

# Improving the session identification using the mean time

C. E. Dinuca, D. Ciobanu

**Abstract**—In the data preprocessing, sessions identification is a very important step. Algorithms used so far to identify sessions use some fixed values to specify the end of a session and to mark the beginning of another. In this paper we explain why the use of fixed values cause errors in identifying sessions and we propose a new method for identifying sessions based on average time of visiting web pages

We implemented in Java programming language by using NetBeans IDE, two algorithms to identify sessions. The first uses a fixed value of 30 minutes (1800 seconds) to indicate the end of a session and the second using the average time spent on the pages of the website by users. For exemplification we used the NASA log file available online at <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>.

**Keywords**—Average time, Clickstream analysis, Sessions identification, Web server logs.

## I. INTRODUCTION

THE Web is the universal information space that can be accessed by companies, governments, universities students, teachers, businessmen and some users. In this universal space trading and advertising activities are held. A Web site is a set of interconnected web pages that are developed and maintained by a person or organization.

Web mining studies analyzes and reveals useful information from the Web [6]. Web mining is a term used for applying data mining techniques to Web access logs [7]. Data mining is a non-trivial process of extracting previously unknown and potentially useful knowledge from large databases [8].

Web mining can be divided into three categories: Web content mining, Web structure mining and Web usage mining [5]. Web content mining is the process of extracting knowledge from documents and content description. Web structure mining is the process of obtaining knowledge from the organization of the Web and the links between Web pages.

Web usage mining analyzes information about website pages that were visited which are saved in the log files of Internet servers to discover the previously unknown and potentially interesting patterns useful in the future. Web usage mining is described as applying data mining techniques on Web access logs to optimize web site for users.

Manuscript received November 28, 2011; Revised version received November 28, 2011.

C. E. Dinuca is PhD Student at the University of Craiova, Craiova, 200585, Romania (e-mail: clauely4u@yahoo.com).

D. Ciobanu is PhD Student at the University of Craiova, Craiova, 200585, Romania (e-mail: ciobanubedumitru@yahoo.com).

Data mining is thus materialized by applying algorithms to extract patterns from data. Additional steps of the process of discovering knowledge from data such as data preparation, data selection, cleaning phase, the integration of previous knowledge required are in fact an essential step to ensure that will extract useful knowledge from data.

There are two fundamental classes of learning methods:

- *predictive* (based on supervised learning), which uses a set of variables (called predictors) through which predictions are made relative to the values (continuous or discrete) of other variables (called decision variables);
- *descriptive* (based on unsupervised learning), for extraction of patterns (structures understandable) of data.

Predictive models are built based on artificial intelligence in a training phase, in which the model learns to predict the right answer (decision) when the input values is formed with different sets of predictors. After consuming training phase, prediction model can be used to solve, as applicable to classification problems (if the decision variable is nominal or discrete) or regression problems (if the decision variable is continuous).

Descriptive data mining methods form the second largest category of data mining. Unlike predictive models, in descriptive methods (such as clustering) the variables are treated uniformly, without distinguishing between predictors and response (decision) as such is not supervised learning (in terms of learning from examples, that of providing responses in the training phase). Descriptive methods allow the description and explanation of the characteristic phenomena of the system studied based on the patterns found.

Click-stream means a sequence of Web pages viewed by a user; pages are displayed one by one on a row at a time. Analysis of clicks is the process of extracting knowledge from web logs. This analysis involves first the step of data preprocessing and then applying data mining techniques. Data preprocessing involves data extraction, cleaning and filtration followed by identification of their sessions.

## II. DATA PREPROCESSING

### A. Actions done for preprocessing

Log files are created by web servers and filled with information about user requests on a particular Web site. They may contain information about: domains, subdomains and host names; resources requested by the user, time of request,

protocol used, errors returned by the server, the page size for successful requests.

Because a successful analysis is based on accurate information and quality data, preprocessing plays an important role.

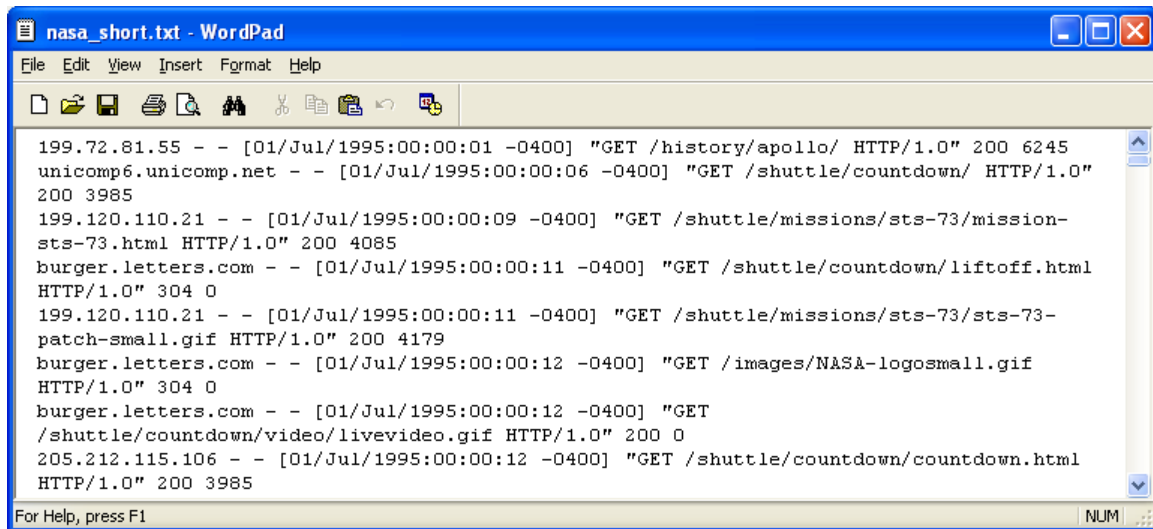
Preparation of the data requires between 60 and 90% of the time necessary for data analysis and contribute to the success rate of 75-90% to the entire process of extracting knowledge [14].

For each IP or DNS determine user sessions. The log files have entries like these:

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

As can be noticed above, each record in the file contain: IP, date and time, protocol, page views, error code, number of bytes transferred.

In Fig. 1. is shown a part of a file with logs. This type of files represent the input for our program.



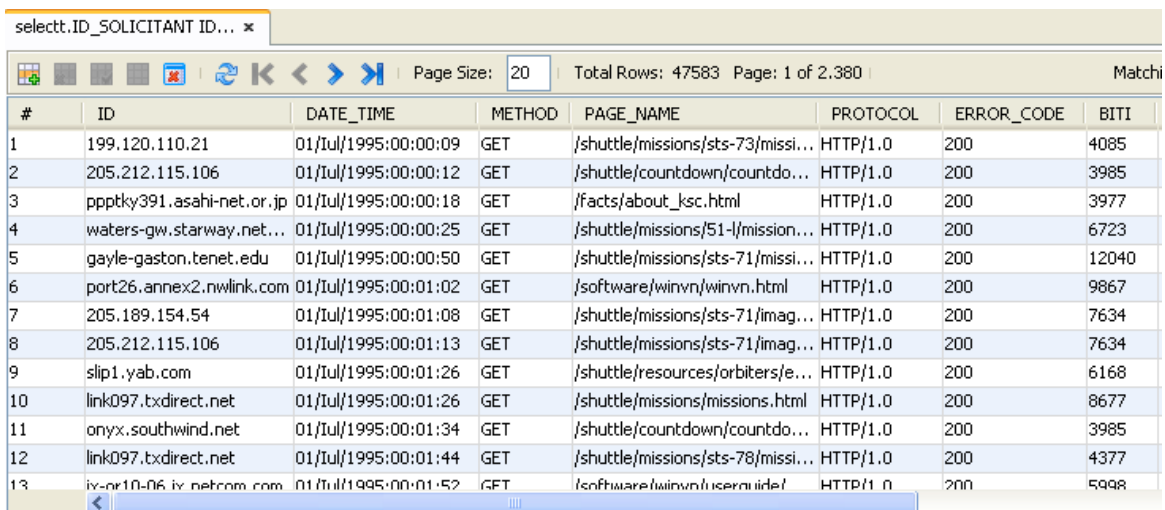
```
nasa_short.txt - WordPad
File Edit View Insert Format Help
199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245
unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0"
200 3985
199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-
sts-73.html HTTP/1.0" 200 4085
burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html
HTTP/1.0" 304 0
199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-
patch-small.gif HTTP/1.0" 200 4179
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif
HTTP/1.0" 304 0
burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET
/shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0
205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html
HTTP/1.0" 200 3985
For Help, press F1 NUM
```

Fig. 1. A text file with logs.

The program reads line by line the text file and use existing string handling functions in Java to split the row into variables and store them into a table.

This will remove the elements that separates fields within a single log record, we remove "--", "-", ":", "]", "[", and quote. Using

the method to separate strings, the following fields are saved in the database: remote host (IP or DNS address of your computer), date and time, HTTP request, status code, the volume of bits transferred.



#	ID	DATE_TIME	METHOD	PAGE_NAME	PROTOCOL	ERROR_CODE	BITI
1	199.120.110.21	01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/missi...	HTTP/1.0	200	4085
2	205.212.115.106	01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985
3	ppptky391.asahi-net.or.jp	01/Jul/1995:00:00:18	GET	/facts/about_ksc.html	HTTP/1.0	200	3977
4	waters-gw.starway.net...	01/Jul/1995:00:00:25	GET	/shuttle/missions/51-l/mission...	HTTP/1.0	200	6723
5	gayle-gaston.tenet.edu	01/Jul/1995:00:00:50	GET	/shuttle/missions/sts-71/missi...	HTTP/1.0	200	12040
6	port26.annex2.nwlink.com	01/Jul/1995:00:01:02	GET	/software/winvni/winvni.html	HTTP/1.0	200	9867
7	205.189.154.54	01/Jul/1995:00:01:08	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634
8	205.212.115.106	01/Jul/1995:00:01:13	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634
9	slip1.yab.com	01/Jul/1995:00:01:26	GET	/shuttle/resources/orbiters/e...	HTTP/1.0	200	6168
10	link097.txdirect.net	01/Jul/1995:00:01:26	GET	/shuttle/missions/missions.html	HTTP/1.0	200	8677
11	onyx.southwind.net	01/Jul/1995:00:01:34	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985
12	link097.txdirect.net	01/Jul/1995:00:01:44	GET	/shuttle/missions/sts-78/missi...	HTTP/1.0	200	4377
13	ix-rr10-06.ix.netcom.com	01/Jul/1995:00:01:52	GET	/software/winvni/userguide/	HTTP/1.0	200	5998

Fig. 2. The table with logs entries.

When the user requests to view a Web page it results more records in the log file as there are loaded graphics and additional scripts to HTML file [21]. Since the main interest of clickstream analysis is to extract patterns of user behavior, it

makes no sense to include in the review pages that were not explicitly required by the user. In this respect, it will remove all entries with the type extensions: gif, GIF, JPEG, JPEG, JPG. There are four classes of status codes: success (200

series), redirect (300 series), failure (400 series) and state error (500 series) [20]. The most common failure codes are 401 - identification failed, 403 - banned from a subdirectory and 404-file not found. All entries which have different series status code different from class 200 are removed. After removing irrelevant information is obtained the log files table that can be seen in Fig. 2.

The steps needed for data preprocessing were presented in detail in [13].

For every record we calculate the timestamp as the differences in seconds between DATE\_TIME and a fixed value; in this case we choose as fixed value "01/JUL/1995:00:00:00".

We coded pages name for making easier to view the results.

### B. Used technology

To develop the current application we have used Java programming language.

Java is an object-oriented programming language. The success of the Java programming language is largely due to ability to work on multiple platforms.

Some of the Java features are [22]:

- interpreted and compiled language. Java programs are first compiled in some similar code assembly files (called byte code, engl.), then they are interpreted by the Java runtime environment platform machine instructions associated with the system.
- platform independent language. When installing Java will create a Java virtual machine that aims to translate Java byte code instruction into machine code for the current platform. These intermediate files can be copied and executed on any platform (Windows, Unix, etc.).
- object oriented language. This is the most important property of Java. Thus, Java highlights all aspects of object orientation: objects, reference parameters, encapsulation, classes, libraries, inheritance, access modifiers.

We work with Java version jdk.1.6.0.

To illustrate and follow the steps described above we have implemented a program in Java using NetBeans IDE 6.9.1.

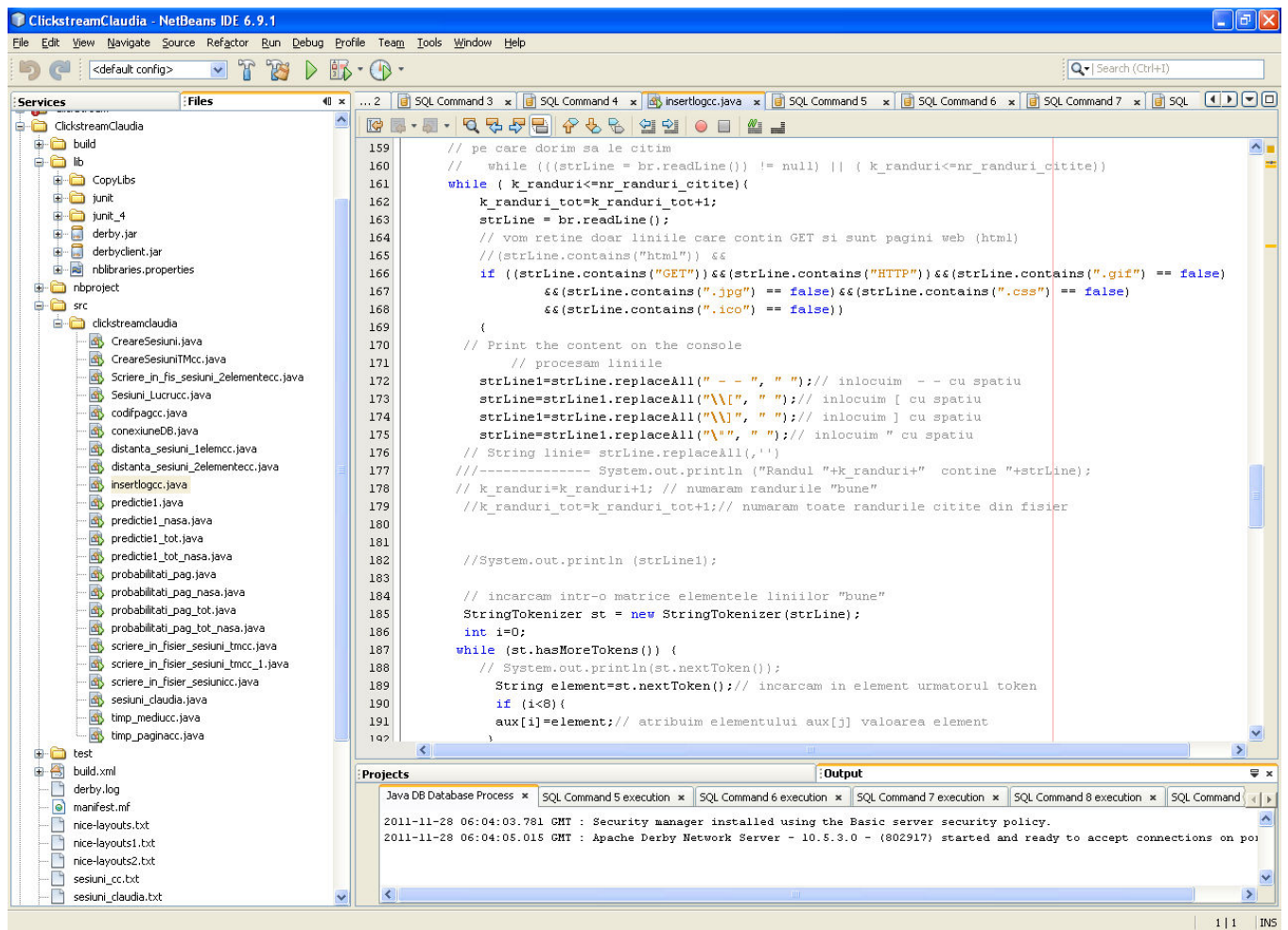


Fig. 3. Working interface for NetBeans IDE 6.9.1.

In Fig. 3 is presented NetBeans interface for creating Java package with its contents. On the right is visible a part of the packages and classes. On the left is visible ClickstreamClaudia class insertlogcc that reads the text file logs, perform

modification operations specified before and write in the table `loguricc` the records obtained, as shown in Fig. 2.

For the current application we create NetBeans project `ClickstreamClaudia`. Java allows us to work with packages. `ClickstreamClaudia` package is created that contains a set of classes implemented.

Examples of classes contained in this package are:

- `InsertLogCC` - class created for the implementation process of data preprocessing. Read the `nasa.txt` log file, line by line, it transforms data file and inserted into `logcc` table in the database;
- `CodifPagCC` - class used to encode pages, assign numbers to the pages in the database easier to apply algorithms;
- `TimpPaginãCC` - calculate the time spent on each page;
- `CreareSesiuni` - create session log file entries.

For the connection to the database is implemented a class as can be seen below.

```
public Connection dbConnection () throws Exception
```

```
{
// Path to database clickdb
    DbUrl String = "jdbc: derby: // localhost: 1527/claudia";
// User
    String user = "clau";
// Password
    String password = "clau";
// Load the driver for connection
    Class.forName ("org.apache.derby.jdbc.ClientDriver");
    Connection c = DriverManager.getConnection (dbUrl, user,
password);
return c;
}
```

In the left, Fig. 4. shows the connection to the database tables and schema `CLAU` with the component tables and in the right side a SQL select syntax together with its result.

The screenshot shows the NetBeans IDE interface. On the left, the 'Services' pane displays a tree view of the database schema 'CLAU' with various tables like LOGURI, PERECHI\_PAG, PROB, etc. The main editor window shows a SQL query: `select t.ID_SOLICITANT ID, t.DATA_ORA DATE_TIME, t.METODA 'METHOD', t.PAGINA PAGE_NAME, t.PROTOCOL PROTOCOL, t.COD_EROARE ERROR_CODE, t.NR_BITI BITI, t.'TIMESTAMP' 'TIMESTAMP', t.COD_PAGINA PAGE_CODE, t.ID_SESIUNE_TH ID_SES_TH, t.ID_SESIUNE ID_SES_30, t.TIMP_PAG TIME_PAG, t.TIMP_MEDIU_PAG TH_PAG from CLAU.LOGURICC t`. Below the query, a table displays the results of the query with columns: #, ID, DATE\_TIME, METHOD, PAGE\_NAME, PROTOCOL, ERROR\_CODE, BITI, and TIMESTAM. The table contains 13 rows of data. At the bottom, the 'Output' pane shows logs for the database process and the Apache Derby Network Server.

#	ID	DATE_TIME	METHOD	PAGE_NAME	PROTOCOL	ERROR_CODE	BITI	TIMESTAM
1	41.54.84.137	27/Jul/2011:08:09:36	GET	/wordpress-themes/category/interior-design/	HTTP/1.1	200	21371	22
2	46.70.130.108	27/Jul/2011:08:11:41	GET	/wordpress-themes/category/music/	HTTP/1.1	200	21875	22
3	67.195.113.239	27/Jul/2011:08:11:41	GET	/web-2-0-templates/tag/3d	HTTP/1.0	200	21780	22
4	49.15.174.49	27/Jul/2011:08:17:26	GET	/css-web-templates/category/politics/	HTTP/1.1	200	21736	22
5	65.52.110.47	27/Jul/2011:08:18:27	GET	/css-web-templates/category/real-estate/	HTTP/1.1	200	21535	22
6	46.73.252.1	27/Jul/2011:08:20:16	GET	/wordpress-themes/	HTTP/1.0	200	21573	22
7	190.88.108.69	27/Jul/2011:08:21:27	GET	/wordpress-themes/category/health-care/	HTTP/1.1	200	21360	22
8	114.17.194.149	27/Jul/2011:08:23:04	GET	/wordpress-themes/category/business/	HTTP/1.1	200	21876	22
9	118.97.15.21	27/Jul/2011:08:25:58	GET	/jquery-templates/	HTTP/1.1	200	21451	22
10	66.249.72.228	27/Jul/2011:08:26:18	GET	/wordpress-themes/category/health-care/	HTTP/1.1	200	21360	22
11	66.249.72.228	27/Jul/2011:08:38:01	GET	/wordpress-themes/category/music/	HTTP/1.1	200	21875	22
12	75.68.70.39	27/Jul/2011:08:38:47	GET	/wordpress-themes/category/music/	HTTP/1.1	200	21875	22
13	207.46.195.233	27/Jul/2011:08:38:59	GET	/css-web-templates/category/art/	HTTP/1.1	200	21228	22

Fig. 4. Working with database using NetBeans.

We used the facilities of working with files in Java.

The results, consisting in sessions that are obtained after preprocessing the data, are written into a file and looks like the ones shown in Fig. 5.

This type of files are used as an input for programs that execute different type of analysis.

In another work we have used files like this for predicting the next page that will be visited by a web user [23].

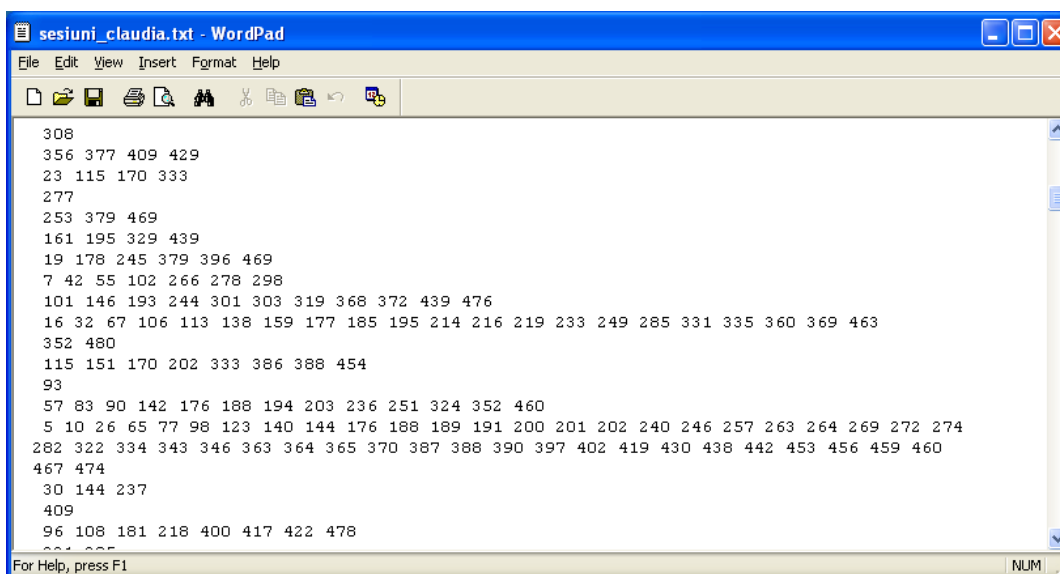


Fig. 5. A part from the file with sessions.

### III. SESSION IDENTIFICATION

For sessions' identification in the first case was considered that a user can not be stationed on a page more than 30 minutes. This value is used in several previous studies, as can be seen in the work [14].

Correct identification of sessions is an important step in preprocessing data from web logs. Some studies indicate a period of 30 minutes between pages viewed as sufficient to establish the end of a session and start another. However, this period may not be sufficient for certain types of websites, for example those which contains documents that the user reads.

Also in this category may fall and commerce sites pages which are opinions about products.

Should be taken into account that different people need different amount of time to cover the same information, for example an elderly person can slowly follow the information presented on the website. Also in the case when a potential client who wants to better inform about a product may exceed this time and the analyst wrongly consider the session ended, longer time spent on the website in this case showing interest in the product and maybe the wish to purchase the product than to leave the website. More bad decisions in sessions' identification can significantly alter the results of applying data mining techniques. In an attempt to reduce errors in session identification, an improved algorithm is proposed to amend the classic algorithm.

The current study intends to add an improvement in sessions' identification by determining an average time of page visiting the sites for the visit duration determined by analysis of web site visit duration, data which can be found in the log files of the site. Thus, for each visited page, is calculated the visit duration, which is determined by the difference between two consecutive timestamps for the same user, which is identified by IP. For records of pages with the highest timestamp among those visited by a user is assigned a predefined value of our choice to 20,000 seconds. We calculate the average visit time for a page by the average of all

the times spent on that page. When calculating the average visiting time we don't take into consideration the time less than 2 seconds and largest than 20,000 seconds.

### IV. ALGORITHMS PRESENTATION

For sessions' identification we use an algorithm which we present below. For each IP we select the visited pages sorted by timestamp. For each page is given a session identification number  $Id\_sesiune$ , and then it is checked after if the time is more than 1800 seconds, in which case we switch to a new session by increasing with one the value of  $Id\_sesiune$ .

#### Model description :

We consider the set of users' IP by  $IP = \{IP_1, IP_2, \dots, IP_n\}$ . The crowd of pages visited by the user identified by  $IP_k$ ,  $PIP_k = \{PIP_{k1}, PIP_{k2}, \dots\}$  and  $TS\_PIP_{ki}$  the timestamp of  $PIP_{ki}$  page. We note by  $ID\_PIP_{ki}$  the session identification number assigned to page  $PIP_{ki}$  page and with  $ID$  the set of these IDs.

#### The pseudo-code Algorithm

```
For each IP  $IP_k$  repeat
If  $|PIP_k|=1$  then  $ID\_PIP_{k1}=\max(ID)+1$ ;
Else  $ID\_PIP_{k1}=\max(ID)+1$ ;
 $I=1$ ;
While ( $I < |PIP_k|$ ) repeat
 $I=I+1$ ;
If  $TS\_PIP_{ki} - TS\_PIP_{ki-1} < 1800$  then  $ID\_PIP_{ki} = ID\_PIP_{ki-1}$ ;
Else  $ID\_PIP_{ki} = ID\_PIP_{ki-1} + 1$ ;
```

In the case of algorithm that uses the average time it is proceeded in the same way. For each IP we select the visited pages and sort them by timestamp. For each page it is given a session  $Id$  and then we verify if the time visiting is great than 300 seconds or more than twice the mean visiting time, in which case it is switched to a new session increasing by one the value of session  $Id$ .

#### The pseudo-code Algorithm

```
For each IP  $IP_k$  repeat
If  $|PIP_k|=1$  then  $ID\_PIP_{k1}=\max(ID)+1$ ;
Else  $ID\_PIP_{k1}=\max(ID)+1$ ;
 $I=1$ ;
```

While ( $I < |PIP_k|$ ) repeat

$I = I + 1$ ;

$ID_{PIP_{ki}} = ID_{PIP_{ki-1}}$ ;

$TMA_{ki} = \max(2 * TM_{ki}, 300)$ ;

If  $TS_{PIP_{ki}} - TS_{PIP_{ki-1}} > TMA_{ki}$  then

$ID_{PIP_{ki}} = ID_{PIP_{ki-1}} + 1$ ;

Where  $TM_{ki}$  is the average time spent by users on the page  $PIP_{ki}$  and  $TMA_{ki}$  is the time used in the modified algorithm instead of the fixed value of 1800 seconds.

Introducing the value of 300 was necessary because in the case of some pages the average time is very low of orders of tens of seconds, which can negatively influence the sessions' identification.

The Java implementation of this algorithm looks like this:

```
package clickstreamClaudia;
import java.sql.*;
public class CreareSesiuniTMcc {
public static void main(String[] args) {
String dbUrl = "jdbc:derby://localhost:1527/claudia";
String user = "clau";// the user
String password = "clau";// password
int id_sesiune=0;
int id2=0;
try//we try to load the drivers necessary for connection
{ Class.forName("org.apache.derby.jdbc.ClientDriver");}
catch(ClassNotFoundException e)
{System.out.println("Eroare incarcare driver!\n" +e);}
try// we try to establish the connection
{
Connection c = DriverManager.getConnection(dbUrl,user, password);
Statement s = c.createStatement();
ResultSet result = s.executeQuery("select distinct id_solicitant from
LOGURlcc ");
int contor=0;
int val_timp=0;
int nri=0;
int nr_pagini_dif_pe_id=0;
while (result.next()) { // process results one row at a time
String val = result.getString("id_solicitant");
if (id2==id_sesiune)
{id_sesiune++;
id2=id_sesiune;}
Statement s1 = c.createStatement();
ResultSet result1=s1.executeQuery("select cod_pagina, timestamp,timp_pag,"
+ "timp_mediu_pag from loguricc where id_solicitant = "
+ val + " order by timestamp");
int count=0;
Integer cod_prima_pagina=0;
Integer ts_prima_pagina=0;
Integer timp_mediu=0;
while (result1.next()){
count++;
Integer cod_pagina1 = result1.getInt("cod_pagina");
Integer ts2 = result1.getInt("timestamp");
timp_mediu=result1.getInt("timp_mediu_pag");
int timp_petrecut=result1.getInt("timp_pag");
Statement s2 = c.createStatement();
boolean execute = s2.execute("update loguricc set id_sesiune_tm="
+ id_sesiune
+ " where timestamp = " + ts2+" and cod_pagina=" +cod_pagina1);
s2.close();
Integer tm=0;
if (timp_petrecut>tm || (timp_petrecut>=20000))
{id_sesiune=id_sesiune+1;}
```

```
}
s1.close();
}
s.close();// inchidem comanda s
}
catch(Exception e)
{
System.err.println("Error 2 : " + e.getMessage());
}
}
```

## V. CASE STUDY

To implement the algorithms presented earlier we used the Java programming language, the code is written using NetBeans editor. We choose Java because it is an object oriented program, it is open source and platform independent. We work with Java Jdk 1.6.0. Also as the code source editor, we choose NetBeans IDE because it is open source. We work with NetBeans IDE version 6.9.1.

We used the database with logs from NASA website that can be downloaded free by accesing the link <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>. After the data preprocessing phase there were obtained 47583 records as can be seen in Fig. 6.

#	1
1	47583

Fig. 6. The total number of records from Loguri table.

After applying the two algorithms to identify sessions we obtained 16505 sessions with preset option for 1800 seconds (30 minutes) and 17178 sessions for the proposed new algorithm, the results can be seen in Fig. 7.

#	ID_SESIUNE	ID_SESIUNE_TM	TIMP_PAG	COD_PAGINA	TIMP_MEDIU_PAG	TIMESTAMP
47564	16500	17173	81	468	267	147739
47565	16500	17173	85	413	55	147820
47566	16500	17173	56	358	46	147905
47567	16500	17173	125	499	96	147961
47568	16500	17173	67	405	27	148086
47569	16500	17173	68	172	62	148153
47570	16500	17173	123	294	212	148221
47571	16500	17173	104	443	99	148344
47572	16500	17173	126	33	409	148448
47573	16500	17173	104	129	481	148574
47574	16500	17173	3225	293	1427	148678
47575	16501	17174	38273	302	1325	151903
47576	16502	17175	437	129	481	190176
47577	16502	17175	120	222	433	190613
47578	16502	17175	20000	449	252	190733
47579	16503	17176	20000	129	481	185097
47580	16504	17177	52	481	332	7749
47581	16504	17177	20000	286	247	7801
47582	16505	17178	70	129	481	4408
47583	16505	17178	20000	129	481	4478

Fig. 7. A part of the content of Loguri table

Fig. 8. shows a case in which using the average time the pages are allocated to the same session and in the classic case the pages are in different sessions.

#	ID_SESIUNE	ID_SESIUNE_TM	TIMP_PAG	COD_PAGINA	TIMP_MEDIU_PAG	TIMESTAMP
47121	16322	16989	201	313	628	59811
47122	16322	16989	18	334	487	60012
47123	16322	16989	31	224	209	60030
47124	16322	16989	254	362	281	60061
47125	16322	16989	1208	225	640	60315
47126	16322	16989	171044	399	1401	61523
47127	16323	16990	2831	207	1569	232567
47128	16324	16990	20000	301	463	235398
47129	16325	16991	2602	72	2018	558
47130	16326	16991	21417	129	481	3160
47131	16327	16992	56	481	332	24577
47132	16327	16992	8030	335	902	24633
47133	16328	16993	55149	207	1569	32663
47134	16329	16994	74967	302	1325	87812
47135	16330	16995	8728	107	577	162770

Fig. 8. An example of wrong session split using 30 minutes for session identification.

Fig. 9. presents a case in which pages are allocated in two sessions using the average time and in the classic case the pages are in the same session.

#	ID_SESIUNE	ID_SESIUNE_TM	TIMP_PAG	COD_PAGINA	TIMP_MEDIU_PAG	TIMESTAMP
11051	3936	4110	158	462	470	156990
11052	3936	4110	74	362	281	157148
11053	3936	4110	428	97	414	157222
11054	3936	4110	192	207	1569	157650
11055	3936	4110	138	449	252	157842
11056	3936	4110	111	73	833	157980
11057	3936	4110	446	228	314	158091
11058	3936	4110	35	301	463	158537
11059	3936	4110	247	486	231	158572
11060	3936	4110	58	161	1053	158819
11061	3936	4110	242	215	463	158877
11062	3936	4110	256	336	521	159119
11063	3936	4110	735	294	212	159375
11064	3936	4111	28	297	43	160110
11065	3936	4111	48	373	276	160138
11066	3936	4111	53	280	2163	160186
11067	3936	4111	36	9	72	160239
11068	3936	4111	20000	202	128	160275
11069	3937	4112	20000	302	1325	215921
11070	3938	4113	20000	92	877	41991

Fig. 9. Another example of wrong session' identification using 30 minutes for session end.

We ran the program for several combinations of mean time and fixed values.

	1800	TM	TM+300	TM+600	TM+900	TM+1200	TM+1500	TM+1800
number of sessions	16467	17326	17141	16970	16675	16571	16389	16254
number of sessions of length 1	7773	8212	8144	8062	7934	7805	7695	7607
number of sessions of length 2	3238	3454	3406	3358	3331	3285	3249	3214
number of sessions of length >2	5456	5660	5591	5550	5410	5481	5445	5433

Fig. 10. Number of session obtained using mean time and different fixed values.

Although the number of sessions with length greater than 2, that we take into consideration, not vary greatly, the structure of sessions is different.

VI. CONCLUSIONS

In the case of modified algorithm other conditions can be chosen to determine with increased precision the separation

between sessions. The number of sessions for the modified algorithm is greater than in the case of classic algorithm. To avoid a division in too many sessions we can increase the value of 300 seconds used in the modified algorithm case. In Fig. 10. is shown the variation of the number of session when we modify the fixed time used.

In the case of the modified algorithm the average visiting time depends on the page, so we can say that using this algorithm to separate sessions better maps the reality than using a single constant value for the algorithm's implementation. Also, for sites running in different areas using the mean time is recommended because it depends directly on the site structure and content pages. The modified algorithm has the same running time as a classical algorithm which is another reason that recommend its use, so it's complexity is not modified.

## REFERENCES

- [1] J. Srivastava, R. Cooley, M. Deshpande, P. -N. Tan, Web usage mining: discovery and applications of usage patterns from web data, SIGKDD Explorations, 1(2), 2000, pp. 12-23.
- [2] B. Mobasher, R. Cooley, J. Srivastava, Creating Adaptive Web Sites through usage based clustering of URLs, IEEE knowledge & Data Engg work shop (KDEX'99), 1999.
- [3] B. Brendt, M. Spiliopoulou, Analysis of Navigation Behaviour in Web Sites Integrating Multiple Information Systems. VLDB, 9(1), 2000, pp. 56-75.
- [4] R. Kohavi, R. Parekh, Ten supplementary analysis to improve e-commerce web sites, Proceedings of the Fifth WEBKDD workshop, 2003.
- [5] O. Zaiane, J. Han, WebML: Querying the World Wide Web for resources and knowledge. In: Workshop on Web Information and Data Management WIDM98, Bethesda, 1998, pp. 9-12.
- [6] Cooley R., Mobasher B., Srivastava J.: Web mining: Information and Pattern Discovery on the World Wide Web. A survey paper. In: Proc. ICTAI-97, 1997.
- [7] Zaiane O.: Conference Tutorial Notes: Web Mining: Concepts, Practices and Research. In: Proc. SDBD-2000, 2000, pp. 410-474.
- [8] Piatetsky-Shapiro g., Fayyad U., Smith P., Uthurusamy R.: Advances in Knowledge Discovery and Data Mining., AAAI/MIT Press, 1996.
- [9] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, Springer Berlin Heidelberg New York, 2006.
- [10] B. Hay, W. Geert, V. Koen, Discovering interesting navigations on a web site using SAM<sup>1</sup>, Springer-Verlag Berlin, 2005.
- [11] T. R. Li, Y. Xu, D. Ruan, W. M. Pan, Sequential Pattern Mining, Springer-Verlag Berlin, 2005.
- [12] L. Clark, I. Ting, C. Kimble, P. Wrigth, D. Kudenko, Combining Ethnographic and Clickstream Data to Identify Strategies Information Research 11(2), paper 249, 2006.
- [13] E. C. Dinucă, The process of data preprocessing for Web Usage Data Mining through a complete example, Annals of the "Ovidius" University, Economic Sciences Series, Volume XI, Issue 1 /2011.
- [14] M. Zdravco, D. T. Larose, Data Minig the Web – Uncovering Patterns in Web Content, Structure and Usage, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.
- [15] W. Taowei, R. Yibo, Research on Personalized Recommendation Based on Web Usage Mining Using Collaborative Filtering Technique, WSEAS Transactions on Information Science and Applications, Vol. 6, No. 1, pp. 62-72, 2009.
- [16] G. Castellano, A. M. Fanelli, M. A. Torsello, Understanding Visitor Behaviors from Web Log Data, WSEAS Transactions on Computer Research, Vol. 2, No. 2, pp. 277-284, 2007.
- [17] G. Castellano, A. M. Fanelli, M. A. Torsello, LODAP: a log data preprocessor for mining web browsing patterns, Proceedings of the 6th Conference on 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, pp.12-17, 2007.
- [18] G. Castellano, A. M. Fanelli, M. A. Torsello, Mining usage profiles from access data using fuzzy clustering, Proceedings of the 6th WSEAS International Conference on SIMULATION, MODELLING AND OPTIMIZATION (SMO '06), pp. 157-160, 2006.
- [19] A. M. Yahya, MD. B. S. Nasir, M. Norwati, I. U. Nur, M. Zaiton, ARS: Web Page Recommendation System for Anonymous Users Based On Web Usage Mining, Proceedings of the WSEAS European conference of systems, and European conference of circuits technology and devices, and European conference of communications, and European conference on Computer science, pp. 115-120, 2010.
- [20] B. T. Sandjay, G. Sangram, A Effective and Complete Preprocessing for Web Usage Mining, International Journal on Computer Science and Engineering, Vol. 02, Nr. 03, pp. 848-851, 2010.
- [21] R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining world wide Web browsing patterns, Journal of Knowledge and Information Systems, 1, pp. 5-32, 1999.
- [22] S. Tănasă, C. Olaru, S. Andrei, *Java from 0 to Expert(in romanian)*, Romania, Ed. Polirom, 2003.
- [23] C. E. Dinucă, D. Ciobanu, *Improving the prediction of next page request by a web user using Page Rank algorithm*, Proceedings of the 1st WSEAS International Conference on Tourism and Economic Development (TED '11), Drobeta Turnu Severin, Romania, pp. 520-524, 2011.