CHAPTER 1

INTRODUCTION

The process of applying statistical and logical techniques for describing and evaluating data is called Data Analysis. It generates inductive inferences from data and thereby helps to distinguish information of interest from that of noise present in the data. A traditional data analyst analyzes data from a single source which is in a structured format. As the volume of data being accumulated in organizations is huge at present, the significance of 'Big Data' and its analysis has more influence. The traditional data analysis methods are greatly challenged by this data tsunami in terms of heterogeneous nature of the data, huge scalability, timeliness and privacy problems.

1.1 Challenges of Traditional Data Analysis Methods

Enormous volumes of data are getting accumulated in big organizations easily, due to improvements in the techniques of data collection and storage. Researchers in medicine, science and engineering are rapidly accumulating data that is key to important discoveries. 'It's a revolution,' says Gary King, director of Harvard's Institute for Quantitative Social Science. 'We're really just getting under way. But the march of quantification, made possible by enormous new sources of data, will sweep through academia, business and government. There is no area that is going to be untouched' [1]. It is a great challenge to extract useful information from such massive volumes of data. Since the data is non-traditional or has huge dimensions, the traditional data analysis tools and techniques cannot be used in these situations. In other situations the the problems cannot be addressed by the existing data analysis methods and hence new methods need to be invented. Data Mining is a solution to overcome much of the challenges faced by the traditional data analysis techniques.

This technology combines traditional data analysis methods with sophisticated algorithms in order to process large volumes of data. It provides lot of opportunities to explore and analyze new types of data. It helps to automatically discover useful information from large data repositories. They help to discover novel and useful patterns that are hidden in large data bases.

Traditional data mining techniques were mainly created for structured data types. Due to various reasons data accumulated in organizations is no longer in traditional format i,e, data is not in a structured table format. Point of sale data collection records up-to-the minute data about customer purchases are in the form of transactions. Observations of the land surface, oceans and atmosphere are provided by a series of earth orbiting satellites which are deployed by NASA. This helps to improve our understanding of the climate system of our earth [1]. Due to the size and the spatio temporal nature of such data, traditional data analysis methods are not suitable to analyze such data sets. Millions of web pages that are added to the World Wide Web (WWW) everyday have data of different formats embedded in them. A web page may contain textual data, traditional data and/or multi-media data. Therefore a web page is said to be semi-structured and mining of web page data is quite challenging. Structured category does not include image data [2], suitable for interpretation by a machine. So, mining useful information from such image data is quite challenging. The ever increasing amounts of patient data in the form of medical images, imposes new challenges to clinical routine such as diagnosis, treatment and monitoring. Hence research in applying traditional data mining techniques to medical images to automate clinical diagnosis is the state of the art.

1.2 Objective, Scope and Limitations

The main objective of this thesis is to design and implement algorithms for improving automatic classification of data that is not in the native structured format ready to be mined. The present framework works for two types of data namely web page and medical image data sets. Classifying a web page/a medical image into one of a pre-defined category is known as web page/medical image classification. The data sets are classified using the content embedded in them. Hence the present classification framework used in this thesis is subject-based. The predictive accuracy of the classification model is improved by a set of pre-processing steps which includes feature extraction, feature selection and discretization techniques. The features are the contents of the data sets that characterize them. For example in case of web page classification WPC, features are the words that are present in a web page. For a medical image, its statistical properties represent its features.

The present framework has been applied to both binary-class and multi-class classification of web page and medical image data sets. It uses the content based features for classifying web pages and medical images. The number of categories experimented in the present work for multi-class WPC and MIC are four and three respectively. However the algorithms presented can be experimented with more categories of web pages and medical images. The modified KNN namely, MKNN used in this thesis uses a feature weighting scheme based on the interestingness measures of association rule mining. The feature weights calculated depend on the user specified threshold namely *min_sup* and *min_conf* as discussed in detail in Section 3.1.5.2. Experimental analysis of the present work is done with only two types of data sets namely, web pages and medical images. However it can be explored with other types of data sets not in native structured format such as gene expression data, etc after transforming them into a suitable format.

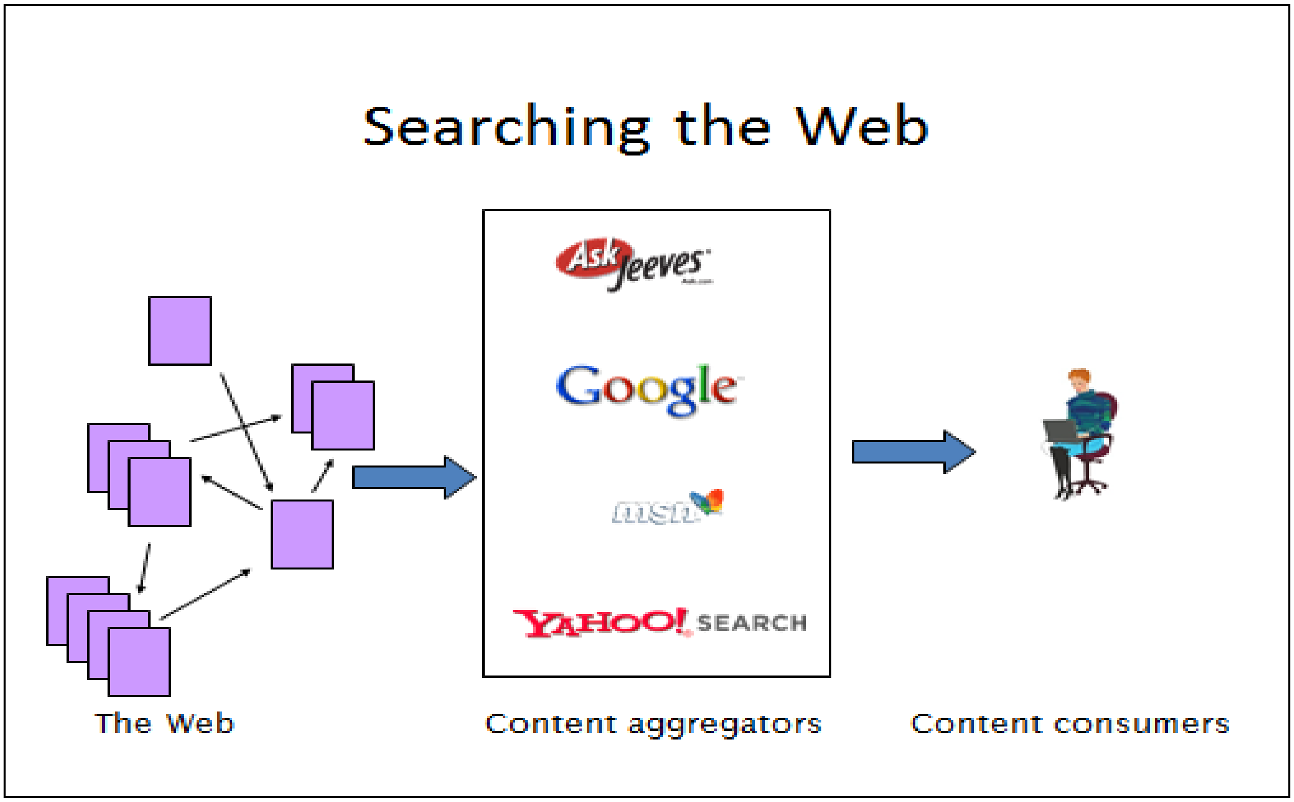


Fig 1.1 Searching the WWW

1.2.1 Web Page Classification

The following Section is a discussion about the need and architecture of a WPC model.

1.2.1.1 Motivation

The World Wide Web WWW can also be defined as a virtual society. It is an Internet based computer network. Using the WWW called the Internet, it is possible for a computer user to access the information stored in any other computer. Navigating through the web is done using browsers like Internet Explorer, Google Chrome, etc. The search engines assist users to locate information of interest on the WWW as in Fig 1.1. The volume of information being added to this huge repository has rapidly increased over the past decade. It is therefore quite challenging for the search engines to retrieve quick and relevant results to a user query. The WWW at present has over 8 billion pages. It has been estimated that every day about 1.5 million pages are being added to this huge repository. Approaches using the conventional search engines will be eventually overwhelmed by the speed at which the size of the WWW increases. Even it is unfortunately possible for highly specific queries to return many thousands of entries and this count may in turn increase over time. So, the web search engine companies will not be able to work successfully always using their standard approaches. Web directories are directories of pre-classified content which allow many users to navigate and find more relevant information in a short span of time. Web directories have a collection of web pages of the same category namely sports, news, course, university, etc. Web page Classification (WPC) helps in creating such web directories. The process of classifying a web page into one of a pre-defined category/label is called WPC. Only a very small percentage of the entire web can be classified manually as demonstrated by the Yahoo system [3].

Manual classification of web pages into one of a pre-defined category requires human expertise and careful review of the web page contents. Any team of human classifiers could be defeated, since 1.5 million of new web pages are being accumulated to the WWW each day. Therefore automatic classification of web pages is necessary. Motivated by these facts this thesis investigates methods of classifying web pages automatically using supervised machine learning methods. Such automated tools also help the search engines to make a relevant and quick retrieval of information for the user query. WPC

- helps in constructing and maintaining web directories and this can improve the quality of the search results.
- It also has applications in improving the quality of answers in a question answering system,

- focused crawling and
- in web content filtering and
- user profile mining.

1.2.1.2 Architecture of a Web Page Classification Model

Assigning a web page into one or more of a predefined categories or labels is called web page categorization or WPC. Let C = $\{C_1, C_2, \ldots\ldots C_k\}$ be the set of predefined categories, W= $\{w_1, w_2, \ldots\ldots w_N\}$ be the set of web pages that needs to be classified, and

Web Pages	Categories				
	C_1	$\cdots$	C_j	$\cdots$	C_k
w_1	a_{11}	$\cdots$	a_{1j}	$\cdots$	a_{1k}
$\cdots$	$\cdots$	$\cdots$		$\cdots$	$\cdots$
w_i	a_{i1}	$\cdots$	a_{ij}	$\cdots$	a_{ik}
$\cdots$	$\cdots$	$\cdots$		$\cdots$	$\cdots$
w_N	a_{N1}	$\cdots$	a_{Nj}	$\cdots$	a_{Nk}

Fig 1.2 Web Page Classification

$A = W\ x\ C$ is a decision matrix of the form shown in Fig 1.2. where every entry is defined as in Equation 1.1

$$a_{ij} = \begin{cases} 0\ if\ w_i\ \notin\ c_j \\ 1\ if\ w_i\ \in\ c_j \end{cases} . \qquad (1.1)$$

Sometimes a web page might belong to multiple categories also. The assignment function f : $W\ x\ C\ \longrightarrow \{0,1\}$ is approximated by the process of WPC using a learned function f' : $W\ x\ C\ \longrightarrow$ $\{0,1\}$ which is called a classification model or a hypothesis. The task of WPC is to maximize the coincidence of f' with f as far as possible [4].

Machine learning techniques are applied onto a set of training web pages to build the hypothesis f'. Each training web page is tagged with predefined category/categories. In this thesis web pages are classified into one of a predefined category using the content/subject embedded in them. Hence the framework is called subject-based/topic based classification. The architecture of a typical WPC model is shown in Fig 1.3. A WPC model is built using a set of labeled web pages as training set. These training web pages are usually of high dimensions/features and are converted into a suitable format for representation using a set of pre-processing steps. One of the pre-processing steps is to decrease the number of dimensions that are needed to represent a web page, which would otherwise degrade its performance. This is achieved using the feature selection process. The class of a new web page can be predicted using the classification model, once it is

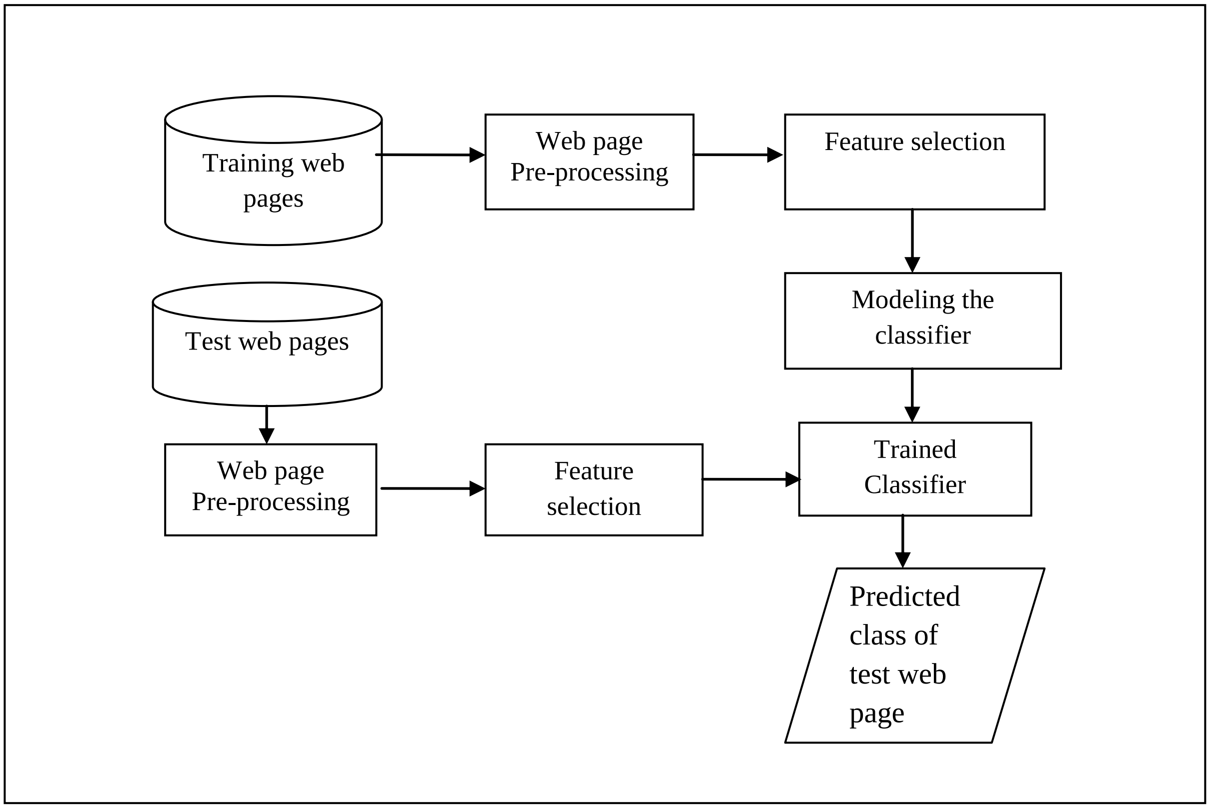


Fig 1.3 Architecture of a Web Page Classification Model

induced. Usually, the labeled web pages which are used to build the model are divided using a 70 – 30 percentage split. The model is trained using 70% of the labeled web pages. The trained model is validated using the remaining 30% of the input. An alternative method of inducing a WPC model is K-fold cross validation and K indicates the number of folds. For a 10-fold cross validation, the input data set is divided into 10 disjoint subsets and these subsets are used to build the model using 10 iterations/folds. In each fold the model is trained using nine subsets and it is validated using the remaining one subset. The classification accuracy of each fold is aggregated to finally estimate the model's performance.

1.2.1.3 Web Page Representation

In subject based classification schemes a web page is represented using its contents like words, phrases and sentences. Firstly, the web pages with hyperlinks, images, strings of characters, and HTML tags are transformed into a feature vector. It helps in eliminating less significant information from a web page and also to extract only the salient features from it. The content of a web page namely words, phrases and sentences in it are used by the subject based classification approaches. The web pages are first pre-processed to extract the most significant features in them. The various step involved in pre-processing are:

1. **Removing the HTML tags**: The format of the web page which is indicated using HTML tags are first removed in this pre-processing step. For example, the pair of tags <title> and </title> is used to indicate the title of a web page. The pair <table> and </table> is used to indicate that the contents are tables.
2. **Stop word removal**: Words that carry no significance in the context they appear are called stop words. Words such as propositions, pronouns and conjunctions are said to be stop

words. Words in a web page are compared with a pre-defined stop word list and are removed if they are stop words.

3. **Word Stemming**: In a text, different forms of the same word, each having the same stem or root may appear. For example, ‘walk’, ‘walking’ and ‘walked’ have the same stem ‘walk’. Much of the stemming algorithms work by suffix/affix stripping. This task is performed using Porter’s algorithm which is one of the common stemming algorithms.

The final vocabulary of words for the entire web page collection is arrived at after the pre-processing phase. With M as the size of the vocabulary, N as the number of web pages in the training corpus; each web page is represented as a M-dimensional vector. Training corpus of web pages is then a N x M matrix.

Bag-of-words Representation:

The bag-of-words representation is one of the commonly used representations of a document in text retrieval. Each web page is represented using this bag-of-words representation method after pre-processing it [4]. The various pre-processing tasks are explained in detail in Section 1.3.2. It is based on the hypothesis that each word in a web page represents some concept in it as illustrated in Fig 1.4. The unique words present in a document and its corresponding frequency can be used to represent a document. The bag-of-words way of representing a web page uses a vector with M number of weighted index words for each web page as in Fig 1.4. This representation is known as bag-of-words representation. The web page feature matrix is shown in Fig 1.5 where $\{C_1, C_2, C_3, \dots \dots \dots . C_N\},$ is one of a predefined set of web page categories. So a web page in this representation is $d_i = (w_{i1}, w_{i2,}, w_{i3}, \dots \dots \dots w_{iM})$ where M is the number of words in the vocabulary and w_{ij} for j = 1 to M is the weight or importance of the j^{th} word in the i^{th} web page d_i. There are different approaches of assigning weights to the words in a vector space model.

A pre-processed web page
overview most relevant cryptographic algorithms two important cryptographic algorithms symmetric cryptographic algorithms symmetric algorithms use same key asymmetric algorithms use two keys

Term	Frequency
cryptographic	3
algorithm	5
Key	2
Relevant	1
Overview	1
Two	2
Most	1
Use	1
important	1
Same	1
symmetric	2
assymmetric	1

Fig 1.4 A Web Page Represented using Bag-of-words in the Vector Space Model [4]

Web Pages	Features				Category
	F_1	F_2	$\dots$	F_M	
1	w_{11}	w_{12}	$\dots$	w_{1M}	C_1
2	w_{21}	w_{22}	$\dots$	w_{2M}	C_2
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
N	w_{N1}	w_{N2}	$\dots$	w_{NM}	C_N

Fig 1.5 Web Page Feature Matrix

Term – Weighting Schemes: The weight w_j of a term t_j in a web page is a binary value that is defined as in Equation 1.2 that follows:

$$w_j = \begin{cases} 1, if\ term\ is\ in\ the\ web\ page \\ 0, otherwise \end{cases} \tag{1.2}$$

This term weighting scheme, looks for exact matches between the web page vectors, in an information retrieval context. It doesn't consider the notion of partial relevance.

Term Frequency Inverse Document Frequency: This way of weighting a term is called TFIDF is proposed by Salton [5]. This is a product of local term importance (TF) and the global term importance (IDF). The term frequency TF(t_j, d_i) is the number of times term t_j occurs in document d_i. A term occurring more frequently in a web page is better indicative of the content of the web page. So, this defines the local importance of a term with respect to a web page. The total number of web pages that contain a term is its document frequency DF(t_j). It specifies the importance of the term globally in the entire training corpus and is defined in as in Equation 1.3

$$w_{ij} = TF\left(t_j, d_i\right).IDF(t_j) \tag{1.3}$$

where $TF\left(t_j, d_i\right)$ is the frequency of term t_j in document d_i. $IDF(t_j)$ is the inverse document frequency of term t_j and is defined as in Equation 1.4 in terms of the document frequency of a term.

$$DF\left(t_j\right) = \sum_{i=1}^{N} \begin{cases} 1, if\ d_i\ contains\ t_j \\ 0, otherwise \end{cases} \tag{1.4}$$

The inverse document frequency IDF(t_j) is defined as in Equation 1.5.

$$IDF\left(t_j\right) = \log(\frac{N}{DF(t_j)}) \tag{1.5}$$

where N is the total number of web pages in the training corpus. If a term occurs in many documents its IDF is low and is high if it occurs in only one document. TFIDF term weighting scheme denotes that a term occurring more number of times in a document i.e with high term frequency, is an important feature in that document. It is insignificant to the document when it appears in many documents, i.e. its IDF is low. Thus the TFIDF weighting scheme is powerful in discriminating the characteristics of terms within a particular document and within a document collection.

1.2.2 Medical Image Classification

The following section is a discussion about the need and architecture of a MIC model.

1.2.2.1 Motivation

Medical Imaging refers to a number of non-invasive methods of looking inside the human body. It enables a doctor to diagnose, treat and cure patients without causing harmful side effects. Medical images are more important assets of clinical history and their analysis is essential in modern medicine. The ever increasing amounts of patient data in the form of medical images, imposes new challenges to clinical routine, such as diagnosis, treatment and monitoring. The process of transforming raw imaging data using knowledge-based data mining algorithms into clinically relevant information is called medical data mining. The target mining model can be used to assist a physician in medical diagnosis. This enables a physician to spend less time in spending on the image volumes to extract the clinical information in it, while improving the diagnostic accuracy.

1.2.2.2 Architecture of a Medical Image Classification Model

Images are a type of structured data and image classification is the process of classifying an image into one of a predefined category using the feature that represent the content of a web page. In this research, a set of retinopathy medical images were used for the experimental analysis. The ever increasing amounts of patient data in the form of medical images, imposes new challenges to clinical routine such as diagnosis, treatment and monitoring. Fundus images are a class of medical images. The retinal fundus images give details of the inner lining of the eye which includes the retinal pigment epithelium, the sensory retina, Bruch's membrane and the choroid. The diabetic retinopathy and its stages can be graded using the patient data in the form of fundus images. This mainly occurs due to damaged blood vessels of the diabetic patient's retina mainly in the posterior part of the eye. In clinics various eye diseases are diagnosed and treated using these retinal fundus images. Diabetic retinopathy in patients is also screened using these images. As more number of patients are undergoing regular screening, an opthalmologists needs more time to analyze and diagnose these fundus images. Medical image mining uses data mining techniques to automate clinical diagnosis and research in this direction is the state of the art. There are four stages in building a medical image classification model namely image preprocessing, feature extraction,

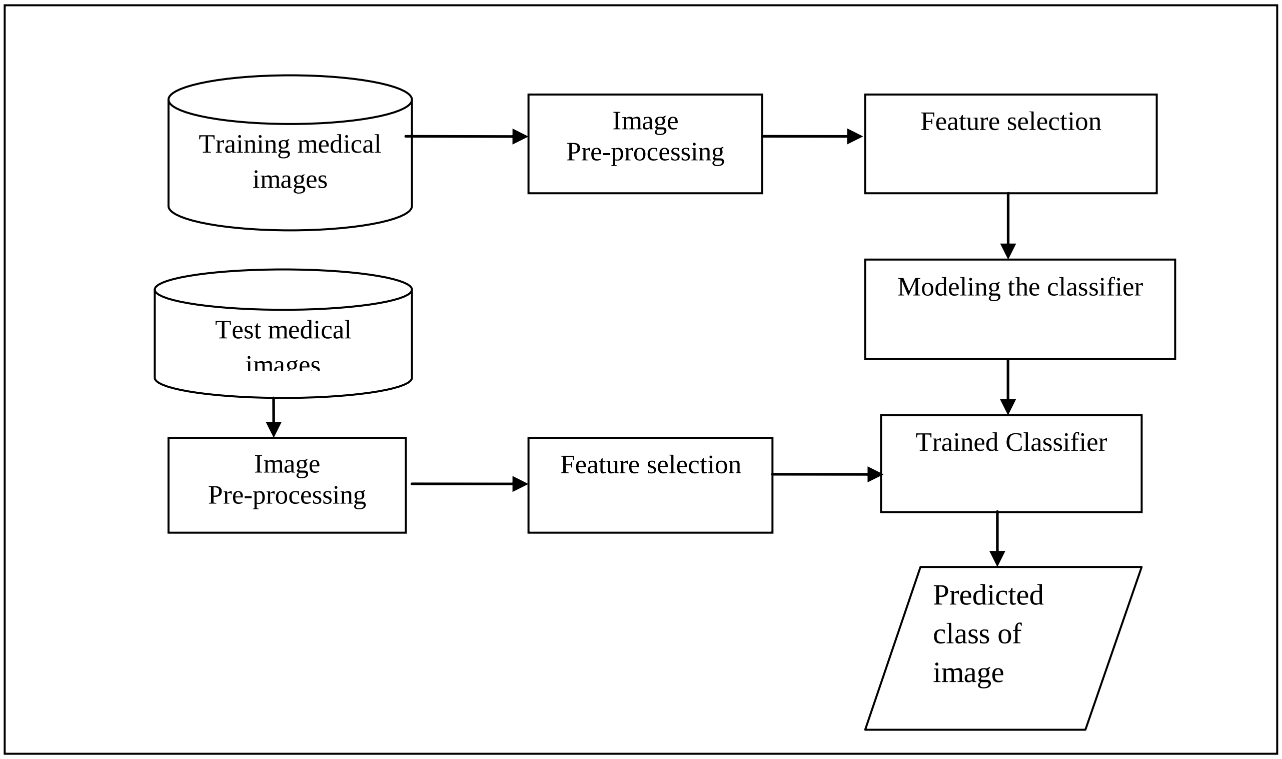


Fig 1.6 Architecture of a Medical Image Classification Model

feature selection and classification as shown in Fig 1.6. The features extracted from images are transformed into a vector space representation. The total number of features extracted from the training image corpus, decides the dimension of each image vector.

Image Pre-processing: The features that represent the content of an image are first extracted by preprocessing. This generally involves various phases namely image filtering, averaging, normalization, object identification and segmentation [6].

Image Feature Extraction: Images have a strong hold in the field of multimedia data. Identifying effective features in images and extracting them is a challenging task in the image mining process. The feature extracted directly influence the accuracy of the mining model. Features that can be extracted from an image are shape, texture, color, histogram features, etc. Features can also be extracted from images by applying transforms such as discrete cosine transform (DCT), wavelet transforms, etc. The features extracted are used to represent the image in a format suitable for the mining model.

1.2.2.3 Image Representation

In subject-based image classification, images are indexed by generating a feature vector using the features extracted that describe the image content. Let $\{F(x, y); x = 1,2, \ldots\ldots X, y = 1,2,3, \ldots .. Y\}$

Images	Features				Category
	F_1	F_2	…	F_M	
1	w_{11}	w_{12}	…	w_{1M}	C_1
2	w_{21}	w_{22}	…	w_{2M}	C_2
…	…	…	…	…	…
…	…	…	…	…	…
N	w_{N1}	w_{N2}	…	w_{NM}	C_N

Fig 1.7 Image Feature Matrix

be a two dimensional image pixel array. For color images $F(x, y)$ indicates the color value of pixel (x, y) i.e $F(x, y) = \{F_R(x, y), F_G(x, y), F_B(x, y)\}$. $F(x, y)$ is the gray scale intensity value of pixel (x, y), in a black and white image. Fig 1.7 represents the Image Feature Matrix, where N is the total number of images present in the training corpus, M is the total number of features extracted from the image collection and $\{C_1, C_2, C_3, \dots \dots \dots . C_N\}$, is one of a predefined set of image categories. In this thesis work, a set of algorithms for improving the classification accuracy of web pages and medical images are presented. The algorithms include pre-processing and classification algorithms themselves. The pre-processing algorithms include feature extraction, two novel feature selection algorithms and a feature discretization algorithm. Two new classification models for classifying web page and medical image data are also designed and implemented. The pre-processing algorithms have contributed to increase in performance of these classification algorithms.

1.3 Background Work

The following sections present a brief introduction to KDD and the significant role of data mining in KDD, the various steps involved in data pre-processing, various data mining tasks and one of the commonly used open source data mining tool.

1.3.1 Knowledge Discovery and Data Mining

Knowledge Discovery in databases (KDD) is said to be the non-trivial method of identifying valid, novel, potentially useful and patterns that are easy to understand in data. It is the overall process of converting raw data into useful information [1]. It involves the following steps iteratively namely : understanding the application domain, extracting the target dataset, data pre-processing, data mining, interpretation and using discovered knowledge. Data Mining is an integral part of KDD.

The various steps of KDD are illustrated in Fig 1.8. It involves a series of transformation steps from data pre-processing to post-processing of the mining results. Data Mining is the process of extracting non-trivial, implicit, previously unknown and potentially useful information from data [7]. Large quantities of data are explored and analyzed by automatic or semi-automatic methods for discovering meaningful hidden information. This is a multi-disciplinary field having its roots in machine learning, artificial intelligence, statistics, database management, information retrieval, visualization and pattern recognition. Traditional techniques of data exploration are unsuitable when data is enormous, high dimensional, heterogeneous and is of distributed nature. Big data mining refers to the process of extracting hidden patterns using data mining techniques from big data repositories like web pages. Research in this field is the state of the art.

Data analysts who are also called data miners begin a data mining application by first trying to understand the application domain. The target data to be mined and necessary data sources are then identified by them. Using the data collected, the data mining process is then carried out in three steps:

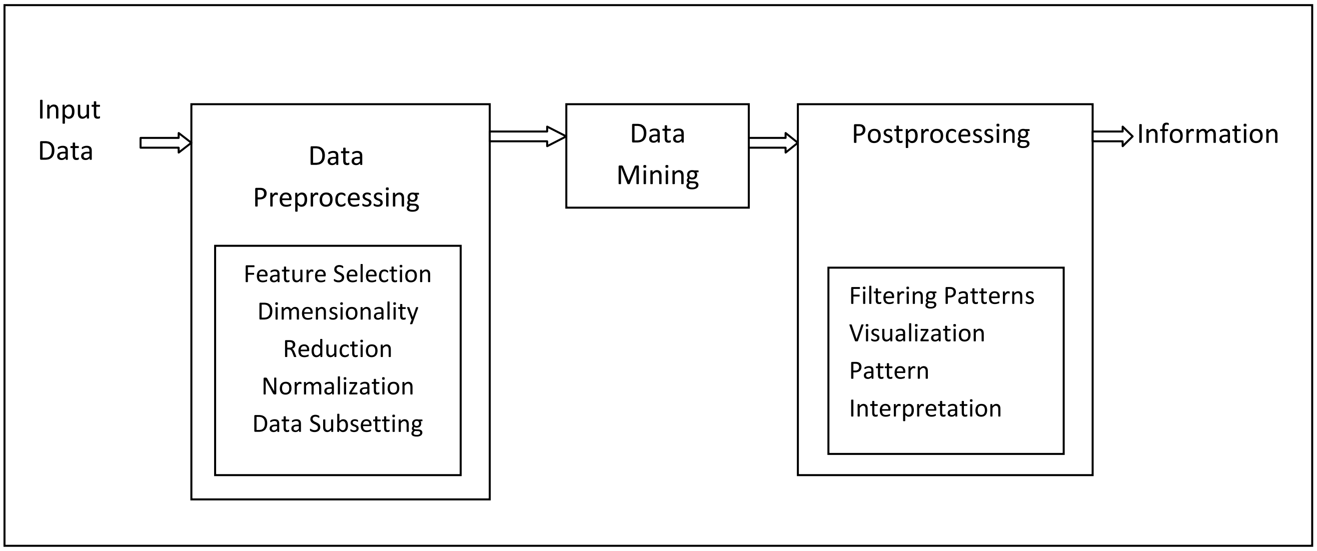


Fig 1.8 Knowledge Discovery in Databases [7]

Pre-processing: The data collected to be mined is usually not in a format ready to be mined because of many reasons. It is therefore required to clean the data such that noises and abnormalities if any present in the data can be removed. Sometimes the data might be very large and might include more number of irrelevant and/or redundant features. This raises the need to use a data reduction technique namely sampling, feature selection, feature projection and so on.

a. **Data Mining**: The mining algorithm is then applied to the pre-processed data which will produce useful patterns or knowledge. This involves functions like regression, classification, clustering, image retrieval, summarization, discovering association rules, rule extraction and extracting functional dependencies, etc.
b. **Post-Processing**: All discovered patterns are not useful in many data mining applications. Visualization and evaluation techniques are used in this step to identify which of the discovered patterns are relevant to the application of interest.

The entire data mining process always runs iteratively until the results produced are finally satisfactory, which is then used in real-world operational task.

1.3.2 Data Pre-Processing

Data to be mined are generally large real world databases and large data warehouses. Pre-processing tasks have to be done on these data before applying any of the mining tasks since such data might be incomplete, noisy and inconsistent. Data might be incomplete due to the following reasons :

- as some attributes of interest may not always be available or
- at the time of data entry, the data might not be considered so significant or
- equipment malfunction.

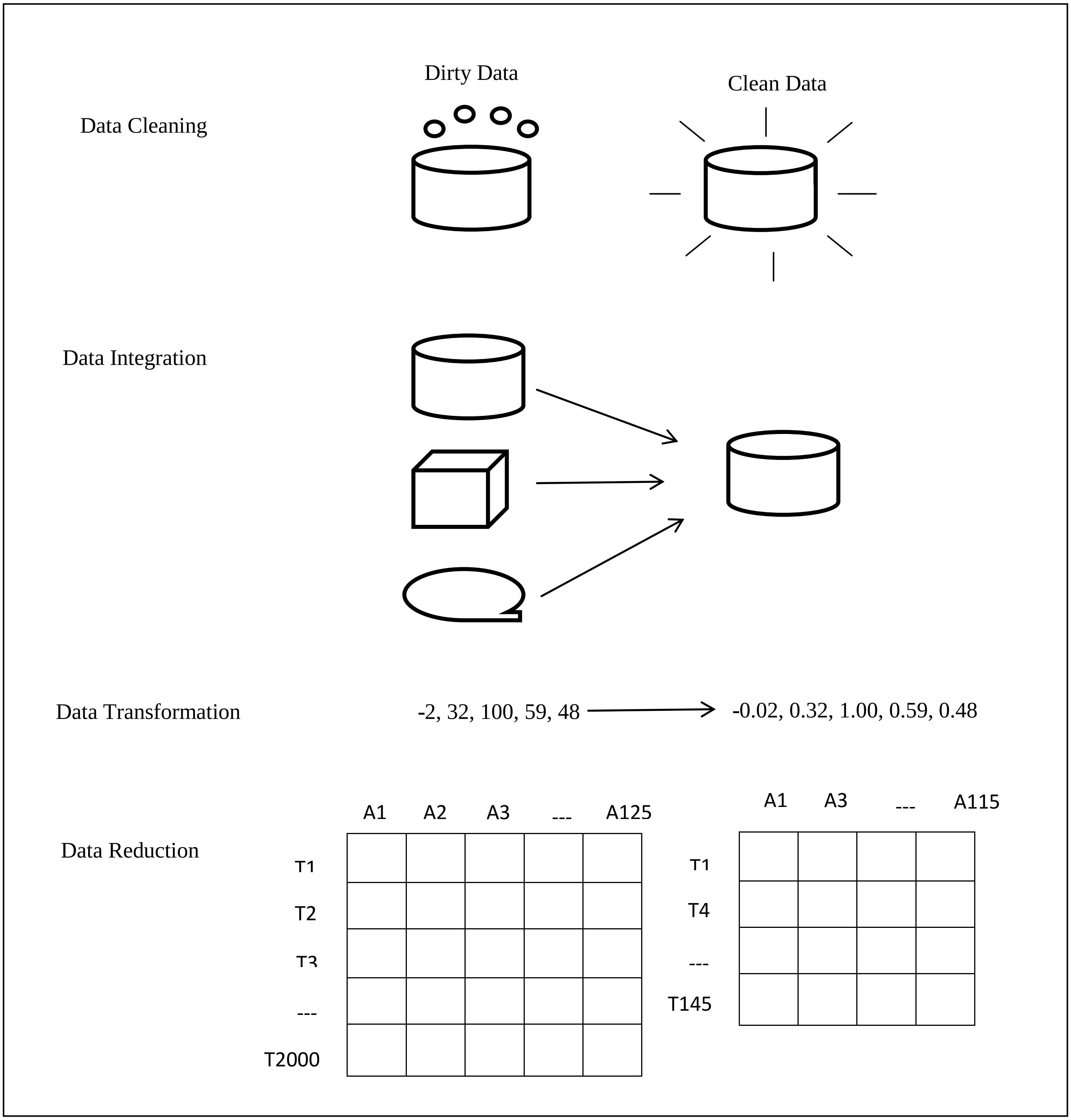


Fig 1.9 Various Forms of Data Preprocessing [16]

Some of the reasons for noisy data (having incorrect attribute values) are error in entry, error in data transmission, inconsistencies in naming conventions. Duplicate tuples also need to be cleansed. Data preprocessing has a significant impact on the mining results as the quality of the data being mined has an impact on the quality of the mining results. The various forms of data preprocessing are illustrated in Fig 1.9. [16].

Data Cleaning: Replacing the missing values in data, smoothening noisy data, identifying and eliminating outliers and resolving inconsistencies is called data cleaning. Dirty data might produce unreliable outputs. Missing values can be filled in using one of the following ways namely: 1) manually, 2) using a global constant, 3) using the mean of the attribute which has a missing value, 4) use the mean of the attribute of all objects of the same class and 5) the most probable value. Noisy data can be smoothened by regression or binning methods.

Data Integration: Merging data that comes from multiple sources into a single coherent data store as in warehousing is the process of data integration**.** These data sources might have multiple data bases, data cubes or flat files. Some of the issues to be addressed while integrating data are schema integration, and redundancy. Real world entities that come from multiple data sources and are equivalent are mapped using a Meta data. The meta data of each attribute including its name, meaning, data type, its permitted range of values and the null rules used for handling zero, blank or null values are used to resolve this entity identification problem. Redundancies can identified using correlation analysis.

Data Reduction: Real world data sets are of huge dimensions and the mining tasks applied to them suffer from the problem called the **curse of dimensionality**. As much of the data in huge dimensions is sparse in nature, it becomes meaningless for many of the mining tasks, especially those that work on proximity measures. Hence dimensionality reduction reductions play a significant role in preprocessing and have a strong impact on the mining results. **Feature selection**, feature projection, feature creation are some of the dimensionality reduction techniques. Feature selection helps in identifying and eliminating redundant features and also irrelevant features from the data set. The techniques of feature selection are generally classified into embedded, filter and wrapper methods. The flowchart in Fig 1.10 shows the various steps involved in a feature subset selection process. Feature selection methods conceptually search over all possible subsets of features. They use a 1) method used for evaluating a subset 2) a search strategy that controls the generation of the new subset of features, 3) a stopping criteria and 4) a procedure to validate the finally selected features.

The data mining algorithm itself does feature selection in the **embedded approaches.** An example is a decision tree based classifier, where the features present in the final tree which is pruned have more predictive capability. In **filter approaches** the features are selected by a different method before running the data mining algorithm. An example is correlation based feature selection method which selects features whose pair wise correlation is less. It looks for features **Wrapper methods** use the target data mining task as a black box to find the best set of attributes, but they do not enumerate all possible subsets. The subset of features selected finally should produce results that are better than or almost as good as those produced when using all the features. Another validation approach is to use different algorithms for selecting features and for generating different subsets of features. Later compare the results of running the data mining algorithm on each subset.

Data Transformation: These techniques are of two types namely normalization and discretization. **Normalization** is the process of transforming the original values of an attribute into a specified range. The three popularly known normalization techniques are 1) min-max 2) z-score and 3) decimal scaling. Normalization helps to improve the performance of the mining tasks that rely on proximity measures. It helps to avoid the proximity measures being dominated by one of the attribute which has a larger range in the data set.

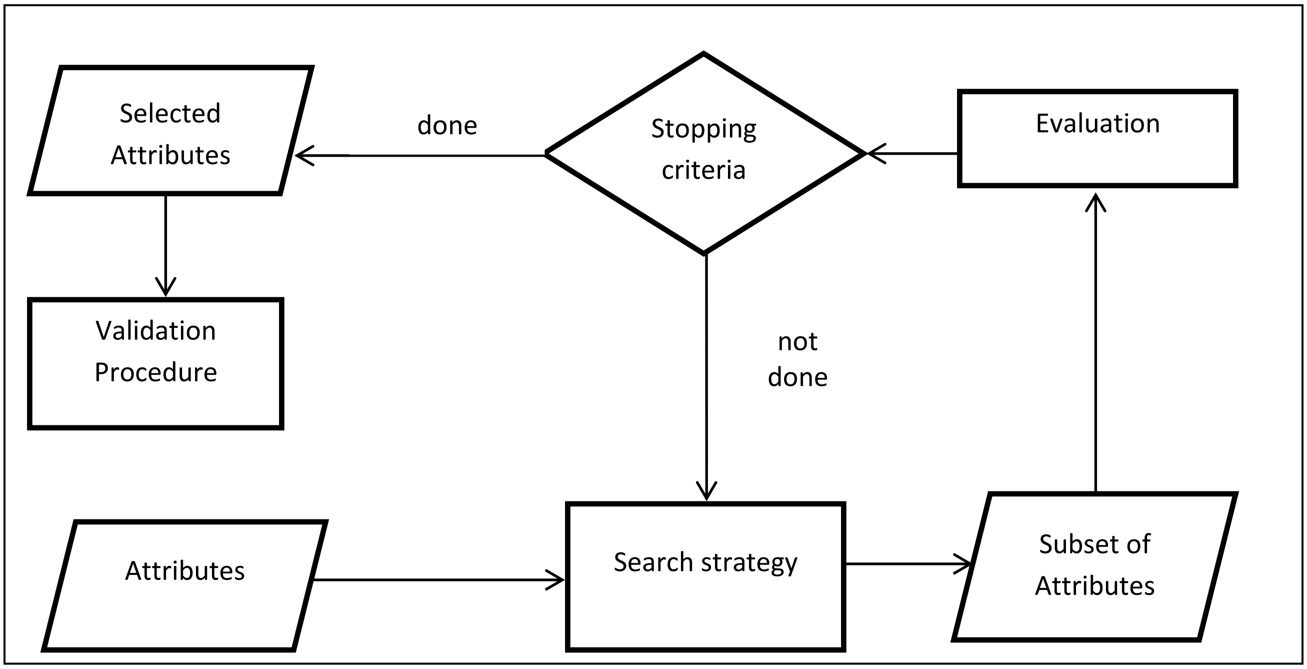


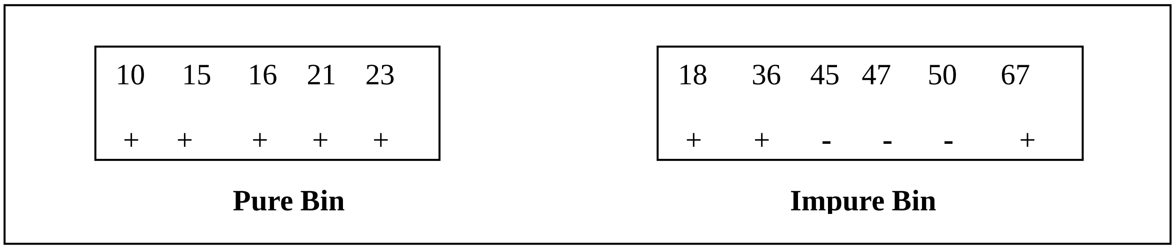
Fig 1.10 The Feature Subset Selection Process [7]

Discretization helps in transforming a numeric attribute to a corresponding discrete/categorical attribute. Some of the machine learning algorithms focuses on learning in discrete feature space. The continuous attributes need to be discretized prior to the mining task or use a different algorithm. Also many studies have shown that algorithms which can handle both continuous and discrete valued attributes perform better in the discrete domain. Induction tasks also benefit from discretization: rules with discrete values are normally shorter and easier to understand and discretization helps to improve the predictive accuracy. Apart from algorithmic requirements, discretization also helps to improve the speed of the induction algorithms.

Based on different criteria discretization methods can be broadly classified as follows:

- Supervised methods or Unsupervised methods
- Direct methods or incremental methods
- Global methods or Local methods
- Static methods or Dynamic methods
- Top-down methods or Bottom-up methods

Supervised methods use the class information in the transformation process, unlike unsupervised methods which do not use the class information. Some of the common unsupervised discretization techniques are equal frequency binning, equal width binning, clustering based techniques etc. Some of the common supervised techniques include minimum description length MDL which uses entropy measures. **Equal width binning** divides the range of possible values of a continuous attribute into N number of bins f the same size, where$bi\ n - width = \frac{(\max value - \min value)}{N}$, N being the number of bins. If there are many occurrences of one range in the data set, it would be useless for the mining task. For a continuous attribute **Equal frequency binning** divides its possible range of values into N bins, where each bin holds the same number of values. Both these methods are



10	15	16	21	23
+	+	+	+	+

Pure Bin

18	36	45	47	50	67
+	+	-	-	-	+

Impure Bin

Fig 1.11 Pure and Impure Bins

unsupervised and has the disadvantage that the user needs to specify N, the number of bins the attribute needs to be discretized. **Entropy based discretization** uses entropy measures based on the class label to identify pure bins in each iteration. If the majority of the values in a bin belong to the same class, then it is a pure bin. If there are k number of classes, entropy of a bin is calculated as $e_i = -\sum_{j=1}^{k} p_{ij}\, log_2 p_{ij}$ where m_i is the total number of values in the i^{th} interval of a partition, m_{ij} is the total number of values of class j in i^{th} interval and $p_{ij} = \frac{m_{ij}}{m_i}$ is the probability of class j in i^{th} interval. For a two class problem, where classes are + and -, if the entropy of a bin is zero, then all values in it belong to the same class. Hence the bin is said to be a pure bin. If the entropy of a bin is 0.5, the bin is totally impure and has an equal distribution of classes. Fig 1.11 illustrates this concept.

A bin with equal class distribution is highly impure and is therefore further partitioned. This discretization method needs the attribute values to be sorted and runs till the user specified number of bins is reached or some stopping criterion is met.

Discretization methods can also be classified as direct methods or incremental methods. Direct methods will divide the range of k values of a continuous attribute simultaneously for example: equal width binning, but the disadvantage with these methods is that the user has to input the total number of intervals for binning. Incremental methods begin with a simple discretization and improves in successive steps. They need to mention a stopping criteria for further splitting. These techniques can be also divided as global methods and local methods. All numeric attributes have to be discretized in the pre-processing step before inducing a classifier by the global techniques. Discretization is in-built with the classifier induction process in the local methods as in C4.5. Empirical results from research literature have stated that global methods can produce better results than the local methods. Discretization methods are also classified as static or dynamic based on whether feature interdependencies are taken into account or not. Static methods like binning, entropy based methods etc, discretize a feature into a certain number of intervals which is determined without considering the other features. Dynamic methods simultaneously search in the space of possible partitions belonging to all features and identify feature interdependencies. The final classification of discretization methods are bottom-up or top-down. Top-down methods like MDL start with a single interval which is big and having all values of a feature known. These are subsequently partitioned into smaller and smaller interval till a stopping criterion is reached. Bottom-up methods start with an initial set of boundary points. The adjacent intervals are then merged in subsequent steps tilt a stopping criterion is reached.

The field of data mining is still not matured and has applications in various fields like computer security, software engineering, web service mining, web content mining, ecommerce, customer relationship management, fault diagnosis of mechanical equipments, chem. informatics, predictive models for future energy demand, medical image mining, drug discovery, stock market price prediction, credit card fraud detection and so on.

1.3.3 Data Mining Tasks

Data mining tasks are broadly classified into predictive and descriptive tasks as in Fig 1.12. Predictive methods are used to predict the value of an unknown variable in terms of the value of all other known variables in the data set. Classification deals with discrete target variables and regression with continuous target variables. Descriptive methods are used to find patterns that can be interpreted by humans and that best describes the data. These patterns may be in the form of rules, correlations or anomalies.

Data mining tasks can also be categorised as supervised tasks or unsupervised tasks. Classification is supervised learning as the model is built from labeled examples. Clustering is an unsupervised learning as these tasks look for similarities in objects and groups them. Association rule mining done on a transaction database generates rules that show the associations/correlations between items involved in a transaction. This is mainly used for shelf management in malls and in inventory management. These algorithms have been tweaked for classification also. Given a set objects each associated with a timeline of events, sequential pattern discovery generates rules that predict strong sequential dependencies among different events. Anomaly detection is the task of detecting objects whose behavior/characteristics significantly deviate from that of all other objects in the data set.

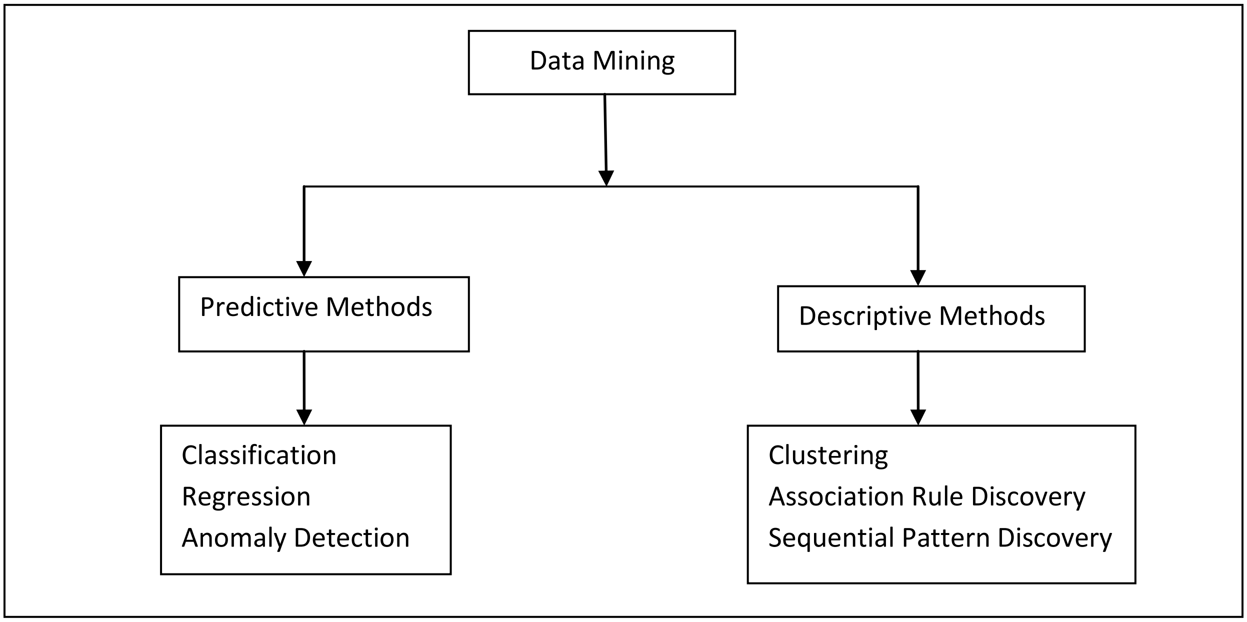


Fig 1.12 Data Mining Tasks

1.3.4 Applications of Data Mining

Some of the interdisciplinary fields in which data mining techniques are applied are explained below. Data to be mined can be structured, unstructured or semi- structured. Applying mining algorithms on traditional table data is called data mining. **Text mining** applys data mining techniques onto unstructured textual data.

Image Mining: Traditional data mining techniques were mainly used with structured data types that are in flat files. The image data cannot be interpreted directly by a machine. Its content is visual in nature and its interpretation is mainly depends on the visual system of the humans. Machine vision is done using image data. The significant features of an image is extracted from it and these features are later interpreted based on the application. This area of study is quiet challenging and is extensively explored in the field of pattern recognition and also in machine vision. It is difficult to define a one suit of algorithms which can claim to compromise an entire group of tools that are used for mining image data. **Medical Image mining** incorporates data mining techniques into medical image analysis and helps in building intelligent computer assisted medical diagnosis. It helps to identify valid, new, potentially useful and patterns that are easy to interpret in large-scale medical image data repositories. This helps in non-invasive and quantitative assessment of human body. The ever increasing amounts of patient data in the form of medical images, imposes new challenges to clinical routine such as patient diagnosis, treating them and monitoring the patient’s progress. It is possible to automate or assist physicians in their clinical decision making using data mining techniques. For example, a framework for classifying a certain class of medical images can be developed using the traditional data mining algorithms. This framework can be helpful to assist the physicians to predict the catefory of an unseen image. Research in this direction is the state of the art.

Spatial Data Mining applies data mining methods to spatial data to extract patterns in data with respect to geography. Some of the critical research challenges to be faced in this field are in developing and supporting geographic data warehouses, better spatio-temporal representations in geographic knowledge discovery and in using diverse data types [8].

Visual Data Mining deals with building interfaces that allow visual presentations of the data being analyzed by the users. These programs must be able to display data in a format which the humans can easily interpret. The overall design of these systems should be precise, easy to interpret, secure from hackers and should be inherently powerful data presentations.

Music Data Mining refers to the process of discovering relevant similarities among music corpora in order to classify music into genres in a more objective manner.

Text Mining: Text data stored in many text databases are mostly unstructured. Large amount of information is available in text or document databases, in the form of e-books, digital libraries, emails, electronic media, technical and business documents, reports, research articles, etc. This is currently under rapid development in scientific research. It involves multi-disciplinary scientific fields like string matching, artificial intelligence, machine learning, information retrieval, natural language processing, statistics, information theory, soft computing, etc. Since texts are unstructured data, text mining involves additional challenges unlike traditional data mining.

Web Mining: Most of the real world data sets are semi structured like web pages. They have tables, textual content and HTML structure embedded in them. **Web mining** applies data mining techniques onto web data for discovering useful patterns from the web hyperlink structure, page content and usage data. It has additional challenges than pure text and data mining.

The largest data source that can be accessed publicly in the world is the web. In the last decade it has received a rapid growth. Thousands of web pages are being added to this repository every day. Mining useful patterns and knowledge from the web has become quite challenging due to its characteristics.. Web data is dynamic in nature, and has data of all types eg., structured tables, web pages that are semi-structured pages, textual data that is unstructured, and multimedia files such as image files, audio files and videos. Information available on the web is noisy and has links to other web pages. Only a percentage of information on the web is considered significant for a particular application and the remaining are treated as noise. All these characteristics of web data have imposed additional challenges for mining and discovering useful patterns and knowledge from it. Although many data mining techniques are used in web mining, it does not apply the conventional data mining techniques directly because of its heterogeneous nature [9]. Web mining tasks are mainly classified into web content mining, web structure mining and web usage mining.

Web structure mining is the process of discovering useful patterns from the web hyperlinks. This helps the search engines to discover important web pages with the help of the links. User communities who share common interests can also be discovered. Such tasks cannot be performed using traditional data mining methods since no link structure is present in tables.

Web Content mining is the process of extracting useful information from the data embedded within the web pages. They can be automatically classified/clustered into one of pre-defined categories. This task is known as web page classification. These are same as traditional data mining. Further, reviews given by customers on a certain product and forum postings can be mined for discovering consumer opinions. These tasks are unlike traditional data mining.

Web usage mining is the process of extracting useful information from the web usage logs. Information about the web access patterns of users can be mined by this task from these logs. They give the details of every click-stream made by individual users. This applies data mining algorithms after intensive pre-processing of the web log data.

1.3.5 Data Mining Tools

Data mining tools provide an easy interface for a data miner to extract hidden knowledge from the data of interest. Several data mining tools help research in machine learning of which some are commercially available and some are freely available in the market. WEKA is one of the open source data mining tool used commonly by the machine learning community. It is developed at the University of Waikato in Newzealand and named after the bird weka [10]. It has a suite of tools and functions for data mining tasks which include filters, select attributes, classify, cluster and associate attributes. It is developed in Java and its classes can be easily imported in any application.

```
% The weather data

@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny, 85, 85, FALSE, no
sunny, 80, 90, TRUE, no
overcast, 83, 86, FALSE, yes
rainy, 70, 96, FALSE, yes
rainy, 68, 80, FALSE, yes
rainy, 65, 70, TRUE, no
overcast, 64, 65, TRUE, yes
sunny, 72, 95, FALSE, no
sunny, 69, 70, FALSE, yes
rainy, 75, 80, FALSE, yes
sunny, 75, 70, TRUE, yes
overcast, 72, 90, TRUE, yes
overcast, 81, 75, FALSE, yes
rainy, 71, 91, TRUE, ?
```

Fig 1.13 The Weather Data Set in ARFF [10]

The data to be mined using this tools has to be in a format called attribute relation file format or ‘arff ’in short. Fig 1.13 shows a sample weather data set in arff. This file format uses three tags starting with the symbol @ namely @relation tag, @attribute tag and @data tag. The @relation tag is used to give the name of the relation. The @attribute tag gives a description of each attribute namely its name, type and possible values if it is nominal. The @data tag gives the actual values of each object with its corresponding attribute values separated by a comma. For supervised learning tasks like classification, the tool assumes that the last column is the dependent attribute.

1.3.6 Supervised Machine Learning

Supervised machine learning uses algorithms that learn from externally supplied labeled examples to produce a general hypothesis. Future instances are then predicted using the hypothesis learnt. These methods try to build a concise model which clearly identifies the distribution of class labels in terms of features used in prediction also called the input features. The final classifier is then used to predict the class labels of the test instances whose predictor features have known values, and unknown value for the class label. Naïve Bayes (NB) uses Bayes theorem and estimates the probability of a certain test record to belong to a certain class. It also assumes that features in the data set are independent of each other.

Out of the plenty of applications for Machine Learning (ML), data mining is the most significant .It is often common for a statistician to make mistakes while analyzing the data and also when trying to identify relationships between the various features in the data set. Therefore finding solutions for certain problem becomes very difficult for them. In such situations, Machine learning can be successfully used and it also helps to improve the efficiency of the results mined.

The input data set used by the machine learning algorithms require that all training examples should be represented by the same number and nature of features. The features may be continuous in nature or categorical type or binary. The learning algorithm is called supervised if the instances have predefined output labels as illustrated in Table 1.1. However, in unsupervised learning, the training instances are unlabeled. Unsupervised methods like clustering help researchers to discover previously unknown, but useful, classes of items.

Issues of Supervised Machine Learning Algorithms

Inductive machine learning learns a set of rules from training instances in the input data set. More generally speaking, they refer to the process of creating a new classifier which knows to generalize from new instances. Various steps involved while applying supervised ML to a real-world problem are explained below:

Table 1.1 Instances with Known Class Labels

Instance No.	**Attribute 1**	**Attribute 2**	**------**	**Attribute n**	**Class**
1	Aaa	A		Aa	Course
2	Aaa	A		Aa	Course
3	Aaa	A		Aa	Student

Data Collection: If the end user is an expert then he/she should identify which attributes/features would be more significant or relevant to the mining task. For example student id is irrelevant to the task of predicting student’s CGPA. If the end user is not an expert, then by a ‘brute-force’ approach collect all available features with the hope that the informative ones can be isolated later. However, a dataset that is collected by the ‘brute-force’ method cannot be directly used for induction. Since many times it may contain noise and missing values for some feature it needs to be pre-processed well before induction.

Data preprocessing. Several methods are suggested by researchers to handle missing data depending on the situation. A survey of various techniques for outlier detection, their advantages and disadvantages is introduced in [11]. It is suggested that instance selection methods are dual purpose. They are used in eliminating noise form the given data set and they also minimize the infeasibility of learning from very large datasets. Instance selection is also defined as an optimization problem, since it attempts to maintain the quality of the mining results at the same time minimizing the size of the training set. It helps a data mining algorithm to work efficiently and produce high quality results especially when dealing with very large data sets. Various approaches for instance selection from large data sets in described in [12].

Feature subset selection is the process of identifying and eliminating features that are irrelevant and redundant in the given data set [13]. It therefore helps to reduce the dimensionality of the data and hence the performance of the data mining algorithms will be fast and more effective. The accuracy of the supervised ML algorithms are greatly influenced by the fact that many features in the given data set are dependent on one another. This problem is usually addressed by either generating new features from the original set of features or transforming the original feature set into a new feature set. These techniques are called feature construction and feature transformation. More accurate classification models could be induced from these features that are newly created. In addition, identifying these meaningful feature set enables the induced classifier to comprehend better. Feature subset selection using this method also helps to understand the learned concept better.

1.3.7 Naïve Bayes Classification

Naive Bayes (NB) is a probabilistic classifier that uses the Bayes theorem (or Bayes rule) and has a strong attribute independence assumption. A hypothesis that the given test data may belong to a certain category is first developed by this classifier. The probability of this hypothesis to be true is then calculated using the Bayes theorem. In medical diagnosis where it is uncertain to decide certain problems the NB models provide the most practical approach.

Bayes Theorem:

Let $P(A)$ refer to the probability that event A will occur. $P(A/B)$ be the probability that event A will occur, given that event B has already occurred. So, it is the conditional probability of A. With these terms the Bayes theorem [14] can be defined as in Equation 1.6

$$P(A/B) \quad = \frac{P(B/A)P(B)}{P(B)} \tag{1.6}$$

If X is an object to be classified using Bayes theorem can be used to compute the probability of X to belong one of the classes C_1, C_2, C_3 etc. by calculating $P((C_i|X)$. Once these probabilities are estimated for all the classes, X is simply assigned to the target category that has the maximum conditional probability. The probabilities $P((C_i|X)$ are calculated as in Equation 1.7

$$P(C_i|X) = \left(P(X|C_i)\,P(C_i)\right) / P(X) \tag{1.7}$$

where $P(C_i|X)$ gives the probability of object X to belong to class C_i. $P(X|C_i)$ is the probability of getting attribute values X if we know that it belongs to class C_i. $P(C_i)$ gives the probability of any object to belong to class C_i. $P(X)$ is the probability of obtaining the attribute values X, irrespective of the class the object belongs to and is calculated as illustrated by Equation 1.7. Since $P(X)$ is independent of any class C_i, it can be omitted when estimating $P(C_i|X)$. So the modified Bayes theorem used for predicting the class probability of any object X is defined as in

$$P(C_i|X) = \left(P(X|C_i)\, P(C_i)\right) \tag{1.8}$$

Equation 1.8. The probability of the output attribute can be estimated by computing estimates of the probabilities of the input attributes. This doesn’t require the values of all the input attributes to be known. This is an advantage of Bayes theorem. This is one of the top ten classification algorithms and has been widely used for medical image classification. Although it assumes that all features are independent of each other which is unlikely most of the time, it possesses several advantages like simplicity, computationally efficient, requires relatively less data for training, do not have lot of parameters and is robust to noisy and missing data. [15]. As it uses all parameters in decision making, it is appealing to physicians, as the decision seems to be natural. The performance of NB classifiers can be improved through discretization.

1.3.8. K Nearest Neighbor Classification

One of the simplest, easy to implement and one of the top ten classification algorithms is the k nearest neighbor also called the KNN classification algorithm in short. Unlike Naïve Bayesian classifier, this is a lazy learner, since the induction process starts only after receiving a test object. This requires i) the training data set, ii) a distance measure to find the distance between the test record and every training record and iii) the value of k, which is the number of nearest neighbors to retrieve from the training set. With all these inputs this it predicts the class of a given test object, through a sequence of steps as described below [16]:

***Algorithm* kNN**

Input: k - the total number of nearest neighbors to retrieve, D - the training records and z – the test record.

Output : $y^{'}$ - the predicted class of the test record

```
for each test record $z = (x', y')$ do
    1.  Find the distance $d(x^{'}, x)$, between z and every training record $(x, y)\ in\ D$the training set.
    2.  Select the k number of closest training records to z. Let this be $D_z$. This will be a subset of D.
    3.  The class of the test record z is now given by simple majority voting as
        $y^{'} = \underset{v}{argmax} \sum_{(x_i, y_i) \subseteq D_z} I(v = y_i)$
end for
```

Disadvantages of KNN:

1. The choice of k is critical. A high value of k would include far away points in the neighborhood. A low value of k, may make the model susceptible to overfitting, because of the presence of noise in the training set.

2. The traditional KNN uses Euclidean distance as a similarity metric to find the nearest neighbors of an unseen instance. This measure has a drawback that all features are given equal significance in this measure. It is therefore possible for a feature with a higher range of possible values to dominate the distance measure and hence the classification results. To avoid this, the data needs to be normalized before classifier induction.

3. A tie may occur when using the simple voting on the class of the k nearest neighbors to decide the class of the test record. One direct solution to resolve this issue is to choose one of these classes at random and this becomes the class of the test record.

4. As it needs to find the distance between the test and every object in the original data, the algorithm requires more memory to store the entire training data.

1.3.9. Association Rule Mining: This data mining task generally done on a transaction data base is used to find items that are strongly associated/correlated in the data base. The traditional application of this task is the market basket data analysis. This identifies the products that are purchased frequently together by customers in a supermarket. The rules generated by the descriptive model helps the shopkeeper in shelf management, sales promotion and targeted marketing. Item sets are collection of one or more items involved in the transaction database. It uses two metrics called support and confidence on the item sets as defined in Equations 1.8 and 1.10 respectively. For a rule $A \rightarrow B$ support of the rule is defined as the fraction of transactions that involve both the items A and B. Confidence of the rule is defines as the ratio of support of item A and item B together to the support of item A.

$$support(A \rightarrow B) = \frac{Number\ of\ transactions\ having\ A\ and\ B}{N} \qquad (1.9)$$

$$confidence\ (A \rightarrow B) = \frac{support\ of\ (AB)}{support\ (A)} \qquad (1.10)$$

Given a transaction data base T, a user specified threshold value for support called min_sup and a threshold value for confidence called min_conf, this task finds association rules that are strong in two steps [14]:

a. Identify all frequent item sets which are item sets whose support is greater than or atleast same as min_sup.

b. Generate strong rules using frequent item sets, which are rules whose confidence is greater than or same as min_conf.

Several algorithms for generating frequent item sets have been indicated in literature such as Apriori, FP-Growth etc. The rules generated help the shopkeepers to promote the sale of items/item sets, shelf management and inventory.

1.4 Research Gap

WPC has been attempted through various approaches as seen in research literature namely, using the URL of a web page, using the HTML tags, using the visual features in the web page, using hyperlinks, using the structure of the web page, clustering approaches, using the web page summaries, using the on page textual features, using features of neighboring web pages etc. Due to the sheer volume of data on the web, manual categorization of web pages is always incomplete. Using URL features although eliminates the need to download a web page, but does not have ideal accuracy; meta tags (HTML tags) cannot be used, since it is possible that a the web page author might include keywords irrelevant to the contents of the web page just to increase the hit-rate of the web page; classifying using visual features depend on the human expertise used to design the web page and are computationally complex; the link based and also structure based approaches are not useful in situations that need to classify a web page from its print version, since they do not contain any link and moreover all web pages do not contain images; clustering approaches are computationally intensive; performance of the classification models using web page summaries depends on the quality of the web page summaries generated; classifying a web page using features of its neighboring pages is based on a ‘strong assumption’ that a web page is more likely to be surrounded by other web pages which belong to the same category. It is proved that this strong assumption works well for classifying using broad categories, but the results for fine grained categories is poor. Also, this strong assumption doesn’t work well in functional classification. Obtaining the features of neighboring web pages is computationally more expensive. It is expressed in [26] that many researches should concentrate on simpler approaches first and the complex ones later.

Motivated by these facts, this thesis investigates the various methods that could improve web page classification using the textual features present directly in a web page and machine learning methods. It is inexpensive to extract on page features from a web page. The web pages are transformed into a format suitable for the traditional machine learning methods. Their performance is also improved by a set of pre-processing tasks. A novel framework for classifying web pages in this direction is presented and experimented in this thesis.

This thesis also presents a classification model for medical images using data mining methods. Different methods have been used to classify medical images namely pattern recognition methods, wavelet based methods, fractal theory based methods, non-linear distortion models, etc. However these methods use the features extracted using intensive image-processing techniques namely using Gabor features, wavelet features, region based features; Gray level Co-Occurrence Matrix features (GLCM) and so on. Extracting the right set of features to train a classification model requires human expertise. In order to manage the accumulating volume of medical data, it is required to minimize human intervention during the data preparation and preprocessing phase. Data Mining is a recently booming research field, and has been widely applied in various domains. In this thesis, a novel framework is proposed for classifying medical images using data mining techniques. It uses simple statistical features obtained from medical images to classify them. Extracting these features requires less domain knowledge and the various pre-processing phases used in this thesis have helped to improve the model's performance.

1.5 Organization of the Thesis

A novel framework for classifying web page and medical image data sets is proposed in this thesis. The framework involves various phases including pre-processing and classification. Following illustrates the method in which the rest of the thesis chapters are organized.

Chapter 2 presents an intensive survey of the various methods and approaches that have been adopted in research literature for web page and medical image classification. It highlights the latest state of the art techniques in this field.

Chapter 3 presents the details of the present framework with the algorithms for classifying web page and medical image data sets. It discusses the various phases of the present framework namely, feature extraction, feature selection, feature discretization and classification. Two novel feature selection methods are presented.

- The first one is a hybrid model which uses the correlation based feature selection CFS followed by the decision tree induction algorithm namely C4.5.
- The second method uses the Ward's minimum variance measure for dimensionality reduction. It identifies clusters of redundant features using the Ward's measure. Later, it uses the information measure to select the most predictive feature from each cluster.

  This chapter also identifies that the predictive accuracy of many of the supervised machine learning classifiers can be improved by modeling them in the discrete domain than in continuous domain. It also presents two new classification methods for web page and image data sets namely,

- A classifier that uses Bayes theorem called PWPC/PMIC, a probabilistic web page classifier/ medical image classifier.
- A modified kNN classifier called MKNN which uses the interestingness measures which are originally used in association rule mining.

Chapter 4 presents the description of the data set used for the experimental analysis. It also includes the results and discussion of each phase of the present work.

Chapter 5 presents the general conclusions of the thesis and specific contributions of this thesis work to research literature. It also throws light on future research in this direction.

CHAPTER 2

RELATED WORK

In this thesis algorithms for improving automatic classification of subject based classification of web page and medical image data sets are designed and implemented. These algorithms include methods for feature extraction, feature selection, feature discretization and classification. The present algorithms are implemented both on web page and medical image data sets. This chapter gives a discussion of the various approaches and methods for web page and medical image classification as found in research literature. Section 2.1 is a survey of the various methods used for web page classification WPC found in research literature. Section 2.2 is a survey of the various methods recommended for medical image classification MIC found in research literature and Section 2.3 is a summary of the contents in this chapter.

2.1 Survey of Web Page Classification Methods

Many approaches for automatic web page classification have been witnessed over years in research literature. With no preprocessed data there is no quality mining results. The performance of the web page classifiers are improved from different perspectives, namely by dimensionality reduction (feature selection), using the word occurrence statistics in a web page (content based), using the relationship between different web pages (link based), using the association between queries and web pages (query log based) and by using the structure of the page, the images, links contained in the page and their placement (structure based).

A WPC model is proposed by Shen, Chen, Yang, Zeng, Zhang, Lu and Ma [17] using the summary of the web page generated by human experts. By exploiting the characteristics of Chinese web pages, a new feature selection method by assigning weights to the HTML tags is proposed by Chen, Du, Zhang and Han [18]. The structure of the web pages are used to classify them into information, research and personal home pages [19]. Blocks [20] are units that compose a web page namely, paragraphs, tables, lists and headings. The association between these blocks, web pages and the queries are used to frame a query with content based classification framework to classify a web page. Visual features of a web page like color and edge histograms, Gabor and texture features [21] are used to classify it. These approaches of web page classification cannot be applied in situations which suffer from hardware and software limitations. Further, they require lot of human expertise and are computationally complex. A web page classification model using the URL of a web page is proposed Kan [22] and Kan and Thi [23]. While not of ideal accuracy, this approach doesn't require to download the web page. Hence it is especially useful when the web page content is not available or time/space efficiency is of more significance. Classifying web pages using HTML tags is proposed Kwon and Lee [24],[25]. A modified k-nearest neighbor classification algorithm is proposed where terms present in tags are assigned more weights. But, most of the HTML tags concentrate on representations instead of semantics, the web page authors can create different but conceptually equivalent tag structures. Therefore WPC using this approach suffer from the inconsistent formation of HTML documents. The various technologies that can be explored in web information extraction have been explored by Xhemali, Hinde and Stone [26] and the authors have expressed their concern that many researchers start with the complex approaches directly rather than trying out the simpler ones first.

Machine learning methods [27], [28] have also been tweaked to improve performance of content based classification in this domain. Further, when the learning task is to build a model with accurate classification, C4.5 and NB are two very important machine learning algorithms for achieving this task because of their simplicity and high performance. NB models are popular due to their conditional independence characteristic. Each attribute contributes towards the final decision equally without the influence of the other attributes over it. It is proved by Xhemali, Christopher, Hinde and Stone [29] that these models are fast consistent, easy to maintain and accurate in the training courses domain. NB classifier based on Independent Component Analysis [30], Hidden Naïve Bayes [31] with Symmetrical Uncertainty for word selection perform more satisfying in web page categorization. It is identified by Balamurugan, Pramala, Rakalakshmi and Rajaram [32] that during the DT induction algorithms, a tie appears when there are equal proportions of the target class in the leaf nodes, which leads to a situation where majority voting cannot be applied. The DT algorithm is improved to handle those exceptions.

Due to the sheer volume of data on the web, manual categorization of web pages is always incomplete. Clustering approaches are computationally expensive and the full potential of these algorithms depends on making several design choices carefully [33]. Meta tags cannot be used, since there is a possibility for the web page author to intentionally include keywords which do not reflect its content, merely to increase its hit-rate. Link based and structure based approaches also fail in scenarios to correctly classify a web page from its print version, since there is no link in it and not all web pages contain images. But its performance depends on the quality of the web page summaries. Motivated by these facts, this thesis investigates the various algorithms that could improve web page classification using the contents of the web page and machine learning methods. The web pages are transformed into a format suitable for the traditional machine learning methods. Their performance is also improved by a set of pre-processing tasks. A survey of improving web page classification in this direction follows.

Since web pages are of higher dimensions and have noisy information, they need to be properly preprocessed which would otherwise increase the learning time and complexity of the classifiers.

Feature selection is one way of solving the curse of dimensionality for content based web page classifiers. It has been an active research area in pattern recognition, statistics, and data mining communities. The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. Feature selection can significantly improve the comprehensibility of the resulting classifier models and often build a model that generalizes better to unseen points. Further, it is often the case that finding the correct subset of predictive features is an important problem in its own right.

A study of the appropriate feature selection techniques for WPC is explored [34] to obtain a minimum number of highly qualitative features. CFS subset evaluator is combined with term frequency to achieve good classification accuracy. Three distinct features of a web page namely, the URL, title and meta data which are believed to have more predictive information about a web page are used [35] with machine learning methods to classify a web page. The output of PCA principal component analysis is combined with a manual weighting scheme to classify web pages using neural networks [36]. A fuzzy ranking analysis with discriminating power measure [37], rough set theory [38] and an integrated use of ant colony optimization with fuzzy-rough sets [39]**,** is used to reduce the dimensionality of web pages. A study of rough sets, their extension and applications in data mining is explored [40]. A new feature selection method that incorporates hierarchical information about the categories is used by Peng, Ming and Wang [41]. This prevents the classifying process from going through every node in the hierarchy. On – page features (content based) and features of neighboring pages (context based) are used [42] to classify a web page. A genetic algorithm that determines the best features for a given set of web pages is proposed by Ozel [43], to decrease the feature space. The feature space can also be reduced by identifying and eliminating redundant (relevant) features in a web page.

Various feature selection methods for WPC were discussed above. Apart from this, several feature selection methods for traditional structured data sets are also proposed in literature. They can be broadly divided into three categories namely filter models, the embedded models and wrapper models. The filter model uses the general characteristics of the data to evaluate a feature subset and does not involve any mining algorithm and hence not biased to it. Features are selected before running the mining algorithm using some approach that is independent of the mining task. For example, sets of attributes whose pair-wise correlation is as low as possible can be selected. In embedded models feature selection occurs as a part of the data mining task. The mining algorithm itself decides which attributes to use and which to ignore. For example, the decision tree induction algorithms can be used to select the best subset of features which are those features that are present in the final pruned tree. The wrapper approaches use the target data mining algorithm as a black box to identify the best subset of features, but without enumerating all possible subsets. The feature subset selection algorithm forms a wrapper on top of the induction algorithm. The optimal subset of features is found by adding or deleting features from the input feature set, depending on the accuracy of the induction algorithm itself. Correlation based feature selection method which is a filter model for UCI data sets is proposed by Hall [44]. It selects subset of features that are highly correlated with the class label and low inter-correlation using a pair-wise selection strategy. A filter method which can identify relevant features and the redundancy among relevant features without pair-wise correlation analysis is proposed Yu and Liu [45]. A modified pair-wise strategy using mixed univariate and bivariate feature evaluation based on correlation between the features is proposed by Michalak and Kwasnicka [46]. The decision tree algorithm is used to select the best statistical features for fault diagnostics of roller bearing [47]. The wrapper technique proposed by Kohavi and John [48] starts with an empty set of features and searches the feature space for the optimal subset. A major disadvantage with these approaches is the computational cost involved. Faster feature subset evaluation techniques such as CFS correlation based feature selection, information gain, support vector machine based feature evaluation, etc have been evolved to reduce the computation.

A feature selection method using Bayes theorem is proposed by Balamurugan and Rajaram [49] and has been experimented on the bench marking UCI data sets. The dependence between two attributes is determined based on the probabilities of their joint values that contribute to positive and negative classification decisions. A new feature subset selection method using class association rules is proposed by Zhang and Zhou [50]. First association rules with features as antecedents and classes as consequent are generated. Features that are present in the strong rules generated are the optimal features identified. A feature selection algorithm using constraint based association rule mining is proposed by Wang and Song [51]. Experiments on UCI data sets show that this algorithm outperforms many of the existing algorithms. A feature subset selection method using the JRip classifier and association rule mining is proposed by Shahzad, Asad and Khan [52]. First the JRip classifier is used to extract the rules form the given data set and then association rules are used to rank the features. A propositional FOIL algorithm is used to select feature subset for high dimensional data [53]. It first merges all features that appear in the antecedent of the FOIL's rules and later uses a metric called CoverRatio to obtain the final feature subset.

Machine learning algorithms have been applied in real-world classification tasks like WPC. Some of these algorithms focus on learning in discrete feature space. They can be applied only to data described by discrete numerical or nominal attributes (features). In the case of continuous attributes, there is a need for a discretization algorithm that transforms continuous attributes into discrete ones, or to use a different algorithm. Also, algorithms which can handle both continuous and discrete features perform better with the discrete-valued attributes. Discrete values play an important role in data mining and knowledge discovery. Many studies have shown that induction tasks can benefit from discretization: rules with discrete values are normally shorter and easy to understand and discretization can lead to improved predictive accuracy [54]. Apart from the algorithmic requirements, discretization also helps in increasing the speed and accuracy of induction algorithms. It makes the results of the induced classifier shorter, compact and easier to understand than those generated using continuous features. Feature selection and discretization are the major preprocessing done before induction.

Most of the machine learning algorithms take a longer induction time, when the data to be modeled is continuous in nature. Their performance can be improved by discretizing the data into finite intervals. Discretization methods are classified into supervised and unsupervised depending on whether the class information is taken into account or not during the discretization process. Two examples of unsupervised methods are equal width binning and equal frequency binning. 1R is a binning method that uses the class information. It first sorts the continuous values and then divides the range of continuous values into a number of disjoint intervals. The boundaries of these intervals are then adjusted according to the class labels associated with them [55]. A comparison of unsupervised and supervised methods is reported by Dougherty, Kohavi and Sahami [56] and the authors conclude that supervised methods give less classification errors than the unsupervised ones.

Another way of categorizing methods is direct vs. incremental [57]. Direct methods like simple binning divide the range of continuous values into k intervals simultaneously, where k is an input from the user. Incremental methods start with a simple discretization and proceeds by merging or splitting two adjacent intervals until a stopping criterion is met. Another distinction of discretization methods is global or local. Global methods discretize each attribute as a pre-processing step, before the mining algorithm is induced. Local methods discretize a feature during the induction process. Empirical results show that global methods perform better than local methods. The distinction between static and dynamic methods depends on whether feature interdependencies are considered during discretization. Static methods discretize each feature independent of the other like binning, entropy based partitioning and the 1R algorithm. Dynamic methods capture interdependencies in feature discretization. Top-down discretization methods start with one big interval and gradually split this into smaller and smaller sub-intervals. Bottom-up methods start with a number of intervals and merge them into larger and larger intervals until the stopping criterion is met.

A recent survey of discretization techniques major theoretical issues and future research directions are covered by Kotsiantis and Kanellopoulous [58]. A supervised discretization algorithm, CAIM (class-attribute interdependence maximization) [59], is proposed to maximize the class-attribute interdependence and to generate a (possibly) minimal number of discrete intervals. The algorithm does not require the user to predefine the number of intervals, as opposed to some other discretization methods. Marzuki and Ahamad [60] highlight that discretization algorithms designed to operate in one domain are inappropriate to the other domain. A discretization method using the Chi-square (X^2) statistical measure is proposed by Kerber [61]. Using this measure the similarity of two adjacent intervals is found. It then tests the hypothesis that two adjacent intervals are independent of the class. If they are independent, they are merged. Minimum Description Length (MDL) uses the entropy measures to evaluate the candidate cut points [62]. It is stated that optimal cut points for entropy minimization must lies between examples of different classes. The minimum description length principle MDLP is used as a stopping criterion. The MDLP principle was originally used to find the cost of communication between a sender and a receiver. If a cut-point is identified for a set of values, then it is acceptable only if the cost of sending the values after partitioning them at that cut point is less than the cost of sending the values before partitioning.

Since real time data sets have continuous features, Liu and Setiono [63] have proved that feature selection can be simultaneously done while discretizing the continuous data. NANO, a supervised feature selection and discretization algorithm is implemented [64]. Experimental results of NANO over the UCI data sets have reduced the number of features significantly and also increased the predictive accuracy. It is stated by Hacibeyoglu, Arslan and Kahramanli [65] that classification accuracy can be improved with discretization on data sets including continuous features. The effectiveness of nine discretization methods with Naïve Bayesian classifier is evaluated [66] and a new discretization method namely weighted non-disjoint discretization is proposed. A new discretization method based on equal width binning and error minimization is introduced [67] and this method is found to improve the classification performance of Naïve Bayes NB as compared to J48 classifier. Empirical analysis of why discretization improves the performance of Naïve Bayes and a lazy discretization method is proposed by Hsu, Huang and Wong [68]. A novel incremental discretization method for NB called incremental flexible frequency discretization IFFD, which discretizes the values of a quantitative attribute into intervals of flexible sizes is proposed by Lu et al [69]. It allows online insertion and splitting of intervals. Hence this incremental discretization enhances the incremental learning power of NB. Most of the supervised discretization methods need the attribute values to be sorted. An unsupervised method based on k-means clustering is proposed by Joita [70], which avoids the O(n log n) time requirement for sorting. However, this algorithm needs the user to specify the value of k, the number of intervals or bins.

An intensive survey of web page classification approaches, suitable features and algorithms is done by Qi and Davison [71]. For classification tasks, supervised machine learning techniques are more appropriate and a survey of supervised machine learning techniques is proposed by Kotsiantis [72]. A review of the top 10 data mining algorithms as identified by the IEEE international Conference on Data Mining (ICDM) in December 2006, is discussed in detail by Wu et al[73]. These algorithms are the decision tree- based C4.5, k-means, Support vector machine SVM, Apriori for association rule mining, Expectation Maximization EM, PageRank, an ensemble method AdaBoost, KNN, Naïve Bayes NB and CART. It includes a description of these algorithms, a discussion of their impact, current review and further research in this direction.

Two WPC models are presented in this thesis: one is a modified KNN model and the other is based on Naïve Bayes theorem. The following section is a survey of the methods used for improving KNN classifiers. These classifiers are identified as the most straightforward in machine learning [74]. These classifiers are more popular today as the increase in computing power has resolved the issues of poor run-time performance. They are very sensitive to irrelevant or redundant features since all features participate in the distance measure to find the nearest neighbors to the test data. This problem is overcome in this thesis by careful feature selection and feature weighting. A survey of improving KNN classifiers from different perspectives is discussed in detail by Jiang, Cai, Wang and Jiang [75]. It includes improving by distance function using attribute weighting, improving by choosing the right value of the neighborhood size i.e, the value of k and improving by class probability estimation. Experiments are done on 36 UCI [76] data sets and WEKA, the data mining tool. The performance of KNN is improved using Genetic Algorithms GA [77], where GA is used to identify the k nearest neighbors straightaway, and hence this avoids calculating the similarity between the test and all training samples. The traditional KNN uses majority voting to decide the class of the test data based on the class of the k nearest neighbors. Instead a weighting scheme is proposed by Parvin, Alizadeh and Minaei-Bidgoli [78] to assign a weight to each training instance prior to building the classification model. These weights are used in the final prediction instead of simple majority voting. Frequent itemsets generated by association rule mining is used to calculate feature weights [79] and the traditional distance formula is modified using these weights to find the k nearest neighbors. However in all the modified versions of KNN mentioned above, experiments are done only on the data in traditional format.

2.2 Survey of Medical Image Classification Methods

In clinical history, the digital revolution has provided relatively inexpensive ways to collect and record huge amounts of patient data in the form of medical images. It is always difficult for a physician to analyze such huge volumes of data for medical decision making. Medical Image classification MIC models assist a physician to automatically classify a given medical image as one of a pre-defined category. Different methods have been used to classify medical images namely wavelets [80],[81], fractal theory [82], statistical methods [83] etc. Features are extracted using image processing techniques in these approaches. Other methods found in literature are using the fuzzy set theory [84], using Markov models [85], etc. Recently data Mining techniques have been successfully applied for tumour detection in digital mammography [86]. Neural networks and association rule mining techniques are investigated and a classification accuracy of 70% was achieved. Many factors affect the success of machine learning on medical data sets, one being the quality of the data. Feature selection is a technique for identifying and removing much of the irrelevant and redundant data from the training data and a survey of this was discussed in Section 2.1. A hybrid feature selection algorithm (CHI-WSS) using NB classifier is proposed [87], to improve the classification accuracy of 17 natural medical data sets. Discretization has also proved to improve the performance of many classifiers and particularly Naïve Bayes models. A new discretization algorithm called effective Bayesian Discretization has been proposed to classify bio medical data sets [88]. The performance of SVM, NB and Random Forest RF is found to improve with discrete domain. When it comes to clinical decision making, the NB classifiers are the more commonly used. The computational complexity of NB is linear with respect to the training data and hence it is better than the exponential complexity of the non-NB approaches [89]. An association rule based method is proposed [90] to enhance the diagnosis of medical images namely mammograms. The method assigns multiple keywords per image to suggest a diagnosis with high values of accuracy. A different feature extraction technique is used to represent X-ray images into 2 groups [91]: low-level image representation using GLCM, Canny edge operator, local binary pattern (LBP), pixel value and local patch-based information representation such as Bag of Words BoW. Using intelligent techniques such as neural networks, fuzzy logic and hybrid systems for classifying and pre-processing MRI medical image data to identify tumor in human brain is discussed by Hota, Shukla and Kiran [92]. A hybrid approach using GA and Particle Swarm Optimization PSO is commonly used for feature extraction and feature selection for MIC [93]. Artificial Intelligence AI techniques such as SVM neural network and Fuzzy C-means are also used for MIC. A hybrid approach using association rule mining and decision tree method is used to classify CT scan images [94]. The frequent pattern tree FP-tree algorithm is used to generate frequent patterns from the CT scan images from which association rules are mined. These rules are combined with the rules generated by the decision tree algorithm to classify the medical images. Experimental results show that this hybrid approach has better accuracy and sensitivity than using the two methods individually. A new SVM based approach for two-class medical image classification is proposed by Le, Tran, Ma and Sharma [95]. The approach identifies an optimal hypersphere such that the interior margin between the surface of this sphere and the normal data and the exterior margin between the surface of this sphere and abnormal data is as large as possible. Features of the bacterial image are extracted and SVM is used to classify them [96]. Feature selection using rough set theory and ant colony optimization for medical image classification is proposed Gnanasekar et al [97]. An artificial neural network based classification model is used to detect diabetic retinopathy images by Nayak, Bhat, Acharya. Lim and Kagathi [98]. The images are preprocessed using morphological techniques and texture analysis methods to extract features such as hard exudates, area of the blood vessels and contrast. These features are then used to train the classification model.

2.3 Summary

This chapter has started with a detailed survey of various approaches used for web page classification WPC as in research literature. A review of various methods for classifying medical images is also explored. The related research issues in these areas are also highlighted. In summary, this chapter has investigated the related knowledge of the area of research and sets up the research problem. This forms the basis of the present learning system. A discussion of the present framework for web page and medical image classification is in the next chapter.

CHAPTER 3

ALGORITHMS FOR IMPROVING SUBJECT BASED CLASSIFICATION OF WEB PAGE AND MEDICAL IMAGE DATA

The present framework for classifying web page and medical image data includes different phases namely feature extraction, feature selection, feature discretization and classification. Section 3.1 includes a description of the algorithms used in this thesis for improving classification of web pages. Section 3.2 includes a description of the algorithms used for improving classification of medical images. Section 3.3 includes a description of the various metrics and methods for evaluating the web page classification and medical image classification models.

3.1 Present Framework for Web Page Classification WPC

In real time data sets which are of high dimensions, the time taken to model the classifier will increase, if features are not properly selected. After extracting the initial set of features, feature selection and data tuning is done on the training web pages prior to classifier induction. This helps to minimize the induction time of the classifier, resolves the problem of sparse data in high dimensions and improves the predictive accuracy of the classifiers. The various methods to improve the performance of the web page classification models are illustrated in Fig 3.1 and a detailed description of them is in the following sections.

3.1.1 Feature Extraction and Web page Representation

Feature extraction plays a significant role in any classifier induction process. The quality of the mining result greatly depends on the initial set of features extracted from the data set for which the mining model is built. The various steps that are implemented in this thesis to extract features from the web page collections are listed below.

a. Convert each web page in the collection of web pages to a text file.

a.1 **Remove HTML tags, punctuations, digits, hyphens and stop words.**

Words that are too frequent in the web page collection are not good discriminators of the particular category of the web page. A word that occurs many times in the web page collection has less significance in identifying the category of the web page. Such words are called stop words and are generally removed in the pre-processing phase. Stop words include articles, prepositions and conjunctions.

a.2 **Words are reduced to their root, using stemming algorithm**.

For grammatical reasons words may be present in different forms in a text. The syntactic variations of a word prevent a perfect match between a query word and a respective word in the web page. This problem is overcome by substituting the words by their corresponding stems and the process is known as stemming. A stem is a part of a word that remains after removing its affixes which could be either prefixes or suffixes. For example the word comput is the stem for the words computer, computers, computing and so on. Stemming helps in reducing the number of dimensions used in representing a web page and hence reduces the problem of curse of dimensionality. Four commonly used stemming approaches are: affix removal, table lookup, successor variety and n-grams. The affix removal stemming is more simple, intuitive and can be also be implemented easily. As most variations of a word are created by introducing suffixes to it, in this thesis a suffix stripping algorithm proposed by Porter [99] is used to reduce words in the web page collections to their stem. This algorithm does suffix stripping using a suffix list. It applies a series of grammar rules as stated below to the suffixes of the words that appear in the text.

$$s \rightarrow \emptyset$$

$$sses \rightarrow ss$$

Hence, by applying these two rules in sequence to the word stresses yields its stem stress.

The web pages are represented using the popular vector space model in this thesis. This has been successfully used in the information retrieval field as it enables partial matching between a query word and a respective word in the text. This is unlike the boolean representation which permits only an exact match. Terms/features are assigned positive non-binary weights in this representation. The term weights are calculated using the popular *tf-idf* weighting scheme. The term frequency *tf* and inverse document frequency *idf* of each new feature in the web pages is calculated as in Equations 3.1 and 3.2 These features will be the best representative features of a web page category, $F_{intial..}$ Each web page is represented as a vector with the weight of a feature f in a *web page*$_i$ calculated as

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i) \tag{3.1}$$

$$tf_{ij} = f_{ij} \;/ max_i\{f_{ij}\} \tag{3.2}$$

where N is the number of web pages in the collection. tf_{ij} is the frequency of term *i* in web page$_j$. df_i is the web page collection frequency of a feature. $max_i\{f_{ij}\}$ is the frequency of the most common term in the web page.

The *tf* and *idf* statistical measures are used in information retrieval domain to identify the terms that occur rarely in a document collection. Such terms are more indicative of their document category than the frequent terms in the document. If D is the total number of

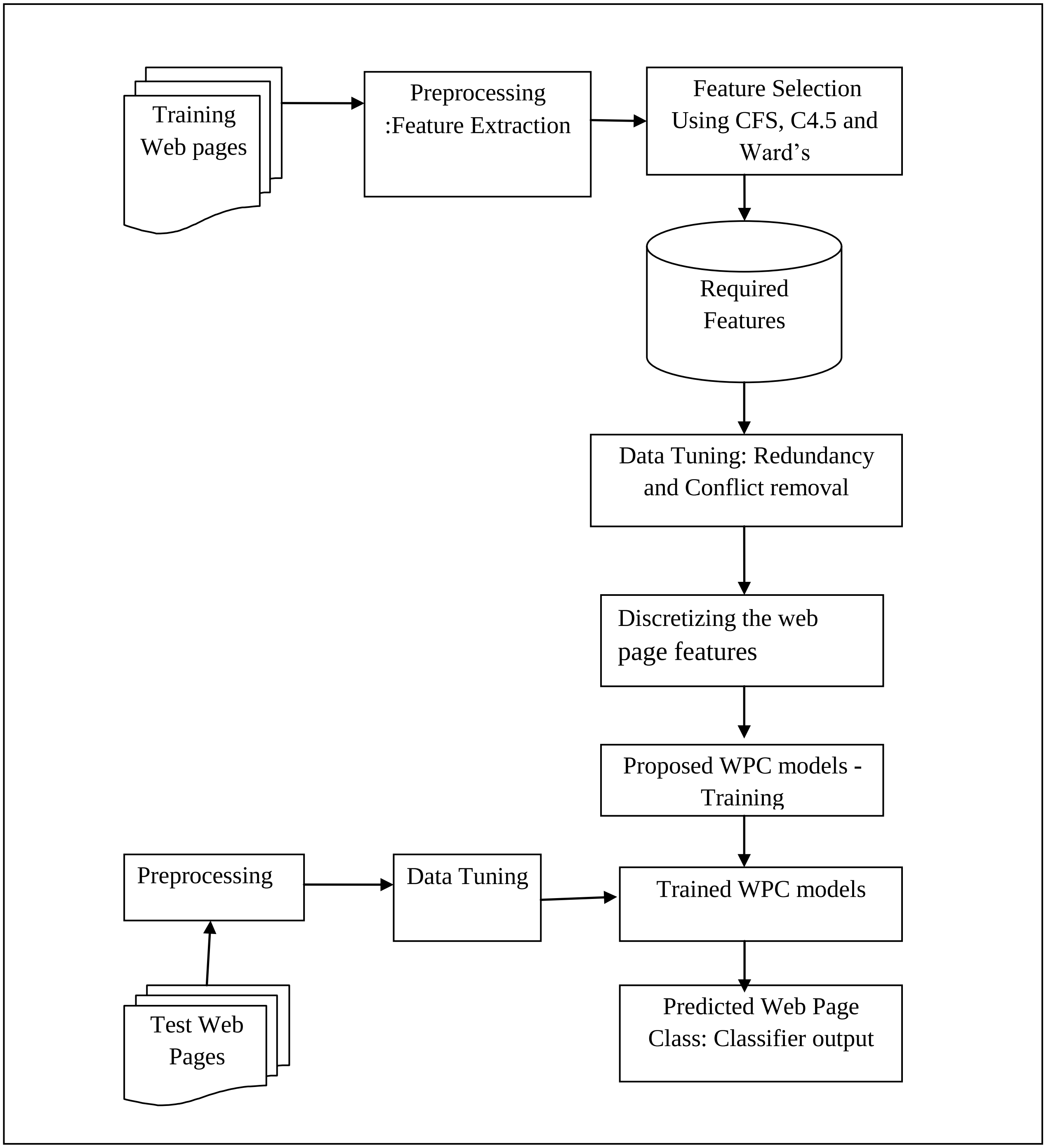

Training Web pages
Preprocessing :Feature Extraction
Feature Selection Using CFS, C4.5 and Ward's
Required Features
Data Tuning: Redundancy and Conflict removal
Discretizing the web page features
Proposed WPC models - Training
Preprocessing
Data Tuning
Trained WPC models
Test Web Pages
Predicted Web Page Class: Classifier output

Fig 3.1 The Present Framework for Web Page Classification

unique words in the collection, each web page is represented as a D-dimensional vector as illustrated in Fig 3.1 and the last column indicates the category of the web page. With N number of web pages in the collection, the web page feature matrix will be of order N x D + 1 as illustrated in Fig 1.13 in Chapter 1. The weight of the i^{th} feature in the j^{th} web page is defined as in Equation 3.3.

$$w_{ij} = \begin{cases} tf_{ij} & if\ it\ is\ present\ in\ the\ web\ page \\ 0 & otherwise \end{cases} \tag{3.3}$$

3.1.2 Feature Selection

The significance of the feature selection step in any data mining model is two fold namely: to avoid the curse of dimensionality and to improve the predictive accuracy. A big challenge faced by the mining algorithms when dealing with high dimensional data is the curse of dimensionality. As the dimensions increase, much of the data becomes sparse in nature. It becomes meaningless for mining algorithms especially those based on proximity measures, to work on such sparse data. The feature selection process also helps in removing redundant and irrelevant features which would otherwise degrade the performance of the mining model. Some of the irrelevant features can be removed immediately using domain knowledge, selecting a subset of features still needs a systematic approach. The subsets of features selected are expected to produce results that are better than or almost as good as those produced when using all features. The worth of the features selected by each model is evaluated using the predictive accuracy of various classification models and is discussed in Chapter 4 on results and discussion. Two feature selection models are designed in this thesis. The first one is a hybrid model which involves a filter based model namely a correlation based method CFS and a wrapper model using decision trees namely C4.5. This is described in Section 3.2.1.1. The second feature selection algorithm is a completely novel algorithm based on the Ward’s minimum variance measure and is discussed in Section 3.2.1.2.

3.1.2.1 Feature Selection Using CFS and C4.5

a. CFS is a correlation based feature selection method. It uses a correlation measure to characterize a feature. A feature is said to be ‘good’, if it is highly relevant to the target category/class and also is also not redundant to any of the other existing relevant features in the data set. It prefers features that are highly correlated with the class but having less correlation with any of the other existing features. The strength of the feature subset is found using a heuristic evaluation on it. This evaluation is based on how effective these features are for predicting the class with the level of inter-correlation within them. If there are ‘*n*’number of features in the data set then 2^n possible features subsets have to be explored to find the optimal subset. The size of the feature subset space is reduced by the filter models using heuristic search strategies like hill climbing, Best First Search etc. CFS begins with a null set of features and uses a best first forward search BFFS in the feature subset space. It uses the terminating criteria of getting successive non-improving subsets [44]. For WPC in this thesis, Cfs is used to select the reduced best features Fbest from Finitial., which is the intial set of features extracted. It has a quadratic computational time complexity in terms of dimensionality which is the number of features or attributes [45].

b. **C4.5** is a decision tree based learning algorithm. This has been successfully applied as a wrapper model for feature selection [47]. The standard tree induced by this algorithm has a root, one/more branches and one/more internal and external nodes. The external nodes correspond to the class labels. Each internal node is associated with an attribute test condition. For each test condition on an attribute a branch descends from this internal node. Subset of examples that satisfy this attribute test condition follow this branch. A procedure for inducing the decision tree and using it for feature selection is explained below.

1) Input to the algorithm is the set of features F_{best}; output is the decision tree.
2) All external nodes represent the class label and internal nodes an attribute.
3) The branches from an internal node represent each possible value of the attribute represented by it, for a nominal attribute. For a numeric attribute, splitting is based on an optimum value of the attribute that results in a pure partitioning of the subsets.
4) Decision to further split a subset of examples in any level of the tree is based on whether the subset is pure/impure. An impure subset is further split using another attribute and its associated test condition. If a subset is pure, the tree growing process terminates there and the subset is replaced by a leaf node whose label is the corresponding class of all examples in the subset. The purity/impurity of a subset is calculated from an information theoretic measure called Information gain and entropy. Information gain measure specifies how well a particular attribute distinguishes the training examples with respect to a target classification. The best splitting attribute at each iteration of the tree growing process is decided using this measure. The attribute that has the highest information gain after splitting a subset of examples using it, is chosen as the best splitting attribute.

   The information *gain (S, A)* of an attribute *A* with respect to a collection of training examples *S* is defined as

$$Gain(S,A) = Entropy\ (S) - \sum_{v\ \in Value(A)} \frac{|S_v|}{|S|}\ Entropy(S_v) \tag{3.4}$$

where *Values(A)* is the set of all possible values of the attribute *A*, and S_v is the subset of training set *S*, which has those training examples where the value of the attribute A is v (i.e $S_v = \{\{s\ \in S|A(s) = v\}\}$. The first term in Equation 3.4 for calculating *Gain(S,A)* is called the entropy. Entropy of a set S which is *Entropy(S)* represents the entropy of the original training set *S*. The second term is the entropy obtained after partitioning *S* using the feature *A*. It is got by adding the entropies of each subset S_v, weighted by the number of samples in *S* that belong to S_v , i.e $(|S_v|||S|)$. The reduction in entropy after splitting the set S using attribute A is given by $Gain(S,A)$. Entropy is a measure of the homogeneity of a given population and it is defined by

$$Entropy(S) = \sum_{i=1}^{c} - P_i\ log_2 P_i \tag{3.5}$$

where c is the total number of classes and P_i is the probability of the set *S* belonging to class i.

c. **Using Decision Trees for selecting the best features for WPC**:

The final set of features, F*final*, with more information gain for WPC is selected using C4.5 [100]. It is clear that the topmost node in the resulting tree is the best feature for classification. The significance of the features descends as we go down the tree. The features that are more significant in identifying the class will be present in the pruned tree.. In this way, the decision tree classification algorithms also help in dimensionality reduction. This reduces the need of domain knowledge to identify the good features for pattern classification problems.

Computational Complexity:

The computational complexity of the tree induction algorithm on a given training set D is $O(m \times |D| \times log(|D|))$ where m is the number of attributes that describe the tuples in D and $|D|$ is the number of training tuples in D [17].

Hence the computational time complexity of the proposed hybrid model of feature selection using CFS and DT is $O(m^2)$ where m is the number of attributes/features. The number of features is further reduced using a novel feature selection method using Ward's minimum variance measure, and is discussed in the next section.

3.1.2.2 Feature Selection Framework using Ward's Minimum Variance Measure

The present method [101] has two steps namely 1) identify clusters of redundant features and 2) eliminate redundant features.

1. Identify clusters of redundant features

Ward's Algorithm [103] is a commonly used algorithm for forming hierarchical groups also called clusters. These clusters are said to be mutually exclusive. Like other clustering algorithms the Ward's method is not based on the distances between clusters. Instead, it identifies clusters with maximum within-clusters homogeneity. The within-group sum of squares is used as the measure of homogeneity. That is, the goal of this method is to reduce the total within-group or within-cluster sum of squares. The clusters formed at each step will have the fewest within-cluster sums of squares. The within-cluster sum of squares that is minimized is also called the error sum of squares (ESS) or variance, $\boldsymbol{E}$. The Ward's method is said to be the very efficient among all clustering methods [103]. Although it tends to create clusters of small size, the intra-cluster similarity will be more. Motivated by this key property of this method, this thesis proposes to use the same for identifying redundant features in a web page.

Algorithm Clusters-of-Redundant-Features

Input : Feature matrix ***F*,** of order ***n x m***, where ***n*** is the total number of instances and ***m*** is the total number of features.

Output : Clusters with redundant features $\boldsymbol{C_{redundant}}$, where $C_{redundant} = \{C_1, C_2, C_3, \dots\dots\ C_{n_k}\}$. Each C_i in $C_{redundant}$ is a cluster of redundant features, p_i is the size of cluster $C_i$and n_k is the number of clusters.

Method

1. Initially each feature itself is in an individual cluster, with variance, $\boldsymbol{E} = 0$.
2. Repeat the following steps for each adjacent pairs of rows $\boldsymbol{F_i F_{i+1}}$, where $i = 0$ *to* $n-1$.

   (Note : Clusters formed in an earlier stage are never unmerged)

   2.1 Find all possible mergers of features in the two rows.

   2.2 Calculate the mean and variance, ***E*** of each possible merger.

   2.3 Choose the merger with minimum variance, *E,* where *E* of a possible merger is defined by

$$\boldsymbol{E} = \sum_{i=1}^{n_k}\sum_{j=1}^{p_i}\left[F_{ij} - m_i\right]^2 \tag{3.6}$$

   where n_k = number of clusters in the current merger

   m_i = mean of cluster *i* in the current merger

   F_{ij} = value of j^{th} feature in cluster *i*

   p_i = number of features in cluster *i*

   2.4 Calculate the vote of each merger with minimum ***E*,** identified for these two rows.

3. By majority voting, choose the merger with highest vote. Each merger will have clusters with different number of features. Clusters with more than one feature are said to have redundant feature groups.

2. Eliminate redundant features

```
Algorithm Eliminate-Redundant-Features
Input:  Clusters with redundant features $C_{redundant} = \{C_1, C_2, C_3, \dots \dots \; C_{n_k}\}$, where $n_k$
        is the number of clusters, $p_i$ is the size of a cluster.
Output : Reduced Feature set, $R = \{C_{1best}, C_{2best}, \dots \dots \dots . C_{n_k best}.\}$
Method $for\ i = 1\ to\ n_k do$
            $p_i$ = size of the $i^{th}$ cluster $C_i$
            If $p_i > 1$ then
            begin
                  $for\ j = 1\ to\ p_i\ do$
                        1) Rank each feature $F_j$, using its information gain in predicting the
                           class label.
                        2) Select the feature $F_j$ with the highest rank.
                    $end\ for$
            $end$
               $end\ for$
```

Computational Complexity to identify clusters of redundant features:

The computational time complexity analysis of the first phase of the present feature selection framework namely identifying clusters of redundant features is discussed below:

1. The entire process is repeated for every adjacent pair of rows as stated in step 2. Hence it is executed n-1 times if n is the total number of instances.
2. The time complexity of all agglomerative hierarchical clustering algorithms is $O(n^2 \log n)$ [8]. For Ward's method the proximity between two clusters is defined as the increase in the squared error when two clusters are merged. Although this feature of Ward's method shows it distinct from the other hierarchical methods, Ward's is similar to the group average method, when the proximity between two points is defined as the square of distance between them [8]. In this thesis, the Ward's method is used to identify clusters of redundant features. Hence the time taken to identify the clustering of features with minimum variance for each adjacent pair of rows is $O(m^2 \log m)$, where m is the total number of features.
3. As discussed in step 2.4, using a hash table to keep track of the votes of each distinct clustering formed in each run, takes constant time to increment the vote of a clustering after each run. The size of the hash table will be 2^m where m is the number of features. So, this signifies a smaller value of m would result in a smaller hash table. Hence the hybrid model of feature selection executed before this phase in the proposed model has helped to achieve this. The maximum value of m used in the experimental analysis is 14 which is illustrated in Table 4.7 of Chapter 4.
4. A heap is also constructed simultaneously having all possible clustering and their corresponding votes. The time to insert a distinct clustering into the heap is proportional to the height of the heap namely $O(\log 2^m)$ which is roughly $O(m)$. Finding the clustering with the highest vote can be done in constant time i.e., $O(1)$.

5. Repeating this for n-1 times, gives the total computational complexity as $n - 1\ x\ [O(m^2 \log m) + O(m) + O(1)]$. This means the computational cost of identifying clusters of redundant features grows atmost $n - 1\ x\ (m^2 \log m)$ with n as the number of instances and m as the number of features.

Computational Complexity to eliminate redundant features:

The computational time complexity analysis of the second phase of the present feature selection framework namely eliminating redundant features is discussed below:

1. In the best case, the clustering result has a single cluster with two features and the remaining (*m*-*2)* clusters with one feature each. Hence the number of clusters n_k, is (*m – 1*) in the best case. In the worst case n_k is 2, where there is one cluster having (*m*-1) features and the second one with a single feature.
2. The inner loop is executed only for clusters with more than one feature. The number of iterations of the inner loop depends on the value of p_i, the size of each cluster C_i in $C_{redundant}$. Hence p_i is 2 in the best case where the information gain has to be computed only for the two-member cluster. In the worst case p_i is (*m*-1) when the information gain has to be computed for all (*m*-1) features in that cluster.
3. Computing the information gain of an attribute is based on estimating the conditional probabilities of a class given a feature and the entropy calculations [110]. The time complexity for probability estimation and entropy computations are *O(n)* and *O(m c)* respectively, where *n* is the number of instances, *m* is the number of features and *c* is the number of classes.
4. Using a heap to store the information gain of each feature involves $O(\log m)$ to insert each

feature into it. Selecting the feature with the highest gain can be done using a heap in constant time.

5. Hence in the best case the inner loop is executed twice and so the time complexity is $2\,x\;[O(n) + \;O(m\,c) + \;O(\log m) + O(1)]$. In the worst case the time complexity of the inner loop is $(m-1)x\;[O(n) + \;O(m\,c) + O(\log m) + \;O(1)]$.
6. Therefore the total time complexity is influenced by p_i. In the best case m = 2 and the time complexity is $2\,x\;[O(n) + \;O(2\,c) + \;O(\log 2) + O(1)]$ which is $O(n)$ and it depends on the number of instances. In the worst case it is $(m-1)x\;[O(n) + \;O\big((m-1)c\big) + O(\log(m-1) + \;O(1)]$ which is $O\big((m-1)n\big) + \;O(m^2) + \;O(\log m - 1) + O(1)$ and subsequently $O(mn)$where *m* is the number of features and *n* is the number of instances.

The total time computational complexity of the proposed framework in the best case is $(n-1)x\;O(m^2\log m) + \;O(n)$ which is $O(n\,m^2\log m)$ approximately. The total computational complexity in the worst case is $(n-1)x\;O(m^2\log m) + \;O(mn)$ and is $O(n\,m^2\log m)$ approximately.

Performance Evaluation: Feature Selection using the present methods has reduced the problem of higher value of dimensionality significantly. As there should not be any compromise in the predictive accuracy after feature selection, the worth of the features selected is evaluated using a set of supervised learning algorithms. The results discussed in Chapter 4 show a significant reduction in dimensionality with no compromise in predictive accuracy.

3.1.3 Data Tuning

This phase of the thesis identifies and eliminates conflicting and duplicate web pages as explained below. Since the performance of any machine learning algorithm will be affected by their presence, such web pages are removed as a pre-processing step before model induction.

Remove web pages with null attribute values.

1. Identify and eliminate the duplicate and conflicting web pages.
2. Using vector space model represent each web page W_i as a vector $\boldsymbol{V_{ij}}$, where j= 1 to *m*, *m* being the number of features selected.

3.1.4 Discretization

The web page data after preprocessing in the vector space model are continuous in nature depending on the weights of each feature in the web page collection. Machine learning algorithms have been applied in real-world classification tasks like WPC. Majority of these algorithms focus on learning in discrete feature space. They can be applied only to data described by discrete numerical or nominal attributes (features). In the case of continuous attributes, there is a need for a discretization algorithm that transforms continuous attributes into discrete ones, or to use a different algorithm. Also, algorithms which can handle both continuous and discrete features perform better with the discrete-valued attributes. Discrete values play an important role in data mining and knowledge discovery. Many studies have shown that induction tasks can benefit from discretization: rules with discrete values are normally shorter and easy to understand and discretization can lead to improved predictive accuracy [54]. Apart from the algorithmic requirements, discretization also helps in increasing the speed and accuracy of induction algorithms. It makes the results of the induced classifier shorter, compact and easier to understand than those generated using continuous features.

The algorithm used in this thesis for web page classification is supervised, incremental, global, static and uses a bottom up approach [104]. It automatically identifies the number of intervals every feature needs to be discretized and this varies with each feature.

Algorithm Discrete-Domain

Input: Web page Feature Vectors $\boldsymbol{WFV}$, Web page Classes, $\boldsymbol{C}$, the threshold B_{size},

I$_{\boldsymbol{min}}$ the inconsistency threshold within an interval.

Output : Discretized web page Feature Vectors, $\boldsymbol{DWFV}$.

```
1.  For each web page feature $\boldsymbol{f}$ in $\boldsymbol{WFV}$ do
        1.1   The values of $\boldsymbol{f}$ are first sorted into ascending order with their respective class labels.
        1.2   Introduce a cut point where two consecutive values in $\boldsymbol{f}$ have different class labels.  Let $\boldsymbol{C_1}$ be the set of all such cut points.
        1.3   for each bin of values $\boldsymbol{B}$ in $\boldsymbol{C_1}$ do
              1.3.1   Find the majority class of the bin, $\boldsymbol{B_{maj}}$.
              1.3.2   Two consecutive bins are merged, if their corresponding size is less than $\boldsymbol{B_{size}}$ i,e., $|\boldsymbol{B_{maj}}| < \boldsymbol{B_{size}}$
              1.3.3   The new cut points are saved in $\boldsymbol{C_2}$.
        1.4   for each bin in $\boldsymbol{C_2}$ do
              1.4.1 The inconsistency measure $B_I$ of two consecutive bins is found as

                        $\boldsymbol{B_I} = \frac{|\boldsymbol{B}| - |\boldsymbol{B_{maj}}|}{|\boldsymbol{B}|}$          (3.7)

                    where, $|\boldsymbol{B_{maj}}|$ is the number of feature values in the bin that belong to the majority class and $|\boldsymbol{B}|$ is the number of feature values in a bin.
              1.4.2. Two consecutive bins are merged if they have the same majority class and also have inconsistency measure within $\boldsymbol{I_{min}}$
        1.5   The new cut points are saved in $C_3$
        1.6   A label is assigned to each cut point in $C_3$
        1.7   The web page feature values in $f$ are replaced with the corresponding
              bin label.
    2. end for
    3. The discretized web page feature vectors with their respective class labels are stored in
       DWFV
    4. Stop.
```

Computational Complexity:

The algorithm is implemented on one feature at a time. The following is an analysis of the computational complexity for discretizing one feature using this method

1. Sorting the values of the feature to be discretized as described in step 1.1 takes $O(n \log n)$ time in the best case, where n is the size of the array having the values of the feature.
2. Establishing the initial set of cut points ***C₁*** as stated in step 1.2 runs n times and hence the complexity is $O(n)$.
3. Establishing the cut points $\boldsymbol{C_2}$ as stated in step 1.3 involves finding the majority class of the bin and merging two consecutive bins, which involves $O(n)$ time complexity.
4. Step 1.4 to find the cut points $\boldsymbol{C_3}$ involves finding the inconsistency of each bin and then merging two consecutive bins, which takes $O(n)$ time.
5. In the worst case, if each numeric value of the feature is mapped to a distinct discrete label, step 1.6 takes $O(n)$ time.

6. Replacing each numeric value of the feature by its corresponding bin label, involves $O(\log n)$ to search and replace. Hence for n values of the feature, total time taken will be $O(n \log n)$.

Therefore the total time complexity to discretize one feature is $O(n \log n)$. If the total number of features in the input are m, the total computational complexity to generate the discretized data set will be $m\ x\ O(n \log n)$.

3.1.5 The Web Page Classification Algorithms

Two novel web page classification algorithms are present in this thesis. The first one, a probabilistic web page classifier PWPC is based on the Bayes theorem and is discussed in detail in Section 3.1.5.1. The second one is based on the traditional k nearest neighbor KNN algorithm called MKNN. It uses the interestingness measures support and confidence to calculate the feature weights and is discussed in detail in Section 3.1.5.2.

3.1.5.1 The Probabilistic Web Page Classifier PWPC

The proposed web page classifier PWPC [105], uses Bayes probability theorem to find the predictive power of each attribute-value towards the class labels. An attribute-value similarity measure between the test web page and each of the training web pages is then used to predict the class of the test web page. The present classifier PWPC is suitable for multi-class web page classification and is explained below.

Algorithm PWPC

Input : The training set *D* **-** Discretized web page Feature Vectors with *n* number of training web pages, c number of web page classes, and the test web page *T*

Output : The predicted class $y'$of the test web page *T*

1. Partition the training set *D* into c disjoint subsets where c is the number of class labels.
2. Calculate the predictive power, *PP* of each attribute-value in each class partition, using Bayes theorem as stated below

   2.1 $PP\ (X|C_i) = \frac{N\ (X|C_i)}{N\ (C_i)}\ \ for\ i = 1\ to\ c$

   where $\mathrm{N}((\mathrm{X}/|\mathrm{C_i}))$ = number of web pages where the attribute – value X has the class label $\mathrm{C_i}$. $\mathrm{N}(\mathrm{C_i})$ = number of web pages that belong to the category $\mathrm{C_i}$.

   2.2 Identify the most predictive attribute value(s). If an attribute has the same PP for all its values, then it is least significant, hence discarded.
3. Calculate the attribute-value similarity measure, *AVS* for every training web page in *D* with the test web page *T*. This identifies the nearest neighbors of the test web page and to what extent they are nearest.
4. Partition the training records/web pages into descending order of their *AVS* measure. An *AVS* partition has web page(s) that have same *AVS* value.
5. For each *AVS* partition do

   5.1 Find the sum of the predictive power *PP* of the influencing attribute values in each web page in this partition.

   5.2 Find the web page(s) with highest *PP* i.e., with more influencing attribute values.

   5.3 Predict the class $\mathrm{y}'$of the test web page *T* directly by majority voting.

   5.4 If there is an equal class probability distribution then proceed with the next *AVS* partition.

Computational Complexity:

The computational complexity analysis of the PWPC is discussed below where n is the number of training web pages, m is the number of feature, c is the number of classes.

1. Step 1 of the method as stated above involves $O(n)$ time, since it needs to process each training web page. Simultaneously, a counter for each attribute value - class combination, keeps track of its corresponding number of occurrences.
2. Computing the *PP* of each attribute-value class combination as in step 2.1 therefore involves time complexity of $O(n\ x\ m\ x\ c)$.
3. Identifying the most predictive attribute values as in step 2.2 needs to process all the counters. Hence this step is bound by *max*, the largest number of intervals a feature is discretized. In the worst case if all features have *max* number of distinct labels, the time complexity of this step is $O(m\ x \max x\ c)$.
4. Calculating the *AVS* value of each training web page as stated in step 3, involves $O(n)$ time.
5. Creating *AVS* partitions, with descending values of *AVS* as stated in step 4, needs to sort the training records and hence involves $O(n \log n)$ time complexity.
6. Predicting the class of the test web page as discussed in step 5 needs to process all *AVS* partitions in the worst case and hence involve $O(n)$ time. In the best case, it is required to process only the first partition.

Therefore the total time complexity of the present classification method is $O(n \log n)$ and is influenced by n, the number of training web pages.

3.1.5.2 The modified k Nearest Neighbor Classification MKNN

The traditional KNN classifier is improved for web page classification using a feature weighting scheme using the interestingness measures min_sup and min_conf of the association rules. For a rule of the form, $ithaca = 16 \rightarrow class = student$, support of the rule gives the fraction of transactions having both the rule's antecedent and consequent. Confidence of the rule is the ratio of number of transactions having both antecedent and consequent to the number of transactions having the rule's antecedent. By this method [106] classifying a new web page involves two steps namely 1) calculate the feature weights using association rules and 2) predict the category of the test web page using the modified k nearest neighbor algorithm.

1. Calculate the feature weights

Algorithm Feature-Weighting

Input: Minimum support ***min_sup,*** minimum confidence ***min_conf,*** **t**he web page feature vectors with discrete features and ***m*** the number of features.

Output: The weight of each feature in a vector, ***weight***.

Method:

```
1. for i = 1 to m  number of features do
    1.1   Generate association rules with every value of the feature and web page category
          combination.
    1.2   Calculate the support and confidence of each rule.
    1.3   Find the maximum support max_sup and maximum confidence max_conf of the
          feature.
    1.4   if $max\_sup_i < min\_sup$ or $max\_conf_i < min\_conf$ then
```

```
                $weight_i = 0$
        $else$
                $weight_i = 1/(1 - max\_sup_i)$        (3.8)
    $end\ for$
```

Computational Complexity:

The computational complexity of the feature weighting scheme used depends on the number of possible values/discrete labels of each feature. This in turn is dependent on the discretization method used. The features in the data set used for experimental analysis is discretized using the method illustrated in Section 3.1.4. The computational complexity of calculating the weight of one feature is discussed below where ***n*** is the number of training records, ***nl*** is the number of distinct labels of a feature and ***c*** is the number of classes:

1. The size of the rule set is $c\ x\ nl$. To generate these rules every training record need to be processed once. Hence the total computational complexity is $O(n)$.
2. Calculating the support of each rule as stated in step 1.2, is done using a hash table whose size is $c\ x\ nl.$ Each entry of the hash table represents one distinct rule and also stores the number of times it appears in the training set. Each distinct rule is identified a unique key in the hash table. The key is the concatenation of the characters corresponding to the feature value and the class label. Constructing the hash table can be done in $O(n)$ time and *incrementing* the count of a rule when it occurs in the training set takes $O(1)$ time.
3. A second hash table is used to calculate the confidence of each rule. For each distinct left hand side of a rule, its corresponding entry in the hash table stores it's the number of times it appears in the training set. Hence calculating the confidence of a rule also involves $O(n)$

time.

4. Finding the max $\_sup$ and max $\_conf$ of a feature involves $O(c\ x\ nl)$ where, $c\ x\ nl$ is the number of support and confidence values a feature has, one for each distinct rule generated from it.
5. Assigning a weight to each feature as stated in step 1.4 can be done in $O(1)$ time.
6. Hence the total time to calculate the weight of one feature is $O(n) + O(n) + O(\ c\ x\ nl)$. Therefore the total time complexity is $O(\ c\ x\ nl)$ since the size of the rule set depends on the number of distinct labels *nl,* and the number of classes *c*.

Therefore the complexity to calculate the weights of all m features in the data set is $m\ x\ O(\ c\ x\ nl)$.

2. Improved KNN for classifying a new web page

Algorithm MKNN

Input: The feature weight vector **weight**, the web page feature vector with numeric features, value of k and the test web page.

Output: The predicted category of the test web page.

Method:

1. Find the k nearest neighbors to the test web page using the feature weighted distance formula as below

$$distance(X,Y) = \sqrt{\sum_{i=1}^{n} weight_i((x_i - y_i)^2)} \qquad (3.9)$$

2. A distance weighted voting on the class of the k nearest neighbors is used to predict the category of the test web page.

   2.1 Assign a weight to each of the k nearest neighbor as below

$$w'_i = \begin{cases} \frac{d(x',x_k^{NN}) - d(x',x_i^{NN})}{d(x',x_k^{NN}) - d(x',x_1^{NN})} & if \ d(x', x_k^{NN}) \neq d(x', x_1^{NN}) \\ 1 & if \ d(x', x_k^{NN}) = d(x', x_1^{NN}) \end{cases} \tag{3.10}$$

where x' is the test web page, x_k^{NN} is the k^{th} nearest neighbor to the test web page, x_1^{NN} is the first nearest neighbor to the test web page and x_i^{NN} is the i^{th} nearest neighbor to the test web page.

2.2 Predict the class y' of the test web page as

$$y' = \text{argmax}_y \sum_{(x_i^{NN}, y_i^{NN})} w'_i \text{ x } \delta\ (y = y_i^{NN}) \tag{3.11}$$

$$\text{The function } \delta\ (y = y_i^{NN}) = \begin{cases} 1, & y = y_i^{NN} \\ 0, & y \neq y_i^{NN} \end{cases} \tag{3.12}$$

where y_i^{NN} is the class label of the i^{th} nearest neighbor, y is the set of all class labels.

Computational Complexity:

If each web page is represented by ***m*** number of features, finding the distance between the test web page and every training web page using Equation 3.9, involves $n\,x\,O(m)$ computational complexity, where ***n*** is the number of training records.

1. To find the **k** nearest neighbors, each time the minimum of the distance array is found, and it is replaced by the largest element of the array. This can be repeated k times, with the size of the distance array decreasing by one each time. Finding the smallest in the distance array can be done in $O(n)$ time, where ***n*** is the size of the array. Removing the smallest element can be done in constant time. For example if k = 3, the total computational complexity of this step is $O(n) + O(n-1) + O(n-2)$. Hence this is approximately $O(n)$ in the worst case.

2. Assigning a weight to each of the k nearest neighbors as explained in step 2.1 can be done in time $k\ x\ O(1)$.
3. Finding the sum of the weights of the nearest neighbors that belong to the same class involves constant time using k number of counters one for each class. This step depends on the value of the constant k.
4. Finding the class with the largest count can be done in constant time.
5. Hence the total time complexity is $n\ x\ O(m) + O(n)$.

The total time complexity of both phases of this present model of MKNN classifier is $m\ x\ O(\ c\ x\ nl) + O(m) + O(n)$. In an ideal data set, the number of features *m* and the number of distinct labels of a feature *nl* will be relatively less than *n*, the size of the training set. Hence the computational complexity of this present MKNN classifier is bound by *n*, the size of the training set.

3.2 Present Framework for Medical Image Classification MIC

Due to the ever increasing amount of patient data in the form of medical images, it is quite challenging to clinical routine such as diagnosis, treatment and monitoring. Hence an automatic medical image classification model is needed to assist physicians in patient diagnosis. The architecture used in this thesis for medical image classification framework is illustrated in Fig 3.2.

3.2.1 Image Preprocessing

Due to the non-uniformity in the color distribution of fundus images among the different subjects, each image is preprocessed for having uniform distribution of gray levels. This non uniformity is mainly due to non-uniform illumination and variation present in the pigment color in the eye. In order to overcome this a technique called adaptive histogram equalization [7] is applied to the image before it is processed further. This technique manages this local variation in contrast by increasing the contrast in lower contrast area and decreases the contrast in high contrast area.

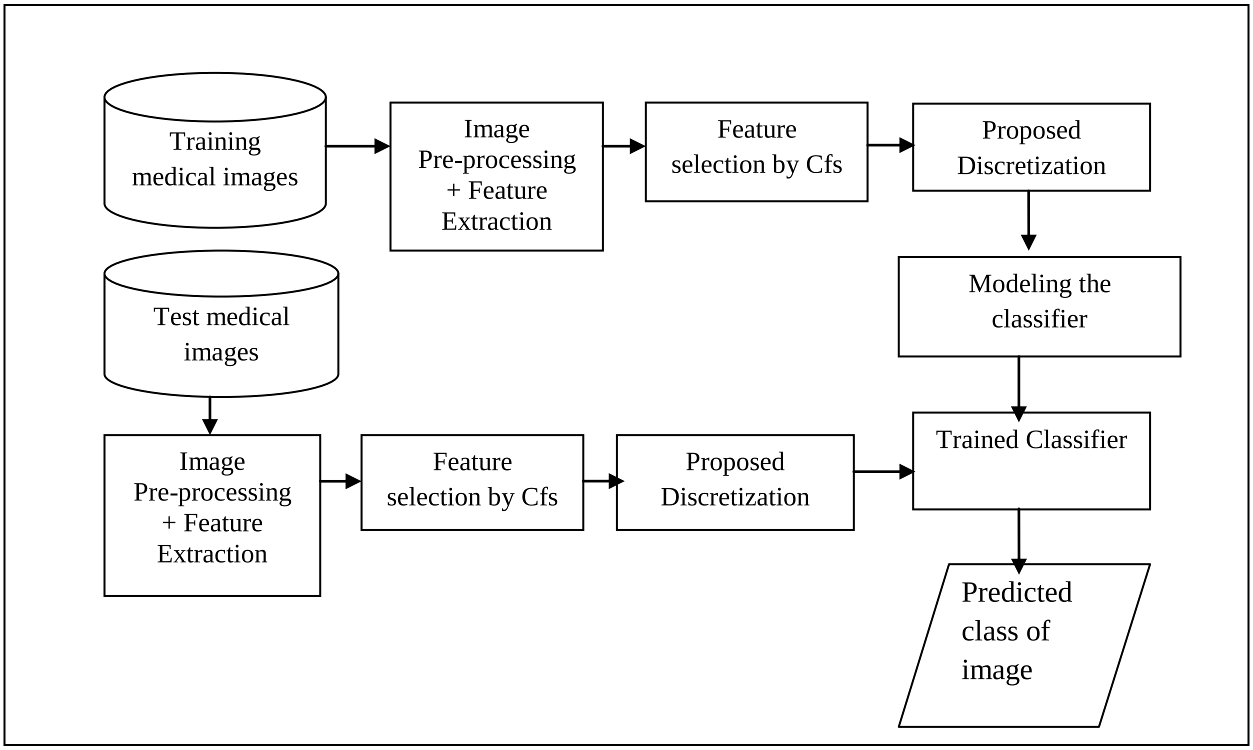


Fig 3.2.The Present Framework for Medical Image Classification

3.2.2 Feature Extraction

Features are extracted from the retinal fundus images of size 576x720 pixels. Localized statistical features are computed by dividing the image into sub-images. More localization of the image will yield accurate features. In this proposed system fundus images are divided into sub images of size 36x90 pixels, which will result in a feature vector of size 128. The four statistical features such as mean, variance, skewness and kurtosis were extracted from the images [107]. Total size of the feature vector is 512. These features are computed as follows.

a. $$Mean\ \bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{3.13}$$

where N is the number of data points.

b. $$Variance = \frac{\sum_{i=1}^{N} (x- \bar{x})^2}{N} \tag{3.14}$$

c. $Skewness = \frac{1}{N}\left(\frac{(x-\bar{x})}{\sigma}\right)^3$ (3.15)

d. $Kurtosis = \frac{1}{N}\left(\frac{(x-\bar{x})}{\sigma}\right)^4 - 3$ (3.16)

After feature extraction, each image is transformed into a feature vector and are represented as $\{f_1, f_2, f_3, \dots . f_n, Image\ category\}$, where each f_i is a continuous feature and Image category is the pre-defined category of the image.

3.2.3 Feature Selection

With no quality data, there is no quality mining results. So, in order to reduce the hypothesis space for the classifiers and to reduce the average classification error, feature selection is performed using CfssubsetEval, a correlation based method as discussed in Section 3.2.2.1 of this chapter. This method evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low inter correlation are preferred.

3.2.4 Feature Discretization

The image features are then discretised by a series of split and merge using the supervised discretization algorithm as explained in Section 3.2.4. Results discussed in Chapter 4, signify that the performance of the three most commonly used classification models in medical domain namely NB, SVM and KNN in discrete domain is significantly better than in the continuous domain. The classification models are also compared based on a metric called area under the receiver operating characteristics curve, AUC which is used in to assess predictive ability in medical domain.

3.2.5 Classification Algorithms for Medical Images

The two novel classification algorithms namely a probabilistic medical image classifier PMIC and a modified k nearest neighbor MKNN that is discussed in Section 3.2.5 are used to train classifiers with medical images as training data. They are modeled to predict both binary and multi-class medical images. The training and the testing phases of the MKNN classification model are illustrated in Fig 3.3 and Fig 3.4 respectively. The testing phase of the MKNN classification model is illustrated in Fig 3.4.

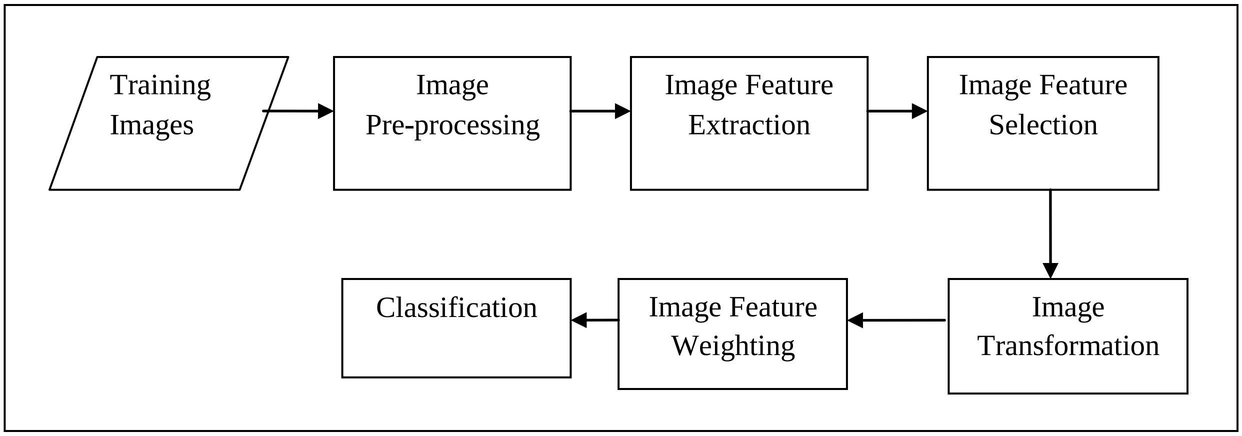


Fig 3.3. The Training Phase of the Present Framework for MIC

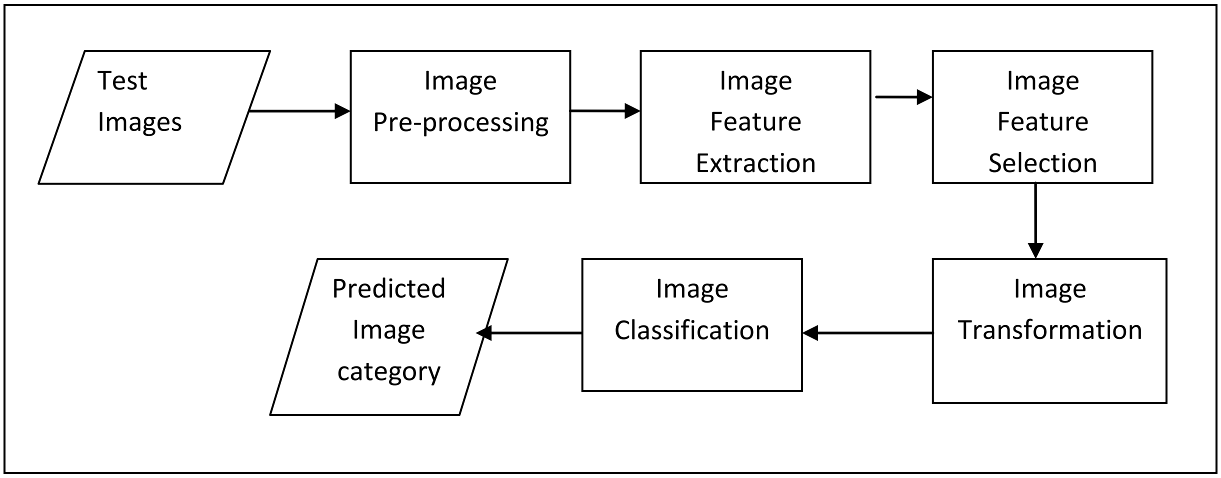


Fig 3.4. The Testing Phase of the Present Framework for MIC

The performance of these classifiers are compared with the other existing models based on predictive accuracy and AUC and is discussed in detail in Chapter 4.

3.3 Performance Evaluation Metrics and Methods

Methods for Evaluating Classifier's Performance: Some of the techniques used for evaluating the performance of a classifier are holdout method, random subsampling, k-fold cross validation, leave one out cross validation and stratified sampling [8]. The hold out method divides the original data into two disjoint subsets using a percentage split. Usually a 70 – 30 % split divides the original data into a training set with 70% of the original data and a test set having 30% of the original data. The training set is used to build the model and the test set is used to evaluate the accuracy of the induced model. Random subsampling is repeated holdout for some k times. In this technique there is no control over the number of times each record is involved in training and testing. The most popular of all methods is the k-fold cross validation. It partitions the original data into k disjoint subsets. In each of the k iterations, the model is trained on k-1 partitions and tested on the remaining one partition. The accuracy in each iteration is then consolidated. This is the most reliable estimate of a model's performance, since each object in the original data is involved both in training and testing exactly once. In this thesis, the models are induced using both cross validation and percentage split methods.

Metrics for Evaluating Classifier's Performance: The most important metric for estimating the performance of a classifier is its predictive accuracy. This is the proportion of a set of test instances that the classifier correctly classifies. The break down of a classifier's performance is given as a confusion matrix. Table 3.1 shows a confusion matrix for a two class problem. In a two class problem, one of the classes is often regarded as positive and the other as negative. The confusion matrix is a 2 x 2 matrix, showing the correctly and incorrectly classified examples of both classes.

Table 3.1 The Confusion Matrix [3]

Actual Class	Predicted Class	
	+	-
+	TP	FN
-	FP	TN

where TP is true positives i.e the number of positive instances that are correctly classified as positive. FP is false positives i.e., the number of negative instances that are incorrectly classified as positives, FN is false negatives i.e., the number of positive instances that are incorrectly classified as negative and TN is true negatives i.e, the number of negative instances that are correctly classified as negatives. If P and N represent the total number of positive and negative instances respectively, the predictive accuracy of a classifier gives the proportion of instances that are correctly classified and is calculated as defined in Equation 3.17.

$$Predictive\ accuracy\ = \frac{(TP+TN)}{(P+N)} \tag{3.17}$$

Receiver Operating Characteristics Graph ROC: In medical image decision making, the performance of the classifiers are compared using the area under the ROC curve called AUC. The ROC of a classifier is a plot between its True positive rate TPR and its false positive rate FPR [4]. This was originally used in signal processing applications. The TPR of a classifier gives the proportion of positive instances that are correctly classified as positive and is calculated as in 3.18.

$$TPR = \frac{TP}{P} \tag{3.18}$$

It is also known as hit rate or recall or sensitivity. The FPR of a classifier gives the fraction of negative instances that are erroneously classified as positive and is calculated as defined in 3.19.

$$FPR = \frac{FP}{N} \tag{3.19}$$

On a ROC graph as in Fig 3.6 FP rate is marked on the horizontal axis and TP rate on its vertical axis. Each classifier is represented as point (x, y) on the ROC space where the coordinates x and y are its FP rate and TP rate respectively. The points (0,1) , (1,0), (1,1) and (0,0) correspond to the perfect classifier, the worst possible classifier, the ultra-liberal classifier and the ultra-conservative classifier respectively. The diagonal line indicates the random guessing, whatever the probability of the positive class may be. If a classifier guesses positive and negative instances at random with equal frequency, it will classify positive instances correctly 50% of the time and negative instances incorrectly as positive 50% of the time. So the TP and FP rate of this classifier will be 0.5 and it lies on the diagonal line. Classifiers whose performance is better than the random guessing will be in the upper left hand triangle and those that are worse than random guessing will be in the lower right-hand triangle. In this way the ROC graph can be used to compare the relative performance of classifiers. To calculate the Area under the ROC curve called AUC, the TPR and FPR values of the classifier for various input files are plotted on the ROC graph. The area under the curve is calculated using the trapezoidal rule as mentioned in Algorithm 2 of [107]. The AUC value of a classifier is always between 0 and 1.0. A realistic classifier has AUC above 0.5 and AUC being 1 for a perfect classifier [108]. One of the reasons for characterizing a classifier using its TPR and FPR is that they do not depend on the relative sizes of P and N. In this thesis the two classification models that are proposed for medical image classification are evaluated using their area under the ROC graph.

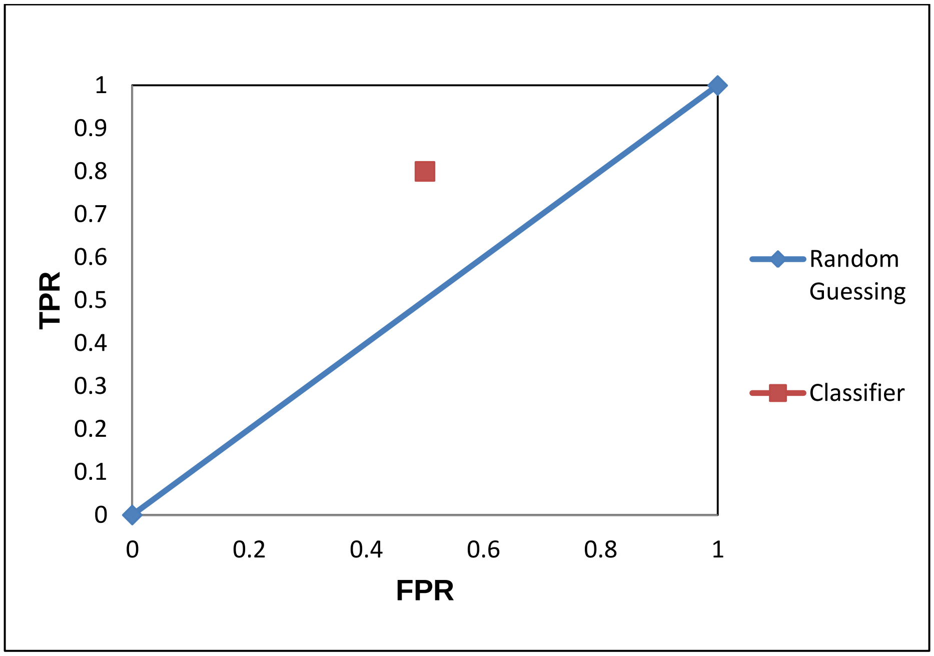


Fig 3.5 The ROC Graph [4]

.

3.4 Summary

This chapter has a detailed discussion of the algorithms used for improving subject based classification of web pages and medical images. The framework involves a series of phases namely feature extraction, feature selection, feature discretization and classification. Two feature selection methods one being a hybrid model and the other using Ward's minimum variance measure is discussed in detail in Section 3.1.2. A supervised discretization algorithm to transform the features from numeric domain to discrete domain is discussed in Section 3.1.4. Two new classification models namely 1) PWPC / PMIC based on Bayes theorem and 2) MKNN based on the traditional KNN and association rule mining is discussed in Section 3.1.5.

All the algorithms presented are implemented with both web page data and medical image data. Section 3.3 is a description of the metrics used in this thesis for evaluating the performance of the proposed classification models. A detailed illustration and discussion of the results of implementing the present framework for web page and medical image classification is in Chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

The Web page classification and the medical image classification model are induced using various phases of feature extraction, feature selection, feature discretization and classification as described in Chapter 3. The algorithms designed for these phases are implemented on a collection of web pages and medical images. The results and the detailed discussion after each of these phases for classifying both data sets are discussed in this chapter. Section 4.1 gives the details of the experimental set up for implementing the proposed algorithms. Section 4.1.1 includes the data set description used in the experimental analysis and Section 4.1.2 is a brief description of the implementation of the present algorithms. Section 4.2 is a detailed discussion of the results obtained for both binary and multi-class web page classification. Section 4.3 is a detailed discussion of the results obtained for both binary and multi-class medical image classification. Section 4.4 gives the details of the Area under the ROC graph for medical image classification. Section 4.5 highlights the summary of this chapter.

4.1 Experimental Setup

The following Section 4.1.1 is a detailed description of the web page and medical image data sets used in the experimental analysis. A sample data of each category of web page and medical image is also included in it. Section 4.1.2 gives an overview of the implementation of the algorithms presented in Chapter 4.

4.1.1 Data Set Description

The web pages used in this study are collected from the bench marking data set namely WebKB [111]. This data set contains WWW –pages collected from computer science departments of various universities by the world wide knowledge base project of the CMU text learning group. It has 8282 pages which are manually classified into the following categories namely student, faculty, staff, department, course, project and others. For each class the data set contains pages from the four universities namely Cornell (867), Texas (827), Washington (1205), Wisconsin (1263) and 4,120 miscellaneous pages collected from other universities. A directory structure is maintained for these files with one directory per category. Each of these seven directories contains 5 subdirectories, one for each of the 4 universities and one for the miscellaneous pages. The web pages are contained in these directories. The file name of each page corresponds to its URL, where '/' was replaced with '^'. Some of the pages do not contain useful information. For example, about 80 pages only contain information for redirecting the browser to a different location. These are not evenly distributed over the different classes. In this thesis binary class web page classifier is modeled using course and student category of web pages as training set. The multi class web page classifier is modeled using four categories of web pages namely course, student, faculty and project. A sample of each category of web page from the benchmarking data set WebKB [111] can be seen in the Appendix A.

The medical images used in this study were provided by Kasturba Medical College Hospital, India and are used in the thesis work [112]. These images are diabetic retinopathy images captured using retinal fundus camera. The inbuilt imaging software stores the images in JPEG format with a resolution of 576 x 720. However the feature extraction method used in the present work will work for images of any resolution. The repository includes 145 patient data. It is a collection of normal retinal images, moderately infected retinal fundus images and severely infected retinal fundus images. They give the details of the inner lining of the eye which includes the sensory retina, the retinal pigment epithelium, Bruch's membrane and the choroid.

The images in this repository were previously labeled into one of three categories using the domain knowledge and expertise. The various categories of these images are normal (61 images), moderate (52 images) and severe (32 images). In this thesis binary class medical image classifier is modeled using normal and severe category of images as training set. The multi class medical image classifier is modeled using three categories of images namely normal, severe and moderate.

4.1.2 Implementation Overview

The algorithms for classifying web page and medical images were discussed in the previous chapter namely Chapter 3. The various phases of these algorithms include feature extraction, feature selection, feature discretization and classification. A brief description of the implementation details of these algorithms is in Appendix B. It gives a list of functions and its description in each implementation. The following figures Fig 4.1, 4.2 and 4.3 shows a sample of each category of medical image.

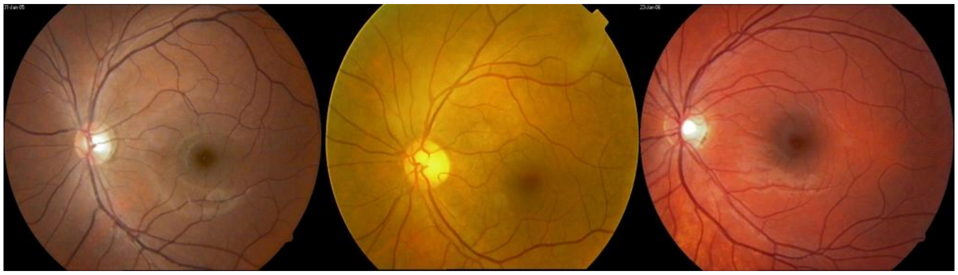


Fig 4.1 Sample Medical Images of Normal Category

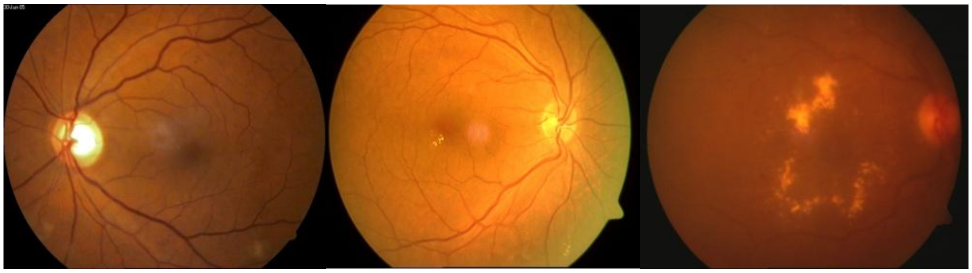


Fig 4.2 Sample Medical Images of Moderate Category

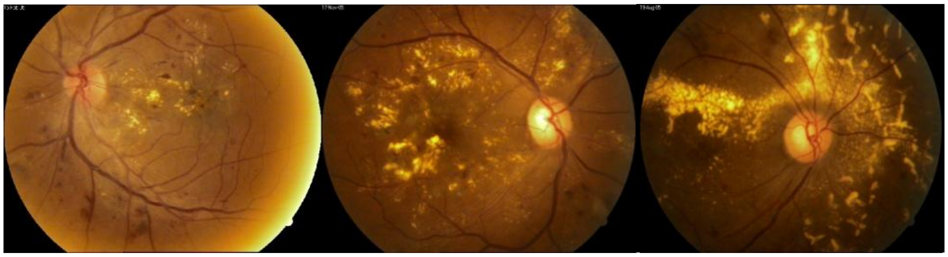


Fig 4.3 Sample Medical Images of Severe Category

used for experimental analysis. A detail of the system configuration and the software used for the experimental analysis of this research is presented in Appendix C.

4.2 Web Page Classification

The present framework for WPC involves feature extraction, feature selection, feature discretization and classification. A classification model that is induced to learn and distinguish n between two classes is a binary classifier. On the other hand, a classifier that is capable of distinguishing more than two classes is a multi-class classifier. The algorithms designed for each of these phases were experimented under both binary classification and multi-class classification. The results of these two classification framework are discussed in the following sections.

4.2.1 Binary class WPC

The binary class web page classifier is modeled to learn two categories of web pages namely course and student. The experimental results of the various phases involved during modeling are discussed below.

4.2.1.1 Feature Extraction

From the WebKB repository, the course categories of web pages are considered as positive examples and the student category of web pages as negative examples for binary web page classification. The HTML tags, stop words, punctuations, digits and hyphens are removed from the web pages in the preprocessing phase. Then, the web pages are subsequently stemmed. Following which the relevant features and instances are selected through feature selection and data tuning phases. The resultant web pages are stored as sparse instances using the SPARSE arff format supported by WEKA, a machine learning tool [11]. Sparse ARFF files are very similar to arff files, but data with value 0 are not explicitly represented. Since after preprocessing the web pages were of high dimensions they were stored in this format. Sparse ARFF files have the same header as an ARFF (i.e **@relation** and **@attribute** tags) but the data section is different in this case. Instead of explicitly representing each value in a data row as:

```
@data
 0, X, 0, Y, ‘class A’
 0, 0, W, 0, ‘class B’
```

the non-zero attributes in each row are explicitly identified and are represented using an attribute number and their value stated as:

```
@data
 {10 X, 30 Y, ‘class A’}
 {20 W, 40 Z, ‘class B’}
```

Each instance is surrounded by curly braces. The format for each entry is: <index> <space> <value> where index is the attribute index which starts from 0. The omitted values in a sparse instance are **0**, and are not ‘missing’ values.

Various web page collections with different size of positive and negative examples are used in the experiments. The initial feature set $\boldsymbol{F}_{initial}$ of all the web page collections after preprocessing are as listed in Table 4.1. The web pages are transformed to feature vectors after the feature extraction phase. In each of the input size, the first number indicates the number of positive examples and the second one indicates the number of negative examples. For example an input size of 70-30 indicates that 70 number of course web pages and 30 number of student web pages are involved in the experiments.

Table 4.1 Results after Feature Extraction for Binary Class WPC

Input Size	**No. of Instances**	**No. of Features**
70- 30	100	2,774
100 – 100	200	4,185
200 – 200	400	6,654
300 – 200	500	7,874
300 – 300	600	8,963
350 – 150	500	7,651
400 – 200	600	8,508
400 – 300	700	9,563
400 – 400	800	10,363

The second and the third columns indicate the total number of web pages and the total number of distinct features identified in the web page collection respectively. This defines the size of the vocabulary of this web page collection. The weight of a feature is its *tf-idf* value if it is present in the web page or zero otherwise. Various machine learning classifiers namely rule based (oneR), nearest neighbor based (kStar), Naïve Bayes (NB), decision tree based (J48) and two neural network based namely Multilayer perceptron (MLP) and Radial Basis Function (RBF) were induced with this high number of initial features using 10-fold cross validation. The percentage classification accuracy of these classifiers is shown in Table 4.2.

As seen from the results in Table 4.2, the performance of kstar is very poor. For input files of larger size, the neural network classifiers MLP and RBF did not produce any classification model. This is due to the huge number of dimensions and the presence of more redundant and irrelevant features in the input file after feature extraction. Thus feature selection is necessary to prevent this performance degradation.

Table 4.2 Classification Accuracy after Feature Extraction for Binary Class WPC

Input Size	**oneR**	***K*Star**	**NB**	**J48**	**MLP**	**RBF**
70- 30	89	70	87	88	-	85
100 – 100	80	50	97	94.5	99.5	98
350 – 150	85	70	88.8	91.6	-	-
200 – 200	76.75	50	92.3	91.5	-	-
400 – 200	75.5	66.7	90.8	91.2	-	-
400 – 300	80.6	57.1	91.6	92	-	-
300 – 300	80.66	50	91.5	91.8	-	-
400 – 400	81.25	-	91.88	92.125	-	-
300 – 200	79.2	60	91.4	90.8	-	-
Average	**80.88**	**59.23**	**91.36**	**91.50**	**99.5**	**91.5**

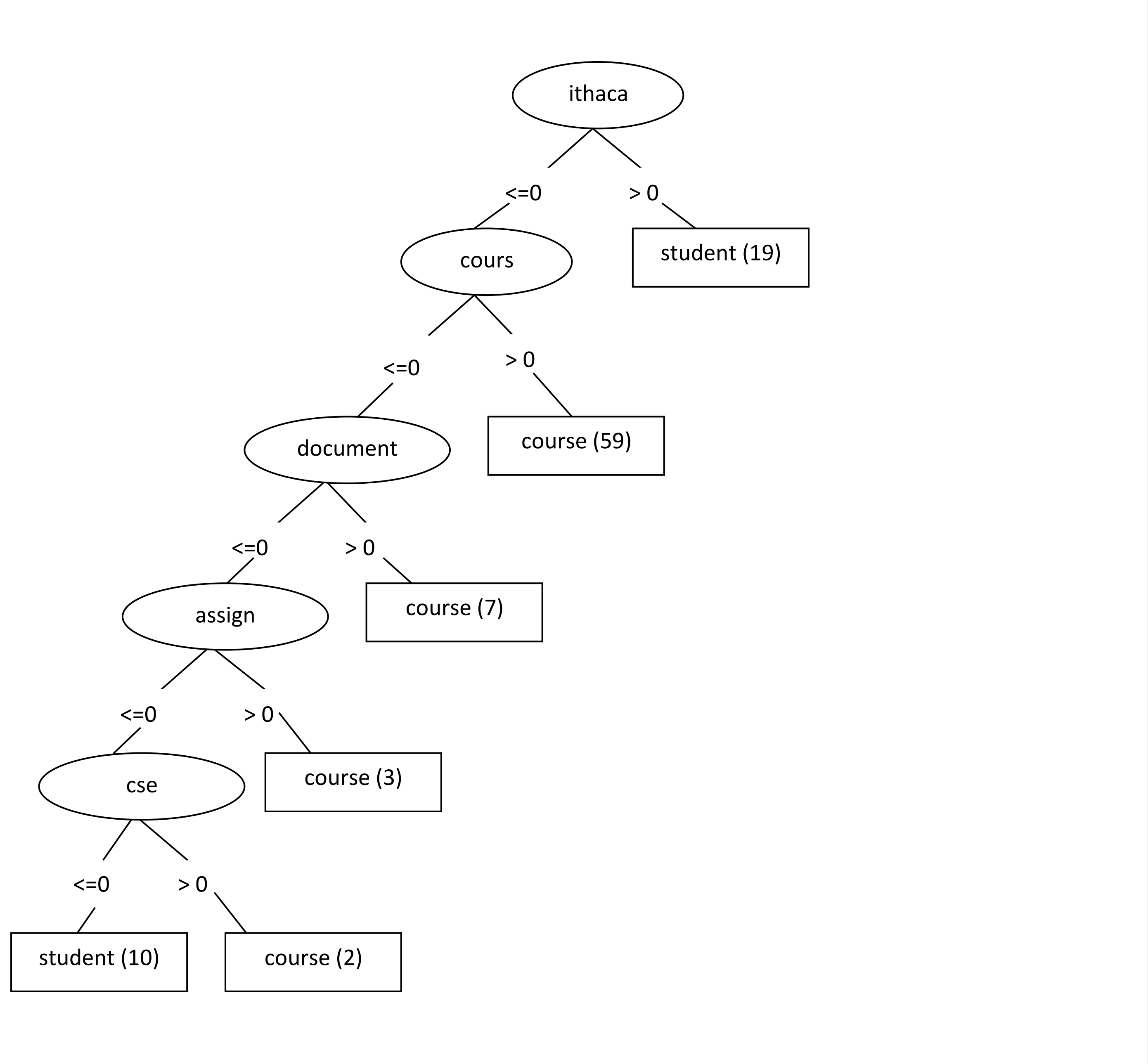


Fig 4.4 Feature Selection using J48

4.2.1.2 Feature Selection

As the initial number of features extracted is high, the classifiers occupy more memory space, need more induction time and exhibit poor classification accuracy. As the input size increases, the performance of the neural network classifiers MLP and RBF was very worse. They took a longer prediction time or did not predict certain inputs completely. In a web page only the features that are more predictive of its category have to be selected to resolve this resource underutilization and the problem of the curse of dimensionality. This is the objective of the feature selection phase of the present work.

A hybrid model of feature selection: Feature selection reduces the learning time of the machine learning classifiers and helps in improving the predictive accuracy. Two feature selection methods are presented in this thesis as discussed in Chapter 3.

The first being a hybrid model is a combination of the correlation based feature selection CFS and the decision tree algorithm J48. CFS selects features that are highly correlated with the class and having low inter correlation. The 70-30 input file after feature selection using CFS in sparse arff is shown in Appendix D. J48 is the implementation of the decision tree based algorithm called C4.5. This algorithm chooses the subset of features that present in the final pruned tree. The features that represent the internal nodes of this tree are selected using the entropy and information gain measures. The decision tree used for feature selection for the 70-30 input file is shown in Fig 4.4. The 70-30 input file after feature selection using CFS and J48 in sparse arff is shown in Appendix E. The subset of features selected by this hybrid model of feature selection for each input file is shown in Appendix F.   As seen from the tree, the significance of the feature in predicting the class decreases as we descend the tree from root. The number of features selected finally by the present method is shown in Table 4.3. It is also compared with two other existing feature selection methods namely, consistency subset and CFS itself.

Table 4.3 Number of Selected Features by each Feature Selection Algorithm for Binary Class WPC

Input Size	Cfs	Consistency Subset	Present Method
70- 30	31	6	5
100 – 100	33	10	6
350 – 150	42	22	13
200 – 200	44	18	13
400 – 200	48	15	15
400 – 300	49	16	17
300 – 300	49	13	14
400 – 400	47	15	17
300 – 200	39	13	9

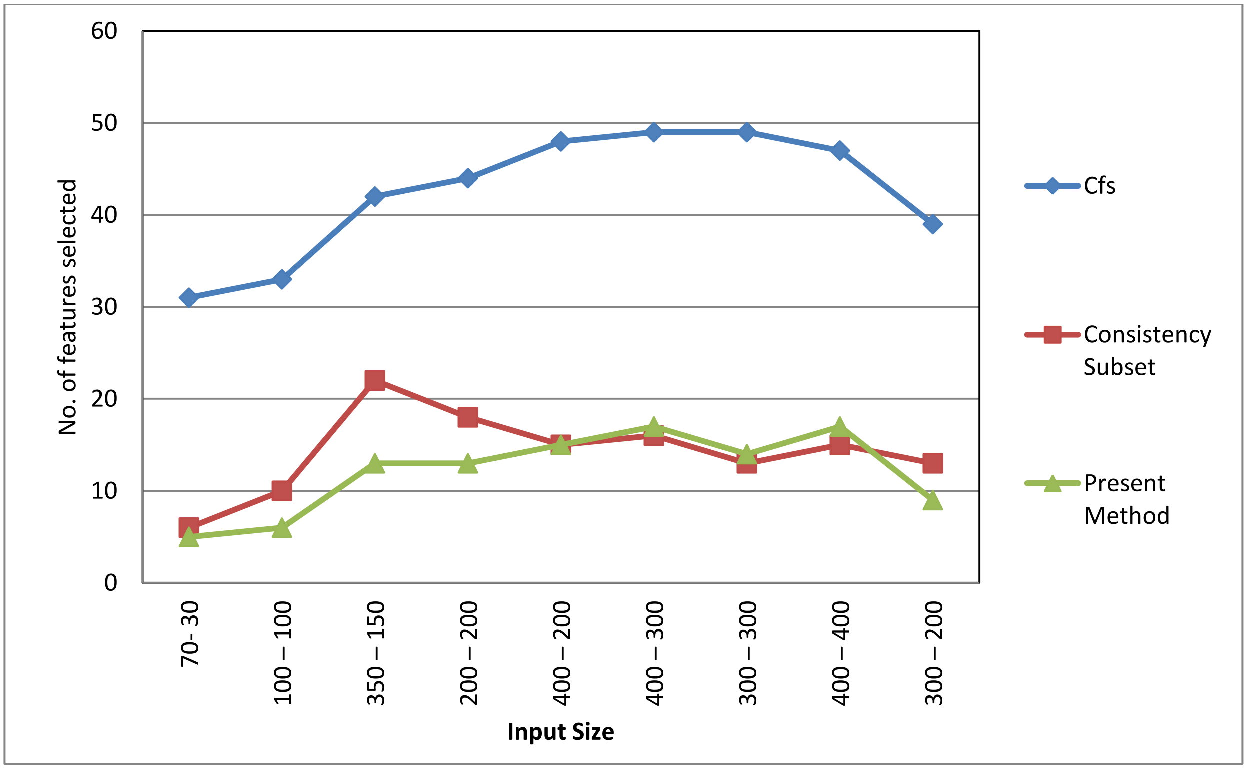


Fig 4.5 Number of Features Selected by the Present Hybrid Method for Binary Class WPC

It can be observed from Table 4.3 and Fig 4.5 that the number of features selected by the present hybrid method is significantly less than the other two methods. Table 4.4 shows the percentage reduction in the number of features from the initial set of features.

The data sets are further fine-tuned by removing noisy web pages, i.e, web pages with null values as feature weights and conflicting web pages. To justify the need of all these preprocessing and fine tuning the web pages, different classifiers are modeled using 10- fold cross validation with the reduced feature set. The percentage classification accuracy of all these classifiers are shown in Table 4.5.

Table 4.4 % Reduction in the Number of Features by the Present Hybrid Model for Binary Class WPC

Input Size	**No. of Instances**	**No. of Features**	**% Reduction in no. of features**
70- 30	56	5	99.82
100 – 100	92	6	99.86
200 – 200	291	13	99.80
300 – 200	298	9	99.89
300 – 300	414	14	99.84
350 – 150	391	13	99.83
400 – 200	422	15	99.82
400 – 300	557	17	99.82
400 – 400	585	17	99.83

There is a significant increase in the accuracy of all classifiers after all the preprocessing steps as illustrated in Table 4.5. Some of the neural network classifiers like MLP and RBF which took a very long learning time on $\boldsymbol{F}_{\boldsymbol{initial}}$, are now modeled quickly on the fine tuned data.

Table 4.5 % Classification Accuracy of Binary Class WPC with Numeric Features after Feature Selection and Data Tuning

Input	**oneR**	***J48***	***NB***	***Kstar***	***SVM***	***Boosting***	***EM***
70- 30	100.00	100.00	80.00	100.00	100.00	100.00	44.64
100 – 100	100.00	100.00	100.00	100.00	96.74	96.74	46.73
200 - 200	73.91	94.20	98.55	97.11	95.87	91.75	61.51
300 – 200	81.33	86.66	97.33	96.00	96.64	96.64	61.74
300 – 300	54.8	76.92	85.57	79.80	96.13	94.44	57.00
350 – 150	85.71	88.77	97.95	98.97	96.67	95.65	55.49
400 – 200	85.84	89.11	97.16	99.05	96.99	95.14	64.12
400 – 300	72.14	92.85	96.42	92.14	97.66	94.07	66.60
400 – 400	68.02	89.11	94.55	95.23	96.14	92.75	54.53
Average Accuracy	**80.19**	**90.84**	**94.17**	**95.36**	**96.98**	**95.24**	**56.92**

Ward’s minimum Variance based feature selection: The number of features is further reduced using a novel feature selection framework present in this thesis using the Ward’s minimum variance measure. This method involves two steps namely 1) identifying clusters of redundant features using Ward’s minimum variance measure and 2) eliminating redundant features in each cluster by selecting the feature with the highest information gain. Table 4.6 shows the detailed working of this method.

The clusters with more than one members are features with minimum variance, ***E***, found by the Ward’s method. So, features in these clusters are identified as redundant features. The complete list of features selected by this Ward’s method for each of the input files is shown in Appendix G. Modeling a classifier using all these redundant features will require maximum utilization of computer resources. Therefore, only the feature in a cluster that has more predictive information of

Table 4.6 Wards Cluster of Redundant Features Formed and the Final Selected Features

Data Set	Clusters Formed	Selected Features
70- 30	(1) (2,3,4,5)	1,5
100 - 100	(2,4,5,6) (3) (1)	1,3,5
200 – 200	(2,3,12) (11,13) (1) (4) (5) (6) (7) (8) (9) (10)	1,4,5,6,7,8,9,10, 12,13,
300 – 200	(4,5,7,9) (8) (1) (2) (3) (6)	1,2,3,4,6,8
300 – 300	(12) (3,4,5,14) (1) (2) (5) (6) (7) (8) (9) (10) (11) (12) (13)	1,2,6,7,8,9,10,11, 12,13,14
350 – 150	(2,3,4,13) (1) (5) (6) (7) (8) (9) (10) (11) (12)	4,1,5,6,7,8,9,10,11,12
400 – 200	(2,3,5,15) (1) (4) (6) (7) (8) (9) (10) (11) (12) (13) (14)	2,1,4,6,7,8,9,10,11,12,13,14
400 – 300	(1,4,5,17) (2) (3) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16)	2,3,6,7,8,9,10,11,12,13,14,15,16,17
400– 400	(1,4,5,17) (2) (3) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16)	2,3,6,7,8,9,10,11,12,13,14,15,16,17

the category of a web page is retained and the others are eliminated. Such best representative feature in each cluster of redundant features is selected by ranking them using information gain. Table 4.7 compares the performance of the present method with the features selected by some of the other feature selection algorithms namely, Principal Components PCA, Info Gain, Relief, GainRatio, oneR attribute evaluation and Symmetric uncertainty attribute evaluation.

Table 4.7 Comparison of the Wards Feature Selection with other Feature Selection Methods for Binary Class WPC

Data Set	**Original No. of Attributes**	**Ward's method**	**PCA**	**Ingo Gain**	**Relief**	**Gain Ratio**	**oneR**	**Symmetric**
70-30	5	2	5	5	5	5	5	5
100-100	6	3	6	6	6	6	6	6
200-200	13	10	12	13	13	13	13	13
300-200	9	6	8	9	9	9	9	9
300-300	14	11	13	14	14	14	14	14
350-150	13	10	12	13	13	13	13	13
400-200	15	12	14	15	15	15	15	15
400-300	17	14	16	17	17	17	17	16
400-400	17	14	16	17	17	17	17	17

Compared to all feature selection methods, it can be inferred from Table 4.7 and Fig 4.6, that the present method using Ward's achieves the highest level of dimensionality reduction. The methods Info gain, Relief, Gain Ratio, oneR and symmetric uncertainty attribute evaluation suggest that all attributes are significant for classification. Principal component analysis reduces the number of features for seven input data sets.

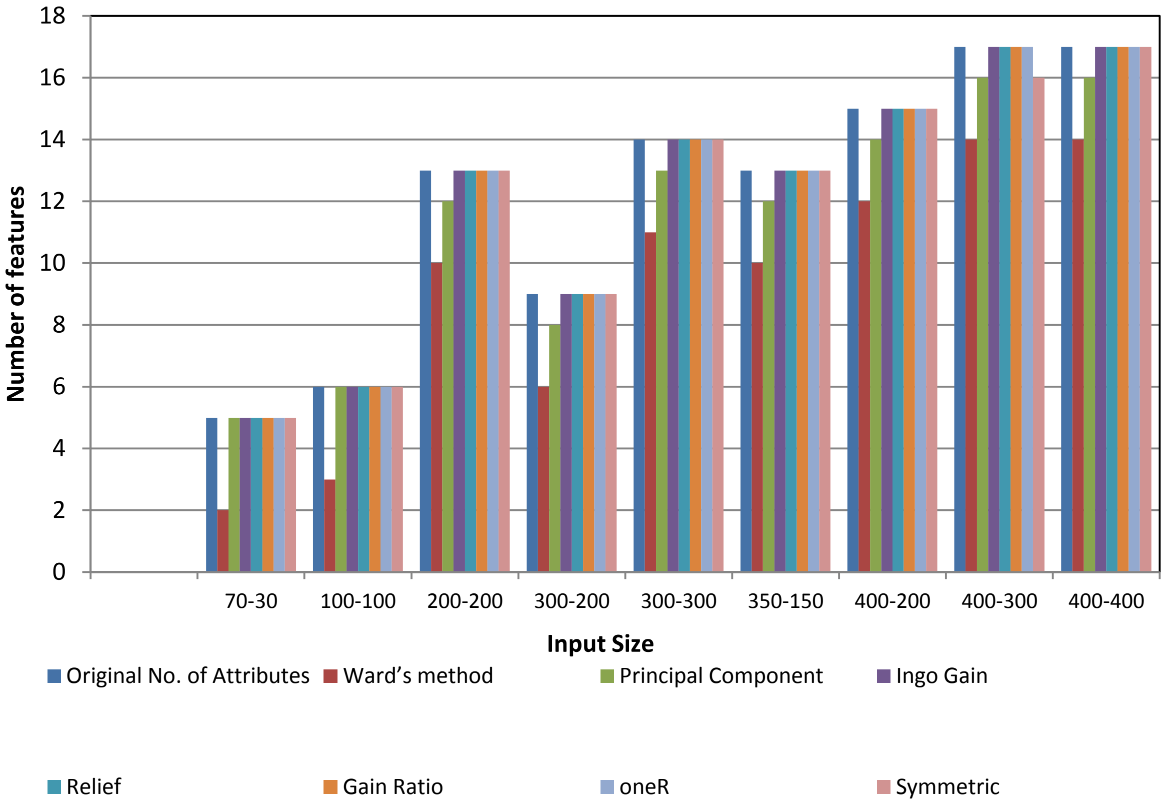


Fig 4.6 Comparison of the Wards Feature Selection with other Methods for Binary Class WPC

The worth of the features selected is tested by running various machine learning classifiers namely, k nearest neighbor (KNN), support vector machine (SVM) and ensemble boosting method. The performance of the classifiers are evaluated with three set of features namely full set, feature subset chosen by Ward's and feature subset chosen by PCA. The classification accuracy of the KNN classifiers with features selected by various methods are shown in Table 4.8.

Any feature selection method should result in less number of features and either has to improve or maintain the classification accuracy when compared with the original set of features. The results in Table 4.8 show that the classification accuracy is considerably same in all three cases. With the reduced number of features the present method is able to maintain the classification accuracy. The features selected are themselves are more predictive of the category of the web page.

Table 4.8 Classification Accuracy of KNN Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	96.42	98.21	96.42
100-100	94.56	96.74	94.56
200-200	96.21	90.03	94.15
300-200	94.29	91.95	93.95
300-300	95.65	95.17	96.13
350-150	97.18	96.68	96.93
400-200	96.71	96.06	96.99
400-300	95.33	95.51	95.51
400-400	96.06	94.70	96.06
Average	**95.82**	**95**	**95.63**

Table 4.9 Classification Accuracy of NN Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	100	100	100
100-100	92.39	96.74	92.39
200-200	94.84	89.69	93.15
300-200	93.28	91.28	93.23
300-300	96.61	95.65	95.63
350-150	95.90	95.95	95.39
400-200	96.75	96.79	96.52
400-300	96.65	96.25	96.12
400-400	95.21	95.56	95.12
Average	**95.74**	**95.33**	**95.28**

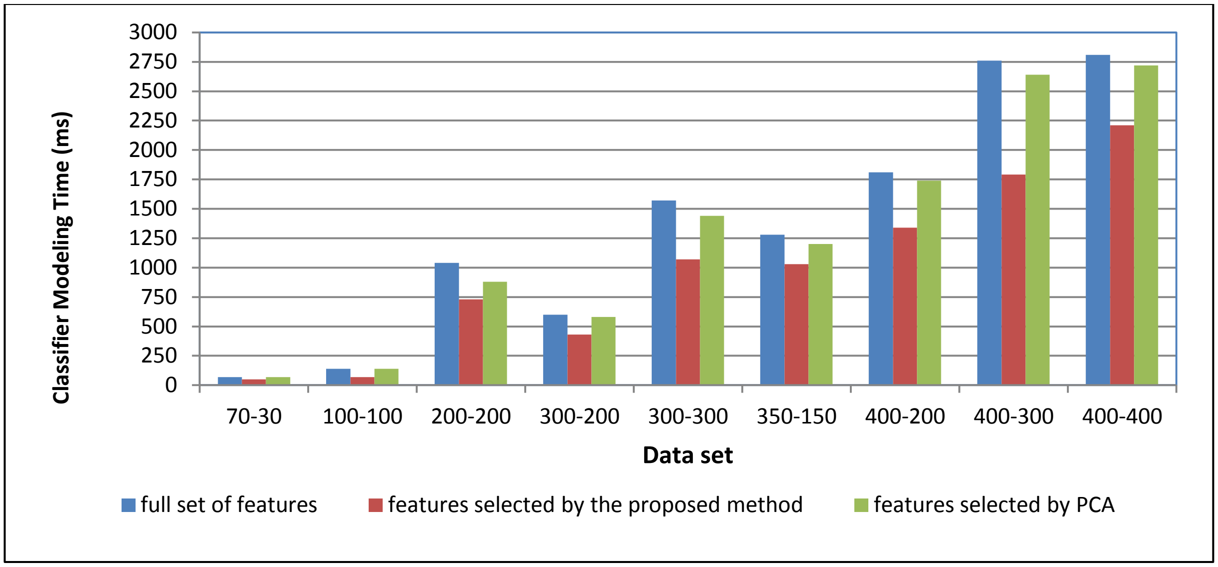


Fig 4.7 Comparison of the Modeling Time for NN Classifier with various Features for Binary Class WPC

The performance of the neural network classifier NN and the time taken to model it for each input size is illustrated in Table 4.9 and Fig 4.7. The classification accuracy of the ensemble classifier namely AdaBoost and SVM classifier with features selected by various methods are shown in

Table 4.10 Classification Accuracy of Boosting Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	100	100	100
100-100	96.74	96.74	96.74
200-200	91.75	90.03	90.72
300-200	96.64	91.61	93.62
300-300	94.44	94.44	94.68
350-150	95.65	94.37	93.86
400-200	95.13	93.75	93.75
400-300	94.07	94.97	94.43
400-400	92.80	92.14	92.47
Average	**95.24**	**94.23**	**94.47**

Table 4.10 and Table 4.11 respectively Table 4.10 and Table 4.11 show that the classification accuracy is the same in all three cases. However, the classification accuracy is maintained with the reduced number of features selected by the present method rather than using the full set of features.

Table 4.11 Classification Accuracy of SVM Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	100	100	100
100-100	96.74	96.74	96.74
200-200	95.87	92.09	94.84
300-200	96.64	91.95	93.28
300-300	96.13	96.13	96.13
350-150	96.67	96.16	96.67
400-200	96.54	95.83	96.06
400-300	97.66	95.87	97.48
400-400	96.41	96.07	95.89
Average	**96.96**	**95.65**	**96.34**

Table 4.12 Classification Accuracy of NB Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	100	100	100
100-100	93.47	88.04	93.47
200-200	94.15	91.75	93.81
300-200	96.64	90.27	93.95
300-300	95.89	95.65	95.89
350-150	96.41	94.63	95.65
400-200	93.98	93.29	93.75
400-300	95.15	95.15	94.97
400-400	94.70	94.70	94.18
Average	**95.58**	**93.71**	**95.07**

Table 4.13 Classification Accuracy of J48 Classifier on the Features Selected by each Feature Selection Method for Binary Class WPC

Data Set	**Full features**	**Ward's method**	**Principal Component Analysis**
70-30	100	100	100
100-100	96.73	96.74	96.12
200-200	92.09	89.04	91.40
300-200	95.64	91.95	93.25
300-300	94.44	91.55	91.00
350-150	93.35	94.25	93.13
400-200	94.44	93.50	92.50
400-300	92.99	93.72	93.17
400-400	93.5	93.25	93.00
Average	**94.78**	**93.78**	**93.73**

4.2.1.3 Feature Discretization

This thesis also focuses learning in discrete feature space. The numeric features after the hybrid model of feature selection are transformed into discrete space using a supervised feature discretization algorithm that is discussed in Chapter 3. This method identifies the number of intervals each feature needs to be discretized automatically. Table 4.14 shows the web page data set before and after feature discretization. For experiments the threshold B_{size} is set to 2. This ensures that each interval of the feature when discretized has atleast two values. The inconsistence threshold I_{min} is set to 0.5. The number of intervals each feature is discretized by the present method for two of the input data sets is shown in Table 4.15.

Table 4.14 Web Page Data Set before and after Discretization

Input File	File Description
(a) Before discretization	@relation webkb-CFS-DT-1-Rnd-DR-r6 @attribute assign numeric @attribute cours numeric @attribute cse numeric @attribute document numeric @attribute ithaca numeric @attribute class {course, student} @data 0.0, 0.0, 0.0, 0.0, 0.73, student 0.4, 0.2, 0.73, 0.73, 0.0, course

(b) After discretization	@relation webkb-cfs-dt-1-rnd-dr-numericclass-r6
	
	@attribute assign {3, 4, 5, 6}
	@attribute cours {7, 8, 9, 10}
	@attribute cse {11, 12, 13, 14}
	@attribute document {15, 16, 17, 18}
	@attribute ithaca {19, 20}
	@attribute class { course, student}
	
	@data
	{0 3, 1 7, 2 11, 3 15, 4 20, 5 student}
	{0 6, 1 10, 2 14, 3 18, 4 19, 5 course}

Table 4.15 The Number of Intervals Identified for each Feature by the Present Discretization Method for Binary Class WPC

Input Size	Features and the number of intervals they are discretized
70 – 30	assign – 4, cours – 4, cse – 4, document – 4, ithaca – 2
100 – 100	cse -8, document – 6, homework – 7, hour – 6, ithaca – 3, materi – 6

The performance of the classifiers in numeric and discrete space is compared based on the classification accuracy and the modeling time. Table 4.16 gives the classification accuracy of various classifiers with features discretized by the proposed method. These results are compared with the classification accuracy of numeric features after feature selection using the hybrid model as in Table 4.5. Results where accuracy in the discrete domain is greater than or same as that in numeric domain are highlighted in Table 4.16.

Table 4.16 Classification Accuracy of Binary Class WPC with Features Discretized by the Present Method

Input Size	**oneR**	***J48***	***NB***	***Kstar***	***SVM***	***Boosting***	***EM clustering***
70- 30	**100.00**	**100.00**	**100.00**	96.43	**100.00**	**100.00**	**87.50**
100 – 100	96.74	96.74	96.74	95.65	**96.74**	94.56	**89.13**
200 - 200	**75.26**	90.03	95.88	93.47	95.18	90.03	58.07
300 – 200	**87.25**	**91.28**	96.64	**96.31**	**96.97**	95.97	**73.48**
300 – 300	**75.36**	**90.10**	**97.10**	**95.41**	**96.13**	**94.68**	**60.38**
350 – 150	84.91	**89.51**	95.65	96.42	**96.93**	94.63	**72.38**
400 – 200	**88.66**	**91.20**	95.14	98.15	96.30	**95.60**	**77.54**
400 – 300	**79.17**	90.66	**96.95**	**96.05**	97.35	**94.62**	63.19
400 – 400	**76.92**	**92.99**	**96.75**	**96.41**	**96.67**	**92.75**	**60.00**
Average Accuracy	**84.92**	**92.50**	**96.76**	**96.03**	**96.92**	**94.76**	**71.30**

The accuracy of the NB classifier is significantly improved by the proposed discretization method. A comparison based on average classification accuracy as shown in Fig 4.8 shows that classifying in discrete domain exhibits good accuracy than in continuous domain.

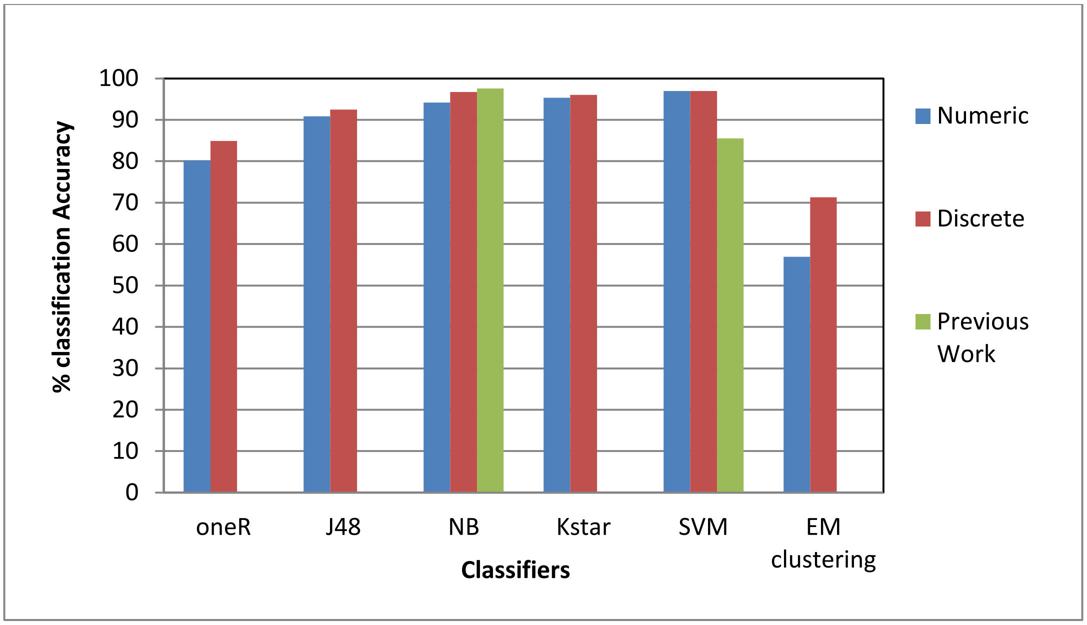


Fig 4.8 Comparison of % Classification Accuracy of Binary Class WPC in Numeric and Present Discrete Domain

Fig 4.8 also shows the comparison between the classification accuracy of NB and SVM classifiers in the numeric domain after pre-processing the web pages as proposed in [34]. However their approach uses features that appear a particular ‘N’ times as the initial set of features. The performance of the NB classifier is same as the pre-processing present in this case. However the performance of the SVM classifier with web pages pre-processed as in presented in this thesis is significantly better than their approach. The present discretization method is compared with the other commonly used discretization method namely, simple binning in Weka. However this method needs the user to specify the number of intervals a feature needs to be discretized. The percentage classification accuracy of various classifiers modeled using 10-fold cross validation is shown in Table 4.17.

A comparison of the average classification accuracy of various classifiers using three types of features namely 1) numeric features 2) features discretized by simple binning and 3) features discretized using the present method is illustrated in Fig 4.9.

Table 4.17 Classification Accuracy by Simple Binning for Binary Class WPC

Input Size	**oneR**	***J48***	***NB***	***Kstar***	***SVM***	***Boosting***	***EM clustering***
70- 30	100	100	96.43	98.21	100	100	87.5
100 – 100	96.74	96.74	95.65	96.74	96.74	96.73	89.13
200 - 200	75.26	87.28	95.53	93.81	95.87	90.37	94.50
300 – 200	87.25	95.30	96.30	95.64	95.30	95.30	96.30
300 – 300	75.60	91.54	96.13	95.41	95.65	95.16	96.85
350 – 150	85.93	89.76	96.93	95.39	96.41	94.11	97.18
400 – 200	87.03	87.96	97.22	98.37	97.22	95.60	96.06
400 – 300	79.17	92.81	97.66	95.51	97.30	94.97	96.76
400 – 400	76.92	92.13	96.58	96.75	95.55	92.99	96.23
Average	**84.88**	**92.61**	**96.49**	**96.20**	**96.67**	**95.03**	**94.50**

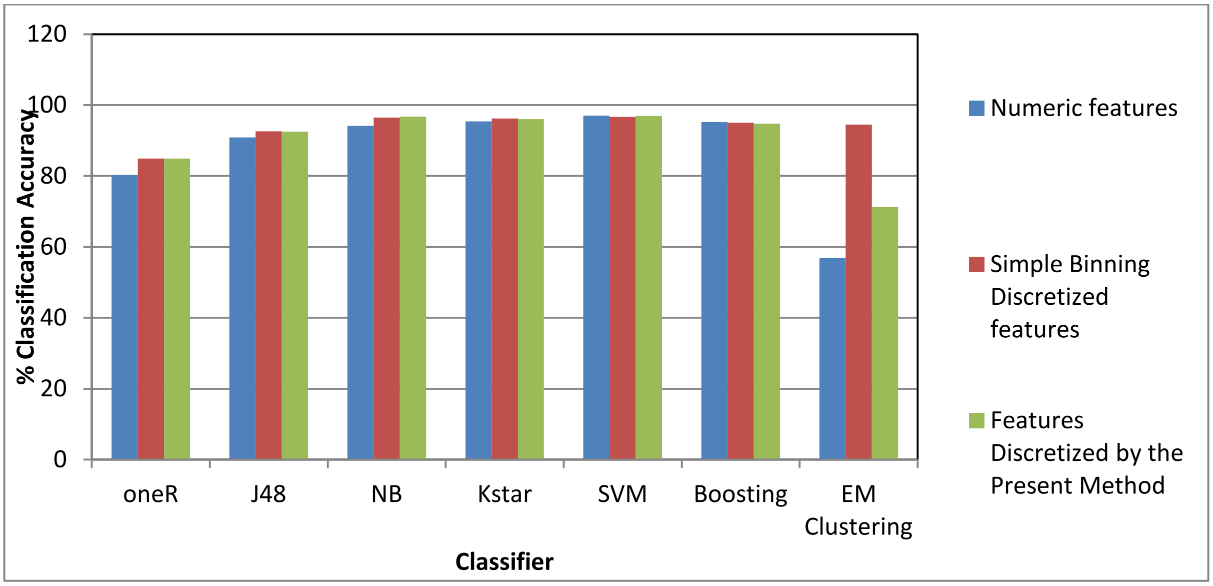


Fig 4.9 Comparison of % Classification Accuracy of Binary Class WPC with Various Features

Besides improving classification accuracy, modeling web page classifiers with discrete features has also reduced the time taken to induce these classifiers. The modeling time in msecs of various web page classifiers in numeric domain and discrete domain are illustrated in Table 4.18 and Table 4.19 respectively.

Table 4.18 Modeling Time of Binary Class WPC in msecs with Numeric Features

Input Size	**oneR**	***J48***	***NB***	***kstar***	***SVM***	***Boosting***	***EM clustering***
70- 30	0	0	0	0	20	0	280
100 – 100	0	20	0	0	20	0	670
200 - 200	0	30	0	0	20	30	2640
300 – 200	0	20	0	0	20	20	5990
300 – 300	0	20	0	0	30	30	24040
350 – 150	0	20	0	0	20	30	12230
400 – 200	0	30	0	0	20	30	13570
400 – 300	20	30	20	0	50	50	27490
400 – 400	0	30	20	0	30	50	28460
Average Induction time	**2.22**	**22.22**	**4.44**	**0**	**25.55**	**26.66**	**12818.88**

Table 4.19 Modeling Time of Binary Class WPC in msecs with Features Discretized by the Present Method

Input Size	**oneR**	***J48***	***NB***	***kstar***	***SVM***	***Boosting***	***EM clustering***
70- 30	0	0	0	0	20	0	130
100 – 100	0	20	0	0	0	20	190
200 - 200	0	20	0	20	60	20	1500
300 – 200	0	0	0	0	30	30	1060
300 – 300	20	30	0	0	60	30	1840
350 – 150	0	20	0	0	50	30	1970
400 – 200	0	20	0	0	2700	30	2390
400 – 300	0	30	0	0	130	30	3200
400 – 400	0	30	0	0	130	50	3319
Average Induction time	**2.22**	**18.88**	**0**	**2.22**	**353.33**	**26.66**	**1733.22**

From the results in Table 4.18 and Table 4.19, it can be inferred that discretization helps to reduce the induction time of the classifiers J48, NB and clustering algorithms. However the induction time of oneR and Boosting methods remain the same in numeric and discrete domain. But the induction time of SVM and kstar classifiers in discrete domain is more than that in the continuous domain. Hence discretization helps in improving the predictive ability of all the classifiers used in the experiments. Comparative analysis of the performance of the various classifiers from these two perspectives is summarized in Table 4.20.

Table 4.20 Performance Comparison of Classifiers in Numeric and Present Discrete Domain for Binary Class WPC

Classifier	**Accuracy in discrete domain as compared to numeric domain**	**Induction time in discrete domain as compared to numeric domain**
NB, J48, EM clustering	Greater	Significantly less
oneR	Greater	Same
Boosting	Less	Same
SVM	Less	Greater
Kstar	Greater	Greater

Two classification algorithms for web page classification are presented in this thesis namely a PWPC a probabilistic web page classifier and MKNN an improved k nearest neighbor algorithm.

4.2.1.4 The Probabilistic Web Page Classifier PWPC

The PWPC works on a discrete web page collection, where each web page is represented as a discrete feature vector. The performance of this classifier is evaluated using web pages that are discretized using two methods namely simple binning and using the discretization algorithm proposed in this thesis. Table 4.21 illustrates a comparative analysis of the predictive accuracy of various classifiers with PWPC. The classifiers are modeled using the features discretized by simple binning. The range of values of each feature is transformed into discrete labels using 10 numbers of bins.

Table 4.21 Comparison of Classification Accuracy with the Present PWPC Classifier for Binary Class WPC Using Simple Binning

Input Size	**ID3**	**oneR**	**Decision table**	***K*star**	**NB**	**J48**	**PWPC**	**MLP**	**SVM**
70- 30	100.00	100.00	100.00	100.00	80.00	100.00	100.00	100.00	100.00
100 – 100	86.95	100.00	86.95	100.00	100.00	100.00	100.00	91.31	94.56
350 – 150	91.83	85.71	91.83	98.97	97.95	88.77	97.95	95.91	96.16
200 – 200	85.65	73.91	94.21	97.11	98.55	94.20	97.10	97.10	94.84
400 – 200	91.5	85.84	90.56	99.05	97.16	87.73	98.10	95.28	96.99
400 – 300	90.71	72.14	87.14	92.14	96.42	92.85	92.14	95.71	96.22
300 – 300	69.23	54.80	89.42	79.80	85.57	76.92	79.01	78.84	95.89
400 – 400	89.79	68.02	92.51	95.23	94.55	89.11	95.23	93.87	95.38
300 – 200	92.00	81.33	86.66	96.00	97.33	86.66	96.00	96.00	96.64
Average	**88.63**	**80.19**	**91.03**	**95.37**	**94.17**	**90.69**	**95.06**	**93.78**	**96.29**

The results in Table 4.21 and Fig 4.10 illustrate that the PWPC exhibits good performance than most of the existing machine learning classifiers namely, ID3, J48, oneR, decision table, MLP and NB. The performance of the SVM and kstar classifiers is better than PWPC. But the SVM classifiers despite their good theoretic foundations and good capability of generalization face a big challenging task with large scale datasets due to their training complexity, high memory requirements and slow convergence.

The performance of the PWPC is also experimented on the web page features discretised by the method present in this thesis. For experiments the threshold B_{size} is set to 2. This ensures that

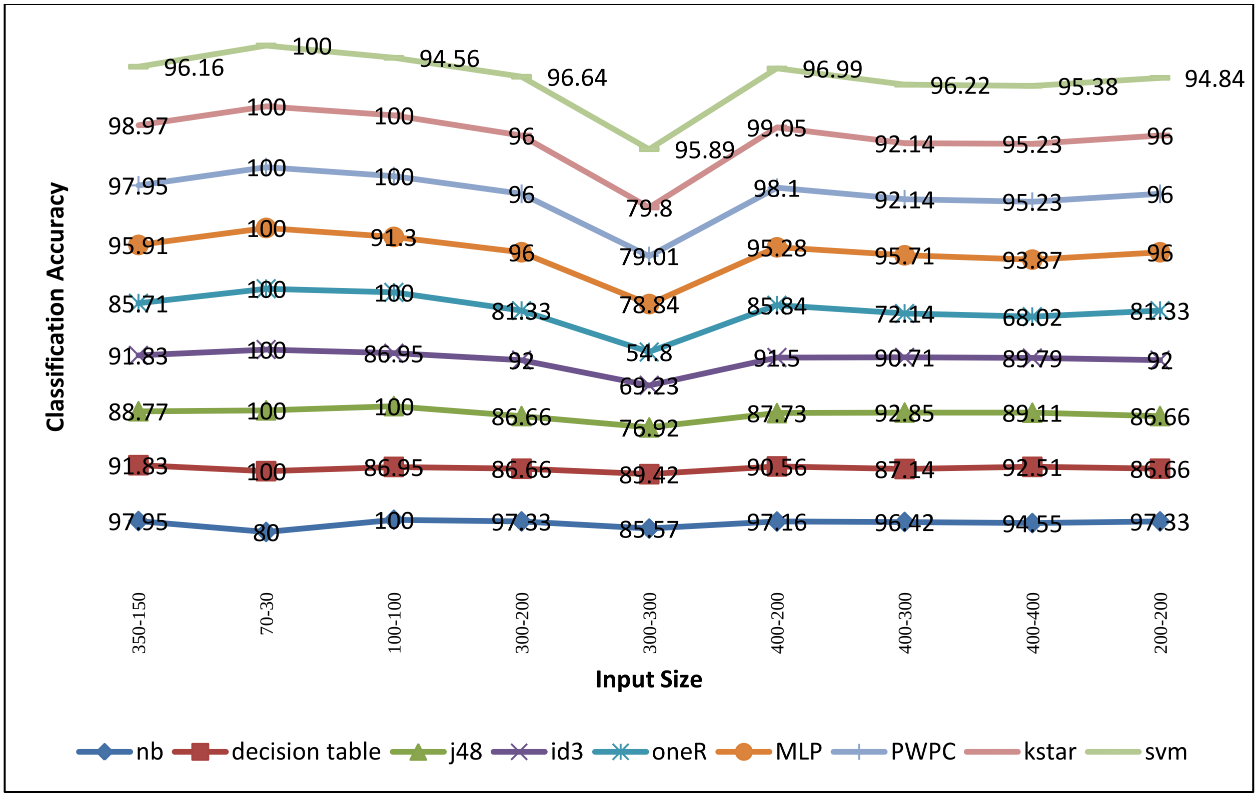


Fig 4.10 Comparison of the Performance of the PWPC using Simple Binning with other Classifiers for Binary Class WPC.

each interval of the feature when discretized has atleast two values. The inconsistence threshold I_{min} is set to 0.5. Table 4.22 illustrates a comparative analysis of all the PWPC classifier with features discretized by both simple binning and discretization by the method proposed in this thesis.

The input data sets where the predictive accuracy of PWPC with the web page features discretized by the method presented in this thesis exceeds or same as that of simple binning features are highlighted in the table.

Table 4.22 Accuracy of the PWPC Classifier on the Present Discretised Features and Simple Binning Discretized Features for Binary Class WPC

Input Size	Present Discretised Features	Simple Binning Features
70- 30	**82.38**	79.53
100 – 100	84.33	85.33
200 - 200	88.7	90.7
300 – 200	**93.42**	91.22
300 – 300	**93.42**	92.48
350 – 150	93.51	93.76
400 – 200	**95.93**	95.25
400 – 300	93.13	93.83
400 – 400	91.92	93.98

4.2.1.5 The Modified k nearest neighbor MKNN Classifier

The second classifier presented in this thesis namely MKNN aims to improve the performance of the traditional KNN. It uses a feature weighting scheme based on the two interestingness measures namely minimum support *min_sup* and minimum confidence *min_conf* that are used in the association rule mining task. Using the features discretised by the proposed discretization method, the feature weights are initially calculated assuming *min_sup* = 0.25 and *min_conf* = 0.755. Table 4.23 gives a comparative analysis of the predictive accuracy of traditional KNN and MKNN for various values of k. The classifiers are modeled using 70 – 30 % split where 70% of the input data set is used or inducing the classification model and the remaining 30% of the input data set is used to validate the model.

Table 4.23 Comparison of Classification Accuracy between KNN and MKNN for Binary Class WPC

Value of K	**K = 9**		**K = 10**		**K = 11**		**K = 12**		**K = 13**	
Input Size/ Classification Algorithm	**KNN**	**MKNN**	**KNN**	**MKNN**	**KNN**	**MKNN**	**KNN**	**MKNN**	**KNN**	**MKNN**
70- 30	82.35	100	82.35	100.00	82.35	100.00	82.35	100.00	82.35	100.00
100-100	82.14	96.43	82.14	96.43	82.14	96.43	82.14	96.43	82.14	96.43
200-200	88.51	92.05	88.51	92.05	89.66	92.05	89.66	92.05	90.80	92.05
300-200	93.26	93.33	93.26	93.33	92.13	93.33	91.01	93.33	91.01	94.44
300-300	95.97	92.74	95.97	93.55	95.97	93.55	96.77	93.55	96.77	93.55
350-150	95.73	100	95.73	100.00	96.58	100.00	96.58	100.00	96.58	100.00
400-200	96.92	95.38	96.15	95.38	96.15	96.15	96.15	96.15	96.92	96.15
400-300	94.61	96.41	94.61	96.41	94.61	97.01	94.61	97.60	95.81	97.60
400-400	92.57	95.43	93.71	96.00	93.71	96.00	94.29	96.00	94.29	96.57
Average	**91.34**	**95.75**	**91.38**	**95.91**	**91.48**	**96.06**	**91.51**	**96.12**	**91.85**	**96.31**

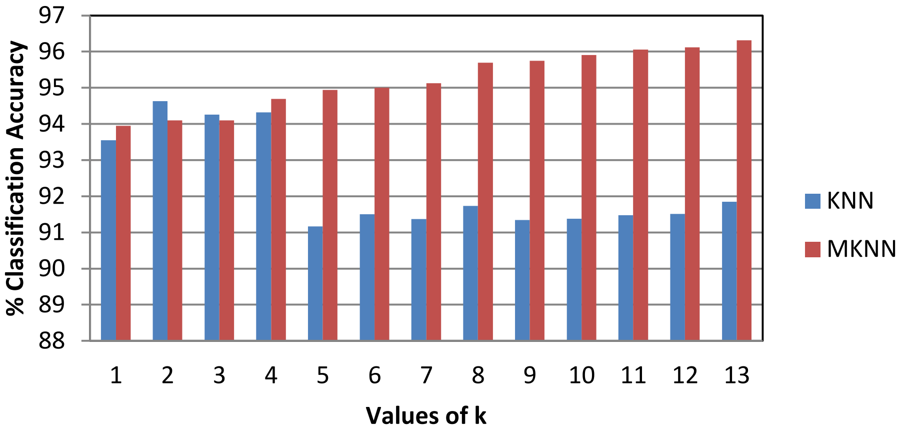


Fig 4.11 Comparison of Average Classification Accuracy between KNN and MKNN for Binary Class WPC

It can be observed from Table 4.23 and Fig 4.11 the performance of the traditional KNN is improved by the proposed feature weighting scheme. MKNN also uses a distance weighted voting scheme, whereby the votes given by k nearest neighbors are weighted using their distance to the test data. This is unlike simple majority voting that is used in the traditional KNN.

Table 4.24 Comparison of Classification Accuracy of the MKNN Binary Class Web Page Classifier with other Existing Classifiers

Data Set	**J48**	**NB**	**KNN**	**MLP**	**SVM**	**MKNN**
70-30	100.00	100.00	82.35	100.00	100.00	**100.00**
100-100	92.86	92.86	82.14	92.86	92.86	**96.43**
200-200	85.06	90.80	89.66	-	88.51	**92.05**
300-200	91.01	92.13	91.01	-	88.76	**94.44**
300-300	91.13	95.97	95.16	-	96.77	93.55
350-150	90.50	97.44	96.58	-	96.58	**100.00**
400-200	90.00	92.31	96.92	-	96.15	96.15
400-300	87.43	93.41	95.21	-	95.21	**97.60**
400-400	90.86	94.86	93.71	-	94.26	**96.57**
Average	**90.98**	**94.42**	**91.42**	**-**	**94.34**	**96.31**

Table 4.24 gives a comparative analysis of various classifiers namely decision tree based (J48), Naïve Bayes (NB), traditional nearest neighbor (KNN), neural network classifier (MLP), support vector machine based classifier (SVM) and the proposed modified KNN called MKNN. Value of k for KNN and for MKNN is 13. Based on the average classification accuracy, the performance of MKNN classifier is better than the traditional KNN and the other classifiers involved in the analysis. The neural network classifier took a longer induction time as the input size increases.

4.2.2 Multi-class Web Page Classification

The multi-class WPC framework proposed in this thesis is developed using a collection of 400 web pages from the WebKB repository. This collection includes labeled web pages of four different categories namely faculty, course, student and project. These web pages are preprocessed through a sequence of steps as discussed in Section 3.1.1. To avoid the problem of the higher value of dimensionality with the feature set initially extracted, the correlation based feature selection algorithm namely CFS is used to identify the most significant features. Table 4.25 illustrates the percentage reduction in dimensionality achieved by this method. The arff file after feature selection using CFS is listed in Appendix H.

Table 4.25 Results after Feature Extraction and Selection by CFS for Multi Class WPC

No. of Web Pages	Initial No. of features	No. of features after CFS	% Reduction
400	7619	38	99.50

The reduction in the number of dimensions achieved this way helps to optimize the resource utilization during the classifier induction process. It also helps in improving the predictive accuracy of these classifiers as the irrelevant and redundant features are eliminated by this feature subset selection process. Feature Selection for multi class WPC is done only using one algorithm namely CFS unlike binary WPC. This is due to the fact although the number of dimensions became significantly less, but the accuracy of the classifiers also decreased with multiple levels of feature selection.

Table 4.26 Comparison of Classification Accuracy with various Features for Multi Class WPC

Features	**oneR**	**J48**	**NB**	**KNN K = 1**	**SVM**
Numeric	34.82	87.5	91.07	76.47	90.17
Simple Binning	34.82	83.03	84.82	73.27	87.5
Present Discrete Features	34.82	78.57	80.35	73.50	87.5
Average	**34.82**	**83.03**	**85.41**	**74.41**	**88.39**

The web page feature vectors are then transformed to discrete domain using the supervised discretization algorithm proposed in this thesis. For experiments the threshold B_{size} is set to 2. This ensures that each interval of the feature when discretized has atleast two values. The inconsistence threshold I_{min} is set to 0.5.

The results in Table 4.26 shows that as the number of classes increases the performance of the classifiers is better in the numeric domain than in the discrete domain. Also, the NB classifiers which are proved to work better with discrete values have given better results only in the numeric domain. The present MKNN classifier works on the features discretized by the present method. This has helped to improve the performance of the traditional KNN algorithm as shown

Table 4.27 Performance Comparison of the PWPC and MKNN for Multi Class WPC with other Classifiers

Classifier	**oneR**	**KNN**	**NB**	**J48**	**PWPC**	**MKNN**	**SVM**
% Accuracy	34.82	76.47	91.07	87.5	81.67	80.43	90.17

in Table 4.27. The value of k in MKNN is 1 and is chosen by cross validation. The best value of k for MKNN is 13.

The results in Table 4.27 show that the performance of MKNN is better than traditional KNN. The PWPC which works on the features discretized by the present method also works better than the traditional KNN. Feature selection has helped to reduce the number of dimensions and hence the performance of traditional KNN which uses distance measures is also increased.

4.3 Medical Image Classification

The experimental results of each phase of the binary-class MIC is discussed in Section 4.3.1 and that of the multi-class MIC is discussed in Section 4.3.2.

4.3.1 Binary Class Medical Image Classification

The binary class MIC is modeled using two classes of retinal fundus images as training set namely normal and severe. The resulting model is able to predict the class of a given test image as one of these two categories. The results of each phase of modeling the binary class MIC is discussed below.

4.3.1.1 Image Feature Extraction and Feature Selection

Due to the non-uniformity in the color distribution of the images, they are pre-processed using histogram equalization. This technique adjusts the local variation in contrast by increasing the contrast in lower contrast area and lowering the contrast in high contrast area. The images are then divided into sub-images of size 36x90 pixels, and the four statistical features namely mean, variance, skewness and kurtosis are extracted from each block. This results in a total feature vector

Table 4.28 Results after feature Extraction and Feature Selection for Binary Class MIC

Input Size	No. of Instances	No. of Features	No. of features by CFS
30-30	60	513	32
50-20	70	513	24
61-32	93	513	43

of size 512. In order to avoid the curse of dimensionality, features that are highly correlated with the class and less inter-correlation are selected using the correlation based feature selection namely CFS. The results after feature extraction and selection of the training images are shown in Table 4.28. The first column in Table 4.28 indicates the various sizes of the two categories of the medical images used in the experiments. For example 30-30 indicates that 30 numbers of normal images and 30 numbers of severe images are used in the study. The second and the third column indicate the total number of images and the total number of features extracted from each image respectively. The fourth column indicates the total number of features selected using the CFS algorithm. This shows that there is a significant reduction in the number of dimensions. Any feature selection algorithm should reduce the number of dimensions without degrading the classifier's performance. To validate this, the performance of various classifiers is evaluated

Table 4.29 Classification Accuracy after Image Feature Extraction for Binary Class MIC

Input Size	oneR	Kstar	NB	J48	SVM	MLP
30-30	70.00	53.33	88.33	68.33	83.33	-
50-20	80.00	71.42	77.14	77.14	87.14	-
61-32	82.79	67.74	81.72	77.41	91.39	-
Average	**77.59**	**64.16**	**82.40**	**74.29**	**87.29**	**-**

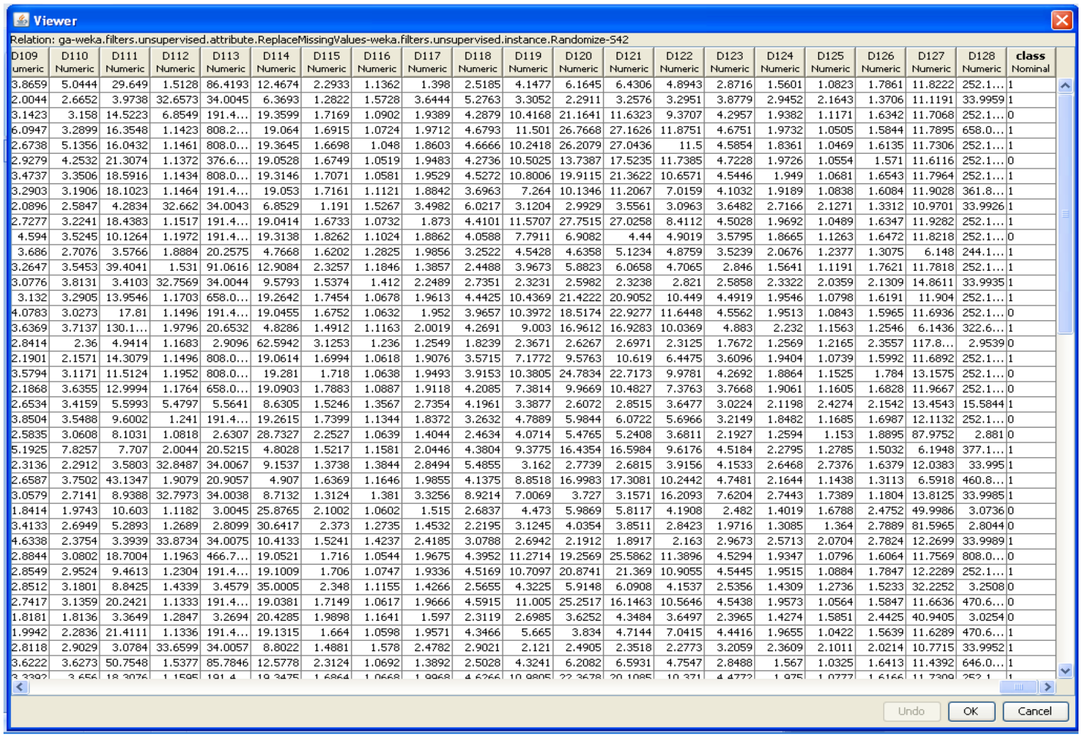
before and after feature selection. The results in terms of percentage classification accuracy are shown in Table 4.29 and 4.30.

The neural network classifier namely MLP, took a longer induction time and could not be modeled due to the huge number of dimensions in the input file. On an average feature selection has helped to improve the performance of all classifiers particularly Kstar, NB, J48, SVM and MLP.

Table 4.30 Classification Accuracy after Feature Selection by CFS for Binary Class MIC

Input Size	**oneR**	**Kstar**	**NB**	**J48**	**SVM**	**MLP**
30-30	68.33	65	93.33	71.67	86.67	86.67
50-20	87.14	78.57	95.71	81.42	88.57	94.28
61-32	74.19	72.04	97.84	86.02	91.39	92.47
Average	**76.55**	**71.87**	**95.63**	**79.70**	**88.88**	**91.14**

The images are represented using the vector space model where each image is a 513 dimensional vector, with the last dimension being the category of the image. The image file in the vector space model is converted to the attribute relation file format called ‘arff’. This is in order to compare the performance of various classifiers before and after feature selection using the data mining tool, WEKA. Fig 4.12 shows a portion of the 61-32 image file in the vector space representation after feature extraction. The 30-30 file after feature selection using CFS in arff is illustrated in Appendix I.



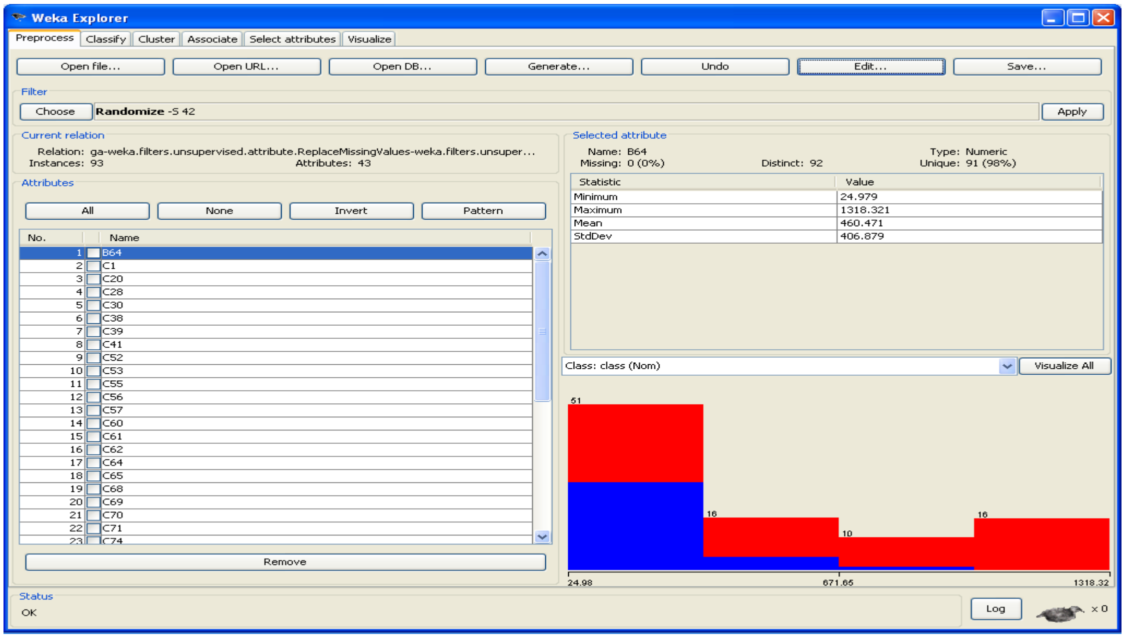
Relation: ga-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.instance.Randomize-S42

D109	D110	D111	D112	D113	D114	D115	D116	D117	D118	D119	D120	D121	D122	D123	D124	D125	D126	D127	D128	class
umeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Nominal
3.8659	5.0444	29.649	1.5128	86.4193	12.4674	2.2933	1.1362	1.398	2.5185	4.1477	6.1645	6.4306	4.8943	2.8716	1.5601	1.0823	1.7861	11.8222	252.1...	1
2.0044	2.6652	3.9738	32.6573	34.0045	6.3693	1.2822	1.5728	3.6444	5.2763	3.3052	2.2911	3.2576	3.2951	3.8779	2.9452	2.1643	1.3706	11.1191	33.9959	1
3.1423	3.158	14.5223	6.8549	191.4...	19.3599	1.7169	1.0902	1.9389	4.2879	10.4168	21.1641	11.6323	9.3707	4.2957	1.9382	1.1171	1.6342	11.7068	252.1...	0
6.0947	3.2899	16.3548	1.1423	808.2...	19.064	1.6915	1.0724	1.9712	4.6793	11.501	26.7668	27.1626	11.8751	4.6751	1.9732	1.0505	1.5844	11.7895	658.0...	1
2.6738	5.1356	16.0432	1.1461	808.0...	19.3645	1.6698	1.048	1.8603	4.6666	10.2418	26.2079	27.0436	11.5	4.5854	1.8361	1.0469	1.6135	11.7306	252.1...	1
2.9279	4.2532	21.3074	1.1372	376.6...	19.0528	1.6749	1.0519	1.9483	4.2736	10.5025	13.7387	17.5235	11.7385	4.7228	1.9726	1.0554	1.571	11.6116	252.1...	0
3.4737	3.3506	18.5916	1.1434	808.0...	19.3146	1.7071	1.0581	1.9529	4.5272	10.8006	19.9115	21.3622	10.6571	4.5446	1.949	1.0681	1.6543	11.7964	252.1...	1
3.2903	3.1906	18.1023	1.1464	191.4...	19.053	1.7161	1.1121	1.8842	3.6963	7.264	10.1346	11.2067	7.0159	4.1032	1.9189	1.0838	1.6084	11.9028	361.8...	1
2.0896	2.5847	4.2834	32.662	34.0043	6.8529	1.191	1.5267	3.4982	6.0217	3.1204	2.9929	3.5561	3.0963	3.6482	2.7166	2.1271	1.3312	10.9701	33.9926	1
2.7277	3.2241	18.4383	1.1517	191.4...	19.0414	1.6733	1.0732	1.873	4.4101	11.5707	27.7515	27.0258	8.4112	4.5028	1.9692	1.0489	1.6347	11.9282	252.1...	1
4.594	3.5245	10.1264	1.1972	191.4...	19.3138	1.8262	1.1024	1.8862	4.0588	7.7911	6.9082	4.44	4.9019	3.5795	1.8665	1.1263	1.6472	11.8218	252.1...	0
3.686	2.7076	3.5766	1.8884	20.2575	4.7668	1.6202	1.2825	1.9856	3.2522	4.5428	4.6358	5.1234	4.8759	3.5239	2.0676	1.2377	1.3075	6.148	244.1...	1
3.2647	3.5453	39.4041	1.531	91.0616	12.9084	2.3257	1.1846	1.3857	2.4488	3.9673	5.8823	6.0658	4.7065	2.846	1.5641	1.1191	1.7621	11.7818	252.1...	1
3.0776	3.8131	3.4103	32.7569	34.0044	9.5793	1.5374	1.412	2.2489	2.7351	2.3231	2.5982	2.3238	2.821	2.5858	2.3322	2.0359	2.1309	14.8611	33.9935	1
3.132	3.2905	13.9546	1.1703	658.0...	19.2642	1.7454	1.0678	1.9613	4.4425	10.4369	21.4222	20.9052	10.449	4.4919	1.9546	1.0798	1.6191	11.904	252.1...	1
4.0783	3.0273	17.81	1.1496	191.4...	19.0455	1.6752	1.0632	1.952	3.9657	10.3972	18.5174	22.9277	11.6448	4.5562	1.9513	1.0843	1.5965	11.6936	252.1...	0
3.6369	3.7137	130.1...	1.9796	20.6532	4.8286	1.4912	1.1163	2.0019	4.2691	9.003	16.9612	16.9283	10.0369	4.883	2.232	1.1563	1.2546	6.1436	322.6...	1
2.8414	2.36	4.9414	1.1683	2.9096	62.5942	3.1253	1.236	1.2549	1.8239	2.3671	2.6267	2.6971	2.3125	1.7672	1.2569	1.2165	2.3557	117.8...	2.9539	0
2.1901	2.1571	14.3079	1.1496	808.0...	19.0614	1.6994	1.0618	1.9076	3.5715	7.1772	9.5763	10.619	6.4475	3.6096	1.9404	1.0739	1.5992	11.6892	252.1...	1
3.5794	3.1171	11.5124	1.1952	808.0...	19.281	1.718	1.0638	1.9493	3.9153	10.3805	24.7834	22.7173	9.9781	4.2692	1.8864	1.1525	1.784	13.1575	252.1...	0
2.1868	3.6355	12.9994	1.1764	658.0...	19.0903	1.7883	1.0887	1.9118	4.2085	7.3814	9.9669	10.4827	7.3763	3.7668	1.9061	1.1605	1.6828	11.9667	252.1...	0
2.6534	3.4159	5.5993	5.4797	5.5641	8.6305	1.5246	1.3567	2.7354	4.1961	3.3877	2.6072	2.8515	3.6477	3.0224	2.1198	2.4274	2.1542	13.4543	15.5844	1
3.8504	3.5488	9.6002	1.241	191.4...	19.2615	1.7399	1.1344	1.8372	3.2632	4.7889	5.9844	6.0722	5.6966	3.2149	1.8482	1.1685	1.6987	12.1132	252.1...	0
2.5835	3.0608	8.1031	1.0818	2.6307	28.7327	2.2527	1.0639	1.4044	2.4634	4.0714	5.4765	5.2408	3.6811	2.1927	1.2594	1.153	1.8895	87.9752	2.881	0
5.1925	7.8257	7.707	2.0044	20.5215	4.8028	1.5217	1.1581	2.0446	4.3804	9.3775	16.4354	16.5984	9.6176	4.5184	2.2795	1.2785	1.5032	6.1948	377.1...	1
2.3136	2.2912	3.5803	32.8487	34.0067	9.1537	1.3738	1.3844	2.8494	5.4855	3.162	2.7739	2.6815	3.9156	4.1533	2.6468	2.7376	1.6379	12.0383	33.995	1
2.6587	3.7502	43.1347	1.9079	20.9057	4.907	1.6369	1.1646	1.9855	4.1375	8.8518	16.9983	17.3081	10.2442	4.7481	2.1644	1.1438	1.3113	6.5918	460.8...	1
3.0579	2.7141	8.9388	32.7973	34.0038	8.7132	1.3124	1.381	3.3256	8.9214	7.0069	3.727	3.1571	16.2093	7.6204	2.7443	1.7389	1.1804	13.8125	33.9985	1
1.8414	1.9743	10.603	1.1182	3.0045	25.8765	2.1002	1.0602	1.515	2.6837	4.473	5.9869	5.8117	4.1908	2.482	1.4019	1.6788	2.4752	49.9986	3.0736	0
3.4133	2.6949	5.2893	1.2689	2.8099	30.6417	2.373	1.2735	1.4532	2.2195	3.1245	4.0354	3.8511	2.8423	1.9716	1.3085	1.364	2.7889	81.5965	2.8044	0
4.6338	2.3754	3.3939	33.8734	34.0075	10.4133	1.5241	1.4237	2.4185	3.0788	2.6942	2.1912	1.8917	2.163	2.9673	2.5713	2.0704	2.7824	12.2699	33.9989	1
2.8844	3.0802	18.7004	1.1963	466.7...	19.0521	1.716	1.0544	1.9675	4.3952	11.2714	19.2569	25.5862	11.3896	4.5294	1.9347	1.0796	1.6064	11.7569	808.0...	0
2.8549	2.9524	9.4613	1.2304	191.4...	19.1009	1.706	1.0747	1.9336	4.5169	10.7097	20.8741	21.369	10.9055	4.5445	1.9515	1.0884	1.7847	12.2289	252.1...	1
2.8512	3.1801	8.8425	1.4339	3.4579	35.0005	2.348	1.1155	1.4266	2.5655	4.3225	5.9148	6.0908	4.1537	2.5356	1.4309	1.2736	1.5233	32.2252	3.2508	0
2.7417	3.1359	20.2421	1.1333	191.4...	19.0381	1.7149	1.0617	1.9666	4.5915	11.005	25.2517	16.1463	10.5646	4.5438	1.9573	1.0564	1.5847	11.6636	470.6...	0
1.8181	1.8136	3.3649	1.2847	3.2694	20.4285	1.9898	1.1641	1.597	2.3119	2.6985	3.6252	4.3484	3.6497	2.3965	1.4274	1.5851	2.4425	40.9405	3.0254	0
1.9942	2.2836	21.4111	1.1336	191.4...	19.1315	1.664	1.0598	1.9571	4.3466	5.665	3.834	4.7144	7.0415	4.4416	1.9655	1.0422	1.5639	11.6289	470.6...	1
2.8118	2.9029	3.0784	33.6599	34.0057	8.8022	1.4881	1.578	2.4782	2.9021	2.121	2.4905	2.3518	2.2773	3.2059	2.3609	2.1011	2.0214	10.7715	33.9952	1
3.6222	3.6273	50.7548	1.5377	85.7846	12.5778	2.3124	1.0692	1.3892	2.5028	4.3241	6.2082	6.5931	4.7547	2.8488	1.567	1.0325	1.6413	11.4392	646.0...	1
3.3392	3.656	18.3076	1.1595	191.4...	19.3475	1.6864	1.0668	1.9968	4.6266	10.9805	22.3678	20.1085	10.371	4.4772	1.975	1.0777	1.6166	11.7309	252.1...	1

Fig 4.12 Image File after Feature Extraction in ARFF for Binary Class MIC

The last column in the file as in Fig 4.12 is a nominal attribute that indicates the category of the image i.e., 1 indicates that the image belongs to the ‘normal’ category and 0 indicates that the image belongs to the ‘severe’ category.

The image file in arff after feature selection using CFS is illustrated in Fig 4.17. It illustrates the number of images and the number of features selected for the 61-32 image file.



Randomize -S 42

Relation: ga-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsuper...
Instances: 93 Attributes: 43

No.		Name
1	□	B64
2	□	C1
3	□	C20
4	□	C28
5	□	C30
6	□	C38
7	□	C39
8	□	C41
9	□	C52
10	□	C53
11	□	C55
12	□	C56
13	□	C57
14	□	C60
15	□	C61
16	□	C62
17	□	C64
18	□	C65
19	□	C68
20	□	C69
21	□	C70
22	□	C71
23	□	C74

Name: B64 Type: Numeric
Missing: 0 (0%) Distinct: 92 Unique: 91 (98%)

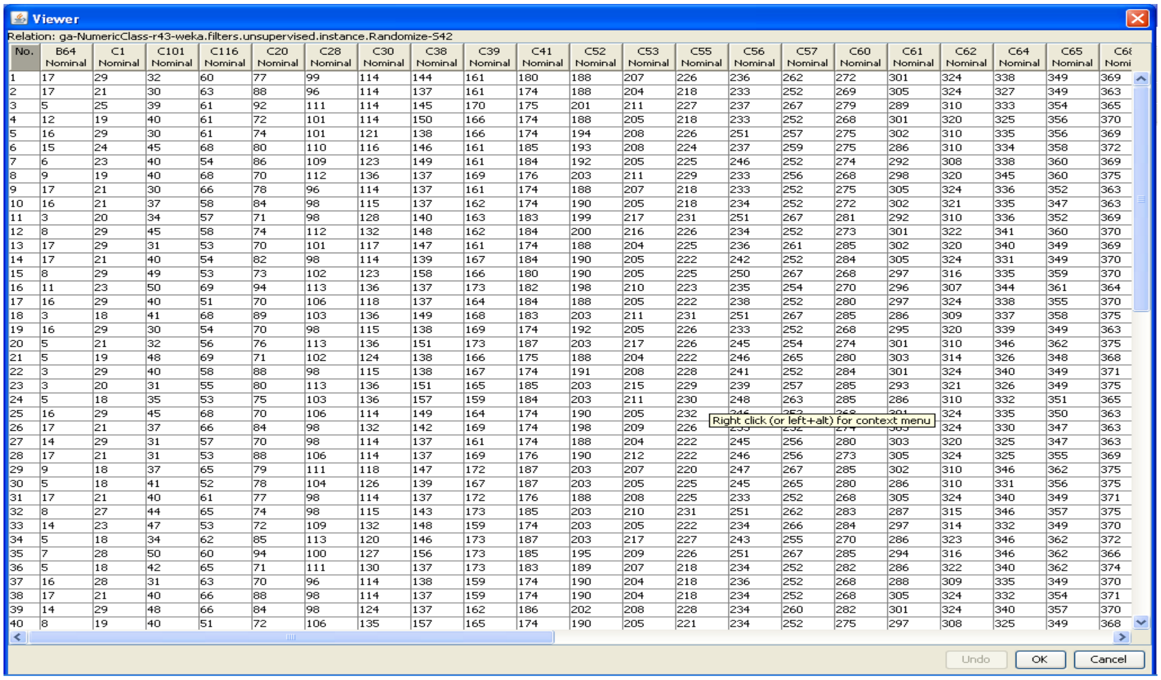
Statistic	Value
Minimum	24.979
Maximum	1318.321
Mean	460.471
StdDev	406.879

Class: class (Nom)

OK

Fig 4.13 The Image File after Feature Selection in ARFF for Binary Class MIC

As seen from the screen shot in Fig 4.13, the data type of each feature is numeric.



Relation: ga-NumericClass-r43-weka.filters.unsupervised.instance.Randomize-S42

No.	B64	C1	C101	C116	C20	C28	C30	C38	C39	C41	C52	C53	C55	C56	C57	C60	C61	C62	C64	C65	C6
	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nomi
1	17	29	32	60	77	99	114	144	161	180	188	207	226	236	262	272	301	324	338	349	369
2	17	21	30	63	88	96	114	137	161	174	188	204	218	233	252	269	305	324	327	349	363
3	5	25	39	61	92	111	114	145	170	175	201	211	227	237	267	279	289	310	333	354	365
4	12	19	40	61	72	101	114	150	166	174	188	205	218	233	252	268	301	320	325	356	370
5	16	29	30	61	74	101	121	138	166	174	194	208	226	251	257	275	302	310	335	356	369
6	15	24	45	68	80	110	116	146	161	185	193	208	224	237	259	275	286	310	334	358	372
7	6	23	40	54	86	109	123	149	161	184	192	205	225	246	252	274	292	308	338	360	369
8	9	19	40	68	70	112	136	137	169	176	203	211	229	233	256	268	298	320	345	360	375
9	17	21	30	66	78	96	114	137	161	174	188	207	218	233	252	275	305	324	336	352	363
10	16	21	37	58	84	98	115	137	162	174	190	205	218	234	252	272	302	321	335	347	363
11	3	20	34	57	71	98	128	140	163	183	199	217	231	251	267	281	292	310	336	352	369
12	8	29	45	58	74	112	132	148	162	184	200	216	226	234	252	273	301	322	341	360	370
13	17	29	31	53	70	101	117	147	161	174	188	204	225	236	261	285	302	320	340	349	369
14	17	21	40	54	82	98	114	139	167	184	190	205	222	242	252	284	305	324	331	349	370
15	8	29	49	53	73	102	123	158	166	180	190	205	225	250	267	268	297	316	335	359	370
16	11	23	50	69	94	113	136	137	173	182	198	210	223	235	254	270	296	307	344	361	364
17	16	29	40	51	70	106	118	137	164	184	188	205	222	238	252	280	297	324	338	355	370
18	3	18	41	68	89	103	136	149	168	183	203	211	231	251	267	285	286	309	337	358	375
19	16	29	30	54	70	98	115	138	169	174	192	205	226	233	252	268	295	320	339	349	363
20	5	21	32	56	76	113	136	151	173	187	203	217	226	245	254	274	301	310	346	362	375
21	5	19	48	69	71	102	124	138	166	175	188	204	222	246	265	280	303	314	326	348	368
22	3	29	40	58	88	98	115	138	167	174	191	208	228	241	252	284	301	324	340	349	371
23	3	20	31	55	80	113	136	151	165	185	203	215	229	239	257	285	293	321	326	349	375
24	5	18	35	53	75	103	136	157	159	184	203	211	230	248	263	285	286	310	332	351	365
25	16	29	45	68	70	106	114	149	164	174	190	205	232	246	252	268	301	324	335	350	363
26	17	21	37	66	84	98	132	142	169	174	198	209	226	235	252	274	303	324	330	347	363
27	14	29	31	57	70	98	114	137	161	174	188	204	222	245	256	280	303	320	325	347	363
28	17	21	31	53	88	106	114	137	169	176	190	212	222	246	256	273	305	324	325	355	369
29	9	18	37	65	79	111	118	147	172	187	203	207	220	247	267	285	302	310	346	362	375
30	5	18	41	52	78	104	126	139	167	187	203	205	225	245	265	280	286	310	331	356	375
31	17	21	40	61	77	98	114	137	172	176	188	208	225	233	252	268	305	324	340	349	371
32	8	27	44	65	74	98	115	143	173	185	203	210	231	251	262	283	287	315	346	357	375
33	14	23	47	53	72	109	132	148	159	174	203	205	222	234	266	284	297	314	332	349	370
34	5	18	34	62	85	113	120	146	173	187	203	217	227	243	255	270	286	323	346	362	372
35	7	28	50	60	94	100	127	156	173	185	195	209	226	251	267	285	294	316	346	362	366
36	5	18	42	65	71	111	130	137	173	183	189	207	218	234	252	282	286	322	340	362	374
37	16	28	31	63	70	96	114	138	159	174	190	204	218	236	252	268	288	309	335	349	370
38	17	21	40	66	88	98	114	137	159	174	190	204	218	234	252	268	305	324	332	354	371
39	14	29	48	66	84	98	124	137	162	186	202	208	228	234	260	282	301	324	340	357	370
40	8	19	40	51	72	106	135	157	165	174	190	205	221	234	252	275	297	308	325	349	368

Right click (or left+alt) for context menu

Fig 4.14 The Image File after Discretization for Binary Class MIC

4.3.1.2 Image Feature Discretization

In order to explore the performance of the classifiers in the numeric and the discrete domain, each feature in the input file is discretized using the present method, with the inconsistence threshold I_{min} set to 0.5. The resulting file with discrete features is then converted to arff, ready to explore the performance of the classifiers in numeric and discrete domain as shown in Fig 4.14. The features are also discretized using some of the existing discretization methods namely simple binning and entropy based discretization MDL. Table 4.31 shows the performance of various classifiers using numeric features and the features discretized using all three methods. The classifiers are modeled using 10-fold cross validation.

Table 4.31 Comparative Analysis of Classification Accuracy for Binary Class MIC with various Features

Input Size	**Discretization Methods**	**Classifiers**				
		oneR	**J48**	**NB**	**Kstar**	**SVM**
30-30	No Discretization	68.33	71.66	93.33	65	86.67
	Simple Binning	76.67	76.67	90	76.67	88.36
	MDL	78.33	83.33	100	98.33	96.67
	Present Method	**90**	**85**	**100**	**100**	**100**
50-20	No Discretization	87.14	81.42	95.71	78.57	88.57
	Simple Binning	74.28	81.42	95.71	84.28	90
	MDL	90	91.42	100	95.71	95.71
	Present Method	**95.71**	**95.71**	**100**	**97.14**	**100**
61-32	No Discretization	74.19	86.02	97.84	72.04	91.39
	Simple Binning	66.67	75.26	93.51	77.41	92.47
	MDL	78.49	83.87	100	93.54	97.84
	Present Method	**89.24**	**84.94**	**100**	**100**	**100**

Results in Table 4.31 shows that for all three input files the performance of NB , KNN and SVM classifiers are better in discrete than in numeric domain. The performance of the J48 classifier with two of the input files is better in discrete than in numeric domain. The present discretization method has helped to improve the performance of all these classifiers. The most frequently used performance evaluation metric in medical decision making is the area under the ROC called AUC. The AUC value of a classifier has to be between 0.5 and 1, where a classifier with an AUC of being a perfect classifier. The AUC for binary class MIC is calculated using the True positive rate (TPR) and the false positive rate (FPR) of the classifier as shown in Table 4.32.

Table 4.32 Comparative Analysis of Binary Class MIC with various Features using AUC

Input Size	**Discretization Methods**	**Classifiers**				
		oneR	**J48**	**NB**	**Kstar**	**SVM**
30-30	No Discretization	0.70	0.88	0.99	0.83	0.89
	Simple Binning	0.58	0.70	0.99	0.93	0.90
	MDL	0.73	0.87	1	0.99	0.97
	Present Method	**0.86**	0.824	**1**	**1**	**1**
50-20	No Discretization	0.68	0.78	0.99	0.90	0.87
	Simple Binning	0.76	0.84	0.99	0.87	0.88
	MDL	0.78	0.80	1	1	0.97
	Present Method	**0.93**	**0.95**	**1**	**1**	**1**
61-32	No Discretization	0.70	0.87	0.99	0.83	0.90
	Simple Binning	0.57	0.70	0.99	0.93	0.90
	MDL	0.73	0.87	1	0.99	0.97
	Present Method	**0.86**	0.82	**1**	**1**	**1**

The AUC values in Table 4.32 illustrate that the performance of NB, KNN and SVM is better in the discrete domain than in numeric domain. It also shows that present discretization has enhanced their performance and identified them as perfect classifiers.

4.3.1.3 The Probabilistic Medical Image Classifier PMIC and MKNN Medical Image Classifier

The image file with the features discretized by the present method is used to calculate the weight of each feature. The feature weights are calculated by generating strong association rules using a support threshold of 0.5 and a confidence threshold of 0.75. These feature weights together with the distance weighting scheme is used to improve the performance of the traditional KNN. The classifiers are modeled using 70-30% split. Table 4.33 shows the comparative analysis of a KNN algorithm and MKNN. The value of k for KNN and MKNN is chosen by cross validation.

Table 4.33 Comparative Analysis of Classification Accuracy of Binary Class MIC with KNN and MKNN

Input Size	**KNN**	**MKNN**	**PMIC**
30-30	81.66 (k=1)	88.88 (k =1)	85.70
50-20	90.00 (k = 1)	95.38 (k= 13)	85.00
61-32	82.79 (k = 1)	92.85 (k = 1)	90.23
Average	**84.82**	**92.37**	**86.98**

The results in Table 4.33 show that the performance of MKNN is better than traditional KNN for all input files.

Table 4.34 Comparative Analysis of Classification Accuracy of PMIC with other classifiers for Binary Class MIC

Input Size	PMIC	oneR	J48	SVM
30-30	85.70	68.33	71.66	86.67
50-20	85.00	87.14	81.42	88.57
61-32	90.23	74.19	86.02	91.39
Average	**86.97**	**76.55**	**86.02**	**88.87**

The PMIC classifier works on the discrete features. Based on the average classification accuracy the performance of PMIC is better than the rule based and decision tree classifiers. Although SVM classifiers have a good theoretic foundation and good capability of generalization, they face a big challenging task with large scale data sets due to their training complexity, high memory requirements and slow convergence.

4.3.2 Multi-Class Medical Image Classification

The multi-class MIC is modeled using three categories of retinal fundus images namely normal, moderate and severe. The images are first pre-processed using adaptive histogram equalization. They are then divided into sub-images and four statistical features namely mean, variance, skewness and kurtosis are extracted from each block. Each image is represented using a 513-dimensional vector with the last feature being the image category. This image file is stored in the arff format. Table 4.35 shows for each input combinations the total number of images, number of features extracted and the number of features selected using CFS. Appendix J illustrates the 30-50-30 file after feature selection using CFS.

Table 4.35 Results after Feature Extraction and Feature Selection for Multi Class MIC

Input Size	No. of Instances	No. of Features	No. of features by CFS
30-30- 30	90	513	34
60-30-30	120	513	36
30-50-30	110	513	39
61-52-78	145	513	47

Table 4.36 Classification Accuracy after Feature Extraction for Multi Class MIC

Input Size	oneR	J48	NB	Kstar	SVM
30-30-30	56.67	46.67	64.44	35.55	63.33
30-50-30	45.45	54.54	60.90	29.09	60.90
60-30-30	59.16	55.00	58.33	51.67	65.83
61-52-32	51.72	62.75	62.06	43.44	62.09
Average	**53.25**	**54.74**	**61.43**	**39.94**	**63.04**

The results indicate that feature selection has significantly reduced the number of dimensions. In order to explore the advantages of this dimensionality reduction in terms of predictive accuracy, various classifiers were modeled using the features extracted initially and the features that resulted after CFS. Their predictive accuracy in percentage is given in Table 4.36 and 4.37.

Table 4.37 Classification Accuracy after Feature Selection for Multi Class MIC

Input Size	oneR	J48	NB	Kstar	SVM
30-30-30	60.00	58.89	64.44	67.78	60.00
30-50-30	45.45	54.54	66.36	60.90	63.63
60-30-30	68.33	57.50	67.50	60.83	67.50
61-52-32	49.65	63.44	67.58	56.55	71.72
Average	**55.86**	**58.59**	**66.47**	**61.52**	**65.71**

There is a significant improvement in the predictive accuracy of all classifiers after feature selection.

Table 4.38 Comparative Analysis of Classification Accuracy for Multi Class MIC with various Features

Input Size	**Discretization Methods**	**Classifiers**				
		oneR	**J48**	**NB**	**KNN**	**SVM**
30-30-30	No Discretization	60.00	58.89	64.44	67.78	60.00
	Simple Binning	48.88	42.22	72.22	53.33	71.11
	MDL	61.11	70.00	84.44	81.11	80.00
	Present Method	46.67	55.55	**91.11**	**84.44**	**91.11**
30-50-30	No Discretization	45.45	54.54	66.36	60.90	63.63
	Simple Binning	63.63	56.36	69.09	61.81	64.54
	MDL	66.36	75.45	85.45	76.36	86.36
	Present Method	61.81	64.54	**88.18**	**80.90**	**87.27**
60-30-30	No Discretization	68.33	57.50	67.5	60.83	67.5
	Simple Binning	60.83	57.50	73.33	66.67	69.16
	MDL	56.67	72.5	84.16	72.50	76.67
	Present Method	65.83	**72.5**	**92.5**	**86.67**	**90.83**
61-52-32	No Discretization	49.65	63.44	67.58	56.55	71.72
	Simple Binning	62.06	53.10	75.17	58.62	67.58
	MDL	64.82	68.27	82.06	71.72	75.17
	Present Method	**73.79**	**71.72**	**91.03**	**83.45**	**86.89**

The image features are then discretized by the present method discussed in Section 3.1.4. A comparative analysis of the predictive accuracy of various classifiers using various features namely numeric features, features discretised by simple binning, features discretised by entropy method and the present method is shown in Table 4.38. The accuracy of NB, KNN and SVM classifiers with features discretised by the present method is more than the other types of features for all input sizes. However the accuracy of the other classifiers namely oneR and J48 with features discretised by the present method increases with the increase in the input size. As the size of the input increases, the number of images used to train the model also increases. Hence the model learns well to distinguish the various categories of images.

Table 4.39 Comparative Analysis of AUC for Multi Class MIC with various Features

Input Size	**Discretization Methods**	**Classifiers**				
		oneR	**J48**	**NB**	**KNN**	**SVM**
30-30-30	No Discretization	0.7	0.73	0.84	0.84	0.74
	Simple Binning	0.61	0.59	0.88	0.76	0.80
	MDL	0.70	0.77	0.95	0.91	0.89
	Present Method	0.6	0.61	**0.96**	**0.91**	**0.94**
30-50-30	No Discretization	0.57	0.65	0.80	0.77	0.92
	Simple Binning	0.7	0.66	0.86	0.76	0.77
	MDL	0.69	0.80	0.96	0.90	0.90
	Present Method	**0.7**0	0.71	**0.96**	0.88	**0.90**
60-30-30	No Discretization	0.74	0.68	0.87	0.79	0.75
	Simple Binning	0.68	0.69	0.90	0.82	0.78
	MDL	0.59	0.80	0.96	0.88	0.84
	Present Method	**0.74**	**0.83**	**0.97**	**0.95**	**0.96**
61-52-32	No Discretization	0.61	0.75	0.87	0.76	0.79
	Simple Binning	0.69	0.66	0.89	0.79	0.77
	MDL	0.71	0.82	0.95	0.87	0.84
	Present Method	**0.79**	0.78	**0.96**	**0.92**	**0.92**

A comparative analysis of the area under the receiver operating characteristics curve, AUC, of various classifiers using various features namely numeric features, features discretised by simple binning, features discretised by entropy method and the present method is shown in Table 4.39. The results show that the performance of the various classifiers in the discrete domain is better than the numeric domain. A comparative analysis of KNN and MKNN is shown in Table 4.40. The value of k for KNN is chosen by cross validation and is 1. For the 30-30-30 input file, the highest accuracy of MKNN is 70.37 for both the value of k as 1 and 2. For the 30-50-30 input file, the highest accuracy of MKNN is 66.67 for values of k from 1 to 4. For the 60-30-30 input file, the highest accuracy of MKNN is 75.00 for values of k from 4 to 6. For the 61-52-32 input file, the highest accuracy of MKNN is 76.74 for both the value of k as 11 and 12. This gives the value of k with the highest accuracy using cross validation for each input file.

Table 4.40 Comparative Analysis of classification Accuracy of KNN and MKNN for Multi Class MIC

Input Size	**KNN**	**MKNN**
30-30-30	67.78	70.37
30-50-30	60.90	66.67
60-30-30	60.83	75.00
61-52-32	56.55	76.74
Average	**61.52**	**72.20**

Table 4.40 also shows the predictive accuracy of the MKNN classifier is better than KNN for all input files. A comparative analysis of the present PMIC with some of the existing classifiers is shown in Table 4.41.

Table 4.41 Comparative Analysis of Classification Accuracy of PMIC with other Classifiers for Multi Class MIC

Input Size	**PMIC**	**oneR**	**J48**	**SVM**
30-30-30	80.00	60.00	58.84	60.00
30-50-30	76.67	45.45	54.54	63.63
60-30-30	80.00	68.33	57.50	67.50
61-52-32	80.68	49.65	63.44	71.72
Average	**79.33**	**55.86**	**58.58**	**65.71**

Based on the average predictive accuracy as illustrated in Table 4.41, the performance of the PMIC is significantly better than the other classifiers used in the analysis.

4.4 Area under the curve AUC for MKNN Medical Image Classification

In medical decision making, the area under the receiver operating characteristics curve called AUC of a classifier is another metric used for evaluating its performance. It is a plot between TPR and FPR of a classifier. For an acceptable classifier the AUC is between 0.5 and 1, where an AUC of 1 represents a perfect classifier. AUC analysis for the binary class MIC is done using the 30-30 input file. This file is randomized 10 times and each of the randomized file is saved. The MKNN classifier is modeled using a 70-30% split using each one of this randomized file. The TPR and FPR value obtained in each case is shown in Table 4.42. A detailed discussion of the same is in Section 3.3.

Table 4.42 TPR and FPR for Binary Class MIC

Input File	**TPR**	**FPR**
1	1	0.17
2	1	0.11
3	1	0.11
4	1	0.22
5	1	0.11
6	1	0.40
7	1	0.36
8	0.8	0.50
9	1	0.22
10	1	0.25

The confusion matrix for the first input file is shown in Table 4.43

Table 4.43 Confusion Matrix for Binary Class MIC

Actual Class	Predicted Class	
	Normal	Severe
Normal	6	0
Severe	2	10

Hence $TPR = \frac{TP}{P} = \frac{6}{6} = 1$ and $FPR = \frac{FP}{N} = \frac{2}{12} = 0.17$

The ROC plot for binary class MIC is shown in Fig 4. 15. As seen from the graph the classifier is a point above the diagonal line for all the input files. It is a perfect classifier for nine of the input files. The AUC calculated using the trapezoidal rule for this classifier is 0.89. Table 4.44 gives the TPR and FPR values for the multi-class MIC using the 30-30-30 input file.

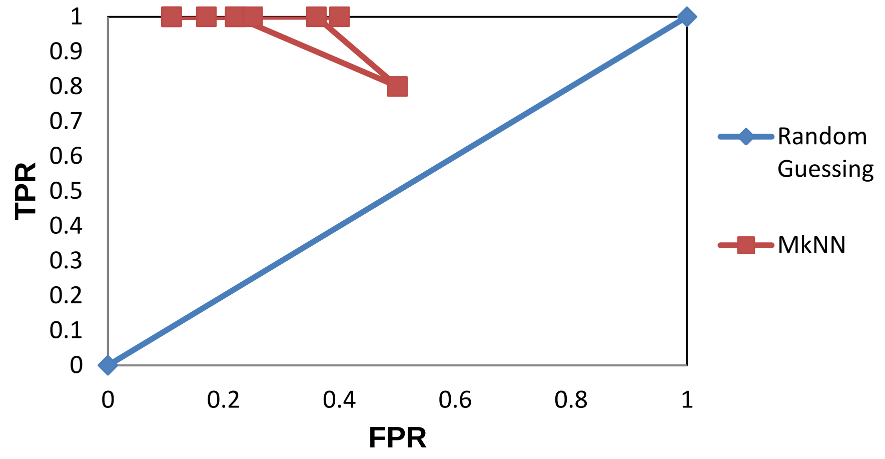


Fig 4.15 ROC Graph for Binary Class MIC

Table 4.44 TPR and FPR for Multi Class MIC

Input file	TPR	FPR
1	0.796	0.121
2	0.783	0.138
3	0.472	0.286
4	0.574	0.214
5	0.603	0.202
6	0.734	0.164
7	0.549	0.260
8	0.645	0.220
9	0.675	0.179
10	0.728	0.176

Table 4.45 Confusion Matrix for Multi Class MIC

Predicted Class	Actual Class		
	Normal	Moderate	Severe
Normal	9	0	0
Moderate	1	5	4
Severe	1	1	6

The confusion matrix for the first input file is shown in Table 4.45. For multi-class classification with 'm' number of classes, the TPR and FPR values of the classification model is calculated in 'm' iterations. Each one of the iterations assumes one of the classes as positive and all the other classes as negative. The weighted average of the TPR and FPR value obtained in each iteration gives the final TPR and FPR of the classifier.

So, TPR for 'Normal' = $\frac{9}{9}$ = 1.

TPR for 'Moderate' = $\frac{5}{10} = 0.5$ and

TPR for 'Severe' = $\frac{6}{8} = 0.75$.

Hence TPR by weighted average = $\big((1\ x\ 11) + (0.5\ x\ 6) + (0.75\ x\ 10)\big)/27$

= $(11 + 3 + 7.5)/27$

= $21.5/27$

= 0.796

The FPR for 'Normal' = $\frac{2}{27-9} = \frac{2}{18} = 0.111$.

FPR for 'Moderate' = $\frac{1}{27-10} = \frac{1}{17} = 0.0588$ and

FPR for 'Severe' = $\frac{4}{27-8} = \frac{4}{19} = 0.21$.

Hence FPR by weighted average = $\big((0.111\ x\ 9) + (0.0588\ x\ 10) + (0.21\ x\ 8)\big)/27$

= $(0.999 + 0.588 + 1.68)/27$

= $3.267/27$

= 0.121

The ROC for multi-class MIC is shown in Fig 4.16. This is plotted using the TPR and FPR values shown in Table 4.44. For all the input files the classifier is a point above the diagonal line which is the random guessing line. The AUC for this classifier using Algorithm 2 [107] is 0.6803.

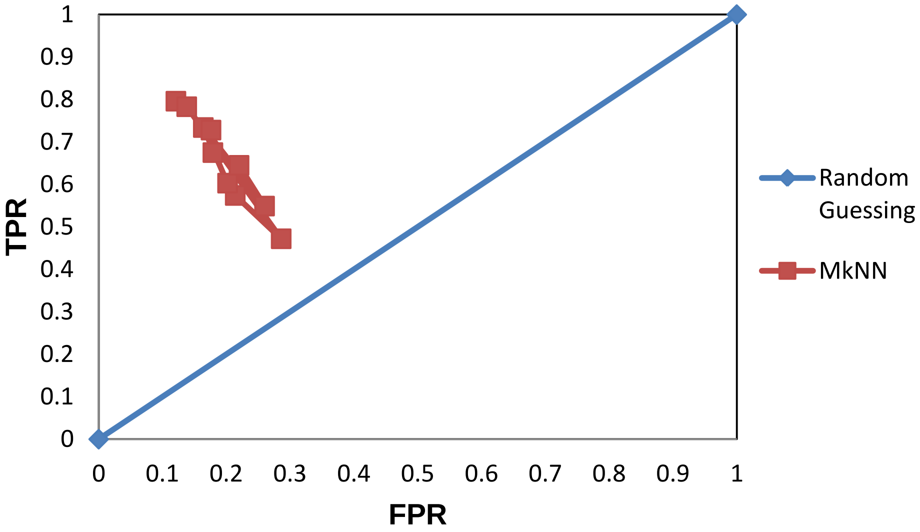


Fig 4.16 ROC Graph for Multi Class MIC

4.5 Summary

In this chapter the results and discussion of the present framework for web page and medical image classification models. It starts with a description of the data sets used in the analysis. The results of each phase of the WPC and the MIC models for both binary and multi-class models are summarized. The classifiers are evaluated using two metrics namely predictive accuracy and area under the receiver operating characteristics curve, AUC. The present feature selection and discretization methods have helped to improve the performance of the classifiers. The experimental results show that the present MKNN classification model performs better than traditional KNN. The performance of the PWPC and PMIC is also better than the existing classifiers used in the analysis.

CHAPTER 5

CONCLUSION

The volume of data generated and accumulated in large organizations increases exponentially every year. So the traditional data analysis methods are no longer suitable in such situations. They are greatly challenged by this data tsunami in terms of heterogeneous nature of the data, huge scalability, timeliness and privacy problems.

Data Mining is a technique that combines the traditional data analysis methods with sophisticated algorithms to process large volumes of data. It enables to explore and analyze new types of data. However these traditional data mining techniques are developed to work only on data that is in structured format, i.e. a standard table. But due to many reasons data accumulated is no longer structured like the earth's observation data, textual data, web page data, medical image data etc. Hence discovering useful patterns using data mining techniques from such non-traditional data formats is quite challenging.

Motivated by these facts in this thesis algorithms for improving subject based classification of web page and medical image data were designed and implemented. A detailed description of each algorithm was discussed in Chapter 3. It includes various pre-processing steps and classification. The pre-processing steps include feature extraction, selection and discretization. Experimental results and analysis of the present framework using web pages and medical images was discussed in Chapter 4. Analysis of the present work was done for both binary and multi-class classification. The results of the analysis have signified that feature selection helps in improving the predictive accuracy. A significant observation made from the experimental results of feature selection for binary class and multi class WPC is that multiple levels of feature selection has improved the performance of only the binary class classification. For multi class WPC feature selection is done only using CFS algorithm. This is due to the fact although the number of dimensions became significantly less, but the accuracy of the classifiers also decreased with multiple levels of feature selection.

The performance of many of the binary-class classifiers was better in the discrete domain than in continuous domain. The predictive accuracy of the present MKNN classification framework with both web page and image data sets was better than the traditional KNN for both binary-class and multi-class classification. The performance of the present PWPC and PMIC was also better than many of the existing classifiers for both binary-class and multi-class classification. The following Section 5.1 illustrates the specific contributions of this research and Section 5.2 throws a light on future research scope in this direction.

5.1 SPECIFIC CONTRIBUTIONS

Following is a list of specific contributions made by this research:

- The web page classification model presented in this research represents each web page using the Bag of Words BoW representation in vector space model. It uses the on page features that are extracted from each web page.
- Handling high dimensional data is one of the challenges faced by the data mining algorithms. Two new feature selection methods for classifying web page data sets are presented in this research. The first one is a hybrid model which uses the correlation based feature selection (CFS) followed by the decision tree induction algorithm namely C4.5. CFS chooses subsets of features that are highly correlated with the class label and low intra-correlation. With these features a decision tree model is first induced for the data set

of interest using the C4.5 algorithm. The features that are present in the final pruned tree are identified to be the best representative features.

- A novel feature selection framework for classifying web page data sets using the Ward's minimum variance measure is presented in this research. This measure was initially used to cluster data of similar characteristics together. As a novel approach, it has been proved in this thesis that this measure can be used for dimensionality reduction also. The classification accuracy of many machine learning algorithms was found to increase after feature selection.
- The performance of the Naïve Bayes (NB) and Decision Tree (DT) WPC model is improved using a discretization algorithm. This algorithm makes NB to increase its classification accuracy in discrete domain than in the numeric domain. Also, the time taken to build the NB and Decision tree (DT) model with discrete features is less than with numeric features.
- A probabilistic web page classifier (PWPC) is presented. It uses the Bayes probability theorem to find the predictive power of each attribute-value towards the class labels. An attribute-value similarity measure between the test web page and each of the training web pages is then used to predict the class of the test web page. Experimental results show that this model works well with binary class and multi class WPC.
- A modified K Nearest Neighbor Algorithm (MKNN) for WPC is designed and implemented in this research. The MKNN uses 1) a feature weighting scheme based on the interestingness measures used in the association rule mining and 2) a distance weighted voting instead of simple majority voting. These two have made MKNN to perform better than the traditional KNN for both binary class and multi class WPC.

- A novel framework for classifying binary class medical Images namely the retinal fundus images is presented in this thesis. This framework represents each image in the vector space model using the statistical features extracted from them. The Feature extraction phase does not require domain knowledge of the data set being mined. The number of features are further reduced using CFS. Features are then quantized using the present discretization method.
- Two classification algorithms for medical images namely the Probabilistic Medical Image Classification (PMIC) and the Modified K Nearest Neighbor (MKNN) are designed and implemented. Both of the methods are same as those used with web page data sets. These algorithms work well for both binary class and multi class MIC.

5.2 FUTURE SCOPE OF WORK

Some possibilities of future research in the direction presented in this thesis are:

- The number of categories experimented in the present work for multi-class WPC and MIC are four and three respectively. But this may reduce the predictive accuracy of the classifiers since it uses only the on-page features. Inorder to avoid this, the feature set can be refined by adding other representative features of the data. For example, the features present in the URL of a web page can be added with its on-page features to improve the performance of the classification model.
- The various phases of the present framework namely feature selection; discretization and classification methods can be experimented with other types of data sets like biological gene expression data sets, etc. However, these data sets need to be transformed into a format suitable for the proposed model in advance.

- The supervised discretization algorithm used in this research can be extended for feature selection also. A feature whose values are mapped to the same discrete label/interval after discretization is said be to less significant in predicting the target variable. Hence it can be discarded.
- The feature selection framework for binary class WPC presented in this research involves two phases namely 1) identify clusters of redundant features using the Ward’s measure and 2) eliminate redundant features. In this research the second phase chooses the best feature from each cluster based on the information gain. Instead a different scheme can be used to choose the best of a subset of features in each cluster.
- The performance of the MKNN presented in this thesis can be improved by exploring a different instance weighting scheme.
- In this research, statistical features are extracted from images for MIC. Instead a different image representation scheme can be adopted and the performance of the present feature selection, discretization and classification methods can be experimented.

REFERENCES

1. Steve Lohr. (2012, Feb 11). *The Age of Big Data* [Online]. Available: http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=all

2. Mitra, S. and Acharya, T. *Data Mining: multimedia, soft computing, and bioinformatics*. Hoboken, NJ: John Wiley,2003.

3. Bramer, M. *Principles of Data Mining*. London: Springer-Verlag, 2007.

4. Choi, B. and Yao, Z. “Web Page Classification”, in *Foundations and Advances in Data Mining*., Chu, W., and Lin, Y. T (Eds). Berlin Heidelberg: Springer-Verlag, 2005, pp. 221-274.

5. Buckley, C., Salton, G., and Allan, J. ‘The effect of adding relevance information in a relevance feedback environment’, in *Proc. of SIGIR ’94:17th ACM Int. conf. on research and development in information retrieval,* Dublin, Ireland, 1994, pp: 292-300.

6. Gonzalez, R. C., and Woods, R. E. *Digital Image Processing*, 2nd Edn, New Jersey: Prentice Hall. 2001.

7. Tan, P.N., Steinbach, M., & Kumar, V. *Introduction to Data Mining,* New York: Pearson Addison Wesley, 2006.

8. Miller, Harvey, J., and Han, J. *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis, 2003.

9. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, 2nd edition Berlin Heidelberg: Springer-Verlag, 2011.

10. M. A Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. "The WEKA data mining software: an update", *ACM SIGKDD Explorations*, Vol. 11. No. 1, pp: 10-18, 2009.

11. Hodge, V., Austin, J. "A survey of outlier detection methodologies", *Artificial Intelligence Review*, Vol. 22. No. 2, pp. 85-126, 2004.

12. Reinartz, T. "A unifying view on instance selection", *Data Mining and Knowledge Discovery*, Vol. 6. pp. 191–210, 2002.

13. Yu, L., and Liu, H. "Efficient feature selection via analysis of relevance and redundancy", *J. Machine Learning Research*, Vol. 5. pp. 1205-1224, 2004.

14. Gupta, G. K. *Introduction to Data Mining with Case Studies*. New Delhi: Prentice Hall India, 2006.

15. Al Aidaroos, K. M., Bakar A. A., and Othaman, Z. (2010) 'Naïve Bayes variants in classification learning', *Proc. Intl. Conf. Information Retrieval & Knowledge Management*, Malaysia, 2010, pp.276 – 281.

16. Han, J., and Kamber, M. *Data Mining Concepts and Techniques*, 2nd Edition, San Francisco, CA: Morgan Kaufmann Publishers, 2006.

17. Shen, D., Chen, Z., Yang, Q., Zeng, H., Zhang, B., Lu, Y., and Ma, W. **"**Web-page classification through summarization"**,** *Proc. 27th Int. ACM SIGIR 04, Conf. Research and Development in Information Retrieval*, New York, ACM Press, 2004, pp.242- 249.

18. Chen,W., Du, Y., Zhang, P., and Han, B. "The effective classification of the Chinese web pages based on KNN" , *J. Computer Information Systems*, Vol. 6, pp: 2925-2932, 2010.

19. Asirvatham, A. P., and Ravi, K. K. "Web page classification based on document structure", *IEEE National Level Student Paper Contest*, 2001.

20. Dai, W et al., "A novel web page categorization algorithm based on block propagation using query-log information", *Advances in Web-Age Information Management Lecture Notes in Computer Science*, Vol. 4016/2006. pp. 435-446, 2006.

21. De Boer, V. Someren, V., and Lupascu. T. "Classifying web pages with visual features". *Proc. 6th Int. Conf Web Information Systems*, Valencia, Spain, 2010.

22. Kan, M.Y. “Web page classification without the web page”, *Proc. 13th Int. Conf on* WWW, 2004, pp.262-263.

23. Kan, M.Y., and Thi, H. O. N. “Fast web page classification using URL features”, *Proc. of the 14th ACM Int. Conf on Information and Knowledge Management*, 2005, pp.325-326.

24. Kwon, O.W., and Lee, J. H. “Web page classification based on k-nearest neighbor approach”, *Proc. of the 5th Intl. Workshop on Information Retrieval with Asian Languages*, 2000, pp.9-15.

25. Kwon, O.W., and Lee, J. H. “Text categorization based on k-nearest neighbor approach for web site classification”, *Information Processing and Management,* Vol. 29, No. 1, pp.25-44, 2003.

26. Xhemali, D., Hinde, C.J., and Stone, R.G. “Embarking on a web information extraction project”, *Proc. of Int. Conf on Computational Intelligence*, 2007.

27. Tsukada, M., and Washio et al, “Automatic web page classification using machine learning methods”, *LNCS: Proc. of First Asia-Pacific Conference on Web Intelligence : Research and Development*, Vol. 2198. 2001, pp.303-313.

28. Zhang, Y. Z., "The automatic classification of web pages based on neural networks", *Proc. of the 2001 Int. Conf on Neural Information Processing*, China, Vol. 2. 2001, pp.570-575.

29. Xhemali, D., Christopher, J., Hinde, and Stone, R. G. 'Naïve Bayes vs. decision trees vs. neural networks in the classification of training web pages', *Int. J. Computer Science Issues*, Vol. 4.No. 1. pp.16-23, 2009.

30. He, Z., and Liu, Z. "A novel approach to naïve bayes web page automatic classification", *Proc. 5th Int. Conf. on Fuzzy Systems and Knowledge Discovery*, Vol. 2. 2008, pp .361-365.

31. Bo, S., Qiurui, S., Zhong, C., and Zengmei, F. "A Study on Automatic Web Pages Categorization", *Proc. of IEEE IACC*, India, 2009, pp. 1423-1427.

32. Balamurugan, S., Pramala, S., Rajalakshmi, B., and Rajaram, R. "Improving decision tree performance by exception handling", *Int. J. Automation and Computing*, Vol. 3. pp. 372 – 380, 2010.

33. Jain A.K., Murty M.N., and Flynn P.J. "Data Clustering: A Review", *ACM Computing Surveys*, Vol.31. No. 3, pp.264-323, 1999.

34. Devi I. M, Rajaram R, Selvakuberan K, “Generating best features for web page classification”, *Webology*, Vol. 5. No. 1, 2008.

35. Han, L. W., and Alhashmi, S. M. “Joint web-feature (JFEAT) (2010): a novel web page classification framework”, *Communications of the IBIMA*, Article ID 73408, 2010.

36. Salamat, A., and Omata, S. “Web page feature selection and Classification using neural networks”, *Information Science*, ACM, Vol. 158. No. 1, pp. 69-88, 2004.

37. Chen, C., Lee, H., and Chang, Y. ‘Two novel feature selection approaches for web page classification”, *Expert Systems with Applications*, Vol. 36. pp. 260-272, 2009.

38. Wakaki, T., Itakura, H., and Tamura M. “Rough set-aided feature selection for web page classification”, *Proc. IEEE/WIC/ACM Int. Conf. on Web Intelligence,* 2004, pp.70-76.

39. Jensen, R., and Shen, Q. “Web page classification with ACO-enhanced fuzzy-rough feature selection”, *LNCS*, Vol. 4259. pp.147-156, 2006.

40. Shen, Q., and Jensen, R. “Rough sets, their extensions and applications”, *Int. J. of Automation and Computing*, Vol. 4. No.1, pp. 100-106, 2007.

41. Peng, X., Ming, Z., and Wang, H. ”Text learning and hierarchical feature selection in web page classification”. *LNCS: Advanced Data Mining and Applications*, Vol. 513. pp. 452 – 459, 2008.

42. Farhoodi, M., Yari, A., and Mahmoudi, M. “A persian web page classifier applying a combination of content-based and context-based features”,. *Int. J. Information Studies,* Vol. 1. No. 4, pp. 263 – 271, 2009.

43. Ozel, S. A. “A genetic algorithm based optimal feature selection for web page classification”. *Proc. of Intl Symposium on Innovations in Intelligent Systems and Applications,* 2009, pp. 282 – 286.

44. Hall, M. “Correlation based feature selection for machine learning”, Ph. D Thesis, Dept. Computer Science, University of Waikato, New Zealand, 1999.

45. Yu, L., and Liu, H. “Feature selection for high-dimensional data: a fast correlation based filter solution”, *Proc. 20th Intl Conference on Machine Learning (ICML 2003),* Washington, 2003.

46. Michalak, K., and Kwasnicka, H. “Correlation-based feature selection strategy in classification problems”, *Int. J. of Applied Mathematical Computational Science*, Vol.16. No.4, pp. 503-511, 2006.

47. Sugumaran, V., Muralidharan, V., and Ramachandran, K. I. “Feature selection using decision tree and classification through proximal support vector machine for fault diagnosis of roller bearing”, *Mechanical Systems and Signal Processing*, Elsevier, Vol. 21, pp. 930 – 942, 2007.

48. Kohavi, R., and John, G.H, “Wrappers for feature subset selection”, *Artificial Intelligence*, Vol. 97. No.2, pp. 273 – 323, 1997.

49. Balamurugan, S., and Rajaram, R. ”Effective and efficient feature selection for large-scale data using baye’s theorem”, *Int. J. of Automation and Computing*, Vol. 6. No. 1, pp. 62-71, 2009.

50. Zhang, S., and Zhou, Q. “New feature subset selection algorithm using class association rules mining”, *Key Engineering Materials*, Vol. 474-476, pp. 622 – 625, 2011.

51. Wang, G., and Song, Q. “Selecting feature subset via constraint association rules”, *Advances in Knowledge Discovery and Data Mining*, LNCS, Vol. 7302, pp. 304-321, 2012.

52. Shahzad, W., Asad, S., and Khan, M. A. “Feature subset selection using association rule mining and JRIP classifier”, *International Journal of Physical Sciences*, Vol. 8. No. 18, pp. 885-896, 2013.

53. Guangtao Wang, G., Song, Q., Xu, B., and Zhou, Y. “Selecting feature subset for high dimensional data using the propositional foil rules”, *Pattern Recognition*, Vol. 46. pp.199-214, 2013.

54. H. Liu, H., Hussain, F., Tan, C.L. and Dash, M. “Discretization: an enabling technique”, *Data Mining and Knowledge Discovery*, Springer Verlag, Vol. 6. No. 4, pp. 393-423, 2002.

55. Holte, R.C. “Very simple classification rules perform well on most commonly used data sets”, *Machine Learning*, Vol. 11. pp.63 – 91, 1993.

56. Dougherty, J., Kohavi, R. and Sahami, M. “Supervised and unsupervised discretization of continuous features”, *Proc. of the 12th Int. Conf. on Machine Learning*, 1995, pp. 194—202.

57. Elomaa, T. and Rousu, J.”Efficient multi splitting revisited: optima preserving elimination of partition candidates”, *Data Mining and Knowledge Discovery*, Vol. 8. No.2, pp. 97 – 126, 2004.

58. Kotsiantis, S. and Kanellopoulos, D. “Discretization techniques: a recent survey”, *International Transactions on Computer Science and Engineering*, Vol. 32. No.1. pp.47-58, 2006.

59. Kurgan, L.A., and Cios, K.J. “CAIM discretization algorithm”, *Knowledge and Data Engineering*, Vol.16. No. 2. pp. 145–153, 2004.

60. Marzuki, Z., and Ahmad, F.”Data mining discretization methods and performances”, *Proc. of* ICEEI 2007, 2007.

61. Kerber, R. “Chimerge : discretization of numeric attributes”, *Proc. of the National Conference on Artificial Intelligence American Association*, 1992, pp.123-128.

62. Fayyad, U.M., and Irani, K.B. “Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning”, *Proc. 13th Intl. Conf. on Artificial Intelligence*. 1993, pp. 1022–1029.

63. Liu, H., and Setiono, R. “Feature selection via discretization”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9. No. 4, pp.642-645, 1997.

64. Senthil Kumar, J., Manjula, D., and Krishnamoorthy, R. “NANO: A new supervised algorithm for feature selection with discretization”, *Proc. of the IEEE International Advance Computing Conference (IACC 2009), 2009.*

65. Hacibeyoglu, M., Arslan, A., and Kahramanli, S. “Improving classification accuracy with discretization on data sets including continuous valued features”, *Proc. of WASET,* 2011.

66. Yang, Y., and Webb, G. I. "A Comparative study of discretization methods for naïve bayes classifiers", *Proc. of the Pacific Rim Knowledge Acquisition Workshop*, 2002, pp.159-173.

67. Faith Kaya, "Discretizing Continuous Features for Naïve Bayes and C4.5 Classifiers", Dept. of Computer Science, University of Maryland, College Park, [Online] Available : http://www.cs.umd.edu/Grad/scholarlypapers/papers/fatih-kaya.pdf

68. Hsu, C.N., Huang, H. J., and Wong, T.T. "Why discretization works for naïve bayesian classifiers", *Proc. of the 17th Intl Conf. on Machine Learning*, 2000, pp.309 – 406.

69. Lu, J., Yang, Y., and Webb, G., I. "Incremental discretization for naïve bayes classifier", *Proc. 2nd Intl Conf on Advances in Data Mining and Applications*, 2006, pp.223 – 238.

70. Joita, D. "Unsupervised static discretization methods in data mining", Faculty of Science and Technology Information, Titu Maiorescu University, Romania, Megabyte, December 2008.

71. Qi, X., and Davison, B., D. "Web page classification: features and algorithms", *ACM Computing Surveys*, Vol. 41, No.2, Article No. 12, Feb 2009.

72. Kotsiantis, S. B. "Supervised machine learning: a review of classification techniques", *Informatica*, Vol.31. pp. 249-268, 2007.

73. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., and et al. “Top 10 algorithms in data mining”, *Knowledge Information Systems*, Vol. 14. pp. 1-37, 2008.

74. Cunningham, P., and Delany, S. J. (2007), ‘K-nearest neighbor classifiers’, University College, Dublin, Ireland, Dublin Technical Report UCD-CSI-2007-4, 2007.

75. Jiang, L., Cai, Z., and Wang, D., and Jiang, S. “Survey of Improving k-Nearest Neighbor for Classification”, *Fuzzy Systems and Knowledge Discovery*, pp. 679-683, 2007.

76. Bache, K., and Lichman, M. (2013). “UCI Machine Learning Repository”, Irvine, CA: University of California, School of Information and Computer Science, Available: http://archive.ics.uci.edu/ml.

77. Suguna, N., and Thanushkodi, K. “An Improved k-Nearest Neighbor Classification Using Genetic Algorithm”, *Intl. J. of Computer Science Issues*, Vol.7. No.4. pp.18-21. 2010.

78. Parvin, H., Alizadeh, H., and Minaei-Bidgoli, B. “MKNN: Modified k-Nearest Neighbor”, *Proc. of World Congress on Engineering and Computer Science*, 2008.

79. Muflikhah, L., and Adnyana, M. P. “Classifying categorical data using modified k-nearest neighbor weighted by association rules”, *Proc. of the Intl Conf. on Future Information Technology*, Vol.13. pp. 347-351, 2013.

80. Chen, C., and Lee. G. ”Image segmentation using multiresolution wavelet analysis and expectation maximization (em) algorithm for digital mammography”. *Intl. J. of Imaging Systems and Technology*, Vol. 8. No.5, pp.491–504, 1997.

81. Wang, T., and Karayiannis. N. “Detection of microcalcification in digital mammograms using wavelets”. *IEEE Transactions on Medical Imaging*, Vol.17. No.4, pp. 498–509, 1998.

82. Li, H. et al. “Fractal modeling and segmentation for the enhancement of microcalcifications in digital mammograms”. *IEEE Transactions on Medical Imaging*, Vol.16. No. 6, pp.785– 798, 1997.

83. Chan, H. et al. “Computerized analysis of mammographic microcalcifications in morphological and feature spaces”, *Medical Physics*, Vol. 25. No. 10, pp. 2007–2019, 1998.

84. Brazokovic, D., and Neskovic, M. “Mammogram screening using multiresolution-based image segmentation”, *Intl. J. of Pattern Recognition and Artificial Intelligence*, Vol.7. No.6, pp.437–1460, 1993.

85. Li, H., et al. “Marcov random field for tumor detection in digital mammography”, *IEEE Trans. Medical Imaging*, Vol. 14. No. 3, pp.565–576, 1995.

86. Antonie, M., Osmar, R., Jane, Z., and Coman, A. “Application of data mining techniques for medical image classification”, *Proc. of the 2nd Intl. Workshop on Multimedia Data Mining, with ACM SIGKDD conference*, 2001,pp.94 – 101.

87. Abraham, R., Jay, B., Simha and Iyengar, S.S. “Effective discretization and hybrid feature selection using naïve bayesian classifier for medical data mining”, *Intl J. of Computational Intelligence Research*, Vol.5. No.2, pp. 116 – 129, 2009.

88. Lustgarten, J., L, Gopalakrishnan, V., Grover, H., and Visweswaran, S. “Improving Classification Performance with discretization on Biomedical Data Sets”, *Proc. of AMIA Annual Symposium*.2008, pp.445 – 449.

89. Zhang, H., Ling, C. X., and Zhao, Z. “The learnability of Naïve Bayes”, *Proc. of the Canadian Artificial Intelligence Conference*, 2000, pp. 432–441.

90. Ribeiro, M.X., Traina, A.J.M., Traina, C., and Azevedo-Marques P.M. “An Association Rule-Based Method to support Medical Image Diagnosis with efficiency”, *IEEE transactions on Multimedia*, Vol.10. No. 2, pp. 277-285, 2008.

91. Zare, M. R., Seng, W. C., and Mueen, A. “Automatic Classification of Medical X-ray images”, *Malaysian Journal of Computer Science*, Vol.26. No.1, pp.9 – 22, 2013.

92. Hota, H.S., Shukla, S.P., and Kiran**,** G.K. (2013), **"**Review of Intelligent Techniques Applied for Classification and Preprocessing of Medical Image Data", *Intl J of Computer Science Issues,* Vol.10. No.1, pp. 267-272, 2013.

93. Deepa, S. N., and Devi, B.A. "A survey on artificial intelligence approaches for medical image classification'", *Indian Journal of Science and Technology*, Vol. 4. No.11, pp.1583-1595, 2011.

94. Rajendran, P., and Madheswaran, M. "Hybrid Medical Image Classification Using Association Rule Mining with Decision Tree Algorithm", *Journal of Computing,* Vol. 2. No.1.pp. 127-136, 2010.

95. Le, T., Tran, D., Ma, W., and Sharma, D. "A New Support Vector Machine Method for Medical Image Classification", *EUVIP 2010*, 2010, pp. 165-170.

96. Vanitha, L., and Venmathi.A.R. "Classification of Medical Images Using Support Vector Machine", *Proc. of the Int. Conf on Information and Network Technology,* 2011, Vol.4. pp. 63-67.

97. Gnanasekar, P., et al. "Investigation on Feature Extraction and Classification of Medical Images", *WASET*, 2011, pp.327 – 332.

98. Nayak J., Bhat P. S., Acharya R., Lim., and Kagathi M. "Automated Identification of Diabetic Retinopathy Stages Using Digital Fundus Images", *J. of Medical Systems,* Vol.

32, No. 2, pp. 107 – 115, 2008.

99. Frankes, W.B., and Baeza_Yates, R. *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

100. Nayak J., Bhat P. S., Acharya R., Lim., and Kagathi M. "Automated Identification of Diabetic Retinopathy Stages Using Digital Fundus Images", *J. of Medical Systems*, Vol. 32, No. 2, pp. 107 – 115, 2008.

101. Mangai, J. A., and Kumar, S. K. "A Novel Approach for Web Page Classification using Optimum features', *Intl J of Computer Science and Network Security*, Vol. 11. No.5.pp. 252 – 257, 2011.

102. Mangai, J. A., Kumar, S. K. and Balamurugan, S. "A Novel Feature Selection Framework for Automatic Web Page Classification", *Intl J of Automation and Computing*, Springer Verlag, Vol. 9. No.4, pp.442 – 448, 2012.

103. Ward, J. H. "Hierarchical grouping to optimize an objective function*", J of the American Statistical Association*, Vol. 58. No.301, pp. 236 – 244, 1963.

104. Soman, K.P., Diwakar, S., and Ajay, V. *Insight into Data Mining*, Prentice Hall, 2006.

105. Mangai, J. A., Kumar, V. S., and Kothari, D.S. "A Supervised Discretization Algorithm for Web page Classification", *Proc. of the 8th Intl Conf. on Innovations in Information Technology,* 2012, pp. 226 – 231.

106. Mangai, J. A., Kumar, V. S., and Balamurugan, S. "A Novel Approach for Effective Web Page Classification", *Intl J of Data Mining, Modelling and Management*, Inderscience Publishers, Vol.5. No.3, pp. 233-245, 2013.

107. Mangai, J. A., Kumar, V. S., Wagle, S. M. "A Novel Web Page Classification Model using an Improved k Nearest Neighbor Algorithm", *Proc. of the 3rd Int. Conf. on Intelligent Computational Systems (ICICS 2013),* Singapore, 2013, pp. 49 – 53.

108. Mangai, J. A., Kumar, V. S., and Nayak, J. "A Novel Approach for Classifying Medical Images Using Data Mining Techniques", *Proc. of the 2nd International Conference on Data Mining and Its Applications (ICDMA'13),* 2013, pp.41 – 45.

109. Fawcett, T. "An Introduction to ROC Analysis", Pattern Recognition Letters, Vol.27. pp. 861-874, 2006.

110. Slaby, A. "ROC Analysis with Matlab", *Proc. of the 29th Intl. Conf. on Information Technology Interfaces*, 2007, pp.191-196.

111. Yang, Y., and Pedersen, J. O. "A Comparative Study on Feature Selection in Text Categorization", *Proc. of the Int. Conf. on Machine Learning,* 1997, pp. 412-420.

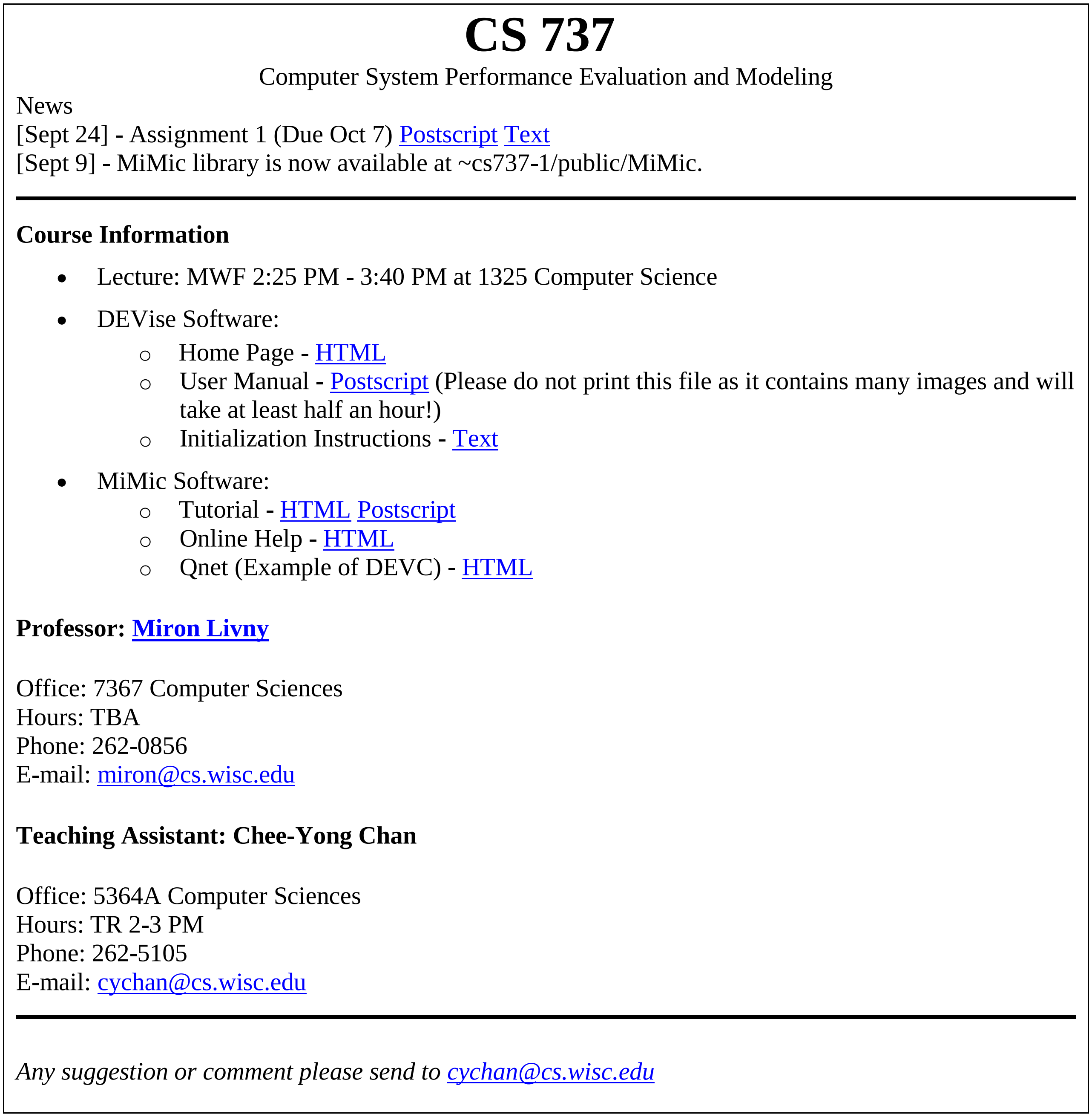
112. The 4 Universities Data set [Online] Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

113. Nayak J, “Automated Detection of Eye Abnormalities and Patient Data Handling”, Ph. D Thesis, Dept. Electronics and Communication Engg, NIT, Surathkal, 2008.

APPENDICES

APPENDIX A: Sample Web Pages of Each Category

1. Course Category



CS 737

Computer System Performance Evaluation and Modeling

News
[Sept 24] - Assignment 1 (Due Oct 7) Postscript Text
[Sept 9] - MiMic library is now available at ~cs737-1/public/MiMic.

Course Information

- Lecture: MWF 2:25 PM - 3:40 PM at 1325 Computer Science
- DEVise Software:
  - Home Page - HTML
  - User Manual - Postscript (Please do not print this file as it contains many images and will take at least half an hour!)
  - Initialization Instructions - Text
- MiMic Software:
  - Tutorial - HTML Postscript
  - Online Help - HTML
  - Qnet (Example of DEVC) - HTML

Professor: Miron Livny

Office: 7367 Computer Sciences
Hours: TBA
Phone: 262-0856
E-mail: miron@cs.wisc.edu

Teaching Assistant: Chee-Yong Chan

Office: 5364A Computer Sciences
Hours: TR 2-3 PM
Phone: 262-5105
E-mail: cychan@cs.wisc.edu

Any suggestion or comment please send to cychan@cs.wisc.edu

2. Student Category

Al Carruth
Please send mail to *carruth@cs.utexas.edu* if you have any questions or suggestions.

Introduction

I am a Ph.D. candidate at UT-Austin in the Department of Computer Sciences. My supervising professor is Jayadev Misra and my dissertation topic is *Real-Time UNITY.* I am a member of Professor Misra's PSP research group.

I am extending the UNITY theory in order to express finite time bounds on the usual UNITY operators for progress and safety. I am also interested in functional programming languages, partial order semantics and automated theorem proving.

Contact Information

□ my personal home page
□ Office address: UA-9 4.116G
□ Office phone: 512-471-9764
□ Home phone: 512-302-3276
□ Email address: *carruth@cs.utexas.edu*

U.S. mail:

Al Carruth
Department of Computer Sciences
Taylor Hall 2.124
University of Texas at Austin
Austin TX 78712-1188

Links to other World Wide Web pages

3. Faculty Category:

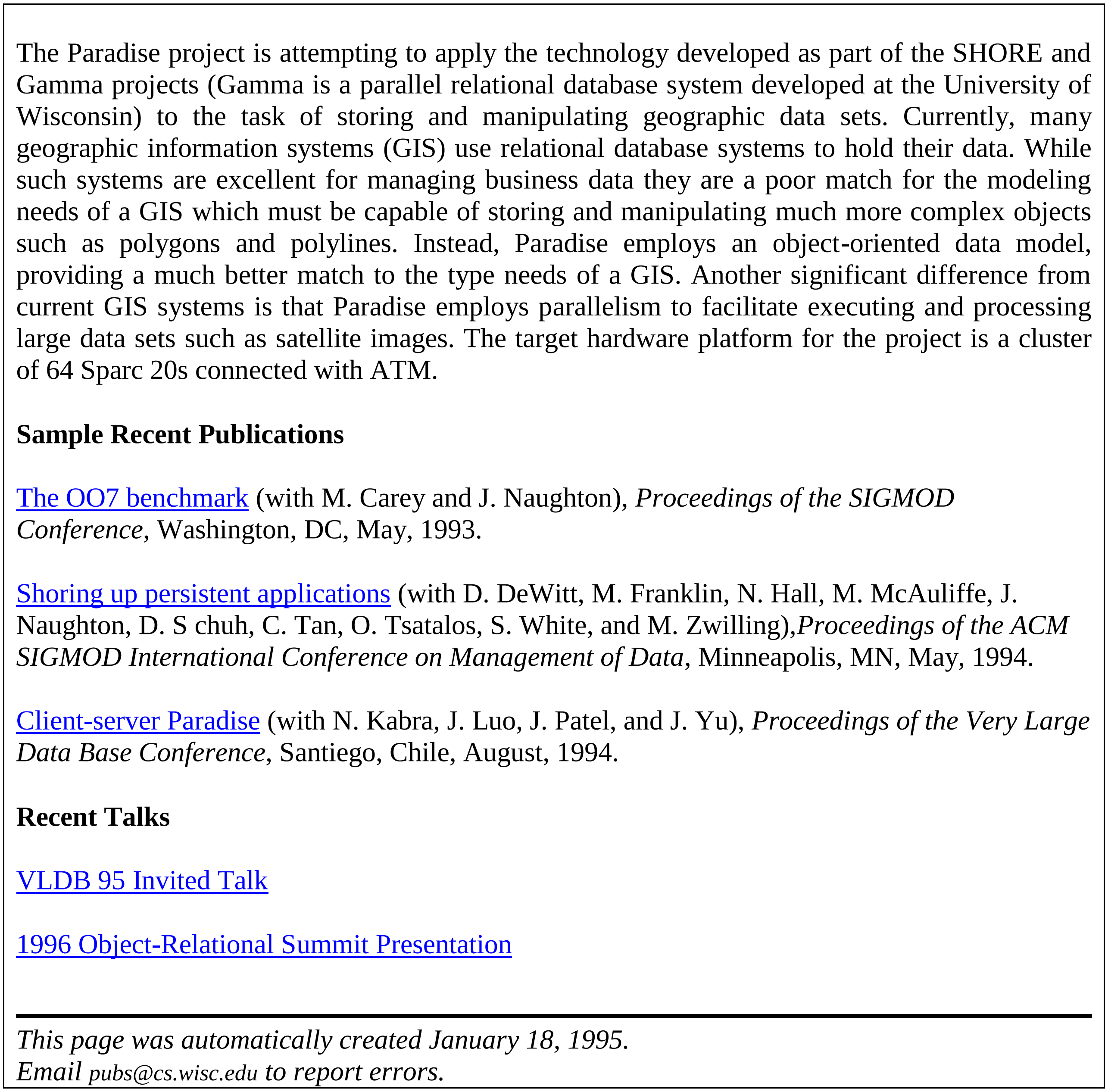
David J. DeWitt
Professor and Romnes Fellow

Computer Sciences Department
University of Wisconsin
1210 W. Dayton St.
Madison, WI 53706-1685

Telephone: (608) 262-1204
Fax: (608) 262-9777
Email: dewitt@cs.wisc.edu
Ph.D., University of Michigan, 1976
Interests: Object oriented database systems, parallel database systems, database benchmarking, geographic information systems

Research Summary

My two main research projects are SHORE and Paradise. The objective of SHORE is to design, implement, and evaluate a persistent object system that will serve the needs of a wide variety of target applications including hardware and software CAD systems, persistent programming languages, geographic information systems, satellite data repositories, and multimedia applications. SHORE expands on the basic capabilities of the widely-used Exodus Storage Manager (developed at Wisconsin, funded by ARPA) in a number of ways including support for typed objects, multiple programming languages, a `Unix-like' hierarchical name space for named objects, and a Unix-compatible interface to objects with a `text' field. This interface is intended to ease the transition of applications from the Unix file system environment to SHORE as existing Unix tools such as vi and cc will be able to store their data in SHORE objects without modification (basically a Unix file becomes either a single SHORE object or the text field of a complex object). SHORE is being targeted at a wide range of hardware environments, scaling all the way from individual workstations to heterogeneous client/server networks to large multiprocessors such as the Intel Paragon. SHORE is a joint project with Profs. Carey, Naughton, and Solomon.



The Paradise project is attempting to apply the technology developed as part of the SHORE and Gamma projects (Gamma is a parallel relational database system developed at the University of Wisconsin) to the task of storing and manipulating geographic data sets. Currently, many geographic information systems (GIS) use relational database systems to hold their data. While such systems are excellent for managing business data they are a poor match for the modeling needs of a GIS which must be capable of storing and manipulating much more complex objects such as polygons and polylines. Instead, Paradise employs an object-oriented data model, providing a much better match to the type needs of a GIS. Another significant difference from current GIS systems is that Paradise employs parallelism to facilitate executing and processing large data sets such as satellite images. The target hardware platform for the project is a cluster of 64 Sparc 20s connected with ATM.

Sample Recent Publications

The OO7 benchmark (with M. Carey and J. Naughton), *Proceedings of the SIGMOD Conference*, Washington, DC, May, 1993.

Shoring up persistent applications (with D. DeWitt, M. Franklin, N. Hall, M. McAuliffe, J. Naughton, D. S chuh, C. Tan, O. Tsatalos, S. White, and M. Zwilling),*Proceedings of the ACM SIGMOD International Conference on Management of Data*, Minneapolis, MN, May, 1994.

Client-server Paradise (with N. Kabra, J. Luo, J. Patel, and J. Yu), *Proceedings of the Very Large Data Base Conference*, Santiego, Chile, August, 1994.

Recent Talks

VLDB 95 Invited Talk

1996 Object-Relational Summit Presentation

This page was automatically created January 18, 1995.
Email pubs@cs.wisc.edu *to report errors.*

APPENDIX B: BRIEF DESCRIPTION OF THE IMPLEMENTATION OF THE PRESENT ALGORITHMS

1. Brief description of Feature Extraction for WPC implemented in JAV A

Function	Description
IFSelMain	Initialize webkb directory using webkb. Reads the category names and the count of each category from the command line. Calls **getTermFreqInvDocFreqTreeMap** to transform each web page into vector space model. Calls **writeArffFile** to store this in ARFF format for easy processing in WEKA.
Webkb	Initialize internal data structures to hold category-wise list of the web pages
getTermFreqInvDocFreqTreeMap	Compute the Term Frequency Inverse Doc Frequency for all the web pages. i) For each category getWebPageTermFreqTreeMapArray is called to get the map (keyword, freq) for all the webpages belonging to the category. ii) Computes the cummulativeTermFreq of all the webpages in the given category.
getWebPageTermFreqTreeMapArray	Obtain the Term Freq for all the webpages belonging to a category. This is done by calling these functions repeatedly processWebPage(),convertToPlainText(), stopWordRemove for each of the webpages. Finally a map (keyword, freq) for all the webpages in the category is returned.
processWebPage	Obtain the Term Freq for a given web page

Function	Description
convertToPlainText	Remove the HTML Tags in a given webpage and create a plain text string
stopWordRemove	For a given webpage, it tokenizes the plain text string, for each token (i,e keyword) it check if it's a a stop word, applies stemming algorithm, and accumulate the freq of the word in a map (keyword, freq). Finally it returns a map (keyword, freq) for all the keywords encountered in the given webpage.
Tfid	Constructs the tfid object to hold the final TermFreqInvDocFreq of all webpages.
Compute	Computes the tfid for the entire collection of webpages as follows. i) Computes the termFreq in a double dimensional array indexed by (webpage_id, term_id). ii) Computes the documentFreq() of a given term. ii) Returns a double dimentional array indexed by (webpage_id, term_id) which represent the tfid weights.
writeArffFile	The tfifd obtained in the previous step is written to an ARFF file, for easy processing with WEKA.
ARFF	This creates the Arff file.
wrtieHeader	This outputs the @attribute <term> numeric.
writeData	This generates the sparse representation each webpage and corresponding non-zero term in the webpage and writes into the ARFF file.
closeArff	This closes the ARFF file.

2. Brief description of Feature Extraction for MIC implemented in MATLAB

Function	**Description**
Gamain	Reads all images of a category of interest stored in the home directory, counts the number of images. For each image it does the following • Resizes the image to uniform length using imresize function. • Calls **blkproc** function which extracts the statistical features from the sub images of size 36 x 90. The statistical parameters are extracted from each block using the functions **my_mean**, **my_var**, **my_skew** and **my_kurt**. • Calls the function **arffwrite** to write the features extracted into an output file which is in attribute relation file format.
arffwrite	• Reads the file name for writing data, relation name for the arff file, attribute name for each variable, data type for each attribute and the input data for writing arff formatted. Creates an output file in arff
my_var	• Computes the variance of the given image block
my_mean	• Computes the mean of the given image block
my_skew	• Computes the skewness of the given image block
my_kurt	• Computes the kurtosis of the given image block

3. Brief description of the hybrid model of Feature Selection implemented using Weka

Function	Description
Weka.attributeSelection. CfsSubsetEval	Evaluates the worth of the subset of input attributes based on the individual predictive ability of each feature with the degree of redundancy between them. The features selected are the inputs to the J48 tree growing algorithm.
Weka.classifiers.trees.J48	Creates the pruned decision tree for the input data set. The features in the pruned tree are the finally selected features.

4. Brief description of the Ward's Feature Selection Framework implemented in VB

Function	Description
Main	• Reads the input file with the pre-processed web pages. Each web page is represented using the same number of features. • Prints the clusters of redundant features by calling the functions **Process**, **PrintClusters** and **Voting** in this order.
Process	For each adjacent pair of rows in the input file, it calls **Mini** to find the merger(s) with minimum variance.
Mini	Finds all possible mergers of features in the two rows. Calculates and returns to **Process** the mean and variance of each possible merger
PrintClusters	Prints each possible merger and its variance
Voting	Calculates the vote of each merger with minimum variance. Finds and prints the merger with the highest vote.

5. Brief description of the discretization algorithm implemented in JAV A

Function	Description
Main	• Reads the input file with continuous features. • Discretizes each feature in the input file, by calling function **Sort** and **Cu**t in this order. • Creates an output file with the discretized intervals of each feature.
Sort	• Arranges the values of the feature being discretized, into ascending order along with their class labels.
Cut	• Accepts the sorted values of the feature • Establishes the cut points C_1, C_2 and C_3 in this order on the feature values by a series of splits and merges. • Replaces the final set of cut-points C_3 by automatically generated discrete labels starting from 1, 2 and so on.

6. Brief description of the MKNN algorithm implemented in JAVA

Function	Description
Main	Reads the value of k, the number of web page/image categories, the names of the categories, min_sup and min_conf threshold. Reads a user choice to model the classifier and for ROC calculation. Reads a training file with numeric attributes and its corresponding file with discrete attributes. Divides the training file with numeric attributes using 70 – 30% split and uses the 30% as test file. Calls the other functions in this order.

Function	Description
ClassAttribute	Isolates the class attribute from each training example.
RuleSetGen	Generates the rule set with every attribute value – class combination using the training file with discrete attributes.
FeatureWeighting	Calculates the support and confidence of each rule. Finds the weight of each feature using its max_support and max_confidence and the input threshold.
MKNN	Finds the k nearest neighbors to each of the test example in the test file using a feature weighted distance measure. Applies distance weighted voting on the k nearest neighbors to predict the class of each test example. Calculates the confusion matrix and the classification accuracy of the classifier.
ROC	Randomizes the training file to create 10 input training and test files. Calculates the TPR and FPR of MKNN for each of these files. These TPR and FPR values are written to an output file to generate the ROC graph in Excel.

7. Brief Description of the Area under the ROC curve implemented in MATLAB

Function	Description
Areaundercurve	Reads the TPR and FPR values. Calculates the area using the Trapezoidal rule.

8. Brief Description of the PWPC/PMIC implemented in VB

Function	Description
Main	Reads the training file, test file with their names and the output file name. Finds the number of distinct class labels in the training file and the class labels also. Splits the training file into subsets based on the class label. Calls **Influence** on each class-split which returns the most predictive attribute value in this split. Repeats the following for each test record • Finds the AVS measure of this test with every training record. • Partition the training records/web pages/images into descending order of their *AVS* measure. An *AVS* partition has web page(s) that have same *AVS* value. • Calls **Classify** with the AVS partitions, which returns the predicted class of the test record.
Influence	Takes a class partition. Find the predictive power of each attribute value in it using the Bayes theorem. Returns the most predictive attribute value (s).
Classify	Takes the highest AVS partition. Finds the sum of the predictive power of the influencing attribute values in each web page/image in this partition. Finds the web page(s)/image(s) with highest *PP* i.e., with more influencing attribute values. Predicts the class $y'$of the test record directly by majority voting on the class of the training records in this AVS partition. If there is an equal class probability distribution then proceeds with the next AVS partition.

9. Brief Description of the usage of WEKA classes

Each of the following weka classes are invoked as on the arff_file (generated from IFSel or Ga) java –Xmx1024m –cp weka.jar <weka_procedure> <arff_file>	
Weka _procedure	**Description**
weka.attributeSelection.CfsSubsetEval	Invoke CFS Subset Evaluation on the input arff_f
"weka.filters.unsupervised.attribute.Remove	Remove the selected attributes attributes from the input arff_file
"weka.filters.unsupervised.instance.Randomize	Randomizes the arff_file.
"weka.classifiers.trees.J48	Invoke the J48 classifier on the arff_file and
weka.filters.unsupervised.attribute.Discretize	WEKA class for Simple Binning based discretization
weka.classifiers.rules.OneR	WEKA class for rule based classifier called OneR.
weka.classifiers.lazy.KStar	WEKA class for K Nearest Neighbor classifier called KStar.
weka.classifiers.trees.Id3	WEKA class for Id3 classifier
weka.classifiers.bayes.NaiveBayes	WEKA class for the Naïve Bayes classifier

10. Brief Description of the custom built java programs and their respective Functionality

Custom built java programs to perform the respective Funcionality java –jar <java_program> <arff_file>	
Java Program	**Description**
Rcndrc.jar	Remove duplicates and conflicting tuples in the input file.
S2N_Arff.jar	Converts the input Sparse Arff to Normal Arff
N2S_Arff.jar	Converts the Normal Arff to Sparse Arff

APPENDIX C: SYSTEM CONFIGURATION

The experimental analysis of the present work in this thesis was conducted in the computer programming lab of BITS Pilani, Dubai Campus. A detail of the system configuration used in the study is as follows:

System: Microsoft Windows XP Professional Version 2002, Service Pack 3, v.3311

Computer:

- Intel ®, Core ™ 2 CPU
- 4400 @ 2.00 Ghz
- 2.00 Ghz, 0.99 GB of RAM

Software used:

- WEKA 3.6 , an open source data mining tool
- Java version 1.6
- Visual Basic 2008.
- MATLAB V7.7 (R2008b)

APPENDIX D: THE 70-30 FILE AFTER FEATURE SELECTION BY CFS IN SPARSE ARFF FOR BINARY CLASS WPC

```
@relation 'webkb-weka.filters.unsupervised.attribute.Remove-V-
R105,150,189,232,450,544,558,691,710,1019,1052,1058,1116,1128,1203,1217,1262,1385,1507,1
510,1666,1693,1704,1826,1864,1918,2084,2352,2353,2466,2592,2774-
weka.filters.unsupervised.instance.Randomize-S42'

@attribute announc numeric
@attribute assign numeric
@attribute babylon numeric
@attribute berkelei numeric
@attribute comment numeric
@attribute cours numeric
@attribute cse numeric
@attribute document numeric
@attribute due numeric
@attribute graduat numeric
@attribute hall numeric
@attribute handout numeric
@attribute homework numeric
@attribute hour numeric
@attribute inform numeric
@attribute instructor numeric
```

```
@attribute ithaca numeric
@attribute lectur numeric
@attribute master numeric
@attribute materi numeric
@attribute network numeric
@attribute note numeric
@attribute ny numeric
@attribute ph numeric
@attribute pm numeric
@attribute pretti numeric
@attribute resum numeric
@attribute structur numeric
@attribute student numeric
@attribute thesi numeric
@attribute univers numeric
@attribute class {course,student}

@data

{14 0.32,16 0.73,18 1.05,20 0.86,26 1.05,31 student}
{1 0.4,5 0.2,6 0.73,7 0.73,8 0.73,9 0.89,13 0.41,15 0.68,18 1.05,28 0.55}
{9 0.89,10 0.73,16 0.73,22 0.8,28 0.45,29 1.7,30 0.33,31 student}
{0 0.7,1 0.4,5 0.17,6 0.73,12 0.45,13 0.46,15 0.61}
```

```
{18 1.05,20 0.86,26 1.05,31 student}
{6 0.81,14 0.32,30 0.3}
{10 0.73,23 1.36,26 1.05,28 0.45,29 1.7,30 0.3,31 student}
{6 0.75,11 0.62,12 0.5,17 0.52}
{1 0.4,11 0.62,13 0.46,14 0.38,17 0.45,21 0.46}
{1 0.5,5 0.21,6 0.79,7 0.57,10 0.73,13 0.46,14 0.32,21 0.57}
{0 0.7,5 0.2,14 0.32,19 0.54,30 0.3}
{1 0.4,4 0.86,5 0.22,7 0.57,8 0.94,12 0.59,15 0.61,17 0.45,22 0.8,24 0.66,28 0.5,30 0.37}
{1 0.5,5 0.17,6 0.7,12 0.55,13 0.46,15 0.61,21 0.46,27 0.77}
{31 student}
{1 0.4,11 0.62,13 0.46,14 0.38,17 0.45,21 0.46}
{10 0.73,16 0.73,22 0.8,28 0.45,30 0.35,31 student}
{0 0.78,1 0.52,4 0.77,5 0.23,8 0.73,11 0.75,12 0.5,13 0.52,15 0.76,17 0.62,19 0.63,21 0.54,23
1.36,24 0.66,28 0.55,30 0.37}
{1 0.4,4 0.77,5 0.2,6 0.75,7 0.57,12 0.45,13 0.48,14 0.38,15 0.61,17 0.5,30 0.3}
{1 0.4,4 0.77,5 0.2,6 0.81,7 0.67,12 0.45,13 0.41,14 0.39,17 0.45,30 0.33}
{1 0.53,5 0.2,6 0.81,7 0.57,8 0.89,12 0.45,13 0.46,14 0.32,15 0.61,17 0.45,21 0.63,24 0.77,30 0.3}
{1 0.54,5 0.21,8 0.93,10 0.87,11 0.7,13 0.46,14 0.36,15 0.61,17 0.5,19 0.54,28 0.5}
{0 0.78,5 0.17,11 0.62,12 0.58,15 0.61,17 0.45,21 0.51}
{0 0.82,1 0.4,4 0.77,5 0.17,6 0.82,7 0.64,13 0.46,14 0.36,15 0.73,17 0.54,21 0.46,24 0.59,30 0.3}
{5 0.17,9 0.99,14 0.36,16 0.81,18 1.05,20 0.86,22 0.89,28 0.45,30 0.3,31 student}
{0 0.7,5 0.2,19 0.63,21 0.46,27 0.77,30 0.3}
```

{1 0.55,5 0.19,8 0.94,11 0.75,12 0.62,13 0.46,14 0.36,17 0.57,19 0.54,21 0.51,24 0.59,27 0.77,28 0.52,30 0.3}

{25 1.7,26 1.05,30 0.3,31 student}

{9 0.89,10 0.73,16 0.73,22 0.8,28 0.45,30 0.3,31 student}

{5 0.17,12 0.45,14 0.32,17 0.62}

{0 0.78,1 0.45,4 0.77,5 0.19,11 0.7,12 0.62,13 0.51,14 0.36,15 0.61,21 0.46,24 0.75,27 0.86}

{4 0.77,5 0.21,13 0.41,14 0.38,28 0.5}

{6 0.77,7 0.64,12 0.45,13 0.41,14 0.36,15 0.61,17 0.45,27 0.77,30 0.3}

{4 0.77,10 0.73,16 0.73,22 0.8,23 1.16,28 0.45,30 0.33,31 student}

{16 0.73,22 0.8,28 0.45,30 0.33,31 student}

{14 0.32,16 0.81,20 0.96,22 0.89,26 1.05,28 0.45,30 0.33,31 student}

{16 0.73,18 1.05,22 0.8,28 0.45,31 student}

{23 1.16,28 0.45,30 0.3,31 student}

{5 0.17,16 0.73,20 0.96,22 0.8,26 1.05,30 0.3,31 student}

{5 0.17,10 0.73,16 0.81,20 0.86,23 1.16,28 0.45,30 0.35,31 student}

{1 0.47,5 0.23,7 0.57,8 0.87,9 0.89,10 0.73,11 0.62,12 0.45,13 0.48,14 0.4,17 0.52,19 0.68,21 0.55,24 0.59,28 0.54}

{4 0.86,7 0.57,19 0.6,21 0.46,28 0.5,30 0.3}

{5 0.2,9 0.89,10 0.73,14 0.32,18 1.17,20 0.86,25 1.7,28 0.5,30 0.3,31 student}

{0 0.7,5 0.2,8 0.73,12 0.54,13 0.46,14 0.32,17 0.45,21 0.54}

{5 0.19,15 0.61,17 0.52,21 0.51}

{0 0.7,1 0.4,6 0.73,7 0.64,14 0.36,24 0.59}

{0 0.7,1 0.5,5 0.17,6 0.73,10 0.81,13 0.46,14 0.4,15 0.61,27 0.86,28 0.45}

```
{5 0.17,6 0.73,12 0.45,14 0.32,21 0.46,27 0.77}

{1 0.4,5 0.17,6 0.7,11 0.62,12 0.45,19 0.54,24 0.59}

{0 0.78,1 0.53,4 0.77,5 0.21,11 0.81,13 0.52,15 0.61,17 0.66,21 0.54,24 0.74,27 0.95,30 0.3}

{1 0.4,4 0.77,5 0.19,6 0.7,7 0.67,8 0.94,11 0.78,12 0.59,19 0.54,24 0.59}

{7 0.64,17 0.45,28 0.5}

{5 0.19,16 0.81,18 1.05,20 0.86,22 0.8,28 0.45,30 0.3,31 student}

{5 0.19,11 0.7,12 0.52,13 0.41,21 0.46}

{4 0.77,5 0.17,6 0.79,14 0.32,30 0.33}

{5 0.17,8 0.93,11 0.75,12 0.58,13 0.46,14 0.32,15 0.61,17 0.63,21 0.63}

{5 0.17,8 0.89,12 0.54,14 0.32,28 0.45}

{1 0.51,5 0.17,6 0.7,11 0.62,12 0.45,13 0.46,15 0.61,17 0.45,24 0.59,27 0.77,30 0.3}

{0 0.78,1 0.4,5 0.21,7 0.64,12 0.45,14 0.36,17 0.52,19 0.63,21 0.51,24 0.66,30 0.3}

{5 0.22,13 0.41,17 0.45,19 0.54,20 0.86,21 0.46,24 0.73}

{2 1.7,5 0.17,9 0.89,16 0.73,18 1.05,28 0.45,30 0.3,31 student}

{1 0.5,5 0.17,10 0.85,13 0.48,17 0.45,21 0.46,27 0.77}

{1 0.4,5 0.21,11 0.62,13 0.5,17 0.45,19 0.54,21 0.51,30 0.3}

{7 0.57}

{5 0.22,7 0.57,12 0.52,14 0.39,17 0.5,19 0.68,21 0.51,24 0.69}

{5 0.17,10 0.73,16 0.73,22 0.8,30 0.3,31 student}

{0 0.78,1 0.53,5 0.22,6 0.78,8 0.95,10 0.89,12 0.5,13 0.48,14 0.32,15 0.61,17 0.45,19 0.67,20 0.86,21 0.54,24 0.69,28 0.56,30 0.3}

{4 0.77,5 0.2,19 0.54,28 0.45,30 0.3}

{3 1.53,9 0.99,10 0.73,16 0.81,20 0.96,22 0.89,28 0.5,30 0.33,31 student}
```

{5 0.2,13 0.41,14 0.36,15 0.61,17 0.45,19 0.54,24 0.59}

{5 0.21,7 0.57,16 0.73,18 1.05,26 1.05,30 0.3,31 student}

{3 1.53,26 1.05,30 0.33,31 student}

{4 0.77,5 0.2,19 0.54,28 0.45,30 0.3}

{5 0.19,6 0.7,12 0.45,14 0.32,15 0.61,19 0.54,27 0.77}

{0 0.7,1 0.45,5 0.17,6 0.78,7 0.67,11 0.62,12 0.5,14 0.4,17 0.45,21 0.46,30 0.3}

{1 0.51,5 0.17,6 0.7,8 0.91,12 0.5,13 0.48,14 0.39,21 0.46,24 0.66}

{7 0.57}

{1 0.54,4 0.77,5 0.23,8 0.92,11 0.73,13 0.51,14 0.32,17 0.55,19 0.65,21 0.46,24 0.69,27 0.77,28 0.54}

{1 0.49,5 0.17,12 0.45,14 0.32,17 0.45}

{20 0.96,23 1.16,28 0.5,30 0.3,31 student}

{1 0.4,4 0.77,13 0.46,21 0.55}

{0 0.7,1 0.45,5 0.19,8 0.81,19 0.54}

{0 0.78,5 0.2,8 0.73,19 0.54,27 0.77,30 0.3}

{5 0.17,9 0.89,10 0.73,14 0.32,16 0.81,20 0.96,22 0.8,23 1.16,28 0.5,30 0.3,31 student}

{1 0.45,5 0.17,10 0.85,12 0.55,13 0.48,17 0.45,21 0.46,27 0.77}

{9 0.99,10 0.73,16 0.73,22 0.8,28 0.5,31 student}

{1 0.4,4 0.86,5 0.22,8 0.73,12 0.52,13 0.46,14 0.32,15 0.68,17 0.45,19 0.69,20 0.86,21 0.54,27 0.77}

{9 0.89,28 0.45,29 1.53,31 student}

{3 1.53,9 0.89,28 0.45,30 0.33,31 student}

{2 1.7,10 0.73,30 0.3,31 student}

```
{5 0.22,7 0.57,8 0.92,12 0.62,13 0.48,14 0.42,15 0.71,19 0.6,21 0.46,24 0.71,27 0.86,30 0.3}

{1 0.4,5 0.19,9 0.89,16 0.73,22 0.8,26 1.05,30 0.33,31 student}

{0 0.7,1 0.4,5 0.21,7 0.57,11 0.62,13 0.41,14 0.32,17 0.45,19 0.54,21 0.46,24 0.69,27 0.86,30 0.3}

{5 0.19,12 0.54,14 0.32,19 0.54,24 0.71}

{5 0.19,7 0.57,11 0.7,14 0.32}

{0 0.7,1 0.5,5 0.21,7 0.57,8 0.85,12 0.56,14 0.36,19 0.63,21 0.46,30 0.3}

{1 0.4,5 0.21,7 0.64,11 0.62,12 0.45,13 0.46,14 0.4,15 0.68,17 0.5,21 0.46,24 0.66,28 0.55}

{1 0.4,5 0.19,6 0.75,7 0.57,11 0.62,13 0.46,14 0.32,15 0.68,19 0.54,24 0.59}

{0 0.7,5 0.2,14 0.38,19 0.54,30 0.3}

{0 0.82,1 0.47,5 0.22,6 0.82,7 0.67,11 0.62,12 0.45,13 0.41,14 0.4,17 0.52,19 0.54,24 0.66,30 0.3}

{7 0.64,11 0.73,13 0.46,14 0.36,15 0.61,24 0.66}
```

APPENDIX E: THE 70-30 FILE AFTER FEATURE SELECTION BY THE HYBRID MODEL IN SPARSE ARFF FOR BINARY CLASS WPC

```
@relation 'webkb-weka.filters.unsupervised.attribute.Remove-V-R105,150,189,232,450,544,558,691,710,1019,1052,1058,1116,1128,1203,1217,1262,1385,1507,1510,1666,1693,1704,1826,1864,1918,2084,2352,2353,2466,2592,2774-weka.filters.unsupervised.instance.Randomize-S42-weka.filters.unsupervised.attribute.Remove-V-R2,6,7,8,17,32'

@attribute assign numeric
@attribute cours numeric
@attribute cse numeric
@attribute document numeric
@attribute ithaca numeric
@attribute class {course,student}

@data
{4 0.73,5 student}
{0 0.4,1 0.2,2 0.73,3 0.73}
{4 0.73,5 student}
{0 0.4,1 0.17,2 0.73}
{5 student}
{2 0.81}
{5 student}
{2 0.75}
```

```
{0 0.4}

{0 0.5,1 0.21,2 0.79,3 0.57}

{1 0.2}

{0 0.4,1 0.22,3 0.57}

{0 0.5,1 0.17,2 0.7}

{5 student}

{0 0.4}

{4 0.73,5 student}

{0 0.52,1 0.23}

{0 0.4,1 0.2,2 0.75,3 0.57}

{0 0.4,1 0.2,2 0.81,3 0.67}

{0 0.53,1 0.2,2 0.81,3 0.57}

{0 0.54,1 0.21}

{1 0.17}

{0 0.4,1 0.17,2 0.82,3 0.64}

{1 0.17,4 0.81,5 student}

{1 0.2}

{0 0.55,1 0.19}

{5 student}

{4 0.73,5 student}

{1 0.17}

{0 0.45,1 0.19}

{1 0.21}
```

```
{2 0.77,3 0.64}
{4 0.73,5 student}
{4 0.73,5 student}
{4 0.81,5 student}
{4 0.73,5 student}
{5 student}
{1 0.17,4 0.73,5 student}
{1 0.17,4 0.81,5 student}
{0 0.47,1 0.23,3 0.57}
{3 0.57}
{1 0.2,5 student}
{1 0.2}
{1 0.19}
{0 0.4,2 0.73,3 0.64}
{0 0.5,1 0.17,2 0.73}
{1 0.17,2 0.73}
{0 0.4,1 0.17,2 0.7}
{0 0.53,1 0.21}
{0 0.4,1 0.19,2 0.7,3 0.67}
{3 0.64}
{1 0.19,4 0.81,5 student}
{1 0.19}
{1 0.17,2 0.79}
```

```
{1 0.17}

{1 0.17}

{0 0.51,1 0.17,2 0.7}

{0 0.4,1 0.21,3 0.64}

{1 0.22}

{1 0.17,4 0.73,5 student}

{0 0.5,1 0.17}

{0 0.4,1 0.21}

{3 0.57}

{1 0.22,3 0.57}

{1 0.17,4 0.73,5 student}

{0 0.53,1 0.22,2 0.78}

{1 0.2}

{4 0.81,5 student}

{1 0.2}

{1 0.21,3 0.57,4 0.73,5 student}

{5 student}

{1 0.2}

{1 0.19,2 0.7}

{0 0.45,1 0.17,2 0.78,3 0.67}

{0 0.51,1 0.17,2 0.7}

{3 0.57}

{0 0.54,1 0.23}
```

```
{0 0.49,1 0.17}

{5 student}

{0 0.4}

{0 0.45,1 0.19}

{1 0.2}

{1 0.17,4 0.81,5 student}

{0 0.45,1 0.17}

{4 0.73,5 student}

{0 0.4,1 0.22}

{5 student}

{5 student}

{5 student}

{1 0.22,3 0.57}

{0 0.4,1 0.19,4 0.73,5 student}

{0 0.4,1 0.21,3 0.57}

{1 0.19}

{1 0.19,3 0.57}

{0 0.5,1 0.21,3 0.57}

{0 0.4,1 0.21,3 0.64}

{0 0.4,1 0.19,2 0.75,3 0.57}

{1 0.2}

{0 0.47,1 0.22,2 0.82,3 0.67}

{3 0.64}
```

APPENDIX F: FEATURES SELECTED BY THE HYBRID MODEL FROM EACH INPUT FILE FOR BINARY CLASS WPC

Input File	Features selected
70 – 30	assign, cours, cse, document, Ithaca
100-100	cse, document, homework, hour, ithaca, materi
200-200	cours,credit,cse,document,graduat,homework,hour,inform,instructor, materi,research,seattl,univers
300-200	cours,hour,inform,ithaca, materi,pm,resum,syllabu,univers
300-300	assign,cours,cse,document,fax,graduat,homework,instructor,materi, resum,seattl,syllabu,univers,usa
350-150	assign,comment,cool,cornel,cours,cse,grade,homework,hour,ithaca, master,research,univers
400-200	assign, cornel,cours,credit,document,grade,graduat,homework, instructor, ithaca,materi,ph,pm,syllabu,usa
400-300	anim, assign, cours, credit, cse, document, fax, graduat, homework, inform, instructor, materi, research, resum, syllabu, univers, usa
400-400	anim, assign,cours,credit,cse,document,fax,graduat,homework, inform,instructor,materi,person,resum,syllabu,univers,usa

APPENDIX G: FEATURES SELECTED BY THE WARDS METHOD FROM EACH INPUT FILE FOR BINARY CLASS WPC

Input File	Features selected
70 – 30	assign, Ithaca
100-100	cse, homework, Ithaca
200-200	cours,document,graduat,homework,hour,inform,instructor, materi,seattl,univers
300-200	cours, hour, inform, ithaca, pm, syllabu
300-300	assign, cours, graduat, homework, instructor, materi, resum, seattl, syllabu, univers, usa
350-150	assign, cornel, cours, cse, grade, homework, hour, ithaca, master, research
400-200	assign, cornel,cours, grade, graduat, homework, instructor, ithaca, materi, ph, pm, syllabu
400-300	assign, cours, document, fax, graduat, homework, inform, instructor, materi, research, resum, univers, usa
400-400	assign, cours, document, fax, graduat, homework, inform, instructor, materi, person, resum, syllabu, univers, usa

APPENDIX H: THE SPARSE ARFF AFTER FEATURE SELECTION USING CFS FOR MULTI CLASS WPC

```
@relation 'webkb-weka.filters.unsupervised.attribute.Remove-V-R57,191,410,413,552,1387,1415,1460,1672,1684,1908,2145,2248,2767,2915,2944,3153,3190,3305,3708,4042,4047,4674,4972,5013,5278,5497,5546,5614,5638,5669,7099,7135,7160,7334,7361,7454,7619-weka.filters.unsupervised.instance.Randomize-S42'

@attribute acm numeric
@attribute algorithm numeric
@attribute assign numeric
@attribute associ numeric
@attribute base numeric
@attribute cornel numeric
@attribute cours numeric
@attribute cse numeric
@attribute department numeric
@attribute describ numeric
@attribute dyer numeric
@attribute exam numeric
@attribute fax numeric
@attribute handout numeric
@attribute homework numeric
@attribute hour numeric
```

```
@attribute inform numeric
@attribute instructor numeric
@attribute ithaca numeric
@attribute lectur numeric
@attribute master numeric
@attribute materi numeric
@attribute ny numeric
@attribute peopl numeric
@attribute ph numeric
@attribute professor numeric
@attribute recent numeric
@attribute region numeric
@attribute research numeric
@attribute resum numeric
@attribute right numeric
@attribute univers numeric
@attribute upson numeric
@attribute util numeric
@attribute washington numeric
@attribute web numeric
@attribute wisc numeric
@attribute class {course,student,faculty,project}
```

```
@data
{0 0.83,1 0.64,3 0.8,4 0.71,9 1.32,16 0.4,23 0.88,26 0.74,28 0.36,31 0.21,33 1.35,36 1.08,37 project}
{3 0.8,4 0.63,8 1.4,12 0.75,22 0.82,24 0.93,25 0.59,26 0.79,28 0.41,31 0.27,36 1,37 faculty}
{0 0.93,1 0.64,3 0.8,20 0.91,24 0.72,25 0.71,28 0.38,31 0.25,34 0.6,37 faculty}
{3 0.8,12 0.75,13 1.01,16 0.37,23 0.88,24 0.72,25 0.59,28 0.36,31 0.21,36 1,37 faculty}
{5 0.44,16 0.33,18 0.84,22 0.91,26 0.63,29 1.1,31 0.24,37 student}
{16 0.41,31 0.26,37 project}
{1 0.64,4 0.82,6 0.45,10 1.7,16 0.39,37 project}
{2 0.77,13 1.01,15 0.86,16 0.39,19 0.74,23 0.88,32 0.81,35 0.58}
{5 0.41,18 0.76,22 0.82,31 0.21,32 0.81,37 student}
{0 1.04,3 0.8,4 0.63,20 0.91,24 0.8,25 0.59,26 0.7,28 0.38,31 0.24,34 0.7,37 faculty}
{5 0.43,6 0.45,9 1.13,12 0.75,18 0.76,25 0.59,26 0.63,28 0.32,31 0.21,32 0.81,35 0.58,37 faculty}
{25 0.59,28 0.32,31 0.21,36 0.9,37 faculty}
{23 0.98,28 0.36,33 1.5,34 0.6,37 project}
{0 0.97,3 0.93,5 0.46,15 0.77,16 0.37,18 0.76,19 0.74,20 0.91,22 0.82,24 0.72,25 0.69,28 0.4,31 0.21,32 0.81,34 0.6,37 faculty}
{5 0.41,24 0.72,28 0.36,31 0.21,37 student}
{37 student}
{28 0.32,37 student}
{7 0.96,14 1.12,15 0.86,16 0.33,17 0.97,21 0.92,34 0.6}
{2 0.96,6 0.54,7 1.01,15 0.86,16 0.33,25 0.59,34 0.7}
```

```
{2 0.9,6 0.45,7 0.89,11 1.23,13 1.13,14 0.91,15 0.86,16 0.39,17 0.97,19 0.74,28 0.36,34 0.6,35
0.58}
{5 0.41,19 0.74,23 0.98,28 0.32,32 0.81,37 student}
{2 0.9,3 0.89,4 0.74,6 0.59,11 1.32,13 1.01,14 0.91,15 0.9,16 0.41,19 0.87,21 1.15,23 0.88,25
0.65,26 0.63,28 0.4,32 0.98}
{24 0.72,25 0.71,26 0.63,28 0.38,31 0.27,34 0.6,37 faculty}
{9 1.25,10 1.89,26 0.63,37 project}
{6 0.52,11 1.02,14 1.1,15 0.86,16 0.33,19 0.74}
{5 0.41,18 0.84,20 0.91,22 0.91,29 1.1,31 0.21,37 student}
{4 0.74,9 1.13,23 0.88,28 0.4,34 0.6,37 project}
{16 0.33,26 0.63,31 0.21,36 0.9,37 project}
{1 0.72,3 0.8,5 0.41,19 0.82,24 0.72,25 0.59,26 0.63,28 0.32,31 0.27,37 faculty}
{2 0.77,6 0.5,7 0.89,13 1.01,15 0.86,16 0.39,17 0.97,19 0.74,28 0.36,34 0.6}
{37 project}
{5 0.45,12 0.75,23 0.88,29 1.1,31 0.24,37 student}
{4 0.63,5 0.41,13 1.18,15 0.86,16 0.37,17 0.97,28 0.32,32 0.95,35 0.64}
{0 0.83,1 0.72,5 0.43,8 1.4,28 0.4,31 0.27,37 faculty}
{5 0.37,18 0.76,22 0.82,29 1.1,37 student}
{7 0.98,11 1.02,14 0.91,15 0.77,16 0.37,17 0.97,19 0.74,31 0.21,34 0.67,35 0.7}
{37 student}
{1 0.72,3 0.8,4 0.78,9 1.13,16 0.33,21 0.92,24 0.72,28 0.41,31 0.27,33 1.35,34 0.6,36 1,37
project}
{37 faculty}
```

```
{1 0.64,6 0.45,7 0.93,14 0.91,16 0.33,34 0.6,35 0.58}
{16 0.37,28 0.32,31 0.24,35 0.68,37 project}
{4 0.63,27 1.52,37 project}
{16 0.33,36 1.08,37 project}
{5 0.43,6 0.5,18 0.84,20 0.91,22 0.82,31 0.21,34 0.6,37 student}
{9 1.13,16 0.33,28 0.36,31 0.24,34 0.7,37 project}
{16 0.33,23 0.88,31 0.21,37 project}
{0 0.93,4 0.63,5 0.43,8 1.4,16 0.33,20 0.91,25 0.59,28 0.32,31 0.21,37 faculty}
{1 0.83,4 0.71,9 1.13,16 0.39,28 0.4,35 0.68,36 1,37 project}
{1 0.72,2 0.96,6 0.45,7 0.93,11 1.14,15 0.86,16 0.41,17 0.97,34 0.6,35 0.64}
{2 0.77,4 0.63,6 0.45,7 0.96,13 1.01,14 0.91,16 0.42,19 0.74,25 0.59,35 0.68}
{5 0.43,16 0.33,20 0.91,31 0.21,37 student}
{6 0.45,7 1.01,16 0.33,31 0.24,34 0.72,35 0.7}
{0 0.83,1 0.83,3 0.89,4 0.63,5 0.48,8 1.4,16 0.42,18 0.76,23 1.03,24 0.72,25 0.65,26 0.74,28
0.38,37 faculty}
{0 0.83,1 0.84,5 0.44,12 0.75,16 0.33,18 0.76,19 0.82,21 0.92,27 1.78,28 0.36,31 0.21,32 0.81,37
student}
{5 0.37,37 student}
{0 1.02,1 0.64,3 0.89,4 0.71,6 0.45,8 1.4,12 0.75,15 0.86,16 0.4,24 0.9,25 0.65,26 0.78,28 0.42,31
0.27,36 1.05,37 faculty}
{1 0.72,37 project}
{5 0.37,6 0.58,14 1.07,16 0.4,19 0.82,21 1.15,35 0.64}
{1 0.64,5 0.44,18 0.76,22 0.82,31 0.24,37 student}
```

{30 1.82,37 project}

{1 0.84,2 0.77,4 0.63,6 0.59,7 0.96,11 1.02,14 1.16,15 0.77,16 0.33,17 1.08,19 0.87,28 0.36,34 0.6}

{37 student}

{0 0.83,1 0.64,3 0.89,4 0.74,5 0.48,16 0.37,18 0.76,20 0.91,23 0.88,25 0.65,28 0.36,31 0.21,35 0.58,37 project}

{2 0.77,4 0.63,5 0.37,9 1.13,16 0.4,20 0.91,37 project}

{16 0.37,23 0.88,31 0.21,36 0.9,37 project}

{2 0.95,6 0.45,14 0.91,16 0.33,19 0.74}

{0 0.83,4 0.63,12 0.75,16 0.39,24 0.72,25 0.59,26 0.7,28 0.36,31 0.25,34 0.6,36 1,37 faculty}

{3 0.8,16 0.39,27 1.52,28 0.32,31 0.21,35 0.71,37 project}

{7 0.8,29 1.1,31 0.21,34 0.67,37 faculty}

{1 0.82,4 0.79,6 0.45,10 2.31,12 0.75,24 0.93,25 0.59,26 0.76,28 0.38,31 0.24,36 1,37 faculty}

{0 0.93,3 0.8,12 0.75,16 0.39,24 0.72,25 0.71,26 0.74,28 0.4,31 0.24,34 0.7,35 0.58,37 faculty}

{0 0.83,4 0.71,5 0.44,21 1.03,31 0.21,35 0.64}

{1 0.64,2 0.96,3 0.8,6 0.45,7 0.89,14 1.12,15 0.86,17 0.97,34 0.67,35 0.58}

{5 0.37,23 0.88,28 0.36,37 project}

{0 0.83,3 0.8,5 0.46,12 0.75,16 0.37,18 0.76,22 0.82,24 0.72,25 0.59,28 0.38,31 0.24,32 0.81,35 0.64,37 student}

{37 student}

{0 1.2,1 0.86,3 0.96,4 0.84,5 0.46,9 1.13,12 0.75,16 0.39,19 0.96,22 1.12,24 0.91,25 0.69,26 0.63,28 0.4,31 0.29,34 0.72,36 0.9,37 faculty}

```
{1 0.79,3 0.8,4 0.81,6 0.5,12 0.75,16 0.41,24 0.72,25 0.59,26 0.79,28 0.4,31 0.24,36 1.13,37 faculty}
{2 0.77,6 0.52,7 0.93,15 0.77,17 1.08,20 0.91,25 0.59,34 0.6}
{5 0.37,6 0.52,11 1.14,15 0.77,16 0.37,17 0.97,19 0.74,21 0.92,35 0.58}
{5 0.37,6 0.5,11 1.14,13 1.13,14 1.07,15 0.77,35 0.58}
{23 0.88,28 0.32,37 project}
{2 0.86,6 0.45,7 0.96,11 1.02,13 1.01,14 1.12,15 0.86,16 0.37,17 0.97,26 0.63,31 0.21,34 0.67,35 0.68}
{0 0.83,6 0.45,12 0.75,24 0.72,25 0.59,26 0.63,28 0.36,31 0.24,36 0.9,37 faculty}
{0 0.83,5 0.46,6 0.45,12 0.75,19 0.74,20 1.02,24 0.72,25 0.65,28 0.36,31 0.26,32 0.81,37 faculty}
{12 0.75,24 0.72,25 0.59,26 0.63,28 0.36,31 0.24,36 1,37 faculty}
{4 0.71,5 0.41,14 0.91,16 0.33,28 0.32,31 0.21,37 student}
{2 0.86,5 0.43,6 0.45,14 1.12,15 0.9,19 0.74,32 0.95}
{9 1.39,16 0.37,20 0.91,23 0.98,26 0.63,34 0.67,35 0.58,37 project}
{0 0.93,3 0.93,5 0.37,20 0.91,24 0.84,25 0.73,28 0.38,31 0.26,34 0.6,37 faculty}
{1 0.75,3 0.8,4 0.63,5 0.41,12 0.83,18 0.76,22 0.82,25 0.59,26 0.7,28 0.32,31 0.21,37 faculty}
{5 0.45,9 1.13,12 0.75,18 0.84,20 0.91,22 0.91,26 0.63,31 0.24,32 0.81,35 0.64,37 student}
{4 0.63,12 0.75,16 0.37,25 0.59,28 0.4,31 0.24,33 1.35,37 project}
{1 0.64,4 0.63,12 0.75,16 0.43,20 1.14,26 0.63,28 0.36,31 0.26,34 0.72,35 0.71,37 faculty}
{16 0.37,23 0.88,28 0.32,31 0.25,37 project}
{5 0.43,18 0.76,22 0.82,31 0.21,32 0.81,37 student}
{1 0.78,3 0.8,12 0.75,19 0.74,24 0.72,25 0.59,26 0.7,28 0.4,31 0.24,33 1.35,36 1,37 faculty}
{5 0.43,18 0.76,22 0.82,29 1.1,31 0.21,35 0.68,37 student}
```

```
{2 0.86,6 0.5,7 0.93,15 0.86,16 0.39,17 1.08,34 0.6,35 0.64}
{9 1.13,10 1.7,16 0.4,37 project}
{28 0.32,31 0.21,36 0.9,37 project}
{}
{5 0.44,6 0.45,16 0.39,18 0.76,35 0.58,37 faculty}
{0 0.83,1 0.64,3 0.8,4 0.71,9 1.32,16 0.4,23 0.88,26 0.74,28 0.36,31 0.21,33 1.35,36 1.08,37 project}
{0 1.07,1 0.87,2 0.77,3 0.8,5 0.43,12 0.83,18 0.76,22 0.82,25 0.59,26 0.7,28 0.41,31 0.21,32 0.81,37 faculty}
{1 0.64,2 0.77,5 0.41,6 0.54,11 1.02,13 1.01,15 0.77,16 0.33,19 0.74,21 0.92,31 0.21}
{5 0.41,16 0.33,18 0.76,20 0.91,28 0.38,29 1.1,31 0.21,37 student}
{5 0.43,18 0.76,22 0.82,24 0.72,28 0.32,31 0.24,32 0.81,37 student}
{20 0.91,29 1.1,37 student}
{10 1.7,28 0.32,37 project}
{5 0.41,16 0.43,21 1.03,28 0.4,33 1.35,37 project}
{5 0.43,18 0.76,22 0.82,28 0.38,31 0.24,37 student}
{5 0.37,37 student}
{2 1.06,5 0.37,6 0.5,13 1.22,14 1.26,15 0.86,16 0.37,19 0.94,21 0.92,23 0.88,26 0.63,31 0.21,32 0.9,35 0.64}
{2 0.77,3 0.8,5 0.47,6 0.58,11 1.14,14 1.2,17 0.97,19 0.74,22 0.82,31 0.27,32 0.98}
{3 0.93,5 0.43,6 0.52,8 1.4,19 0.74,31 0.24,37 faculty}
{0 1,1 0.64,5 0.37,6 0.5,12 0.75,24 0.8,26 0.7,28 0.36,31 0.24,36 0.9,37 faculty}
{6 0.54,15 0.77,16 0.39,35 0.58}
```

```
{1 0.72,3 0.8,4 0.63,5 0.41,6 0.5,12 0.75,28 0.36,31 0.21,35 0.58,37 faculty}
{28 0.32,37 project}
{5 0.43,16 0.33,18 0.76,22 0.82,23 0.88,31 0.21,37 student}
{}
{5 0.41,24 0.84,28 0.36,29 1.1,31 0.21,37 student}
{1 0.72,3 0.8,6 0.57,15 0.77,19 0.74,21 0.92,32 0.81}
{0 0.93,5 0.47,12 0.75,16 0.33,18 0.91,19 0.82,22 0.96,24 0.72,26 0.63,28 0.36,31 0.25,32 0.81,35 0.68,37 student}
{1 0.75,2 0.77,4 0.63,6 0.5,7 0.89,11 1.28,13 1.01,14 1.22,15 0.93,16 0.33,17 0.97,19 0.82,21 0.92,34 0.72}
{1 0.79,4 0.63,12 0.75,16 0.33,24 0.72,25 0.59,26 0.7,28 0.32,31 0.24,36 1,37 faculty}
{2 0.77,5 0.41,6 0.5,18 0.76,22 0.82,28 0.32,29 1.1,31 0.24,35 0.58,37 student}
{4 0.74,5 0.46,9 1.13,28 0.36,31 0.21,37 project}
{2 0.86,4 0.63,6 0.45,7 1,13 1.01,14 1.02,16 0.41,19 0.74,25 0.59,31 0.21,34 0.6,35 0.71}
{0 1,1 0.78,4 0.71,5 0.37,12 0.75,22 0.82,24 0.72,25 0.59,26 0.63,28 0.36,31 0.24,34 0.6,36 0.9,37 faculty}
{2 0.99,6 0.58,7 0.98,9 1.25,11 1.14,15 0.77,16 0.42,19 0.82,21 1.03,26 0.63,31 0.21,34 0.67,35 0.71}
{0 0.83,5 0.45,12 0.75,16 0.33,18 0.76,22 0.82,28 0.32,31 0.27,32 0.81,35 0.58,37 student}
{1 0.64,4 0.83,16 0.4,25 0.59,28 0.39,31 0.21,35 0.64,37 project}
{1 0.64,2 0.77,6 0.45,7 0.89,13 1.01,14 0.91,21 0.92,34 0.6,35 0.64}
{1 0.64,3 0.8,6 0.45,16 0.4,25 0.65,26 0.63,28 0.4,31 0.21,36 1.05,37 faculty}
{1 0.64,5 0.44,6 0.45,16 0.33,18 0.84,20 0.91,22 0.82,29 1.1,31 0.25,37 student}
```

```
{2 0.99,6 0.45,7 0.89,11 1.02,14 1.02,15 0.9,16 0.4,25 0.59,34 0.6}
{3 0.8,4 0.71,5 0.44,6 0.45,12 0.75,18 0.84,22 0.91,23 0.98,25 0.59,28 0.32,31 0.24,32 0.81,37 faculty}
{0 1,1 0.75,5 0.43,6 0.52,9 1.32,15 0.77,16 0.33,21 0.92,24 0.72,25 0.59,26 0.63,28 0.45,31 0.25,32 0.81,33 1.35,35 0.58,37 faculty}
{6 0.52,7 1.01,13 1.01,15 0.86,16 0.33,17 1.13,19 0.89,21 1.03,34 0.7,35 0.58}
{5 0.41,6 0.52,21 0.92,31 0.21,35 0.68}
{4 0.63,5 0.37,19 0.74,28 0.32,35 0.58}
{12 0.75,28 0.4,31 0.24,37 project}
{6 0.5,7 0.89,13 1.01,15 0.77,16 0.37,19 0.82,34 0.6,35 0.58}
{4 0.63,23 0.88,28 0.36,31 0.24,35 0.7,37 project}
{2 0.77,4 0.63,5 0.43,6 0.56,9 1.25,13 1.01,14 0.91,15 0.86,16 0.41,17 1.08,19 0.82,28 0.32,32 0.98,35 0.64}
{0 1.02,1 0.64,3 0.8,6 0.5,8 1.4,12 0.75,24 0.91,25 0.59,26 0.74,28 0.42,31 0.25,36 1.05,37 faculty}
{4 0.63,5 0.43,16 0.33,18 0.76,20 0.91,28 0.32,31 0.24,35 0.58,37 student}
{24 0.72,25 0.59,28 0.32,31 0.21,36 1,37 faculty}
{5 0.41,18 0.76,22 0.82,29 1.1,31 0.21,32 0.81,37 student}
{4 0.63,5 0.45,6 0.58,11 1.14,14 1.27,15 0.9,16 0.43,17 1.13,21 1.03,23 0.88,28 0.32,31 0.21,32 0.9}
{1 0.75,5 0.41,12 0.75,18 0.76,22 0.82,25 0.59,26 0.63,28 0.36,31 0.21,37 faculty}
{2 1.01,6 0.57,7 1,11 1.31,14 1.02,15 0.9,16 0.33,17 0.97,19 0.74,21 1.13,26 0.63,31 0.21,34 0.67,35 0.71}
```

```
{1 0.64,3 0.8,4 0.63,12 0.75,24 0.72,25 0.59,28 0.32,31 0.24,36 0.9,37 faculty}
{4 0.63,5 0.44,26 0.63,28 0.38,31 0.21,32 0.81,37 student}
{2 0.77,6 0.45,7 0.96,11 1.02,13 1.01,14 1.18,15 0.9,16 0.37,17 0.97,26 0.63,28 0.32,31 0.21,34 0.67,35 0.64}
{37 student}
{2 0.77,13 1.01,15 0.86,16 0.39,19 0.74,23 0.88,35 0.58}
{0 0.93,4 0.76,5 0.37,16 0.33,23 0.88,37 project}
{2 0.77,6 0.45,7 1.04,11 1.14,15 0.86,16 0.37,17 1.17,19 0.89,31 0.21,34 0.7,35 0.72}
{7 0.93,11 1.02,17 0.97,34 0.6,35 0.64}
{5 0.41,6 0.52,16 0.39,21 0.92,26 0.63,31 0.21}
{16 0.33,24 0.72,26 0.63,31 0.21,33 1.35,34 0.6,37 project}
{37 student}
{3 0.8,5 0.45,24 0.72,28 0.32,31 0.21,35 0.58,37 student}
{5 0.45,18 0.76,22 0.82,24 0.72,28 0.36,31 0.24,37 student}
{4 0.63,9 1.13,16 0.33,21 1.03,23 0.98,31 0.21,34 0.7,37 project}
{3 0.8,5 0.44,16 0.33,18 0.76,20 0.91,22 0.82,29 1.1,31 0.25,35 0.64,37 student}
{2 0.9,9 1.25,16 0.33,27 1.9,28 0.32,31 0.21,36 0.9,37 project}
{5 0.37,18 0.76,22 0.82,37 student}
{2 0.77,6 0.52,7 0.96,14 0.91,15 0.9,16 0.39,17 0.97,19 0.82,31 0.21,34 0.67,35 0.68}
{0 1.04,1 0.72,4 0.71,12 0.83,16 0.37,25 0.59,26 0.7,28 0.41,31 0.24,36 0.9,37 faculty}
{5 0.41,18 0.76,22 0.82,31 0.21,32 0.81,37 student}
{0 0.83,4 0.71,9 1.13,26 0.63,28 0.32,30 1.82,35 0.58,37 project}
{5 0.41,18 0.76,22 0.82,31 0.21,32 0.81,37 student}
```

```
{5 0.37,16 0.33,20 0.91,28 0.36,31 0.21,37 student}
{5 0.37,16 0.33,37 student}
{5 0.41,6 0.45,20 0.91,31 0.21,37 student}
{4 0.71,10 1.7,16 0.33,26 0.63,27 1.52,37 project}
{37 project}
{5 0.37,16 0.33,18 0.76,20 0.91,29 1.1,37 student}
{4 0.71,7 0.8,16 0.39,28 0.36,31 0.21,33 1.35,34 0.7,35 0.7,37 project}
{6 0.5,9 1.13,16 0.33,26 0.78,28 0.36,31 0.21,36 1,37 project}
{4 0.71,9 1.25,16 0.43,23 0.88,26 0.63,28 0.39,36 1.15,37 project}
{1 0.82,2 1.01,5 0.47,6 0.55,11 1.14,13 1.31,15 0.98,17 0.97,19 1.09,31 0.21,32 1.08,35 0.68}
{5 0.37,6 0.52,11 1.02,16 0.33,21 0.92,31 0.21,32 0.9}
{0 0.83,2 0.77,7 0.96,15 0.77,16 0.37,34 0.6}
{0 1.04,1 0.78,2 0.77,16 0.39,24 0.8,25 0.59,28 0.38,31 0.21,37 faculty}
{4 0.63,16 0.37,30 1.82,31 0.21,37 project}
{0 0.83,1 0.64,5 0.37,6 0.45,8 1.4,12 0.75,19 0.74,24 0.72,25 0.59,31 0.21,32 0.81,35 0.58,37 faculty}
{5 0.44,6 0.45,16 0.37,18 0.76,20 0.91,22 0.82,28 0.32,29 1.1,31 0.25,35 0.58,37 student}
{1 0.64,5 0.41,8 1.4,19 0.74,25 0.59,31 0.27,37 faculty}
{37 faculty}
{4 0.71,7 0.89,16 0.33,23 0.88,26 0.63,27 1.52,31 0.21,34 0.7,35 0.64,37 project}
{1 0.72,5 0.37,6 0.54,18 0.76,20 0.91,29 1.1,31 0.21,37 student}
{5 0.41,16 0.33,28 0.32,37 project}
```

```
{2 0.9,6 0.57,7 1.04,11 1.02,13 1.01,14 0.91,15 0.77,16 0.41,19 0.87,21 0.92,31 0.21,34 0.67,35 0.68}
{5 0.41,6 0.45,18 0.76,20 0.91,28 0.32,31 0.21,35 0.64,37 student}
{12 0.75,19 0.74,26 0.63,31 0.24,34 0.7,37 faculty}
{1 0.72,6 0.45,11 1.02,13 1.01,14 1.19,17 0.97,19 0.74,32 0.9}
{1 0.64,5 0.41,6 0.45,16 0.33,31 0.21,32 0.81,37 student}
{1 0.64,5 0.44,16 0.37,19 0.74,25 0.59,28 0.36,31 0.26,37 student}
{16 0.33,23 0.88,28 0.36,37 project}
{5 0.41,6 0.45,16 0.37,18 0.84,20 0.91,22 0.91,31 0.21,37 student}
{5 0.41,31 0.24,37 student}
{4 0.63,28 0.32,31 0.24,35 0.58,37 project}
{0 0.93,4 0.78,28 0.38,31 0.28,36 0.9,37 project}
{0 0.93,6 0.45,7 0.89,19 0.74,24 0.8,25 0.71,26 0.63,28 0.38,31 0.21,34 0.6,37 faculty}
{24 0.72,25 0.59,28 0.38,31 0.21,37 project}
{1 0.72,21 1.08,23 0.88,24 0.8,26 0.7,28 0.39,31 0.25,37 project}
{5 0.44,18 0.76,22 0.82,28 0.32,31 0.25,32 0.81,37 student}
{5 0.37,6 0.45,20 0.91,29 1.1,37 student}
{2 0.77,7 0.93,16 0.37,34 0.6,35 0.58}
{5 0.44,6 0.5,12 0.75,16 0.33,18 0.76,20 1.02,22 0.82,29 1.1,31 0.24,35 0.64,37 student}
{7 1.04,16 0.33,28 0.32,31 0.21,34 0.67,35 0.64}
{16 0.33,27 1.52,31 0.21,37 project}
{16 0.33,26 0.63,28 0.38,31 0.21,35 0.58,37 project}
{5 0.44,6 0.45,12 0.75,18 0.76,22 0.82,25 0.59,28 0.36,29 1.1,31 0.21,37 faculty}
```

```
{37 project}
{7 0.93,34 0.6,35 0.58}
{3 0.8,5 0.43,12 0.75,16 0.41,20 0.91,24 0.8,28 0.39,29 1.1,31 0.21,37 student}
{37 student}
{2 0.77,6 0.5,7 0.96,13 1.01,15 0.86,16 0.33,17 1.08,21 0.92,34 0.7,35 0.64}
{5 0.45,26 0.63,28 0.36,31 0.21,35 0.64,37 student}
{12 0.75,16 0.37,24 0.72,25 0.59,28 0.36,31 0.21,33 1.35,34 0.7,37 faculty}
{4 0.63,10 1.89,37 project}
{37 faculty}
{5 0.43,31 0.21,37 student}
{5 0.44,31 0.21,37 student}
{37 student}
{6 0.5,7 0.89,15 0.86,16 0.37,17 0.97,19 0.74,34 0.67}
{1 0.64,5 0.43,23 0.88,28 0.39,37 project}
{5 0.37,6 0.45,23 0.88,26 0.63,31 0.21,35 0.58,37 student}
{2 0.86,5 0.44,6 0.5,11 1.02,13 1.13,14 1.26,15 0.95,16 0.37,17 0.97,25 0.59,32 1}
{5 0.41,6 0.45,18 0.76,22 0.82,29 1.1,31 0.21,35 0.58,37 student}
{0 0.93,1 0.64,3 0.89,4 0.63,6 0.45,12 0.75,16 0.39,20 1.02,25 0.59,26 0.63,28 0.32,31 0.25,34 0.72,37 faculty}
{0 0.93,3 0.93,4 0.74,5 0.43,6 0.45,8 1.4,16 0.44,19 0.82,24 0.72,25 0.59,28 0.4,31 0.26,33 1.58,34 0.67,37 faculty}
{2 0.77,6 0.45,7 0.96,13 1.13,15 0.86,17 0.97,28 0.36,31 0.25,34 0.7}
{4 0.63,5 0.37,28 0.32,37 student}
```

```
{5 0.43,28 0.32,31 0.21,37 student}
{23 0.88,36 0.9,37 project}
{1 0.64,16 0.41,23 0.88,28 0.32,31 0.21,36 1,37 project}
{5 0.41,20 1.07,31 0.21,37 student}
{2 0.99,6 0.45,7 0.89,13 1.01,14 0.91,15 0.86,17 0.97,19 0.74,31 0.21,34 0.72,35 0.58}
{12 0.75,16 0.33,21 0.92,23 0.88,28 0.36,30 1.82,31 0.24,37 project}
{37 project}
{5 0.44,28 0.36,31 0.21,37 project}
{2 0.77,5 0.37,6 0.55,14 0.91,16 0.37,19 0.87,21 1.08,23 0.88,31 0.21,35 0.64}
{37 project}
{1 0.64,3 0.8,7 0.8,16 0.41,25 0.59,26 0.63,28 0.32,31 0.24,33 1.5,34 0.7,37 faculty}
{6 0.45,14 1.1,16 0.33}
{28 0.32,37 project}
{0 0.83,3 0.89,4 0.63,12 0.75,16 0.42,22 0.82,25 0.59,28 0.42,31 0.24,37 project}
{4 0.74,15 0.77,23 0.88,33 1.35,37 project}
{1 0.64,2 1.04,4 0.63,5 0.41,6 0.61,11 1.28,13 1.18,15 0.95,16 0.33,19 0.91,21 1.11,23 0.98,25
0.59,26 0.63,32 0.98,35 0.74}
{4 0.63,9 1.13,16 0.33,27 1.52,34 0.6,35 0.58,37 project}
{0 0.93,1 0.72,3 0.89,5 0.37,16 0.37,24 0.72,25 0.72,28 0.38,31 0.27,34 0.6,37 faculty}
{0 0.83,1 0.75,3 0.89,4 0.74,5 0.43,8 1.4,16 0.41,19 0.91,21 0.92,25 0.59,28 0.4,31 0.27,34
0.72,37 faculty}
{0 0.93,1 0.64,12 0.75,24 0.72,25 0.59,26 0.7,28 0.36,31 0.24,36 1,37 faculty}
{}
```

```
{37 student}
{5 0.41,6 0.45,18 0.76,22 0.82,31 0.21,32 0.81,35 0.58,37 student}
{1 0.64,4 0.63,16 0.33,24 0.72,25 0.69,28 0.32,31 0.21,37 faculty}
{2 0.95,7 0.89,13 1.01,34 0.6}
{5 0.41,6 0.52,21 0.92,31 0.21,35 0.68}
{0 1.07,1 0.85,3 0.89,5 0.47,16 0.39,18 0.84,20 0.91,22 0.91,25 0.59,26 0.63,28 0.39,31 0.26,32 0.81,37 faculty}
{1 0.72,6 0.45,7 0.96,15 0.77,16 0.33,21 0.92,25 0.59,28 0.32,34 0.7}
{3 0.8,4 0.63,5 0.41,25 0.59,28 0.32,31 0.21,37 faculty}
{5 0.44,18 0.76,22 0.82,32 0.81,37 student}
{2 0.77,6 0.52,7 1.03,11 1.26,14 0.91,15 0.77,16 0.4,19 0.74,31 0.24,34 0.72,35 0.72}
{0 1,1 0.72,3 0.8,19 0.89,25 0.71,28 0.4,31 0.26,34 0.7,37 faculty}
{12 0.75,24 0.72,25 0.59,26 0.63,28 0.32,31 0.24,36 1,37 faculty}
{4 0.63,16 0.41,23 0.88,28 0.32,31 0.21,36 1,37 project}
{5 0.37,18 0.76,20 0.91,22 0.82,37 student}
{3 0.8,16 0.37,25 0.59,28 0.39,37 project}
{0 1.04,1 0.87,26 0.63,27 1.52,28 0.4,31 0.27,37 project}
{28 0.32,31 0.24,37 project}
{0 1,3 0.93,7 1,16 0.43,19 0.87,20 0.91,24 0.88,25 0.59,26 0.78,27 1.52,28 0.41,31 0.29,34 0.78,37 faculty}
{1 0.64,2 0.99,5 0.47,6 0.6,9 1.13,11 1.34,13 1.22,14 1.02,15 0.98,17 1.21,19 1.02,21 1.08,23 0.88,24 0.84,31 0.26,32 1.03,35 0.68}
{2 0.77,5 0.45,6 0.54,13 1.01,15 0.93,19 0.74,21 0.92,25 0.59,31 0.21,32 0.95}
```

```
{3 0.96,4 0.79,5 0.43,6 0.45,12 0.75,16 0.39,19 0.74,24 0.84,25 0.59,26 0.63,28 0.41,31 0.26,32
0.81,34 0.6,37 faculty}
{5 0.45,25 0.59,26 0.74,37 faculty}
{2 0.96,5 0.43,6 0.45,15 0.9,19 0.74,32 0.95}
{5 0.41,6 0.45,9 1.13,23 1.03,26 0.63,29 1.1,35 0.68,37 student}
{2 1.03,4 0.63,5 0.41,6 0.56,11 1.23,13 1.13,15 0.86,16 0.37,17 0.97,19 0.82,21 0.92,32 1.03}
{2 0.77,6 0.45,7 0.96,15 0.86,16 0.37,17 0.97,26 0.63,31 0.21,34 0.67,35 0.64}
{1 0.64,3 0.8,4 0.63,6 0.45,7 0.96,12 0.75,16 0.33,24 0.72,25 0.59,26 0.7,28 0.39,31 0.26,34
0.72,35 0.58,37 faculty}
{4 0.71,5 0.43,6 0.5,16 0.33,20 1.07,29 1.1,37 student}
{1 0.79,5 0.37,6 0.45,14 0.91,16 0.33,19 1.02,25 0.59}
{5 0.41,6 0.45,9 1.13,12 0.75,18 0.76,22 0.82,24 0.72,25 0.59,26 0.63,28 0.32,29 1.1,31 0.21,32
0.81,37 faculty}
{7 0.96,11 1.02,13 1.01,14 1.02,19 0.87,34 0.67,35 0.64}
{5 0.43,18 0.84,22 0.82,26 0.63,31 0.21,32 0.81,37 student}
{5 0.41,29 1.1,31 0.21,37 student}
{5 0.47,6 0.45,12 0.75,16 0.33,18 0.84,22 0.82,24 0.72,31 0.21,32 0.81,35 0.64,37 student}
{2 0.96,5 0.37,6 0.55,14 1.14,16 0.37,21 1.08,31 0.21}
{2 0.77,6 0.45,7 0.89,13 1.01,14 0.91,15 0.93,17 0.97,19 0.82,21 0.92,34 0.6}
{4 0.63,6 0.45,25 0.65,28 0.32,31 0.24,34 0.6,37 faculty}
{2 0.9,6 0.52,7 0.89,13 1.13,14 0.91,15 0.77,16 0.39,19 0.74,25 0.59,34 0.6,35 0.64}
{0 0.83,1 0.64,6 0.5,8 1.4,12 0.75,16 0.33,24 0.72,25 0.59,26 0.76,28 0.4,31 0.27,36 1.05,37
faculty}
```

```
{1 0.64,3 0.8,7 0.8,12 0.75,15 0.77,24 0.72,25 0.59,28 0.36,31 0.25,34 0.72,37 faculty}
{5 0.49,6 0.45,11 1.02,18 0.84,24 0.72,28 0.32,31 0.25,37 student}
{2 1,4 0.63,6 0.57,7 1.05,11 1.28,14 0.91,15 0.86,17 0.97,19 0.82,21 0.92,31 0.21,34 0.67,35 0.7}
{2 0.77,6 0.45,7 0.93,14 0.91,15 0.86,17 0.97,34 0.6,35 0.58}
{5 0.41,37 project}
{0 0.97,3 0.89,4 0.63,5 0.43,8 1.4,19 0.91,25 0.59,28 0.38,31 0.27,37 faculty}
{5 0.44,6 0.54,20 1.07,27 1.52,29 1.22,31 0.25,35 0.58,37 student}
{37 faculty}
{0 0.97,4 0.63,5 0.43,16 0.39,18 0.84,22 0.91,24 0.72,25 0.59,28 0.42,31 0.21,32 0.81,37 faculty}
{5 0.41,18 0.76,22 0.82,31 0.24,35 0.64,37 project}
{0 1.07,3 0.93,4 0.78,16 0.47,25 0.65,28 0.44,31 0.32,34 0.67,35 0.73,37 project}
{1 0.64,5 0.46,12 0.83,16 0.33,18 0.76,22 0.82,25 0.65,28 0.36,31 0.26,32 0.81,37 student}
{2 0.77,5 0.44,6 0.45,16 0.33,20 0.91,23 0.88,31 0.24,37 student}
{2 1.02,6 0.52,7 1.03,11 1.14,14 0.91,15 0.86,16 0.33,17 0.97,19 0.74,28 0.32,31 0.21,34 0.72,35
0.64}
{1 0.64,3 0.8,5 0.46,18 0.76,26 0.63,28 0.4,31 0.24,37 faculty}
{5 0.37,18 0.76,22 0.82,29 1.1,37 student}
{7 0.8,16 0.33,23 0.98,33 1.35,34 0.77,37 project}
{5 0.44,11 1.02,16 0.33,18 0.76,20 1.02,22 0.82,31 0.25,37 student}
{1 0.72,6 0.5,7 0.96,16 0.33,17 0.97}
{6 0.45,16 0.33,23 0.88,28 0.32,31 0.24,34 0.7,37 project}
{1 0.72,3 0.8,4 0.76,5 0.41,6 0.52,12 0.75,16 0.37,24 0.91,25 0.59,26 0.63,28 0.41,31 0.25,35
0.58,36 0.9,37 faculty}
```

```
{0 0.83,9 1.13,12 0.75,16 0.39,26 0.63,31 0.21,36 1,37 project}

{2 0.77,6 0.55,7 1,13 1.01,14 0.91,15 0.77,17 0.97,21 0.92,31 0.21,34 0.7,35 0.7}

{2 0.77,6 0.5,7 0.89,11 1.26,13 1.27,14 1.2,21 0.92,34 0.67,35 0.58}

{4 0.63,6 0.5,17 0.97,19 0.87}

{37 faculty}

{5 0.44,6 0.45,16 0.33,18 0.76,21 1.03,22 0.82,24 0.72,28 0.36,31 0.21,37 student}

{5 0.43,16 0.41,37 project}

{1 0.64,12 0.75,16 0.33,20 0.91,28 0.39,31 0.21,34 0.7,37 faculty}

{3 0.8,6 0.45,24 0.72,25 0.59,26 0.63,28 0.36,31 0.26,36 1,37 faculty}

{1 0.75,4 0.63,16 0.33,26 0.63,28 0.36,36 0.9,37 faculty}

{5 0.44,12 0.75,18 0.76,22 0.82,24 0.72,31 0.21,32 0.81,37 student}

{29 1.1,31 0.21,37 student}

{4 0.63,16 0.37,28 0.32,37 project}

{5 0.41,6 0.45,13 1.22,14 1.19,15 0.86,16 0.33,17 0.97,19 1.04,32 0.95}

{5 0.44,18 0.84,20 0.91,22 0.82,24 0.72,28 0.32,31 0.24,37 student}

{31 0.25,34 0.7,37 project}

{6 0.5,14 1.1,16 0.33,21 0.92,23 0.88,32 0.95,35 0.68}

{5 0.37,12 0.75,18 0.84,22 0.91,25 0.59,31 0.21,32 0.81,37 faculty}

{2 0.86,6 0.45,7 0.89,11 1.02,15 0.9}

{5 0.37,6 0.45,15 0.77,32 0.81,37 student}

{1 0.64,6 0.45,12 0.75,15 0.77,16 0.42,23 0.88,25 0.59,26 0.7,31 0.21,36 1,37 faculty}

{4 0.63,16 0.33,37 project}

{6 0.5,13 1.13,16 0.33}
```

```
{1 0.64,4 0.78,6 0.45,12 0.75,16 0.37,24 0.72,25 0.59,26 0.63,28 0.38,31 0.21,36 0.9,37 faculty}
{0 0.83,5 0.41,18 0.76,22 0.82,31 0.21,37 student}
{4 0.63,5 0.45,6 0.52,16 0.33,20 1.02,23 1.11,28 0.38,31 0.21,32 0.9,35 0.58,37 student}
{3 0.8,12 0.75,25 0.59,31 0.24,34 0.7,37 faculty}
{5 0.46,18 0.76,20 0.91,22 0.82,24 0.72,25 0.59,31 0.26,32 0.81,37 student}
{12 0.75,24 0.72,25 0.59,28 0.32,31 0.21,34 0.7,37 faculty}
{7 0.98,31 0.21,34 0.67,35 0.68}
{0 0.97,1 0.79,2 0.77,3 0.8,4 0.63,5 0.41,8 1.4,16 0.37,18 0.76,19 0.82,22 0.82,25 0.59,26 0.7,28 0.39,31 0.24,34 0.6,37 faculty}
{1 0.72,4 0.63,7 0.8,16 0.33,20 0.91,23 0.88,24 0.8,26 0.63,27 1.7,28 0.39,31 0.26,34 0.78,37 project}
{37 faculty}
{3 0.8,12 0.75,23 0.88,28 0.38,30 1.82,31 0.24,37 project}
{5 0.43,6 0.45,20 0.91,28 0.32,29 1.1,31 0.24,37 student}
{5 0.37,16 0.37,28 0.38,37 project}
{5 0.41,12 0.75,18 0.76,19 0.74,22 0.82,28 0.32,31 0.21,32 0.81,37 student}
{5 0.46,18 0.84,22 0.91,26 0.63,31 0.24,32 0.81,37 student}
{1 0.64,6 0.5,7 0.89,11 1.02,14 0.91,16 0.33,17 0.97,21 0.92,34 0.7}
{0 1.04,3 0.8,6 0.45,8 1.4,12 0.75,24 0.72,25 0.59,26 0.74,28 0.41,31 0.26,36 1,37 faculty}
{4 0.71,16 0.39,28 0.32,37 project}
{1 0.64,12 0.75,16 0.39,24 0.72,25 0.59,26 0.63,28 0.38,31 0.25,36 1,37 faculty}
{3 0.89,6 0.45,12 0.75,21 0.92,24 0.72,25 0.59,26 0.63,28 0.38,31 0.25,36 1,37 faculty}
{}
```

```
{1 0.72,5 0.44,6 0.45,12 0.75,18 0.76,25 0.59,28 0.38,31 0.24,32 0.81,37 faculty}
{2 0.77,6 0.45,7 0.89,13 1.01,14 0.91,15 0.86,17 0.97,19 0.82,21 0.92,34 0.67}
{28 0.4,37 project}
{0 0.83,4 0.63,5 0.44,9 1.36,16 0.39,24 0.72,26 0.63,28 0.32,31 0.21,37 project}
{0 0.93,1 0.72,4 0.63,12 0.75,16 0.33,24 0.72,25 0.59,26 0.63,28 0.38,31 0.21,33 1.35,36 1,37 faculty}
{3 0.8,4 0.71,28 0.36,37 project}
{5 0.44,6 0.45,18 0.76,20 0.91,29 1.1,31 0.24,37 student}
{1 0.64,3 0.93,5 0.41,8 1.4,16 0.4,25 0.59,28 0.41,31 0.24,37 faculty}
{16 0.33,31 0.21,35 0.58,37 project}
{0 0.93,16 0.37,23 0.88,26 0.63,28 0.36,31 0.21,37 project}
{3 0.89,24 0.72,25 0.59,26 0.7,28 0.38,31 0.25,34 0.7,37 faculty}
{3 0.8,4 0.71,12 0.75,16 0.37,24 0.72,25 0.59,28 0.39,31 0.25,34 0.7,35 0.58,37 faculty}
{5 0.37,6 0.52,21 1.08,31 0.21,33 1.35,35 0.58}
{5 0.37,18 0.76,20 0.91,22 0.82,37 student}
{2 0.77,5 0.37,15 0.86,32 0.9,35 0.58}
{1 0.81,2 0.77,12 0.75,24 0.72,25 0.59,26 0.63,28 0.36,31 0.24,33 1.35,36 1,37 faculty}
{5 0.37,6 0.52,11 1.02,21 0.92,31 0.21}
{5 0.37,37 project}
{0 0.83,24 0.72,25 0.59,28 0.36,31 0.24,37 faculty}
{6 0.45,19 0.82,31 0.21,36 1,37 faculty}
{0 0.93,4 0.63,16 0.33,28 0.32,30 1.82,37 project}
{1 0.64,2 0.77,28 0.39,31 0.24,34 0.67,37 project}
```

```
{2 0.77,4 0.71,7 0.93,9 1.13,16 0.37,23 0.98,26 0.63,28 0.38,34 0.6,37 project}
{1 0.64,2 0.77,5 0.41,6 0.57,14 1.07,15 0.86,16 0.33,17 1.08,19 0.74,21 1.17,23 0.88,32 0.81,35 0.58}
{5 0.47,6 0.45,20 1.02,29 1.1,31 0.21,35 0.71,37 student}
{5 0.41,16 0.37,28 0.32,37 project}
{4 0.74,20 0.91,26 0.63,28 0.32,37 project}
{0 0.83,1 0.75,3 0.8,12 0.75,24 0.72,25 0.59,26 0.74,28 0.32,31 0.24,34 0.6,36 1,37 faculty}
{1 0.64,4 0.74,6 0.45,10 1.7,23 0.88,26 0.63,28 0.38,31 0.24,35 0.58,36 0.9,37 project}
{0 0.93,1 0.64,4 0.74,5 0.37,6 0.5,12 0.75,16 0.33,19 0.74,26 0.63,28 0.32,35 0.68,37 faculty}
{1 0.72,3 0.8,12 0.75,24 0.72,25 0.65,26 0.63,28 0.38,29 1.1,31 0.24,34 0.7,37 faculty}
{2 0.86,6 0.5,11 1.02,21 0.92,32 0.9}
{37 student}
{37 student}
{1 0.75,7 0.8,9 1.13,23 0.98,28 0.4,31 0.24,34 0.72,35 0.58,37 project}
```

APPENDIX I: THE 30-30 FILE AFTER FEATURE SELECTION BY CFS IN ARFF FOR BINARY CLASS MIC

```
@relation 'ga-weka.filters.unsupervised.attribute.ReplaceMissingValues-
weka.filters.unsupervised.instance.Randomize-S42-weka.filters.unsupervised.attribute.Remove-V-
R201,279,283,293-295,297,300,308-
309,312,317,327,330,332,334,341,343,368,370,373,390,410,421,440-
441,443,457,480,499,508,513'

@attribute B73 numeric
@attribute C23 numeric
@attribute C27 numeric
@attribute C37 numeric
@attribute C38 numeric
@attribute C39 numeric
@attribute C41 numeric
@attribute C44 numeric
@attribute C52 numeric
@attribute C53 numeric
@attribute C56 numeric
@attribute C61 numeric
@attribute C71 numeric
@attribute C74 numeric
@attribute C76 numeric
```

```
@attribute C78 numeric
@attribute C85 numeric
@attribute C87 numeric
@attribute C112 numeric
@attribute C114 numeric
@attribute C117 numeric
@attribute D6 numeric
@attribute D26 numeric
@attribute D37 numeric
@attribute D56 numeric
@attribute D57 numeric
@attribute D59 numeric
@attribute D73 numeric
@attribute D96 numeric
@attribute D115 numeric
@attribute D124 numeric
@attribute class {Normal,Severe}

@data
2379.7201,-0.6626,3.4477,0.2186,1.5709,0.3187,1.3282,-0.5674,2.2503,-0.2118,1.0047,-
0.5925,0.7219,0.6257,-0.7568,3.1497,0.4032,-0.0529,-0.0304,4.252,-
0.831,3.1355,40.712,4.3082,3.1442,2.9863,2.7151,1.5916,5.8317,2.0753,1.9867,Severe
```

```
19.4514,-0.6124,-0.3756,-1.0321,-0.794,0.1547,-0.1534,1.2979,-0.7972,-
0.2313,1.0931,0.4386,0.7884,1.7734,0.3859,0.3892,-1.0551,-0.4307,-0.2179,4.2667,-
0.8466,5.1078,3.4183,4.9964,6.7021,10.9072,43.5552,7.186,12.0648,1.8262,1.8665,Severe
710.5976,-0.1545,0.8446,-0.278,-1.518,-0.2616,0.0639,-0.7627,-
0.9696,1.9975,1.363,0.1061,2.1634,3.9813,3.1245,0.6869,-0.6175,2.2975,-0.0789,4.7128,-
0.6189,1.7837,2.8961,2.0176,8.1167,5.6945,2.7509,10.2183,17.4368,2.1189,1.3099,Severe
38.9888,-0.3483,0.0503,-1.2937,-1.1629,-0.2358,-0.3435,-0.2377,-1.0755,-0.6749,-0.9862,-
1.002,0.0838,-0.4112,-1.0593,-1.2071,-1.5669,-0.0077,5.5299,1.9964,-
1.0882,34.8261,2.4186,5.3367,4.0939,2.5789,5.2013,2.4281,32.6888,1.1632,3.1083,Normal
50.3969,0.1734,-0.0354,-0.0763,0.2994,0.0483,-0.0284,-0.7516,-1.4152,0.742,1.0762,-
0.6408,0.8401,-0.5925,-0.6801,-0.4892,1.1526,-0.0546,-0.1883,4.2454,-
0.4947,4.44,3.9673,4.4718,4.1076,8.8918,3.0777,3.0824,7.3819,1.8043,1.8522,Severe
193.8003,0.392,-0.0611,-1.6276,0.2968,0.9283,2.5288,-0.4559,-1.4032,-0.2788,1.2959,0.6028,-
0.2934,0.1135,0.2354,0.3115,-1.0654,-0.0993,-0.3125,4.2457,-
0.9169,4.975,2.9615,5.77,4.8758,2.7552,2.0753,6.63,14.6593,1.7265,1.9287,Severe
49.2642,-0.2159,-0.1165,-1.6296,-1.6429,-0.9953,-0.0844,-0.5942,-1.504,-1.5811,0.0289,-
0.3032,-0.3767,-0.2848,-0.4555,-0.2622,-0.7009,-0.4792,-0.3324,4.2438,-
0.9384,4.9061,2.407,6.3557,3.1815,3.5419,1.9818,2.5575,7.9903,1.6835,1.9297,Normal
91.7712,0.3317,0.0551,-0.3628,-0.1912,-0.8494,0.6157,-0.4601,-0.4959,-1.7844,-0.2166,-0.7513,-
0.5296,-0.1961,-1.2642,-1.079,-1.0969,-0.7958,-0.3279,4.2463,-
0.9639,4.033,2.2477,3.9123,3.3556,3.1341,2.775,2.7555,9.7646,1.6915,1.9732, Normal
```

```
36.3484,0.3742,1.7462,0.2141,-0.1883,-0.0324,-0.3912,0.5137,-0.5775,-0.7832,-0.5886,0.1711,-
0.4264,1.1768,1.2423,0.7708,0.3812,0.6031,-0.2818,4.2486,-
0.8399,3.2876,2.3697,2.9908,3.0742,2.5982,2.502,3.2207,11.8226,1.7883,1.9061,Severe
1161.857,0.0348,0.821,0.3778,1.1312,1.9292,-
0.1028,1.1046,0.4351,0.8319,0.8358,0.758,0.8337,0.3136,0.4116,0.2537,0.9026,0.6306,-
0.1272,4.8342,-
0.5656,1.6065,4.5861,2.2942,3.358,2.7527,2.7357,2.58,2.6373,2.2764,1.3775,Severe
89.4285,-0.1736,-0.4855,-0.2746,0.0336,0.0563,-0.1939,-0.4477,-0.4128,2.1897,-
0.4775,0.1429,2.7842,-0.3482,0.3983,0.0304,-0.3866,0.8034,0.0777,5.2822,-
0.3024,1.6143,3.6729,2.416,2.9444,2.2621,2.5742,1.8816,21.0512,2.373,1.3085,Severe
12.0049,0.4803,-0.8327,-0.0211,0.4763,-0.4264,0.1972,-0.4001,-1.1565,0.596,0.0852,-
0.1985,0.0488,-0.3845,0.0392,-0.9257,-0.8952,1.829,-0.3361,4.2454,-
0.9247,5.0622,7.5776,2.8995,2.8056,2.7536,6.3408,2.7871,14.7981,1.6749,1.9726,Severe
115.8672,1.2142,-0.9658,-1.8537,-0.1219,-0.052,2.3305,0.4257,-0.5208,-0.4763,-0.253,0.4714,-
0.0451,-0.6204,-0.2391,0.1298,-0.2692,-0.2203,-0.2706,4.2635,-
0.9344,4.5297,2.3851,7.8703,2.8328,2.4317,2.9632,2.7949,8.2535,1.7454,1.9546, Normal
66.4113,-0.3613,-0.4394,-0.8483,-1.09,-0.4583,-0.3634,0.0703,-0.3454,-0.8868,0.1839,-0.6845,-
0.2515,-0.4322,-1.0269,-0.6112,-0.1038,0.1118,-0.6992,2.6737,-
0.995,69.3589,3.7467,4.1924,2.218,2.2299,2.2234,1.9708,10.3706,1.5246,2.1198, Normal
123.1021,0.2141,-0.2541,-0.6731,-0.7953,0.6242,-0.0676,-1.2359,-0.0214,-1.0654,0.3984,-
0.7175,-0.328,-0.4904,-1.6389,-0.6586,-1.1298,-0.6626,5.5837,2.7734,-
0.9938,11.7132,3.8981,5.1618,2.2864,1.9957,3.0243,1.9122,32.2504,1.3738,2.6468, Normal
```

```
81.3732,-0.2788,-0.2628,-0.5871,0.0634,-0.0412,-0.206,-0.2645,0.1142,-0.7063,0.2561,-1.0447,-
0.4936,-0.6569,-1.0447,0.1413,-1.0482,0.6402,4.604,2.6259,-
1.3762,75.7159,4.2131,4.3203,2.1957,2.4543,4.2333,2.2807,32.6409,1.4982,2.9752, Normal
60.3728,-0.2939,-0.2528,-0.5597,-1.1065,-0.6842,-0.6545,-0.3228,0.3004,-0.7667,0.1924,-
0.7206,-0.0997,-0.6362,-1.3758,-0.1152,-0.4647,0.3681,5.7259,2.959,-
0.7485,64.7046,3.1305,3.2256,1.8784,2.0522,2.3431,2.5644,33.4927,1.5241,2.5713, Normal
35.0828,-0.8396,0.0824,-1.4939,-1.4148,-0.8436,-0.9883,-0.1686,-1.9358,-1.1241,-0.375,-
1.1369,-0.404,-0.1176,-1.5921,-0.8484,-1.1248,-
0.6815,5.7157,2.6771,0.6612,84.8044,2.6919,5.0809,2.4268,2.7941,3.8419,1.9543,33.6619,1.714
6,2.196, Normal
2181.1987,1.0397,-0.8959,3.4199,4.2593,-0.215,0.2162,0.0513,2.7853,1.5876,2.9646,-
0.4012,0.3646,1.5879,5.2006,-0.7078,1.4381,2.5548,-0.3072,5.6393,-
0.5662,1.5104,5.0818,17.2919,13.2675,2.9728,14.2195,7.3158,16.0318,2.348,1.4309,Severe
10.9721,-0.3535,-0.8222,-0.3085,0.3073,-0.064,0.6097,-0.4084,-1.0771,0.0041,-0.1123,-1.2947,-
0.3573,-0.435,-1.2588,-0.6261,-0.7564,-0.4062,-0.3283,4.268,-
0.9277,4.7969,3.6302,3.4304,2.4646,2.6623,3.3709,2.6642,13.7409,1.7071,1.949, Normal
51.3599,-0.9123,-0.0538,-1.2168,-1.1862,-0.0371,0.8534,-0.4281,-0.5622,0.5522,0.8183,-
0.6833,0.3993,0.3869,-0.2011,0.2143,-0.0906,-
0.7217,5.0853,7.8801,6.2369,3.9323,4.0222,3.8845,4.1352,3.1105,2.5778,2.666,31.8639,47.2109,
56.049, Normal
1088.0221,-0.1399,-0.0018,0.3936,0.2494,0.8766,0.9251,0.3702,-0.5194,-0.549,0.9356,0.3705,-
0.414,2.2891,3.412,0.7561,-0.945,0.6966,-0.268,5.0499,-
0.0113,1.6262,2.4432,2.1821,4.1975,6.3959,3.1771,8.6308,14.0867,4.063,3.0842,Severe
```

```
38.565,-0.2239,0.1139,-1.4343,-1.9926,-1.9765,-0.6859,-0.3156,-1.2826,-1.4451,-0.5065,-1.061,-
1.1879,-0.094,-0.7508,-0.6276,-1.0117,0.1291,5.56,2.2148,-
1.2886,50.6675,2.105,4.3421,3.4391,2.8702,2.5864,2.8912,32.3754,1.2822,2.9452, Normal
149.962,0.2627,0.5115,-0.1541,0.052,-0.189,4.1263,-0.3084,0.6757,2.5948,1.9179,-
0.2294,2.0679,-0.2788,-0.3975,-0.89,-
1.6012,2.7441,0.1007,4.9898,0.081,1.4709,3.4571,2.718,12.3605,5.4873,3.0955,1.8797,6.4206,3.
0559,1.6775,Severe
187.8146,-0.2513,-0.6511,-0.459,0.3674,0.0856,-0.9475,-0.0856,-0.4096,0.0644,-0.2722,-0.0133,-
0.3715,-0.1617,0.3176,-0.5071,-1.2673,-0.325,-0.339,4.2407,-
0.8835,4.7634,4.4235,3.2449,2.2229,1.8051,3.3345,2.3575,12.2064,1.6738,1.9413, Normal
137.6009,1.8877,1.5547,2.2249,2.3571,2.3237,1.8123,3.299,1.5248,1.4905,0.5112,0.4067,-
0.0303,1.6577,0.869,0.2977,0.1121,1.2046,-0.286,4.2832,-
0.8573,3.4677,9.6974,8.9684,3.652,6.4338,14.5538,8.7348,9.2752,1.694,1.9472,Severe
24.9514,-0.6534,0.9015,-0.4855,3.3252,1.9099,0.7731,0.4737,-0.6143,1.0384,0.9633,-0.2903,-
0.1397,-0.7214,2.6965,1.835,-0.0581,-1.1011,-0.1807,4.2632,-
0.6398,5.0127,2.1816,3.6358,5.4521,3.917,6.7388,10.3991,11.655,1.7399,1.8482,Severe
23.9143,-1.1375,-0.7701,0.0171,0.1928,-0.6209,-0.8237,0.0177,-0.0038,-0.6769,0.6734,-0.3952,-
1.1733,-0.8007,-0.3996,-0.7315,-1.4135,-1.0241,5.5781,2.6877,-
1.3999,30.1534,2.592,3.0704,4.0618,3.0778,2.3812,3.6807,33.0734,1.3124,2.7443, Normal
102.8658,0.0062,-0.3862,-1.4555,-1.0739,-0.5062,0.0033,-0.8468,-1.8145,-0.9519,-0.6107,-
1.2366,-0.3456,-1.0174,-0.7086,-0.8366,-0.6808,-0.5549,-0.3014,4.2553,-
0.961,4.201,2.8894,4.5768,4.1523,3.7087,3.5157,2.6496,10.5169,1.7099,1.9445, Normal
```

```
84.8089,-0.5522,-0.668,-0.8031,-1.4997,-1.2493,-0.4943,-0.6493,-0.9104,0.2792,-0.466,-
0.2425,0.4061,0.2157,-0.7704,-1.2679,-0.8343,0.9519,5.5901,2.9111,-
0.3641,54.672,3.4636,3.5095,2.3288,2.0067,2.7093,1.9812,31.6057,1.6008,2.4552, Normal
63.0036,-0.7583,-0.0802,0.0435,-1.2548,-1.8829,-0.6912,-0.5102,-1.3267,-0.9332,0.2931,-
0.9125,-0.1733,-0.428,-1.5199,-0.9758,-0.8872,-0.0162,-0.2892,4.2432,-
0.8536,4.8557,3.9455,2.9049,2.2351,2.1524,2.7773,3.4398,8.8669,1.6747,1.894, Normal
83.9558,-0.2485,-0.6495,-0.176,0.5196,-0.6191,-0.4555,-0.5453,-0.5264,-0.3253,-
0.452,0.3449,0.2135,-0.3452,-0.5919,0.3705,-1.1521,0.0133,-0.3231,4.2553,-
0.7433,4.7261,3.122,3.105,2.6527,2.0492,3.8181,3.2988,12.2307,1.7156,1.9606, Normal
34.3387,-0.078,-0.4784,0.5914,-0.8035,-0.6121,-0.1221,-0.3244,-0.4431,-1.001,-0.437,0.1249,-
0.9669,-0.9293,-0.3096,-0.8106,-0.8345,-0.3016,-0.307,4.254,-
0.9347,4.2326,2.4317,2.9641,3.7565,2.3347,1.9896,3.4803,11.3485,1.7371,1.9318, Normal
65.336,-0.3604,-0.6669,-0.2892,-1.1253,-0.5515,-0.1972,-1.1458,-0.03,-0.9558,-0.2721,-0.9064,-
0.4897,-0.8067,-1.7924,-0.6622,-1.3452,-0.0509,-0.3213,4.2463,-
0.8236,5.1322,3.7235,3.5251,2.7468,1.7218,2.8851,1.8653,7.9109,1.6994,1.9404, Normal
550.5459,-0.0131,0.7556,-0.1827,0.1158,-0.4462,-0.2725,-1.8039,-0.1287,-0.5917,1.2299,-
0.1634,2.4108,3.8873,-0.4182,-0.1808,1.0493,-0.1361,-0.3082,4.249,-
0.9282,4.8414,2.8699,1.9433,5.2564,4.5019,2.9813,15.8066,11.4027,1.6859,1.968,Severe
37.7381,-0.3242,-0.3734,-0.6847,-0.7931,-0.8306,-0.9465,-1.7892,-1.3805,-0.5185,0.2058,-
1.173,-0.4674,0.2421,-0.1806,-0.0998,-1.738,0.8914,-0.2631,4.2491,-
0.7921,4.6755,3.2441,4.2568,1.5574,2.0579,3.9185,2.7408,12.7416,1.7124,1.9427, Normal
```

```
178.236,0.2111,-0.0644,-0.5527,1.0656,1.6272,-0.9468,0.2592,-0.1517,-0.6305,0.5616,-
0.3539,1.1443,0.2583,-0.6312,-0.8231,-1.0641,1.1865,-0.324,4.2456,-
0.693,5.0368,2.2264,3.4175,3.0946,3.0915,4.171,2.1569,12.237,1.7161,1.9189, Normal
80.0137,2.3916,-0.7946,-0.0069,-0.0332,2.6091,2.2169,-1.0089,-1.5399,-0.0331,0.6175,-0.0281,-
0.1745,-0.2865,0.2081,-0.6689,-1.4536,1.7363,-0.0877,5.0888,-
0.561,1.7198,3.6152,3.2213,2.5023,2.9463,12.2379,2.2283,26.0071,2.2527,1.2594,Severe
886.8032,1.7906,1.2571,1.1382,2.1103,-0.0051,0.6118,-
0.2823,0.8209,0.5567,0.4444,1.1718,0.857,-0.4567,-0.1198,0.4379,1.7139,0.9964,-
0.1097,5.2917,-
0.3437,1.936,1.9875,4.4139,2.3219,2.0656,4.0103,2.9572,2.9959,2.4867,1.353,Severe
17.6254,-0.0452,-
0.0445,1.5506,2.7762,4.1237,1.921,0.1061,0.4564,2.7085,1.2131,0.6572,1.5161,-
0.5898,1.8001,1.5265,-0.3359,2.3311,-0.0038,4.7451,-
0.3687,1.4647,2.2848,5.317,4.9484,3.6028,2.9741,3.5237,13.6156,2.2861,1.4539,Severe
40.0209,-0.2638,-0.4086,-0.3804,-0.461,-0.6347,-0.3291,-0.1714,0.254,0.2486,1.0068,-
0.3225,0.4969,-0.3676,-0.8414,-1.2853,-0.0063,0.0972,-0.2969,4.2423,-
0.9425,4.0565,3.0447,2.8938,3.3633,3.1233,2.8947,1.9574,12.4089,1.6685,1.8914, Normal
96.8771,-0.1174,-0.8023,-0.8592,-0.8133,-1.18,-0.182,-0.964,-0.3957,0.1754,-0.1753,-0.67,-
0.4544,0.0781,-0.2708,-1.9616,-0.7627,-0.5416,5.5741,2.8544,-
0.4648,54.0081,3.1163,3.862,2.5029,2.179,2.8533,2.4925,30.2316,1.5374,2.3322, Normal
23.9615,-2.17,0.1368,0.3369,0.6802,0.3885,-1.3321,-0.3741,0.9735,-0.2049,-0.0479,-1.3435,-
1.0691,-0.9185,-0.9243,-1.3084,0.1352,-0.3348,-0.0964,4.2997,-
0.5135,1.3773,2.9669,2.8608,2.977,3.8147,6.6784,3.6919,8.7894,1.9898,1.4274,Severe
```

```
127.4792,-0.3288,-1.0372,-1.3635,-1.2838,-1.1039,-0.446,-0.2447,-2.2568,-1.2458,-0.4887,-
0.9536,0.1115,-0.2397,-0.6792,-0.9105,-0.0698,0.1598,5.6972,2.7071,-
0.5394,68.6373,2.3696,6.1165,3.0521,3.4545,5.1011,2.0242,33.1926,1.4881,2.3609, Normal
12.745,-0.0879,-0.0192,-0.0225,-0.04,1.3273,0.4664,1.5058,-0.3314,-
0.1963,0.6507,2.2551,2.147,0,2.0362,1.5576,-1.0699,-0.0121,0.0369,7.4277,-
0.2911,1.7473,2.5711,3.0354,2.669,6.375,12.7675,3.7942,25.033,3.1253,1.2569,Severe
19.0214,-0.0357,0.7911,-0.3551,-0.5659,0.0948,1.8775,2.3004,1.2188,0.8147,0.3363,1.5137,-
1.1036,0.6869,0.2615,0.9885,-0.6418,-1.5488,-0.3439,4.2442,-
0.9481,4.4991,2.6609,3.6467,2.5293,1.8555,3.0087,2.4934,14.1038,1.7149,1.9573,Severe
225.9997,-1.4477,0.1838,-1.726,-2.1067,-1.2448,-0.4701,-0.1663,-0.1657,-1.2456,-0.4953,-
0.6012,-0.1382,-0.5638,-2.0587,-0.7017,-1.6228,-1.5688,5.0223,2.8516,-
1.0139,37.3655,2.4742,5.1624,3.1241,2.1246,4.3739,1.7998,29.0509,1.4804,2.2244, Normal
32.1718,-1.1528,-0.2603,-2.166,-1.5702,-1.6733,-0.7695,-1.1817,-1.1293,-1.3871,-0.0386,-
0.9447,-1.0135,-0.0079,-0.8662,-0.1823,-1.0517,-0.9201,5.5606,2.3061,-
1.2651,44.5436,2.2382,7.8632,2.7608,3.0519,2.7018,2.6209,32.2433,1.191,2.7166, Normal
133.9518,0.1009,-0.3567,-0.248,-0.4287,-0.1419,0.2257,0.4858,-0.315,-0.542,-0.3066,-
0.1893,0.3462,1.2595,-0.0249,-0.2649,0.2339,0.4185,-0.5795,4.6362,-
0.76,1.2756,2.3225,2.6902,3.1583,2.0705,2.2989,2.1736,22.4421,1.9526,1.727,Severe
277.5424,0.634,-0.6588,2.4645,2.9092,1.4775,0.7723,1.0953,2.4718,1.6995,2.6361,-
0.5697,2.0085,1.2733,-1.2274,-0.615,1.8846,2.3638,-0.24,4.2638,-
0.9181,3.2362,5.3216,8.7367,11.8403,6.8442,3.0248,9.4885,11.6746,1.718,1.8864,Severe
```

```
2181.1987,1.0397,-0.8959,3.4199,4.2593,-0.215,0.2162,0.0513,2.7853,1.5876,2.9646,-
0.4012,0.3646,1.5879,5.2006,-0.7078,1.4381,2.5548,-0.3072,5.6393,-
0.5662,1.5104,5.0818,17.2919,13.2675,2.9728,14.2195,7.3158,16.0318,2.348,1.4309,Severe
70.78,-0.0776,-0.4829,-1.302,-0.2513,0.2099,0.0083,-0.7935,-0.6372,-0.8348,-0.3407,-
0.85,0.1253,0.4701,0.324,-1.1327,-0.6675,0.8975,5.545,2.4956,-
0.3312,72.2195,4.3742,5.7223,2.6102,2.3078,3.1691,2.6302,33.3356,1.3817,2.8594, Normal
115.7276,-0.259,-0.2802,-0.812,-0.7555,-1.2539,-0.4666,-1.7267,-1.1561,-0.9901,0.0133,-
1.2669,-0.1162,-0.5172,-1.0596,-0.7503,-1.2671,-0.9894,-0.3254,4.2443,-
0.8544,4.863,2.9114,3.4901,2.8884,4.1352,3.0065,2.0949,13.0034,1.6796,1.89, Normal
26.0925,1.9459,0.58,2.1313,1.3858,2.925,-0.7315,-0.3002,0.3146,-0.3709,0.4134,-0.5508,-
1.0624,-0.073,-0.5632,-0.7573,-1.4713,1.8044,-0.3201,4.2449,-
0.9132,4.8633,33.5744,7.4672,2.7423,1.852,3.2094,2.4646,14.1995,1.6752,1.9513,Severe
3429.3161,0.3813,-0.7439,-0.1124,0.6247,-0.7589,0.1524,0.5468,0.0179,-
0.7964,0.5682,0.505,1.3066,1.2389,-0.7029,-1.2102,-1.0963,1.4583,0.9256,4.2719,-
0.8825,4.8988,24.557,2.6734,2.5566,2.5062,3.1461,1.4038,13.0816,1.7169,1.9382,Severe
146.3948,1.2474,0.3655,-0.8471,-0.923,0.3926,1.5439,-0.6297,-1.5054,-2.5481,0.0348,-
0.4213,0.138,0.0396,-0.6787,-1.3632,-0.3307,1.2673,-0.3314,4.2441,-
0.9487,4.8108,1.8582,3.7152,1.5752,1.792,4.3893,3.5822,9.4594,1.7205,1.9678, Normal
0.0425,1.0134,0.5548,1.2527,1.1683,0.9813,1.6968,0.2573,-0.8336,1.5989,0.8123,-
17.0389,1.7821,-7.0336,-0.961,-1.3684,-0.0782,0.2255,-0.2533,3.033,-
0.8202,1.2254,2.5052,5.4601,2.9682,3.5274,128.9177,20.5465,17.9409,1.4974,1.8398,Severe
```

21.6964,-0.1928,-0.4212,-1.3029,-0.7144,-0.0956,0.3937,0.7696,-0.7786,-1.394,1.0497,-0.2451,1.385,3.686,1.1436,0.022,-0.0985,2.4577,-0.2307,4.3352,-0.6955,1.5996,2.0296,4.4834,2.8576,3.7763,2.4646,4.4256,25.0125,1.9243,1.4088,Severe

231.0135,0.9392,-0.3541,2.6729,-0.544,0.4282,-0.116,-1.7302,2.0798,-0.442,-0.1118,-0.1903,0.8542,1.9901,0.6722,-0.0642,0.0017,0.865,-0.284,4.2525,-0.9548,4.5028,4.6805,22.6672,6.2501,4.966,2.5721,8.4231,13.2684,1.6866,1.9699,Severe

1145.8664,-0.1441,0.612,0.7102,0.6608,-0.3329,0.3256,-0.1532,0.2494,1.9457,-0.1092,0.4465,0.2325,0.4786,0.2505,0.9436,2.1086,2.3922,-0.21,4.8421,-0.6334,1.9553,3.3459,2.6753,1.6016,1.8193,3.2774,1.799,28.9507,2.1002,1.4019,Severe

APPENDIX J: THE 30-50-30 FILE AFTER FEATURE SELECTION BY CFS IN ARFF FOR MULTI CLASS MIC

```
@relation 'ga-weka.filters.unsupervised.attribute.ReplaceMissingValues-
weka.filters.unsupervised.instance.Randomize-S42-weka.filters.unsupervised.attribute.Remove-V-
R14,138,142-143,172,185,188,199,202,209,240,264,281,283,288,293-296,299-
300,309,311,323,330-333,341,343,346,362,368,395,440,442,457,481,513'


@attribute A14 numeric
@attribute B10 numeric
@attribute B14 numeric
@attribute B15 numeric
@attribute B44 numeric
@attribute B57 numeric
@attribute B60 numeric
@attribute B71 numeric
@attribute B74 numeric
@attribute B81 numeric
@attribute B112 numeric
@attribute C8 numeric
@attribute C25 numeric
@attribute C27 numeric
@attribute C32 numeric
@attribute C37 numeric
```

```
@attribute C38 numeric

@attribute C39 numeric

@attribute C40 numeric

@attribute C43 numeric

@attribute C44 numeric

@attribute C53 numeric

@attribute C55 numeric

@attribute C67 numeric

@attribute C74 numeric

@attribute C75 numeric

@attribute C76 numeric

@attribute C77 numeric

@attribute C85 numeric

@attribute C87 numeric

@attribute C90 numeric

@attribute C106 numeric

@attribute C112 numeric

@attribute D11 numeric

@attribute D56 numeric

@attribute D58 numeric

@attribute D73 numeric

@attribute D97 numeric

@attribute class {1,2,3}
```

```
@data
23.6225,508.0833,629.8479,626.5977,50.2966,60.8366,34.8064,42.4507,86.8509,900.3576,1475.
3558,8.574,-0.126,-0.2628,5.5632,-0.5871,0.0634,-0.0412,-0.2424,-0.2401,-0.2645,-0.7063,-
0.0381,-0.3884,-0.6569,0.6597,-1.0447,-0.6502,-1.0482,0.6402,-
0.6527,0.2399,4.604,79.7875,2.1957,2.276,2.2807,22.049,1
21.9123,727.9562,189.1371,0.381,76.3613,458.6733,83.3269,1500.3916,74.8486,85.2269,627.42
81,-0.7311,-0.2922,0.0029,0.3505,1.5287,1.0977,1.6691,2.2421,-0.0345,-
0.9499,1.6315,1.2448,2.7647,1.8522,-1.2026,-1.4738,-1.3318,2.6855,2.5158,-0.9317,-1.8417,-
0.4488,1.2204,4.4197,13.3257,8.9949,1.5779,2
17.1185,204.9268,614.2681,150.7738,48.5564,53.236,17.4638,213.1265,599.6955,111.9843,1048
.0039,-4.5755,0.5511,0.7556,-0.4245,-0.1827,0.1158,-0.4462,0.9215,1.2012,-1.8039,-0.5917,-
0.1228,-0.6775,3.8873,-0.327,-0.4182,0.0213,1.0493,-0.1361,2.3577,0.3354,-
0.3082,4.7289,5.2564,5.4051,15.8066,1.1816,3
22.162,395.587,147.9284,0.2571,60.3591,1240.8165,30.0987,859.7036,149.6727,71.1673,67.359,
-1.0165,2.1751,0.5115,0.2402,-0.1541,0.052,-0.189,1.1747,0.2539,-0.3084,2.5948,1.4034,-
2.1462,-0.2788,-0.0158,-0.3975,-0.2324,-1.6012,2.7441,-0.0498,-
0.1993,0.1007,1.4237,12.3605,2.5939,1.8797,1.4945,3
14.8676,115.9475,463.0072,115.3298,29.8558,155.0016,21.1436,44.1408,52.2997,108.1685,468.
3737,-3.5788,-0.1376,-0.4805,-0.3457,-1.4251,-0.1634,-1.3433,-1.1003,-0.7748,-0.9545,1.424,-
0.1293,-1.9224,-0.3107,-0.9356,-1.7588,-0.0125,-2.5277,-0.8169,-1.3327,-0.7095,-
0.318,4.9149,4.6579,5.2462,2.1934,1.1878,2
```

```
12.5247,98.0416,328.1482,82.6016,440.8613,271.787,355.2357,110.4042,209.0232,183.9169,325
.6294,-0.1873,3.9806,1.5547,-0.4052,2.2249,2.3571,2.3237,1.4391,0.8594,3.299,1.4905,0.8896,-
1.3034,1.6577,0.546,0.869,0.2027,0.1121,1.2046,1.3404,0.294,-
0.286,4.9134,3.652,7.3112,8.7348,1.452,3
24.6398,962.57,237.4272,2.2474,30.1203,23.7908,21.959,18.4846,2320.1109,50.2646,70.6862,-
0.0671,0.3239,-0.8959,0.0731,3.4199,4.2593,-
0.215,0.4508,0.1561,0.0513,1.5876,1.283,0.5538,1.5879,0.7938,5.2006,-
0.7776,1.4381,2.5548,4.1431,-0.25,-0.3072,1.5706,13.2675,14.5112,7.3158,1.2811,3
10.5988,72.5296,234.9566,42.4027,20.9865,77.685,52.511,73.1024,157.906,164.3181,284.898,-
3.819,-0.7289,-0.6495,-0.3964,-0.176,0.5196,-0.6191,-0.8002,-0.3126,-0.5453,-0.3253,-0.4777,-
0.754,-0.3452,-0.7193,-0.5919,0.1037,-1.1521,0.0133,-0.6333,-0.2975,-
0.3231,4.8233,2.6527,2.2543,3.2988,1.1762,1
28.3472,1951.4131,738.2175,0.6791,42.8422,22.7331,25.5427,21.1537,30.4663,53.0761,2317.37
37,-1.0857,-0.8067,-0.3277,0.1298,2.7936,1.0828,0.3393,-0.0304,-0.9035,-0.4874,0.946,0.2328,-
0.2363,-0.5628,-0.5775,0.052,-0.76,0.0144,0.054,-0.5335,-1.2626,-
0.7052,1.3197,2.9785,4.5322,2.8801,1.6583,2
27.6485,1398.2845,534.9276,1.2254,195.7788,472.9446,89.7645,34.0817,427.5997,55.5591,714.
8153,-1.2141,1.1801,0.2861,0.1046,-0.9439,0.2562,1.0733,0.0723,-1.1467,-
1.2936,1.6881,0.2896,4.5275,2.4717,2.0932,1.1296,0.6478,2.6999,4.7435,1.9481,2.5228,-
0.1811,1.528,6.2327,11.595,2.039,1.3094,2
17.483,417.5051,638.0096,139.173,51.3617,56.5447,14.4065,126.5087,83.0517,127.5142,584.34
81,-0.5236,0.2064,-0.3541,-0.4189,2.6729,-0.544,0.4282,0.1799,-0.3625,-1.7302,-0.442,2.028,-
```

```
0.2585,1.9901,-0.3674,0.6722,-0.3855,0.0017,0.865,-0.2423,-0.4223,-
0.284,4.8226,6.2501,3.7349,8.4231,1.1812,3
15.1111,277.7351,485.8574,119.9052,43.9646,46.2595,21.3409,16.2409,38.0743,67.2839,219.62
79,1.4055,-0.4468,-0.9658,-0.3662,-1.8537,-0.1219,-0.052,3.3511,0.7082,0.4257,-0.4763,-
0.4017,-0.6534,-0.6204,0.1481,-0.2391,-0.2336,-0.2692,-0.2203,0.2453,-0.2617,-
0.2706,3.2614,2.8328,2.7012,2.7949,1.1907,1
21.9883,711.2448,162.0653,0.255,5.2662,36.4493,4.9202,11.0569,38.7129,24.4835,23.7233,-
1.2091,0.617,0.3525,0.2661,0.1448,-0.7425,-0.3996,0.0451,-0.0889,0.145,-0.9249,-
0.5353,0.021,1.779,1.3842,1.7588,4.6024,-0.3387,-
1.8109,2.4431,0.7079,0.0695,1.3804,3.5417,15.9187,18.1419,1.4945,2
21.1395,239.2501,913.4779,188.7208,131.2174,39.0381,35.4372,71.1837,24.233,185.7095,982.2
167,-0.594,-0.1743,0.2393,-0.4141,0.1311,0.5105,0.3144,0.143,-0.3203,-1.0226,0.3766,-0.0802,-
0.1958,1.0871,1.4617,0.2237,-0.169,-0.3719,-0.8584,2.0432,2.254,-
0.3224,4.7943,2.7449,4.9099,4.3364,1.1866,2
21.8858,1190.6764,183.5214,0.6302,3350.4947,1010.0218,90.3633,474.9788,39.5185,103.7718,3
79.3238,-
0.9213,0.7261,0.5548,0.5209,1.2527,1.1683,0.9813,1.11,0.524,0.2573,1.5989,0.0636,0.7434,-
7.0336,-1.5723,-0.961,-1.5567,-0.0782,0.2255,0.9289,0.9458,-
0.2533,1.1825,2.9682,8.2454,20.5465,1.8767,3
24.4877,825.6972,287.2027,0.5436,66.7221,41.9703,29.2991,415.6847,40.9988,113.1281,224.42
27,-1.2721,-0.2702,-0.4855,0.1604,-0.2746,0.0336,0.0563,-0.1111,-0.3151,-
0.4477,2.1897,1.0135,-0.2738,-0.3482,1.4859,0.3983,0.2414,-
0.3866,0.8034,2.0429,0.4643,0.0777,1.5135,2.9444,2.7181,1.8816,1.3511,3
```

```
11.1716,72.6977,263.5649,71.6649,43.234,104.6175,69.1897,60.1154,96.8556,91.433,633.9374,-
3.057,-0.1638,-0.1145,-0.3621,-0.3118,-0.3073,-0.7389,-0.3197,0.5644,-0.0406,0.3497,-0.7573,-
0.8813,-0.2825,0.2436,0.2129,-0.1205,-0.9648,-0.3562,0.3876,0.7741,-
0.3337,4.6445,2.1642,2.6939,3.0389,1.3056,2
23.8531,526.7764,627.9599,623.9649,38.9999,28.2185,34.5436,16.6284,54.8191,1086.811,1445.
1938,8.3055,-0.2373,-0.4829,5.6179,-1.302,-0.2513,0.2099,-0.0884,-0.1556,-0.7935,-0.8348,-
0.5601,-1.1282,0.4701,-0.5355,0.324,0.5706,-0.6675,0.8975,-0.2739,-
0.601,5.545,79.4872,2.6102,6.5865,2.6302,29.6597,1
27.8713,913.4601,627.3267,0.8166,7.2897,14.6188,13.4966,15.662,12.1698,64.6519,513.5627,-
1.4494,0.1828,0.1835,0.1805,-0.0817,0.3945,0.7283,0.3168,-0.1582,-0.028,2.6554,1.9739,-
0.9467,0.3412,-0.1145,-0.0791,0.1465,-0.5544,0.0261,-0.1179,-0.3535,-
0.0379,1.5288,4.6309,3.1426,2.2115,1.4377,2
26.2269,521.3077,652.938,629.8068,30.2588,52.9357,16.3886,107.5623,66.7313,872.7844,1443.
3014,6.9185,-0.3566,-0.8023,5.4355,-0.8592,-0.8133,-1.18,-0.4648,-0.4988,-0.964,0.1754,-
0.3712,-1.3506,0.0781,0.2982,-0.2708,-0.2805,-0.7627,-
0.5416,1.0041,0.4636,5.5741,63.5199,2.5029,2.048,2.4925,22.3087,1
8.413,52.8997,148.8705,31.2757,29.1034,47.6825,29.4768,76.9008,92.8494,86.2441,116.627,-
2.4878,0.7634,-0.0709,-0.2928,0.0373,1.7996,1.7669,1.2149,0.0054,-0.337,0.2738,0.2746,-
0.2607,-0.2387,-0.4898,-0.1266,0.0082,-1.0049,0.5319,0.169,0.2212,-
0.2728,4.554,2.388,2.7197,2.597,1.2269,2
18.2333,63.9092,9.9239,0.2379,10.7923,53.1716,35.5918,23.9339,213.0277,140.9691,65.262,-
0.3233,-0.4073,-0.3567,0.2561,-0.248,-0.4287,-0.1419,-0.2565,0.2801,0.4858,-
```

```
0.542,0.201,1.0193,1.2595,2.3203,-0.0249,0.0298,0.2339,0.4185,0.1608,-0.1339,-
0.5795,1.5536,3.1583,2.702,2.1736,1.351,3
17.4031,183.2884,631.7183,147.4865,34.6574,62.254,27.1447,100.1625,57.6107,157.3786,579.8
691,-4.5223,0.3315,1.5221,-0.4232,2.2643,1.5032,0.8642,-0.4451,0.9468,1.3106,0.2055,3.2981,-
0.9902,0.0572,-0.1036,-0.024,-0.0943,-1.338,1.6605,-0.1979,-0.0149,-
0.3057,4.1864,19.3418,3.2149,2.6199,1.1741,2
17.241,1.5704,0.4288,0.1469,10.283,9.0415,40.1799,272.1687,444.2034,52.722,54.8061,0.3659,0
.133,0.1558,0.7443,0.2023,0.179,0.3491,0.192,-0.0464,0.3964,1.4199,-
0.4794,2.7903,4.5151,1.3405,-0.06,0.2899,-0.8125,1.5785,1.3115,2.0296,-
0.3029,2.6693,2.8837,2.3747,53.4232,1.2047,2
19.5605,601.1923,27.7117,0.3965,67.3982,1175.3018,468.0399,2529.0415,832.9075,9.1173,58.1
295,-0.4835,1.5795,1.2571,0.2385,1.1382,2.1103,-0.0051,-0.4847,0.3213,-
0.2823,0.5567,0.6505,0.1775,-0.4567,0.6281,-0.1198,0.4233,1.7139,0.9964,1.872,0.2315,-
0.1097,1.6468,2.3219,2.4127,2.9572,1.5555,3
11.6623,73.1144,284.9245,71.0497,12.6968,21.7034,12.9836,22.9662,25.7195,45.009,203.2619,-
4.3132,-0.3107,-0.3756,-0.4138,-1.0321,-0.794,0.1547,-0.3749,1.0458,1.2979,-0.2313,0.5631,-
0.8299,1.7734,0.4608,0.3859,0.0116,-1.0551,-0.4307,0.1212,1.1298,-
0.2179,4.9058,6.7021,12.2866,7.186,1.1929,3
13.9784,72.4539,409.1709,120.1461,37.6576,42.9175,45.3678,69.2415,118.0484,79.1079,1297.9
908,-4.3085,-0.0479,0.0374,-0.3839,-0.0928,-0.753,-0.4461,-0.4533,2.1145,-0.0982,-0.8631,-
0.0639,-0.8357,-0.3927,-0.3506,-1.1645,-1.0746,-1.196,-1.0615,2.4047,1.0055,-
0.3309,5.0127,2.6356,4.9321,2.9306,1.199,2
```

```
12.5722,113.4217,330.0609,74.6913,60.2163,22.379,120.4248,61.4576,29.9801,150.1917,273.62
1,-2.6463,0.9963,0.0312,-0.423,-0.8771,-0.8642,0.6028,0.1821,-1.0895,-1.2705,-0.717,0.0657,-
1.1647,-0.3864,-0.4507,-0.8031,-0.8925,-0.9122,0.1924,0.1606,0.1649,-
0.3082,4.8522,4.4451,2.977,2.3995,1.1819,2
23.8284,531.9527,627.9088,623.883,23.7706,44.207,19.4933,37.1074,30.3452,959.9136,1480.54
64,8.1068,-0.1952,-1.0372,5.6457,-1.3635,-1.2838,-1.1039,0.2076,-0.7527,-0.2447,-1.2458,-
0.1395,-0.6799,-0.2397,-0.7469,-0.6792,-1.1269,-0.0698,0.1598,-1.3932,-
0.4101,5.6972,73.6254,3.0521,3.7043,2.0242,27.3853,1
17.2636,92.1776,625.6885,189.4514,25.0086,15.3372,26.2715,20.1757,44.1618,134.1022,1428.1
49,-2.3111,0.1127,-0.4086,-0.3722,-0.3804,-0.461,-0.6347,-0.8552,-0.3183,-
0.1714,0.2486,0.1113,-0.0009,-0.3676,-0.3564,-0.8414,-0.7857,-0.0063,0.0972,-0.5711,-0.5034,-
0.2969,4.7919,3.3633,2.8956,1.9574,1.1757,1
19.4204,380.0407,788.1221,165.2567,195.8115,35.479,34.7967,12.5037,120.7563,100.288,631.4
755,1.0732,0.8319,0.0551,-0.3784,-0.3628,-0.1912,-0.8494,0.1037,0.5046,-0.4601,-1.7844,-
0.731,-0.8526,-0.1961,-0.6518,-1.2642,-0.8712,-1.0969,-0.7958,-0.6171,-0.6683,-
0.3279,4.1474,3.3556,2.6287,2.7555,1.1992,1
32.4821,565.6937,795.7965,669.7417,335.7175,197.8094,165.7528,40.7041,35.1436,1000.265,14
27.6423,1.8507,0.9722,-0.2541,5.1108,-0.6731,-0.7953,0.6242,0.4217,-0.5446,-1.2359,-
1.0654,0.2934,-1.314,-0.4904,-0.7162,-1.6389,-0.9707,-1.1298,-0.6626,-0.3091,-
0.0553,5.5837,25.7883,2.2864,2.5258,1.9122,26.3956,1
13.8318,116.0541,398.001,88.1404,17.765,59.0634,28.1448,76.6691,47.7709,144.9582,475.9148,
-3.4781,-0.1109,-0.1003,-0.3982,2.1683,2.8354,0.1318,-0.4322,-0.0587,0.0169,3.6467,0.1919,-
```

```
0.3955,-0.339,-0.4934,-0.4441,-0.3998,0.899,0.439,-0.1463,-0.9653,-
0.3008,4.6495,2.6836,2.5445,2.079,1.1945,2
30.9525,500.9053,1938.9085,461.3486,42.929,52.7187,90.4207,60.8079,38.8053,358.3614,2126.
7474,-4.5481,-0.2079,-0.8592,-0.3819,-0.7845,-0.559,-0.8151,-1.229,-0.4656,-0.6496,-0.2071,-
1.2543,-0.826,-0.1681,-0.419,0.6998,1.1202,-1.7365,-0.4974,0.4167,-0.2869,-
0.2601,5.1541,4.9401,2.7671,3.1425,1.2163,2
10.9562,107.6079,286.3574,83.9088,107.1834,56.0093,24.9067,53.8278,56.5342,121.1515,287.6
891,-1.6978,0.4736,0.5921,-0.3745,-0.2486,-1.3582,2.2056,0.5243,1.1678,-0.3385,0.0626,-
0.3472,-0.9069,-0.2345,-0.3933,-0.949,-0.5752,-0.3873,-0.9617,-0.8071,0.1114,-
0.3107,3.6745,3.2358,4.2299,2.2868,1.2457,2
18.2775,570.4573,9.1792,0.223,7.8699,22.1004,21.9725,180.6963,20.677,11.8484,43.4675,0.440
2,0.6038,0.7405,0.4754,-1.0623,-1.3714,3.0575,2.5817,0.2242,-0.29,-0.3016,1.6571,-
0.2771,4.161,0.0541,0.002,0.0738,-
0.1341,3.3272,3.8935,1.8333,0.0408,4.829,18.0335,13.9755,19.8353,1.3521,2
26.1926,785.5591,457.4288,0.5395,41.0301,349.4001,34.2017,40.7593,380.2509,45.166,1333.21
33,-1.3616,0.178,-0.4212,0.1284,-1.3029,-0.7144,-0.0956,-0.1547,0.421,0.7696,-
1.394,0.5471,0.0985,3.686,1.8798,1.1436,-0.2607,-0.0985,2.4577,2.7406,-0.035,-
0.2307,1.5093,2.8576,3.3982,4.4256,1.489,3
7.7108,54.8983,126.9972,27.2919,50.5828,118.3686,15.1316,12.4012,40.0695,53.4407,70.9008,-
3.0366,1.8926,0.1032,-0.3135,-0.2787,0.2147,0.7927,0.2555,1.9332,0.1884,-0.0448,1.751,-
0.8589,2.021,2.0043,2.2713,3.6733,-0.5229,-0.4803,0.4211,-0.4253,-
0.213,4.6876,4.6827,4.361,7.0506,1.1854,2
```

```
25.7253,498.3881,653.9085,629.1715,54.0426,57.5126,18.2995,20.4343,51.5282,905.4721,1452.
9065,7.9503,-0.3892,0.1139,5.2934,-1.4343,-1.9926,-1.9765,-1.3922,-1.7955,-0.3156,-1.4451,-
0.5472,-0.9806,-0.094,-0.6487,-0.7508,-1.1233,-1.0117,0.1291,-0.4003,-
0.2211,5.56,75.0822,3.4391,2.434,2.8912,23.2642,1
11.2907,67.1057,268.6329,62.884,14.5205,64.4145,44.9371,706.4268,1040.3926,96.4247,941.51
05,-2.7496,3.599,-0.7439,-0.3883,-0.1124,0.6247,-0.7589,0.1127,0.4639,0.5468,-0.7964,0.7875,-
1.6345,1.2389,0.0075,-0.7029,-1.5107,-
1.0963,1.4583,1.1301,0.2848,0.9256,4.9457,2.5566,2.4595,1.4038,1.1931,3
26.1136,485.6473,655.2529,631.3539,42.8278,65.9221,19.2386,30.905,40.5457,896.2374,1444.6
716,7.3102,-0.3223,-0.2603,5.3605,-2.166,-1.5702,-1.6733,-1.8262,-0.0981,-1.1817,-1.3871,-
0.6152,-0.8915,-0.0079,-0.6224,-0.8662,-1.4795,-1.0517,-0.9201,-
0.6619,0.155,5.5606,67.9463,2.7608,2.8124,2.6209,23.5804,1
21.5077,325.5811,986.6674,237.0401,187.1178,527.9638,152.7485,280.0525,443.118,399.4996,1
805.5904,-3.6159,-0.2271,-0.6511,-0.3979,-0.459,0.3674,0.0856,-0.5971,1.3799,-
0.0856,0.0644,0.0704,-0.7662,-0.1617,-0.096,0.3176,-0.7401,-1.2673,-0.325,-0.637,-0.5005,-
0.339,4.7671,2.2229,2.8667,2.3575,1.1645,1
33.1889,380.1032,2259.7414,677.1232,190.5522,196.4217,204.1478,331.6325,50.4027,327.228,9
455.7732,-5.4829,0.308,-0.0354,-0.3717,-0.0763,0.2994,0.0483,0.0277,-0.5253,-
0.7516,0.742,1.5209,-0.1771,-0.5925,-0.8885,-0.6801,-0.4897,1.1526,-0.0546,-0.9861,0.6636,-
0.1883,5.0451,4.1076,3.1747,3.0824,1.2748,3
14.9747,118.1248,467.7727,99.8189,5.6501,52.4711,23.9058,81.3154,59.4798,163.102,842.1905,
-5.3812,-0.3059,0.2621,-0.4334,0.4618,2.3295,2.587,1.1936,-0.0372,0.1394,-0.6269,2.6983,-
```

```
0.7827,-0.265,-0.1856,0.4964,0.1165,-0.3057,1.4555,1.4205,1.5972,-
0.3348,4.9902,2.6886,3.628,2.1952,1.1686,2
27.0364,443.5475,685.7276,634.7503,16.5286,24.7946,31.0728,24.6969,13.5529,801.5367,1419.
3609,8.7107,-0.1777,0.0503,5.4528,-1.2937,-1.1629,-0.2358,-0.7949,-0.9446,-0.2377,-0.6749,-
0.1647,-1.0469,-0.4112,-1.1429,-1.0593,-1.0649,-1.5669,-0.0077,-0.2014,-
0.3967,5.5299,61.7194,4.0939,3.997,2.4281,16.7113,1
18.4509,77.1809,12.8979,0.2461,66.9014,54.3123,24.3942,27.7791,350.6249,101.5214,31.1075,0
.6116,1.5407,1.4742,0.3491,-0.6444,-0.4863,-0.0816,1.6788,1.171,-1.319,-1.3804,-0.7073,-
0.3329,1.7503,1.7634,3.25,-0.2202,-1.0413,-0.7072,0.06,0.1338,-
0.0972,2.0444,2.7946,2.4208,3.5472,1.335,2
11.0188,653.5247,396.1599,19.5713,46.4483,338.3361,84.4127,201.561,90.496,43.4579,569.544
9,-2.1116,-0.0624,0.029,0.1216,-0.1421,-0.1137,0.5089,6.1427,-0.487,-0.3436,-0.0461,-0.0528,-
1.332,-0.3684,-0.1624,-0.632,-0.6324,-0.6702,0.5596,-
0.5466,0.4066,0.6042,2.3074,3.1634,6.7215,1.6518,1.207,2
20.8728,258.5792,918.7007,242.2861,53.5309,147.1258,60.9237,10.5601,28.0276,152.1188,1144
.8179,-3.3671,0.4632,-0.0611,-0.4192,-1.6276,0.2968,0.9283,1.8009,-0.394,-0.4559,-0.2788,-
0.4237,0.0756,0.1135,0.0602,0.2354,0.2986,-1.0654,-0.0993,-0.7948,-0.178,-
0.3125,4.9675,4.8758,1.8697,6.63,1.1864,3
15.9148,171.6524,527.8952,110.7052,322.0994,103.9313,186.9909,22.4192,110.5598,183.6695,6
58.9087,-3.0157,0.397,0.3655,-0.3893,-0.8471,-0.923,0.3926,-0.4549,-1.0637,-0.6297,-2.5481,-
0.3493,-0.986,0.0396,1.23,-0.6787,-0.3261,-0.3307,1.2673,-1.624,-0.0811,-
0.3314,4.8928,1.5752,1.9733,3.5822,1.1749,1
```

```
15.0833,187.3955,479.3558,87.3264,262.2763,65.9931,66.3249,44.4891,133.9349,111.27,720.62
61,-1.809,0.4004,0.0386,-0.3378,-0.9977,-1.1986,-0.9141,1.3208,-0.3913,-1.2797,0.0494,-0.137,-
1.0448,-0.4415,-0.0637,-0.8011,-0.971,-1.1721,0.1441,-1.0911,-0.2179,-
0.322,4.7667,3.0146,1.9989,2.378,1.1988,2
14.3293,154.5944,432.6439,79.1896,126.3582,43.8879,75.4139,32.472,15.4638,133.6194,429.01
24,-4.1446,0.071,-0.2131,-0.3591,-1.0026,1.3299,1.8928,2.7452,-0.5429,0.115,-1.0495,-0.0814,-
0.2864,-1.3681,-0.6506,-0.9706,-1.3859,-0.5485,-0.3652,-1.3726,0.0963,-
0.2372,4.7064,5.0915,2.1766,3.0981,1.2005,2
17.1185,204.9268,614.2681,150.7738,48.5564,53.236,17.4638,213.1265,599.6955,111.9843,1048
.0039,-4.5755,0.5511,0.7556,-0.4245,-0.1827,0.1158,-0.4462,0.9215,1.2012,-1.8039,-0.5917,-
0.1228,-0.6775,3.8873,-0.327,-0.4182,0.0213,1.0493,-0.1361,2.3577,0.3354,-
0.3082,4.7289,5.2564,5.4051,15.8066,1.1816,2
21.4694,188.7577,80.4671,1.91,322.8488,807.5123,87.7462,342.6574,7.5016,14.3514,166.234,-
1.2941,-0.086,-0.0192,0.0188,-0.0225,-0.04,1.3273,2.0399,2.6688,1.5058,-
0.1963,0.975,1.3638,0,2.3068,2.0362,1.3033,-1.0699,-
0.0121,0.4539,0.3675,0.0369,1.684,2.669,3.4278,3.7942,1.2133,3
15.742,285.2229,517.5734,114.645,49.2918,9.7815,22.7163,32.9136,11.4736,177.934,459.4714,0
.4041,-1.0223,-0.8222,-0.4225,-0.3085,0.3073,-0.064,0.2509,-0.3709,-0.4084,0.0041,0.0285,-
0.7934,-0.435,-1.1184,-1.2588,-1.033,-0.7564,-0.4062,-0.2066,-0.9296,-
0.3283,4.657,2.4646,2.9669,2.6642,1.1807,1
17.0815,187.9985,613.6383,150.062,235.0719,473.0085,199.8071,86.2503,53.8964,35.9404,212.
5829,-2.3077,1.2076,0.6675,-0.3926,-1.4998,-1.0883,-0.7367,0.5513,-0.0435,0.5072,-
```

```
1.0536,0.4288,-0.3217,0.4783,0.2723,0.5117,0.5442,1.6051,1.6548,1.1038,-0.1645,-
0.3129,4.4419,2.0626,2.2515,1.9882,1.2083,2
25.5386,1046.6439,369.0112,0.69,9.9844,44.0012,52.6935,41.008,27.6229,130.0036,229.8263,-
1.3076,0.6196,-0.2179,0.1571,-0.7894,-0.4653,-0.2684,1.8379,-0.6087,-0.1178,3.7902,-
0.2323,0.9075,0.2876,-0.2061,-1.0164,-0.0519,1.6669,1.7023,-1.061,0.4415,-
0.1021,1.4992,2.8821,4.9042,2.9685,1.626,2
13.921,102.8306,407.0009,91.0408,17.6792,105.5504,25.8546,553.0874,30.0991,64.6424,416.48
69,-2.1854,-0.1864,0.1226,-0.3661,1.8149,1.6786,0.3055,2.3792,0.4102,0.4969,1.2841,2.841,-
1.5727,0.3761,-0.2056,-0.2068,-0.3613,-0.881,1.4655,0.0423,0.5502,-
0.2776,4.7815,5.3449,7.6462,4.8768,1.1877,2
17.4151,6.8,1.229,0.1665,35.3243,28.6705,37.6086,30.538,19.495,69.9144,128.2388,-
0.0507,0.0495,0.5171,0.6976,-0.4789,-0.4861,-0.1426,-0.5576,-0.6628,-0.3817,-0.8775,-
0.1279,2.4371,0.8833,0.5813,0.1647,-1.0528,0.6248,1.3542,0.7358,-
0.9919,0.1011,2.1173,2.8824,7.7525,6.0769,1.6153,2
20.4173,433.0527,66.0129,0.3417,73.5421,367.7974,132.0146,44.4759,281.3215,79.9498,506.23
41,-0.8186,2,2.6517,0.3109,-0.4274,-0.1964,-0.19,-0.5484,1.1055,0.6394,-0.8041,0.3878,-
0.1703,0.9766,-0.1247,0.4507,0.4434,0.1258,1.3184,-0.6486,1.9176,-
0.1942,1.5155,2.4061,5.8283,19.7201,1.3924,2
22.9469,500.0257,621.1914,621.2959,15.7963,78.8791,18.6203,66.7056,54.3488,20.3422,7.682,7
.9045,-0.4293,-0.4394,-0.4025,-0.8483,-1.09,-0.4583,-0.3209,-0.2436,0.0703,-0.8868,-0.0708,-
0.0757,-0.4322,-0.8397,-1.0269,-1.3376,-0.1038,0.1118,1.1711,0.2942,-
0.6992,78.1395,2.218,2.0274,1.9708,1.7659,1
```

```
30.3438,119.2563,266.8765,73.0957,59.8275,27.1194,24.0343,17.4601,57.5262,970.9207,1436.7
712,-0.0111,0.3049,-0.0538,4.9133,-1.2168,-1.1862,-0.0371,2.0073,1.9717,-0.4281,0.5522,-
0.1612,-1.0674,0.3869,0.1222,-0.2011,0.0404,-0.0906,-0.7217,-0.4303,-
0.2323,5.0853,3.9028,4.1352,2.1988,2.666,26.3236,1
29.5633,1306.0576,630.8648,6.2286,155.8731,59.8241,32.9689,536.5396,269.965,69.0932,765.7
699,-1.4622,0.8326,0.8446,0.0262,-0.278,-1.518,-0.2616,1.6522,0.9384,-0.7627,1.9975,0.7855,-
0.4681,3.9813,1.507,3.1245,0.6835,-0.6175,2.2975,5.3091,0.1506,-
0.0789,1.6684,8.1167,3.2537,10.2183,1.4266,3
10.4997,68.8576,230.7583,53.4776,5.3929,76.3033,32.2575,21.3131,26.6266,92.2462,218.8714,-
3.2763,-0.0266,0.2357,-0.3962,-0.3475,0.2165,0.4923,0.2893,-0.1024,-0.1085,0.3161,-0.11,-
0.0927,0.2726,0.2909,1.3345,0.9183,-0.2644,1.0017,-0.262,0.3756,-
0.3119,4.7224,1.9644,2.0082,2.7198,1.1847,2
27.7349,799.3167,521.4088,2.8221,74.2324,102.5101,141.6875,57.0721,34.8926,159.0522,2277.
7246,-1.4909,-1.1924,-0.7946,0.0749,-0.0069,-0.0332,2.6091,4.5423,1.6183,-1.0089,-
0.0331,1.7279,-1.5181,-0.2865,-0.5048,0.2081,-0.1542,-1.4536,1.7363,0.7022,-0.47,-
0.0877,1.6206,2.5023,6.1951,2.2283,1.3587,3
14.7488,164.221,455.1212,107.0791,31.7538,256.261,28.8117,48.0913,112.4487,79.1153,405.49
4,-2.333,1.9955,0.5432,-0.4215,-2.0499,-0.5587,0.0621,0.4182,1.056,-0.254,-1.4485,0.2622,-
0.8273,1.5881,0.1492,0.1408,0.3328,-
0.0581,1.7375,1.3932,0.4845,4.4321,4.9415,2.5049,4.2563,5.0503,1.2218,2
24.6398,962.57,237.4272,2.2474,30.1203,23.7908,21.959,18.4846,2320.1109,50.2646,70.6862,-
0.0671,0.3239,-0.8959,0.0731,3.4199,4.2593,-
```

```
0.215,0.4508,0.1561,0.0513,1.5876,1.283,0.5538,1.5879,0.7938,5.2006,-
0.7776,1.4381,2.5548,4.1431,-0.25,-0.3072,1.5706,13.2675,14.5112,7.3158,1.2811,3
16.9957,148.8167,604,126.2298,42.8936,28.3488,51.219,44.3006,97.893,111.4393,627.6523,-
4.8129,-0.0404,-0.2802,-0.3961,-0.812,-0.7555,-1.2539,-0.4155,-1.7311,-1.7267,-0.9901,0.3948,-
1.2693,-0.5172,-0.2636,-1.0596,-1.0942,-1.2671,-0.9894,-0.2665,1.4174,-
0.3254,4.7258,2.8884,3.7352,2.0949,1.1761,1
18.6775,145.2202,718.7305,214.9076,16.5008,6.3247,20.334,16.962,30.4196,107.3692,747.7597,
-5.9072,0.0315,0.4117,-0.3846,-0.138,0.3472,-1.1489,-0.9599,-0.2391,-0.619,-0.3993,-
0.7964,0.6066,-0.2742,0.0676,-0.5804,-0.8409,-0.1361,-0.2224,-0.0916,-0.5924,-
0.3263,4.9093,3.4063,2.5641,2.4785,1.1805,2
37.2941,349.9702,978.927,718.4343,35.3611,113.6463,21.4753,49.347,71.4134,717.3552,1432.4
521,6.4923,-0.4795,0.1838,4.3901,-1.726,-2.1067,-1.2448,-1.2367,-0.6242,-0.1663,-1.2456,-
1.164,-0.734,-0.5638,-1.4518,-2.0587,-1.331,-1.6228,-1.5688,-1.5153,-
0.5925,5.0223,38.0826,3.1241,2.9717,1.7998,12.427,1
22.3247,773.6213,172.0619,0.3462,43.5697,73.8473,19.2339,83.5358,131.7272,61.3999,159.691
8,-1.0837,-0.0472,0.0141,0.2768,1.143,1.9337,1.4756,4.0348,1.745,-0.6367,2.3576,0.7558,-
0.6284,-0.1448,-0.3876,-0.4587,-1.5285,0.9701,-1.4534,1.0863,0.3072,-
0.0774,1.3506,12.715,2.6893,2.6558,1.4121,2
14.4034,98.1155,424.7372,117.4126,76.7222,353.8193,48.655,40.9254,86.2461,180.8833,1188.0
673,-2.8575,0.8901,0.0757,-0.3065,1.022,0.7925,-0.0937,-0.02,-0.4869,-0.141,-0.0679,0.512,-
0.8211,-0.1758,-0.0489,-0.0547,0.1015,-0.4553,0.844,-1.3173,-1.144,-
0.3139,4.6172,1.9188,1.9265,2.8565,1.2271,2
```

```
23.362,1125.177,240.985,0.549,253.8656,52.3847,31.7092,85.655,67.489,131.9812,583.0626,-
0.9146,0.9881,0.0624,0.2407,-0.7829,-0.8818,-1.4997,0.834,-0.8125,-
1.1492,0.4316,0.0988,0.5503,0.5345,0.54,-0.5402,1.1096,0.071,2.292,2.3819,1.5657,-
0.5148,1.2566,7.0797,4.4052,3.0141,1.3891,2
9.7432,196.1351,199.9452,47.4684,21.6933,12.9951,9.5146,20.5783,29.7359,102.1672,275.2405,
0.0303,0.4583,0.9015,-0.3762,-0.4855,3.3252,1.9099,1.3336,1.5051,0.4737,1.0384,2.1202,-
1.0417,-0.7214,1.6341,2.6965,-0.171,-0.0581,-1.1011,-0.5723,-0.3361,-
0.1807,4.3385,5.4521,4.3568,10.3991,1.211,3
9.6114,54.0084,195.7029,50.6346,44.286,56.0009,15.7383,29.8022,49.0686,111.9788,785.7439,-
1.6401,-0.124,0.6672,-0.3086,0.3949,0.4303,0.2126,0.2149,5.5848,1.8854,0.1265,0.3325,-
0.2045,0.2484,0.2957,0.0083,-0.3669,-0.5106,-0.8551,0.5037,-0.1652,-
0.2719,3.893,2.2101,4.7465,2.6706,1.215,2
32.4636,571.1416,777.7183,667.2188,86.0331,56.2856,36.2776,28.5542,47.8745,1033.7642,1445
.5592,2.4735,-0.1952,-0.7701,5.1883,0.0171,0.1928,-0.6209,-0.0708,0.0205,0.0177,-0.6769,-
0.2485,-0.5156,-0.8007,-1.0633,-0.3996,-0.3653,-1.4135,-1.0241,-
0.0694,0.2064,5.5781,24.295,4.0618,1.955,3.6807,28.5724,1
18.3327,30.6051,9.8053,0.1973,25.4518,127.331,60.2632,24.7297,1178.3452,35.4079,20.6242,-
0.1435,0.34,-0.0018,0.3226,0.3936,0.2494,0.8766,0.4304,-0.923,0.3702,-0.549,0.1595,-
0.1971,2.2891,1.2854,3.412,2.2671,-0.945,0.6966,2.7547,0.6546,-
0.268,1.6809,4.1975,2.1872,8.6308,1.6993,3
24.0117,278.8676,1214.0838,320.1245,110.3774,143.4636,111.685,80.2221,79.4175,214.3195,62
2.5445,-4.5973,-0.4475,-0.6669,-0.4088,-0.2892,-1.1253,-0.5515,-0.3539,-1.5734,-1.1458,-
```

```
0.9558,0.0161,-2.0343,-0.8067,-1.305,-1.7924,-1.1498,-1.3452,-0.0509,0.0828,-0.2813,-
0.3213,4.9929,2.7468,2.5861,1.8653,1.1782,1
16.4938,122.7065,552.6088,122.6846,32.3194,366.4683,61.4109,21.641,30.6211,139.2855,845.4
798,-4.5009,0.7501,-0.0306,-0.3944,-0.005,1.2972,-0.1986,0.1141,1.2368,-0.2379,-0.3064,0.462,-
0.8215,-0.6387,-0.4359,0.1561,-0.2365,-0.2907,0.4071,0.0928,0.1979,-
0.3012,5.112,7.4903,8.6245,2.4771,1.1841,2
12.3636,222.5799,318.2642,75.8159,52.9155,14.0123,32.3055,17.6076,11.1269,133.2958,343.26
22,0.2411,-0.3993,-0.0205,-0.3988,-0.6569,0.7154,1.6402,1.4969,0.2934,0.0556,-0.7134,-
0.1414,0.4952,0.1876,3.5471,-0.8536,-0.5047,1.2973,-0.0904,2.017,2.1899,-
0.312,3.7658,3.7314,2.9383,2.3291,1.1849,2
25.4571,522.0142,643.4109,626.0072,36.3089,76.4501,29.7565,92.2281,63.5105,891.2955,1446.
1472,6.937,-0.3836,-0.668,5.5177,-0.8031,-1.4997,-1.2493,-0.1679,-0.6938,-
0.6493,0.2792,0.1604,-1.3231,0.2157,0.089,-0.7704,-0.5281,-0.8343,0.9519,0.6507,-
0.2396,5.5901,61.3336,2.3288,2.855,1.9812,23.4719,1
20.8444,799.4487,71.2379,0.3919,194.9379,817.5086,1155.1126,1521.4175,1366.2685,33.5151,1
448.648,-0.1048,0.6811,0.612,0.6599,0.7102,0.6608,-0.3329,-0.6355,0.6855,-
0.1532,1.9457,0.998,0.296,0.4786,0.3742,0.2505,0.2859,2.1086,2.3922,-0.3183,0.4016,-
0.21,1.4404,1.6016,3.1621,1.799,1.3751,3
11.4861,306.2522,277.0316,74.6656,19.3732,82.7835,387.0858,63.2759,145.7026,73.6195,225.2
311,-1.6455,0.4548,1.7462,-0.1986,0.2141,-0.1883,-0.0324,-0.2789,1.2392,0.5137,-0.7832,-
0.5808,-1.3371,1.1768,0.768,1.2423,0.398,0.3812,0.6031,-0.0074,-0.0748,-
0.2818,3.8114,3.0742,2.0366,3.2207,1.1965,3
```

```
18.1525,981.0772,907.7186,70.0258,25.6648,178.0821,39.3032,201.5119,729.6614,23.4397,1228
.2821,-2.6523,-0.0382,-0.0871,-
0.0488,0.1397,0.1125,1.3666,0.2845,0.1844,0.4249,0.4047,0.2697,-0.5958,0.5625,0.2877,-
1.1334,-0.8801,0.0486,-0.0152,0.818,0.4267,0.6279,2.8233,2.0025,2.2011,1.7496,1.1426,2
2.5012,76.2004,19.3276,0.8725,865.4363,741.7478,120.1448,134.0258,110.6506,90.3663,34.121
3,-
0.232,1.4098,0.0262,0.4115,0.2443,2.0387,3.2958,2.8106,1.4265,1.0204,2.0296,2.3961,2.8828,-
0.4508,0.531,-0.0781,-0.691,-0.9704,3.1168,0.5796,-
0.2359,1.0191,2.673,19.5831,15.8222,1.6579,1.4158,2
19.5133,303.3989,28.8414,0.3535,75.4106,1625.9999,202.8225,785.0006,984.4943,15.928,32.03
36,0.342,0.3588,0.821,0.447,0.3778,1.1312,1.9292,-0.0181,1.3983,1.1046,0.8319,1.0008,-
0.1673,0.3136,-0.242,0.4116,-0.0029,0.9026,0.6306,0.641,-0.1056,-
0.1272,1.8748,3.358,1.9443,2.58,1.4214,3
26.0151,390.7684,1410.9318,319.7166,28.7726,16.6627,9.3862,30.7976,8.8643,205.4488,1465.8
637,-4.9381,2.1469,-0.8327,-0.4332,-0.0211,0.4763,-0.4264,0.6096,-0.5412,-
0.4001,0.596,0.0505,-0.3897,-0.3845,-0.099,0.0392,0.4345,-0.8952,1.829,2.5119,0.9909,-
0.3361,4.9219,2.8056,2.636,2.7871,1.1734,3
14.7009,87.109,454.7142,108.402,8.8006,40.2882,18.5582,27.3641,69.1271,134.6796,730.6616,-
5.2675,2.7043,0.0865,-0.4088,-0.5517,-0.1771,0.7124,1.3683,-0.2487,0.0142,-0.5473,0.0474,-
0.1403,-0.1964,-0.2337,-0.2638,-0.0346,-1.6565,-2.0151,2.2236,0.2991,-
0.3424,4.9097,2.2659,5.7226,2.8228,1.2155,2
23.8256,655.3232,225.6346,0.8369,21.07,20.4611,23.8747,29.2036,11.417,99.3728,160.6817,-
1.1656,-0.1402,-0.1323,0.1001,-0.2584,-0.7631,-0.4843,0.1206,-0.6769,-0.2482,-1.3786,-1.0579,-
```

```
1.0687,0.4055,-0.5077,-1.4524,-1.5286,-1.5161,-0.1747,1.2964,-0.7786,-
0.048,1.5027,3.3831,4.594,17.0124,1.366,2
22.1991,567.103,618.525,622.3432,23.7375,120.5686,25.2055,51.2044,30.1274,1046.9252,1490.
6046,7.7785,-0.701,-0.2528,5.7062,-0.5597,-1.1065,-0.6842,-0.5391,-0.4627,-0.3228,-0.7667,-
0.5978,-1.9872,-0.6362,0.0472,-1.3758,-0.6948,-
0.4647,0.3681,1.6129,0.9472,5.7259,73.4829,1.8784,2.509,2.5644,28.4841,1
11.2611,79.4842,265.8484,59.3239,92.4713,309.4145,27.3602,49.5443,167.1381,114.0424,310.6
901,-3.9036,0.1102,0.4709,-0.3821,0.6132,0.609,2.8664,1.6458,0.6746,-0.2954,0.2588,-0.0362,-
0.1935,1.8255,1.5953,-0.0556,-1.0049,-1.1103,1.3142,-0.8214,0.2458,-
0.2299,4.2354,1.7299,2.606,13.1664,1.3657,2
13.3466,207.6865,365.3757,79.5292,177.9987,36.9493,158.1617,25.9024,30.6124,95.3427,454.4
958,0.6529,0.5578,-0.0683,-0.3493,-1.5547,-0.7655,1.0136,1.0108,-0.9064,-
1.7425,0.3651,1.0064,0.954,-0.0676,1.8273,0.9806,-0.1783,0.8822,0.2171,1.2729,2.9771,-
0.2504,3.3527,2.7734,6.0776,11.0308,1.229,2
15.5086,133.3558,504.7125,125.3277,65.6224,173.4938,49.7376,81.9455,27.314,205.2857,455.1
761,-3.8845,-0.2335,-0.3734,-0.4116,-0.6847,-0.7931,-0.8306,-1.0641,-0.2614,-1.7892,-0.5185,-
0.3104,-0.1312,0.2421,-1.1852,-0.1806,-0.2531,-1.738,0.8914,-0.752,-0.2703,-
0.2631,4.8627,1.5574,2.1445,2.7408,1.178,1
20.3315,216.0882,865.9006,215.8681,15.4376,168.3063,34.4977,64.4234,56.5119,169.6252,716.
6278,-3.852,-0.6801,-0.0802,-0.4272,0.0435,-1.2548,-1.8829,-1.6501,-0.1413,-0.5102,-
0.9332,0.0825,-0.902,-0.428,-0.4204,-1.5199,-0.8902,-0.8872,-0.0162,1.9354,0.2292,-
0.2892,4.9664,2.2351,2.47,3.4398,1.1714,1
```

```
7.6475,49.5911,123.9579,30.4462,27.2023,880.5544,51.6444,294.1667,965.8262,769.529,163.33
11,1.9445,2.4661,3.4477,-0.3105,0.2186,1.5709,0.3187,0.685,1.2928,-0.5674,-
0.2118,0.0073,0.2805,0.6257,1.2982,-0.7568,-0.301,0.4032,-0.0529,0.7101,0.2998,-
0.0304,4.4003,3.1442,10.9218,1.5916,1.1967,3
22.7059,545.1582,617.773,622.2394,14.0266,11.6221,12.3678,19.2461,16.1155,1214.5925,1495.
7383,9.1316,-0.2021,0.0824,5.6981,-1.4939,-1.4148,-0.8436,-0.8185,-0.0552,-0.1686,-1.1241,-
1.1216,-0.8959,-0.1176,-1.0943,-1.5921,-1.0884,-1.1248,-0.6815,-0.8632,-
0.9965,5.7157,84.2238,2.4268,2.7005,1.9543,32.7402,1
22.4414,584.9208,160.2664,0.4041,31.6288,67.8816,14.2195,46.4966,10.0841,31.4375,130.5306,
-1.0701,-0.5917,-0.0445,0.2477,1.5506,2.7762,4.1237,1.382,0.5498,0.1061,2.7085,1.1626,-
1.1322,-0.5898,2.1742,1.8001,1.2265,-0.3359,2.3311,3.4216,1.1191,-
0.0038,1.4084,4.9484,2.4733,3.5237,1.3639,3
21.6142,276.2729,976.9045,243.1264,299.2719,119.5479,24.5341,22.9918,21.6727,161.0843,633
.7799,-2.4973,0.632,0.7911,-0.3777,-0.3551,-0.5659,0.0948,0.0186,-
0.0867,2.3004,0.8147,0.1114,1.9313,0.6869,0.2074,0.2615,0.5789,-0.6418,-
1.5488,0.7093,0.8207,-0.3439,4.6203,2.5293,3.2951,2.4934,1.1715,3
27.2025,524.839,1551.2798,337.0982,566.7556,351.6375,195.6126,154.4618,92.941,227.6854,14
09.2365,-3.3157,0.3252,-0.4784,-0.4123,0.5914,-0.8035,-0.6121,-0.0431,-0.4448,-0.3244,-1.001,-
0.4633,-0.2902,-0.9293,-1.0658,-0.3096,-0.9702,-0.8345,-0.3016,-0.903,0.269,-
0.307,4.367,3.7565,2.912,3.4803,1.1767,1
27.8682,1524.5712,723.0129,0.4346,9.2271,10.5471,5.2087,7.4104,6.217,76.8172,2821.155,-
1.2174,-0.2643,-0.3508,0.2187,-0.6047,-0.6677,-0.5193,-0.1495,-0.6741,-0.5243,-0.3272,-
```

```
0.3555,-0.5779,-0.1205,-0.0006,0.0807,0.1879,0.0344,0.3073,0.0207,0.2258,-
0.2933,1.3494,2.6404,3.119,2.9925,1.5675,2
9.5951,165.2581,194.0422,37.671,55.2432,10.4693,51.5971,270.9549,118.9539,74.6596,330.311
3,0.3112,0.8933,-0.6588,-
0.3959,2.4645,2.9092,1.4775,0.1485,1.4306,1.0953,1.6995,2.8599,1.6144,1.2733,-0.5497,-
1.2274,-1.3103,1.8846,2.3638,0.6299,1.0572,-0.24,3.3698,11.8403,4.7114,9.4885,1.1834,3
8.3324,55.0955,147.2538,23.4954,6.081,102.2438,11.8372,237.7907,94.6593,169.1528,163.7442,
-3.0971,0.3843,-0.2685,-0.3775,-0.2705,-0.393,0.2117,0.1832,-0.2362,-0.3426,0.0265,0.3364,-
1.4269,0.8795,0.7956,-0.1021,-0.7451,-0.7654,-1.3014,-0.0489,0.618,-
0.2976,4.8783,2.0303,2.0419,2.3712,1.1715,2
16.0954,216.5748,544.8304,112.2999,145.6661,19.743,21.5758,18.1956,47.1652,194.6737,646.2
273,-2.7537,0.0879,-0.3862,-0.3691,-1.4555,-1.0739,-0.5062,0.0686,-1.0446,-0.8468,-0.9519,-
0.8117,-0.388,-1.0174,-0.6682,-0.7086,-0.4456,-0.6808,-0.5549,-0.627,-0.3504,-
0.3014,4.4127,4.1523,4.1023,2.6496,1.1939,1
19.3852,170.379,792.8926,191.3235,45.5786,124.5487,33.2179,39.0303,25.6836,175.7211,1100.
7076,-5.3436,-0.131,0.58,-0.3928,2.1313,1.3858,2.925,-0.7823,1.8607,-0.3002,-0.3709,0.2783,-
0.7863,-0.073,-0.9363,-0.5632,-1.3712,-1.4713,1.8044,-0.9243,0.429,-
0.3201,5.1141,2.7423,2.4816,2.4646,1.1697,3
19.3543,106.5617,33.1328,0.1555,5.9016,28.8219,18.3428,6.8548,7.1557,7.7397,493.4496,-
0.9578,0.1383,0.163,0.3022,-0.0748,0.1173,0.1673,0.188,0.0053,-0.2215,-0.0185,0.1996,-
0.5737,-0.4167,-0.1861,0.1684,-0.1891,-0.6729,-0.7358,0.2116,1.5789,-
0.3621,1.4151,4.0412,7.5421,2.8519,1.3001,2
```

```
15.8583,147.1865,523.3652,111.7557,25.9181,125.7743,71.9263,211.877,119.5789,192.6182,458
.6628,-5.0987,0.2944,-0.0644,-0.3657,-0.5527,1.0656,1.6272,0.0708,0.077,0.2592,-
0.6305,1.1321,-1.9203,0.2583,-0.7202,-0.6312,-1.3728,-1.0641,1.1865,-1.4286,0.2306,-
0.324,5.0842,3.0946,1.808,2.1569,1.1852,1
18.1667,775.3815,945.2522,68.3326,8.7724,20.8977,8.3892,55.1178,9.5069,27.0416,79.7556,-
3.1478,-0.0097,-0.1774,-0.0848,-0.1445,0.1689,0.0466,0.1183,0.0145,0.393,-0.0933,0.1611,-
1.0505,1.2323,1.6433,0.361,-0.34,-1.0244,-
0.5983,2.6891,0.1725,0.8941,2.918,2.4749,3.198,4.4684,1.2721,2
18.9676,159.2433,752.4414,189.5167,18.4181,33.3751,24.5924,44.3381,127.1211,149.1213,598.
286,-4.6481,-0.5101,-0.1165,-0.4139,-1.6296,-1.6429,-0.9953,-0.4304,-0.4402,-0.5942,-1.5811,-
0.8003,-0.868,-0.2848,-0.4774,-0.4555,-0.5542,-0.7009,-0.4792,-0.2876,-0.1244,-
0.3324,5.0688,3.1815,3.4175,2.5575,1.1697,1
16.166,1375.829,545.8008,118.7193,62.6327,38.0636,60.6765,225.3061,151.8134,80.3226,323.2
863,-1.2664,0.8865,-0.2508,-0.3681,-0.8965,-0.5454,-1.0966,-1.2537,0.1776,0.029,-0.4042,-
1.8004,0.0973,3.9153,1.8719,-0.2695,0.1349,-1.5208,0.6691,1.7242,0.2074,-
0.2862,2.8012,4.7438,2.467,13.4104,1.1895,2
25.4451,1015.3455,466.5292,0.2834,15.5207,21.6857,66.8542,94.5223,30.2797,273.7178,301.49
54,-1.1474,-1.2353,0.1368,0.2486,0.3369,0.6802,0.3885,-1.0123,0.1112,-0.3741,-0.2049,-0.4023,-
0.2073,-0.9185,-0.9823,-0.9243,-1.6727,0.1352,-0.3348,-1.1176,-1.8314,-
0.0964,1.3298,2.977,5.0297,3.6919,1.3581,3
27.3895,513.8112,1493.4253,359.1194,45.6791,25.5696,48.1548,51.4728,182.9706,473.9231,305
1.1414,-1.1553,0.3791,-0.2763,-0.3471,-0.4335,0.4315,0.7813,-0.2018,0.6182,-0.5283,-0.8292,-
```

0.0641,-1.9949,-0.2932,-0.3432,-0.8008,-0.4956,-1.4839,-0.6806,-0.402,0.42,-0.303,4.7165,2.4898,2.902,2.1704,1.2148,2

LIST OF PUBLICATIONS AND PRESENTATIONS

Book Chapter

1. Mangai, J. A., Kumar, V. S., and Ramesh, K, “Web Page Classification using MDAW*KNN,”* in *Encyclopedia of Business Analytics and Optimization*, John Wang, Pennsylvania, IGI Global Publishers, (Accepted for Publication).

International Journals

1. Mangai, J. A., and Kumar, V. S. “A Novel Approach for Web Page Classification using Optimum features”. *Intl. J. of Computer Science and Network Security*, Vol. 11, No. 5, pp. 252 – 257, 2011.

2. Mangai, J. A., Kumar, V. S., and Balamurugan, S. “A Novel Feature Selection Framework for Automatic Web Page Classification”. *Intl J of Automation and Computing*, Springer Verlag, Vol. 9, No. 4,442 – 448, 2012.

3. Mangai, J. A., Kumar, V. S., and Kothari. D. S. “A Novel Approach for Automatic Web Page Classification Using Feature Intervals”. *Intl J of Computer science Issues*, Vol. 9, No. 2, pp. 282-287, 2012.

4. Mangai, J. A., and Kumar, V. S. ‘Towards Improving Automatic Web Page Classification in the discrete domain”. *Intl J of Data Analysis and Information Science*, Vol. 4, No. 2, pp.81-91, 2012.

5. Mangai, J. A., Kumar, V. S., and Balamurugan, S. “A Novel Approach for Effective Web Page Classification” , *Intl J of Data Mining, Modeling and Management*, Inderscience Publishers, Vol.5, No. 3, pp. 233-245, 2013.

INTERNATIONAL CONFERENCES

1. Mangai, J. A., Kumar, V. S., and Kothari, D.S. “A Supervised Discretization Algorithm for Web page Classification”, *Proc. of the 8th Intl. IEEE Conf. on Innovations in Information Technology,* 2012, pp. 226 – 231.

2. Mangai, J. A., Kumar, V. S., and Wagle, S.M. “A Novel Web Page Classification Model using an improved k Nearest Neighbor Algorithm”. *Proc. of the 3rd Int. Conf. on Intelligent Computational Systems (ICICS 2013),* 2013, pp. 49 – 53.

3. Mangai, J. A., Kumar, V. S., and Nayak, J. “A Novel Approach for Classifying Medical Images Using Data Mining Techniques*”, Proc. of the 2nd Int. Conf. on Data Mining and Its Applications (ICDMA’13),*2013, pp. 41 – 45.

4. Mangai, J. A., Kumar, V. S., and Wagle, S. M. “An Improved k Nearest Neighbor Classifier using Interestingness Measures for Medical Image Mining”. *Int. Conf. on Health and Medical Informatics (ICHMI’13)* 2013, pp. 364-368.

BRIEF BIOGRAPHY OF THE CANDIDATE

Ms. J. ALAMELUMANGAI graduated as Bachelors in Computer Science and Engineering from Bharathidasan University, India in 1994. She worked as a Lecturer for nearly 7 years in engineering colleges affiliated to Anna University, India. She then completed her Masters in Computer Science and Engineering from Annamalai University, India in 2005. In 2006, she joined BITS Pilani, Dubai Campus as a Lecturer in the Department of Computer Science. Since 2007 she continues her service in the same campus as a Senior Lecturer in Computer Science. Her research interests include web mining, data mining algorithms and applications. She has published her research results in various international journals and conferences.

BRIEF BIOGRAPHY OF THE SUPERVISOR

Dr. V. SANTHOSH KUMAR graduated as Bachelors in Computer Engineering from Mangalore University, India in 1990. He then completed his Masters in Computer Science from Birla Institute of Technology and Science, Pilani, India in 1997. In 2007, he completed his Ph. D in Computer Science from Indian Institute of Science, Bangalore, India. His professional experiences include both academic and industry. He had worked as Staff Software Engineer in IBM Software Labs, Bangalore, India during 2006 – 2009. Since 2009, he is working as an Assistant Professor in the Computer Science Department of BITS Pilani, Dubai Campus. His research interests include data mining and performance evaluation of computer systems. He has published his research results in various international journals and conferences.