# WEB PAGE ACCESS PREDICTION USING HIERARCHICAL CLUSTERING BASED ON MODIFIED LEVENSHTEIN DISTANCE AND HIGHER ORDER MARKOV MODEL

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**CH CHARISHMA (1210315307)**

**S HANISHA SREE(1210315315)**

**HEMANTH KUMAR (1210315324)**

**FARHAT KHAN(1210315365)**

Under the Esteemed Guidance of

**Dr. G.A.RAO , ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**Gandhi Institute of Technology and Management (GITAM)**
**VISAKHAPATNAM – 530045**
**2015-2019**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be university)**

# CERTIFICATE

This is to certify that the final year project report entitled “**Web Page Access Prediction Using Hierarchical Clustering Based on Modified Levenshtein Distance and Higher Order Markov Model**” submitted by **CH CHARISHMA**(1210315307),**S HANISHA SREE** (1210315315), **HEMANTH KUMAR**(1210315322) and **FARHAT KHAN** (1210315365) in complete fulfillment for the award of B.Tech in Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM Deemed to be University, Visakhapatnam, during the year 2018-2019 is a record of bonafide work carried out under the guidance and supervision of

**Project Guide**

Dr.G.A.Rao
Assistant Professor
Department of C.S.E

GITAM Deemed to be University

Visakhapatnam- 530045

# DECLARATION

The Project “**Web Page Access Prediction Using Hierarchical Clustering Based on Modified Levenshtein Distance and Higher Order Markov Model”** is submitted in complete fulfillment of the requirements for the award of the degree of B.Tech (Dept. of Computer Science and Engineering). The results embedded in this documentation have not been submitted to any other University, or Institute for the award of any other degree or diploma.

**CH CHARISHMA(1210315307)**

**S HANISHA SREE(1210315315)**

**K HEMANTH KUMAR(1210315322)**

**FARHAT KHAN(1210315365)**

# ACKNOWLEDGMENT

We are pleased to acknowledge our project guide, Dr.G.Appa RAO, Assistant Professor, GITAM Institute of Technology, GITAM Deemed to be University for his guidance and supervision during this project work.

We are also grateful to Dr. Konala Thammi Reddy, Professor, HOD, Department of Computer Science and Engineering, for giving us permission to do this project work.

We are greatly indebted to our Project reviewers Dr. Ch Shanthi, Professor and Dr. S Anuradha, Assistant Professor for providing their valuable advice, constructive suggestions, positive attitude, and encouragement without which it would not been possible to complete this project. We hope that we can build upon the experience and knowledge that we have gained and make a valuable contribution towards the IT industry in coming future.

s

# ABSTRACT

Web Page access prediction is a challenging task in the current situation, which pulls the attention of many researchers. Predictions need to keep track of history data to analyze the usage behavior of the users. Web Usage behavior of a user can be analyzed using the web log file of a specific website. User behavior can be analyzed by observing the navigation patterns. This approach needs user session identification, clustering the sessions into similar clusters and developing a model for prediction using the current and earlier accesses. Most of the previous works in this field have used K-Means clustering technique with Euclidean distance for computation. The drawbacks of K-Means is that deciding on the number of clusters, choosing the initial random center are difficult and the order of page visits are not considered. In this project we uses hierarchical clustering technique with modified Levenshtein distance, Page Rank using access time length, frequency and higher order Markov model for prediction. Experimental results prove that the proposed approach for prediction gives better accuracy over the existing techniques.

# TABLE OF CONTENTS

# 1.INTRODUCTION

## 1.1 MOTIVATION

This work can be used to prefetch the web pages before they are actually being requested by the user, this reduces the access latency.

## 1.2 PROBLEM STATEMENT

Thirst for knowledge and increasing demand for accessing information slows down the performance of satisfying the request by the user, hence this project aims on Web Page Access Prediction using Hierarchical Clustering Based on Modified Levenshtein Distance and Markov Model. Web page prediction technique is proposed which involves user and session identification, clustering the user and the higher order Markov model is used for prediction

## 1.3 PROGRAM OBJECTIVES

### 1.3.1 DATA MINING

**Data mining** refers to extracting or “mining” knowledge from large amounts of data. Data mining is also known as “Knowledge discovery in databases” or KDD. Data mining is applicable to any kind of repository which includes relational databases, data warehouses, transactional databases, advanced database systems, flat files and the World Wide Web. The goal of data mining process is to extract information from a data set and transform it into an understandable structure for further use.

Data mining is a methodology that is used for extracting knowledge from data. Currently, data mining applications focus on web mining. Web mining is the application of data mining techniques that are used to extract knowledge from web data, including web documents, hyperlinks between documents, usage logs of websites, etc.

Web mining is classified into three types based on extracting knowledge. They are web structure mining, web content mining, and web usage mining. Web content and structure mining utilizes the real or primary data on the web. On the contrary, Web usage mining mines the secondary data derived from interactions of users with the web server. Web data are noisy, incomplete, inconsistent and difficult to analyze and mine. Superior quality of data gives superior quality of output; superior quality of output gives reliable information. For this reason, web data mining offers data preprocessing techniques.

### 1.3.2 WEB MINING

With the increase in technologies, the need to apply data mining techniques to electronic data has increased in order to find out useful information. Data mining is the process to explore large amounts of data in order to extract useful data . Web Mining is the process of applying data mining techniques to Web data. Web mining is used widely in e-commerce, businesses in decision making, assisting in the design of good websites and assisting the user when navigating the web.

Web Mining can be divided into three categories. They are Web Structure Mining. Web Content Mining, Web Usage Mining.

**Need for Web Mining**

Webmining is a technique to crawl through various web resources to collect required information which enables an individual or a company to promote business understanding ,marketing dynamics,new promotions floating on the internet etc.Web mining is used to understand customer behaviour,evaluate the effectiveness of a particular website,and help to quantify the success of a marketing campaign.It allows you to look for patterns in data through web structure mining,content mining and web usage mining. The information gathered through web mining is evaluated by using traditional data mining parameters such as clustering and classification,association and examination of sequential patterns.

### 1.3.3 WEB STRUCTURE MINING

It aims at generating structured summary about Websites and Web pages in order to identify relevant documents. Web structure mining mainly focuses on link information, which is an important aspect of Web data. Web structure mining can be used to reveal the structure or schema of Web pages which would help in Web document classification and clustering based on its structure.

### 1.3.4 WEB USAGE MINING

It involves the automatic discovery and analysis of patterns in usage data as a result of the user's interactions with one or more websites. It focuses on tools and techniques used to study and understand the users navigational preferences and behavior by discovering their Web access patterns. These techniques  help e-commerce ,business to improve their websites in an efficient manner. The goal of Web usage mining is to capture, model and analyze the user's behavioral patterns. It involves three phases: Preprocessing of Web data, Pattern discovery and pattern analysis. Web usage mining can be classified further depending on kind of usage data considered.

- Web Server Data: The user logs are collected by Web server. It includes IP address, page reference and access time.
- Application Server Data: commercial application servers such as Web logic Story Server have significant features to enable E-commerce applications to be built on top of them with little effort.
- Application Level Data: New kinds of events can be defined in an application, and logging can be turned on for them- generating histories of these specially defined events.

### 1.3.5 PREPROCESSING

Before starting any mining technique, web data has to be cleaned and preprocessed. Preprocessing prepares data for the pattern discovery stage. Preprocessing includes cleaning, normalization, transformation, selection. The missing values, out of range values, noise can be removed during preprocessing. The product of preprocessing is the final training set.

### 1.3.6 PATTERN DISCOVERY

During this stage, algorithms are run on the data and patterns are extracted from it. Pattern discovery involves employment of data mining techniques in order to extract knowledge from collected and preprocessed data. Widely used pattern discovery techniques are association rule mining, clustering, classification, sequential patterns.

- **Association rule mining:** Association rule mining refers to the sets of pages that are accessed together in a single server session. These rules are used to identify items that are likely to be purchased or viewed in a similar session.
- **Clustering:** Clustering aims at identifying data items with same characteristics. Clustering is the task of grouping a set of objects in the same group are more similar to each other than to those in other groups.
- **Classification:** Classification aims at finding common properties among a set of objects and mapping those objects into a set of predefined classes.
- **Sequential patterns:** Sequential patterns attempts to find patterns such that a set of items is followed by another item within a certain period of time and in a certain server session.

### 1.3.7 AGGLOMERATIVE HIERARCHICAL CLUSTERING

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

The hierarchy within the final cluster has the following properties:

- Clusters generated in early stages are nested in those generated in later stages.
- Clusters with different sizes in the tree can be valuable for discovery.

**Process**

- Assign each object to a separate cluster.
- Evaluate all pair-wise distances between clusters
- Construct a distance matrix using the distance values.
- Look for the pair of clusters with the shortest distance.
- Remove the pair from the matrix and merge them.
- Evaluate all distances from this new cluster to all other clusters, and update the matrix.
- Repeat until the distance matrix is reduced to a single element.

**Advantages**

- It can produce an ordering of the objects, which may be informative for data display.
- Smaller clusters are generated, which may be helpful for discovery.

### 1.3.8 HIGHER ORDER MORKOV MODEL

The Markov property specifies that the probability of a state depends only on the probability of the previous state. But we can build more “memory” into our states by using a higher order Markov model in an nth order Markov model

$P(X_i \mid X_{i-1}, X_{i-2},..., X_1) = P(X_i \mid X_{i-1},..., X_{i-n})$

# 2.LITERATURE SURVEY

DONG Yihong proposed a work on “A Novel incremental mining algorithm of frequent patterns for Web usage mining”. The proposed paper presents a novel algorithm for updating global frequent patterns. A rapid clustering method is introduced to divide database into n parts, where the data are similar in the same part. Then, the nodes in the tree are adjusted dynamically in inserting process by pruning and laying back to keep the frequency in descending order.

UjwalaManojPatil and J.B.Patil proposed a work on “Web Data Mining Trends and Techniques”. The proposed paper explains about Web data mining and the categories of web data mining – Web content mining, Web Usage mining, Web structure mining. The challenges and issues pertaining to Web Data Mining are also discussed in this paper.

BamshadMobasher proposed a work on “Effective Personalization based on association rule discovery from Web Usage Data”. In this paper the author proposed an effective and scalable technique for Web personalization based on association rule discovery from usage data. A detailed experimental evaluation on real usage data is performed maintaining a computational advantage over direct approaches to collaborative filtering such as the k-nearest neighbour strategy.

JORG SANDER and MARTIN ESTER, proposed a work on “ Density-Based Clustering in Spatial Databases: GDBSCAN and Its Applications” .It states that the clustering algorithm DBSCAN relies on a density-based notion of clusters and is designed to discover clusters of arbitrary shape as well as to distinguish noise. The GDBSCAN algorithm can cluster point objects as well as spatially extended objects according to both, their spatial and their nonspatial attributes. The algorithm GDBSCAN generalizes DBSCAN in two important ways. First, we can use any notion of a neighbourhood of an object if the definition of the neighbourhood is based on a binary predicate which is symmetric and reflexive. Second, instead of simply counting the objects in the neighbourhood of an object, use other measures, e.g., considering the nonspatialattributes such as the average income of a city, to define the “cardinality” of that neighbourhood.

Srivatsava proposed a work on “Web usage mining: Discovery and applications of usage patterns from web data”. It states that applications of data mining techniques to the world wide web,referred as web mining has been the focus of several research projects. The proposed paper presents a taxonomy of web mining and place various aspects of web mining in their proper context

Yong Wang proposed a work on”Mining sequential Association rule for improving web document prediction”. It states that providing a comparative study on different kinds of sequential association rules for web document prediction. By the method of variance analysis, we explore the effect of sequence and temporal information on influencing the precision of prediction. We show that sequence constraints, the temporal constraints and the interaction between these two can effect the precision of prediction.

Cui Wei proposed a work on “Algorithm of Mining Sequential Patterns for Web Personalization Services”. This paper focuses on the requirements of webpersonalization service for sequential patterns and sequential mining algorithms. Previous sequentialmining algorithms treated sequential patterns uniformly, but individual patterns in sequences often have differentimportance weights. To solve this problem, we proposea new algorithm to identify weighted maximal frequentsequential patterns. Web usage mining has been used effectively to informweb personalization and recommender systems, andthis new algorithm provides an effective method foroptimizing these services.

Mukund Deshpande proposed a model on “Selective Markov models for predicting web page accesses”. It states the importance of Markov models for solving the problem of predicting a user’s behaviour on a website and the need to personalize and influence a user’s browsing experience. Of the different variations of the markovmodels ,it is generally found that higher-order markov models display high predictive accuracies on web sessions that they can predict. However higher-order models are also extremely complex due to their large number of states, which increases the space and run-time requirements. Here they presented different techniques for intelligently selecting parts of different order Markov models so that the resulting model has a reduced state – complexity, while maintaining a high predictive accuracy.

## 3.EXISTING SYSTEM

Web page recommendation gained its importance from the ever increasing number of e-commerce web information systems and e-business. Web page recommendation involves personalizing the web users browsing experiences, assists web users in navigating the site, accessing the information they need and improves the site topology. Existing methodology used to predict WebPages are not efficient and based on Euclidean Distance and cannot give exact prediction .When only Markov models are used they suffered from the limitation of high state space complexity .When lower order Markov models are used they lacked accuracy because of the limitation in covering enough browsing history.

## 4.PROPOSED SYSTEM

Proposed system reduces the error rate in prediction of web pages based on two inputs which are preprocessed and then clustered using Hierarchical Clustering which is based on modified Levenshtein Distance based on Higher Markov model. Markov models and association rules to achieve better web page access prediction performance. Association rules are simple to implement but they suffer from the problem of identifying the correct prediction out of many rules that lead to a large number of predictions. Clustering can be used to improve the personalization task but they are not appropriate to be used as predictive models on their own. In clustering the prediction is performed on the cluster sets rather than the actual sessions. So in our project we integrate all the three techniques to overcome the problems that are faced by using these techniques individually.

# 5.SYSTEM ANALYSIS AND DESIGN

## 5.1 REQUIREMENTS ANALYSIS

### 5.1.2FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs .Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

- **Input :** A web log file containing requests from Clients
- **Intermediate Input :**No of Clusters to Be Formed
- Two input Pages From clusetered Data
- **Output**: Targeted page

### 5.1.3 NON-FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design.

**User interface:** The system should provide “click and go” type of graphical interface with buttons and links that can be easily understood by the user. So that the errors in entering the data in appropriate fields can be reduced

**Performance:** The system should perform its objectives efficiently and effectively as per the requirements. Response time should be very fast.

**Error Handling:** During data input if errors are occurred then appropriate messages should be displayed so that the user can easily identify and rectify them

## 5.2 SPECIFICATIONS

### 5.2.1 SOFTWARE REQUIREMENTS

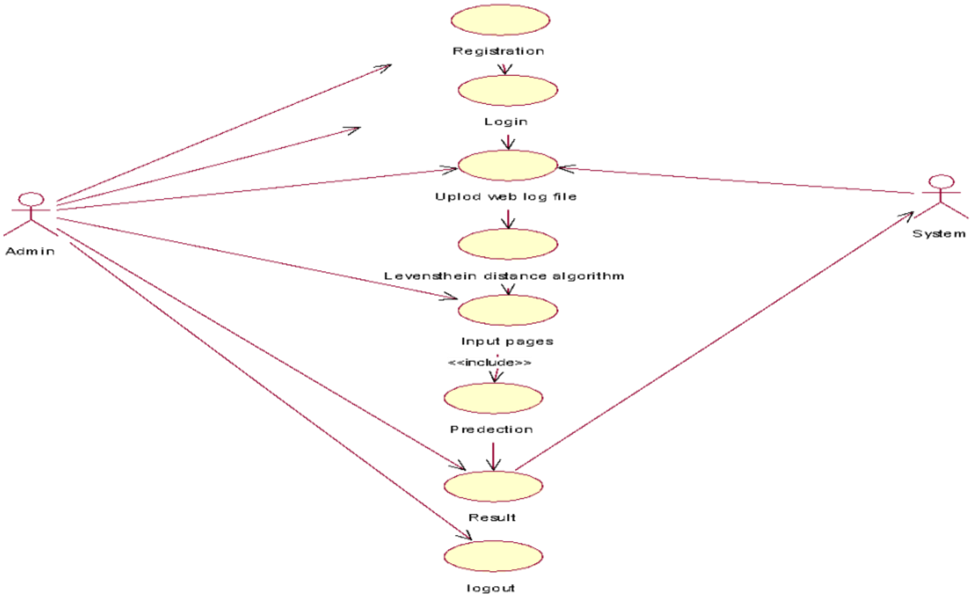
1. Operating System: Windows 10 OS (preferred)
2. Language used: Java
3. Web Server: Tomcat 7.x
4. Web technologies: Servlets, JSP
5. Database: Oracle 10g

### 5.2.2 HARDWARE REQUIREMENTS

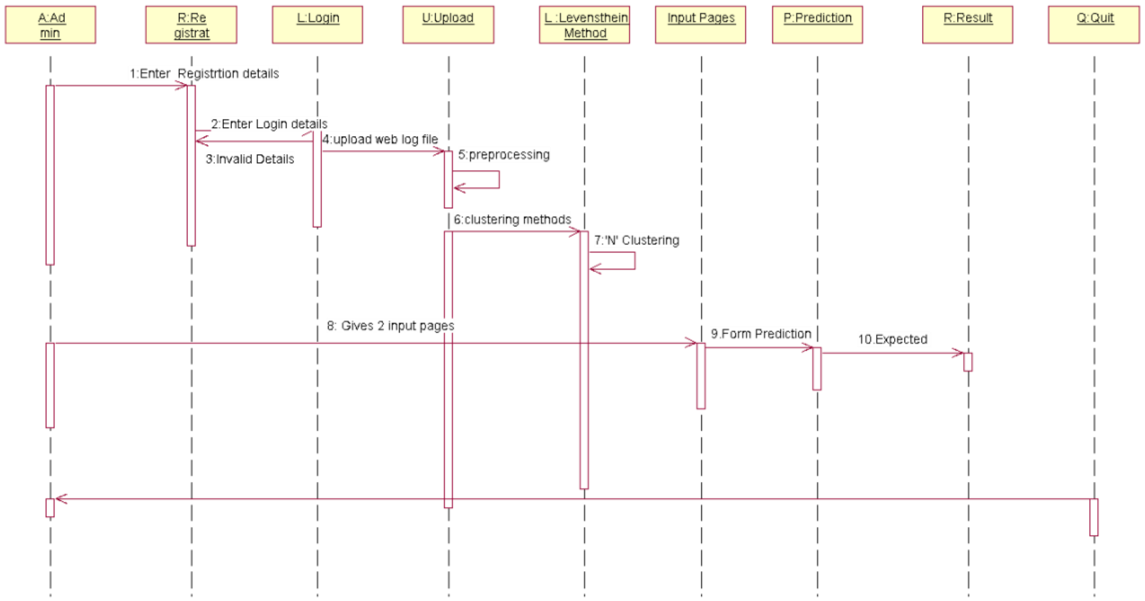
1. Ram: 4GB required
2. Rom: minimum 100GB
3. CPU: Dual- core 64-bit x86 CPU running at 2.5GHz

# 5.3 UML DIAGRAMS

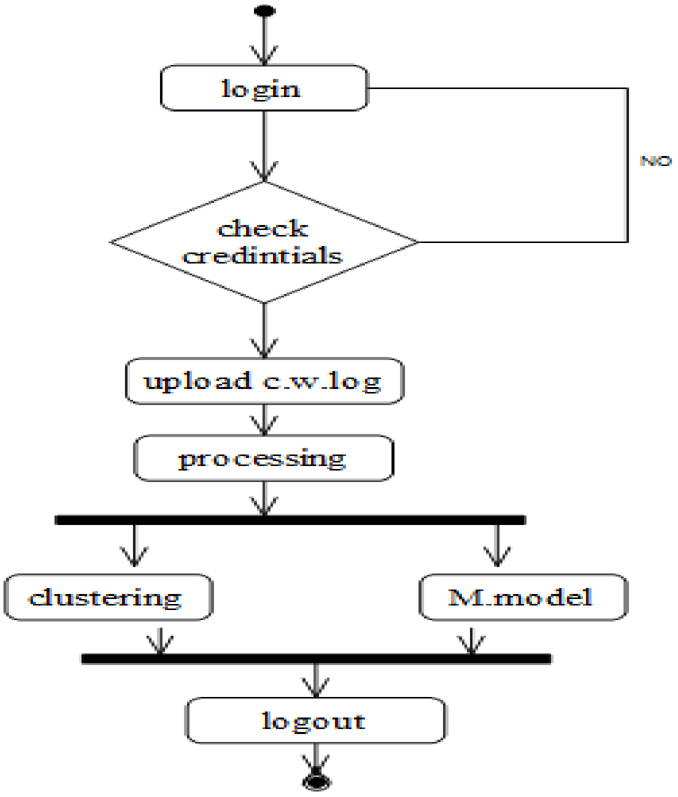
## 5.3.1 Use case Diagram:



## 5.3.2 Sequence Diagram:

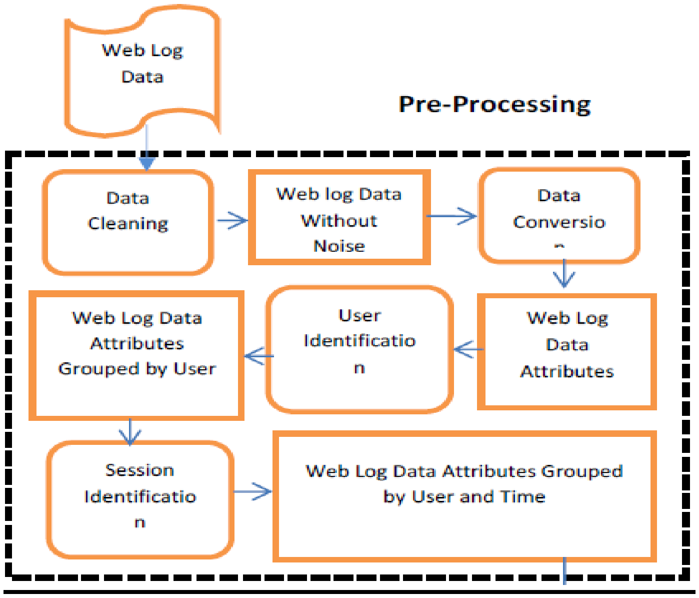


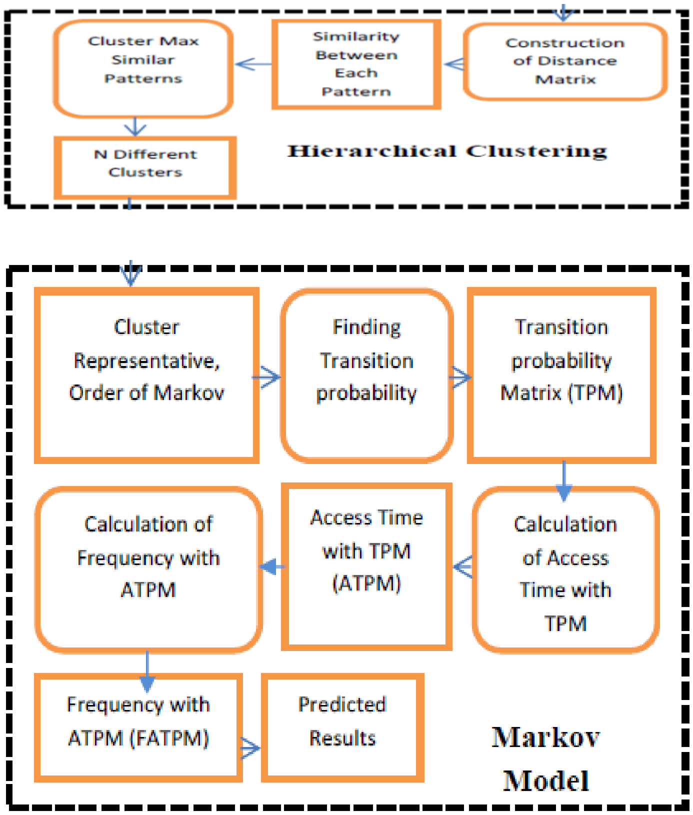
### 5.3.3 Activity diagram:



# 6.METHODOLOGY

## 6.1 SYSTEM ARCHITECTURE




Cluster Max Similar Patterns
Similarity Between Each Pattern
Construction of Distance Matrix
N Different Clusters
Hierarchical Clustering
Cluster Representative, Order of Markov
Finding Transition probability
Transition probability Matrix (TPM)
Calculation of Frequency with ATPM
Access Time with TPM (ATPM)
Calculation of Access Time with TPM
Frequency with ATPM (FATPM)
Predicted Results
Markov Model

## 6.2 ALOGARITHMS

**Algorithm:** Levenshtien distance

**Input :**Set of sessions stored in sessions DB

**Output :**k clusters

```
Step1:Set n to be the length of s.
          Set m to be the length of t.
Step 2:   if m=0 return m exit
             If n=0 return n exit.
              Construct a matrix containing 0..m rows and 0..n columns.
 Step 3:  initialize the first row to 0..n.
             Initialize the first column to 0..m.
```

```
Step 4:  Examine each character of s (i from 1 to n).
         Examine each character of t (j from 1 to m).
Step 5:  If s[i] equals tp t[j], the cost is 0.
        if s[i] doesn't equal t[j], the cost is 1.
Step 6:  Set cell d[I,j] of the matrix equal to maximum of.
         a. The cell immediately above plus 1:d[i-1,j]+1.
         b. The cell immediately to the left plus 1:d[i,j-1]+1.
         The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
```

**Algorithm**: Morkov model

**Input :**Cluster representative

**Output :**Transition probability matrix

**Method**:

```
    Step 1: Construct Transition Probability Matrix.
    Step 2: Calculate probability for each state.
P[i,j] = n[i, j] /$\sum_{j=0}^{n} n[i,j]$
    Step 3: update Transaction probability Matrix.
    Step 4: compute maximum probability.
```

# 7.IMPLEMENTATION

```
package com.IRM.algorithm;

import java.util.ArrayList;

public class AggCluster implements Cluster{

        private ArrayList<RecordAgg<String>> records;

        private Boolean justUpdated;

        private Boolean toInitUpdate;

        private RecordAgg<String> currentMean;

        private RecordAgg<String> oldMean;
```

```
    private void init(RecordAgg<String> currentMean){

    this.justUpdated = true;

    this.toInitUpdate = true;

    this.currentMean = currentMean;

    this.oldMean = null;

}

    public AggCluster(RecordAgg<String> mean){

        init(mean);

                records = new ArrayList<RecordAgg<String>>(); }

    public ArrayList<RecordAgg<String>> getRecords(){

    return records; }

public void addRecord(RecordAgg<String> record){

    justUpdated = false;

    this.records.add(record); }

public void setRecord(int index, RecordAgg<String> record){

    justUpdated = false;

    this.records.set(index, record);}

private void updateMean(){

        oldMean = currentMean;

        System.out.println("mean generating..");
        currentMean = generateMean();

        System.out.println("mean generated..");

        justUpdated = true;

        this.toInitUpdate = false;   }

private RecordAgg<String> generateMean(){

        RecordAgg<String> tempMean = this.getMean();

        System.out.println("in generate mean...");

             if(records.size()>0){

             RecordAgg<String> curRecord = records.get(0);

             if(curRecord!=null){

            int attrCount = records.get(0).getSize();

            System.out.println("in generate mean...2");

            tempMean = new RecordAgg<String>(attrCount);

            int recordCount = records.size();

            System.out.println("in generate mean...3");

            String[] attrCols = new String[recordCount];

            System.out.println("record count "+recordCount);

            System.out.println("attr count "+attrCount);

            for(int i=0;i<attrCount;i++){

                for(int j=0;j<recordCount;j++){
                    attrCols[j] = records.get(j).getAttribute(i);

                }

                tempMean.setAttribute(i,MathUtilAgg.getMean(attrCols));

            }   }  }

    return tempMean;}

public void removeAllRcords(){

    updateMean();

    justUpdated = false;

    this.records = new ArrayList<RecordAgg<String>>();}

public RecordAgg<String> getOldMean(){

    return oldMean;}

public RecordAgg<String> getMean(){

    return currentMean;}

    public Boolean isFullyFormed(){

    boolean fullyFormed = false;

    RecordAgg<String> curMean = generateMean();

    System.out.println(" is fully  "+curMean+" our mean "+getMean());

    if(curMean.equals(getMean())){

        fullyFormed = true;}

    return fullyFormed;}}
```

```
package com.IRM.algorithm;
import java.util.ArrayList;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.List;
public class AggClusterGenerator {
   public  ArrayList<AggCluster>  generateClusters(DataSet<RecordAgg<String>>  dataSet,  int noOfClusters){
        if(dataSet.length()<noOfClusters){
            return null;}
        ArrayList<AggCluster> clusters = new ArrayList<AggCluster>(noOfClusters);
        DataSet<RecordAgg<String>> cloneSet=new DataSet<<Strint>>(dataset.getAll());
        for(int i=0;i<dataSet.length();i++){
                        RecordAgg<String>mean=dataSet.remove(new
            java.util.Random().nextInt(dataSet.length()));
           clusters.add(new AggCluster(mean));    }
           System.out.println("data set size after removal.."+dataSet.length());
        dataSet =  new DataSet<RecordAgg<String>>(cloneSet.getAll());
        arrangeClusters(dataSet,clusters);
        printClusters(clusters);
```

```
 System.out.println("clusters..printed..");

 int count =0;

 while(!testForFinal(clusters)){

     count++;

     System.out.println("updating the means...");

     updateMeans(clusters);

    arrangeClusters(dataSet,clusters);

     if(count>5)

         break; }

 System.out.println("*** clusters formed ***"+count);

 this.printClusters(clusters);

 return clusters;}

private boolean testForFinal(ArrayList<AggCluster> clusters){

boolean temp = true;

for(AggCluster c:clusters){

    if(!c.isFullyFormed()){

        temp = false;

        break;}       }

System.out.println("testing for matching..."+temp);

return temp;}
```

```
    private void arrangeClusters(DataSet<RecordAgg<String>> dataSet,ArrayList<AggCluster>
clusters){
      int dsLen = dataSet.length();
      RecordAgg _rec = null;
      int mostRelIndex = -1;
      for(int i=0;i<dsLen;i++){
         _rec = dataSet.get(i);
         mostRelIndex = findMostReleventClusterIndex(clusters,_rec);
         System.out.println("most relative index.."+mostRelIndex);
         clusters.get(mostRelIndex).addRecord(_rec);}}
   private void updateMeans(ArrayList<AggCluster> clusters){
       System.out.println("updating the means...start..");
      int cLen = clusters.size();
      for(int i=0;i<cLen;i++){
         System.out.println("updating..."+i);
         clusters.get(i).removeAllRcords();}
      System.out.println("updating the means...end..");}
    private int findMostReleventClusterIndex(ArrayList<AggCluster> clusters,RecordAgg rec){
      int index = 0;
      double minDistance = ClusterUtilOne.getDistance(clusters.get(0).getMean(),rec);
    double curDistance = 0;

    int cLen = clusters.size();

    for(int i=1;i<cLen;i++){

        curDistance = ClusterUtilOne.getDistance(clusters.get(i).getMean(), rec);

        if(minDistance>curDistance){

            index = i;

            minDistance = curDistance;}}

    return index;}

    private void printClusters(ArrayList<AggCluster> clusters){

    System.out.println("  Printing the clusters ");

    AggCluster cur = null;

    for(AggCluster c :clusters){

        System.out.println(" Mean : "+c.getMean());

        for(int i=0;i<c.getRecords().size();i++){

        System.out.println(i+"  "+c.getRecords().get(i)); }}}

public static void main(String args[]) throws FileNotFoundException, Exception{

java.io.InputStream in = new FileInputStream(new File("D:\\cherry\\pt\\Projects\\doc.txt"));

DataSet<RecordAgg<String>> dSet = new DataSet<RecordAgg<String>>();

RecordAgg<String> record = new RecordAgg<String>(20);

for(int i=0;i<dSet.length();i++){
  record = new RecordAgg<String>(20);

  for(int j=0;j<20;j++){

     record.setAttribute(j, dSet.get(i).getRecord()[j]);}

         dSet.add(record);}

System.out.println("initial data set...");

for(int i=0;i<dSet.length();i++){

   System.out.println("  "+dSet.get(i));}

   System.out.println("clusters generated..."); }
```

# 8.RESULTS

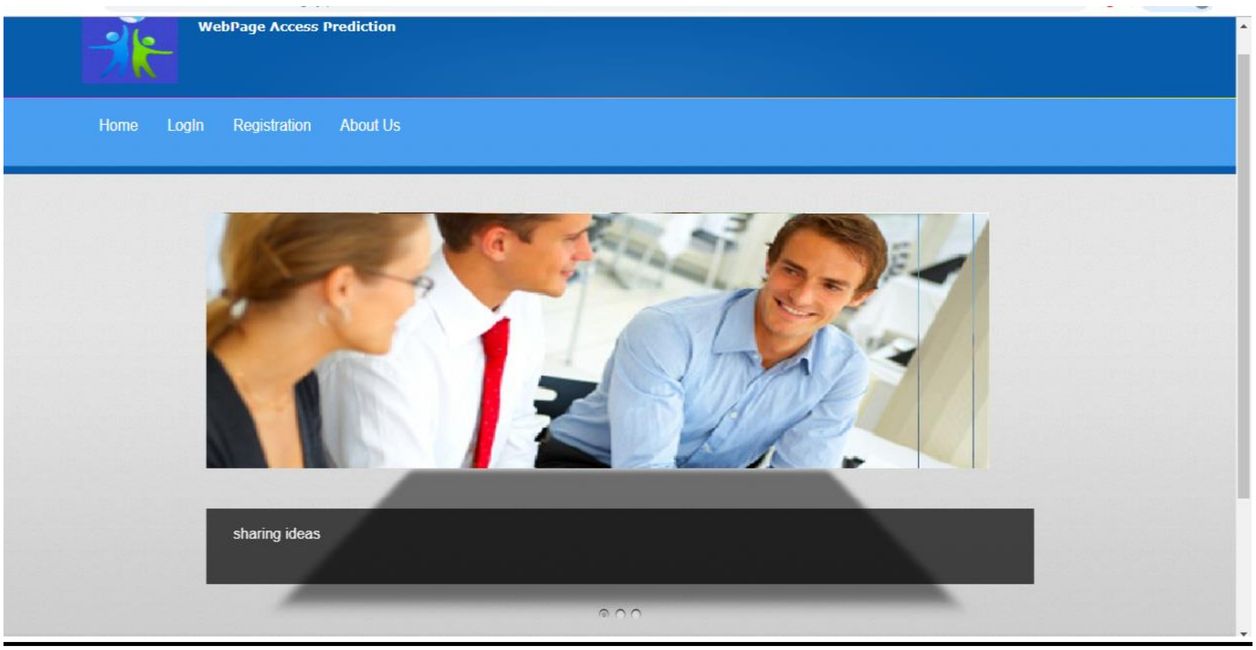
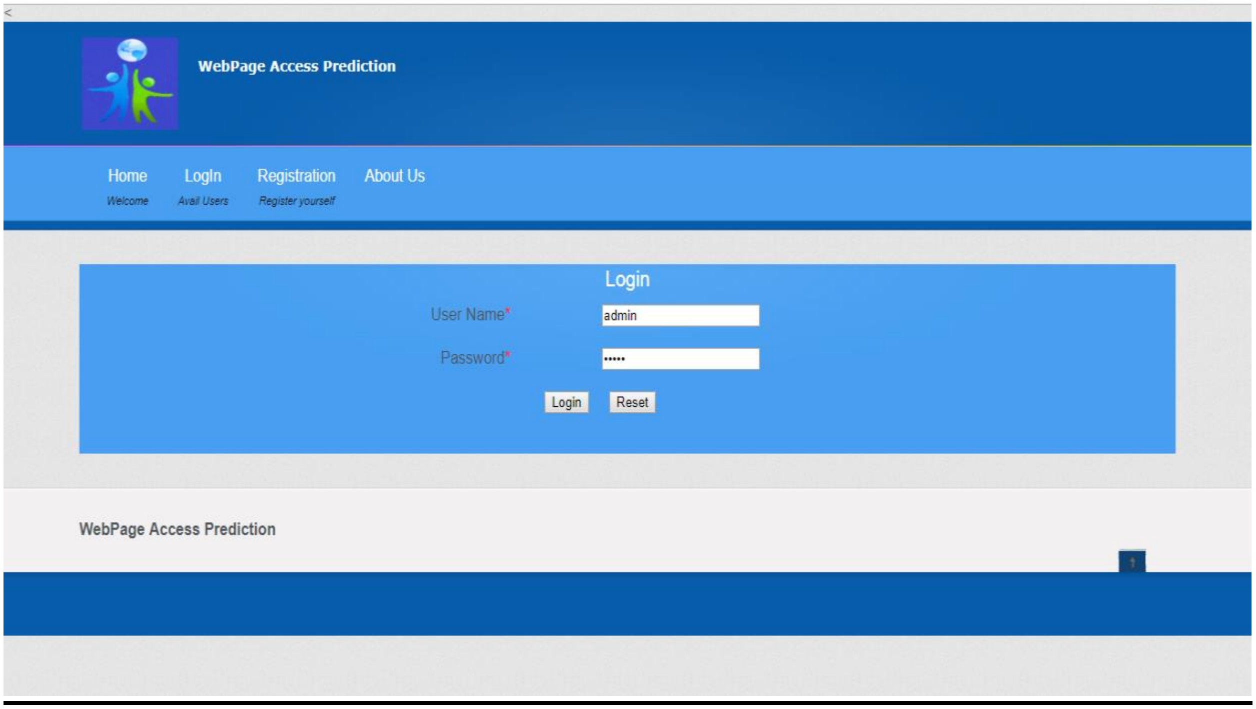


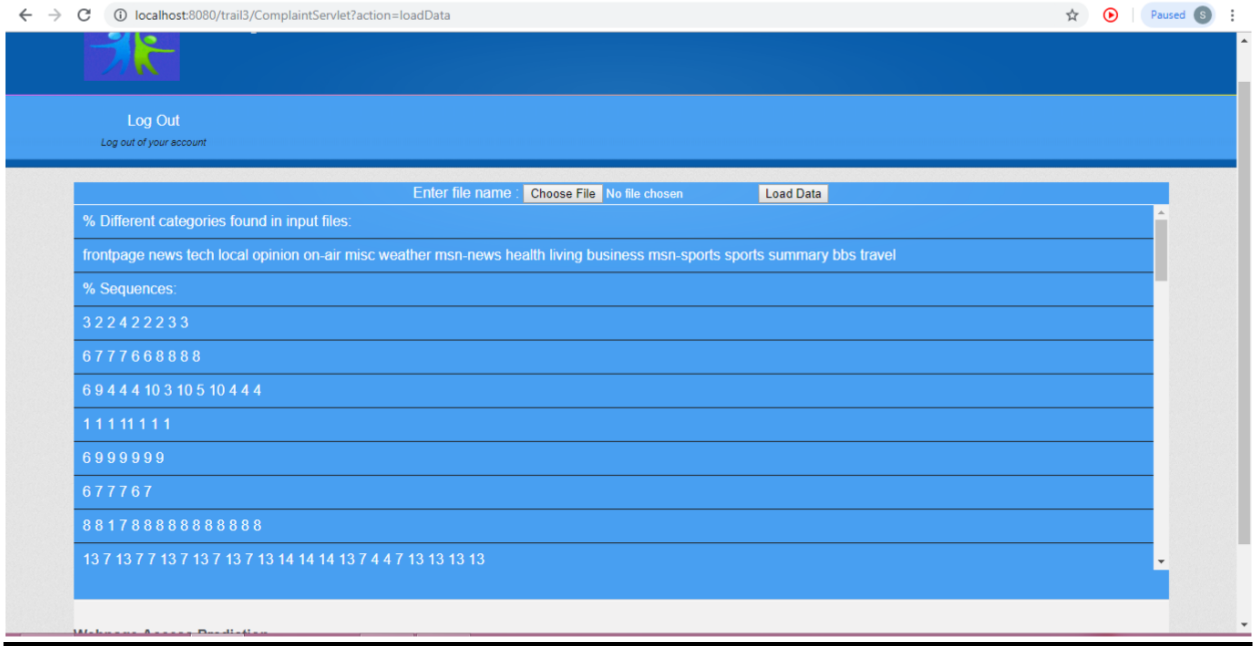
Fig 8.1: Home page



User Name* admin

Password* •••••

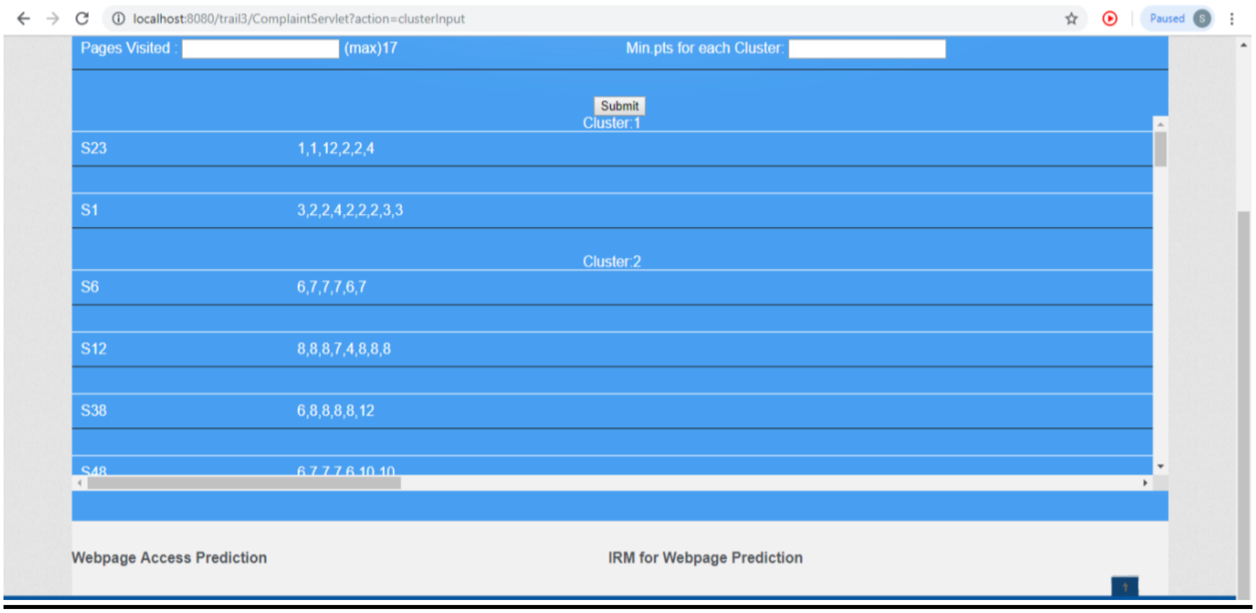
Fig 8.2:Login page



Enter file name : No file chosen

```
% Different categories found in input files:
frontpage news tech local opinion on-air misc weather msn-news health living business msn-sports sports summary bbs travel
% Sequences:
3 2 2 4 2 2 2 3 3
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
6 9 9 9 9 9 9
6 7 7 7 6 7
8 8 1 7 8 8 8 8 8 8 8 8 8 8 8
13 7 13 7 7 13 7 13 7 13 7 13 14 14 14 13 7 4 4 7 13 13 13 13
```

Fig 8.3: Uploading dataset



localhost:8080/trail3/ComplaintServlet?action=clusterInput

Pages Visited : (max)17 Min.pts for each Cluster:

Cluster:1

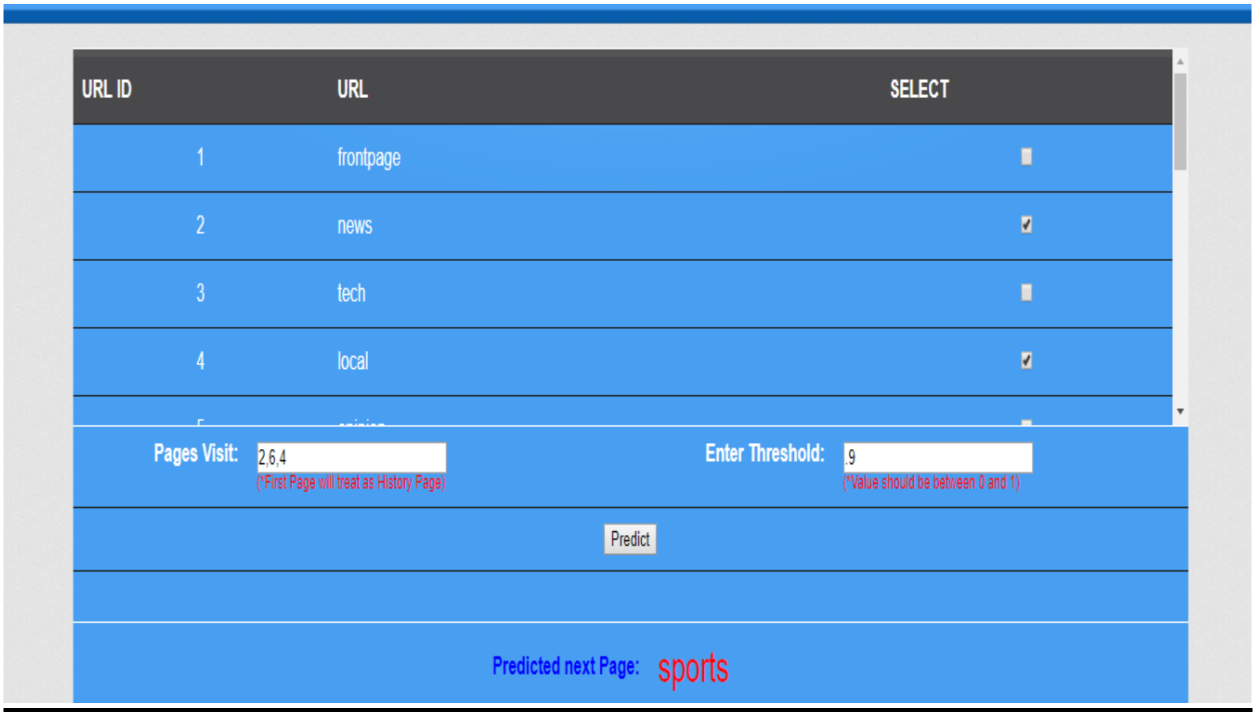
| | |
|---|---|
| S23 | 1,1,12,2,2,4 |
| S1 | 3,2,2,4,2,2,2,3,3 |

Cluster:2

| | |
|---|---|
| S6 | 6,7,7,7,6,7 |
| S12 | 8,8,8,7,4,8,8,8 |
| S38 | 6,8,8,8,8,12 |
| S48 | 6,7,7,7,6,10,10 |

Webpage Access Prediction IRM for Webpage Prediction

Fig 8.4:Clusters



| URL ID | URL | SELECT |
|---|---|---|
| 1 | frontpage | □ |
| 2 | news | ☑ |
| 3 | tech | □ |
| 4 | local | ☑ |
| 5 | opinion | □ |

Pages Visit: 2,6,4
(*First Page will treat as History Page)

Enter Threshold: .9
(*Value should be between 0 and 1)

Predicted next Page: sports

sssFig 8.5: Predicted page

# 9.CONCLUSION

In this project work a new methodology using hierarchical clustering based on modified Levenshtein distance and higher order Markov model is proposed to improve the web prediction accuracy. The proposed work can be used to prefetch the web pages before they are actually being requested by the user, this reduces the access latency.

# 10.REFERENCES

[1] Phyu Thwe, "Using Markov Model and Popularity and Similarity Based PageRank Algorithm for Web Page Access Prediction", *International Conference on Advances in Engineering and Technology (ICATE),*March 29th-30th, 2014, Singapore.

[2] Y.Z Guo et al., "Personalized PageRank for Web Page Prediction Based on Access Time Length and Frequecy", *In Web Intelligence, IEEE/WIC/ACM International Conference,* Page 687-690.

[3] Smriti Pandya et al., "Review Paper on Web Page Prediction Using Data Mining", *International Journal of Computer Engineering and Intelligent Systems,* Vol 6, No. 7, ISSN 2222-1719 (Paper), ISSN 2222-2863 (online)-2015.

[4] Tingzhong Wang, "The Development of Web Log Mining Based on Improved K-Means Clustering Analysis", *Advances in CSIE,* Vol 2, AISC 169, PP 613-618 Springer Verlag Berlin Heidelberg, 2012.

[5] Sonia Setia et al., "Survey of Recent Web Prefetching Techniques", *International Journal of Research in Computer and Communications Technology,* Vol 2, Issue 12, Dec-2013 ISSN: 2278-5841 (Online), ISSN: 2320-5156 (Print)

[6] Bindu Madhuri et al., "Analysis of User's Web Navigation Behaviour Using GRPA with Variable Length Markov Chains", *International Journal of Data Mining and Knowledge Management Process (IJDKP),* Vol 1, No. 2, March-2011.

[7] Sweah Liang Yong, "Ranking Web Pages Using Machine Learning Approaches", *In International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM,* 2008.

[8] Sonal Vishwakarma, “Web User Prediction By: Integrating Markov Model with Different Features”, *International Journal of Engineering and Science and Technology,* Vol 2, No. 4, Nov-2013, ISSN: 2319-5991.
[9] S. Prince Mary et al., “An Efficient Appraoch to Perform Pre- Processing”, *International Journal of Computer Science and Engineering,* Vol 4, No. 5, Oct-Nov 2013, ISSN: 0976-5166
[10] A. Anitha, “A New Web Usage Mining Approach for Next Page Access Prediction”, *International Journal of Computer Application,* Vol 8, No. 11, Oct-2010, ISSN: 0975-8887.