A Hybrid Recommendation Model for Web Navigation

Conference Paper · December 2015

DOI: 10.1109/IntelCIS.2015.7397276

3 authors:

No'aman Muhammad
Port Said University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ahmed M. Gadallah
Institute of Statistical Studies and Research
13 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Hesham A. Hefny
Institute of Statistical Studies and Research
186 PUBLICATIONS **369** CITATIONS
SEE PROFILE

A Hybrid Recommendation Model for Web Navigation

No'aman M. Abo Al-Yazeed
Computer Sciences Dept.
HIMC, Port Said University
Port Said, Egypt
no3man_mohamed@himc.psu.edu.eg

Ahmed M. Gadallah
Computer and Information Sciences Dept.
ISSR, Cairo University
Giza, Egypt
ahmgad10@yahoo.com

Hesham A. Hefny
Computer and Information Sciences Dept.
ISSR, Cairo University
Giza, Egypt
hehefny@hotmail.com

***Abstract*— Nowadays, users rely on the web for information gathering. Accordingly, web usage mining becomes one important subject of research. Such research area covers prediction of user near future intentions, web-based personalized services, customer profiling, and adaptive web sites. Web page prediction is strongly limited by the nature of web logs, the intrinsic complexity of the problem and the tight efficiency requirements. This paper proposes a hybrid page ranking model based on web usage mining technique by exploiting session data of users, to enhance the recommendations of the next candidate web page to be accessed. The proposed approach represents a combination between two page ranking approaches. The first one computes the frequency ratio indicating the number of occurrences of each page in the search result. On the other hand, the second approach computes the coverage ratio from similar behavior patterns. As a result of the proposed approach, a set of candidate pages are ranked and the page with highest rate is recommended. The proposed approach has been tested on real data collected and extracted from the web server log file of CTI main web server.**

Keywords*— *Adaptive Web Sites, Navigation Pattern Mining, Recommender System, Web-Based Recommendation Systems, Web Log, Web Mining, Web Personalization, Web Usage Mining.

I. INTRODUCTION

The volume of information available on the internet is increasing rapidly with the ongoing growth of the *World Wide Web* and the huge amounts of web-based applications. Almost, web searching by users suffer from the information overload problem because they are provided with more information and service options. Accordingly, it becomes more difficult for them to find the "*right*" or "*interesting*" information easily [1]. Commonly, modeling of human behavior is one of the major challenges of computer science in the new century. Therefore, web mining becomes essential not only to better understand human behaviors, but also to allow dynamic restructuring of contents and adaptive navigation of the internet. The explosive growth of web users data, in the form of log files, increases the interest of web usage mining to enhance the web navigation process [1]. Alternatively, recommender systems approaches can be used to handle the problem of information overload [2]. It aims to adapt the content and structure of web sites respecting previous navigation behaviors of web users stored in the web server log files. Commonly, recommender systems guide users toward the more interesting or useful objects in a large space of possible options [3].

The rest of this paper is organized as follows: A brief overview about recommender systems is given in section **II.** Section **III** introduces some of previous and related works. The proposed approach is presented in section **IV**. An illustrative case study is outlined in section **V**. Finally, section **VI** presents the conclusions and the future work.

II. RECOMMENDER SYSTEMS

Recently, there has been an increasing interest to develop a web recommender systems using the application of web usage mining techniques [4-6]. Traditional techniques are mainly depends on users ratings on different items or other explicit feedbacks provided by the users [7, 8]. Generally, Web server access log records the user's browsing history that contains an abundance of hidden information about users and behaviors of their own navigation. Web page recommender systems use the log files of web server access as input. Consequently, they use some of data mining techniques such as association rules mining and clustering to extract implicit navigational patterns, this patterns considered to be potentially useful which then used to provide a recommendations. Therefore, such mining techniques represent good alternatives to the explicit user rating or feedback about the navigation behavior. They discover the user's preferences from their implicit feedbacks, namely the navigated web pages.

Several approaches like clustering and collaborative filtering preparative to integrate both binary and non-binary weights of pages, binary weights are commonly used for computing efficiency [9]. Binary weights of page views representing the existence or nonexistence of a product-purchase or a documents access in the transaction; non-binary can be a function of the duration of the associated page view in the user's session [10].

On the other hand, association rule mining (***AR***) can lead to higher recommendation precision and are easy to scale to large datasets [11]. Yet, incorporating page weight into AR models has not been explored in previous studies. Weighted association rule (***WAR***) mining allows different weights to be assigned to different items, and is a possible approach to improving the AR model in the web personalization process [9].

Generally, Markov Model (***MM***) represents one of the most common sequential pattern mining techniques used for web recommendations [12]. In general, it can achieve good prediction accuracy considering consecutive and sequential page accesses. Markov model has many orders based on the number of considered pages in the web log file entries determines the order of the model. It was extended to Lower-Order and Higher-Order. Lower-order markov model provides high coverage, but with low accuracy. On the other hand, higher-order markov model gives low coverage but high accuracy with more time complexity [13]. In first order markov model, each state represents a single web page and each pair of visited pages corresponds to a state transition. Two artificial states, start and final, are incorporated in the model. On the other hand, in second-order markov model each state corresponds to a sequence of two visited web pages and so on [14].

III. Previous Work

The importance of study and modeling the behaviors of web users led to a number of research works in this area. However, most of these works are hindered by some kinds of limitations [15, 16]. Different combinations of mining techniques already proposed to the recommendations of web access. Commonly, many approaches were introduced to achieve such recommendation based on markov model which is considered the widest one used to model user's web navigation.

R. Bhushan et al. in [17] introduced a new model based on learning from web logs and recommends users a list of pages which are considered relevant to him respecting the user's historical patterns. Then, the search result list is optimized by a re-ranking process. On the other hand, D. Dhyani et al. introduced another model based on markov model for web access prediction [18]. Yet, such model has the drawback of its high complexity due to considering all access sequences throughout the prediction process. Also, K. Han-Gyu et al. [19], proposed a recommendation approach which work at content level, recommendations are done across different categories considers information semantic for contents and user interests. The underlying structure of the contents semantics is Linked Data, its consider the source to find the relevant results. Retrieved data are then grouped together based on their similarity and relevance to form a clusters. Finally they recommends contents which are more likely to be interesting to general users based on the content consumption trends that monitored by user groups more prominent proactive and often consume contents.

Also, A. Maratea et al. in [20] introduced a heuristic approach based on majority intelligence technique, its adapted to changes of the navigational patterns easily; while users surfs the web, it's provide them with recommendations with low cost. In an unidentified environment, the proposed technique tries mimic human behavior. This approach works perfectly in real time with good accuracy in case of web surfing by several users in parallel.

On the other hand the authors of [21] introduced a new approach to predict users browsing behavior at two levels to meet the nature of the navigation. The first one is the category stage and the other is the web page stage. The first stage is concerned with predicting the category under interest. Accordingly, the unnecessary categories are excluded. The scope of calculation is massively reduced. In consequence, pruned markov models with higher order were used in the second level to predict the users browsing pages. Also, A. Anitha introduced a new approach for next page access prediction [22]. It uses a hybrid approach which combines an integration of markov model with clustering based pair wise nearest neighbor technique. The application of the proposed model has led to improve the quality of the resulting patterns and reduce the size of the data used in the sequential mining process significantly. Ignoring the loosely connected sequences of web access in the mining process is the major drawback of this work.

Additionally, M. Jalali et al. in [23] introduced a novel web recommendation system called WebPUM. The developed system use web usage mining techniques to predict the near future intentions for web users online based on classifying user's navigation patterns. Modeling of user's navigation patterns at first phase is done by the new graph partitioning algorithm. Moreover, classifying current user activities are based on using of longest common subsequence algorithm to predict the next user's action. This work was hindered its moderate accuracy in spite of highly computations complexity.

Also, Y. AlMurtadha et al. proposed an improved web page recommendation algorithm using profile aggregation based on clustering of transactions (***IPACT***) [24]. This algorithm is a two phases recommender algorithm (offline and online) based on clustering technique. Offline phase is responsible for data pre-processing and session clustering using previous log data of profiles. The input for the online phase is the output of the offline phase. Using offline & online session clustering, the online phase will generate the session classification. After that recommendation engine will generate the proper recommendation based on session clustering & classification.

The main task of both phases is to generate the navigation patterns profiles to get recommendation. The experiment result is based on accuracy of precision & coverage. Result shows that how prediction engine is accurate in recommendation to particular user profile.

R. Mishra et al. [25], introduced a novel approach to predict the next movement for web user taking into account current sequential information with content information were exist in web navigation patterns. This model apply soft clusters through clustering process to improve captured multiple user's concerns. Its utilized a rough set based similarity upper approximation using both sequential similarity and content similarity. To generate a recommendation for users, it's also using singular value decomposition (SVD). System evaluation proves that the proposed approach is applicable.

IV. The Proposed Model

This work proposes a more efficient hybrid recommendation model for web navigation based on the historical navigation behaviors existed in the web logs. Generally, the proposed model incorporates three main phases: ***Preliminary Preparation Phase, Data Summarization and Transformation Phase, and Recommendation Phase.***

A. Preliminary Preparation Phase

This phase is concerned with web logs data preprocessing. Generally speaking, web access log mining process starts with parsing the web access log file and importing the required data into a relational database. Consequently, a preparation process takes place on the imported data as shown in Fig. 1. Such preparation phase consists of the following steps:

- Data cleaning and filtering the irrelevant data which is not appropriate for the mining process.
- Identify each unique user by considering each IP address represents one user. For more logs, if the IP address is the same, but the log shows a change in the user agent, then such IP address represents a different user. Using the access log in conjunction with the referrer logs and site topology to construct browsing paths for each user; if requested page 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.
- Creating user sessions which is considered one of the major steps in data preparation phase, in this step the page accesses of each user is divided into individual sessions according to a time threshold (*in our experiment 30 minutes*), If the time between two pages request exceeds a certain limit, it is assumed that the user is starting a new session.
- Path completion was done to fill in these missing references page that are not recorded in the access log by using the site topology and link structure in web site.

B. Data Summarization and Transformation Phase

In this phase many computations were done for data summarization and visualization purpose, the resulting statistical data (*e.g. web page frequency, mean duration, etc.)*, are used later in the recommendation phase and also reported to web server administrator, advertising agencies, and etc. As follow, explanation of this phase's tasks in detailed:

- Page ranking is the first step in this phase. This step is concerned with creating interesting information such as page frequency (PF) and page mean duration (MD). A page frequency represents the number of times the page was visited as depicted in Eq. (1). On the other hand, a page mean duration indicates the users' mean stay time in exploring such page as depicted in Eq. (2). Accordingly, such relation can be used to identify the pages with high and low requests. Such information can help web server admin to rearrange the web site to reach the more requested pages easily and quickly.

$$PF(p_i) = \textbf{SELECT COUNT (URI)}\ \textbf{FROM Training Log}\ \textbf{WHERE URI LIKE } `p_i\text{'} ; \quad (1)$$

Where "p_i" represents the selected page.

$$MD(p_i) = \frac{\sum_{n=1}^{f}(D_n(p_i))}{PF(p_i)} \quad (2)$$

Where "D_n", represents web page surfing duration at visit number "n" for the web page 'p_i'.

- Basket transformation [26], is the next step in this phase. In this step sessions are treated as patterns representing sequences of web pages. Each pattern is weighted by the number of its occurrence. In consequent, this step reduces the data of the search space by over 70%. On the other hand, not all sessions and web pages are considered. That is, the sessions with less than 2 visited pages and web pages with a frequency less than a specified threshold are excluded. Consequently, certain thresholds are set to eliminate search space and increase the recommendation accuracy.

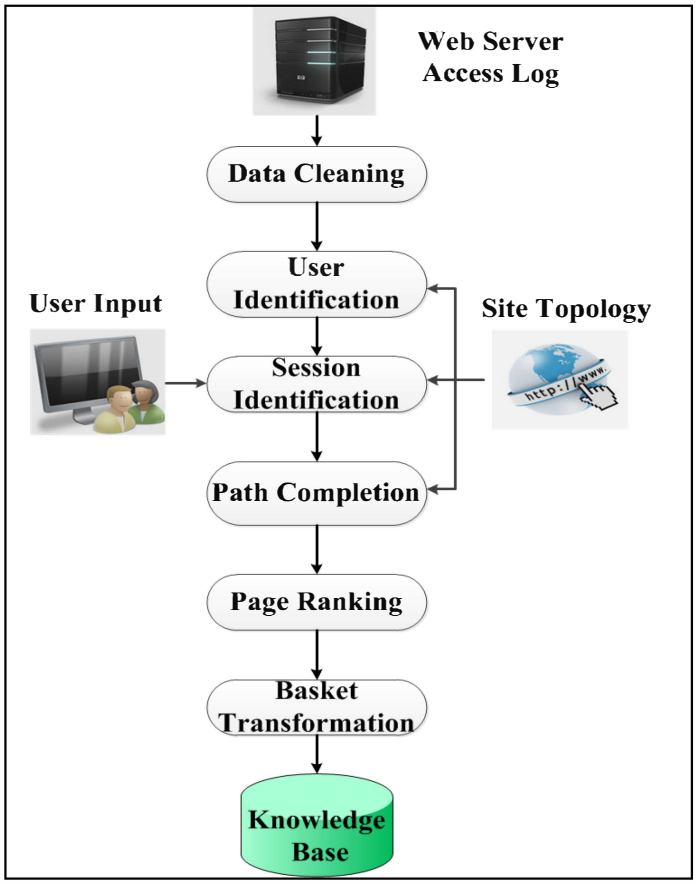

Fig. 1: Web Log Preprocessing Steps.

C. Recommendation Phase

In this phase the proposed recommendation algorithm is applied on the transformed data which are resulted from previous phase. The inputs of this algorithm are the processed web access log, and the current user's session data .On the other hand, the output represents a set of recommended web pages which are more appropriate for the current user navigational behavior.

- The algorithm starts with receiving the parameters consists of current user browsing history "*input pattern*". Consequently, it searches the reached transformed data, from phase two, for sessions which contain one or more pages from the input pattern.
- Next, each distinct web page is selected from the search result and removes input pattern web pages from selection.
- For each web page resulted in previous step, compute the number of times the page is requested (**F**) (See Eq.(3)), divided by the total number of pages requests in search result "***Frequency Ratio***" as shown in Eq. (4).

$$F_{p_i} = \sum_{\substack{i=1 \\ p_i \in vp_i}}^{m} w(vp_i) \quad (3)$$

Where F_{p_i} is the page p_i frequency and $w(vp_i)$ is the occurrence of the visited pattern vp_i including page p_i.

$$FR_{p_i} = \frac{F_{p_i}}{\sum_{n=1}^{d}(F_{p_n})} \quad (4)$$

Where F_{p_n} is the frequency of page n in the searched web log data set.

- In consequence, the input pattern frequency "*IF*" is computed by counting the number of times the input pattern is requested in the search result as shown in Eq. (5). If no matched result (*IF = 0*) then the first page is removed from input pattern to be (*L - 1*). This step is repeated in *a dynamic pruned search* fashion until (*IF > 0*).

$$IF = \sum_{\substack{i=1 \\ ip \in vp_i}}^{m} w(vp_i) \quad (5)$$

Where IF represents the frequency of the input pattern ip and $w(vp_i)$ is the occurrence of the visited pattern vp_i including the input pattern ip.

- Consequently, the support (S) and coverage ratio (CR) for each distinct web page are computed using Eq. (6) and Eq. (7) respectively.

$$S_{cp_i} = \sum_{\substack{i=1 \\ ip_i \in vp_i \\ cp_i \in vp_i}}^{m} w(vp_i) \quad (6)$$

Where S_{cp_i} represents page support for candidate page p_i, and $w(vp_i)$ is the occurrence of the visited pattern vp_i that include the input pattern page ip, and include also the candidate page cp_i.

$$CR = S / IF \quad (7)$$

- Finally a recommendation relation is constructed which contain candidate pages sorted by coverage and frequency ratios. If two or more pages have identical rank, page frequency is used to re-rank the pages. The top page is selected and recommended to the user. The proposed web page recommendation algorithm is shown in Algorithm 1.

Algorithm 1: web page recommendation algorithm.

Inputs: Transformed Log Relation (**TL**), Page Frequency (**PF**), Page rank relation (**PR**) and Input Pattern (**IP**).
Output: Top rated recommended pages
Begin

```
foreach (var Page in IP)
  {//Exclude Low Frequency Page From Input Pattern
        if (!(TL.Contain(Page)))
              IP.Remove(Page);
}//End foreach
if (IP.Length >= 1) //Input Pattern Length Must Be >=1.
{
   Select Sessions From "TL" into "IH" Where Session
   Contains any Page From "IP" As Visit Action;
   Select Distinct Pages and Its Frequency "F" from "IH"
   into "CP";
   From "CP" Sum Pages Frequency So:
   sum = ∑_{n=1}^{d} (f_n);
   foreach Page in "CP"
   {
        Compute: FR = F / Sum;
   }//End foreach
   //Computing Input Pattern Frequency for sessions in "IH"
   int IF = 0;
   While (IF < 1) //There is No Matched Pattern.
   {
      for each (var M in IH)
      {
          if ( M.Contains(IP))
              IF += M[Weight];
      } //End for each
      if (IF == 0)
         IP.Remove(First Page);
              //Exclude First Page From "IP"
   }//End While
   //Computing the support of Candidate Pages
   int S = 0 ;     //Count of current page visits After "IP"
   foreach (var Page in CP)
   {
        IP.Add(Page);
        foreach(var M in IH)
        {
            if(M.Contains(IP))
                S += M[Weight];
        } //End foreach
        IP.Remove(Page);//Compute Coverage Ratio
        CR = S / IF;
        S = 0;
    }//End foreach
    Sort Pages in "CP" By CR, FR , PF;
    Select Top Page From "CP";
} else
{   Select Top Page From "PR"; }//end if
Recommend Selected Page;
```

End

V. An Illustrative Case Study

The experiments have been conducted on DePaul University CTI (http://www.cs.depaul.edu) logs file dataset, which contains the log file for CTI main web server during the month of April 2002. The original file size was 252 MB which includes of 1051105 records. Each record represents a request to the server. The Microsoft log parser 2.2.10 is used as a command line utility to import the log file into an SQL Server Database. This tool provides a universal query access to text-based data such as log files, XML files and CSV files. The obtained log file is in the original log format used by IIS. The only cleaning operation performed on this data by owners was to remove references to auxiliary files such as images, sound and video files, etc., to make the data more manageable. Only references to the content files and scripts that generate content pages are included in the data. No other cleaning, such as removing erroneous references or spider navigational references have been performed by owners.

Commonly, Log files data includes user IP address, user name, date and time of the request, request method, requested page, web site name, network web server name, server IP, port number, user search query, time taken, host name, protocol version, status code, referred page, and user agent. Reasonably, gathered data had to be prepared; it has some fields which are not relevant to the purpose of our experiments, redundant information like time taken, port number, etc. are removed. The next step is to create time stamp to each request. Next; log data should be cleaned and filtered, cleaning the data involved removing erroneous and invalid pages, the status code less than **200** and greater than **299** were removed, which represents a successful request. Table 1 shows the frequency of various status codes involved in the log file. Cleaning phase also includes deleting records with request method other than "**GET**", in addition to deleting all requests that contain graphical extensions or any other content pages.

TABLE 1: TOP STATUS CODES FREQUENCIES FOR WEB LOG FILE RECORDS.

Status Code	Records Count	Percentage
200	789703	75.1352%
302	244475	23.2602%
500	7747	0.7371%
401	3769	0.3586%
206	2696	0.2565%
304	1727	0.1643%
403	318	0.0303%
207	270	0.0257%
405	208	0.0198%

Table 2 shows the count of each request method on the underlined log. On the other hand, Table 3 shows the number of occurrences of the logged web pages respecting their extensions. Consequently, in order to ensure data and results quality, non-human behavior like spiders, crawlers and

automatic web bots (*e.g. "fastwebcrawler /3.5+(atwcrawler+at+fast+dot+no;+ http:// fast. n o /support.php?c=faqs/crawler)"*) are deleted.

TABLE 2: REQUEST METHODS FREQUENCIES IN THE LOG FILE.

Request Method	Request Count	Percentage
get	999005	95.0489%
post	51023	4.8545%
head	413	0.0393%
propfind	311	0.0296%
options	287	0.0273%
connect	4	0.0004%

TABLE 3: TOP WEB LOG FILE STATISTICS BASED ON THE PAGE TYPES.

Page Extension	Request Count	Percentage
.asp	1028169	97.8237%
None	9773	0.9298%
.pdf	5053	0.4808%
.htm	2608	0.2481%
.dll	2292	0.2181%
.lpk	1271	0.1209%
.doc	695	0.0661%
.rtf	320	0.0304%
.html	221	0.0210%
.pl	186	0.0177%
.cab	179	0.0170%
.exe	105	0.0100%

On the other hand, the frequencies of human and automatic web bots records in the log file are shown in Table 4.

TABLE 4: HUMAN AND AUTOMATIC WEB BOTS STATISTICS.

Type	Count.	Percentage
Bot	22666	2.16%
Not Bot	1028377	97.84%

In consequent to log file preparation, the user and session identification is the next step to identify each user and divide user transactions into sessions respecting a 30-minute session timeout into consideration. The frequency of each page visited by the user was calculated. Short sessions were then removed and only pages with high frequency were considered, Table 5 shows changes of data size across adjacent processing phases.

TABLE 5 : DATA SIZE ACROSS PROCESSING PHASES.

Phase	No. of Records	No. of Users
Raw Data	1051043	34778
Cleaned Data	740822	33424
User Identification	740822	44660
Session Identification	740822	102277
Training Log	518575	90264
Test Log	222247	27924
Filtered Sessions — Training	490560	62249
Filtered Sessions — Test	214789	20466
Transformed Log (Weighted)	17386	56089

Top sessions length and its frequency presented in Table 6, its show that most sessions consists of 2 : 10 visited pages, session with length 1 and too long sessions considered a noise and must be deleted to maintain recommendation accuracy.

TABLE 6: TOP FREQUENT SESSION LENGTH.

Session Length	Training Set Frequency	Test Set Frequency	Total
2	12812	4245	17057
4	9407	3049	12456
3	8270	2833	11103
5	6423	2032	8455
6	4308	1391	5699
7	3219	1027	4246
8	2614	824	3438
9	2036	750	2786
10	1683	528	2211
11	1339	439	1778

Yet, the data are not in a format that is appropriate for mining purpose, so further transformation should done, a new table are created to represent sessions in such a way, each one session in the form of vector, its consist of sequence of visited pages, session length, and session duration. Not all web pages are involved in this representation, frequent pages only involved in this process, to involve a particular page in transformation process; certain threshold are set "*Only web pages with Frequency > 10 were considered*". Duplicate sessions are taken into consideration, so we have created a new variable called "*weight*", which assign an integer value to each session represents the number of times the session was repeated, this method reduces the data size by over **70%**. So, it's helps in the search process by reducing the search space leading to speed the search process, and therefore reduce recommendation time, Table 7 show most frequent user's sessions, visited pages, session weight, and session duration.

TABLE 7: TOP FREQUENT SESSIONS AND ITS MEAN DURATIONS.

Length	Visited Pages	Weight	Duration
2	P392, P479	3902	6
4	P392, P479, P166, P169	3163	25
4	P392, P111, P116, P113	2063	34
2	P392, P97	1322	19
5	P392, P111, P116, P113, P169	1115	23
2	P33, P170	1007	19
3	P392, P17, P335	847	473
4	P392, P479, P321, P169	732	39

3	P33, P170, P392	623	437
3	P392, P166, P169	520	15

Sample of dataset has been chosen randomly, the algorithm is tested in such a way that, by eliminate the last page in each test session. The proposed algorithm repeated through test sessions. Firstly, algorithm takes each user session in the form of sequence of visited web page as input parameter. Next, its search in transformed log for each page in input sequence to determine whether it's involved in it or not to exclude not found web pages. Searching for sessions contain any element of input pattern in transformed log. Then, exclude search pattern elements from search result and compute frequency ratio for each page in search result, each page is multiplied by session's weight.

In the next step, "*A Dynamic Pruned Search Technique*" was introduced. This technique has been applied in such a way. First, by counting the number of times the occurrence of search pattern; if there is no match "*Input Pattern Frequency = 0*", first page are excluded from search pattern and so on while "*Input Pattern Frequency > 0*", this step emphasizes that the algorithm covers all cases. Then, a new variable called "*Candidate Page Support*" is computed for each page in search result by counting the number of times the occurrence of page given input pattern has occurred. Finally, another variable called "*Coverage Ratio*" is computed for each candidate page by divide Candidate Page Support value by Coverage Ratio value. Resulted pages are then ordered by coverage ratio, frequency ratio and page frequency; selecting the highest rated Web page and then a recommendation is made.

To evaluate the proposed hybrid model for recommendation, it has been providing an "*Accuracy Metric*". Accordingly, accuracy has been defined as the ratio of the number of correct recommendations to the number of total recommendations, as shown in Eq. (8). In the recommendation phase, system recommends to the user the next possible page to be visited. Next, we compare the recommended page with actual visited page; If the recommended page is the same as of the actual visited page, then, the event termed as success. Else, "*web pages are not the same*", the event termed as fail.

Accuracy "*ACC*": Computing the proportion of the number of successful predictions to the number of total predictions.

$$Acc = No.\ of\ success\ /\ (No.\ of\ success + No.\ of\ fail). \quad (8)$$

For evaluating and testing of the proposed model, it has been compared with other similar models in terms of prediction accuracy to prove the extent of the proposed model's ability to give appropriate recommendations to clients.

In the first experimental comparison between the proposed model with WebPUM model [23], a sample of data set was selected randomly , then the sample has been formed into 10 parts, whereby each part is a percentage of the sample. We started the experiment from 10% to 100% of the sample. From TABLE 8, we can observe some aspects of data, and its distribution; the average length of sessions is almost 13 page views. The proportion of the training set to the test set is 70 to 30 respectively. As observed from test results, shown in TABLE **9**, and Fig. 2, the proposed model achieves moderate prediction accuracy with small portion of training data set. This behavior can be interpreted because of loss of background for proposed model due to low training. Also, the WebPum model achieves moderate prediction accuracy with slightly increasing over the proposed model.

Thereafter, while the data set are growing, it's also resulting in increase of the result's accuracy. Fig. 2 shows that the result's accuracy of the proposed model has been increased in compared with the WebPum model. Such behavior can be interpreted that, the proposed model have less complexity and covers all cases of recommendation respecting different input patterns. On the other hand, in WebPum model the result's accuracy remain stable over the increase of data set "*Results were not affected by varying of the proportion of data set*".

TABLE 8: SHOWS SOME STATISTCS OF THE EXPEREMINT USED DATA SET.

Data Set Records	**Training Set**	**Test Set**	**Total**	**Page Per Session**
	Sessions			
10000	570	290	860	11.62791
20000	990	610	1600	12.5
30000	1530	909	2439	12.30012
40000	2111	1207	3318	12.05546
50000	2732	1401	4133	12.09775
60000	3101	1710	4811	12.47142
70000	3440	2025	5465	12.80878
80000	3824	2243	6067	13.18609
90000	4385	2318	6703	13.42682
100000	4693	2516	7209	13.87155

TABLE 9: TEST RESULT ACCURACY.

Data Set Records	**Test Set** Sessions	**Success Predictions**	**Percentage (%)**
10000	290	116	40.00
20000	610	251	41.15
30000	909	446	49.06
40000	1207	626	51.86
50000	1401	796	56.82
60000	1710	1144	66.90
70000	2025	1545	76.30
80000	2243	1783	79.49
90000	2318	1851	79.85
100000	2516	2010	79.89

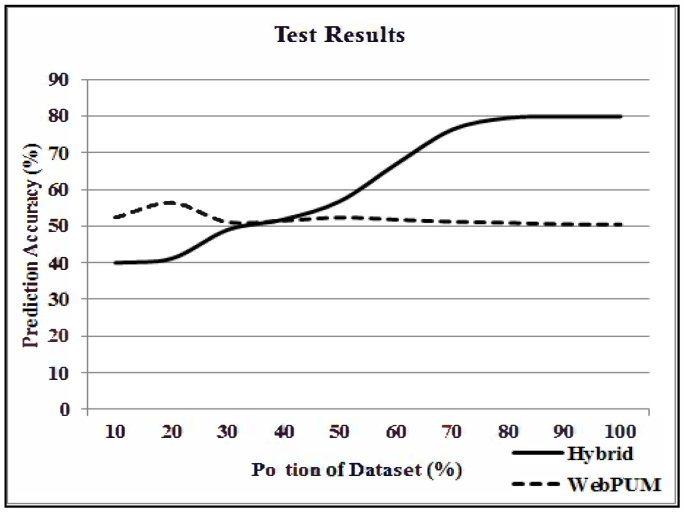

Fig. 2: Prediction Accuracy of the Proposed Model and WebPUM Model.

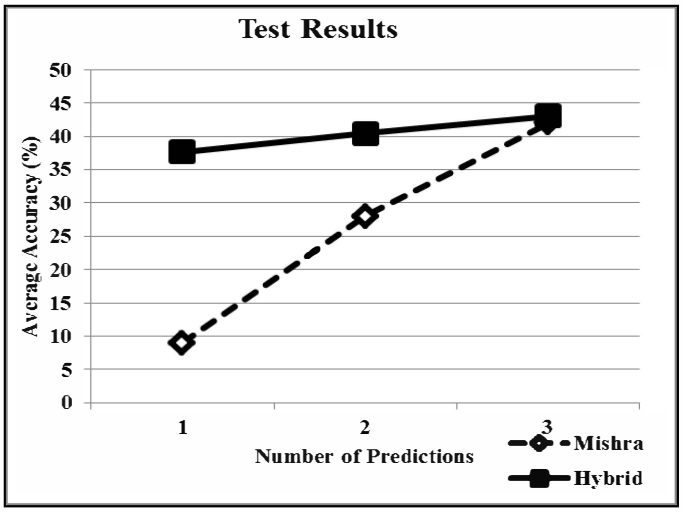

Fig. 3: Prediction Accuracy of the Proposed Model and Mishra's Model.

The next comparison was done between the proposed model and the model introduced by Rajhans Mishra, et al. [25], in terms of prediction accuracy. For testing purpose, 5,000 sequences of user sessions have been taken as training set, and select 10 random different groups from a dataset of size 2,000 sequences of user sessions for each one as test set. Table 10 shows the results accuracy of the proposed model across various samples of the CTI dataset.

TABLE 10: TEST RESULT ACCURACY OF THE PROPOSED MODEL ACROSS NUMBER OF PREDICTIONS.

Test Set	No. of Predictions		
	1	2	3
Sample 1	40.60	41.00	45.15
Sample 2	37.50	38.65	40.65
Sample 3	39.00	41.30	43.95
Sample 4	37.10	45.45	47.15
Sample 5	34.00	35.50	38.40
Average	37.64	40.38	43.06

Comparison made between test results of the proposed model and the results of Mishra's model using a user defined parameter **M** (*number of clusters chosen for constructing the response matrix A*), where **M = 32**. Fig. 3: shows the results accuracy of the proposed model compared with the first prediction is greater than Mishra's model; while success of second and third prediction has slight effects on improve result. But in Mishra's model, the number of the predictions has a significant impact in improving the accuracy of the results.

Finally, the proposed model were compared with IPACT model [24]. Recalling that, the aim of the experiments is to evaluate the both models ability on predicting the next web page visit based on the current navigation session for testing the prediction accuracy of the proposed algorithm. In our experiments, coverage ratios have been used instead of recommendation score and apply recommendation threshold to the set of recommended pages. Fig. 4 show test results for both models. The results of the two models are close together, but in the case of the proposed model the more restrictive of the recommendation score have a greater effect on improving the quality of predictions than the comparative model.

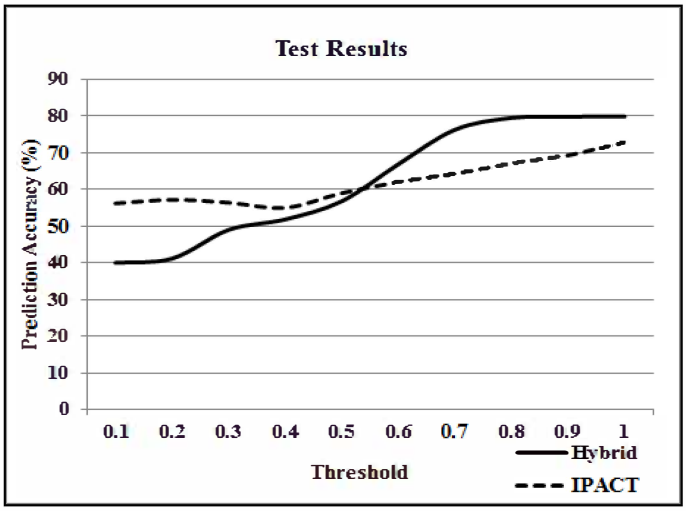

Fig. 4: Prediction Accuracy of the Proposed Model and IPACT Model.

VI. CONCLUSIONS

World Wide Web is growing rapidly with a huge number of users and massive amounts of data presented in web pages. A web user aims mainly to obtain quick answers of his/her generated queries by recommending the more closed web

pages to the specified criterion in a flexible and reliable manner. Indeed, many approaches were introduced to achieve such recommendation based on markov model which is considered the widest one used to model user's web navigation. Yet, there is a need to other approaches to enhance and facilitate web browsing be recommending the web user by the more suitable web page respecting both of his/her input navigation pattern and the historical navigational pattern existed in the web log files. Consequently, a web user will be recommended by the more suitable web page in a more reliable manner.

Accordingly, the proposed approach for web usage mining and recommendation is essential. It is necessary to analyze the user's web navigation historical behaviors stored in the web log files in order to improve the quality of web services offered to the web user. This paper proposed a hybrid model based on page ranking algorithm. It has been tested and compared with other models. As shown in the results of the illustrative case study, and compared with some of previous works, the proposed approach is more efficient and reliable. It has been tested with some different cases concerning the size historical data set. In the future, there is a plan to design an incremental algorithm for the proposed approach so that it does not start from scratch each time in web page prediction.

REFERENCES

[1] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), pp. 558-567, Nov. 1997.

[2] Paul Resnick, and Hal R. Varian, "Recommender Systems," *Communications of the ACM,* vol. 40, no. 3, pp. 56-58, Mar. 1997.

[3] Robin Burke, "Hybrid Recommender Systems: Survey and Experiments," *User modeling and user-adapted interaction,* vol. 12, no. 4, pp. 331-370, Nov. 2002.

[4] Xiaobin Fu, Jay Budzik, and Kristian J. Hammond, "Mining Navigation History for Recommendation," In Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, Louisiana, USA, pp. 106-112, 2000.

[5] Yi-Hung Wu, Yong-Chuan Chen, and Arbee L. P. Chen, "Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors," In Proceedings of the 11th International Workshop on Research Issues in Data Engineering, pp. 17-24, 2001.

[6] Bamshad Mobasher, "Web Usage Mining and Personalization," In Practical Handbook of Internet Computing, M. P. Singh (ed.), CRC Press, pp. 264-265, 2004.

[7] Mukund Deshpande, and George Karypis, "Item-Based Top-N Recommendation Algorithms," *ACM Transactions on Information Systems,* vol. 22, no. 1, pp. 143-177, Jan. 2004.

[8] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99), Berkeley, California, USA, pp. 230-237, 1999.

[9] R. Forsati, and M. R. Meybodi, "Effective Page Recommendation Algorithms Based on Distributed Learning Automata and Weighted Association Rules," *Expert Systems with Applications,* vol. 37, no. 2, pp. 1316-1330, Mar. 2010.

[10] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa, "Effective Personalization Based on Association Rule Discovery From Web Usage Data," In Proceedings of the 3rd international workshop on Web information and data management, Atlanta, Georgia, USA, pp. 9-15, 2001.

[11] Weiyang Lin, Sergio A Alvarez, and Carolina Ruiz, "Collaborative Recommendation via Adaptive Association Rule Mining," In Proceedings of the International Workshop on Web Mining for E-Commerce (WebKDD '2000), Boston, Aug. 2000.

[12] A. Anitha, and N. Krishnan, "A Web Usage Mining Based Recommendation Model for Learning Management Systems," In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-4, Dec. 2010.

[13] Faten Khalil, Jiuyong Li, and Hua Wang, "A Framework of Combining Markov Model with Association Rules for Predicting Web Page Accesses," In Proceedings of the 5th Australasian Conference on Data Mining and Analystics (AusDM '06), Sydney, Australia, pp. 177-184, Dec. 2006.

[14] Bhawna Nigam, Sanjiv Tokekar, and Suresh Jain, "Evaluation of Models for Predicting User's Next Request in Web Usage Mining," *International Journal on Cybernetics & Informatics (IJCI),* vol. 4, no. 1, pp. 1-13, Feb. 2015.

[15] J Vellingiri, and S Chenthur Pandian, "A Survey on Web Usage Mining," *Global Journal of Computer Science and Technology,* vol. 11, no. 4, pp. 66-72, USA, Mar. 2011.

[16] Mathias Géry, and Hatem Haddad, "Evaluation of Web Usage Mining Approaches for User's Next Request Prediction," In Proceedings of the 5th ACM International Workshop on Web Information and Data Management (WIDM '03), New Orleans, Louisiana, USA, pp. 74-81, Nov. 2003.

[17] Ravi Bhushan, and Rajender Nath, "Recommendation of Optimized Web Pages to Users Using Web Log Mining Techniques," In Proceedings of the 3rd IEEE International Advance Computing Conference (IACC), pp. 1030-1033, Feb. 2013.

[18] Devanshu Dhyani, Sourav S Bhowmick, and Wee-Keong Ng, "Modelling and Predicting Web Page Accesses Using Markov Processes," In Proceedings of the14th International Workshop on Database and Expert Systems Applications, pp. 332-336, Sept. 2003.

[19] Han-Gyu Ko, Eunae Kim, In-Young Ko, and Deokmoon Chang, "Semantically-Based Recommendation by Using Semantic Clusters of Users Viewing History," In Proceedings of the International Conference on Big Data and Smart Computing (BIGCOMP), pp. 83-87, Jan. 2014.

[20] Antonio Maratea, and Alfredo Petrosino, "An Heuristic Approach to Page Recommendation in Web Usage Mining," In Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 1043-1048, 2009.

[21] VVR Maheswara Rao, and V Valli Kumari, "An Efficient Hybrid Successive Markov Model for Predicting Web User Usage Behavior using Web Usage Mining," *International Journal of Data Engineering (IJDE),* vol. 1, no. 5, pp. 43-62, 2010.

[22] A Anitha, "A New Web Usage Mining Approach for Next Page Access Prediction," *International Journal of Computer Applications (IJCA),* vol. 8, no. 11, pp. 7-10, Oct. 2010.

[23] Mehrdad Jalali, Norwati Mustapha, Md Nasir Sulaiman, and Ali Mamat, "WebPUM: A Web-Based Recommendation System to Predict User Future Movements," *Expert Systems with Applications,* vol. 37, no. 9, pp. 6201-6212, Sept. 2010.

[24] Yahya AlMurtadha, Md. Nasir Bin Sulaiman, Norwati Mustapha, and Nur Izura Udzir, "IPACT: Improved Web Page Recommendation System Using Profile Aggregation Based On Clustering of Transactions," *American Journal of Applied Sciences,* vol. 8, no. 3, pp. 277-283, 2011.

[25] Rajhans Mishra, Pradeep Kumar, and Bharat Bhasker, "A Web Recommendation System Considering Sequential Information," *Decision Support Systems,* vol. 75, pp. 1-10, Jul. 2015.

[26] Zdravko Markov, and Daniel T. Larose, "Preprocessing for Web Usage Mining," *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*, Wiley Series on Methods and Applications in Data Mining, Daniel T. Larose, ed., pp. 156-176: John Wiley & Sons, Inc., 2007.