ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

# New Framework for Web Access Prediction

**Sawan Bhawsar, Kshitij Pathak, Vibhor Patidar**

***Abstract* -In recent years, the problem of modeling and predicting a user's surfing behavior on a web-site has obtained a lot of research interest.Markov models& its variations have also been used to analyze web navigation behavior of users. A user's web link transition on a particular website can be modeled using first, second-order or higher-order Markov models and can be used to make predictions regarding future navigation and to personalize the web page for an individual user. As it can be used to improve the web cache performance, recommend related pages , improve search engines , understand and influence buying patterns , and personalize the browsing experience. Modeling user web navigation data is a challenging task that is continuing to gain importance as the size of the web and its user-base increases. Data characterizing web navigation can be collected from the server or client- based log files, enabling the reconstruction of user navigation sessions.**

***Index Terms*—web access prediction, markov model, page rank.**

## I. INTRODUCTION

Basically the web mining is the task related to the database to collect the information related to the web page access in the form of web session for the future analysis. The web mining enables the prediction of page access and the analysis of user navigation Behavior.

## II. TAXONOMY OF WEB MINING

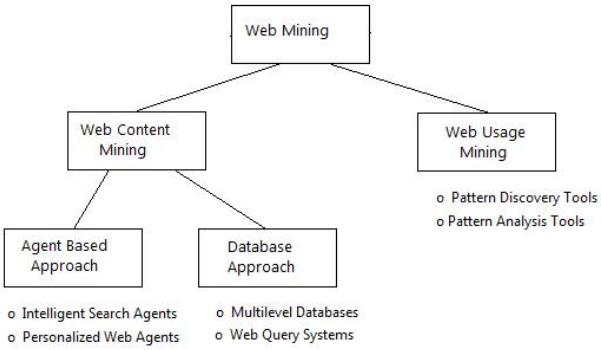



**Fig1: Taxonomy of Web Mining.**

Here our task is related to the web usage mining which basically Consist task related to the use of web where the access of the web will considered and the navigation pattern and the prediction operation will performed in the mining of this kind we will use the database in the form of the web log files and we will generate the results on the basis of the database given.Markov models have been used for studying and understanding stochastic processes, and were shown to be well-suited for modeling and predicting a user's browsing behavior on a web-site.In general, the input for these problems is the sequence of web-pages that were accessed by a user and the goal is to build Markov models[2] that can be used to model and predict the web-page that the user will most likely access next[3]. In many applications, first-order Markov models are not very accurate in predicting the user's browsing behavior, since these models do not look far into the past to correctly discriminate the different observed patterns.As a result, higher-order models are often used. Unfortunately, these higher-order models have a number of limitations associated with high state-space complexity, reduced coverage, and sometimes even worse prediction accuracy.One method proposed toovercome the problem is the clustering and cloning to duplicate the state corresponding to page that require a longer history to understand the choice of link that users made.Initially when the web log is not available means the web site is newly launched the prediction or the navigation decision will mad on the page rank our page rank strategy will also used to resolve the ambiguity of the model.Our model will use the basic strategy for the preparing the model is the page rank , and variable length markov model, the problem of ambiguity in the markov model will solve on the basis of the page rank and the page rank will also used in the initial stage when the web log file is not available.

## III. MARKOV MODEL AND PAGE RANK

As discussed in the introduction, techniques derived from Markov models have been extensively used for predicting the action a user will take next given the sequence of actions he or she has already performed.For this type of problems, Markov models are represented by three parameters < A; S; T >, where A is the set of all possible actions that can be performed by the user; S is the set of all possible states for which the Markov model is built; and T is a |S|*|A| Transition Probability Matrix (TPM),

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

where each entry tij corresponds to the probability of performing the action j when the process is in state i . The state-space of the Markov model depends on the number of previous actions used in predicting the next action.The simplest Markov model predicts the next action by only looking at the last action performed by the user. In this model, also known as the first-order Markov model, each action that can be performed by a user corresponds to a state in the model. A somewhat more complicated model computes the predictions by looking at the lasttwo actions performed by the user.This is called the second-order Markov model, and its states correspond to all possible pairs of actions that can be performed in sequence. This approach is generalized to the Kth -order Markov model[7], which computes the predictions by looking at the last K actions performed by the user, leading to a state-space that contains all possible sequences of K actions.

For example, consider the problem of predicting the next page accessed by a user on a web site. The input data for building Markov models consists of web-sessions, where each session consists of the sequence of the pages accessed by the user during his/her visit to the site.In this problem, the actions for the Markov model correspond to the different pages in the web site,and the states correspond to all consecutive pages of length K that were observedin the different sessions. In the case of first-order models, the states will correspond to single pages, in the case of second-order models, the states will correspond to all pairs of consecutive pages, and so on.Once the states of the Markov model have been identified, the transition probability matrix can then be computed. There are many ways in which the TPM can be built.. For example consider the web-session A3; A5; A2; A1; A4shown in Figure 1. If we are usingfirst-order Markov model then each state is made up of a single page, so the first page A3 corresponds to the state s3. Since page A5 follows the state s3 the entry t35 in the TPM will be updated. Similarly, the next state will be s5 and the entry t52 will be updated in the TPM. In the case of higher-order model each state will be made up of more than one actions, so for a second-order model the first state for the web-session WS2 consists of pages A3, A5 and since the page A2 follows the state A3, A5 in the web-session the TPM entry corresponding to the state A3, A5 and page A2 will be updated.

Once the transition probability matrix is built making prediction for web sessions is straight forward. For example, consider a user that has accessed pages A1; A5; A4. If we want to predict the page that will be accessed by the user next, using a first-order model, we will first identify the state s4 that is associated with page A4 and look up the TPM to find the page Ai thathas the highest probability and predict it. In the case of our example the prediction would be page A5.

Web Session:

A3, A2, A1
A3, A5, A2, A1, A4
A4, A5, A2, A1, A5, A4
A3, A4, A5, A2, A1
A1, A4, A2, A5, A4

**Table1: First order transition Probability Metrics**

| First order | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A1 | 0 | 0 | 0 | 2 | 1 |
| A2 | 4 | 0 | 0 | 0 | 1 |
| A3 | 0 | 1 | 0 | 1 | 1 |
| A4 | 0 | 1 | 0 | 0 | 2 |
| A5 | 0 | 3 | 0 | 2 | 0 |

Similarly the second order probability metrics can be developed, for the creation of the second order model the sequence of two consecutive pages are considered. For the example in the above web session what the count of A2 followed the sequence A4, A5 For all the pair of the two page sequence the probabilistic model is given below.

**Table2: Second order transition Probability Metrics.**

| Second order | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A1, A4 | 0 | 1 | 0 | 0 | 0 |
| A1,A5 | 0 | 0 | 0 | 1 | 0 |
| A2,A1 | 0 | 0 | 0 | 1 | 1 |
| A2,A5 | 0 | 1 | 0 | 0 | 0 |
| A3,A2 | 1 | 0 | 0 | 0 | 0 |
| A3,A5 | 0 | 1 | 0 | 0 | 0 |
| A4,A2 | 0 | 0 | 0 | 0 | 1 |
| A4,A5 | 0 | 2 | 0 | 0 | 0 |
| A5,A2 | 3 | 0 | 0 | 0 | 0 |
| A3,A4 | 0 | 0 | 0 | 0 | 1 |

Page Rank:PageRank is a numeric value that represents how important a page is on the web. When one page links to another page, it is effectively casting a vote for the other page. The more votes that are cast for a page, the more important the page must be.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

Also, the importance of the page that is casting the vote determines how important the vote itself is. How important each vote is taken into account when a page's Page Rank is calculated.PageRank is Google's way of deciding a page's importance. It matters because it is one of the factors that determine a page's ranking in the search results. It isn't the only factor that Google uses to rank pages, but it is an important one.Brin and Page have simply transferred this premise to its web equivalent: the importance of a web page can be judged by the number of hyperlinks pointing to it from other web pages

Formula to find the Page Rank

It may look daunting to non-mathematicians, but the PageRank algorithm is in fact elegantly simple and is calculated as follows:

$$PR(A) = (1-d) + d\,(PR(B)/C(B) + ...+ PR(Tn)/C(Tn))$$

Where PR(A) is the page rank of a page A, PR(B) is the Pagerank of the page B C(B) is the number of outgoing links from the page B. d is the damping factor in the range 0<d<1, usually set to 0.85.

The PageRank of a web page is therefore calculated as a sum of the PageRanks of all pages linking to it (its *incoming links*), divided by the number of links on each of those pages (its *outgoing links*).

## IV. METHODOLOGY

The creation of the dynamic markov model based on clustering technique. Considering the web site of seven web Pages {A1,A2,A3,…..A7}

And the following web sessions with NOS as the number of times the session performed-

**Table3: Performed session with NOS.**

| Performed Session | NOS |
|---|---|
| A1; A2 ;A3 | 3 |
| A1; A2; A4 | 2 |
| A5; A2; A3 | 1 |
| A5; A2; A4 | 1 |
| A6; A2; A3 | 2 |
| A6; A2; A4 | 1 |
| A1; A2; A4; A7 | 1 |
| A5 ; A2 ; A4 ; A7 | 3 |
| A6 ; A2 ; A4 ; A7 | 2 |

..

With corresponding to the above web session the first order model will be calculated as follows-

$W_i$ =no of times the page traversed

$W_{ij}$ =number of times the link visited;

Pij (Transition Probability)= $W_{ij}/W_i$;

On the basis of above probability formula the transition matrix is as follows

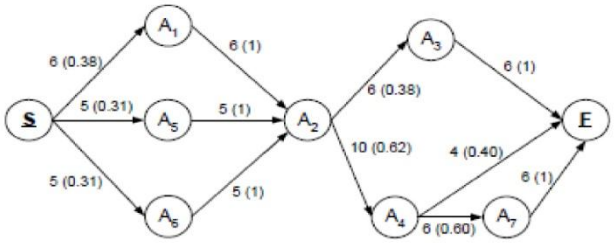



**Fig2: first order HPG model.**

Probability for the S-A1 is calculated as

6(0.38)=6(no of time link visited ) (6/(no of times the s visited)16)

same as all above link probability calculated.

We let $P_{ij} = W_{ij}/W_i$ be the First-order transition probability from $A_i$to $A_j$ ,and $P_{ijk} = W_{ijk}/W_{ij}$, be the second-order transition probability. Also, the accuracy threshold, P sets the highest admissible difference between a first-order and a second-order probability a model is said to be accurate if there is no link that violates the constraint set by $\gamma = 0.1$ .

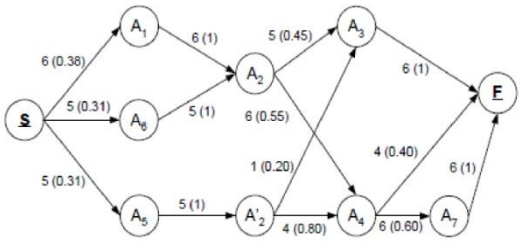
Now to check the accuracy of the model the combination of the states will be used as follows

If, $|P_{ijk} - P_{jk}| > 0$ then the State j is not accurate like

In the example given in Figure, the user's past navigation behavior implies that $P_{123}$- $P_{124}$ = 0.5. Therefore, for $\gamma$ = 0.1, the state *A*2 is not accurate, since $P_{123}$-$P_{23}$> 0.1, and needs to be cloned. To clone state *A*2, we let each in-link define a vector of second-order probabilities each of the vector's components Corresponds to an out-link from state *A*2. In the example, state

*A*2 has three in-links and two out-links, inducingthree vectors of second-order probabilities for I={3,4} we have $P_{12i}$ = {0.5,0.5} and$P_{52i}$ ={0.2,0.8} and $P_{62\,I}$ ={0.4,0.6}.

**Fig3:The second order HPG model obtained when applying the dynamic clustering method with $\gamma$ = 0.1 to the first-order model given in example.**



ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

The method applies a k-means clustering algorithm to the collection of second-order vectors, in order to identify groups of similar vectors with respect to γ . Figure shows the result of applying the method to state A2. Since $|P_{1,2\ i}$-$P_{6,2\ i}|$<0.1 for i={3,4} links from A1 and A6 are assigned to one clone, the link from A5 is assigned to the other clone. The transition counts for the out-links are updated as follows

$W_{23}= W_{123}+W_{623}$

$W_{24}= W_{124}+W_{624}$

$W_{2'3}=W_{523}$

$W_{2'4}=W_{524}$

Note that, state A4 is accurate, since all its in-links have an equivalent source state, and, moreover, every state having just one out-link is accurate by definition. Therefore, the model given in Figure 2 accurately represents every second-order transition probability.Similarly the third order transition probability is calculated and the higher order model is created and the process is followed until the accurate model is not achieved if it is difficult to achieve the accurate model the third order model will be considered.Now After finding the higher order model the transition probability model is considered to predict the web page prediction.

4.1 Finding the Page Rank (The Topology Used By Google)Simplified algorithm

Assume a small universe of four web pages: **A**, **B**, **C** and **D**. The initial approximation of PageRank would be evenly divided between these four documents. Hence, each document would begin with an estimated PageRank of 0.25.

In the original form of PageRank initial values were simply 1. This meant that the sum of all pages was the total number of pages on the web. Later versions of PageRank (see the below formulas) would assume a probability distribution between 0 and 1. Here a simple probability distribution will be used- hence the initial value of 0.25.If pages **B**, **C**, and **D** each only link to **A**, they would each confer 0.25 PageRank to **A**. All PageRank **PR ( )** in this simplistic system would thus gather to **A** because all links would be pointing to **A**.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

This is 0.75.

Again, suppose page **B** also has a link to page **C**, and page **D** has links to all three pages. The value of the link-votes is divided among all the outbound links on a page. Thus, page **B** gives a vote worth 0.125 to page **A** and a vote worth 0.125 to page **C**. Only one third of **D**'s PageRank is counted for A's PageRank (approximately 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

Counted for A's PageRank (approximately 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

In other words, the PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the normalized number of outbound links **L( )** (it is assumed that links to specific URLs only count once per document).

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

In the general case, the PageRank value for any page **u** can be expressed as:

$$PR(u) = \sum_{v \in Bu} \frac{PR(v)}{L(v)}$$

i.e. the PageRank value for a page **u** is dependent on the PageRank values for each page **v** out of the set **Bu** (this set contains all pages linking to page **u**), divided by the number*L*(*v*) of links from page**v**.

Damping factor

The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor *d*. various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.
The damping factor is subtracted from 1 (and in some variations of thealgorithm, the result is divided by the number of documents in the collection) and this term is then added to the product of the damping factor and the sum of the incoming Page Rank scores. That is,

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$

Or (*N* = the number of documents in collection)

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

A page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. However, the solution is quite simple. If the random surfer arrives at a sink page, it picks another URL at random and continues surfing again.When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability of usually $d$ = 0.85, estimated from the frequency that an average surfer uses his or her browser's bookmark feature.So, the equation is as follows:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

where $p_1,p_2,...,p_N$ are the pages under consideration, $M(p_i)$ is the set of pages that link to $p_i$, $L(p_j)$ is the number of outbound links on page $p_j$, and $N$ is the total number of pages.

The PageRank values are the entries of the dominant eigenvector of the modified adjacency matrix. This makes PageRank a particularly elegant metric: the eigenvector is

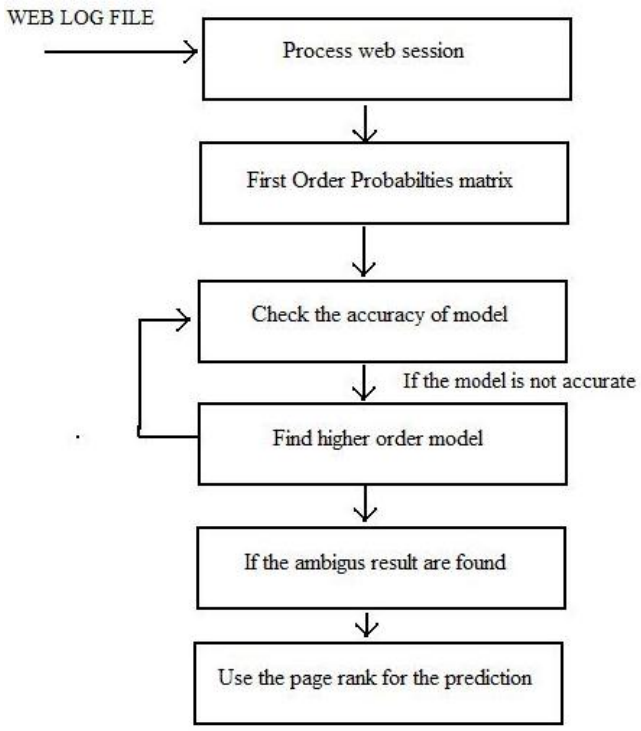
$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

Where **R** is the solution of the equation

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_2,p_1) & \cdots & \ell(p_N,p_1) \\ \ell(p_1,p_2) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_1,p_N) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

So as above the page rank is calculated and this page rank is used.

**ALGORITHM**: **Steps Followed in the model**




4.2 Algorithm for finding the page rank

PR (A) is the rank for the page A, L(O) is the number of outlink to page A. d is the dumping Factor.

For each page of web site find the following

$$PR(A) = \frac{(1-d)}{n} + \sum \frac{PR(O)}{L(O)}$$

4.3 Algorithm for finding the dynamic order model

$W_i$- the Weight of state W,$W_{ij}$ is the Weight of link i to j, P is the transition probability. accuracy threshold $p$=0.1.

n number of states

Find the First Order transition Probability Matrix

Repeat until the accurate model

For each value of x=1 …. N

For state x

If $|P_{ixk} - P_{xk}|$>0.1 then state x is not accurate make clone X and X'

(To find the new link weight)

For each in link y and y' and out link z,z'

IF $|P_{yxz}-P_{y'xz}| < 0.1$ then keep y, y' in same clone

Update the weight

$W_{xz}=W_{yxz}+ W_{y'xz}$;

End

## V. CONCLUSION

The primary goal of this paper isto identify frame work which may be used to predict a web page access for a website or web application at server side. The frame work included the concept of variable length markov model and page rank , page rank concept may be used when the website is newly launched and the weblog is not sufficiently created so page rank may be used to predict the page and it may be also used when the ambiguity will arrived in the markov model.

## REFERENCES

[1]. J. Borges A data mining model to capture user web navigation patterns. PhD thesis, University College London, London University, 2000.

[2]. Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 1991

[3]. V.N. Padmanabham and J.C. Mogul. Using predictive prefetching to improve world wide web latency. *Computer Communication Review*, 1996

[4]. J. Borges and M. Levene. A dynamic clustering-based markov model for web usage mining. cs.IR/0406032, 2004.

[5]. Eugene Charniak. Statistical Language Learning. The MIT Press, 1996.

[6]. X. Chen and X. Zhang. A popularity-based prediction model for web prefetching. Computer, 36(3):63{70, March 2003.

[7]. M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. ACM Transactions on Internet Technology, 4(2):163{184, 2004.

[8]. X. Dongshan and S. Junyi. A new markov model for web access prediction. Com- puting in Science and Engineering, 4(6):34{39, November/December 2002.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 2 , Issue 1

[9]. M. Levene and G. Loizou. Computing the entropy of user navigation in the web. Int. Journal of Information Technology and Decision Making, 2:459{476, 2003.

[10]. B. Mobasher. Web usage mining and personalization. In Munindar P. Singh, editor, Practical Handbook of Internet Computing. Chapman Hall & CRC Press, 2004.

[11]. Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: conceptual frame- work and case study. Arti¯cial Intelligence, 118(2000):245{275, 2000.

[12]. J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In Proc. of the 2nd Usenix Symposium on Internet Technologies and Systems, pages 139{150, Colorado, USA, October 1999.

[13]. Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: Extracting usable structures from the web. In *CHI-96*, 1996.

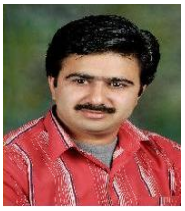
[14]. Ramesh R. Sarukkai. Link prediction and path analysis using markov chains. In *Ninth International World Wide Web Conference*, 2000.

[15]. S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. In *Seventh International World Wide Web Conference*, 1998.

[16]. Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *22nd annual international ACM SIGIR conference on Research and development in information retrieval*,1999.

[17]. J. Borges and M. Levene. Data mining of user navigation patterns. In B. Masand and M. Spliliopoulou, editors, Web Usage Analysis and User Pro¯ling, Lecture Notes in Arti¯cial Intelligence (LNAI 1836), pages 92{111. Springer Verlag, 2000

[18].J. Borges and M. Levene. An average linear time algorithm for web usage mining. Int. Jou. of Information Technology and Decision Making, 3(2):307{319, June 2004.



Er. Sawan Bhawsar, has been working as a Faculty in the Department of Information Technology, Malwa Institute of science &Technology Indore, He has a rich academic and industry experience of more than 5 years, He has participated and contributed in various national/international conferences/seminars & workshops. His area of interest and research are: Computers Networking, Data Communication, Data Mining, Algorithms, etc.

Prof. Kshitij Pathak has been working as a Faculty in the Department of Information Technology, Mahakal Institute of Technology Ujjain since August 2005. He is also heading the Software Cell of the institute. He served as a faculty in Ujjain Engineering College Ujjain during 2003 to 2005. He has a rich academic and research experience of more than 7 years.

Mr. Pathak has a meritorious academic career and did his B.E. (CSE) and M.Tech (IT) with distinction. He is actively involved in Management Committee of Computer Society of India – Ujjain Chapter for the duration of 2011-12. He is instrumental in developing an ERP package for MIT group of Institutions. He has participated and contributed in various National/International Conferences/Seminars/Workshops/FDPs'. His area of interest and research are: Data Mining, Algorithms, Computational Learning and Software Engineering etc.

Er. Vibhor Patidar, has been working as a Faculty in the Department of Computer Science , TRUBA College of Engineering & TechnologyIndore, He has a rich academic and industry experience of more than 4 years, He has participated and contributed in various national/international conferences/seminars & workshops. His area of interest and research are: Java Programming, DBMS, software engg., Computers Networking, Data Communication, Data Mining, Algorithms, etc