



Improved Web Usage Mining Process and Efficient Online Pattern Prediction

¹Parag Agrawal, ²Suresh K. Shirgave

¹C.S.E, D.Y. Patil College of Engineering & Technology, Kasaba Bawda, Kolhapur, Maharashtra, India

²I.T, D.K.T.E. Society's Textile & Engineering Institute, Rajwada, Kolhapur, Maharashtra, India

Abstract— *Accurate web log mining results and efficient online navigational pattern prediction are utmost crucial for enrichment of web contents and thereupon retaining the visitors. Similar to data mining task, web log mining starts with data cleaning and preparation to eventually discover relevant hidden knowledge which cannot be extracted by usual methods. The quality of the results primarily depends on good quality input data. Thus, data cleaning and pre-processing is of paramount importance. Scalability is the other concerning aspect faced in online prediction, in which improvements have been suggested. Proposed work demonstrates an improved web log mining process and the online navigational pattern prediction. We propose refined time-out based heuristic for session identification. Thereafter, a specific density based algorithm for navigational pattern discovery is suggested. Lastly, a novel approach for efficient online prediction is proposed. Performance of the suggested online prediction technique has also been evaluated using standard measure viz. precision and recall. The results show that the proposed method is able to achieve around 25%-30% better accuracy as compared to conventional KNN method. Also, the proposed method is able to address the scalability problem of conventional KNN method by using it in conjunction with inverted index.*

Keywords—*Web Usage Mining ; Pattern Mining; Clustering; Indexing; Online Prediction*

I. INTRODUCTION

Organizations, companies and institutions are relying more and more on their websites to interact with clients. Retaining current clients and attracting potential ones push these organizations, companies and institutions to look for attractive ways to make their websites more useful and efficient. To achieve this goal, navigational history of the clients that is automatically recorded is analyzed and the Web site is tuned up accordingly. This kind of analysis is referred to as Web Usage Mining (WUM). The patterns extracted by applying WUM techniques can be used to maintain Web sites by improving their content and structure in a way that meets the requirements of both Web site owner and user which will consequently increase the overall profit of the business. The proposed work is centered on the process of WUM and predicting the pattern of the session in real time. Proposed work starts with data cleaning and then focus on better session identification for pattern mining in subsequent steps. Further, a specific density based clustering algorithm is suggested for mining the navigational patterns. Then, an approach is also proposed for navigational pattern prediction of users in real time.

II. RELATED WORK

In WUM for finding navigational patterns, it is mandatory to know what visitors have looked at each time they have visited the Web site. A session is defined as the period of time that a unique user interacts with a Web application. Identifying users' sessions from the Web log is not easy as it may seem. Logs may span long period of time during which visitors may come to the Web site more than once. Therefore, sessions' identification is the task of dividing the sequence of all page requests made by the same user during that period into subsequences. Many approaches have been used by researchers for sessions' identification. According to [1], the most popular session identification techniques use a time gap between requests. It has been mentioned in [2] that many commercial products use 30 min as default time-out threshold. However, many thresholds can be found in the literature. These thresholds vary from 10 min to 2 hours.

Session's identification by referrer has been described in [3]. According to the comparative study conducted by [3], sessions' identification by referrer exhibited very poor performance in the presence of framesets. However, time-oriented heuristics were less affected by the presence of framesets. Earlier research in data pre-processing [4] and [5] presented a comprehensive algorithm that cleans the data, identifies users and sessions. The other research [5] has proposed a complex system to reconstruct sessions. According to the authors real sessions can be obtained using referrer information available in the log.

Association rule discovery or sequential pattern mining can be used to mine for navigational patterns. Both of these two techniques require a user input parameter such as the minimum support. Clustering algorithm like K-means can also be used for mining navigational patterns but it requires the number of clusters to be known by the user beforehand. In the context of WUM, two types of clustering can be performed viz. Clustering users or Clustering pages. Users clustering should result in groups of users visiting similar pages. Researchers have proposed clustering techniques for both types of clustering. The authors of [6] have proposed an algorithm called PageGather to cluster pages based on cliques. The authors

of [7] have suggested the use of K-means clustering algorithm to cluster users. The paper [8] gives a detailed survey of five density based clustering algorithms like DBSCAN, VDBSCAN, DVBSAN, ST-DBSCAN and DBCLASD based on the essential requirements required for any clustering algorithm in spatial data. Each algorithm is unique with its own features. A comparative study is shown in terms of the input parameters, shapes of the cluster, density and the type of the data. [9] and [10] proposed a system for discovering user navigation patterns using a graph partitioning model. An undirected graph based on connectivity between each pair of Web pages was considered and weights were assigning to edges of the graph.

The paper [11] presents the Prediction of User navigation Patterns Using Clustering and Classification (PUCC) from web log data. In the first stage PUCC focuses on separating the potential users in Web log data, in the second stage clustering process is used to group the potential users with similar interest and in the third stage the results of classification and clustering are used to predict the users' future requests.

According to the literature, there are two main approaches in predicting navigational patterns in real time. These two approaches are clustering based and the K-nearest-neighbours approach. T. M. Cover and P. E. Hart [12] propose k-Nearest Neighbour (KNN) in which nearest neighbour is calculated on the basis of value of K, that specifies how many nearest neighbours are to be considered to define class of a sample data point. It has been observed that KNN gives more accurate results than clustering.

III. PROPOSED WORK

Overall architecture of the proposed method is represented in Fig.3.1.

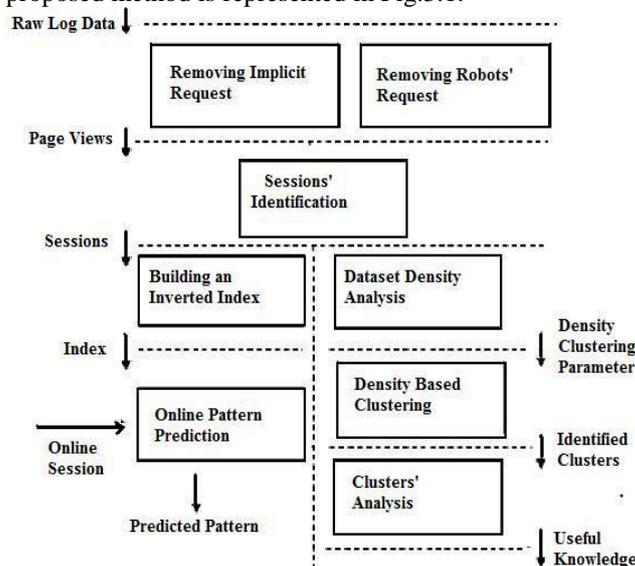


Fig.3.1. Architecture of Proposed System

Proposed framework contains three different components. First, a refined time-out based heuristic for session identification is provided. Second, the usage of a specific Density based algorithm for navigational pattern discovery is shown. Finally, a new approach for an efficient online prediction is also suggested. The Proposed system includes following modules:-

A. Data Cleaning and Preprocessing

The proposed framework accepts raw log data and cleans it to produce page views as shown in the Fig.3.1. The cleaning and pre-processing step consists mainly of removing implicit requests and removing requests made by robots. A Web robot (WR) (also called spider or bot) is a software tool that periodically scans a Website to extract its content. WRs automatically follow all the hyperlinks from a Web page. Some of the traffic in web logs is not generated by real visitors. These robot requests may generate unwanted hits which do not really help in revealing the actual visitor's patterns. Removing WR-generated log entries not only simplifies the mining task, but it also removes uninteresting sessions from the log file.

B. Session's Identification

To produce good quality sessions, the improvement we are proposing at this step of the WUM process is based on the idea that returning visitors without any repeated requests will not help in discovering navigational patterns. Therefore, after detecting sessions using a time-out threshold such as 30 min or 10 min we proceed to check whether the detected sessions of the same user share a pattern or not. Shared pattern is nothing but the common pages between the sessions. For example let's suppose that session1 corresponds to eight visited pages; say P1, P3, P6, P8, P12, P122, P123 and P600. On the other hand session 2 corresponds to 7 visited pages; say P3, P21, P122, P600, P706 and P900 then a shared pattern would be P3, P122 and P600. We will consider shared pattern with and without considering page order. If a shared pattern exists then the identified sessions get approved. Otherwise, the sequence which is being tried to be divided into sessions will be skipped.

C. Refined Pattern Mining Process

After the raw log data has gone through all steps of cleaning, pre-processing and session’s identification, it is ready for mining navigational pattern. In the proposed work, DBSCAN [13] a density based clustering algorithm, is applied. Unlike K-means clustering algorithm, this algorithm detects outliers and it also does not require the number of clusters to be known by the user beforehand. However, it requires another input parameter and to guess a good value for this parameter is very hard for the user. Further, DBSCAN is very sensitive for this input parameter. For this reason, the proposed approach makes use of another algorithm called OPTICS [14] which helps in selecting an appropriate value for the input parameter of DBSCAN. Before clustering, identified sessions are taken by the OPTICS as input and outputted them in a certain order according to their closeness to each other. OPTICS gives a clear picture of the dense regions within the dataset. This makes it very easy for user to select the appropriate density parameter i.e. epsilon for DBSCAN to cluster the dataset.

D. Online Prediction

To predict online patterns, we propose a scalable kNN-based approach. A classical kNN-based approach would look for k nearest neighbors of a point in a dataset by computing distances between that specific point and all other points in the dataset. Then these points are sorted according to their instances to the original point in an ascending order and only the first k points are outputted. In the context of an online application, this might be acceptable only when the size of the dataset is relatively small. However, if the dataset size is huge this becomes unacceptable. The proposed scalable kNN solution is based on the idea of using indexes. Here sessions are viewed as text documents and an online session is considered as a query. Therefore, looking for the k nearest sessions to an online session will be translated to looking for the documents which are most relevant to the current query. The suggested approach consists of the adoption of the well known technique used in information retrieval systems, namely TF-IDF combined with the cosine similarity measure to find the closest sessions to a current online session. Also in order to speed up the above mentioned search process, an inverted index is built from all sessions. This method is depicted in Fig.3.1.where identified sessions go through an inverted index process then they are compared with an online session.

IV. EXPERIMENTAL RESULTS

This section describes the experiments conducted and evaluates the results obtained. NASA space center web server log have been used to conduct series of tests. This is a well known dataset used by many researchers. The log data was recorded according the common log format and it spans the whole month of July 1995. The computer that is used to conduct all experiments has the characteristics shown in Table I. All algorithms have been implemented using java programming language. The process begins by filtering out all noisy data, implicit requests, entries related to errors and any entries related to requests made by robots. Tables II and III provide an overview of the work done at this stage.

TABLE I TESTING ENVIRONMENT

Main Memory	3 GB
CPU	2.2 GHz
Operating System	Windows Vista Home Basic

TABLE II DATASET USED AND NUMBER OF PAGE VIEWS IDENTIFIED

Dataset	Period	Number of entries	Number of page views
NASA web log	Month of July 1995	1,891,715	523,160

TABLE III A SAMPLE OF ROBOTS DETECTED IN DATA SET

NASA dataset
hydra.wcupa.edu
bang.engin.umich.edu
wfr-20-1.rz.uni-frankfurt.de
query2.lycos.cs.cmu.edu

A. Session’s Identification

For session identification, the improvement that is proposed is based on the idea that returning visitors without any repeated requests will not help in discovering navigational patterns. Therefore, after detecting sessions using a time-out threshold such as 10 min we proceed to check whether the detected sessions of the same user share a pattern or not. If a shared pattern existed we approve the identified sessions. Otherwise, the sequence we are trying to divide into sessions is skipped.

The following investigation has been conducted to see the effect of imposing our suggested additional constraint on the quality and quantity of the identified sessions. In our investigation we have used 10 min time-out threshold. For this threshold we keep 3 as the required minimum length of a shared pattern. Also for 10 min threshold and for the value of the required minimum length of a shared pattern, we introduced another variable R which varies from 10% to 100%. A 100% value of R means that all concerned sessions must share a pattern. A 50% value means that only half of the concerned sessions must share a pattern and so on. For instance, the first line of Table V corresponds to the results obtained by our enhanced heuristic using a 10 min time-out threshold where all (100%) concerned sub-sequences must share a pattern of length at least 3. Each time we run our sessionizer, we record the total number of correctly identified sessions, the total number of identified sessions and the number of false positives. In order to be able to compare the quality of the results returned by our proposed heuristic and the results returned by the classical time-out based heuristic, first we have run the classical time-out based heuristic for 10 min threshold and the results are as shown in Table IV. Second, we have run our proposed sessionizer using the 10 min time-out threshold and the results are shown in Table V. It is observed that even though the classical heuristic was able to correctly identify a high number of sessions it has also included a very high number of false positives which is 30.59% as shown in Table IV. This is due to blindly dividing a sequence into a series of sessions based on a predefined threshold only. These high numbers of false positives are nothing but a source of misleading results. On the contrary, the results obtained by our proposed heuristic have a very low number of false positives. The lowest ratio we have got is 2.6% as shown in Table V. We noticed that when we are reducing the number of false positives to its minimum, the number of correctly identified sessions is reduced as well.

TABLE IV RESULTS OF CLASSICAL TIME-OUT BASED SESSIONIZER

Threshold	Correctly identified sessions	False positives	Total	Ratio of false positives/Total(%)
10 min	16,173	7129	23,302	30.59

TABLE V ENHANCED SESSIONIZER RESULTS; TIME-OUT= 10 MIN, LENGTH SHARED PATTERN>=3

R(%)	Correctly identified sessions	False positives	Total identified sessions	Ratio of false positives/Total(%)
100	7835	210	8045	2.6
90	7835	210	8045	2.6
80	7900	222	8122	2.7
70	8230	280	8510	3.2
60	9140	321	9461	3.4
50	9400	350	9750	3.5
40	9450	350	9800	3.5
30	9604	440	10044	4.3
20	11090	505	11595	4.3
10	12105	634	12739	4.9

Therefore, the correctly identified session might not be representative of all the real navigational patterns. This concern is similar to the one triggered by the elimination of all requests coming from hosts having the term proxy. To conclude, the experiments conducted showed clearly that we can get good quality results by taking into consideration not only the time gap between two consecutive requests of the same user but also by checking whether there exist any shared pattern between the concerned sub-sequences. The idea behind our proposed sessionizer can be plugged to any non-time-based heuristics as well, such as the one based on the referrer if the log is following the extended format.

B. Clustering

The raw data has gone through all steps of cleaning, pre-processing and sessions' identification before it was ready for our clustering algorithm. In fact, it has gone into another step before clustering. It has been analyzed by OPTICS [2] first. OPTICS has taken the identified sessions in a form of a matrix as input and outputted them in a certain order according to their closeness to each other. The number of columns of the input matrix is the number of pages in the website and its number of rows is the number of sessions. In other words, each session is represented by a vector.

We have fed OPTICS with a binary matrix. Each cell in the binary matrix represents whether a specific page has been requested during a specific session or not, 1 for yes and 0 for no as shown in Fig. 4.1.

	P1	P2	P3	P4
	0	1	0	1
	1	1	1	0
	0	1	1	1

Fig 4.1. An example of a binary matrix that can be used as input for OPTICS

The results of OPTICS have been plotted as shown in Figs. 4.2. In Fig. 4.2 the calculated reach-ability distances of all points (sessions) translated into a binary matrix are plotted according to the order OPTICS has put them in. When we draw a horizontal line at a value 0.0042 from the Y-axis, we see that this value results in two different clusters. OPTICS gives a clear picture of the dense regions within the dataset. This makes it very easy for us to select the appropriate density parameter epsilon for DBSCAN to cluster the dataset.

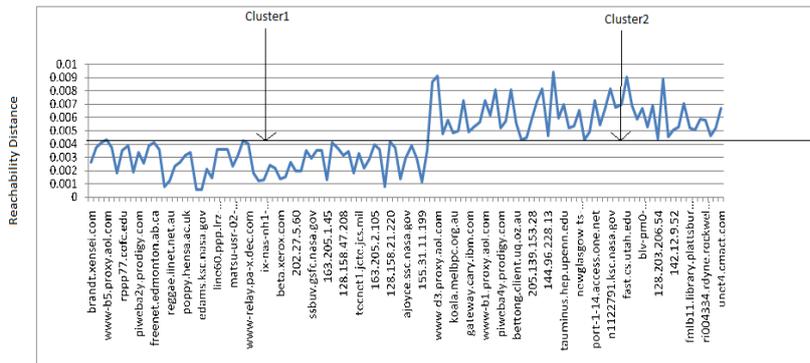


Fig 4.2. Reachability distances graph for OPTICS

C. Online Prediction

At this final phase and as we have mentioned in the previous section, each session is represented by a vector. Each single value in the vector is the TF-IDF value of the corresponding page if it does exist in the session or it is zero if the page does not exist in the session. The goal of the experiments conducted at this stage is to show how much time can be saved by the usage of an inverted index. Also, the goal is to show that the way the inverted index is built helps in getting only sessions that contain exactly the pages contained in the sliding window, in the exact same order and as a contiguous sequence. So here we will test the efficiency and accuracy of the proposed KNN-based online pattern prediction.

As an example, we have considered the following sequence as the actual content of our sliding window, p47, p47. The results obtained in terms of running time and in terms of the most relevant sessions are shown in Tables VI and VII. To check how much we will save in running time by using an inverted index, we conducted ten experiments as shown in Table VI.

TABLE VI KNN RESPONSE TIME IN MILLISECONDS

No. of Sessions	No Index	With Index
1000	1015	30
2000	4327	36
3000	8312	40
4000	12570	43
5000	20412	47
6000	36412	57
7000	47412	61
8000	61129	67
9000	79618	73
10000	85922	73

First, we took the first 1000 sessions from our original dataset and we measured how long it took to get the five nearest neighbors by using an index and without using an index. Then we took the first 2000 sessions and measured also how long it took to get results in both cases. We kept adding each time 1000 sessions till we reached the size of ten 10,000 sessions. The results shown in Table VI speak for themselves and there is a very huge difference in running time. Also when it comes to the quality of the results returned in both cases and as we can see in Table VII, the usage of the inverted index helps in finding the closest sessions that contain exactly the query.

TABLE VII KNN RESULTS

Without use of Index	With the use of Index
P47,P47	P47,P47
P47,P47,P47, P47	P47,P47,P47,P49,P49,P49,P244,P244, P244
P47,P47,P47,P2,P2,P2,P8,P8 ,P8,P11,P11, P11	P5,P64,P16,P45,P47
P47,P47,P47,P11,P11,P11	P64,P2,P47

P47,P47,P47,P11,P11,P11	P2,P2,P2,P4,P4,P4,P44,P44,P44,P65,P65,P65,P495,P495,P495,P493,P493,P493,P76,P76,P76,P47,P47,P47,P2,P2,P2,P4,P4,P4
-------------------------	---

For evaluating the accuracy of the method used in online prediction, standard measures like precision and recall have been used and are defined below:

Assume that we have transaction t (taken from the evaluation set) viewed as a set of Page views, and that we use a window w subset of t (of size $|w|$) to produce a predicted patterns set P using the proposed algorithm (KNN with index) in online prediction. Then the precision of P with respect to t is defined as:

$$\text{precision}(P,t) = \frac{|P \cap (t - w)|}{|P|}$$

and the Recall of P with respect to t is defined as:

$$\text{recall}(P,t) = \frac{|P \cap (t - w)|}{|t - w|}$$

Fig.4.3. gives the comparison of KNN without index and KNN with index method based on the metric precision. Fig.4.4. gives the comparison of performance of KNN with index and KNN without index based on the metric recall.

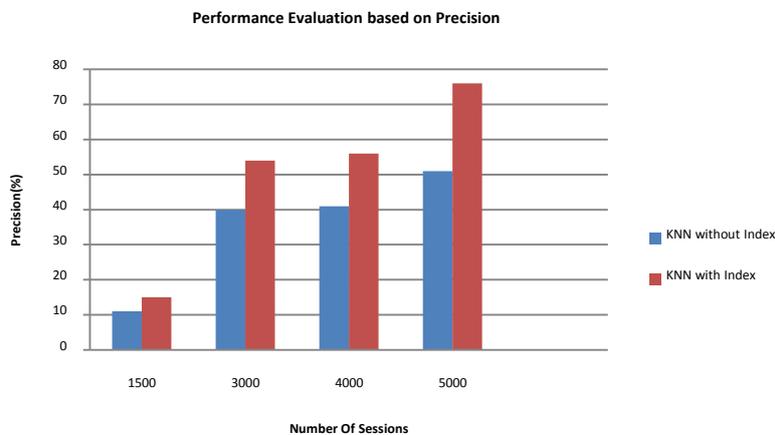


Fig.4.3. Performance comparison based on precision

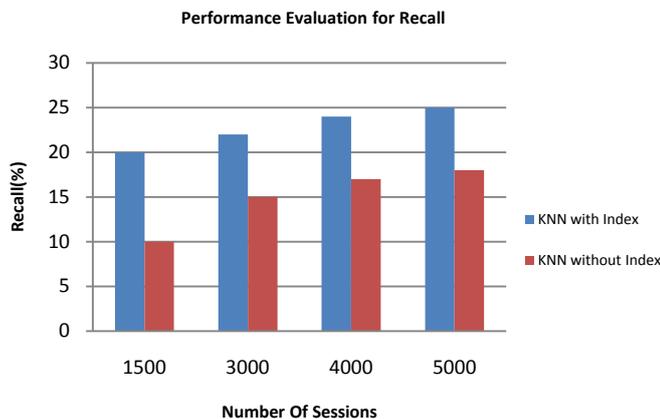


Fig.4.4. Performance comparison based on recall

Fig.4.3, gives the comparison of conventional KNN and KNN with index based on the metric precision. When number of sessions increases, precision of KNN with index is high when compared with conventional KNN. KNN with index achieves 75% precision whereas conventional KNN achieves only 50% precision.

Fig.4.4, gives the comparison of conventional KNN and KNN with index based on the metric recall. When number of sessions increases, recall of KNN with index is high when compared with conventional KNN. KNN with index achieves 25% recall whereas conventional KNN achieves only 17% recall.

V. CONCLUSION

In this paper, we proposed an efficient framework for web log mining and for online navigational behavior prediction. We have reviewed all steps of this process and we have analyzed existing approaches and made an effort to make a contribution at each step. First our framework accepts the raw log data and cleans it to produce page views. The page

views enter a second phase of session identification. The proposed time-out heuristic has less value of false positive as shown in Table V, as compared to existing time-out heuristic approach which means good quality input data for mining algorithms. At the refined pattern mining step we used DBSCAN clustering that does not require a prior knowledge about the number of clusters and that detects outliers. We implemented this clustering algorithm and we also implemented OPTICS and we showed how OPTICS can help in selecting the appropriate input for DBSCAN. The combination of the two algorithms helps in detecting all patterns, not only the most frequent ones.

The last part of our suggested framework is an efficient online navigational behavior prediction component. Online pattern prediction is very useful. For example, it helps in reducing the server's response time by caching pages that are more than likely to be requested by a visitor. Also, it helps in recommending products, links, online services and so on. We proposed an efficient kNN based approach to do online pattern prediction. Here we used KNN approach for finding the relevant sessions with respect to online session and also used inverted index for making this process fast and accurate.

REFERENCES

- [1] Zidrina Pabarskaite and Aistis Raudys. A process of knowledge discovery from web log data: Systematization and critical review. *Journal of Intelligent*.
- [2] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava, et al. Data preparation for mining world wide web browsing patterns. *Knowledge and information systems*, 1(1):5-32, 1999.
- [3] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In *WEBKDD 2002-Mining Web Data for Discovering Usage Patterns and profiles*, pages 159-179. Springer, 2003.
- [4] Zhang Huiying and LiangWei. An intelligent algorithm of data pre-processing in web usage mining. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 4, pages 3119-3123. IEEE, 2004.
- [5] Jie Zhang and Ali A Ghorbani. The reconstruction of user sessions from a server log using improved time-oriented heuristics. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 315-322. IEEE, 2004.
- [6] Mike Perkowitz and Oren Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *AAAI/IAAI*, pages 727-732, 1998.
- [7] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data mining and knowledge discovery*, 6(1):61-82, 2002.
- [8] M Parimala, Daphne Lopez, and NC Senthilkumar. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology*, 31:59{66, 2011.
- [9] Mehrdad Jalali, Norwati Mustapha, Ali Mamat, and Md Nasir B Sulaiman. A new clustering approach based on graph partitioning for navigation patterns mining. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1-4. IEEE, 2008.
- [10] Raymond Kosala and Hendrik Blockeel. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter*, 2(1):1-15, 2000.
- [11] V Sujatha et al. Improved user navigation pattern prediction technique from web log data. *Procedia Engineering*, 30:92-99, 2012.
- [12] Thomas Cover and Peter Hart. Nearest neighbor pattern Classification. *Information Theory, IEEE Transactions on*, 3(1):21-27, 1967.
- [13] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226-231,1996.
- [14] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jorg Sander. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49-60, 1999.