International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

Data Pre-Processing on Web Server Access Logs of University for User Interaction Patterns

Chaitra H.K, Suneetha K.R

Abstract - In the current digital Era, websites are developed and organized into multifaceted in nature. It is essential to distinguish user sessions/intent and browsing behavior from logs in order to recommend appropriate content for the web designers and administrators. This paper focuses on data preprocessing of the weblogs received from Kannada University Hampi, Vidyaranya Karnataka state are cleaned viably by applying various pre-processing methodologies. The work identifies the superior quality of data to discover user interactions, user sessions, the specific web pages, and the regularly visited Uri's, most visited pages, most time spent on pages and incorrect webpages served to users. These pre-processed webserver access log files will be utilized to discover patterns, fine grained analysis and study. This paper also focuses on challenges of log file analysis.

Keywords: Data cleaning, User Analysis, Log files, Data Preprocessing.

I. INTRODUCTION

Websites are massive collection of structured and unstructured related web resources like images, text document and multimedia yet it's a difficult task from website to retrieve the relevant data [1]. Every interaction of the user with web pages is logged in a single record as a text file is known as web log file. The weblogs accommodate abundant information, which include irrelevant and relevant data too. Data Pre-processing is one of the phases of data science, which is used to get rid of the inconsistent, irrelevant and redundant data. the idea of knowledge pre-processing is to convert raw log files into the cleaned log files which will be given for further processing of pattern discovery and pattern analysis [2][3].

II. RELATED WORK

This area presents related work in this space, in the digital era data science [18] is one of the emerging area of artificial intelligence where data analyzing is the most important in the websites to track the user interest and browsing behavior of the users from the access log files [10]. Most of the modern applications in today's market are deployed on multiple production web servers in different data centers or in the cloud-based web servers that are distributed geographically. These web servers which, host various applications, record different user and application activity and save it in the form of messages in a file called log file. University website provides all the pages related to new invites for admissions, fresh announcements, lists different details like admission, career info, new happenings, university events like workshops, seminars, technical and non-technical events etc. We use all this info for mining user behaviors, analysis browsing patterns, developed over the period of time includes 404,401,400,500,502 errors occurred while browsing for necessary information by the users[8] which help system admin and web designer to improve their system. According to Srivastava, et al., [5]data preprocessing is defined as to convert the usage, content and structure information into the data abstractions necessary for pattern discovery, Ling Zheng [11] has used an improved preprocessing expertise for the purpose of solving some current issues in traditional information preprocessing in the web log mining. According to Jiang Chang-bin and Chen Li [12], the web log data preprocessing algorithm is based on collaborative filtering which will identifies the session easily and quickly.

Pushpa .V. et al., [14] used web explorer tool, to analyze the web server log and generated reports. The rate at which log files are produced in modern distributed applications ranges from several terabytes to petabytes per day. Among the different types of logs recorded on the application servers, access logs contain the information related to user navigational behavior and user access patterns. Tsuyoshi et.al [16] has developed a method for clarifying user's interests based on an analysis of the site keyword graph. This method is used for extracting sub graphs representing user's main interests from a site keyword graph which is generated from web log data.

III. WEB LOG FILES

A web log files [8] is the significant source utilized in the web usage mining process. When the user requests the particular web page on the website, entry will be created as a record on the server automatically, by maintaining a history of user browser behaviour. Typically, the web log contains set of fields such as IP address, date & time, request method, status, bytes, referrer, user agent etc.

A. Various Sources of Data

Web logs are collected from different data sources like web server, proxy server, and client server [8] [14].

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
*** Correspondence Author**
Chaitra H K, Assistant Professor, Department of CSE, SJB Institute of Technology, Bangalore, India. Email: chaitrahk82@gmail.com
Dr. Suneetha K R, Associate Professor, Department of CSE, Bangalore Institute of Technology, Bangalore, India. Email: suneetha.bit@gmail.com

- **Web server logs**: The log files are collected from web server [5], which is contains user log records. Server log maintains the relationship between the web server and the users. Generally, four types of server log files are available [4], which are named as access log, error log, referrer log and agent log.
  - Access logs: access log files store all incoming requests of the user access
  - Error logs: error log is used to store all the failed http error files
  - Referrer logs: the referrer logs maintain the records, how the user has linked to each site and each page.
  - Agent logs: it indicates information about browser, operating system that is used by the user.
- **Proxy server logs**: The proxy server log files are stored in the proxy server, which is used to handle the user request page when the main server is unable to respond to the user access.
- **Client server logs**: The client server log is also called as browser log, which consist of client browser. Client server logs helps to handle the web page caching and session reconstruction problems with the use of HTTP cookies.

B. ***Web Log File Format***: There are different types of weblog format[9]available in the various web servers such as

- W3C extended format – W3C is defined by the World Wide Web consortium, which is an access log for web server. It contains data about each access request.
- NCSA common format – this is regularizing text file format used by the server, which format is fixed ASCII text so the user unable to customize it.
- IIS log format- this is a fixed ASCII text format. But it contains more information than the NCSA log format. These three are common web log file format on the web server.

C. Web Log structure :Web server los are plain text files, which is independent from the server platform, the following is the segment of the server logs,

137.226.113.28 - - [17/Dec/2019:23:38:56 +0000] "GET / HTTP/1.1" 200 275 "-" "Mozilla/5.0 zgrab/0.x (compatible;Researchscan/t13rl;+http://researchscan.comsys.rwth-aachen.de)"

The above code reflects the information such as , remote IP address or domain name, Authuser ,entering time and exiting date and time ,modes of request such as GET,POST or HEAD method ,HTTP status code [17] ,remote log and agent log, remote log and agent log, ”request” line from the client , requested URL.

IV. DATA PREPROCESSING

The weblogs usually contain the adequate information about the click stream data of user requests. It may be incomplete, noisy and unstructured data. Data pre-processing is the one of the phases of web usage mining, which is used to remove the inconsistent, irrelevant and redundant data [13] [14]. The main intention of data pre-processing is to transform raw logs into the cleaned logs that can be given for further processing of pattern analysis [4] [5]. The data pre-processing has main stages like data cleaning which reduces database size, user identification, session identification, path completion [15] used to retain relevant fields of the weblogs effectively.

Our work focuses on analysation of web log file; hence the contents of the access log file needs to be pre-processed.

A. Attributes Extraction

When importing log files, it is highly desirable to extract structured data. Most logs have some sort of structure to them. For instance, each line in a web access log can be decomposed into fields such as the URI, HTTP method, client IP address, user-agent, etc. By extracting structured attributes, we greatly expand our ability to query and visualize log data. [7]. Delimiter based Field Extraction algorithm is given below.

Input to the algorithm: Raw Server Log File
Output: cleaned log Database
Open the Raw Server Log File and
Read every attribute contained in Raw Server Log File
Separate out the entire Attribute using the delimiter
Extract all fields and Add into the Log Table (LT) and
Close the file

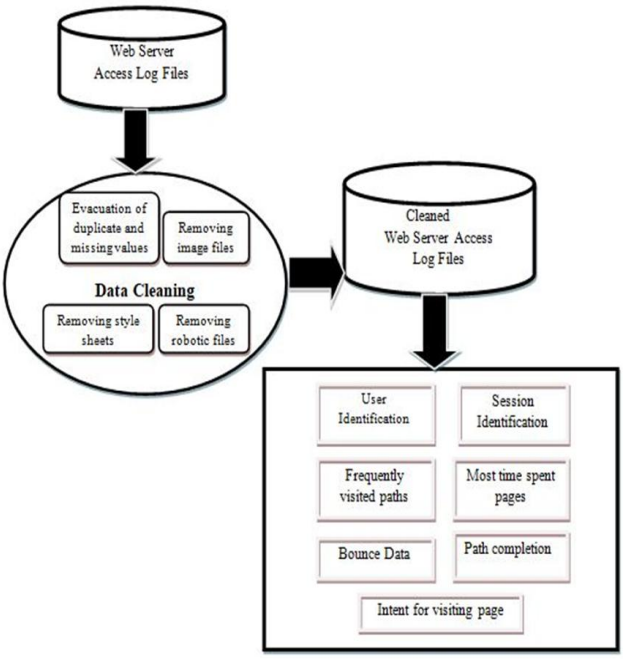

Fig.1.Data Pre-processing stages and outcomes

Each and every section of the log entries has been separated to attributes for easy cleaning of unclean data from the large data source.

B. Data Cleaning

Data cleaning is an essential process of data pre-processing which is used to remove the unwanted information from the web logs [14]. The log files include very large data and it takes around 85% process work in the data cleaning step, the cleaned data used further for effective pattern discovery. Irrelevant information is removed during data cleaning which is mentioned below:

- Remove Image File:

The log file entries contain different extension format. But we need only relevant extension data. The extension files of image file, graphics or multimedia and style sheet pages will be eliminated from log files like .gif, .jpg, .jpeg, .css [6]. By applying Regex Expressions,

search = recompile(log_re). search

matches = (search(line) for line in file (filename))

return 'server_ip':x. group('ip'), 'uri':x.group('uri'), 'time':x.group('time'), 'status_code':x.group('status_code'), 'referral':x.group('referral'), 'agent':x.group('agent'),

```
{
attributes: {},
lineMatchers: [
 {
  //All Access logs Types
  attributes: {"logType": "access log"},
  /**
  All consecutive lines matching this pattern are included in the group. The first line (the line that matched the start pattern) does not need to match the continue Through pattern
  **/
  extractors: ["(ip=([^\" ]+)(\"[^\"]*\")) (user=[^ ]+) (authUser=[^ ]+) \\[(timestamp=[^\\]]*)\\] \"(method=[^ \"]+) (uri=[^ \"]+) (protocol=[^\"]+)\" (status=[^ ]+) (bytes=[^ ]+) \"(referrer=[^\"]+)\" \"(agent=[^\"]+)\""],
  /**
  If an extractor does not match, it is simply ignored; the overall matcher is still used. However, if a non-matching extractor has required: true, then the matcher does not apply to this log record.
**/
  pattern: "(([^\" ]+)(\"[^\"]*\")) (([^\" ]+)(\"[^\"]*\")) (([^\" ]+)(\"[^\"]*\")) \\[[a-z0-9:\/ +-]+\\] \"[^ \"]+ [^ \"]+ [^\"]+\""

 },
 /** If Above Pattern is not matching **/
 {
  attributes: {"logType": "unparseable"},
  pattern: ""
 }
]
}
```

- Remove Failed Status Code:

The log entries with failed status code which is to be removed from the weblogs. There is different failed status codes available it may be like this, 400 - Invalid request page, 403 - Forbidden page, 404 - Page not found, 206 - Partial content, 304 –Not modified, 412 - Condition failed. These are the some of the failed status code, but we need only the log entries with success that is 200 status codes and rest of others will be removed from the weblogs.

- Remove Spider/ Robots Files:

The weblogs also entries with automated search engine such as robots, spider and crawler files removed from web logs. The robot file extension with robots.txt that is has to be removed.

C. **User Identification** There is various methods for identifying. A simple method for the identification of session is the timeout method. In this method a predefined threshold value is used. If the inter-arrival time between the two tcp connections is less than the threshold value, then both connections are belonging to the same session and if it is greater than the threshold value then the first connections belongs to the current session and the second connection belongs to the next session. The algorithm to representation the User Identification is given below.

Unique User Identification algorithm from logs (UUI)
Read each entry in LT
If an Timeout and IP address not exist then
Consider the user as new user
End if
If Timeout and IP address exists and the ((browser version or Operating System) is not exist) then
Consider the user entry as new user
Else if
Next entry

D. Session Identification

In Real Time Scenario, it's difficult to know when the user is finished with the browsing. In many servers there's a timeout that naturally closes a session except if another page is accessed by a user. The first run through a user associates a session/cookie ID is made (how it's done relies upon the web server programming and the kind of verification/login you're utilizing on the webpage). Like treats, this normally doesn't get sent in the URL any longer since it's a security issue. Rather it's put away alongside a lot of other stuff that all in all is additionally alluded to as the meeting. Session factors resemble treats - they're key-value sets sent alongside a solicitation for a page, and returned with the page from the server - however their names are characterized in a web standard [7]

Time Oriented Heuristic Algorithm for Session Identification (TOH)

TOH depend on time confinements on all out meeting time or page-stay time. There are two kinds of time arranged heuristics. In the first, the duration of session can't be more noteworthy than a predefined upper bound A. The upper bound A is generally acknowledged as 30 minutes as indicated by any page mentioned with timestamp (I) can be affixed to the present session if the time distinction between the mentioned page's timestamp and the timestamp of the main page I0 of that meeting is littler than A ($i - i\,0 \leq A$). The main web page with time stamp more prominent than $I0 + \delta$ turns into the primary page of the new meeting. As it were, if [Page1, Page1, …., Page N] are site pages framing a meeting (in expanding request of access time), at that point get to time(Page) – get to time(Page) ≤ A. The pseudocode for TOH utilizing all out time limitations given underneath:

```
Input:
L : The set of input logs, |L| : the number of input logs
δ : user defined upper bound
Output:
 FinalSessionSet

FinalSessionSet = {}
Function Log_Parser_TimeOrineted_I ( |L|,L, δ)
 For each Li of L 47
        If Methodi is 'GET' AND Urli is 'WEBPAGE'
                If ∃ Sk ∈ FinalSessionSet with IPk = IPi then
                        If ( (Timei – START_TIME(Sk) ) < δ) then
                Sk = (IPk , PAGESk ● Urli)
        Else
                Close_session(Sk)
                Open_session(IPi, Urli)
        End if
        Else
                Open_session(IPi, Urli)
        End if
        End if
 End For
```

. **Most Frequently visited pages**

Most Frequently visited pages are identified based on classifying the users browsing behaviours based on whether they visited a frequently visited page or not.

E. Most time spent pages

Sources of info are taken from the log file and Count the quantity of one of a kind users visited pages.

Using the user passage time and the leave time of a specific page, Calculate the all-out time spent on that specific page, List the pages in the request for the most time spent pages and afterward distinguish the top pages.

F. Path Completion

Path completion is the final step of pre-processing which are acquiring the entire path of the user access. The user page request is could not recorded in the log entries during the caching problems, POST method and during the use of "forward" or "back" button of a browser. If the user page request is not directly linked to the last page requested, then check the recent history, in case entries are not available in recent history then have to check the referrer URL in which the page request closest to the unknown page request that source is filled in this path and the pages is added to the user session.

G. Bounce Rate

A **bounce rate** is the percentage of users who leave the site after just viewing one page. Since the user only viewed a single page, it is considered as a bounced user. Bounce rate calculated using the following formula (1):

$$\text{BROP} = \frac{\text{Total number of singlepage sessions}}{\text{Total number of entrances on the page}} \quad \text{-- (1)}$$

A high bounce rate implies that the site or page isn't pertinent to what users are searching for. Thus, user will open the site yet then hit the back catch or exit without communicating with some other page. Site stacking speed is perhaps the most compelling motivation for a high skip rate.

In the event that the site sets aside a long effort to stack, at that point the users would get a poor client experience and would choose to leave. As the load time of a page increases, the probability of bounce rate also increases as shown in Table- I. Further in the next section we show the bounce rate by device type.

Table- I: The thumb rule of Bounce rate

Sl.No.	% of Bounce rate	Description
1	80%+	very bad
2	70 – 80%	poor
3	50 – 70%	average
4	30 – 50%	excellent
5	20% or below	likely a track in error

V. EXPERIMENTS AND RESULTS

For experimental analysis datasets are collected from a single webdomain**"http//www.kannadauniversity.org/log/access.log"**which is dated from 1.12.2019 to 31.12.2019. All the information used in this work is associated with the Kannada University Hampi, Vidyaranya Karnataka state. This section provides the experimental results obtained by using all the methodologies described in the previous section, there are totally 130MB raw logs, these access log files details are taken as input and then followed by data cleaning and testing the log parsers for attributes, extractors and pattern matching the experimental results obtained are total entries, total number of hits, failed requests, total number of unique users identification and also number of unique sessions.

The work is implemented using colabs Notebook in which it provides jupyter notebook environment runs entirely in the cloud.

The recorded entries of an access log which contains, image files, error files, other files. The collected raw log file is shown in the Fig.2.

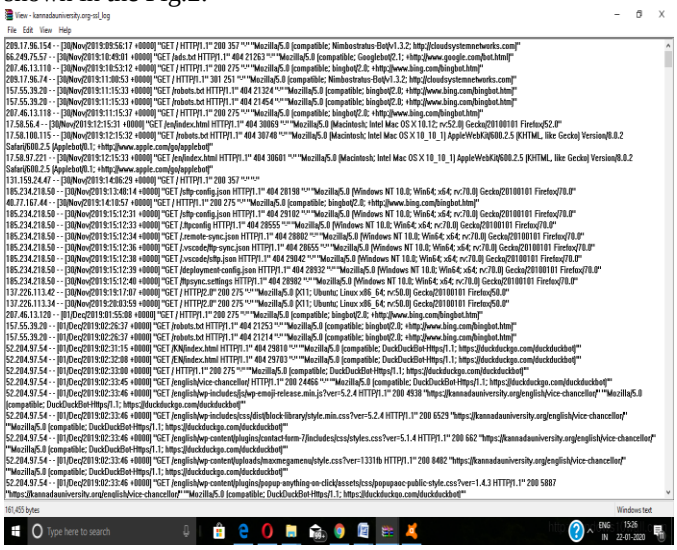

Fig.2. The raw web log files.

The irrelevant fields like - .jpg, .jpeg, .png, Error code - 404, 206, 304, Other files - .js, .css,.log are eliminated from weblogs and only retain the useful fields, Kannada university server log files of one-month data is processed in our work Table II gives complete idea of percentage of reduction compare to original size as shown in the Table- II

Table- II: The Results of Pre-processed data

Server log file	*Kannada University Hampi*
Duration (no. Of Days/Months)	31
Original size	130 MB
Reduced size after Pre-processing	20 MB
Percentage in Reduction	85%
Total Number of unique users	2031

The access logs of Kannada University are taken from their website; the raw web log format has multiple attributes that are represented using the single field which needs separation during pre-processing. Using the methodology the log files are separated according to ip address, user id, host id, date time, method, path, status code, bytes, and referral path and user agent.[18]. Table III shows the results obtained after testing the log parsers.

Table - III: Parsed raw logs information

IP Address User id, and Host id	
Fields	*Count of web log entries*
IP Address	230000
User id	4222
Host id	3211
method field	
Method Field	*Count of Weblog entries*
GET	295095
HEAD	569
POST	13911
Protocol field	
Protocol field	*Count of web log entries*
HTTP 1.0	4209
HTTP 1.1	-
Results of successful status code field	
Status code	*Count of web log entries*
200	239453
204	0
206	24747
User agent field	
User Agent	*Count of web log entries*
Mozilla/5.0	278969
Opera/9.80	33063
Mozilla/4.0	36
Google	244939
other	82
Successful status codes	
Status code	*Count of web log entries*
200	239453
204	0
206	24747

The results obtained by page extension field [18] are shown in the Table-IV.

Table - IV: Page extension field obtained

Page extension field		
Extension field	Count of web log Entries	Percentage%
.jpg	42224	18.35
.css	77582	33.73
.png/.jpg	14226	6.185
.pdf	37368	16.24
.rar	4155	1.8
.txt	5172	2.24
.ico	7721	3.35
.woff	0	0
.php	5068	2.2
Others	321	0.139

Unsuccessful request code is a three digit response from the browser, when a users request is not succeeded then web logs will record this request as unsuccessful request code is shown in Table-V

Table -V: Unsuccessful status code field

Results of unsuccessful status code field		
Status code	Count of web log entries	Percentage%
404	13950	76.22
409	3526	19.26
301	215	1.17
500	96	0.524
503	86	0.469
403	78	0.426
400	3	0.0163
405	1	0.0054

By observing the Table- VI the system administrators can able to guess the number of entries, IP address, unique users, hits and failures occurred during this period and can also predict the most preferable time to shut down the server.

Table - VI: User Profile information

Number of entries	*9112*
Number of IP address	3403
Number of unique users	2031
Number of Hits	260453
Number of failures	18302

Users will get required information in the website, when the website is used by the users, for each of the request from the user's web browser to a web server, a response is generated, and this server request is called Hits[18] and is shown in Table-VII.

Table -VII: The information about the Hits

Results of Hits before pre-processing		
Actions	Count of web log entries	Bytes(MB)
Total hits	5600000	130 MB
Results of Hits After pre-processing		
Actions	Count of web log entries	Bytes(MB)
Total hits	122000	20MB

The information about the hits before preprocessing and after preprocessing is shown in the Table-VII and Fig .3, gives the detailed information about the Hits per minute.

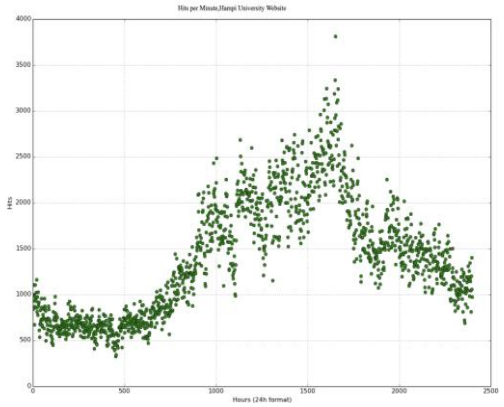

Fig.3. Hits per Minute

Fig. 4 shows the apache web logs extracted information from web server access logs of Kannada University Hampi. The information shown in the Fig. 5 is the data browsed using the different browsers in the website, Fig. 6 shows the users interest browsing and Fig. 7 shows the average bounce rate by device type which is very useful to understand the user interest and admin can update the information related to the Hits per minute, apache web logs, data browsed using different browsers and user interest browsing accordingly.

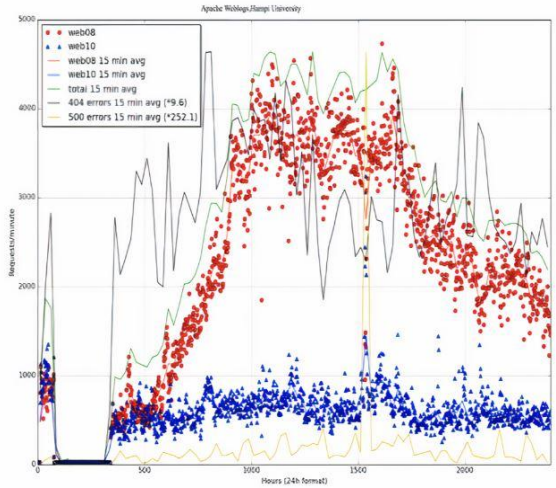

Fig. 4: Apache web logs of university

Ninety-eight percent (98%) of the data is browsed from Mozilla Browser. Kannada University website was accessible and links were successfully browsed from mobile and web 92% of the times. Users mostly browsed Kannada and English refer links.

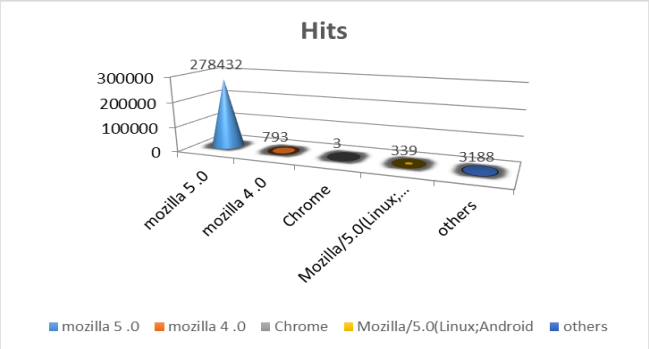

Fig. 5: The data browsed using different browsers in university website

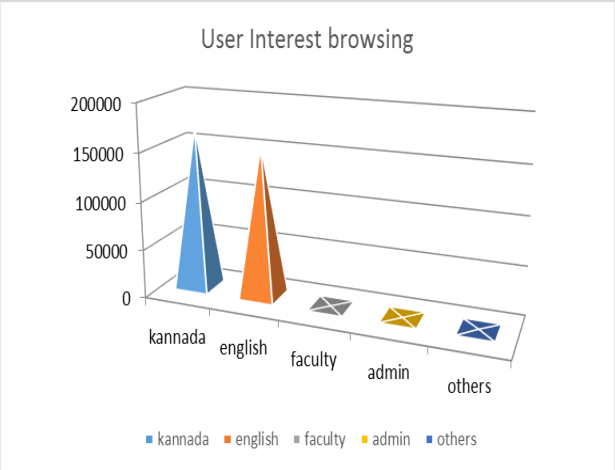

Fig. 6: Users interest browsing.

Thinking about the gadgets, Versatile users are bound to ricochet in all cases, so it ought to sensibly follow that any site with an enormous, developing level of portable traffic will see bounce skip rate. Tablets are not particularly unsurprising-once in a while not as much as work area, at times more. The percentage of bounce rate is shown in the Table-VIII

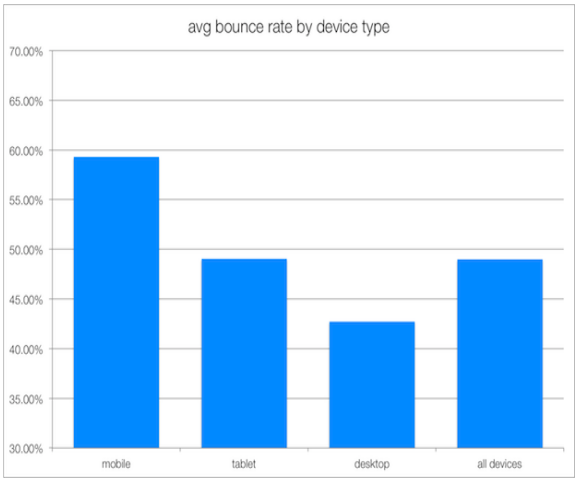

Fig. 7: The average bounce rate by device type

By and large, expect versatile bounce rates to ring in around 10 to 20 percent higher than sample data.

In this we took a gander at the versatile ricochet rate over a little example of sites from 01st Dec 2019 to 31st Dec 2019 (the hour of composing), and the normal portable bounce rate was only a shade under 60%. The rate for tablets was generally 49%. Sample data users bounced just 42% of the time all things considered.

Table -VIII: Percentage of bounce rate by device type

Devices	Bounce rate (%)
Mobile	58
Tablet	49
Desktop	43
All devices	49

VI. CHALLENGES OF WEB SERVER LOG ANALYSIS

The most challenging issue with web server log analysis is that HTTP is a stateless protocol: The transmission protocol liable for communication among server and user handles each request independently. The web server additionally doles out two diverse site hits to a single user – something fairly unfeasible for the processing of a user's general conduct. There are a few different ways to tackle this issue:

Assigning a session ID: The user id is server-produced and put away in the client's program. On the off chance that a client is given such a recognizable Id, at that point all questions they submit to the web server are prepared by means of a similar Id. Every one of their activities is in this way joined into a single user. There are two choices for the task of an ID: One alternative is to utilize treats, which are put away beginning with the user's first solicitation and afterward transmitted on each further contact with the server. Treats are not obvious in the log file however, thus require exceptional programming in the event that you need to have the option to break down them later. The other choice is to transmit the user ID as a URL parameter. Be that as it may, for these client explicit connections, a higher programming exertion is required. From the Search engine optimization (SEO) perspective, the individual URLs can cause issues. A similar substance is available at an alternate URL each time the crawler visits the page, so it could without much of a stretch be confused as copy content.

User identification via IP address: For whatever length of time that the entirety of a user's activities is created to a similar IP address, they can be extraordinarily distinguished utilizing this technique. The essential (in the assessment of numerous information assurance specialists) is that the user has concurred ahead of time to the kitty of their IP address for analysis purposes. Issues emerge when guest users are used their IP powerfully as are checked on numerous occasions, or if a few clients are utilizing a similar IP – for instance, by means of an intermediary server.

Using server-independent measures: Purported following pixels – fundamental structure squares of page labeling – which are obviously implanted in the site, give propelled data, for example, screen goals and which program modules a client has introduced. In mix with the IP address and the data on the program and working framework, clients can be recognized in a specific way. A 100% detachment of clients is preposterous utilizing this strategy. Be that as it may, with the assistance of pixel gadgets or AJAX components, you can follow. In a basic log document examination, you don't get any uncommon data about the manner in which it's utilized.

Another issue with log file processing is the caching capacity of the internet browser or intermediary server. While it's critical for the quick conveyance of the mentioned information, it likewise implies that users don't generally come into contact with the web server. Requests that worry stored content just show up in a constrained manner in the server log records (status code 304 "Not altered"). It ought to likewise be referenced that extra assets are required for the lasting logging, stockpiling, and assessment of the server gets to – particularly for web ventures with a high traffic volume. Also, the log file analysis does exclude significant markers, for example, the bounce rate or the length of visits, which is the reason it should just be utilized as an enhancement to other test instruments and not as the sole strategy for.

VII. CONCLUSION

This paper has bestowed the important points of data pre-processing tasks that are useful for performing pattern discovery and analysis. The experimental results of Kannada University Hampi web server access log file are pre-processed using several techniques. The cleaned log data is passed to the next level for user identification and session identification. The result of data pre-processing has provided the information related to identify the unique users, sessions, total number of Hits, apache web logs details ,successful request, browsers information and also user interest browsing, investigations are done to explore the percentage of bounce rate by device type. Experimental result shows the efficacy of the algorithms and heuristic approaches. However, with this there are few problems such as data collection, accuracy measures further needs to work on overcoming the challenges of web server log analysis such as assigning a session ID and To use server-independent measures. Even though some of the work focused on these areas but still more work needs to be inspect. These pre-processed weblogs information is used for further stages of pattern discovery and pattern analysis.

ACKNOWLEDGMENT

The authors Chaitra H .K and Dr. Suneetha K .R would like to thank Kannada University Hampi, Vidyaranya Karnataka (KUH) for providing the necessary resources for accomplishing the research work.

REFERENCES

1. R. Kosala, H. Blockeel, Web mining research: a survey, SIGKDD: SIGKDD explorations: newsletter of the special interest group (SIG) on knowledge discovery & data mining, ACM 2 (1), 1–15, 2000
2. Bamshad Mobasher et.al "Effective Personalization based on Association rule Discovery from Web usage data" WIDM01 3rd ACM workshop on Web Information and data management, November 9 2001, Atlanta 2001.

3. R. Cooley, B. Mobasher, and J. Srivastava, Data Preparation for Mining World Wide Web Browsing Patterns. Journal of Knowledge and Information Systems, (1), 1999.
4. Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, et. Al, "Data Preprocessing on Web Server Logs for Generalized Association Rules Mining Algorithm", World Academy of Science, Engineering and Technology, 2008
5. Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang Ning Tan "Web usage mining: Discovery and Applications of usage patterns from web data" SIGKDD Explorations- vol-1, issue-2 Jan 2000 pages 12-33.
6. Mobasher, B., Cooley, R., Srivastava J.: Automatic personalization based on web usage mining .ACM43(8),142-151(2000)
7. Radha.M, K. Santhi," An Novel approach on Pre-Processing Techinique on web log mining" International Research journal of engineering and technology, volume 4, issue 5, May 2017.
8. Dr. R. Krishnamoorthi and K. R. Suneetha," Identifying User Behavior by Analyzing Web Server Access Log File", International Journal of Computer Science and Network Security, April 2009, vol 9, no.4, pp.327-332.
9. W.W.W. Consortium the Common Log File format http://www.w3.org/Daemon/User/Config/ Logging.html#common-logfile-format, 1995
10. Lizhen Liu, Junjie Chen, Hantao Song, "The Research of Web Mining", Proceedings of the 4th World Congress on Intelligent Control and Automation, June 10-14, Shanghai/China, 2002
11. Ling Zheng, Hui GUI and Feng Li. 2010. Optimized Data Preprocessing Technology for Web Log Mining. IEEE International Conference on Computer Design and Applications (ICCDA), pp. 19-21.
12. JING Chang-bin and Chen Li. 2010. Web Log Data Preprocessing Based On Collaborative Filtering. IEEE 2nd International Workshop on Education Technology and Computer Science, pp. 11
13. R.M. Suresh, R. Padmajavalli. "An Overview of Data Preprocessing in Data and Web Usage Mining". 2006 IEEE
14. V. Pushpa et al," An Efficient Preprocessing Method to Detect User Access Patterns from Weblogs" International Journal of Computer Science and Mobile Computing, Vol.5 Issue.9, September- 2016, pg. 16-22
15. Zhang Huiying, Laing Wei "An Intelligent Algorithm of Data Pre-processing in Web Usage Mining" Proceedings of the 5th world Congress on Intelligent Control and Automation, June15-19, 2004 Hangzhou, P.R. China.
16. Tsuyoshi Murata and Kota Saito "Extracting Users Interests from Web Log Data" Proceedings of the 2006 IEEE/WIC/ACM International Conference of Web Intelligence (WI 2006 Main Conference Proceedings) (WI'06) 2006 IEEE.
17. Internet: Hypertext Transfer Protocol Overview, http://www.w3.org/Protocols/,http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html, 1995.
18. P.Sukumar,L.Robert,S,Yuvaraj "Review on Modern Data Preprocessing Techniques in web usage mining ",International conference on computational system and information systems for sustainable solutions.2016

AUTHORS PROFILE

Chaitra H.K is working as Assistant Professor in the Department of Computer Science and Engineering at SJB Institute of Technology, Bangalore, India. And pursuing Ph.D. Degree in the Department of Computer Science and Engineering, Research center at Bangalore Institute of Technology, Bangalore, India, Under Visvesvaraya Technological University Belagavi, India. She obtained her BE degree in 2005 and M. Tech degree in 2011 from Visvesvaraya Technological University, Belagavi, India. She is a life member of ISTE; her research interest is in Data Mining, Web Mining, Big Data Analytics and Data Science.

Dr. Suneetha K.R is working as Associate Professor at Bangalore Institute of Technology (from 2002) in the Dept. of Computer Science Engineering. She has obtained her Ph.D. in Information and Communication Engineering from Anna University, Chennai, in the year 2014. She has published several research papers in the reputed Journals, International and National conferences. Her area of interest include Data Mining, Web Mining, Big data, Artificial Intelligence and Data Science.