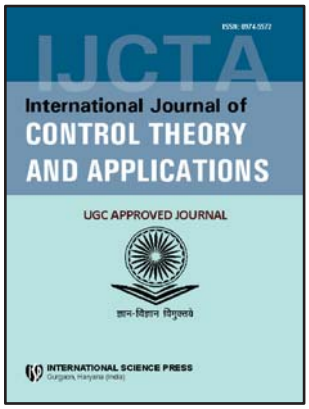

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 31 • 2017

Efficient Data Cleaning Algorithm and Innovative Unique user Identification Algorithm using Hashing and Binary Search Techniques for Web Usage Mining

Ranjena Sriram[a] and S. Sheeja[b]

[a]*Research Scholar, Karpagam University, Coimbatore-641021, Tamil Nadu, India*

E-mail: ranjenasriram@gmail.com

[b]*Associate Professor & Head, Dept of Computer Applications, Karpagam University, Coimbatore, Tamil Nadu, India*

E-mail: sheejaajize@gmail.com

Abstract: This current study focuses on proposing a new pre-processing and unique user identification algorithms for Web Usage Mining to discover and analyse the user's access pattern through mining of log files or log databases and the associated data from a particular website. Pre-processing technique is to clean the data and user identification process to identify unique users. Since number of users interacting with web sites around the world is increasing day by day, the amount of data generated and information gathered could help the organizations to improve their business according to the customers' needs and behavior. This work comes out with an innovative pre-processing technique which uses Generalized Sequence Pattern Algorithm to find the irrelevant data and two new strategies, one to group IP addresses zone wise in separate Hash buckets from Web Log Server file and the other is usage of Binary Search techniques in the proposed User Identification Algorithm to minimize the searching time of unique users. In addition this work fine tunes the previously proposed Hashing function further to improve the performance of Distinct User identification for Web Usage Mining. The proposed pre-processing algorithm and User Identification Algorithm is evaluated by comparing with existing algorithms to prove its accuracy and efficiency. Similarly the modified Hashing techniques is compared with previously proposed Hashing function and existing searching methodologies and it has been proved that the modified Hashing function is quick in searching according to Big O notation. Web Log Server data from renowned Universities like Murdoch from U.A.E, Enmities College of Management and Information Technology from United Arab Emirates and Nehru Arts and Science College from India are used to evaluate the performance of the proposed pre-processing technique and modified Unique User Identification algorithm.

Keywords: *Web Usage Mining, Hashing Techniques, UUI (Unique User Identification).*

1. INTRODUCTION

Internet has become an important source of information for many users around the world. Evolutions take place and we are in the world of accessing robust, versatile and aesthetic websites for day to day transactions. Web servers play an important role in mining these transactions by various ways. The behavior of the Web site can be evaluated from the information stored in Web Log File generated by Web Log Server. Much useful

information like type of users, access time, and number of page hits etc. guides and helps the site developers to an extent to evaluate and thereby do some useful modifications, which can attract more number of visitors to the site. Tragically many web site developers are not focusing on their Web Server Logs to study the performance of their sites. Web Log Server stores huge volume of data and in order to derive useful patterns using mining strategies irrelevant data has to be removed or cleaned. To overcome these problems this work comes out with two important strategies.

1. An efficient algorithm to clean web log data
2. A unique algorithm using Hashing and Binary Search techniques to identify unique users.

The study proposes a fast active distinct user identification algorithm which uses a Hashing technique blended with an IP address and a finite user's inactive time to identify different users in the web log file. Though man works are proposed and implemented, none produced a better quality in the results produced when the data size increases in the Web Server. Experimental results prove that the algorithms proposed in this work shows better results for Web Servers with huge data size. The results also prove the generalized behavior of the algorithms for different Web Serves with different data and attribute information.

2. WEB USAGE MINING

Web Usage mining is the application web Data Mining Techniques to discover usage patterns from Web to understand the better needs of web based applications. IT tries to make use of the information regarding the web surfer's session behavior. The web content and web structure make use of primary data; web usage mining uses secondary data mainly from Web Log Server to understand the user's patterns, user interactions with the Web Server. It analyses the results of user interactions with web server, including weblogs, click streams, and database transactions at a web site of a group of related sites [1].

Web Usage Mining is a three phase process consisting of :

1. **Pre-processing / Data Preparation :** Proper dataset is selected to derive meaningful patterns after mining process. Getting relevant data in today's world from different sources is very difficult due to various reasons. Results generated from irrelevant data is not of much accurate and efficient, hence some processing is done to get accurate data for mining. The data preparation step is most time consuming and difficult process in any mining process. It requires versatile and robotic algorithms with heuristics approach not common to other domains. This process may involve pre-processing the original data, integrating from different sources. And transforming the integrated data into a form suitable for input into specific Data Mining operations collectively called as pre-processing preparation [2].

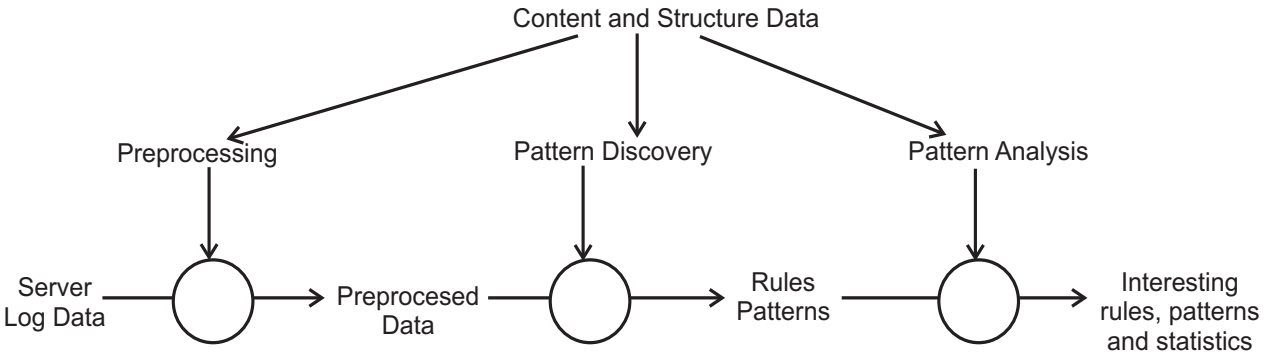

Figure 1: Process of Web Usage Mining

2. **Pattern Discovery:** Meaningful patterns are identified using Statistical, Data mining, Associate rule and Sequential methods. Classification are made simple and faster due to these techniques Statistical methods, Data mining methods, Associate rule, Sequential methods and cluster techniques are used to identify unique patterns.
3. **Pattern Analysis:** The patterns identified and discovered are analyzed using OLAP tools, query management and intelligent smart agent based systems to remove irrelevant data, rules or patterns.

3. SOURCES AND TYPES OF DATA

The major data source for Web Usage Mining is the server log files, which include web server access and application server logs. Apart from this information, additional data sources are essential for data preparation and pattern discovery, which include site files, Meta data, operational databases, application template and vast domain knowledge. In some cases data from client side, proxy level data collection (Internet Service Provider), and demographic data sources provided by data aggregation services are used for huge mining systems.

The data obtained from various sources can categorized into four primary groups.

1. **Usage Data:** The log data collected automatically from the Web Log Server serves as the primary source for Web Usage Mining. Each hit on the Web Server corresponding to the HTTP request, generates a single entry in the server log. Each log entry contains fields identifying the time and date of request, the IP address of the client, method used, the user agent (browser and operating system type, version), the referring Web resource and if available client side cookies which identify a repeat visitor. Depending on the depth of the analysis, the data is transformed and aggregated at different levels of abstraction.
2. **Content Data:** Collection of objects and relations conveyed to the users. Most of the data are in the format of textual and images which are generated from static HTML/XML pages and multimedia files, dynamically generated page segments from scripts and collection of records from operational databases, content in form of Meta data, document attributes descriptor keywords, semantic tags and HTTP variables. Domain ontology such as conceptual hierarchies, product categories, explicit representation of semantic content and relationship via an ontology language such as RDF, or a database schema over the data contained in the operational database.
3. **User Data:** Data collected from operational databases which include demographic information about registered users, user ratings on various objects such as product or movies, past purchases or visit histories of users as well as other explicit or implicit representations of user's interest.

4. RELATED WORK

The study focuses on designing a modified and efficient preprocessing technique in the form of an algorithm, which groups the data based on the elements of the log record. The advantages of the modified algorithm are discussed in later section. Also the work proposes an innovative Unique User Identification algorithm which uses Hashing and Binary Search techniques to locate the user in limited time. An algorithm is proposed to group the IP addresses based on their zones before executing the proposed UUI algorithm.

4.1. Data Cleaning

The principle of Data Cleaning is to remove or reduce extraneous data. The following data is removed.

1. Records containing video, graphics and file extensions of GIF, JPEG and CSS.
2. The log records with status codes over 299 or fewer than 200.
3. Records having value of POST or HEAD.
4. User agents like Crawler, Spider or Robot.

4.2. Proposed Algorithm for Data Cleaning

The generalized approach followed by earlier strategies can't be applied in real time scenarios, which are handling huge volume of data and where time is an important criterion. Considering these prevailing conditions this study introduces an innovative algorithm which follows Generalized Pattern Sequence methodology to check for irrelevant data for Web Usage Mining. Since Generalized Pattern Sequence is a mining algorithm, only the salient features are extracted for pre-processing technique.

Generally from the Web Log record the major issues considered as irrelevant data are discussed above in the generalized algorithm. According to the modified pre-processing algorithm, different groups are created for different conditions like groups for image file extensions, methods and user agents. The input log record is fragmented and the fragments are simultaneously compared with the groups, if either one matches then the record is invalid or considered as irrelevant record and can be eliminated. The algorithm is explained below in detail.

4.3. Proposed Data Cleaning Algorithm using Generalized Pattern Sequence methodology

Three sequences taken for the algorithm

1. File extensions like (*css, jpeg, jpg, js, gif*)
2. Methods (GET, POST)
3. Site Status (301,404,500)
4. User Agents.

Input: Web server Log File

Step 1: Let F be the different Groups

$$k = 2$$

Step 2: Read Log Record from Web Server Log File

Step 3: Fragment Log Record into different elements *fr*.

Step 4: Do while (F_{k-1} != Group Count)

Step 5: Let (*a*) denote individual fragments in Group F_k

For all input fragments from Log Record r in Log file (or) Database D

Step 6: If (*a*) matches (*fr*) then

Step 7: Move the record to the corresponding Group and eliminate the record from Log Database

Else move to next group

$$k = k + 1$$

else

Consider the fragment as outlier.

End if

Step 8: Repeat until eof

End do

Execution of the algorithm

1. Input Log Record from log File
2. Generate different Groups
3. Read Log Record from Log File and repeat until end of file.

4. Fragment the Log Record into individual elements
5. Compare each element in the groups with the input element from Log File
6. If matches move the element to the individual group else move to next group.
7. Eliminate the record from Log File
8. Repeat the process until all groups are visited.

4.4. Advantages

1. Searching time minimizes since the given element from the log record is parallel checked in all groups.
2. Efficient and quick when comparing with other techniques.

5. USER IDENTIFICATION

User's identification is, to categorize who access web site and which pages are accessed. Different users may have same IP address in the log. A referrer-based method is proposed to solve these problems in this study.

The rules adopted to distinguish user sessions can be described as follows:

1. Each IP address represents one user;
2. For more logs, if the IP address is the same, but the agent log shows a change in browser software or operating system, an IP address represents a different user
3. Using the access log in conjunction with the referrer logs and site topology to construct browsing paths for each user. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, there is another user with the same IP address.

This work comes out with an innovative Unique User Identification algorithm using Hashing techniques to locate the user in quick manner, though it is efficient it has its own drawbacks which is modified and proposed as modified algorithm which uses grouping of similar zone Ip's followed by Hashing and Binary Search techniques to locate the user more faster when comparing with this UUI algorithm.

5.1. Proposed Unique User Identification Algorithm Using Hashing Technique

Unique user identification is important process next to data cleaning. Unique users are identified based on the rules suggested in User Identification section. Though many efficient algorithms are there, many fail in accuracy and efficiency (time taken to identify users) when the size of the Log Database increases. Today's modern web servers are capable of handling terabytes of data conventional algorithms are obsolete in handling these scenarios. Considering the above facts, this study proposes an efficient Unique User Identification algorithm that uses modern Hashing techniques to identify unique user quickly inspire the huge size of the database. A new hashing key is prosed and successfully implemented in the algorithm to locate the user [7].

5.1.1. Hashing Techniques

For a huge database structure, it can be almost next to impossible to search all the index values through all its level and then reach the destination data block to retrieve the desired data. Hashing is an effective technique to calculate the direct location of a data record on the disk without using index structure.

Hashing uses hash functions with search keys as parameters to generate the address of a data record [5].

5.1.2. Hash Organization

Generally a hash stores data in the form of a bucket, a bucket is a representative of a storage block which stores one complete disk block, which in turn stores record groups. Searching in Hash table is done by a Hash Function which maps all set of sear keys (K) to the address where the actual records are placed. It is a function from search key to bucket addresses [5].

5.1.3. Dynamic Hashing

The problem with static hashing is that it does not expand or shrink dynamically as the size of the database grows or shrinks. Dynamic hashing provides a mechanism in which data buckets are added and removed dynamically and on-demand. Dynamic hashing is also known as **extended hashing** [6].

Hash function, in dynamic hashing, is made to produce a large number of values and only a few are used initially.

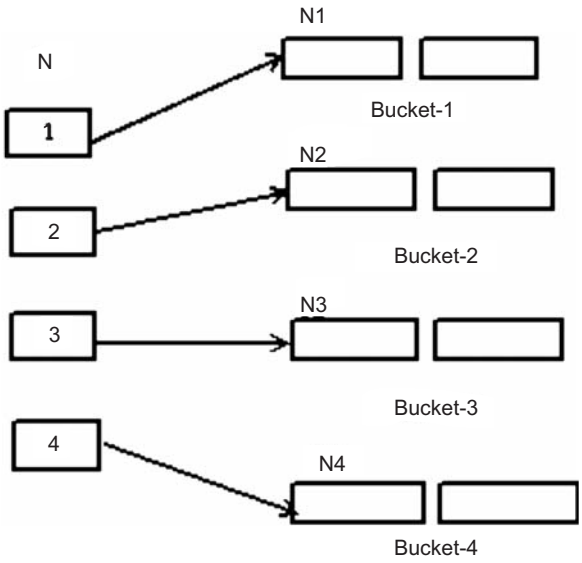

Figure 2: Dynamic Hashing Technique

Generally any Unique User Identification algorithm analyses more factors like users IP addresses, web site topology, browser edition and operating system. The proposed algorithm not only uses IP addresses but also identifies user's session. The proposed algorithm not only uses User IP address, but also based on path chosen by any user, access time with the referred page etc. [4].

When huge databases are taken for considering the time taken to locate the records is much, hence appropriate methodology is incorporated to make the process faster. Taking these prevailing conditions, the study proposes a new Hashing formulation, to minimize the searching time for large datasets. Previous study proposed a Hashing function, which is quick enough to search the unique user's IP address, but when the size of the bucket increases certain pre-processing is done to fasten the searching time of unique user. On considering the prevailing issues, this work substitutes Binary search techniques to minimize the searching time.

A few modifications are done the previously proposed Hashing function to make it generalized and quick in searching patterns.

5.1.4. Proposed Hash Function

$$Nmod_2 * K + d \quad (1)$$

Where N refers the record number indirectly pointing the data an IP address or an Operating system or a browser, (K) refers to the virtual address of the bucket and d refers to the displacement distance. The multiplied factor gives the original location of the data [6].

Substitute $Nmod_2$ with parameter H equation (1.1) becomes

$$H(K) \quad (2)$$

5.1.5. Drawbacks

1. Takes more time to locate the required user.
2. Some pre-processing required minimizing the searching time.
3. Takes user records (IP addresses) as such without splitting zone wise hence more time in identifying users.

   Considering these drawbacks the Hash function specified in equation (1) is generalized. The generalization is done by the following steps.

5.2. Proposed Unique User Identification Algorithm

Web Log server contains accumulated log information, which makes the searching more complicated. In order to minimize the searching time this work includes two main strategies.

1. An algorithm is designed and developed to group IP addresses of similar zones in individual Hash buckets from the Web Log Server file, which minimizes the searching time to a great extent.
2. Binary Search techniques are used along with some string manipulations in the previously proposed UUI algorithm to reduce the searching time.

   Different sets of IP ranges are allocated to particular networks, geographic areas, companies etc. The table below shows several examples of IP ranges and their implementations.

5.3. First Strategy

Table 1
IP addresses of different zones

IP Range	*Description*	*Example*
192.168...	Private Networks	192.168.1.23
172.16... ... 172.31...		
10...		
41...	AfriNIC allocation of IP	102.43.1.65
102...	addresses in Africa	
105...		
81...	European allocation of IP	81.202.17.89
217...	address	
62...		
200	Latin American and	200.100.50.25
	Caribbean	
9	IBM	9.1.2.3
17	Apple	17.19.29.23

Since different zones start with different IP address the IP address of the given user is searched must be differentiated and grouped to their specific zones in separate Hash buckets assigned for different IP zones from other zones of IP addresses. The algorithm explained below shows how specific IP address from a particular zone is extracted and stored in separate Hash bucket in the form of array. This tactic reduces the searching time to much extent which is shown in the result and discussions section.

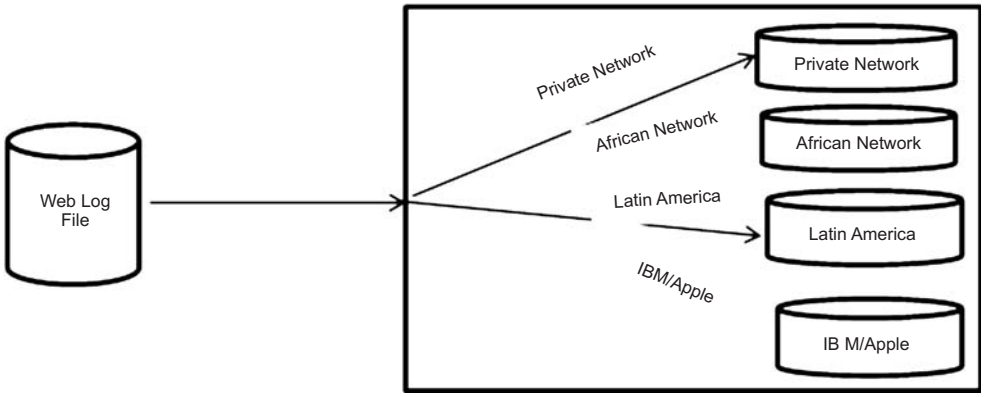

Figure 3: Grouping of IP addresses from Web Log Server in separate Hash buckets using prosed SetIp algorithm

SetIp Algorithm to group IP zone wise from Web Log Server File

```
PN Private Network, AN  – African Network, EN European Network, LA Latin America,  AP = Apple
Networks
Definition: given clean and filtered web log file and record set web log file

Records R – (r1, r2, r3 .... r.n)
Initialize array PN, AN, EN, LA, AP, IBM, GetIP
where n > 0
Step1:   Input Log database RUser of N records
Step2:   Distinct User identification base
Step3:   RUser=Purl, ip_addr, agent, method, operating system, status session id, time_stamp
Step4:   RUser=r1, r2, r3...rn where n!-0,i-0
Step5:   while (i=n)
Step6:   GetIp(i) = substr(R(i),03 // Obtain first three digits of IP Adress
Step7:   SetIp(i) = substr(R(i),-3,3 // Obtain last three digits of IP Adress
Step8:   While (Log database<>eof)
           switch(GetIP(i))
      Begin
          Case 192:          Store the IP address to Private Networks Array PN
                             Assign the PN Array to Hash Storage.
                             Break.
          Case 41, 102, 105: Store the IP address to African address Array AN
                             Assign the AN Array to Hash Storage
                             Break.
          Case 81, 217, 62: Store the IP address to European IP addresses Array EN
                             Assign the EN Array to Hash Storage
                             Break.
          Case 200:          Store the IP address to Latin American IP addresses Array LA
                             Assign the LA Array to Hash Storage
                             Break.
           Case 17:          Store the IP address to Apple IP addresses Array AP
                             Assign the AP Array to Hash Storage
End Switch
Step9: i = i + 1;
Step10: Return IP address
```

Figure 4: Algorithm GETIP to extract IP address of Unique User

5.4. Execution of SetIp Algorithm

1. Read records one by one until end of file
2. Get first three digits from the each record.
3. Match with the case statement.
4. If the record matches either of the case statement, store the IP in either of the arrays assigned to each zone.

5.5. Second Strategy

Now this arrangement facilitates the modified UUI algorithm to search whether the IP of the user exists or not quickly. In order to achieve this task some modifications are done to the previously proposed UUI algorithm.

1. Binary Search techniques are used along with some string manipulations in the previously proposed UUI algorithm to reduce the searching time.

```
Unique User Identification (UUI)
Definition: Given a clean and filtered web log file and record set web log file
Records R = {r1, r2, r3 .... r.n}
where n > 0
Step1: Input Log database RUser of N records
Step1: Distinct User identification base
Step3: DUser = P<url, ip_addr, agent, method, operating system, status, session id, time_stamp>
Step4: RUser=<r1,r2,r3...m> where n! = 0, i = 0
Step5: While(i > n)
Step6: GetIP(i) = substr(R(i),0,3 // Obtain first three digits of IP Address
Step6: While (Log database<>eof)
Step7: Read Log database RUser
Step8: Switch (GetIP(i))
    Begin
      Case 192:          Binary Search(PN, R(i) *(Rmod2 *K(i) + d)).
                         If found extract user information Else // Existing User
                         Store the IP address to Private Networks Array PN // New User
                         Assign the PN Array to Hash Storage.
                         End if
                         Break.
      Case 41,102,105:  Binary Search(AN, R(i) *(Rmod2 *K(i) + d)).
                         If found extract user information Else
                         Store the IP address to African address Array AN
                         End if
                         Assign the AN Array  to  Hash Storage.
                         Break.
      Case 81,217,62:   Binary Search(EN, R(i) *(Rmod2 *K(i) + d)).
                         If found extract user information Else
                         Store the IP address to European IP addresses Array EN
                         Assign the EN Array to Hash Storage.
                         End if
                         Break.
       Case 200:         Binary Search(LA, R(i) *(Rmod2 *K(i) + d)).
                          If found extract user information Else
                          Store the IP address to Latin American IP addresses Array LA
                          Assign the LA Array to Hash Storage.
                          Break.
     Case 17:    Binary Search(AP, R(i) *(Rmod2 *K(i) + d)).
                 If found extract user information Else
                 Store the IP address to Apple IP addresses Array AP
                 Assign the AP Array to Hash Storage.
                 end if

Step9:  End Switch
Step10: End loop (Log database)
Step11: i + i + 1;
Step12: End loop (Web log file)
Step13: End
```

Figure 5: Unique User Identification (UUI) Algorithm

5.6. Execution of the Algorithm

1. Get the input user IP
2. Extract the first three digits from it
3. Check to which zone it belongs using the switch statement.
4. If it matches a particular zone, then search the given user IP in that particular zone Hash bucket using B5. inary Search and Hash function.
6. If found extract the user information, if not assign the IP address to that [articular zone and treat it as new user.

5.7. Advantages of the modified UUI Algorithm

1. Since the IP addresses are grouped zone wise, easy to search and locate the users IP addresses and their relevant information.
2. Binary Search techniques combined with Hash function makes the searching faster minimizing the time.
3. This proposed algorithm proves and shows better results over other UUI algorithms, which are elaborated in the results and discussions section.

6. RESULTS AND DISCUSSIONS

Table 2
Comparison results of Data Cleaning and Unique User Identification Process from previous work with Murdoc University, ECMIT College and Nehru Arts and Science College

Data Sources	*Murdoc University*	*Emirates College of Management and Information Technology*	*Nehru Arts and Science College*
Entries in raw web log	100000279900 (records)	100450279900 (records)	125232787204 (records)
Entries after data cleaning	100000002783 (records)	100270002783 (records)	126100270002 (records)
Number of users	567502876	606920287	
Number of Unique users	436675422	445275422	463278321
Execution time of UUI(Algorithm)Previous Work	3.257(s)	4.437(s)	NIL
Execution time of UUI(Algorithm) Updated Work	3.00(s)	4.1526(s)	4.2432(s)
Number of sessions	546744372	586744372	602765387

To validate the effectiveness and efficiency of the algorithms proposed, an experiment with the web server logs of Murdoch University and Emirates College of Management and Information Technology, Dubai and Nehru Arts and Science College, India, was made. Results obtained from previous work are compared with the updated work. This work proves with better results to validate the work done. The initial data source of our experiment is from JAN 1, 2014 to Aug 3, 2015, with data size of 10^{12} records. Our experiments are per-formed on a 2.8GHz Intel Celeron I, CPU, 2.00 GB of main memory, Windows 2000 professional, SQL Server 2000 and MATLAB (7.9.0.529). MATLAB tool is used to develop applications to evaluate the performance of the proposed algorithms. The table listed below illustrates the overall performance of UUI algorithm. From Table 2 it is clearly evident that the modified algorithm works fine for a large dataset and also proves the improved performance over the previous work in terms of accuracy and efficiency. Results prove that the proposed UUI algorithm consumes relatively less time to find whether the user record already exists or a new one. The following sections describe the performance of the proposed UUI Algorithm with Murdoch University, Dubai server log data.

Table 3
Overall Performance of Proposed UUI Algorithm's for Murdoch University

Month	*Unique Visitors*	*Number of Visits*	*Pages*	*Hits*	*Bandwidth*
Jan 2014	747792371	747592371	947592371	1947592371	1.9 GB
Feb 2014	726527342	736527342	836527342	2836527342	1.85 GB
Mar 2014	718945720	718945720	718945720	1718945720	1.65 GB
Apr 2014	727654381	717654381	817654381	1817654381	1.7 GB
May 2014	625678990	655678990	755678990	1755678990	1.54 GB
June 2014	543298760	443298760	643298760	1643298760	1.22 GB
July 2014	456789321	556789321	656789321	1656789321	1.02 GB
Aug 2014	326789900	326789900	526789900	1526789900	1.00 GB
Sep 2014	0	0	0	0	0
Oct 2014	0	0	0	0	0
Nov 2014	0	0	0	0	0
Dec 2014	0	0	0	0	0
Total	4873476785	4903276785	5903276785	14903276785	11.88

Table 4
Overall Performance of Proposed UUI Algorithm's for Nehru Arts and Science College

Month	*Unique Visitors*	*Number of Visits*	*Pages*	*Hits*	*Bandwidth*
Jan 2015	834784367	847592352	936592378	2967592371	1.9 GB
Feb 2015	825378634	836527242	836527342	2836527342	1.85 GB
Mar 2015	813975254	818945566	818945720	2518945720	1.65 GB
Apr 2015	812745238	817654631	847654381	1927654381	1.7 GB
May 2015	825678990	855678991	855678990	1877678990	1.54 GB
June 2015	643527876	643298760	843298760	1763298760	1.22 GB
July 2015	843278546	656789321	756789321	2633789321	1.02 GB
Aug 2015	826578387	826789900	626789900	2526789900	1.00 GB
Sep 2015	765432781	847592352	866378990	2543592371	1.63 GB
Oct 2015	853452678	836527242	858398760	2647857342	1.04 GB
Nov 2015	853567846	818945566	856788446	2583725720	1.63 GB
Dec 2015	765432876	817654631	847654381	2927654381	2.00 GB
Total	9663833473	9623996554	9951497369	29755106599	18.18GB

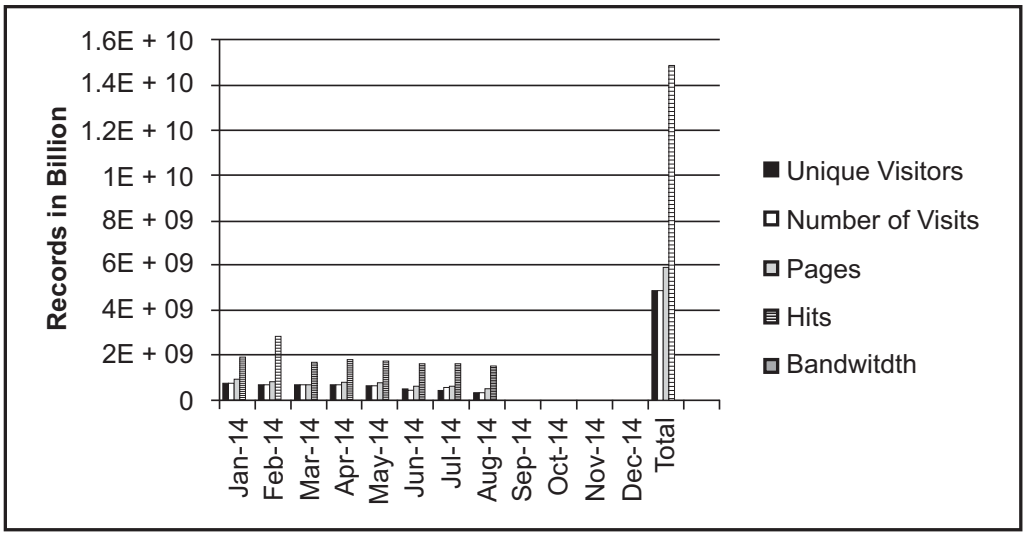

Figure 6: Graphical results of (UUI) Algorithm for Murdoch University

The graphical results displayed in Figures 6 and 7 illustrate the overall performance of the UUI algorithm along with other statistical results. From the results displayed, it is evident that the proposed Data cleaning algorithm performs well along with the proposed UUI algorithm for huge Web Log Server data.

Similarly the proposed modified UUI Algorithm is compared with the works done by Shetal.A.Raiyani in International Journal of Computer Science & Communication Networks, Vol 2 August 2015 [9] and K.R.Suneetha and Dr.Krishnamoorthi in International Journal of Computer Science and Network Security, Vol 9 No 4 April 2009 [11]. The results obtained are displayed in the table below.

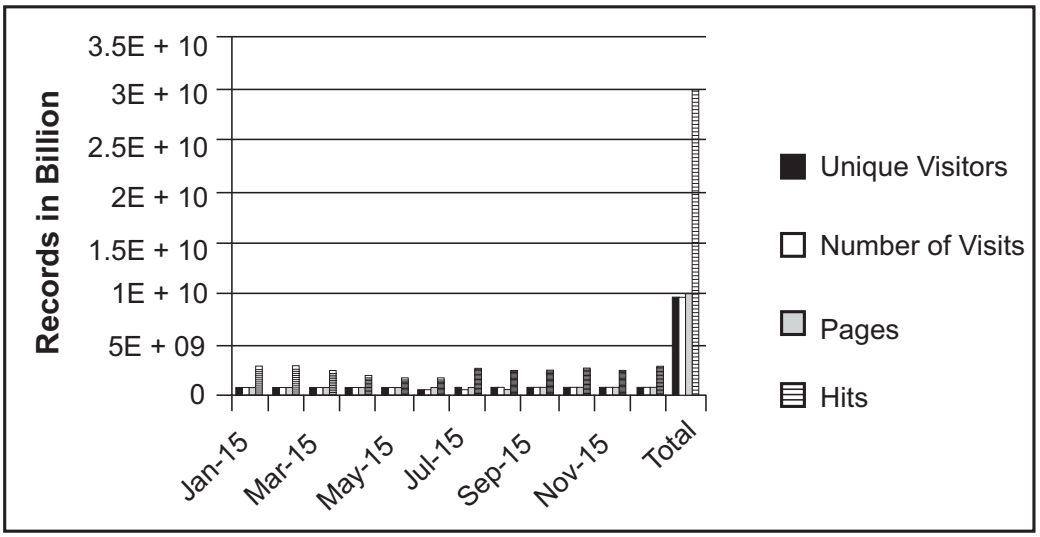

Figure 7: Graphical results of (UUI) Algorithm for Nehru Arts and Science College

Table 5
Performance of Proposed UUI Algorithm with other related works

Performance Analysis	*Database Source*	*Record Size*	*Entries in Raw Web Log*	*Entries After Data Cleaning*	*No of Users*	*No of Unique Users*	*Execution Time(s)*
Unique User Identification Algorithm Proposed by Sheta.A.Raiyani	Web Server Log R.K. University	103	47890	12783	6542	4366	0.2567
Proposed Modified Unique User Identification Algorithm	Web Server Log MURDOC University, Dubai	1012	100000279900	100000002783	567502876	436675422	0.4247
Unique User Identification Algorithm proposed by K.R.Suneetha and Dr.R.Krishnamurthy.	NASA Server Log	104	87233	33657	4000	1765	0.5432
Proposed Modified Unique User Identification Algorithm	Web Server Log WOLLO-NGONG University, Australia.	1012	125644000277	112400027711	577502876	446675422	0.4211

From Table 5 it is clearly evident that the proposed algorithm is far better than the algorithm prosed by Shetal.A.Raiyani in their work. The proposed algorithm shows much clarity in data cleaning and also proves in its efficiency by the consuming less execution time. It takes only 0.4247 second to identify the number of unique users for data size of 10^{12} records, whereas the referred algorithm consumes 0.256 seconds for data size of 10^{3}record sizes, similarly the algorithm proposed by K.R.Sangeetha dn Dr.Krishnamurthy takes 0.5432 seconds to identify 1765 unique users whereas the proposed modified UUI algorithm takes 0.4211 seconds to identify 446675422 unique users , which proves that the proposed UUI algorithm takes less time to execute inspire of the huge data size. Still work is in progress to fine tune the algorithm and improve its efficiency to an appreciable extent

From the above Figures 8 and 9 , it is evident that the proposed Data Cleaning algorithm has performs well. Sample of 642 records were taken from MURDOC University Web Log Server and the data were cleaned using the proposed Data Cleaning algorithm, interestingly 342 irrelevant records were eliminated at a time factor of 1.25 (s). This result is a valid proof for the performance of the proposed Data Cleaning Algorithm.

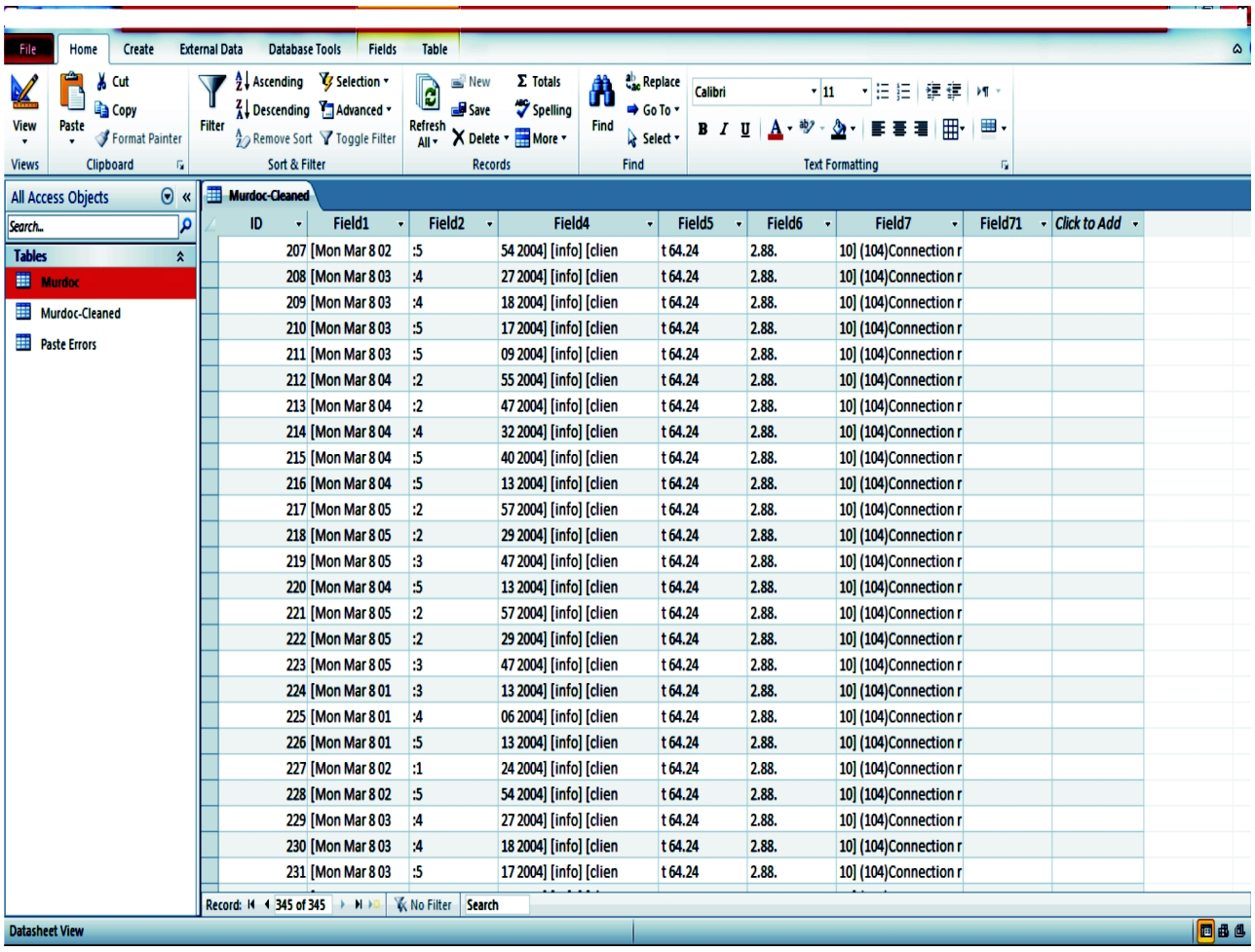

Figure 8: Number of Web Log Server Records of Murdoch University before implementation of proposed Data Cleaning Algorithm

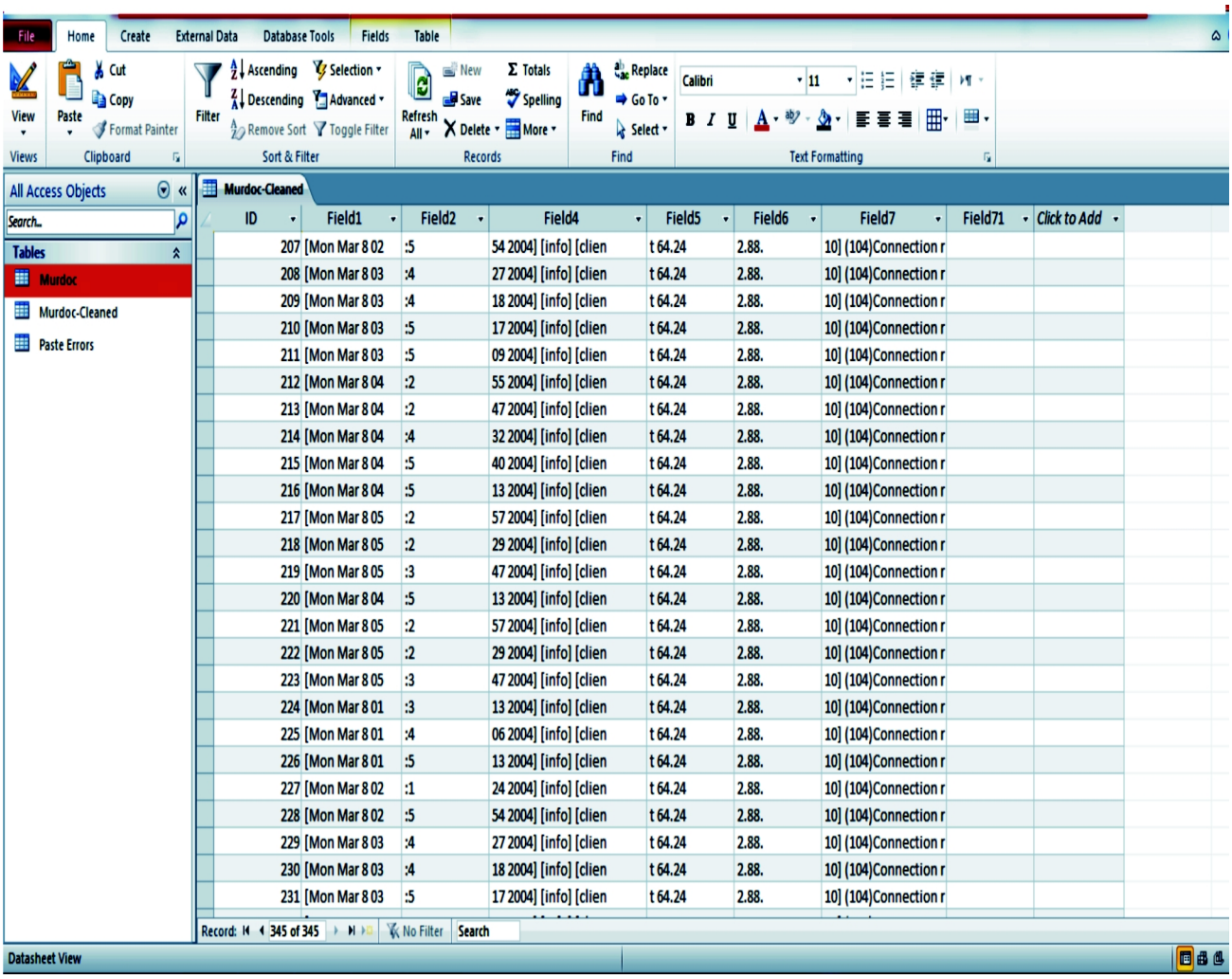

Figure 9: Number of Web Log Server Records of Murdoch University after implementation of proposed Data Cleaning Algorithm

7. CONCLUSION

This paper has comes out with a unique strategy to group IP addresses according to their zone specification. From the grouped IP address, this work uses Binary Search technique to locate the IP of unique user. Inclusion of this strategy in the previously proposed DUI algorithm drastically minimizes the time to search and locate users IP addresses. The algorithm is evaluated with different universities web log server's data to identify the efficiency of cleaning process, to check number of users visited the pages, time taken to identify unique users etc. The algorithm proves and shows much improvement over the previous and other related works. Further improvements are needed to combine the whole process of Web Usage Mining. A complete methodology that covers pattern discovery and pattern analysis will be more useful in user identification process. This work helped the site developers to analyze their sites and also helped then to identify the user types and range of users. It also guided them in further redesigning their sites according to the users requirements.

REFERENCES

[1] BamshadMobasher, Robert Cooley,Jaidep Srivastave. Automatic Personalization based on Web Usage Mining. Communications of ACM Volume 43, Issue 6 Aug 2000,Page(s):142-151.

[2] K.SudeerRedy. An effective data pre-processing method for Web Usage Mining. IEEE, ISBN978-1-4673-5786-9, Page(s):7-10.

[3] Jaideep Srivastava * t, Robert Cooley:l: , Mukund Deshpande, Pang-Ning Tan .Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data University of Minnesota 200 Union St SE Minneapolis, MN 55455.

[4] Shuyan Bai ,Yantai , Qingtian Han , Qiming Liu , Xiaoyan Gao. Research of an Algorithm Based on Web Usage Mining. IEEE, ISBN: 978-1-4244-3893-8, Page(s): 1-4.

[5] Review and Analysis of Hashing Techniques", International Journal of Advanced Research in Computer Science and Software Engineering", Volume 4, Issue 5, May 2014, Page(s): 296-297.

[6] Ranjena Sriram. Dr.R.Mallika.Innovative Pre-Processing Technique and Efficient User Identification Algorithm for Web Usage Mining.International Journal of Advanced Research in Computer Science and Software Engineering.Volume 6, Issue 2, Feb-2016.Page(s):85-90.

[7] Sangeeta Raheja .Comparative study of Hashing Algorithm Using Cryptographic and Steganography Using Audio Files. International Journal of Advanced Research in Computer Science and Software Engineering", Volume 4, Issue 5, May 2014, Page(s):292-294.

[8] Sheetal A. Raiyan. Advanced Pre-processing using Distinct User Identification in web log usage data. International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 6, August 2012 Copyright to IJARCCE www.ijarcce.com 418, Page(s) : 418-420.

[9] Satpal Singh .An Exclusive Survey on Web Usage Mining For User Identification. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 11, November 2014 Page(s):6582-6586.

[10] Shetal.A.Raiyani .Preprocessing and Analysis of Web Server Logs. International Journal of Computer Science & Communication Networks, Vol 2 August 2015, Page(s): 46-55.

[11] K.R.Suneetha and Dr.R.Krishnamoorthi. Identifying User Behavior by Analyzing Web Server Access Log File. International Journal of Computer Science and Network Security, Vol 9 No 4 April 2009, Page(s) :327-332.