Knowledge Discovery from Log Data Analysis in a Multi-source Search System based on Deep Cleaning

Fatma Zohra Lebib[1,2], Hakima Mellah[2] and Abdelkrim Meziane[2]
[1]*University of Science and Technology Houari Boumediene, USTHB, Algiers, Algeria*
[2]*Research Center in Scientific and Technical Information, CERIST, Algiers, Algeria*
{zmatouk, hmellah, ameziane }@mail.cerist.dz

Keywords: Log Files Analysis, Web Usage Mining, Multi-source Search System, Knowledge Extraction, Information Source, User Profile.

Abstract: In a multi-source search system, understanding users' interests and behaviour is essential to improve the search and adapt the results according to each user profile. The interesting information characterizing the users can be hidden in large log files, whereas it must be discovered, extracted and analyzed to build an accurate user profile. This paper presents an approach which analyzes the log data of a multi-source search system using the web usage mining techniques. The aim is to capture, model and analyze the behavioural patterns and profiles of users interacting with this system. The proposed approach consists of two major steps, the first step "pre-processing" eliminates the unwanted data from log files based on predefined cleaning rules, and the second step "processing" extracts useful data on user's previous queries. In addition to the conventional cleaning process that removes irrelevant data from the log file, such as access of multimedia files, error codes and accesses of Web robots, deep cleaning is proposed, which analyzes the queries structure of different sources to further eliminate unwanted data. This allows to accelerate the processing phase. The generated data can be used for personalizing user-system interaction, information filtering and recommending appropriate sources for the needs of each user.

1 INTRODUCTION

Web servers record user activities and store them in log files which include fields related to the user requests. IP address, port number, state code, and access time are commonly used fields in this kind of log files (Mobasher et al., 2002). Different from the past when log files were mostly neglected except for urgent situations when system related problems were experienced, nowadays they give important information to system administrators and are efficiently used for detailed investigations. In addition to their uses for various purposes, intrusion detection and prevention systems and digital forensics analysis are the emerging application fields of log files (Boeck et al., 2010).

The log file data represents the relevant information reflecting the user's interests, which can be exploited in many applications, including recommendation items, adaptation user services and improving information retrieval. The users' behaviour is hidden in these logs, which need to be discovered, and analyzed (Facca and Lanzi, 2005).

Web Usage Mining (WUM) is the use of data mining techniques to automatically discover and extract information from web data (Patel et al., 2011). The main goal of this technique is to discover usage patterns trying to explain the users' interests (Hernández et al., 2017). The user's interests can be extracted in a popular way, from his/her own profile (e.g. interests) or from his/her social behaviour (e.g. tagging behaviour) (Astrain et al., 2010) (Li et al. 2008), his social network (e.g. friends) (Yang et al. 2015), or from log data (Carman et al. 2010). However, detecting user's interests is a crucial problem (Milicevic et al., 2010). In a multi-source retrieval system, several information sources can contribute to provide the user with relevant results in response to his/her query. To improve the system performance and provide the personalized results to each user's profile, it is crucial to collect information characterizing different needs and interests of users.

The main objective of this work is to discover the user's interests from large log data of the SNDL[1]

[1] National online Documentation System (www.sndl.cerist.dz)

Lebib, F., Mellah, H. and Meziane, A.
Knowledge Discovery from Log Data Analysis in a Multi-source Search System based on Deep Cleaning.
DOI: 10.5220/0008121102570264
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 257-264
ISBN: 978-989-758-386-5

system that allows to access different national and international electronic documentations covering all the areas of education and scientific research. The SNDL system covers several information sources and databases, such as Springer Link, Science Direct and ClinicalKey, belonging to different domains of scientific research, including computer science, medicine, and law. The search history of users is recorded in log files, including the queries submitted to sources and the documents downloaded from these sources. These log files may contain a large amount of raw data, which must be transformed into a meaningful format, for example to obtain users' preferences and behaviour, and to understand in-depth the interests of users. An approach based on WUM techniques is proposed to analyze the SNDL log files and extract useful information to model an accurate user-interests profile. The user profile can be used further to customize the search results for each user, to filter the information based on the user's interests, or to recommend the appropriate sources or documents to a specific user.

The remainder of this paper is structured as follows. Section 2 discusses context and related work. Section 3 presents the proposed approach. Section 4 describes implementation of the proposed approach and Section 5 concludes the paper.

2 RELATED WORK

A log file is a text file containing all the events detected by a particular process, including monitoring a network and application. The events, including click, add to cart and execution of an application are dated and in chronological order. It contains all the information that has been defined as relevant by the programmers.

Log files are a valuable source of information that can be exploited for multiple purposes such as issues related to computer security and systems or applications functioning, improving the use of a site or a web application, optimizing the performance of a system and identifying user profiles.

However, it is impossible to manually analyze all this data, which are too large. Web Usage Mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. WUM consists of three phases, namely *preprocessing*, *pattern discovery*, and *pattern analysis* (Srivastava et al., 2000). Preprocessing consists of converting the usage, content, and structure information contained in the various available data sources into the data abstractions necessary for pattern discovery. Pattern discovery draws upon methods and algorithms developed from several fields such as statistics, data mining, machine learning and pattern recognition. Pattern analysis is the last step in the overall WUM process. The motivation behind pattern analysis is to filter out uninteresting rules or patterns from the set found in the pattern discovery phase.

Many researchers have widely used WUM techniques to discover interesting and frequent user navigation patterns from web server logs. In (Sujatha and Punithavalli, 2012), four types of clustering approaches are investigated in web log files to improve the quality of clustering for user navigation pattern in WUM systems, for predicting user's intuition in the large web sites. In (Vellingiri, et al., 2015), authors proposed an approach for discovering web user's navigation patterns and analyzing of web usage data using Weighted Fuzzy Possibilistic C-Means (WFPCM) for clustering and Adaptive Neuro-Fuzzy Inference System with Subtractive Algorithm (ANFIS-SA) for classification. Singh and Meenu (Singh and Meenu, 2017) presented visitor pattern analysis of educational institution web log data using WUM for discovering patterns and generating the reports. In (Dwivedi, 2017), log data has been analyzed using one of the most popular web log expert software tool (WebLog Expert). The process of web log analysis consists of data collection, preprocessing, pattern discovery, pattern analysis, and result analysis visualization. Jaya Kumar and Alagarsamy (Jaya Kumar and Alagarsamy, 2013) discussed about the web usage mining, different file format of log data and carried out the analysis task with the help of web log expert tool. In (Shanthi and Rajagopalan, 2013), an automatic discovery strategy for web server log information using the web page collection algorithm is proposed. The approach uses sequential pattern mining technique to identify the sequence of navigational patterns of web users. The extracted patterns allow to understand the customer interests. In (Mahoto et al., 2016), WUM has been used to predict users' web navigational behaviour. Two main techniques, cluster analysis and sequential pattern mining, have been used to get the desired navigational patterns from real web log datasets.

Some of these approaches used existing web log tools for attending specific objectives or generating the statistics. There are various tools (Kaur and Aggarwal, 2015) for analyzing log files such as Google analytics, Stat Counter, Deep Log Analyzer, Awstates and Web Log Expert. What can be seen is

that some of the cited works focus on the pre-processing step of the Web Data while others try to concentrate on the behavioural models of Internet users visiting websites.

The weaknesses in existing tools are summarized in the following points:

- Their configuration is not compatible with all needs and requires some skills;
- Only statistics results are provided, including number of visitors and the amount of data downloaded;
- No knowledge on the log file content is provided;
- No structuring of data in a database is possible;
- Especially, the SNDL system manages several resources that require in-deep skills.

3 THE PROPOSED APPROACH

The proposed approach allows to extract useful information that reflects user's profile and interests from the SNDL log files. The information needed to be extracted from these log files is as follows:

- User introducing the query (his/her name);
- Session;
- Date and time;
- User query (search / download);
- Keywords (query terms);
- Sources accessed by a user.

The proposed approach consists of two major steps, which are: Log file pre-processing and Log file processing, as shown in Figure 1.

i) The first step, i.e., log file pre-processing, allows to prepare the log file for further processing. It consists of the following three sub-steps:

- Log file cleaning;
- Log event identification;
- Activities segmentation by session.

ii) The second step, i.e., log file processing, allows to perform deep analysis through the following sub steps:

- Extracting and partitioning homogenous data;
- Identifying the source targeted by each query;
- Keywords extraction.

Before describing the different steps of the proposed approach, the log file structure considered in this study is presented in the following section..

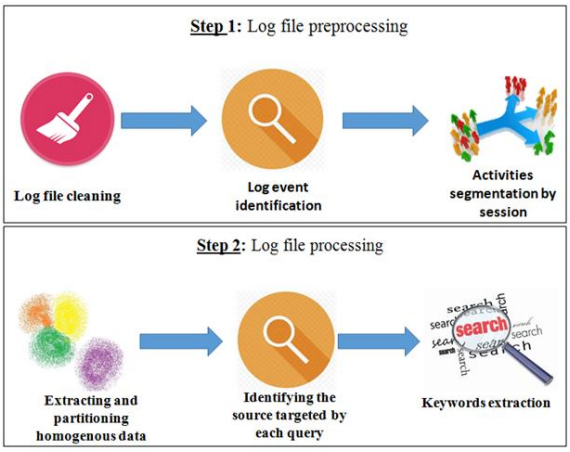

Figure 1: General schema of the proposed approach.

3.1 Structure of the SNDL Log File

A log file of the SNDL system is of a common log format. An event represents a line in a log file. Figure 2 shows an example of an event of this file.

```
41.98.29.193 ilWvTPtsaK6yDwC sahnounefoudil [20/Feb/2013:00:00:18 +0100]
"GET http://www.springermaterials.com:80/docs/pdf/10000866_118.html
HTTP/1.1" 200 15413
```

Figure 2: Example of an event in the SNDL log file.

An event is composed of the following attributes:

- 41.98.29.193: IP address of the user;
- ilWvTPtsaK6yDwC: session ID;
- sahnounefoudil: user name;
- (20/Feb/2013:00:00:18 +0100): date and time of query submission;
- GET : method used by the protocol;
- http://www.springermaterials.com:80/docs/pdf/10000866_118.html : URL of the source;
- HTTP/1.1: protocol used and its version;
- 200: status or code returned from the server;
- 15413: size of the requested page in bytes.

The "common" structure of an event of the SNDL log file is given by the following form.

IP	Session	User	Date	Method	Query	Protocol	Status	Size

3.2 Log File Preprocessing

3.2.1 Log File Cleaning

Log data is typically noisy and unclear, so data cleaning is an essential process for effective mining process. Data cleaning allows to remove irrelevant items stored in the log files that may not be useful

for analysis purposes (Cooley et al., 1999; Cooley et al., 1997), such as access of JPEG, GIF file, Java Scripts, other audio/video files, non-human accesses (accesses made by web robots), and accesses with failed HTTP status codes.

Since the interesting data in the log files is the queries submitted to the sources and the documents downloaded from the sources and not in any system generated data, it is important to keep only this data in the log files after the cleaning process.

The cleaning process is divided into two different steps, namely conventional cleaning and deep cleaning, where:

- Conventional cleaning: is similar to cleaning methods used in most data mining techniques which consist in removing multimedia files, style pagefiles, java script files, error codes, and web robot requests;
- Deep cleaning: analyzes the queries structure and eliminates those that are not relevant to our analysis, which do not contain information on user queries and downloaded documents.

Before describing these two cleaning steps, the following mathematical formulations are given.

Formal Description of the Cleaning Process: Let F be a log file in the common format. Let E be the set of events of F. E = {e_1, e_2, e_3… e_m}, e_k is an event of F where $1 \leq k \leq m$. Let $e_k{}^t$ be an event of F being processed. The cleaning process is performed according to the following rule:

if ($e_k{}^t$ satisfies condition c) **then** (E = E - {$e_k{}^t$})

This means that if the current event meets the predefined condition then remove this event from the log file.

Conventional Cleaning: This first cleaning step is to eliminate lines from the log file that are generally useless in almost every data mining process because they contain no relevant information.

Conventional cleaning rules are defined using the following predefined functions, namely:

Status: $e_k \mapsto$ Status (e_k): returns the value of the "Status" attribute of e_k

Query: $e_k \mapsto$ Query (e_k): returns the value of the "Query" attribute of e_k

Size: $e_k \mapsto$ Size (e_k): returns the value of the "Size" attribute of e_k

User: $e_k \mapsto$ User (e_k): returns the value of the "User" attribute of e_k

Extension: Query (e_k): $\mapsto$ Extension ((Query (ek)): returns the file type of the query associated with e_k

File: Query (e_k) $\mapsto$ File (Query (e_k)): returns the file name of the query associated with e_k

Let D be a set of file extensions, such as: D = {jpg, jpeg, gif, png, bmp, ico, mp3, wma, wav, ogg, mp4, mkv, avi, WebM, CSS, js}

Examples of conventional cleaning rules are presented in Table 1.

Table 1: Conventional cleaning rules.

Rule number	Rule	Description
1	(Status(ek^t) < 200 ∨ Status (ek^t) > 399) ⇒ E = E – {ek^t }	Remove the event with status codes over 299 or under 200 (the failed HTTP status code)
2	Size(ek^t) = 0 ⇒ E = E – {ek^t}	Remove the event that does not contain relevant information, including pages used to allocate a session, open an account, and redirect to another page
3	Extension(ek^t) ⊂ D ⇒ E = E – {ek^t}	Remove the events have filename extension of gif, jpeg, css, and so on (e.g. multimedia files)

Algorithm 1 describes the conventional cleaning process.

Algorithm 1: Conventional cleaning.

Input: the log file F
Output: the log file after the conventional cleaning F_c
begin
while (there are still lines in F) do
ek^t = Read(F) // The current line
if (Status(ek^t) < 200 ∨ Status (ek^t) > 399) then E = E – { ek^t }
else if (Size(ek^t) = 0) then E = E – { ek^t }
else if (Extension(ek^t) ⊂ D) then E = E – { ek^t}
else if (File(ek^t) = robots.txt) then E = E – { ek^t }
else if (User(ek^t) = '-') then E = E – { ek^t }
endwhile
Remove the "IP", "Method", "Status", "Protocol" and "Size" fields from F
F_c = F
End

Deep Cleaning: is performed after the conventional cleaning, to further eliminate lines that are not interesting for this analysis. The events that do not contain information on user query and documents downloaded are removed. This can be done by checking the query content in each event. As the NDL system manages several sources, including Springer Link, Science Direct, and IEEE, different query formats can be found in the log file (each source has its own query format). This step requires

understanding the query structure of each SNDL source in order to eliminate the irrelevant queries from the log file after the conventional cleaning.

Two types of query are distinguished, namely:

- Download query: used to download PDF documents from SNDL sources;
- Search query: containing the keywords entered by the user for SNDL sources.

The study of the queries structure of SNDL sources is based on the main composition of the HTTP query. Indeed, an HTTP query is decomposed in the following attributes (Shepperd et al., 2018):

- Protocol used (http protocol);
- Domain name (Web server or source);
- Path to the file (is the path to the resource);
- Parameters (Each Web server has its own rules regarding parameters).

Figure 3 shows an example of query consisting of the following attributes:

- Protocol: http;
- Domain name: link.springer.com;
- Path: /search;
- Parameters: facet-discipline.

http://link.springer.com:80/search?facet-discipline=%22Physics%22

Figure 3: Example of query from the SNDL log file.

Two pattern dictionaries are created, describing all the query patterns of the SNDL sources, which are:

- Search patterns dictionary: describes the search queries (Table 2);
- Download patterns dictionary: describes the download queries (Table 3).

Table 2: An excerpt from the search patterns dictionary.

Source name	Domain	Path(s)	Relevant parameters
ACM	dl.acm.org	/results.cfm	Query
Springer Link	link.springer.com	/search	query; term
zbmath.org	zbMATH	/;/authors/;/journals/;/classification/	q;f

Table 3: An excerpt from the download patterns dictionary.

Domain	Source name	PDF Marker
portalparts.acm.org	ACM	.pdf
delivery.acm.org		.pdf?
www.cairn.info	CARIN	load_pdf
ieeexplore.ieee.org	IEEE	.pdf?

Each SNDL source is represented by patterns able to determine if the query is a search query, download query or other.

a) Search Pattern: characterized by:

- Domain of URL (the concerned source);
- Source name;
- List of paths to pages dedicated to search;
- Search parameters containing the keywords.

b) Download Pattern: characterized by:

- Domain of URL;
- Source name;
- Download parameters (download markers).

Deep Cleaning Process: For this step, the following functions are defined.

Query: $e_k \mapsto Query(e_k)$: returns the value of the "Query" attribute of e_k

Domain: $Query(e_k) \mapsto Domain(Query(e_k))$: returns the value of the "Domain" attribute of the query associated with e_k

Path: $Query(e_k) \mapsto Path(Query(e_k))$: returns the value of "Path" attribute of the query associated with e_k

Params: $Query(e_k) \mapsto Params(Query(e_k))$: returns the value of "Parameters List" attribute of the query associated with e_k

Let D_R be the set of domains registered in the search patterns dictionary. Let D_T be the set of domains registered in the download patterns dictionary. Let M be the set of the download markers in the download patterns dictionary. Let N be the set of the paths in the search patterns dictionary. Let P be the set of the parameters in the search patterns dictionary. Table 4 describes the deep cleaning rules.

Table 4: Deep cleaning rules.

Rule number	Rule	Description
1	$(Domain(Query(ek^t)) \notin D_R \cup D_T) \Rightarrow E = E - \{ek^t\}$	Remove the event where the query is not associated with any of the SNDL sources
2	$Path(Query(ek^t))=\emptyset \wedge Params(Query(ek^t) = \emptyset)) \Rightarrow E = E - \{ek^t\}$	Remove the event where the query contains no path or parameter
3	$(Domain(Query(ek^t)) \subset D_T \wedge \forall x \in M, x \notin Path(Query(ek^t))) \wedge (Domain(Query(ek^t)) \subset D_R \wedge (Path(Query(ek^t)) \notin N \vee \forall y \in P, y \notin Params(Query(ek^t)))) \Rightarrow E = E - \{ek^t\}$	Remove the event where the query is neither a download query nor a search query

Algorithm 2 describes the deep cleaning process.

Algorithm 2: Deep cleaning.

Input: the log file F_c
Output: the log file after the deep cleaning F_n
begin
while (there are still lines in F_c) do
ek^t = Read(F_c) // the current line
if (rule 1 = true) then E = E – $\{ek^t\}$;
else if (rule 2 = true) then E = E – $\{ek^t\}$;
else if (rule 3 = true) then E = E – $\{ek^t\}$
endwhile
$F_n = F_c$
End

3.2.2 Log Event Identification

After removing irrelevant events from log file, this step consists in identifying the type of each event. An event can be either a search event or a download event according to the type of query. Deep cleaning not only eliminates unnecessary queries, but also identifies search queries and download queries according to rule 3 (Table 4).

3.2.3 Activities Segmentation by Session

It consists in segmenting user activities by session before extracting the data to distinguish the different visits of each user. At time t, a session identifier (ID) is assigned to a user connected to the SNDL system. The user will keep this identifier throughout his/her activity, at the break of the latter, or by closing the browser, a new session is assigned to the user. The probability that two users have the same session ID tends to 0. Due to the complexity of identifier generation, it is assumed below that if the session ID is known, then the name of the user is also known.

3.3 Log File Processing

3.3.1 Extracting and Partitioning Data

The relevant data is extracted and partitioned into two sets, namely a set of search events and a set of download events, such as:

A search event is characterized by:

- Session ID;
- User name;
- Source name;
- Date and time the user accessed the system;
- Search query;
- List of keywords entered by the user.

A download event is characterized by:

- Session ID;
- User name;
- Date and time the user accessed the system;
- Source name;
- Download query.

3.3.2 Source Identification

It consists in identifying the source targeted by each query. For this, patterns dictionaries are used to identify the source name according to the domain in the URL query.

3.3.3 Keywords Extraction

To extract the keywords from user's search query, the search patterns dictionary (Table 2) is used. The user query may contain special characters, such as ":","/", ".", which are encoded in URL format. For this, a URL decoding is performed to convert these characters to original format.

Example: Consider the following keywords entered by the users: **greenhouse%2C+heat+transfer**.

The characters "+" and "% 2C" are the encoding of the space and the comma respectively. After decoding these special characters, the following sentence was obtained: greenhouse, heat transfer.

4 IMPLEMENTATION

The proposed approach is implemented in a client-server architecture (Figure 4). The client accesses the application via a user account. A remote server hosts database containing very large amount of data.

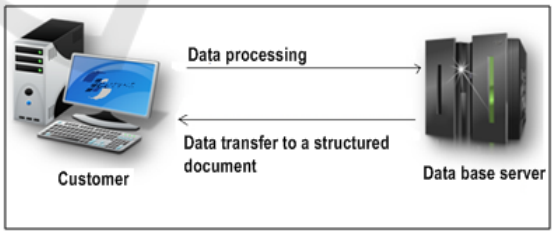

Figure 4: Client/Server Architecture.

Three main actors who interact with the system are identified, namely:

- User: can only download the extracted data;
- Administrator: analyze the log files;
- Maintainer: manages the sources and maintains the two patterns dictionaries (add a new source).

Three log files of different sizes are analyzed. Table 5 shows the amount of data before and after

cleaning process, as well as the remaining data in each file. Various others statistics can be generated, such as the number of events detected regarding error 404, image files, and JavaScript files.

Table 6 shows the results of the cleaning process, illustrating the percentage of the amount of data that was removed by conventional cleaning and deep cleaning, as well as the remaining data in each log file after the cleaning process. It can see that the proposed approach eliminates a large amount of data that is not important for the analysis which speeds up the data extraction process. Deep cleaning removes a considerable amount of unwanted data, e.g. 40% from log file 1 as shown in Table 6.

Table 5: Description of the three log files before and after the cleaning process.

Description	Log File 1	Log File 2	Log File 3
Size (MB)	62.2	140	150
Amount of data[2] before cleaning	336283	421320	412146
Amount of data after conventional cleaning	199268	286446	285412
Amount of data after deep cleaning	133575	119628	122035
Total amount of deleted data	332843	406074	407447
Amount of data remaining	3440	15246	4699

Table 6: Statistical results of the cleaning process.

Description	Log File 1	Log File 2	Log File 3
Data deleted by conventional cleaning	59%	68%	69%
Data deleted by deep cleaning	40 %	28%	30%
Remaining data	1%	4%	1%

Table 7: The extracted data from in each log file.

Description	Log File 1	Log File 2	Log File 3
Research data	2350	4150	3500
Download data	1100	10150	1100

Table 7 shows the amount of data extracted from the three files, namely research data and download data. The extracted data consists of downloaded documents and keywords used by the users, which constitute relevant information that represents the user's interests. This data is structured as Mysql data that can be downloaded as an Excel file or directly access using the SQL language.

In addition to discovering the user's interests from large log data, the proposed approach allows the preparation of relevant statistics that provide information about SNDL users, such as the most used keywords, their activities on SNDL sources, the most downloaded documents and the most requested documents (or sources). These statistics help to improve research on the SNDL system. Table 8 shows an extract of the statistical results of the most used keywords by SNDL users. Table 9 shows the frequency of access to the sources, by calculating the number of users visiting each of the SNDL sources.

Table 8: Example of most searched keywords.

Word	Number of time of search
Dirac oscillators	186
Microfluidic	135
true	90
Fault diagnisis	63
Neutron detection	60

Table 9: Sources access frequency.

Source	Number of visits
ACM	121
Aluka	8
IOP Science	103
Springer Materiels	33
IEEE	218
Science Direct	1040

5 CONCLUSION

Log files are a relevant data source that can be used in various applications, such as customizing a search system. In this paper, WUM techniques are used to analyze the user's behaviour, recorded in large log files of a multi-source search system. The proposed approach includes the preprocessing step that deletes the unwanted data and the processing step that extracts the relevant data describing the user's interests. In the pre-processing phase, a deep cleaning is proposed to remove the largest amount of irrelevant data and therefore reduce the time required for the processing phase and get accurate results. The extracted data is used to create a user profile and understand the user's activities via the generated reports.

The sources management is considered as a limitation of the proposed approach because it requires human intervention to add new sources and update the corresponding patterns dictionaries. In future, it is planned to automate the sources management through the use of learning methods.

Finally, the generated user data can be used later to form clusters, to facilitate cooperation, to emerge social networks that did not exist before, in order to

[2] Amount of data represents the number of events.

personalize the interaction with the system or to recommend the appropriate sources to the user.

REFERENCES

Astrain, J. J., Cordoba, A., Echarte, F., & Villadangos, J., 2010. An algorithm for the improvement of tag-based social interest discoery. SEMAPRO 2010 : *The Fourth International Conference on Advances in Semantic Processing*. (pp. 49-54).

Boeck, B., Huemer, D., & Min Tjoa, A., 20-23 April 2010. Towards More Trustable Log Files for Digital Forensics by Means of "Trusted Computing". *24th IEEE International Conference on Advanced Information Networking and Applications* (AINA), (pp. 1020-1027). DOI: 10.1109/AINA.2010.26

Carman, M. J., Crestani, F., Harvey, M., & Baillie, M., 2010. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and Knowledge Management*, (pp. 1849-1852). Toronto, ON, Canada.

Cooley, R., Mobasher, B., & Srivatsava, J., 3-8 Nov. 1997. Web mining: Information and pattern discovery on the World Wide Web. *9th IEEE Inernational Conference on Tools with Artificial Intelligence*. Newport Beach, (pp. 558-567). CA, USA, USA.

Cooley, R., Mobasher, B., & Srivastava, J., 1999. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, 1(1), 5-32.

Dwivedi, S., 2017. User's Activity Analysis From Weblog: Web Log Expert. *International Journal of Engineering Sciences & Research Technology (IJESRT)*, 6 (12), 547-558.

Facca, F. M., & Lanzi, P. L., 2005. Mining interesting knowledge from Weblogs: A survey. *Data Knowl. Eng*., 53(3), 225-241.

Hernández, S., Álvarez, P., JFabra, J., & Ezpeleta, J., 2017. Analysis of Users' Behavior in Structured e-Commerce Websites, *Access IEEE*, vol. 5, pp. 11941-11958.

Jaya Kumar, V., & Alagarsamy, K., 2013. Analyzing server log file using web log expert in web data mining. *International Journal of Science, Environment and Technology*, ISSN 2278-3687(O) 2(5), 1008-1016.

Kaur, N., & Aggarwal, H., 2015. A Comparative Study of WUM tools to Analyze User Behaviours Pattern from Web Log Data. *International Journal of Advances in Engineering Research*, 10 (VI).

Li, X., Guo, L., & Eric Zhao, Y., 2008. Tag-based social interest discovery. *International Conference on World Wide Web*, (pp. 675-684), Beijing, China.

Mahoto, N. A., Memon, A., Memon, M., Teevno, M. A., 2016. Extraction of Web Navigation Patterns by means of Sequential Pattern Mining. Sindh Univ. Res. *Jour. (Sci. Ser.)* 48 (1), 201-208.

Mobasher, B., Dai, H., Lou, T., Nakagawa, M., 2002. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6, 61-82.

Milicevic, A. K., Nanopoulos, A., Ivanovic, M., 2010. Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 33(3), 187-209.

Patel, K. B., Chauhan, J. A., Patel, J. D., 2011. Web Mining in E-Commerce : Pattern Discovery, Issues and Applications. *International Journal of P2P Network Trends and Technology*, 1 (3), 40-45.

Shanthi, R., & Rajagopalan, S. P., 2013. An Efficient Web Mining Algorithm To Mine Web Log Information. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(7), 1491-1500.

Shepperd E. et al., 2018. Web documentation: What is a URL?https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL. Accessed 06 December, 2018.

Singh, S. P., & Meenu, 2017. User behavior Analysis from Web Log using Web Log Expert Tools. *International Journal of Innovations & Advancement in Computer Science* IJIACS ISSN 2347 – 8616, 6 (1), 58-68.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N., 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *ACM SIGKDD* Explorations Newsletter, 1(2), 12-23.

Sujatha, V., & Punithavalli, 2012. Improved user Navigation Pattern Prediction Technique from Web Log Data. *Procedia Engineering*, 30, 92-99.

Vellingiri, J., Kaliraj, S., Satheeshkumar, S., & Parthiban, T., 2015. A Novel Approach for User Navigation Pattern Discovery and Analysis For Web Usage Mining. *Journal of Computer Science*, 11(2), 372-382.

Yang, P., Song, Y., & Ji, Y., 2015. Tag-Based User Interest Discovery Though Keywords Extraction in Social Network. Y. Wang et al. (Eds.): *BigCom* 2015, 363-372.