CHAPTER 5
PREDICTION MODEL

5.1. INTRODUCTION

In this chapter, ensemble clustering and ensemble classification are used to predict the model for web category. The prediction system is composed of two steps, namely clustering and classification. The research work proposes three categories of clustering based classification systems and two heterogeneous ensemble based classification systems.

The Ensemble Clustering and Classification techniques (ECLU-ECLA) are used to discover the User Navigation Pattern and Next Page Prediction by using two categories

1) Ensemble clustering includes three algorithms such as
   a) Ant based algorithm
   b) Graph partitioned algorithm and
   c) Improved pairwise nearest neighbor algorithm

2) Ensemble classification includes three algorithms such as
   a) Maximum likelihood classification
   b) Longest Common Sequence classification algorithm
c) Markov Model.
The aggregation method used by ensemble systems is the majority voting algorithm.

5.2. PROPOSED ENSEMBLE CLUSTERING AND CLASSIFICATION TECHNIQUES (ECLU-ECLA) FOR USER NAVIGATION PATTERN PREDICTION

The proposed system consists of three main steps such as preprocessing, identification of potential users and prediction system. The preprocessing step consists of cleaning the weblog file by removing irrelevant data, user and session identification, path filling and clustering session. The second step distinguishes the potential users from non-potential users. It reduces the size of the weblog file and focuses only on important users. In this work, it focuses on the prediction system, which develop the prediction engine to discover the requested page by the user. The main objective is to develop a User Navigation Pattern Discovery for Next Page Prediction system by using Ensemble prediction algorithm to mine the knowledge from weblog data in an accurate and efficient manner for improving the browsing experience of users.

The proposed ensemble prediction system uses ensemble clustering and ensemble classification. The proposed system uses heterogeneous clustering ensemble model to group similar browsing sequences together, which is then used by a heterogeneous classification ensemble model to predict the future requested page. The majority voting is used as consensus function in this work. The goal of this combination process is to improve the
quality of clustering and classification. The proposed ensemble clustering and classification technique is shown in Figure 5.1.

![Figure 5.1. Proposed Ensemble Clustering and Classification techniques](image)

5.3. ENSEMBLE SELECTION

Ensembles can be either homogeneous or heterogeneous. The models having same methodology with different vectors are called homogeneous models. The model using a different methodology is called heterogeneous models. Cluster Ensemble method is designed based on quality and diversity. The concept of diversity is used to enhance the ensemble
performance through choosing the ensemble from multiple ensembles. In Ensemble clustering, multiple clustering algorithm results are combined to produce stable and robust single consolidated clustering. In an ensemble prediction model, the set of independently trained classifiers is used in which prediction results are merged by statistical or algebraic methods. High accuracy and diversity between classifier are the two factors for selecting clustering and prediction algorithms for ensemble models.

When a new user request arrives at the server, the URL requested and the session to which the user belongs are first identified. This information is used to update the underlying knowledge and a list of suggestions is appended to the requested page. This necessitates a search through the whole weblog data file. To reduce this search time, the clustering step is introduced. Thus while using clustering, the prediction step attempts to find a cluster with high degree of similarity with the user request and predicts the possible next page visits using cluster. Therefore Clustering and prediction are performed using ensemble concepts.

The objective of ensemble models is to merge the robust result of individual model to enhance the prediction of user’s next request page. The Ensemble selection method of clustering or classifier is shown in Figure 5.2. The $M_5$ denotes the clustering or classification algorithms, $R_5$ are the clustering and classification algorithms results and $n$ denotes the number of clustering or classification algorithms used in ensemble selection method. The Consensus Function used in ensemble selection is majority voting algorithm.
5.4. ENSEMBLE CLUSTERING

In e-commerce, customers with similar browsing navigation are clustered and general features of customers can be identified using cluster analysis techniques. These cluster analysis can assist the users to get better understanding the needs of customer, provision of customer-oriented service, marketing decisions, advertisement, etc. The proposed system uses heterogeneous clustering ensemble system to group similar browsing sequences together. The three clustering algorithms (Ant based clustering algorithm, Improved Pairwise Nearest Neighbour algorithm and Graph partitioning algorithm) are used to frame the heterogeneous ensemble clustering system.
Ant based clustering (AC) algorithm is used to identify the users with a common behavior and sort them accordingly. A graph partitioning algorithm is used to identify the correlated pages between each pair of webpages by giving the weights to the graph. An improved Pair wise Nearest Neighbor algorithm produces the clusters hierarchically through merge operations consecutively and two nearby clusters are combined at each step.

5.4.1. Ant Based Algorithm

Ant based clustering (AC) algorithm group objects to form heaps, identify them into different types of objects and places them with respect to their properties using sorting method are shown in figure 5.3. In toroidal grid, agent’s data are randomly initialized and sorted according to its neighbors. The picking and dropping probabilities, given a grid position and a particular data item j, are computed using density functions.

\[
P_{pick}(j) = \left( \frac{h^+}{h^+ + f(j)} \right)^2
\]

(5.1)

\[
P_{drop}(j) = \begin{cases} 
2f(j) & \text{if } f(j) < h^- \\ 
1 & \text{otherwise}
\end{cases}
\]

(5.2)

Where \( h^+ \) and \( h^- \) are constants, and \( f(j) \) is a neighbourhood function

\[
f(j) = \max \left( 0, \frac{1}{\sigma^2} \sum_{k \in L} \left( 1 - \frac{d(j,k)}{\alpha} \right) \right)
\]

(5.3)
Figure 5.3. Ant Based Clustering Algorithm

The $d(j, k) \in [0, 1]$ denotes the dissimilarity between data points $j$ and $k$, $\alpha \in [0, 1]$ denotes data-dependent scaling parameter and $\sigma^2$ denotes local neighbourhood size $L$. When the algorithm runs, the parameter is adaptively updated. The proposed ant-based data clustering algorithm resembles the ant behavior. An ant based clustering algorithm takes cleaned weblog files as input to identify the user behavior pattern.
5.4.2. Graph Partitioned Algorithm

Perkowitz and Etzioni developed Graph partitioning theoretic approach called graph partitioned algorithm that clusters the users with similar navigation pattern. It is used to identify the correlated pages by dividing the graph, correspond to its connected components. Figure 5.4 shows the algorithm has model the undirected graph \( M = (V, E) \) which is based on connectivity between each pair of webpages accessed.

The set of vertices \( V \) denotes identifiers of different pages hosted on the web server and edge between two vertices is assigned as a weight based on the time connectivity and frequency. Time connectivity calculates degree of visit ordering for each two pages \( x \) and \( y \) in \( i \)th session at time \( T_i \).

\[
TC_{x,y} = \sum_{i=1}^{N} \frac{T_i}{T_{xy}} \times \frac{f_x(p)}{f_y(p)}
\]

\[
T_{xy} = T_x - T_y
\]

\( T_{xy} \) denotes the difference between requested time of page \( x \) and \( y \) in the session and \( f(p) = p \) if webpage appears in position \( p \). Frequency measures the occurrence of the two pages as \( f_x(p) \) and \( f_y(p) \) in each sessions.
\[ F_{x,y} = \frac{N_{xy}}{\text{Max}\{N_x, N_y\}} \] (5.5)

\( F_{x,y} \) denotes the frequency of web page between total numbers of sessions. \( N_{xy} \) denotes number of sessions in both page \( x \) and \( y \) and \( N_x, N_y \) denotes number of session containing only page \( x \) and \( y \).

The frequency values are between 0 and 1, considered as two indicators of the degree of connectivity for each pair of webpages and can be calculated as

\[ DC_{x,y} = \frac{2 \times TC_{xy} \times F_{xy}}{TC_{xy} + F_{xy}} \] (5.6)

\( DC_{x,y} \) is the weight of each edge in the undirected graph. The data structure can be used to store the weights is an adjacency matrix \( M \) where each entry in matrix \( M_{xy} \) contains the weight \( DC_{x,y} \). If the \( M_{xy} \) value is less than the threshold (MinFreq) then it is discarded, to reduce the number of edges in graph.
The Graph partitioning algorithm finds the group of strongly correlated pages by partitioning the graph according to its connected components. Starting from a vertex a Depth First Search (DFS) method is applied on the graph where $M$ is applied to search the connected component reachable to this vertex. Once the component has been found, the algorithm checks if there are any nodes not considered in the visit. If so, the previously connected components are split, and therefore it needs to be identified. To do this the DFS is again applied by starting from one of the nodes not visited. In the worst case, when all the URLs are in the same cluster, the cost of this
algorithm will be linear in the number of edges of the complete graph $G$. Before the clusters put into the navigational pattern profile, they are ranked based on values store in the matrix $M$.

5.4.3. Pairwise Nearest Neighbor

The pair wise nearest neighbor (PNN) method is also called Ward's method or bottom up hierarchical clustering technique, in which every item initially corresponds to individual clusters based on the distance between the items that are merged pair wise at each step to produces single cluster.

This method begins by assigning each data vector $V_i$ to its own code vector, where $c_x$ and $c_y$ are the clusters of pages $x$ and $y$. The codebook size is minimized by combining two neighbor clusters, by finding the similarity between page $x$ and page $y$. The Merge cost of two clusters $c_x$ and $c_y$ is known as distance between the clusters referred by the code book distortion. The distance between two clusters are evaluated as

$$d_{x,y} = \frac{S_x S_y}{S_x + S_y} \| c_x - c_y \|^2$$  \hspace{1cm} (5.7)

Where $S_x$ and $S_y$ are the size of the corresponding clusters $M_x$ and $M_y$. Since the time complexity of this method is lower bounded by $\Omega(N)^2$, it take high
execution time. To overcome the problem, Improved PNN (Pair wise Nearest Neighbor) method is proposed in figure 5.5.

### Figure 5.5. Pair wise Nearest Neighbor Algorithm

```plaintext
PNN(X, M) R,→P
s_i{x_i}∀i ∈ [1, N]; w←M;
REPEAT
(s_a, s_b) ← NearestClusters ();
Merge Clusters (s_a, s_b); w←w-1;
UpdateData Structures (); UNTIL w=W;
```

5.4.4. Improved PNN (Pair wise Nearest Neighbor) Method

The improved pair wise nearest neighbor approach is a bottom up hierarchical clustering technique. Every item in this technique initially corresponds to individual clusters and based on the similarity of items they are merged pair wise at each step to produce a single cluster. The similarity between two transactions is defined as the total number of unique pages referenced to the number of common pages. The improved PNN algorithm is described below.

1. Click stream transactions, formed using IP address and URL from access log and locate each transaction
in individual cluster.

2. Calculate similarity ratio between transactions using the following Equation.

\[
\text{Similarity ratio} = \frac{\text{No. of unique pages referred}}{\text{No. of pages}}
\]  

(5.8)

3. For every transaction, find first ‘k’ neighbors having a similarity ratio greater than threshold and neglect other neighbors.

5. Cluster the pair which has the highest similarity ratio.

6. Change the similarity ratio for objects in the neighborhood of merged pair.

7. Identify new set of k neighbors from 2k neighbors of merged pair.

8. Change the neighbors in the merged pair list.

9. Repeat step (3) and (4) until merging is not possible.

5.5. ENSEMBLE CLASSIFICATION

Given a training set, the classifier classifies the given set into classes. In this work, Ensemble classification model is used in prediction system because it consumes less time and memory utilization. The ensemble classifier is constructed using three heterogeneous classifiers. Maximum likelihood (MLC) algorithm is a statistical decision rule based classifier that computes the probability function of data for each class and places the data having the highest probability class. Longest Common Sequence (LCS) is the second classifier used to identify the longest common subsequence for all set of sequences and distinguishes the current user activities to predict the users’
next movement. Markov Model (MM) algorithm is the third classifier, depends on the sequence of previously accessed pages to predict the users’ next page access.

The identical pages $P_1$, $P_2, ..., P_m$ is contained in the current active session $S$. Each page has distinct codes used to construct the session $S$. If the user perceives a page in a website, the prediction system generates a distinct code (new page request) otherwise it changes the URL with predefined code. The heterogeneous ensemble clustering produces the output with navigation patterns set such as $NP = np_1, np_2, ..., np$ where $np_i$ denotes $k$ webpages in a navigation pattern (i.e. $np_i = P_1, P_2, ..., P_k$, where $k = 1 \leq i \leq n$ and $P_1, P_2, ..., P_k \in P_s$). Current active session $P_s$ contains Sequence $W' = P_1, P_2, ..., P_m$ where $m$ denotes an active session window size. Each classifier takes navigation patterns $np_i$ and active session $S$ as input.

The objective of the classifier is to identify a cluster having the highest degree of similarity with user request. The user active session is used to construct the co-occurrence of matrix $M$. The conditional probability of element $M_{ij}$ in the page $P_i$ and $P_j$ are visited in the same session. Based on its values the co-occurrence of matrix $M$ is calculated. The prediction lists build the pages in the active session window and searches the cluster based classifier for building the prediction list.

In LCS algorithm, the prediction engine chooses the user request suitable with the selected cluster. If diversity between positions of last elements of longest common subsequence discover, the cluster and its position
of first element. This classifier reduces the prediction system and selects the 
cluster of same behavior pattern. If the prediction list is dissimilar to the first 
page in the next user activity, then it is necessary to classify the new user 
activities again. In MLC, the user’s request is mapped to the cluster that has 
the highest probability. In MM classifier, the probability of visiting page Pi 
does not depend on all the pages in a session, it depends on a small set of k 
preceding pages. This model is also called the $K^{th}$ order Markov model. The 
consensus function used in ensemble is the majority voting algorithm.

5.5.1. Maximum Likelihood Classification

Maximum likelihood (MLC) algorithm is a statistical decision rule 
based classifier that computes the probability function of data for each class 
and that places the data having highest probability to the class. In this work, 
the method is used to classify user requests from the weblog data. 
Mahalanobis distance (MD) Ryuei Nishiia and Shinto Eguchi (2005) is used 
to calculate minimum distance classifier.

$$
MD = (x - mI)^T C^{-1}(x - mI)
$$

(5.9)

Here, CI denotes covariance matrix for the left or right particular 
imagined movement, mR and mL denote right and left imagined movement 
classes, CR and CL denotes corresponding covariance matrices and T 
indicates transposition operator. The Mahalanobis distance d from x to each 
class is used to classify the feature vector x and places x to the class for which
the Mahalanobis distance is minimum. Mahalanobis distance (MD) is calculated using the full covariance matrix. The MLC algorithm is used to classify the user requests based on matches with the cluster whose probability is the highest. The covariance matrix of each class is estimated to classify the user request based cluster.

5.5.2. Longest Common Sequence Classification Algorithm

To determine the similarities of two sequences $\alpha$ and $\beta$ is the most termed problems in pattern matching. The longest common subsequence (LCS) is one of the problems to determine the sequences $\alpha$ and $\beta$. The subsequence of maximal length common to both sequences is measured by the LCS string comparison.

In general comparing two sequences of identical vector $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ is a subsequence of $\alpha$ only if increasing sequence $\{j_1, j_2, \ldots, j_n\}$ of $\alpha$ is such that, for all $i=1,2,\ldots,\text{lit}$ has $\alpha_{j_i} = \gamma_i$. In $\alpha$ and $\beta$ sequences if $\gamma$ is a subsequence of both $\alpha$ and $\beta$ then $\gamma$ is common subsequence of $\alpha$ and $\beta$. In order to calculate longest common subsequence (LCS) for the given two sequences $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, $\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$ of page-visits. The optimal sub-structure properties of LCS are as follows:
Theorem: 1 Let $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ and $\vec{\beta} = \langle \beta_1, \beta_2, \ldots, \beta_m \rangle$ be sequences, and let $\vec{\gamma} = \langle \gamma_1, \gamma_2, \ldots, \gamma_n \rangle$ be any LCS of $\vec{\alpha}$ and $\vec{\beta}$.

1. If $\alpha_n = \beta_m$ then $\gamma_1 = \alpha_n = \beta_m$ and $\vec{\gamma}_{i-1}$ is a LCS of $\vec{\alpha}_{n-1}$ and $\vec{\beta}_{m-1}$.

2. If $\alpha_n \neq \beta_m$, then $\gamma_1 \neq \alpha_n$ implies $\vec{\gamma}$ is a LCS of $\vec{\alpha}_{n-1}$ and $\vec{\beta}$.

3. If $\alpha_n \neq \beta_m$, then $\gamma_1 \neq \beta_n$ implies $\vec{\gamma}$ is a LCS of $\vec{\alpha}$ and $\vec{\beta}_{m-1}$.

Where $\alpha_{n-1} = \langle \alpha_1, \alpha_2, \ldots, \alpha_{n-1} \rangle$, $\beta_{m-1} = \langle \beta_1, \beta_2, \ldots, \beta_{m-1} \rangle$ and $\gamma_{i-1} = \langle \gamma_1, \gamma_2, \ldots, \gamma_{n-1} \rangle$. This algorithm outputs the subsequence indices of two sequences that matches in forming the longest common subsequence. Let consider the example of two sequences $\vec{\alpha} = \langle A, B, C, B, D, A, B \rangle$ and $\vec{\beta} = \langle B, D, C, A, B, A \rangle$. Then $\vec{\gamma} = \langle B, C, B, A \rangle$.

The heterogeneous ensemble clustering produces the output with navigation patterns set such as $\vec{n}_p = \langle n_{p_1}, n_{p_2}, \ldots, n_{p_n} \rangle$ where $n_{p_i}$ denotes k webpages in a navigation pattern (i.e. $n_{p_i} = \langle P_1, P_2, \ldots, P_k \rangle$, where $k = 1 \leq i \leq n$ and $P_1, \ldots, P_k \in S$). Current active session $S$ contains Sequence $\vec{\omega} = \langle P_1, P_2, \ldots, P_m \rangle$ where $m$ denotes the active session window size. Each classifier takes the navigation patterns $\vec{n}_p$, and active session $S$ as input. Based on values that are stored in the co-occurrence matrix $M$, the prediction list is built in the active session windows. Each cluster is ranked based on their value. After this step, the prediction system searches the cluster based on the LCS classifier. The prediction list is built by identifying the cluster with highest degree of LCS with respect to sequence $\vec{\omega}$.
In LCS Algorithm, prediction chooses the cluster based on condition that if diversity between positions of last elements of longest common subsequence discovered in the cluster reduces, the prediction system and it selects that cluster. If prediction list is dissimilar with first page in the next user activity, then it is necessary to classify new user activities again. The LCS has two features such as

- The first feature constrains that if two sequences X and Y have same element, then LCS can be identified by deleting the last element and then computing the LCS of the shortened sequence.

- The second feature constrains that if two sequences X and Y do not end with the same element, the LCS of X and Y is the longest sequence of LCS (Xn, Ym-1) and LCS (Xn-1, Ym).

LCS can be represented by equation

\[
LCS(X_i, Y_j) = \begin{cases} 
0 & \text{if } i = 0 \text{ or } j = 0 \\
(LCS(X_{i-1}, Y_{j-1}), x_i) & \text{if } x_i = y_j \\
\text{longest}(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)) & \text{if } x_i \neq y_j 
\end{cases}
\]  
(5.10)

Longest Common Sequence classification procedure

1. Let us consider two sequences X = (x₁, x₂...xₘ) and Y = (y₁, y₂...yₙ) from clustering result.

135
2. From the prediction list, the system must find the cluster that matches with the user request using following steps.

3. LCS \((X_i, Y_j)\) denotes longest common subsequence set.

4. Find the longest subsequences common to \(X_i\) and \(Y_j\), the elements \(x_i\) and \(y_i\) are compared.

5. If equal, then the sequence LCS \((X_{i-1}, Y_{j-1})\) is extended by the element \(x_i\).

6. Else the two longest common sequences, LCS \((X_i, Y_{j-1})\), and LCS \((X_{i-1}, Y_j)\), are retained.

7. If two sequences are of the same length but not identical, it is retained.

**5.5.3. Markov Model**

Markov model is used to predict the users’ next page movement based on the sequence of the previously accessed pages. Markov models are iterative (recursive) decision trees that are utilized for modeling predictable events, which occurs over time. The input data for constructing Markov models is composed of web-sessions, where each session contains the sequence of the pages accessed by the user during the visit to the website. This technique gives more precise result by considering the consecutive orders of preceding pages.

The set of pages in a website is denoted as \(x_1, x_2\ldots x_k\) and \(Y\) denotes the user session that contains sequence of pages visited by the user. If the user has visited \(k\) pages, then probability \(\text{prob} (X_k|Y)\), of the user can visit
the pages next to $X_k$. The probability of accessing the next page $X_{k+1}$ can be formulated as

$$X_{k+1} = \arg \max_{x \in IP} \{ P(X_{k+1} = x | x_k, x_{k-1}, \ldots, x_1) \}$$ (5.11)

Where $k$ denotes the number of previous page access. The probability, $\text{prob}(X_k | Y)$ can be computed using all sequences of the users in training data $Y$. It produces accurate result but leads to unnecessary complexity. Therefore, to overcome this problem, the Markov process constrains the number of previously accessed pages $n$ (i.e. the probability of visiting a page $p_i$ is not based on all the pages in the web session but only on a small set of $n$ preceding pages) and $n < 1$ which can be formulated as

$$X_{k+1} = \arg \max_{x \in IP} \{ P(X_{k+1} = x | x_k, x_{k-1}, \ldots, x_{k-(n-1)}) \}$$ (5.12)

Where $n$ specify the number of the preceding pages. This model is called the $n^{th}$ order Markov model. The Markov model begins with computing the highest probability of the last page visited because during a web session, the user can only link the pages that are currently visited by the next one. $M_j^n$ specifies the object that contains $k$ pages, $M_j^n = \{X_{k-(n-1)}, X_{k-(n-2)}, \ldots, X_k\}$. The probability of $P(X_k | M_j^n)$ can be computed from a training data.

$$P(X_k | M_j^n) = \frac{\text{Frequency}(M_j^n, X_k)}{\text{Frequency}(M_j^n)}$$ (5.13)
The conditional probability is defined as the ratio of the frequency of the sequence occurring in the training set to the frequency of the page occurring directly after the sequence. In Markov models next state is based on the previous k states, if k has the highest page visited with more accurate predictions has two issues. 1) Coverage of model is limited and leaves many states uncovered and 2) The model is Unmanageable because of its complexity. Hence, a modified Markov model is proposed to predict the webpage access and is described below.

1. Markov model: The model reduces the low coverage problem of a high order Markov model. The highest order Markov model uses instance that is utilized to predict the instance in each test instance.

2. Frequency pruned Markov model: The order of n\textsuperscript{th} Markov models results in low exposure part which can lead to exacerbate problem of complexity states of all Markov models are added as a positive part. In many states statistical predictive reliability are low where the occurrence frequencies are less. The removal of these low frequency states affects the accuracy of a Markov model. However, the number of states is pruned and the model is reduced significantly.

3. Accuracy pruned Markov model: Frequency pruned in Markov model does not capture the issue that affect the accuracy of states. A high frequent state can be evaluated for accurate prediction. The mean value is calculated to estimate the predictive accuracy of each state, and these states with low predictive accuracy are eliminated. One way to estimate the predictive accuracy using conditional probability is called confidence pruning. The other way to estimate the predictive accuracy is to count (estimated) the errors involved is called error pruning. The probability of a state is based on
probability of earlier visited pages but additional memory can be obtained with the help of higher order Markov model is shown in figure 5.6.

```
Begin
Identify the cluster having similar access sequence as input
Begin with k highest possible value.
Markov model is applied to find kth order states for the test Session from its cluster. If the support is very less, compute next lower order states for the test session from its cluster Repeat step until convergence
Page with highest probability as suggested page is output
End
```

Figure 5.6 Markov Model Algorithm

5.6. MAJORITY VOTING

Majority voting is used in ensemble techniques to obtain the aggregated (consensus) result. The majority voting algorithm is used in both clustering and classification techniques to select the navigation pattern by combining them to ensemble model. Essentially, the ensemble chooses the class or cluster that is chosen by the majority of the classifiers/ clustering. The i\textsuperscript{th} classifier / clustering decision can be represented as \(d_{k,i}\) where \(k = 1 \leq j \leq N\) and \(i = 1 \leq j \leq C\), \(N\) denotes number of classifiers or cluster and \(C\) denotes number of classes or cluster. If the \(i\textsuperscript{th}\) classifier/clustering chooses class/ cluster \(\omega_k\), then \(d_{k,i} = 1\) and \(0\), otherwise. If a class \(\omega_k\) is chosen, and then majority voting can be defined as

\[
\sum_{k=1}^{N} d_{k,i} = \max_{i=1}^{c} \sum_{k=1}^{N} d_{k,i}
\]  

(5.14)
The certain classifiers in the ensemble are “better” than the others can be identified using a weighted majority voting.

5.7. EXPERIMENTAL RESULTS

The experiments focus on analyzing the performance of the proposed next webpage prediction algorithms. Three performance metrics namely accuracy, coverage, F1-Measure are used for this purpose.

The navigation patterns are identified using the clustering algorithm and the grouped patterns are divided into two sets. The first set is used for generating prediction and the second set is used to evaluate the predictions. Let as$_{np}$ denotes the navigation pattern obtained for the active session’s’ and let T be a threshold value. The prediction set is denoted as P (as$_{np}$, T) and the evaluation set is denoted as eval$_{np}$, the three parameters can then be calculated using Equations (1), (2) and (3).

\[
\text{Accuracy} = \frac{|P(as_{np}, T) \cap \text{eval}_{np}|}{|P(as_{np}, T)|}
\]  
\[\text{(5.15)}\]

\[
\text{Coverage (P(asnp, T))} = \frac{|P(as_{np}, T) \cap \text{eval}_{np}|}{|\text{eval}_{np}|}
\]  
\[\text{(5.16)}\]

\[
\text{F1(P(asnp, T))} = \frac{2 \times \text{Accuracy(P(asnp,T))} \times \text{Coverage(P(asnp,T))}}{\text{Accuracy(P(asnp,T)) + Coverage(P(asnp,T))}}
\]  
\[\text{(5.17)}\]
The experiments were conducted in three stages, where the first stage evaluated the effect of the ensemble clustering algorithm on prediction, the second stage evaluated the effect of clustering-based ensemble prediction and final stage of experiments evaluate the proposed ensemble clustering based ensemble classification.

Accuracy measures the degree to which the prediction algorithms produce accurate recommendations, while coverage measures the ability of the prediction algorithms to produce all page views that are likely visited by the user. The F1 measure attains its maximum value when both accuracy and coverage are maximized.

5.7.1. Performance of Clustering Based Ensemble Clustering Based Ensemble Prediction

![Graph showing accuracy percentages for different methods]

**Figure. 5.7. Overall accuracy in Ensemble Clustering based Ensemble Prediction Algorithm**
Figure 5.8. Overall coverage in Ensemble Clustering based Ensemble Prediction Algorithm

Figure 5.9. F1 Measure in Ensemble Clustering based Ensemble Prediction Algorithm
From Figure 5.7 to 5.9 Ensemble clustering and Ensemble Classification have proved to have better accuracy percentage of 94.64% when compared with the three clustering and classification technique.

5.8. SUMMARY

This chapter focused on the prediction system which discovers the navigation patterns for predicting next page request of the user. The User Navigation Pattern Discovery for Next Page Prediction system is developed using Ensemble prediction algorithm to extract the knowledge from cleaned and interested user weblog data in an accurate and time efficient manner. This system uses heterogeneous clustering ensemble model to group similar browsing sequences together, which is then used by a heterogeneous classification ensemble model to predict the future requested webpage.

The ECLUCLA prediction algorithm showed high accuracy efficiency gain over ECLULCS and GPECLA. The percentage of accuracy in the proposed method is higher than existing technique. Therefore, it is concluded that the proposed Ensemble clustering and Ensemble Classification have better accuracy compared with these three clustering and classification technique.