison of Real-time Data Pre-processing

CHAPTER -6

OBSERVATION AND RESULTS

Chapter 6

OBSERVATION AND RESULTS

This chapter is focused on results of real-time data cleaning and data pre-processing step. The graphical result shows the performance and effectiveness of proposed methodology. A comparative study of the present work with past studies has been made.

6.1 Real-Time Data Cleaning Process

To validate the efficiency of proposed methodology, an experiment is conducted using the log files of TomatoCart application. The different log files collected from Apache Web server were analyzed. The results showed that our methodology reduced the Web access log file down to 60% of the initial size.The Table 6.1 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 6.1 Comparison Matrix of Real-Time Data Cleaning

6.1.1 Comparison of log entries of access.log and access_redudant.log

The access.log and access_redundant.log files were used for this comparison are generated using TomatoCart e-commerce shopping website. Different users are created to login and use the application to purchase sample products from the website. As per the observation the number of log entries in the access.log file has been reduced to 60% of the total number of log entries. There is a significant reduction in the number of log entries in the access.log file and hence this will further reduce the processing time of the log file for Web Usage mining.Figure 6.4 shows Comparison of log entries of access.log and access_redudant.log.

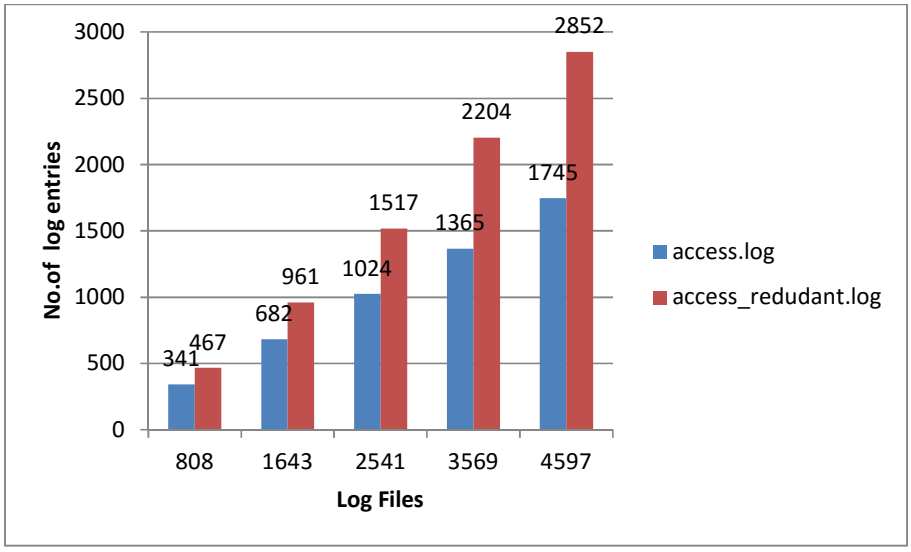

Figure 6.1 Comparison of log entries of access.log and access_redudant.log

6.1.2 Comparison of Multimedia, JavaScript, CSS and error Log entries in access_redundant.log

The figure 6.5 shows the comparison of the various multimedia, javascript, css and error log entries namely, Multimedia Request -86%, CSS Request – 1.26%, JavaScript Request –8.96% and Error Request-1.53%. The Multimedia Request include .png, .jpg, .gif, .ico

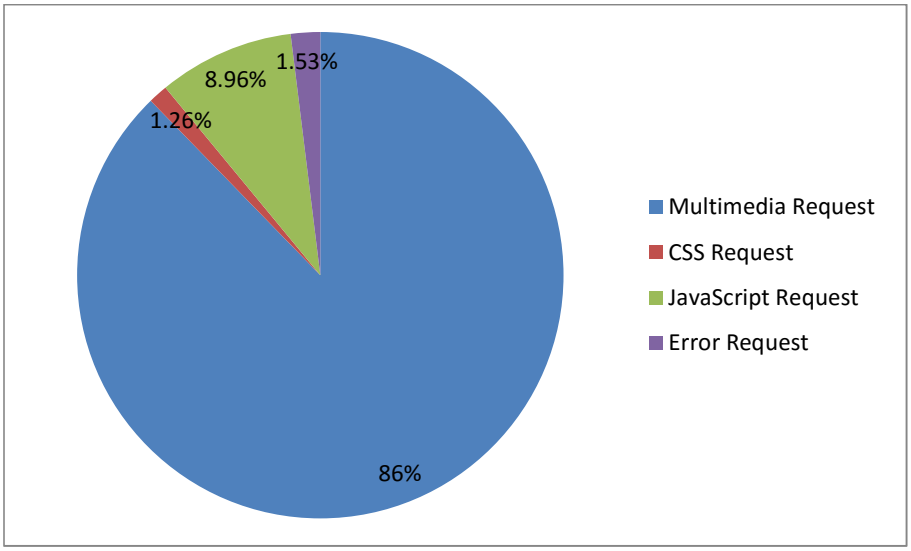

Figure 6.2 Comparison of Multimedia, JavaScript, CSS and Error Log entries

6.1.3 Comparison of Log files size

The figure 6.6 shows the size of the access.log and access_redundant.log files. As per our observation the real time data cleaning solution has made a significant reduction in the size of the access log file.

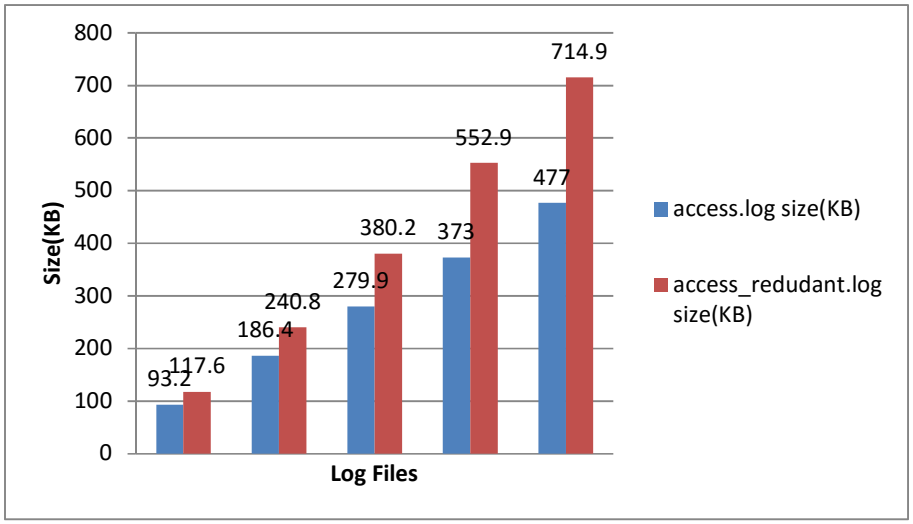

Figure 6.3 Comparison of Log file size

6.2 Time Comparison between Real-Time Data Cleaning 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.

Traditional data cleaning approach is applied on the access_full.log and the execution times are calculated for all the log file samples. The Table 6.1 below shows the time stats of the real-time data cleaning versus the traditional data cleaning approach. 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 6.1 and 6.2 shows Real-time and Traditional data cleaning time. Figure 6.3 shows time comparison of traditional and proposed data cleaning process.

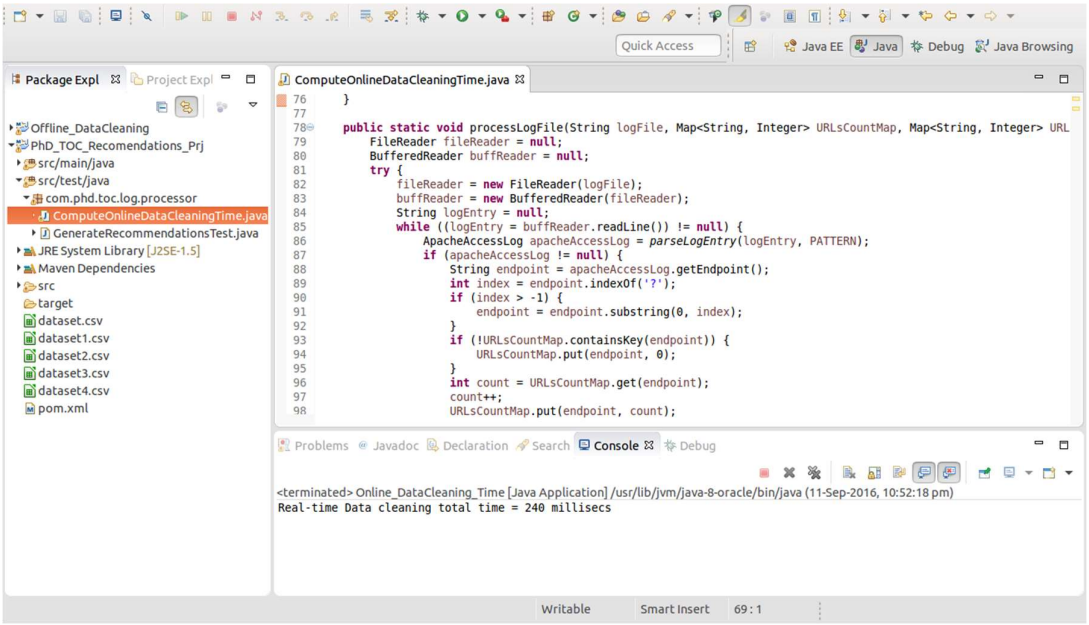

Figure 6.4 Real-Time Data Cleaning Time

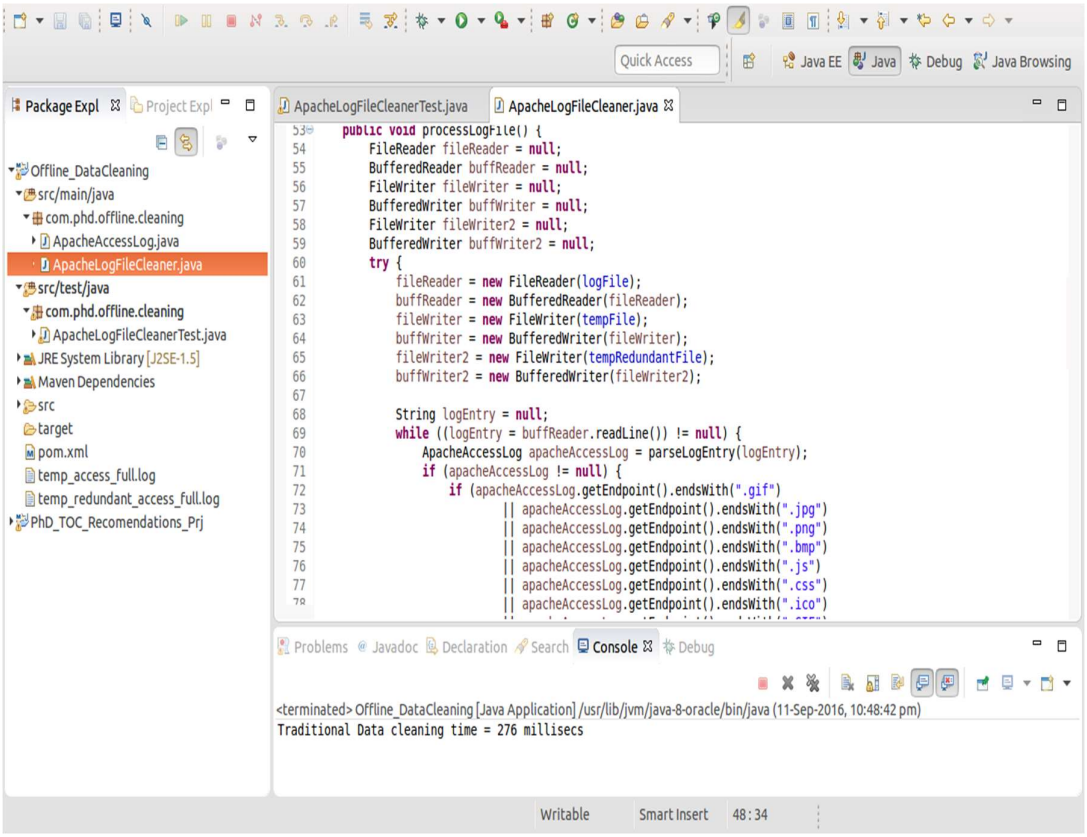

Figure 6.5 Traditional Data Cleaning Time

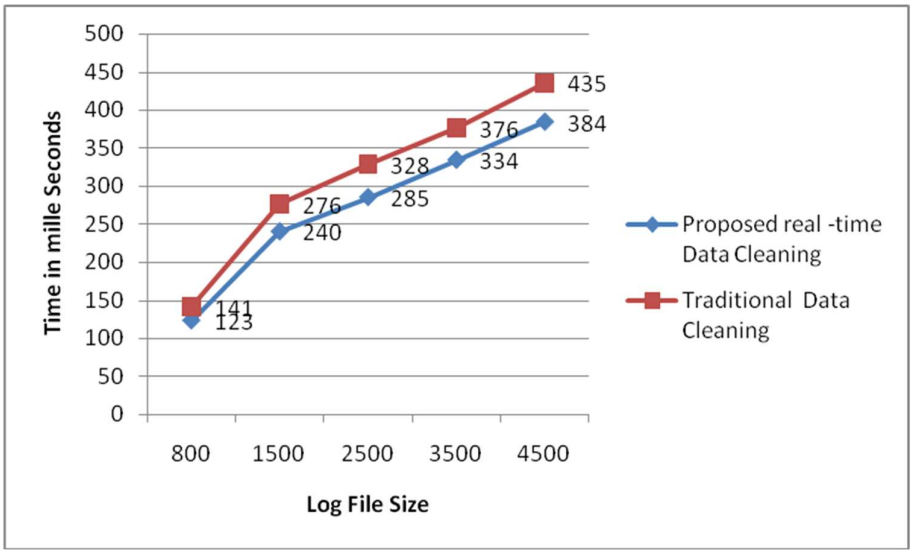

Figure 6.6 Time Comparison of Traditional and Proposed Data Cleaning Process

6.3 Real-Time DataPre-processing

The proposed algorithm for real time data pre-processing in web usage mining is divided into three main steps: User and Session Identification, Transaction Identification, and Recommendation Generation.

6.3.1 User-Session Identification Process

In the proposed system UserId and sessionId are recorded in the log entries of the web requests of the user. Thus one can identify the user and session by processing the log file. The proposed system reads the sessionId from the HTTP request header/cookies .The userId is then read from the application user tables using the sessionId as shown in the table 6.2.

Client IP	URL	User Id	Session Id
127.0.0.1	GET /tomatocart/account.php?login	1	Cei5fbei24ib2uk0e4ejnka3p6
127.0.0.1	POST /tomatocart/json.php	1	Cei5fbei24ib2uk0e4ejnka3p6
127.0.0.1	GET /tomatocart/checkout.php?success	1	Cei5fbei24ib2uk0e4ejnka3p6
127.0.0.1	GET /tomatocart/account.php?logoff	1	Cei5fbei24ib2uk0e4ejnka3p6
127.0.0.1	GET /tomatocart/account.php?login	4	4uctorihvhtakg379g7kdam5s5
127.0.0.1	POST /tomatocart/json.php	4	4uctorihvhtakg379g7kdam5s5
127.0.0.1	POST /tomatocart/json.php	4	4uctorihvhtakg379g7kdam5s5
127.0.0.1	GET /tomatocart/account.php?logoff	4	4uctorihvhtakg379g7kdam5s5

Table 6.2 User Session Identification

6.3.2 Transaction Identification Process

The access log entries are enriched with the semantic information of the user actions in the application. In the case of the sample application 'TomatoCart', the access.log adds the following semantic information in the logs –

a) User adds a product (Product Id - 17) to shopping cart – The Semantic info 'action=add_product,pID=17' is added to the log entry of add product web request.

b) User removes a product (Product Id – 6) from shopping cart – The Semantic info 'action=remove_product,pID=6' or 'action=cart_remove,pID=6' is added to the log entry of the remove product web request.

c) User checksout and confirms the order – The semantic info 'action=success' is added

to the log entry of the confirm order web request.

Figure 6.7 below shows the access.log contains the log entries of the web requests that were part of the customers shopping transaction on TomatoCart.

Figure 6.7 Transaction Identification Process

Figure 6.8 below shows the dataset CSV file generated based on the user order transactions. It contains comma-separated record for each product purchased by a user. The first value in the record is the userId and second value is productId.

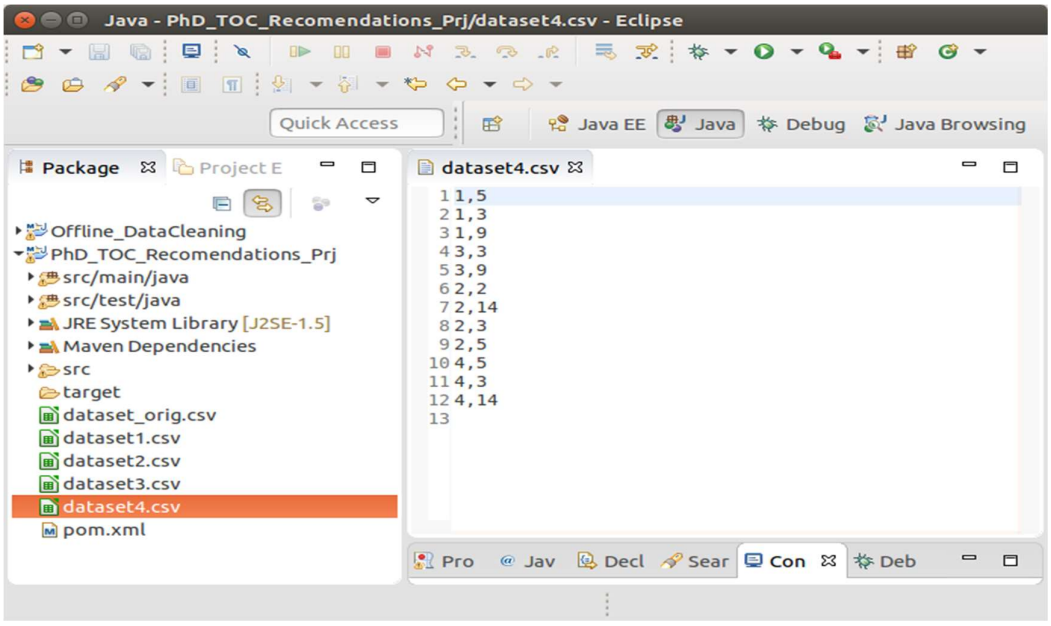

Figure 6.8 dataset.csv

6.3.3 Recommendations Generation

6.3.3.1 Real-Time User Based Recommendation

The recommendation task of Java application produced the recommendations for the dataset CSV file that has UserId and productId. Figure- 6.9 shows real-time user based recommendation.

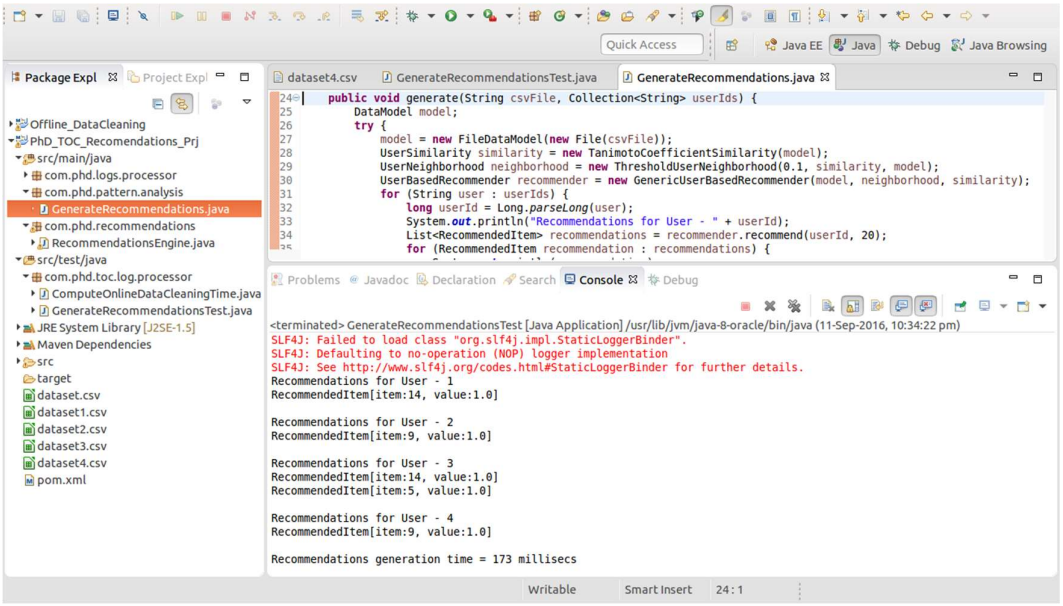

Figure 6.9 Real-time User Based Recommendation Generation

The Tables 6.3 shows the products purchased by each of the users and the recommendations that were generated based on the user similarity.

User		**Products Purchased**		**Recommended Products**	
User login	Login Id	Product Id	Product Name	Product Id	Product Name
tom@example.com	1	5	17" MACBOOK PRO MB166LL/A	14	Toshiba Satellite L305D-S5904 15.4 –Inch Laptop
		3	ThinkCentre M57p		
		9	Lenovo ThinkPad T400 2767		
	2	2	HP Pavilion a6433w-b	9	Lenovo ThinkPad T400 2767
		14	Toshiba Satellite L305D-S5904 15.4 –Inch Laptop		

John.doe@example.com		3	ThinkCentre M57p		
		5	17" MACBOOK PRO MB166LL/A		
lucy@example.com	3	3	ThinkCentre M57p	5	17" MACBOOK PRO MB166LL/A
		9	Lenovo ThinkPad T400 2767	14	Toshiba Satellite L305D-S5904 15.4 –Inch Laptop
viraj@example.com	4	3	ThinkCentre M57p	9	Lenovo ThinkPad T400 2767
		5	17" MACBOOK PRO MB166LL/A		
		14	Toshiba Satellite L305D-S5904 15.4 –Inch Laptop		

Table 6.3 TomatoCart –Real-Time User Based Recommendations

6.3.3.2 Item Based Recommendation

The item based recommendations are generated using Apriori algorithm of TANAGRA data mining tool.Association rules show relationship among different items. The table 6.4 shows association rules are found using support 30%, confidence 75% and lift above 1.0.

Association rule generation comprises of three steps:

1. To find all frequent itemsets in a database, apply minimum support.
2. The association rules are form using frequent itemsets and minimum confidence constraint.
3. Antecedent and consequent appears more often together when lift value greater than 1; this states that the Antecedent is positively correlated with Consequent.

Antecedent	Consequent	Lift	Support (%)	Confidence (%)
"Toshiba Satellite L305D-S5904 15.4-Inch Laptop=true"	"Dell XPS 630 =true"	2.50000	30.000	75.000
"Dell XPS 630 =true"	"Toshiba Satellite L305D-S5904 15.4-Inch Laptop=true"	2.50000	30.000	100.000
"15.4 Macbook Pro MB134LL/A=true"	"APPLE 23" HD CINEMA COLOR DISPLAY=true"	2.00000	30.000	100.000
"HP Pavilion DV7-1240US 17.0-Inch Laptop=true" - "HP Compaq 6735s KS117UT 15.4-Inch Notebook=true"	"Toshiba Satellite A355-S6921=true"	2.00000	30.000	100.000
"Toshiba Satellite A355-S6921=true" - "HP Pavilion DV7-1240US 17.0-Inch Laptop=true"	"HP Compaq 6735s KS117UT 15.4-Inch Notebook=true"	2.00000	30.000	100.000
"Toshiba Satellite L305D-S5904 15.4-Inch Laptop=true"	"SONY DSC-T700(g) DIGITAL VIDEO CAMERA =true"	1.87500	30.000	75.000
"SONY DSC-T700(g) DIGITAL VIDEO CAMERA =true"	"Toshiba Satellite L305D-S5904 15.4-Inch Laptop=true"	1.87500	30.000	75.000
"HP Compaq 6735s KS117UT 15.4-Inch Notebook=true"	"Toshiba Satellite A355-S6921=true"	1.60000	40.000	80.000
"Toshiba Satellite A355-S6921=true"	"HP Compaq 6735s KS117UT 15.4-Inch Notebook=true"	1.60000	40.000	80.000
"Toshiba Satellite L305D-S5904 15.4-Inch Laptop=true"	"HP Pavilion DV7-1240US 17.0-Inch Laptop=true"	1.50000	30.000	75.000
"Toshiba Satellite A355-S6921=true" - "HP Compaq 6735s KS117UT 15.4-Inch Notebook=true"	"HP Pavilion DV7-1240US 17.0-Inch Laptop=true"	1.50000	30.000	75.000
"SONY DSC-T700(g) DIGITAL VIDEO CAMERA =true"	"HP Pavilion DV7-1240US 17.0-Inch Laptop=true"	1.50000	30.000	75.000
"SONY DSC-T700(g) DIGITAL VIDEO CAMERA =true"	"APPLE 23" HD CINEMA COLOR DISPLAY=true"	1.50000	30.000	75.000
"Lenovo ThinkPad X200=true"	"APPLE 23" HD CINEMA COLOR DISPLAY=true"	1.50000	30.000	75.000

Table 6.4 Item Based Recommendation

6.3.4 Time Comparison 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 formula mentioned in chapter-5(Section 5.3) .The average percentage of time for real-time data pre-processing is 62% of the total web usage mining time.

Figure 6.10 shows real-time data pre-processing time. Figure 6.11 shows Time Comparison of Real-time Data Pre-processing.

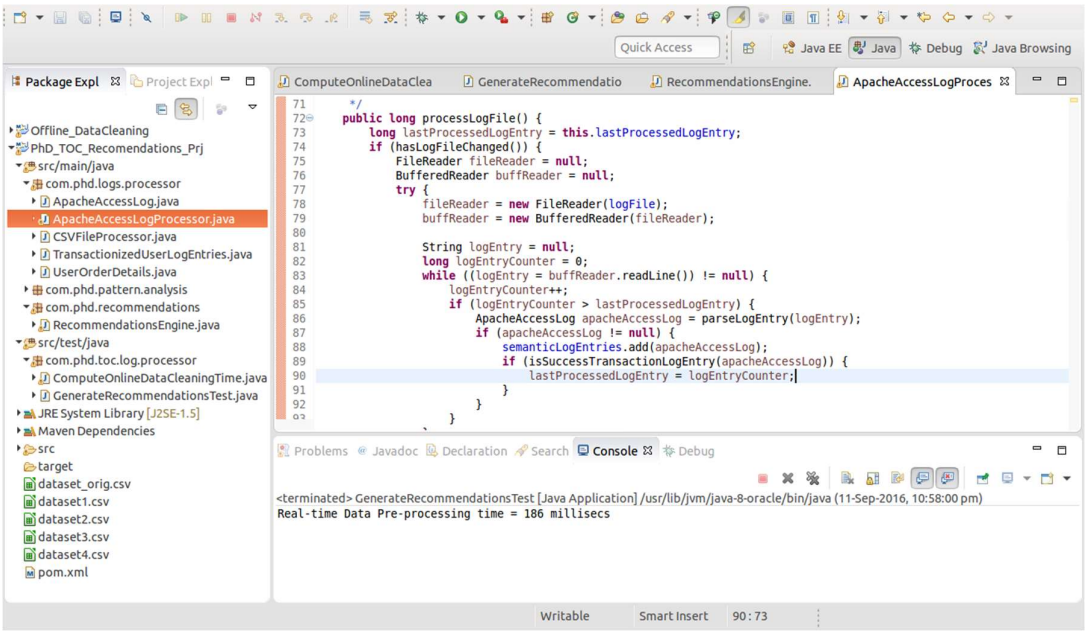

Figure 6.10 Real-Time Data Pre-processing time

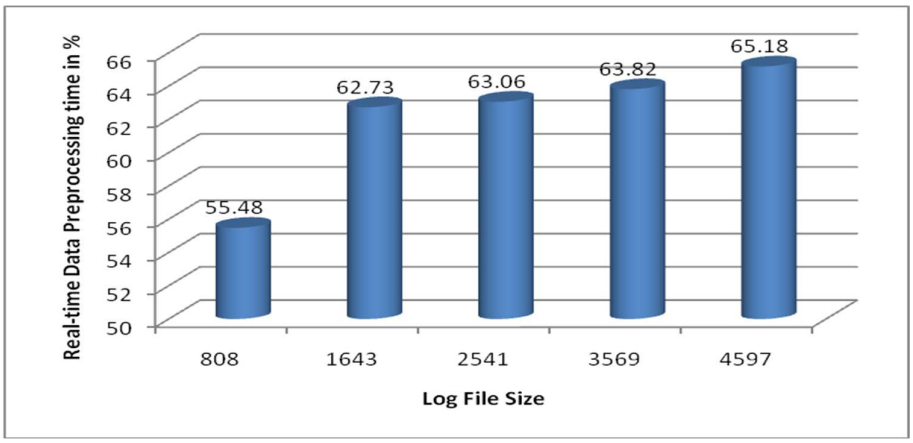

Figure 6.11 Real-time Data Pre-processing time in %

6.4 Comparative Study

A number of studies were reported on data pre-processing techniques, but it seems difficult to compare the results as the web log data used by the authors is usually site-specific. The work reported in the literature is presented in the Tables 6.5 and 6.6.The proposed work shows promising results on data cleaning and data pre-processing.

Author Name	Web Log Customization	Online/Offline	% of Reduction after data Cleaning
HongzhouSha,,Qingyun Liu(2013)	NA	Online	30%
HananeEzzikouri , Mohammed Erritali, Mohamed Oukessou(2015)	NA	Offline	70%
Priyanka VermaDr.NishthaKesswani(2014)	NA	Offline	12 %
Muskan, Dr.Kanwal Garg(2016)	NA	Offline	71%
K. Sudheer Reddy, G. ParthaSaradhi Varma, and M. Kantha Reddy (2014)	NA	Offline	50-55%.
V.VidyaPriya,S. Kalaivani(2015)	NA	Offline	75%
ShailyLanghnoja , MehulBarot, Darshak Mehta	NA	Offline	67%.
Proposed Approach	**Yes**	**Online**	**After Data Cleaning of Web Log File records reduced up to 60%**

Table 6.5 Analysis of Data Cleaning

***NA=Not Available**

Author Name	% of Time Required for Data Pre-processing
Khushbu Patel, Anurag Punde, Kavita Namdev, Rudra Gupta, Mohit Vyas(2015)	70%
Vijayashri Losarwar, Dr.Madhuri Joshi (July 2015-2016)	80%
Mitali Srivastava,RakhiGarg,P. K. Mishra (July 2014)	80%

S. Prince Mary ,E. Baburaj (2013)	80%
Priyanka Verma,Dr.Nishtha Kesswani(2014)	80%
V.Vidya Priya,S. Kalaivani(2015)	80%
C. E. Dinucă (2012)	60 To 90%
Proposed Approach	**62%**

Table 6.6 Analysis of Data Pre-processing

CHAPTER -7

SUMMARY AND CONCLUSION

Chapter 7

SUMMARY AND CONCLUSION

7.1 Conclusion

7.2 Scope of Future Research

This chapter presents contributions and summary of the present research work. Major contribution of the present work and scope for further research are discussed.

7.1 Conclusion

The present work has proposed algorithms for real time data cleaning and data pre-processing. The algorithms are tested with log files of TomatoCart application. In addition, a new structure of web log file has been proposed 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. The average percentage of time required for data pre-processing is 62%.

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 behaviour 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. Item based recommendations are generated offline by using Apriori algorithm.

Association rules are evaluated on the metric of support, confidence and lift.

Major Contributions of the Present Work:

The contributions of the present work summarized as follows:

1. Perform real-time data cleaning process on the Web Server log files to significantly reduce the size of the log files by around 60%.
2. Enrich semantic information of the web requests in the log files thus improve the quality of data for data pre-processing.
3. Proposed and developed algorithm for Data cleaning and Data Pre-processing.
4. Modified the structure of the existing Combined Log file format.
5. The proposed real time data cleaning process required 12% less time than traditional data cleaning process.
6. The average percentage of time required for real time data pre-processing is 62%.
7. Real-time User Based recommendations are generated using Apache Mahout and Item Based recommendation are generated offline using Apriori algorithm.

7.2 Scope for Future Research

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.

BIBLIOGRAPHY

1. Abraham M. and Puthiyidam. J., (2016), "A Survey on Wind Data Pre-Processing in Electricity Generation", International Journal on Cybernetics & Informatics", International Journal on Cybernetics & Informatics, Vol- 5, Issue-2, pp. 407- 415.
2. Aggarwal M. and Bhatia A., (2015), "Pattern Discovery Techniques in Online Data Mining", International Journal of Engineering and Technical Research (IJETR), ISSN: 2321-0869, Vol- 3, Issue -7, pp. 28-31.
3. Aldekhail M. (2016), "Application and Significance of Web Usage Mining in the 21st Century: A Literature Review", International Journal of Computer Theory and Engineering, ISSN: 1793-8201, Vol-8, Issue-1, pp: 41-47.
4. Ashish Vitthalrao Galphade and Dhiraj Bhise (2016), "Suggestion of An Apriory Algorithm For Web Recommendation System", Imperial Journal of Interdisciplinary Research (IJIR), ISSN: 2454-1362, Vol-2, Issue-8, pp:997-1001.
5. Ashishkumar B. Patel, Niravkumar B. Suthar and Jitendra S. Dhobi(2012), "Recommending Top-n Research Papers (Based on With, Without and Boolean Items Preferences: An User Base Collaborative Filtering Approach in Mahout)", International Conference on Computing, Communication and Information Technology, ISBN 978-93-82338-02-4, pp:138-141.
6. Bayir M. A., Toroslu I. H., Cosar A. and Fidan G. (2008) "Discovering More Accurate Frequent Web Usage Patterns".
7. B. Uma Maheswari, Dr. P.Sumathi (2015), "A Comparative Study of Rule Mining Based Web Usage Mining Algorithms", International Journal of Science and Research (IJSR), ISSN: 2319-7064, Vol-4, Issue-11, pp:2540-2543.
8. Bhaiyalal Birla, Sachin Patel (2014**),** "An Implementation on Web Log Mining", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Vol-4, Issue2, pp:68-73.
9. B.Santhosh Kumar, K.V.Rukmani (2010), "Implementation of Web Usage Mining Using

   Apriori and FP Growth Algorithms", International Journal of Advanced Networking and Applications, ISSN: , Vol-01,Issue-06,pp:400-404.
10. Campagni. R, et al (n.d), "Sequential Pattern Analysis in a Student Database", Department Of System And Information, pp. 1-16.

11. C. E. Dinucă (2012), "An Application for Clickstream Analysis", International Journal of Computer and Communication, ISSN: , Vol-6,Issue-1,pp:68-75.
12. Chitraa.V and Thanamani.A.S., (2011), "A Novel Technique for Sessions Identification in Web Usage Mining Pre-processing", International Journal of Computer Applications ISSN:0975 – 8887, Vol-34,Issue-9, pp: 23-27.
13. Chitraa, V., Davamani, A. (2010). "A Survey on Pre-Processing Methods for Web Usage Data", International Journal of Computer Science and Information Security, ISSN: 1947-5500, Vol-7, Issue-3, pp: 78-83.
14. Chen J. and Liu W. (2006), "Research for Web Usage Mining Model", International Conference on Computational Intelligence for Modeling Control and Automation IEEE, ISBN: 0-7695-2731-0.
15. Ciobanu. C. and Dinuca. E., (n.d), "A new method for session identification in click stream analysis", Recent Researches in Tourism and Economic Development, ISBN: 978-1-61804-043-5, pp. 476-479.
16. Cleve. J., et al (n.d)," Data Mining on Transaction Data", Wismar University, Germany, pp. 1-8.
17. Charjan. D. and Pund. M., (2013), "Pattern Discovery For Text Mining Using Pattern Taxonomy", International Journal Of Engineering Trends And Technology, ISSN 2231-2803, Vol-4, Issue-10, pp. 4550-4555.
18. Chandrama W., Devale P. R., & Murumkar R. (2014), "Survey on Data Pre-processing Method of Web Usage Mining", International Journal of Computer Science and Information Technologies, ISSN-0975-9646,Vol-5,Issue- 3 ,pp 3521-3524.
19. Cheng H., et al (n.d), "Discriminative Frequent Pattern Analysis Effective Classification", US national science foundation, pp. 1-10.
20. Dangi A. K. and Sangwan S. (2013), "A New Approach For User Identification In Web Usage Mining Pre-processing", IOSR Journal of Computer Engineering, ISSN: 2278-0661, Vol-11, Issue-3, pp: 57-61.
21. Deshpande K. V., Bagwan A. B. and Deshmukh P. K. (2014), "An Effective Approach for Pattern Discovery in Web Usage Mining", International Journal of

Advance Research in Computer Science and Management Studies,ISSN: 2327782, Vol-2, Issue-12, pp:3664-3667.

22. Dinuca. E. and Ciobanu. D., (2011), "Improving the Session Identification Using the Mean Time", International Journal of Mathematical Models and Methods in Applied Sciences, Vol- 6, Issue-2, pp: 265-272.
23. Dhawan S. and Goel S. (2013), "Web Usage Mining: Finding Usage Patterns from Web Logs", American International Journal of Research in Science, Technology, Engineering and Mathematics, ISSN (Online): 2328-3580, Vol-2, Issue-2, pp: 203-207.
24. Domenech J. M. and Lorenzo J. (2007), "A Tool for Web Usage Mining", 8th International Conference on Intelligent Data Engineering and Automated Learning, pp:1-10.
25. Eirinaki M. and Vazirgiannis M. (2003), "Web Mining for Web Personalization", ACM Transactions on Internet Technology, Vol- 3, Issue-1, pp: 1-27.
26. Grace et al., (2011), "Analysis of Web Logs and Web User in Web Mining", International Journal of Network Security & Its Applications (IJNSA), DOI: 10.5121/ijnsa.2011.3107, Vol-3, Issue-1, pp: 99-110.
27. Herring.S.C, J. Hunsinger, M. Allen, & L. Klastrup (Eds.) (2009), "Web Content Analysis: Expanding the Paradigm", The International Handbook of Internet Research. Springer Verlag, pp: 1-18.
28. Halfaker A., et al (2015), "User Identification Based on Strong Regularities in Inter Activity Time", International World Wide Web Conference Committee, ACM 978-1-4503-3469-3/15/05, pp: 410-418.
29. Hanane Ezzikouri, Mohammed Erritali, Mohamed Oukessou (2015), "Pretreatment of Web Log Files", Journal of Information Sciences and Computing Technologies, ,ISSN:2394-9066, Vol-2, Issue-1, pp:108-121
30. Hellerstein M. (2008), "Quantitative Data Cleaning for Large Databases", United Nations Economic Commission for Europe (UNECE), pp. 1-41.
31. Hongzhou Sha,Qingyun Liu(2013), "EPLogCleaner: Improving Data Quality of Enterprise Proxy Logs for Efficient Web Usage Mining", Information Technology

and Quantitative Management, Elsevier Itqm Procedia Computer Science,ISSN:812 – 818 ,pp:812-818

32. Jiadi. Z. and Hai G., (2016), "Research on User Identification Algorithm Based on Rewriting URL", International Journal of Security and Its Applications, Vol-10, Issue-3, pp. 215 -222.
33. Joslyn C., et al (n.d), "Transaction Hyper Graph Models for Pattern Identification in Bitcoin Blockchain", WA 98109, pp: 1-4.
34. Kamat M. S., Bakal J. W. and Nashipudi M. (2013), "Improved Data Preparation Technique in Web Usage Mining", International Journal of Computer Networks and Communication Security, ISSN 2308-9830), Vol-1, Issue-7, pp: 284-291.
35. Kayuturk. M, et al (2005), "Compression Clustering and Pattern Discovery in Very High Dimensional Discrete Attribute Datasets", IEEE Transaction of Knowledge and Data Engineering, ISSN 1041-4347, Vol-17, Issue-4, pp: 447-461.
36. Kherwa P. and Nigam J. (2015), "Data Preprocessing: A Milestone of Web Usage Mining", International Journal of Engineering Science and Innovative Technology, ISSN: 2319-5967, Vol-4, Issue-2
37. Khushbu Patel, Anurag Punde, Kavita Namdev, Rudra Gupta, Mohit Vyas (2015), "Detailed Study of Web Mining Approaches-A Survey", International Journal of Engineering Sciences & Research Technology", ISSN: 2277-9655, Vol-4, Issue-2, pp:23-30.
38. Kularbphettong K., et al (2010), "A Hybrid System Based on Multi Agent System in the Data Pre-processing Stage", (IJCSIS) International Journal of Computer Science and Information Security, ISSN: 1947-5500, Vol-7, Issue-2, pp: 199-203.
39. Krishnan. S, et al (2016), "Towards Reliable Interactive Data Cleaning: a User Survey and Recommendations", ACM ISBN: 978-1-4503-4207-0, pp: 1-5.
40. K. Sudheer Reddy, G. Partha Saradhi Varma, and M. Kantha Reddy (2014), "An Effective Preprocessing Method for Web Usage Mining" International Journal of Computer Theory and Engineering, DOI: 10.7763/IJCTE.2014.V6.900, Vol-6, Issue-5, pp:412-415.

41. Kulkarni J. and Bakal A., (2014), "Survey on Data Cleaning", International Journal of Engineering Science and Innovative Technology, ISSN: 2319-5967, Vol-3, Issue-4, pp: 615-620.
42. Losarwar. V. and Joshi M., (2012), "Data pre-processing in Web Usage Mining", International Conference on Artificial Intelligence and Embedded Systems, pp: 1-5.
43. Lokeshkumar, R., Sindhuja, R., & Sengottuvelan D. P. (2014). "A Survey on Pre-Processing of Web Log File in Web Usage Mining to Improve the Quality of Data", International Journal of Emerging Technology and Advanced Engineering, ISSN, 2250-2459, Vol-4, Issue-8, pp:229-234.
44. Lee. E, et al (2015), "Hierarchical Cluster Analysis Heatmaps and Pattern Analysis: An Approach for Visualizing Learning Management System Interaction Data", 9th International Conference on Educational Data Mining, pp: 603-604.
45. Makwana. C and Rathod. K(2014), "An Efficient Technique for Web Log Pre-processing Using Microsoft Excel", International Journal of Computer Applications, ISSN 0975-8887, Vol-90, Issue-12, pp:25-28.
46. Mahanta A. N. (2008), "Web Mining: Application of Data Mining", Proceedings of NCKM, pp: 1757–1764.
47. Mitharam M. D. (2012), "Preprocessing in Web Usage Mining", International Journal of Scientific and Engineering Research, ISSN 2229-5518, Vol-3, Issue-2, pp: 1-7.
48. Mitali Srivastava, Rakhi Garg, P. K. Mishra (July 2014), "Preprocessing Techniques in Web Usage Mining: A Survey", International Journal of Computer Applications ,ISSN:0975 -8887, Vol-97, Issue-18, pp:1-9.
49. Muskan, Dr. Kanwal Garg (2016), "An Efficient Algorithm for Data Cleaning of Web Logs with Spider Navigation Removal", International Journal of Computer Application, ISSN-2250-1797, Vol-6, Issue-3, pp:6-12.
50. Nigrel S. et al (2015), "Web Log Pre-processing for Web Usage Mining", International Journal for Scientific Research and Development, ISSN: 2321-0613, Vol-2, Issue-12, pp:604-606.

51. Patel K. et al (2015), "Detailed Study of Web Mining Approaches-A Survey", International Journal of Engineering, Sciences and Research Technology, ISSN: 2277-9655, Vol-4, Issue-2, pp: 23-30.
52. Patel P. and Parmar M., (2014), "A Review on User Session Identification Through Web Server Log", (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN:0975-9646 , Vol-5, Issue-1, pp: 146-148.
53. Patel P. and Parmar M., (2014), "Improve Heuristics for User Session Identification Through Web Server Log in Web Usage Mining", (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN:0975-9646, Vol-5, Issue-3, pp: 3562-3565.
54. Padala. V, et al (2013), "A Novel Method for Data Cleaning and User Session Identification for Web Mining", International Journal of Modern Engineering Research, ISSN:2249-6645, Vol-3, Issue-5, pp:2816-2819.
55. Priyanga P. and Naveen N C (2015), "User Identification, Classification and Recommendation in Web Usage Mining- An Approach for Personalized Web Mining", International Journal of Innovative Science, Engineering and Technology, ISSN 2348 – 7968, Vol-2, Issue-4, pp: 1021-1030.
56. Rao.V.V.R. and Kumari.V.V. (2011), "An Enhanced Pre-Processing Research Framework For Web Log Data Using A Learning Algorithm", Computer Science & Information Technology, DOI: 10.5121/csit.2011.1101, pp. 01–15.
57. Raiyani.S.A. and Jain.S (2012), "Efficient Preprocessing Technique Using Web Log Mining", International Journal of Advancements in Research & Technology, ISSN 2278-7763, Vol-1, Issue-6, pp: 1-5.
58. Raiyani.S.A, Jain.S and Raiyani.A.G., (2012), "Advanced Preprocessing Using Distinct User Identification in Web Log Usage Data", International Journal of Advanced Research in Computer and Communication Engineering , ISSN : 2278 – 1021, Vol-1, Issue-6, pp: 418-422 .
59. Raiyani.A.G. and Pandya.S.S., (2012), "Discovering User Identification Mining Technique For Preprocessed Web Log Data", Journal Of Information, Knowledge And Research In Computer Engineering, ISSN: 0975 – 6760, Vol-2, Issue-2, pp: 477-482.

60. Rahm E. And Do. H, (n.d), "Data cleaning: Problems and Current Approaches", Microsoft research, pp. 1-11.
61. Rathi A. and Raipurkar A. (2016), "Web Usage Mining- A Review", International Journal of Advanced Research in Computer and Communication Engineering,ISSN (Online) 2278-1021, Vol-5, Issue-2, pp: 496-498.
62. Rejena. S. And Malika R., (2016), "Innovative Pre-Processing Technique and Efficient Unique User Identification Algorithm for Web Usage Mining", International Journal of Advanced Research Computer Science and Software Engineering, ISSN: 2277 128X, Vol-6, Issue-2, pp. 85-91.
63. Sagar P. and Nimavat A. V. (2015), "Web Usage Mining: Survey on Process and Methods", International Multidisciplinary Research Journal, 2 (5), ISSN: 2349-7637 (Online), Vol-2, Issue-5, pp: 1-4.
64. Singh S. and Badhe V. (2014), "An Exclusive Survey on Web Usage Mining for User Identification", International Journal of Innovative Research in Computer and Communication Engineering, ISSN (Online): 2320-9801, Vol-2, Issue-11, pp: 6582-6589.
65. Saxena K. and Shukla R., (2010), "Significant Interval and Frequent Pattern Discovery in Web of Data", IJCSI International Journal of Computer Science Issues, ISSN (Online): 1694-0784, Vol-7, Issue-1, pp: 29-36.
66. Shaily Langhnoja, Mehul Barot, Darshak Mehta (2012), "Pre-Processing: Procedure on Web Log File for Web Usage Mining", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Vol-2, Issue-12, pp:419-423.
67. Singh. S And Badhe. V, (2014), "An Exclusive Survey on Web Usage Mining for User Identification", International Journal of Innovative Research in Computer and Communication Engineering, ISSN (Online): 2320-9801, Vol-2, Issue-11, pp. 6582-6589.
68. Srivastave J., Cooley R., Deshpande M. and Tan P. N. (2000), "Web Usage Mining:Discovery and Applications of Usage Patterns from Web Data,", SIGKDD Explorations. ACM SIGKDD, Vol-1, Issue-2, pp:12-23.

69. Sundari R., et al (2014), "A Review Pattern Discovery Techniques of Web Usage Mining", International Journal of Engineering Research and Application, ISSN: 2248-9622, Vol-9, Issue-9, pp:131-136.
70. Steinbach. M., et al (2007), "Objective Measure for Association Pattern Analysis", Contemporary Mathematics, American Mathematical Society, pp. 1-21.
71. S. Prince Mary, E. Baburaj (2013), "An Efficient Approach To Perform Pre-Processing"
    Indian Journal of Computer Science and Engineering (IJCSE), ISSN: 0976-5166, Vol-4, Issue-5, pp:404-410.
72. Tamrakar L. and Ghosh S. M. (2014), "Identification of Frequent Navigation Pattern Using Web Usage Mining", International Journal of Advanced Research in Computer Science & Technology, ISSN: 2347 - 8446 (Online), Vol-2, Issue-2, pp: 296-299.
73. Thakar U., et al (2010), "Pattern Analysis and Signature Extraction for Intrusion Attacks on Web Services", International Journal of Network Security & Its Applications (IJNSA), DOI: 10.5121/ijnsa.2010.2313, Vol-2, Issue-3, pp.190-205.
74. Talakokkula A. (2015), "A Survey on Web Usage Mining, Applications and Tools", Computer Engineering and Intelligent Systems, ISSN 2222-2863 (Online), Vol-6, Issue-2, pp:22-29
75. Tao Y. H., Hong T. P. and Su Y. M. (2008), "Web Usage Mining with Intentional Browsing Data", Expert Systems with Applications Science Direct, DOI: 10.1016/j.eswa.2007.02.017, Vol-34, Issue-3, pp: 1893-1904.
76. Tomar D. and Agarwal S.(2014), "A Survey on Pre-processing and Post processing Techniques in Data Mining", International Journal of Data Base Theory and Application, ISSN: 2005-4270, Vol-7, Issue-4, pp. 99-188.
77. Vellingiri J. and Pandian C. (2011), "A Novel Technique for Web Log Mining with Better Data Cleaning and Transaction Identification", Journal Of Computer Science, DOI: 10.3844/jcssp.2011, Vol-7, Issue-5, pp. 683-689.

78. Verma P. and Kesswani N. (n.d), "Web Usage Mining Framework for Data Cleaning and IP address Identification", International Journal of Advanced Studies in Computer Science and Engineering, ISSN : 2278 7917, Vol- 3, Issue-8, pp:39-43.
79. Vishwakarma A. and Singh K.N., (2014), "A Survey on Web Log Mining Pattern Discovery", (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN:0975-9646, Vol-5, Issue-6, pp: 7022-7031.
80. Vijayashri Losarwar, Dr. Madhuri Joshi(July 15-16), "Data Preprocessing in Web Usage Mining", International Conference on Artificial Intelligence and Embedded Systems, Singapore.pp:1-6.
81. V.Vidya, Priya,S., Kalaivani(2015), "An Efficient Clustering Technique for Weblogs", IJISET-International Journal of Innovative Science, Engineering & Technology , ISSN:2348 -7968, Vol-2, Issue-7, pp:516-525.
82. Yang, et al (2006), "Pre-Processing Time Series Data for Classification with Application to CRM", Springer-Vering Berlin Heidelberg, pp. 133-142.
83. Zhong N., et al (2012), "Effective Pattern Discovery for Text Mining", IEEE Transactions On Knowledge and Data Engineering, Vol-24, Issue-1, pp. 30-44.

Books:

84. Bamshad Mobasher (n.d.), "Web Usage Mining", Chapter-2, pp: 449-483.
85. Ganti. V and Sarma. A, (2013), "Data Cleaning, a Practical Perspective", Morgan and Claypool Publishers, pp: 1-63.
86. Hasvighurst. R, (2007), "User identification and authentication concepts", Taylor & Francis group, LLC, pp: 1-64.
87. Maletic and Marcus (n.d), "Data Cleaning, Data Mining and Knowledge Discovery Handbook", pp: 21-36.

Thesis:

88. Koh J. (2006), "Correlation Based Methods for Biological Data Cleaning", Ph.D. Thesis, National University of Singapore, pp: 1-160.

89. Vahid Fadaeian (2015), "Developing and Evaluating Recommender Systems", Ph.D. Thesis MID Sweden University, pp: 1-53.

Websites:

90. "Apache Mahout," Available: http://mahout.apache.org/

91. "W3C Common Log Format", http://www.w3.org/Daemon/User/Config/Logging.html

92. "W3C Extended Log Format", http://www.w3.org/TR/WD-logfile.html

93. "Apache Mod Perl", https://perl.apache.org/

Publication Related to the Thesis

International Publications

"*Real-Time Data Cleaning and Semantic Enrichment of Web Server Logs*", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN (online): 2277 128X, Volume 6, Issue 7, 30th July 2016, Impact Factor – 2.5, Science Central Evaluation Score: 6.39.

"Real-Time Data Pre-processing Technique in Web Usage Mining", International Journal of Innovative Research in Computer and Communication Engineering, ISSN (Online): 2320-9801, Volume 4, Issue 9, 30th September 2016, Impact Factor-6.5.

"Effective Data Pre-processing Technique in Web Usage Mining", International Journal of Advances in Management, Technology and Engineering Sciences, ISSN: 2249-7455, Volume 2, Issue 4(III), January 2013.