A Feature Group Weighting Method for Classifying High-Dimensional Big Data

Shakila Sarker (ID: 012162007)

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

MSc in Computer Science & Engineering

November 2019

Abstract

Features hold the distinctive characteristics and intrinsic values of data. But it's of no use if the important information and pattern can not be extracted from the data coming from disparate sources and applications. In the area of big data, feature selection is one of the most important pre-processing step in reducing numerous numbers of unessential, irrelevant and noisy features that can seriously affect the outcomes of the classifier models. The main motivation for applying feature selection is to reduce high-dimensionality of large-scale data. As high-dimensional big data has more features for training, it becomes challenging and costly to measure the performances. The aim of the research is to build models with several hybrid feature selection techniques so that the classification algorithms can have only those features that are really relevant and help to achieve better performances. Also, finding the informative features and grouping them so that we can extract the knowledge from Big Data. In this research, we have collected 10 benchmark datasets from UC Irvine Machine Learning Repository. We have applied several feature selection methods and tested their performance (CFS, Chi-Square, Consistency Subset Evaluator, Gain Ratio, Information Gain, OneR, PCA, ReliefF, Symmetrical Uncertainty and Wrapper). The feature grouping methods are named Random Grouping, Correlation based Grouping and Attribute weighting grouping; these groups were experimented with ensemble classifiers: Random Forest, Bagging and Boosting (AdaBoost). With the observed result it has been found that these groups have similar or even better result than the entire feature sets for the datasets. Attribute Weighting grouping method has shown promising performances for the Big Data.

“This dissertation is dedicated to my parents and sister whose unyielding support and encouragement have inspired me to pursue and complete this research.”

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete the thesis for my MSc in Computer Science and Engineering.

I would like to thank my supervisor Dr. Dewan Md. Farid, Associate Professor, Department of Computer Science and Engineering (CSE), United International University (UIU), for his patience, motivation, enthusiasm, and immense knowledge.

I pay my deep sense of gratitude to Dr. Salekul Islam, Professor and Head of the Computer Science and Engineering Department, United International University to provide me the opportunity to complete the thesis.

I express my sincere thanks to Dr. Swakkhar Shatabda, Associate Professor, Department of Computer Science and Engineering, United International University, for his helpful suggestions and for critically reviewing this thesis manuscript.

My deepest gratitude and respect to Dr. Mohammad Nurul Huda, Professor and MSCSE Director, Department of Computer Science and Engineering, United International University for reviewing my thesis. Special thanks to him for his kind and extraordinary support throughout the completion of my thesis.

I also like to acknowledge Md. Saddam Hossain Mukta, Assistant Professor, Department of Computer Science and Engineering, United International University for his valuable time to review the thesis and insightful comments.

Lastly, I am forever grateful to my parents, sister, friends and well wishers for their constant prayer, love, inspiration, moral support and sacrifices during the entire study period, without which this work could not have been possible.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This chapter describes the importance of feature selection process and explain why it is needed to apply feature selection for mining of big data. The motivation and objective of the research are also stated here along with our contribution to the thesis. This chapter ends with the organization of thesis.

1.1 Motivation

With the fast and continuously generated data of all the challenging research areas, huge amount of complex databases are growing. These databses include both structured and unstructured data. Databases forming big data, consist of all possible data storage format such as images, documents, complex data of query, transfer and transactions data [2]. As big data analysis requires integrated, clean, trustworthy, and efficiently accessible data [3], it is a difficult and challenging task of mining these high dimensional Big Data. Most of these big data contains thousand of features which may contain false correlations making mining model more complex. Feature selection has been the most important data pre-processing techniques for decades and accelerated the data mining process without the loss of much information. To deal with the high dimensionality of big data, feature selection methods are significantly considered now-a-days [4].

In present time, extracting, transforming and loading data are using to find useful patterns, predict labels and make assumptions from data. But, these traditional techniques are not enough efficient to analyze the Big Data that has higher velocity and engender high volume. In this age of Internet of Things (IoT) where so many physical objects are connected via internet, massive volumes of data is generated through these devices [5]. This structured and unstructured large amount of data is of no use if the hidden correlation and unseen pattern can not be recognized. Moreover, due to the immense size of big data class imbalance in data becomes high [6]. Though existing machine learning algorithms have huge reputation for increasing performance measurements, but these algorithms face challenges mining big data in terms of scalability or finding patterns and hidden values from data [7]. When these machine learning algorithms are applied with big data, one problem that arises frequently is the high dimensionality of features [4]. These high dimensional feature space force learning algorithms to take account the higher space features. However, this problem can be minimized by applying feature selection methods, which helps to reduce the dimensionality of features and also remove redundant and irrelevant features [8] [9].

1.2 Objectives of the Thesis

The main objective of this thesis is to compare several feature selection methods for mining high-dimensional Big Data. Performances of feature selection method s evaluated by the accuracy of a learning scheme with minimal number of feature subsets. From the above motivation, the following objectives are established in order to reach the aim:

- Design and develop efficient and effective ensemble model with minimum number of subsets of features for classifying high dimensional Big Data.
- Find the informative subspace of feature from high-dimensional big data.
- Apply the subspace with important features from full space of data as groups.
- Compare and test the performance of several feature selection methods.
- Compare the performance of several ensemble learning algorithms with optimum features.

1.3 Thesis Contributions

Thesis contribution are summarized as follows:

1. Reviewed the literature review on various feature selection methods for data analysis and designed ensemble model to classify high-dimensional big data.
2. 15 well known feature selection methods are studied and compared.

3. Used 10 different types of benchmark data sets from UCI machine learning repository which contains multiple classes.

4. Applied feature selection group named Attribute Weighting by measuring feature importance of the features and eliminating lowest important features.

5. To compare feature selection methods, two other types of feature groups named correlation based grouping and random grouping are also experimented.

6. Compared Random Forest, Bagging and Boosting (AdaBoost) algorithms with feature groups.

7. The performances are evaluated by using standard evaluation metrics.

1.4 Organization of the Thesis

The rest of the chapters of the thesis are organized as follows:

Chapter 2 represents the systematic literature review related to the thesis.

Chapter 3 studies the well known feature selection methods and ensemble learning algorithms. And this chapter will also represent methods for feature grouping.

Chapter 4 presents the experimental design and analysis to support the feature group methods. And at the last of this chapter, the analysis results produced through the experiments is delineated.

Chapter 5 summarizes the discussion of thesis analysis; this ends with the conclusions and highlight the future works.

Chapter 2

Related Works

This chapter represents the steps of feature selection processing and the feature selection methods. This chapter also covers the works related to feature selection methods and big data.

2.1 Feature Selection Process

A typical feature selection process consists of four general steps: (1) Subset Generation, (2) Subset Evaluation, (3) Stopping Criterion, and (4) Result Validation. Figure 2.1 shows the four steps of feature selection.

(1) Subset Generation: Subset of features are generated based on some search strategy for subset evaluation. These search strategies are named as sequential search, exponential search and random search [10].

(2) Subset Evaluation: The goodness of the newly generated feature subset is measured by an evaluation function. If this current feature subset is better than the previously selected feature subset after comparing both of them with each other, than this present one is replaced with the previous one [11].

(3) Stopping Criterion: Continuous process of generating new feature subset and comparing them with previous ones, a stop criterion is predetermined for this process. This stop criterion could be a predefined number of iteration or features, or a predefined optimal features or better performance and significant difference [10].

(4) Result Validation: The selected subset of features are verified by using them real world or artificial data sets.

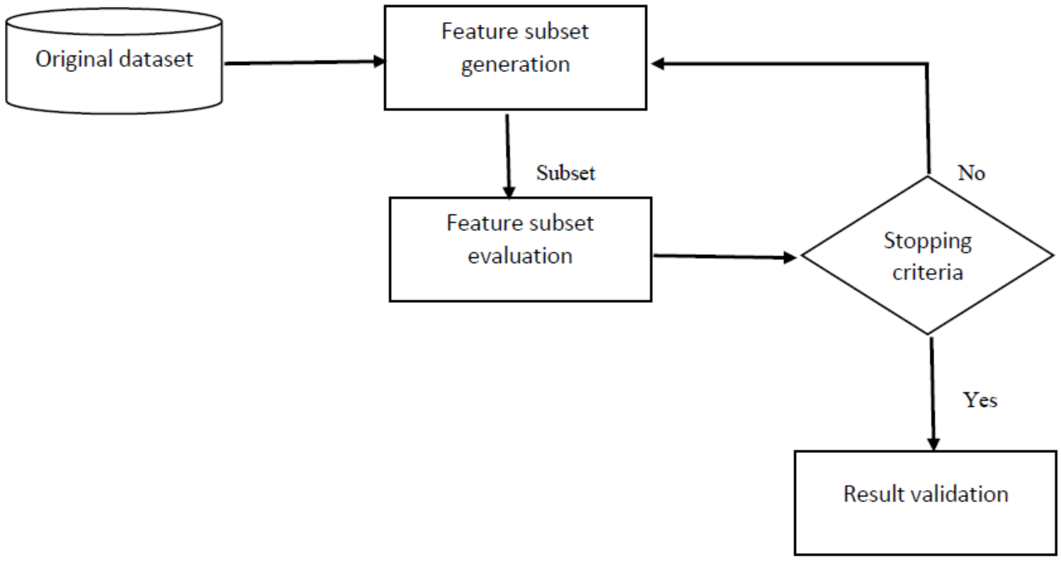


Figure 2.1: Feature selection process.

2.2 Feature Selection Methods

Based on the evaluation criteria of the selected subset of features with the construction of classification there are three feature selection methods: filter methods, wrapper methods and embedded methods [12][13].

2.2.1 Filter Methods

Among all the methods of the feature selection, filter method has received more attention because of it's simple way of approaching high dimensional data [11]. Filter methods are performed for selecting best features for classification models. Using the features that are really important, reduce both the training and evaluation time. Moreover Filter method is not only better for high dimensionality of data but also performs better when it comes to scalability issue [14]. Selection of feature in filter methods is independent from the classifier and used the intrinsic properties of the data. Filters model works based on the association between feature and class label. So for better output feature selection is divided into feature ranking known as weighting and then selecting subset for a better understanding model [15]. Although filter method is an accurate approach when it comes to scalability, it has been also criticised for not having the capacity to deal with the big data [14].

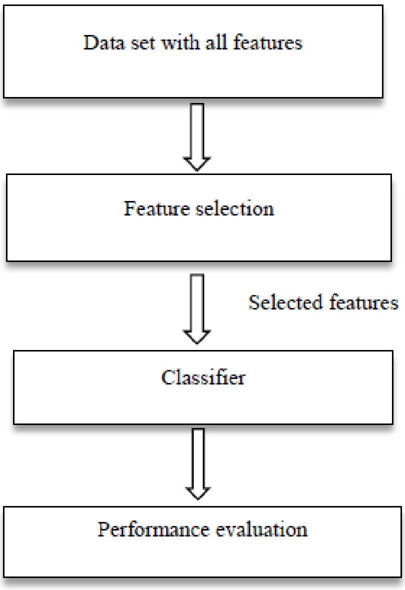


Figure 2.2: Process of filter method.

Chi-Square, Relief based feature selection (ReliefF), Information Gain or Gain Ratio, Mutual information, Feature Selection Perceptron (FS-P), Recursive Feature Elimination for Support Vector machine (SVM-RFE), Correlation based feature selection (CFS), Fast Correlation-Based Filter (FCBF), PCA, Symmetrical Uncertainty (SU) etc. follow the filter approach [14]. In recent years, ensemble learning for classification using feature techniques is obtained to select optimal feature subset. As ensemble algorithm like boosting algorithm has high computational cost, it is important to select feature subset that has minimum numbers of features. For multi label text categorization a rank-and-filter based strategy was introduced that ranks the features and uses only highest-ranked subset. Mutual information had a overall good result but besides it is also said that there is no best feature ranking method in general. It depends mostly on the characteristics of the data sets [16]. Another experiment of Differential Evolution (DE) was introduced inspired by feature ranking and Mutual Information(MI) that showed better accuracy result as small number of feature was used [17]. Also for Intrusion Detection System (IDS) feature selection showed improved accuracy by combining filter technique with Information Gain to select best features and then applying hybrid algorithm [18].

Medical sector is flooded with huge amount of data that contains unwanted or noisy instances. Feature selection method has also improved this sector by selecting most significant features from large data sets. Feature weighting by ReliefF method, ranking the features according to weights and then eliminating features below the specified threshold - combining these three methodology a model was proposed that was experimented on medical data sets. And 50% redundant features were detected from original data [19]. Two cancer microarray gene expression datasets namely Leukemia and breast cancer supervised datasets were experimented with different methods where kNN classifier had better classification accuracy for RF and Fuzzy RoughSet feature selection was said to be a better approach rather than the filter methods for efficiency [20]. The relationship between feature and class and feature and feature were established on medical data that improved classification [21].

2.2.2 Wrapper Methods

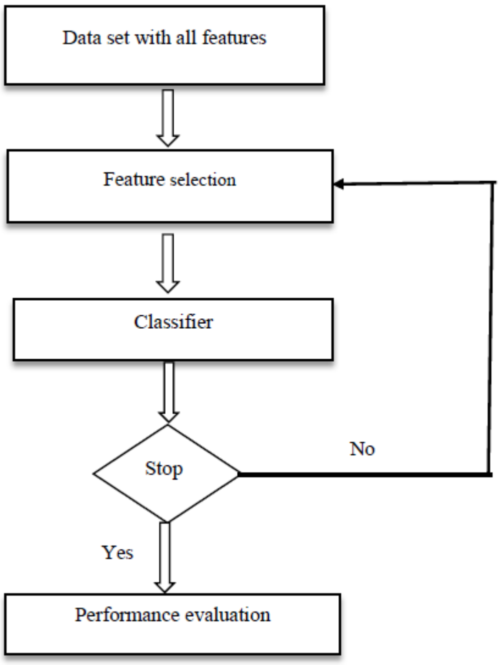


Figure 2.3: Process of wrapper method.

In wrapper method, a subset of features are used to train a model. This method is simple to use and interacts with the classifiers [11]. This method uses something like “Black Box” function which returns the estimation of the quality of model [14]. It uses search algorithm to search through the space of all possible features and evaluate each subset by running a model on the subset. These search method is divided into two classes named deterministic and randomized search algorithm [13]. The disadvantage of using wrapper model is that the computational cost is more for Big Data [14].

Wrapper method is classified into two search algorithms. They are Sequential Selection Algorithm and Heuristic Search Algorithm [22]. In a recent work, selected subset from wrapper based subset selection were given as input to Naïve Bayes classification algorithm and SVM algorithm which returned a good result [23].

2.2.3 Embedded Methods

Embedded method is combined by adopting both the qualities of filter and wrapper methods. Built-in feature selection methods are used in embedded method. The main approach of embedded method is to incorporate the feature selection as a part of training process [22]. And it also interact with the classification model. The search for the optimal subset of features can be described as a search in the mutual space of the hypothesis and the feature subset [13].

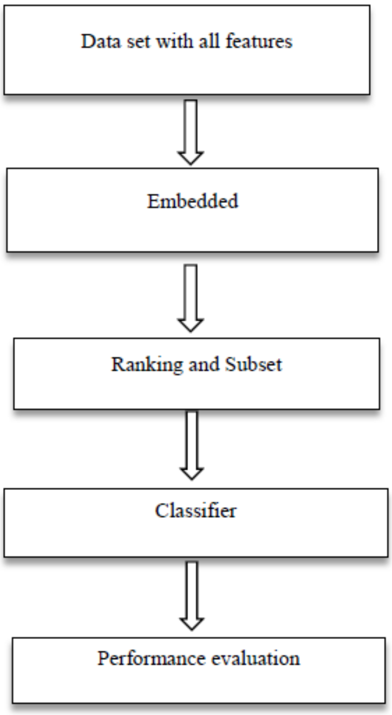
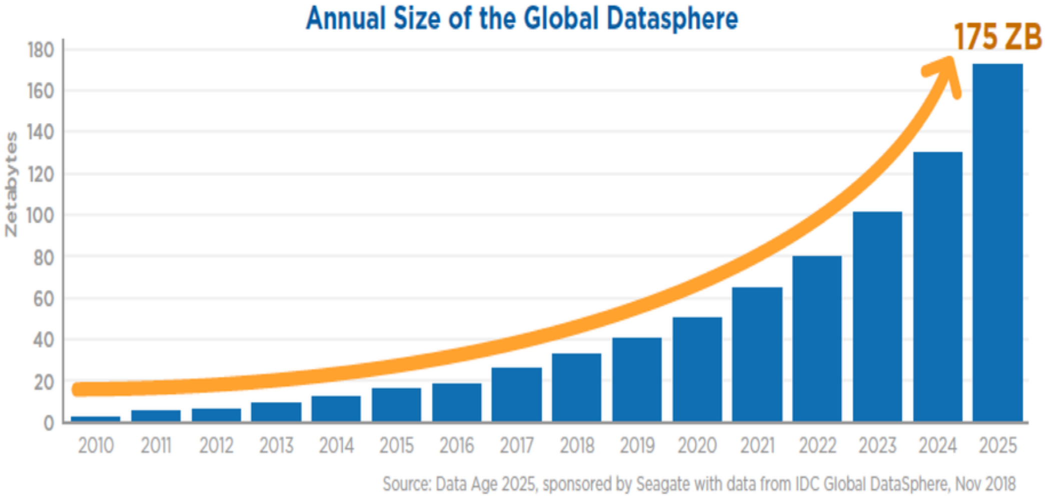


Figure 2.4: Process of embedded method.

Recursive Feature Elimination for Support Vector machine (SVF-RFE) and Feature-Selection Perceptron (FS-P) are the examplpes of embedded feature selection method.

2.3 Big Data

Big data is being generated from diverse sources like digital data, machine data, social data, transnational data and what not. With the exponential growth of internet usage, big data is also growing in an explosive manner [24]. The first thing that comes to mind when we talk about big data is it's size. Big data consists of three Vs generally known as Volume, Velocity and Variety. Gartner in 2012 [25] defined big data in more detailed: "Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making." Big data sizes are reported in multiple terabytes and petabytes. According to IDC (International Data Corporation) report, where they forecasts that the amount of the data will be increased around 40 zettabytes by 2020. Besides, social media like Facebook and Twitter contributes huge number of contents every single minute. Moreover, IDC published a report [1], that shows (Figure 2.5) we will be producing 165 zettabytes per year by 2025.



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

Figure 2.5: Data generation trend [1].

For better analysis and more opportunities, big data needs proper integration. The typical integration systems like ETL(extract, transform, and load) are not sufficient enough to make the most out of these huge informative data. And these rapidly growing data need to be stored and analysed properly to explore the new findings from them.

In this modern age of technology and information, a massive amount of data both structured an unstructured is being generated which is referred as big data. Needless to say, these generated data contain huge numbers of features that are not relevant and hard to process. The challenging part of big data mining is that it contains different formats of data types and high dimensional of features that requires high computational costs [26]. Feature selection is one of the basic preprocessing task of data mining. When we implement feature selection, it assists the algorithm by feeding in only those features that have immediate effect to train the algorithm faster. In most of the cases, algorithms are designed for structured data and performs better when the features are independent in nature. Linked and streaming data also are very difficult to analyse. This is a very challenging task to design such a system that can handle and manager the large scalability that big data generates. To overcome the current challenges of big data mining such as high dimensional small sample size, secure feature selection and big data, many recent development in feature selection research area has been conducted [11]. It has been proved that whenever feature selection is applied it speeds up the computation time and performs accurate results removing unessential and irrelevant features [27]. To reduce dimensionality and decrease the risk of over-fitting of big data, feature selection methods has been used frequently [28]. The application of using different types of feature subset and combinations are becoming popular in today's world. Feature selection techniques were also evaluated in business competitor sector where the amount of data is quite large; as this approach exceeded the manual performance by 10% [29].

Chapter 3

Methodology

This chapter describes the existing feature selection methods and the ensemble learning methods we used in this research. This part also contains the proposed feature grouping methods of the thesis.

3.1 Feature Selection Methods

Feature Selection Method's main focus is to selecting groups of variables that can efficiently perform and provide good results with Big Data having too many irrelevant features [30].

3.1.1 Correlation-Based Feature Selection

Correlation-based feature selection (CFS) is evaluated by a heuristic function which provides subsets of features that are highly correlated with the classes but uncorrelated mutually [31]. Redundant features are removed and highly correlated features are used for better accuracy [14]. Here the main focus is that by using a correlated feature subset the accuracy can be outperformed or equaled. And CFS algorithm needs not to be specified by a threshold number. However, if the features given a class not only are correlated but also have dependencies on each other then this method can fail to select the relevant features.

3.1.2 Chi-Square

Chi-Square method is originally designed to analyze categorical data. It is called a “goodness to fit” statistic as it measures the correlation among the variables. This method is designed according to that principle is that there is no relation among variables and evaluates that they are independent to each other [14]. Chi-Square allows to observe the difference between the actual data and expected data if they do not share any relation [32]. The calculation of Chi-Square is quite straightforward and defined as follows:

$$ChiSquare = \frac{(Observedfrequency - Expectedfrequency)^2}{Expectedfrequency} \tag{3.1}$$

This attribute selection method evaluates subset of features on the entire training data set or a separate hold out testing data set. Here classifiers are used to generate the accuracy of data.

3.1.3 Consistency Subset Evaluator

In Consistency based feature selection, the evaluator is used with the subset of features in conjunction with a random or exhaustive search that finds smallest subset. This smallest subset must have consistency equal to that of the full set of feature. The inconsistency rate is then used to assess its quality [33]. The equation is given by,

$$C_s = 1 - \frac{\sum_{i=0}^{j} |D_i| - |M_i|}{N} \tag{3.2}$$

Where consistency of the subset of feature with N instances is represented by C_s and j is the number of distinct attribute value combination. $D - i$ is the number of occurrences in the i-th attribute value condition and M_i is the majority class with respect to that.

3.1.4 Cost Sensitive Attribute Evaluator and Cost Sensitive Subset Evaluator

These two feature selection methods are highly efficient when it comes to deal with highly imbalanced data of the real world. Both the evaluators consider the costs of missclassifiation. There are very few studies regarding this evaluatiors and it is an issue that needs further studies in broad range [33].

3.1.5 Filtered Attribute and Subset Selection Method

This feature selection method is based exclusively on the training data. The number for feature can always be selected from feature vector and then can be ranked [34]. It eliminates irrelevant attributes but not redundant data, because it only looks upon single attribute at a time. Like the filter attribute selection this method is also based on training data but it evaluates searching method; not ranking method.

3.1.6 Information Gain

Information Gain (IG) is the measurement of information of a feature. It was measured by Claude Shannon who worked on information theory. Recent studies shows that for large datasets with multi-label classification Information Gain works 100 times faster than some other techniques named Binary relevance and Label powerset [35]. The idea is to select a better split for maximal information [36]. This is defined as follows:

$$Info(D) = -\sum_{i=1}^{m} P_i log(P_i) \tag{3.3}$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{3.4}$$

$$Gain(A) = Info(D) - Info_A(D) \tag{3.5}$$

Gain(A) represents the required reduction of the information of the feature A, as we know the value of the feature. A setback of Information Gain method is that even though some features are less informative but they contain more values, the method becomes biased towards those features. To overcome this biasness, a normalization approach named Gain Ratio was introduced later.

3.1.7 Gain Ratio

Gain Ratio (GR) is introduced in decision tree (C4.5) algorithms [37]. Gain ratio is basically the modification of the information gain that reduces the bias on high valued attributes [38]. The merit of the attribute is calculated by measuring the gain ratio with respect to the class.

$$SplitInfo_A(D) = -\sum_{j=1}^{v}(|D_j|)/|D|log_2(|D_j|)/|D| \tag{3.6}$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{3.7}$$

3.1.8 Mutual Information

Mutual Information (MI) is the measurement of shared information between two random variables. If two random variables shares zero mutual information then these variables are considered to be independent of each other [22]. Mutual Information is defined as follows:

$$I(X;Y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p(x,y) log \frac{p(x,y)}{p(x)P(y)} \tag{3.8}$$

3.1.9 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is used for analysing text document and finding the meaning underlying those documents [39]. Latent semantic analysis model represents meaning that derives high-dimensional numerical vectors representing word meanings from the words' distributions in large corpora of natural texts [40]. In LSA, documents are represented ass 'Bags of words', when the order is not important but how many times appears, Concepts are represented as patterns. Words are assumed to have one meaning to make problem tactable.

3.1.10 OneR

OneR rule is simple association based algorithm that generates set of rules and from the set of rules every one rule works with each feature in the condition part. And then it selects the rule that has best error ratio. This algorithm can handle missing values too. A credit card fraud detection approach has been studied using OneR algorithm along with other significant data mining feature selection algorithms which shows this algorithm provides better performance [41].

3.1.11 Principal Component Analysis

Principal Component Analysis (PCA) method is used for dynamically reduction of features. This is an unsupervised method for identifying the important directions in data set. We can rotate the data into coordinate system that is given by those direction and then linear combination of the initial features can be generated. Recent research shows that using PCA feature selection approach can significantly improve classification accuracy of Alzheimer's disease [42].

3.1.12 Symmetrical Uncertainty

This method normalizes the bias of information gain by providing the value of feature to the range [0,1]. Symmetrical Uncertainty method rank features based on how relevant a feature is to a class label. The main difference of Symmetrical Uncertainty feature selection with others is that it calculates feature selection function based on loss [43]. It is given by

$$SU = 2 \times \frac{Gain(A)}{Info(D) + SplitInfo(A)} \tag{3.9}$$

If the equation has value zero, then the two features shares no dependencies and if 1 then by knowing one feature we can use the knowledge to predict another.

3.1.13 ReliefF

ReliefF method is used for multi-class problems. ReliefF method was introduced by Kononenko (1994) [44] who used Manhattan distance for finding near-hit and near-miss instances [19]. Recently, ReliefF method has gained a lot of attraction as it may be applied in many situations including capturing local dependencies and interaction among features.

3.1.14 Fast Correlation-Based Filter

Fast Correlation-Based Filter (FCBF) works by selecting a set of features of highly correlated with class according to Symmetrical Uncertainty(SU) from high dimensional data. FCBF is defined as the ratio between the Information Gain(IG) and Entropy of two features. Then three heuristics are applied to remove redundant features [14].

3.1.15 Recursive Feature Elimination for Support Vector machine

In Recursive Feature Elimination for Support Vector machine (SVF-RFE) method, Feature Selection is done in backward by iteratively training a SVM classifier and removing each time the least important feature according to the SVM weights. L2 norm is used in the SVM minimization problem [22]. L2 regularization have inbuilt penalization to reduce overfitting.

3.2 Feature Grouping Methods

For this research, a feature group weighting method has been implemented. Feature weighting technique is used to identify the lowest important features. To reduce lowest important features for high-dimenisional Big Data, several feature groups have been formed with only important feature seubsets. In similar way, correlation based grouping is formed using corrleation matrix. And, random feature grouping from feature subsets also has been implemented.

For the implementation of the proposed method, a collection of dataset D have been used. These datasets contains more that 15 features. At first, the dataset D was grouped into three forms: random grouping, correlation based weighting(using correlation matrix) and attribute weighting (ranking by feature importance and then filtered out the lowest important features). After grouping the features, dataset D was divided into sub-datasets $\{D_1, D_2, \cdots, D_n\}$ with subset of features $\{x_1, x_2, \cdots, x_N\}$. Then for each and every feature group, an ensemble model is M_i is learned. Finally, a combined voting is being generated from the models so that newly added instances can be predicted.

Algorithm 1 Feature Group Weighting Algorithm

```
Input: Training data, $D$ and a learning model.
Output: Ensemble learning model, $M^*$
Method:
1: group feature from $D$ into several sub-datasets $\{D_1, D_2, \cdots, D_n\}$;
2: for each $D_i = \{A_1, A_2, \cdots, A_N\}$ do
3:     use $D_i$ , and learning scheme to derive a model, $M_i$;
4: end for
To use $M^*$ to classify a new instance, $x_{New}$:
Each $M_i \in M^*$ classify $x_{New}$ and return the majority vote;
```

3.2.1 Attribute Weighting Grouping

Attribute weighting method is basically a data pre-processing task where important features are assigned higher weights and less important features are assigned lower weights. Then the highly weighted features or attributes are used to design the model that reduces high dimensionality of dataset. For this study, attribute weighting method is implemented and then the lowest important features are eliminated. Based on the feature numbers of the dataset that are experimented, lowest 10% features are removed and then rest features are assigned to the first group. If less than 10-15% of the lowest feature are selected then the result has no affect by the change. So a standard thresh (10%) of feature number has been chosen after experimenting with the datasets. Similarly, next groups are formed in the same process until the number of features in a group is not less than 70% of the total features. If groups are formed with less than 70% of the total feature set for our used datasets, performance of the models degrades.

3.2.2 Correlation Based Grouping

In a dataset, many features depend on each another or a cause of another feature. Sometimes, the features are also associated with each another. A correlation matrix is used to show the correlation coefficients among the features. In this study, correlation matrix is used for calculating correlation between the features and highly correlated features are gathered in groups ass feature subset.

3.2.3 Random Feature Grouping

For random grouping, features are selected randomly from the datasets and then group is formed. The number of the features of a group contain same number of features as for attribute weighting for the respective dataset. Sometimes, same features can appear in two or even all the groups as features have been selected randomly.

3.3 Ensemble Learning Algorithms

Ensemble is the way of combining individual learning models to improve the performance of a model. After constructing powerful composite model it classifies a new instance by taking votes from the individual classifiers. When the training data is imbalanced and not sufficient enough then three fundamental reasons (Statistical, Computational and Representational) arise in the way of building a strong hypothesis. Ensemble learning is used to solve these weaknesses of existing individual classifications algorithms[10].

3.3.1 Random Forest

Supervised learning algorithm Random Forest is an ensemble classifier that builds multiple decision trees and ensemble them together to get better prediction. In Random Forest, random subset of the features is operated and the class with the most votes is returned. The advantage of random forest is that random threshold for each feature can be additionally used. Random Forests are trained via bagging or Bootstrap Aggregating methods. In decision tree, an instance goes from root node to the bottom until it is classified in a leaf node, but in Random Forest each instance visits all the different trees of random samples. For classification, most frequent class predicted by individual trees are used and for regression average prediction of each tree is used. To avoid overfitting cross validation should be applied for the models.

3.3.2 Bagging

Bagging stands for Bootstrap Aggregation. It is used to reduce the variance of a decision tree. Leo Breiman (1994) proposed bagging algorithm as the experiment result shows that

Algorithm 2 Random Forest Algorithm

Input: Training data, D, number of iterations, k, and Decision tree (C4.5) induction algorithm.
Output: Ensemble model, M^*
Method:

```
1: create sample $D_i$, by sampling $D$ with replacement;
2: for i = 1 to k  do
3:     derive a tree $DT_i$ from $D_i$ employing C4.5 algorithm by randomly selected features;
4:     compute the error rate of $DT_i$,error( $DT_i$);
5:     if $error(DT_i) \geq 0.5$ then
6:         go back to step 3 and try again;
7:     end if
8: end for
```

To use M^* to classify a new instance, x_{New}:
Each $M_i \in M^*$ classify x_{New} and return the majority vote;

theses bootstrap sampled classifiers result in an optimal classifiers [2]. The main theme is to create several samples from a training set, where the total sample sizes are as same as the original dataset. A sample may have few duplicate instances as they are chosen randomly with replacement. And then some instances may not be appeared for once in the samples. Suppose, a dataset D contains total number of N instances. For iteration i = 1,2,...,k; a training sample or subset Di with n instances is formed with replacement. The learning process generates a classifier model, Mi from each sample Di and then final classifier M* is formed by merging all the k classifiers. The final classifier M* counts the votes of each classifier Mi while predicting classes for sampled dataset. Then M* assigns the class who has most votes to an instance X to classify.

Algorithm 3 Bagging Algorithm

Input: Training data, D, number of iterations, k, and a learning scheme.
Output: Ensemble model, M^*
Method:

```
for i = 1 to k do
    create bootstrap sample $D_i$, by sampling $D$ with replacement;
    use $D_i$, and learning scheme to derive a model, $M_i$;
end for
```

To use M^* to classify a new instance, x_{New}:
Each $M_i \in M^*$ classify x_{New} and return the majority vote;

3.3.3 Boosting

The main functionality of popular Boosting algorithm is to converting weak learners to strong learners. In boosting, converting weak learners to strong learner is done by using weighted average. An equal weight is assigned to each instance. A number of classifiers is learned and the weight of incorrectly classified instance is increased after a classifier is learned. The algorithm pay higher attention to instances having prediction error. Finally it combines the votes from each classifier and builds a strong learner which improves the accuracy of the model. AdaBoost (Adaptive Boosting) is an ensemble Boosting algorithm. For a dataset D, with m labeled training examples, (X1, y1), (X2, y2),..., (Xm, ym) where yi is the class label of instance Xi. AdaBoost assigns an equal weight of 1/m to each training instance so that the probability of each instance of appearing in the training set increases. For k number of classifiers, k rounds of algorithm is required classifiers are trained one at a time. After the first i round, the first classifier model Mi is trained with weight 1/m and then the output weight of this classifier is computed. When an instance is not correctly classified, this instance will be given higher attention in the nest round. Correctly classified instances weight will be decreased and incorrectly classified instances weight will be increased.

Then we calculate the error rate of the first classifier. Error rate is the number of misclassified instances of the training sample divided by the training sample size.A function is used for misclassified instances. If instance is misclassified then it returns 1 and if correctly classified it returns 0. Now we compute another weight for classifiers using weighted error from first classifier. If the error rate is less than 0.5 then we update the correctly classified instances and normalize them. Then each classifier is assigned an weight and which class label has the highest sum of weights of each classifier is returned as the prediction result for a new instance.

Algorithm 4 AdaBoost Algorithm

Input: Training data, D, number of iterations, k, and a learning scheme.

Output: Ensemble model, M^*

Method:

```
1:  initialise weight, $w_j \in D$ to $\frac{1}{m}$;
2:  for i = 1 to k  do
3:    sample $D$ with replacement according to instance weight to obtain $D_i$;
4:    use $D_i$, and learning scheme to derive a model, $M_i$;
5:    compute $error(M_i)$;
6:    if $error(M_i) \geq 0.5$ then
7:      go back to step 3 and try again;
8:    end if
9:    for each correctly classified $w_j \in D_i$ do
10:     multiply weight of $w_j$ by $(\frac{error(M_i)}{1-error(M_i)})$;
11:   end for
12:   normalise weight of instances;
13: end for
```

To use M^* to classify a new instance, x_{New}:

```
1: initialise weight of each class to zero;
2: for i = 1 to n do
3:   $w_i = log\frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
4:   $c = M_i(x_{New})$; // class prediction by $M_i$
5:   add $w_i$ to weight for class $c$;
6: end for
7: return class with largest weight;
```

Chapter 4

Experimental Analysis

This chapter of the thesis represents the experimental procedure and methodology that are embraced for this research. Several major data mining processes including data collection, extraction and pattern analysis are considered to find out useful information. During this stages, various modeling techniques are adopted to arrive at a decision. All the results after experiment and analysis are summarized in tabular form. At first the dataset description are given and then the results for algorithms using feature selection methods are presented.

4.1 Datasets

- **Data Collection**

  As a part of data collection, at first ten standard machine learning datasets are chosen from UCI Machine Learning Repository [45]. These datasets range in from approximately 600 instances to 581000 and having features starting from 14 t0 more than 60. All these datasets are labeled both supervised and semi-supervised containing large amount of unlabeled and a small about of labeled data. We splited the data into 70/30 that means 70% data are training data and rest 30% data are for testing data. The description of the datasets are summarized in Table 4.1.

- **Dataset Description**

  The descriptions of the datasets are given below:

  1. **Adult dataset:** For our first data set Adult, we have 14 number of features which are categorical and integer, has 48842 number of instances with 2 classes.

Table 4.1: Datasets description

No.	Datasets	No. of Features	Types of Features	Instances	Classes
1	Adult	14	Categorical, Integer	48842	2
2	Bach Choral Harmony	17	Text	5665	2
3	Drug Consumption	31	Real	1885	7
4	Dermatology	33	Categorical, Integer	366	6
5	Ionosphere	34	Integer, Real	351	2
6	Soybean	35	Categorical	682	19
7	Census Income	40	Categorical, Integer	299285	10
8	Covertype	54	Categorical, Integer	581012	7
9	Diabetes	55	Integer	10000	3
10	Spambase	57	Integer, Real	4601	2

2. **Bach choral harmony dataset:** Bach choral harmony data set has 17 number of attributes and their type is text, instances number is 5665 and classes are 2.

3. **Drug consumption dataset:** Drug consumption data set has 31 number of instances where the feature type is real, instance number are 1885 and number of classes are 7.

4. **Dermatology dataset:** Dermatology data set has 33 number of attributes of categorical and integer types. Instances are 366 and classes are 6.

5. **Ionosphere dataset:** Ionosphere data set has 34 number of attributes of integer and real type. Instances are 351 and classes are 2.

6. **Soybean dataset:** Soybean data set has 35 number of attributes of categorical type. Instances are 682 and classes are 19.

7. **Census income dataset:** Census income data set has 40 numbers of features and the type of features are Categorical and integer. Number of instances are 299285 and classes are 10.

8. **Covertype dataset:** Next data set Covertype has 54 number of attributes which are all categorical and integer, number of instances re 581012, number of classes are 7.

9. **Diabetes dataset:** Diabetes data set has 55 number of attributes and all are integers, 10000 instances with 3 classes.

10. **Spambase dataset:** The Spambase data set has 57 numbers of attributes with Integer and real type. Instances are 4601 and classes are 2.

- **Data Pre-Processing**

  All the datasets that have been collected for analysis are raw data format. To make raw data to an understandable format pre-processing techniques have been used. Data are transformed into various forms so that all data are in same acceptable form. For categorical type of data, features have been changed into numeric numbers. Missing values have been converted to numeric values too.

- **Missing Value and Label Encoder**

  Dataset contains many instances that have missing or null value and even sometimes value with '?' sign. To handle this type of inconsistency, encoder technique have been used to convert this erroneous values into numeric values. Again, the dataset contains some text values that need to be converted into numeric numbers for analysis. Python library has been used for pre-processing those data named 'Label Encoder' that transforms the texts into numeric numbers.

- **Data Analysis**

  For this research, Python language was used as it contains some packages for scientific use and data analysis such as Numpy, Pandas, Scikit-learn and more. All the methods were implemented in Python 3.6.3 programming language. Random Forest, AdaBoost and Bagging; these three algorithms had been implemented in Python in order to visualize the results of choosing different feature selection methods. Table 4.3 shows the result of the Random Forest, AdaBoost and Bagging algorithms respectively before applying the approached feature selection methods. For better understanding some well known feature selection methods were implemented and summarized in table 4.2.

- **Correlation Matrix**

  For the proposed method, correlation matrix calculation in python using 'pandas' and 'numpy' library had been used. The features were gathered in same group that were highly correlated with each other.

- **Attribute Weighting in Python**

  For ranking and filter, we used attribute weighting in python that is called feature importance. Methods with ensemble of decision trees like extra tree or random forest can compute the relative importance of each feature. We imported 'sklearn.ensemble' from 'ExtraTreesClassifier' library to measure the importance of each attribute and then formed feature subset to create groups. Decision trees make splits that improves the performance measure or we can say maximise the decrease in impurity. Feature importance is then calculated by averaging the number of observations the node is responsible for. The higher the value the more important the feature is.

4.2 Experimental Setup

Accuracy, Precision, F1-Score, Recall are used for calculating and evaluating features performances. And for any kind of performance measurement confusion matrix is required. True positive, True negative and False positive, False negatives are the four parameters used for performance measurement in a confusion matrix. Suppose there are two classes named ‘Yes’ and ‘No’ in a dataset ‘D’ and total number of instances are ‘N’. True Positives (TP) are correctly predicted positive values. That means that the actual and the predicted class both are same and class is ‘Yes’. True Negative(TN) means both the actual and predicted class is ‘No’. Sometimes the actual class value is ‘No’ but predicted class value is ‘Yes’. This term is known as False Positive(FP). And finally when reverse situation happens that means the actual class is ‘No’ and predicted class is ‘Yes’, it is named as False Negative(FN). Let us say, TP + FN = W; TP + TN = X; TP + FP = Y and TN + FN = Z. Based on these measurement parameters, feature weighting or ranking methods are evaluated.

Accuracy: Accuracy is measured by observing the ratio of correctly predicted samples to the overall samples of a dataset. It is often considered that if accuracy of a model is high then the model is best. But this measurement will be best when the value of False Positive(FP) and False Negative (FN) is almost same. Accuracy is defined by,

$$Accuracy = |\frac{TP}{W} - \frac{FP}{X}| \tag{4.1}$$

Precision: Precision is measured by the ratio of correctly predicted positive samples to the total number of predicted positive samples. Precision is defined as follows:

$$Precision = |\frac{TP}{Y}| \tag{4.2}$$

F1-Score: F1-Score formula is measured using both the scores of Precision and Recall . F1-score is very useful to use when the dataset has uneven class distribution. It works best if false positive and false negative value of a sample are almost same. F1-Score is defined as follows:

$$F1 - Score = |\frac{2TP}{W + Y}| \qquad (4.3)$$

Recall: Recall also known as Sensitivity is measured by the ratio of all predicted positive samples to all the observation of a specific class. It is defined as

$$Recall = |\frac{TP}{W}| \qquad (4.4)$$

4.3 Experimental Design

A number of experiment had been run to implement the proposed study. Attribute weighting ranking method was implemented for ranking and filter the features in order to eliminate the features that had lowest importance and then divided them into features groups. The feature number that needs to be eliminated were selected for the experiment depending on the features numbers of used datasets. For attribute weighting grouping maximum 7 and minimum 3 numbers of features were eliminated (based on standard thresh hold of 10%). The eliminated feature numbers were selected following the above mentioned criteria because after elimination of more than eight or ten features, the performance degraded drastically for the formed groups. So based on the feature size of datasets, 10% of the features that possessed lowest ranks were eliminated for every group. Again the random grouping elimination of features was also done randomly but the number of eliminated features was same as the number of the eliminated features of attribute weighting for the respective dataset. For correlation based grouping the groups were formed based on the correlated groups. For some datasets the correlation between the features were so close that led to had only few groups or even all features were grouped in a single group. Table 4.7, table 4.8 and table 4.9 show the individual accuracy results for each feature group. These tables are arranged in the last section of this chapter.

4.4 Results

The classification results for all the databases are summarized in table 4.3. In this table, the actual performance measurements of all the datasets using all features are shown. Table 4.4, table 4.6 and table 4.5 show the comparison of Random grouping, Correlation based grouping and Attribute weighting using three learning algorithms along with three respective algorithms.

According to the comparison from the table 4.4, table 4.6 and table 4.5, **Adult** data set had highest accuracy score about 85.63% accuracy for Random Forest algorithm. For Random Forest algorithm, the precision, recall and f1-score measurements had also higher scores than for the other two algorithms AdaBoost and Bagging. After performing the proposed feature selection methods, Random grouping, Correlation based grouping and Attribute weighting grouping, it was seen from the tables that for Attribute weighting grouping Random Forest algorithm had the most closest accuracy score (85.57%) and also precision, recall and f1-score have closer values. The difference of f1-score measurements is only 0.003 less after performing Attribute weighting grouping method. On the other hand, AdaBoost had a good result by performing Random grouping feature selection method for Adult data set. The accuracy score was almost similar (only 1.02 greater before performing feature selection) after selecting features randomly. Like Random Forest, Bagging algorithm had better results after performing Attribute weighting grouping method. Attribute weighting grouping method had 83.57% accuracy score while the score was 84.57% for Bagging before performing the method(difference is 1). So for Adult data set Attribute weighting grouping method wins as a method and correlation based grouping degrades the performance of algorithms comparing to others.

For **Bach choral harmony** data set Bagging algorithm has almost 99.04% accuracy and other measurements precision, recall and f1-score had great results as well. AdaBoost had also similar valued results. However, Random forest showed less result score compared to them. Three feature selection methods and their results for Random forest, AdaBoost and Bagging could be significantly distinguish from the other data sets for choral harmony data set. For all the algorithms random grouping had outperformed in all the evaluation metrics. On the contrary, the results of correlation attribute weighting and correlation based grouping methods had been decreased eventually.

For **Drug consumption** data set, accuracy for Random forest was 76.05% and recall measurement was 76.00%, higher that all other algorithm's result. Second, Bagging had accuracy of 74.15% and it's f1-score had highest value amongst all(68.00%). Performance of Random forest algorithm had been significantly improved by implementing Attribute weighting grouping method. Random grouping method had also improved accuracy result (76.26%) for AdaBoost algorithm. Correlation based grouping worked better having accuracy of 70.58% and recall of 0.706. These three feature selection methods had a great impact consequently for Bagging algorithm. Almost all the measurements result had been increased after applying these methods.

For **Dermatology** data set, Random forest algorithm has highest accuracy about 96.97% with random grouping. For attribute weighting, Random forest algorithm showed similar accuracy of about 96.96%. After applying AdaBoost algorithm with attribute weighting, accuracy is also quiet good(93.11%). Bagging Algorithm showed a very good accuracy with attribute weighting(96.41%).

For our last data set **Ionosphere**, Random forest algorithm with attribute weighting has highest accuracy about 91.09%. AdaBoost algorithm along with attribute weighting also has a very good accuracy of about 91.13%. And also Bagging algorithm has the high accuracy score of 85.91% with attribute weighing.

For **Soybean** data set, AdaBoost had highest accuracy result (94.25%) for this data set and Bagging algorithm also had similar results. Soybean data set had mixed results. When random grouping method applied for Random forest algorithm it showed the same accuracy result as the result before applying it. And recall measurement also had same results. Attribute weighting method had same value for Precision, recall and f1-score. AdaBoost algorithm's performance has increased by applying Attribute weighting method. Bagging algorithm had better precision and recall values for Attribute weighting method and f1-score has same value.

Census income data set had highest accuracy result for Random forest algorithm, the precision, recall and f1-score is also better for Random forest than other algorithms. And bagging had also better result for this data set. Attribute weighting method outperformed for Random forest algorithm increasing the accuracy and recall by 95.57% and 0.957 respectfully. Besides, precision and recall values were as same as before implementing Attribute weighting. AdaBoost algorithm outperformed for Attribute weighting method and bagging algorithm had the same result value for random grouping. Overall Attribute weighting method wins for census income data set.

Our next data set **Covertype**, Bagging algorithm had overall good result before no feature selection methods had been applied. Accuracy for Bagging algorithm was 95.72%, then Random forest algorithm had also closer results for measurements, having precision result 95.1%. When the feature selection methods were applied for Random forest algorithm, the tables show that Attribute weighting grouping method had better results having 95.15% accuracy. Moreover, Correlation based grouping had same to same result for accuracy, precision, recall, and f1-score before and after applying correlation based method. Closely related feature groups had the same results for RF. For AdaBoost algorithm, Correlation based grouping had the same accuracy, precision, recall and f1-score result after applying the method and attribute weighting had also same results.

For the **Diabetes** data set Random forest had the higher performance compared to AdaBoost and Bagging algorithms. Recall measurement had highest score for Random forest algorithm(60.30%). When feature selection methods were used for diabetes data set, Attribute weighting grouping method increases the performance of Random forest algorithm. All the measurements accuracy, recall, precision and f1-score has increased values after performing Attribute weighting grouping method.Random grouping algorithm had also close results after performing the method. Implementing three feature selection methods showed a mixed type of results for AdaBoost algorithm. For correlation based feature selection had better accuracy result (52.04%);almost 3.32% better than while not performing feature selection and also recall measurement had same type of better result. Then for Attribute weighting method precision and f1-score had better result after applying the method. Comparing the tables it was seen that Attribute weighting method improved the precision measurement(0.537) on diabetes data set and accuracy, recall and f1-score has closest measurement after applying Attribute weighting grouping method. The significant better results for feature selection method attribute weighting was clearly seen for diabetes data set. Random grouping has comparatively closer results but for AdaBoost algorithm, correlation based grouping surely wins at performance as well as Attribute weighting grouping method.

For **Spambase** data set, AdaBoost and Bagging had also closest results. Only Random grouping method showed some constant results close to the results before implementing the feature selection methods. For attribute weighting Random forest had close value for accuracy, precision, recall and f1-score. But all others have decreased values eventually.

The summary of the results on the ten data set for the three feature selection methods with Random Forest, AdaBoost and Bagging algorithms, it could be seen that for adult data set Attribute weighting has improved the performance of Random forest and Bagging while Random grouping improved AdaBoost algorithm. It can be distinguish that Attribute weighting method had improved most of the data set performance and not only that, this method had contributed even better results for some data sets(such as Diabetes, Drug consumption, Covertype) with different algorithms. Random grouping method had also better results for some datasets (Drug consumption, Bach Choral harmony) and another data set who had improved performances (Spambase, Census Income, Adult, Diabetes) with different algorithms. At last

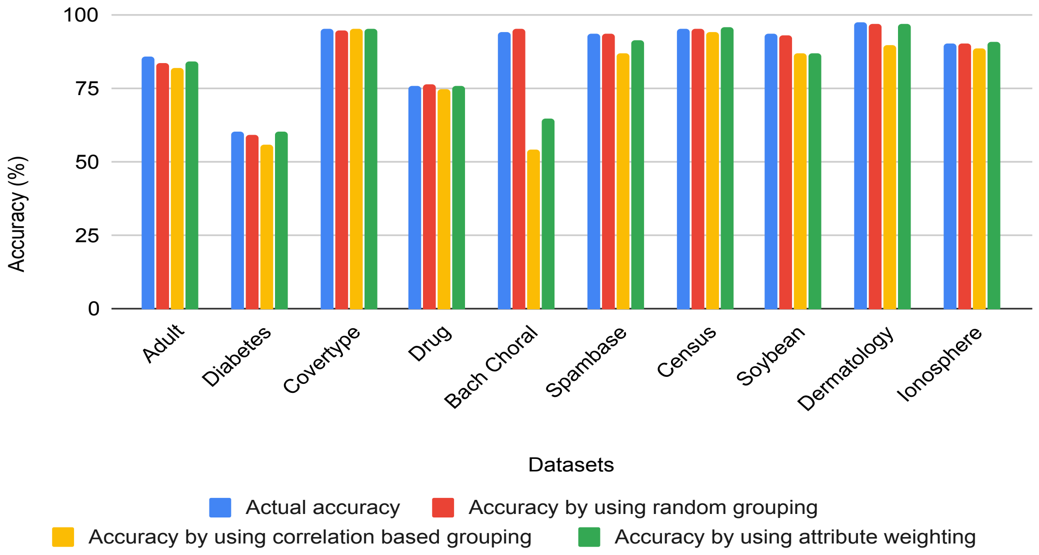


Figure 4.1: Performance of the Random Forest classifier with different feature grouping methods on benchmark datasets.

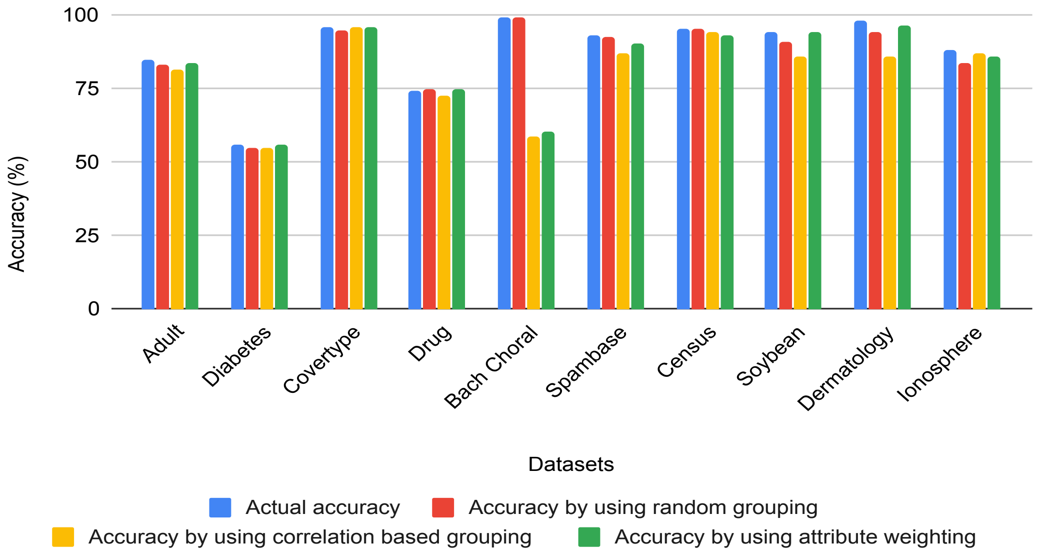


Figure 4.2: Performance of the Bagging classifier with different feature grouping methods on benchmark datasets.

correlation based grouping has least number of data set that have improved result after applying the method. Only Drug consumption had better results and Covertype, Census Income

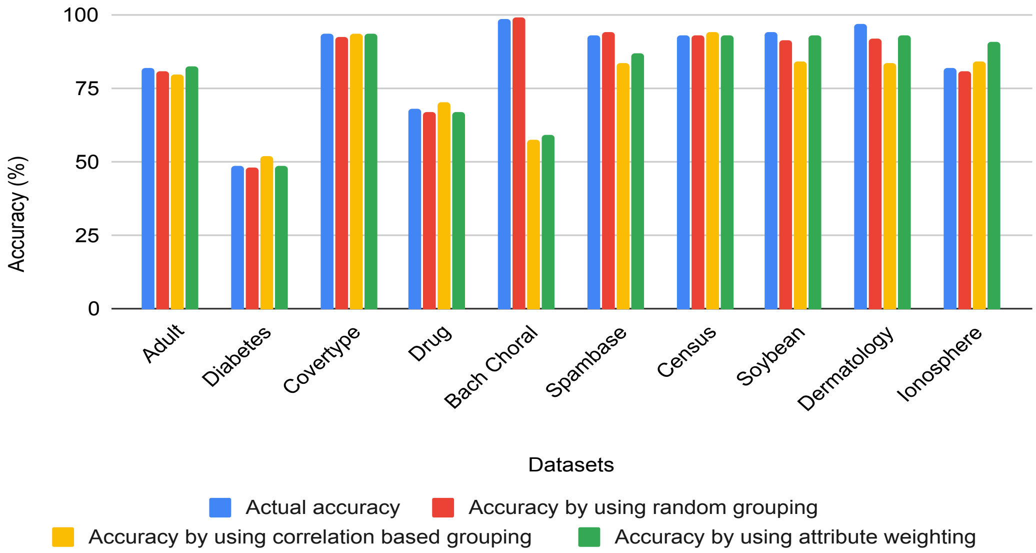


Figure 4.3: Performance of the AdaBoost classifier with different feature grouping methods on benchmark datasets.

have improved results. Rest of the datasets like Ionosphere and Dermatology, the performance is overall same even after reducing features.

Table 4.2: Performance comparison of different types of feature selection methods

FS method	Algo.	Adult	Bach Chor.	Drug Cons.	Derma.	Ionos.	Soy.	Cens. Inc.	Cov.	Diab.	Spam.
CFS	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Chi-Square	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Consistency Sub.	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Cost Sensitive Attr.	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Filtered Attr.	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Gain Ratio	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Info Gain	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
OneR	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
PCA	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
ReliefF	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Symm. Uncertainty	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Wrapper	RF	85.63	92.79	**75.06**	96.69	92.49	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
RFE-SVM	RF	85.63	92.79	**75.06**	96.69	92.49	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
LSA	RF	85.63	92.79	**75.06**	96.69	92.49	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34
Fast Correlation-Based Filter	RF	85.63	92.79	**75.06**	96.69	**92.49**	93.62	92.68	**64.43**	48.25	**89.39**
	AdaB.	**85.70**	**98.87**	68.30	92.56	85.43	**93.80**	92.25	63.51	48.06	86.29
	Bagg.	85.65	98.82	74.10	**97.52**	89.65	92.03	**93.10**	63.59	**48.43**	88.34

Table 4.3: Performance of classifiers on dataset with original feature space

Datasets	Evaluation indices	Random Forest	AdaBoost	Bagging
Adult	Accuracy	**85.63**	81.94	84.57
	Precision	0.85	0.82	0.84
	Recall	0.85	0.82	0.85
	F1-Score	0.85	0.82	0.84
Bach Choral Harmony	Accuracy	94.28	98.72	**99.04**
	Precision	0.94	0.99	0.99
	Recall	0.94	0.99	0.99
	F1-Score	0.94	0.99	0.99
Drug Consumption	Accuracy	**76.05**	68.33	74.15
	Precision	0.65	0.68	0.65
	Recall	0.76	0.68	0.64
	F1-Score	0.68	0.68	0.68
Dermatology	Accuracy	97.52	95.87	**98.35**
	Precision	0.97	0.96	0.98
	Recall	0.97	0.96	0.96
	F1-Score	0.97	0.96	0.96
Ionosphere	Accuracy	**90.52**	81.90	87.93
	Precision	0.90	0.84	0.89
	Recall	0.90	0.82	0.88
	F1-Score	0.90	0.82	0.88
Soybean	Accuracy	**94.25**	**94.25**	**94.25**
	Precision	0.94	0.84	0.94
	Recall	0.94	0.84	0.94
	F1-Score	0.95	0.84	0.94
Census Income	Accuracy	**95.54**	92.95	95.17
	Precision	0.95	0.93	0.95
	Recall	0.96	0.93	0.95
	F1-Score	0.95	0.93	0.95
Covertype	Accuracy	95.04	93.56	**95.72**
	Precision	0.95	0.94	0.96
	Recall	0.95	0.94	0.96
	F1-Score	0.95	0.94	0.96
Diabetes	Accuracy	**60.26**	48.72	55.87
	Precision	0.58	0.49	0.54
	Recall	0.60	0.49	0.56
	F1-Score	0.56	0.49	0.55
Spambase	Accuracy	**93.81**	92.82	92.89
	Precision	0.94	0.93	0.93
	Recall	0.94	0.93	0.93
	F1-Score	0.94	0.93	0.93

Table 4.4: Performance of Random Forest classifier with Feature Selection Methods

Datasets	Evaluation indices	Random grouping Methods	Correlation based grouping	Attribute weighting based grouping
	Accuracy	85.04	82.80	**85.57**
Adult	Precision	0.84	0.81	0.82
	Recall	0.85	0.81	0.85
	F1-Score	0.85	0.82	0.83
	Accuracy	**99.09**	60.48	74.65
Bach Choral Harmony	Precision	0.54	0.53	0.74
	Recall	0.99	0.54	0.73
	F1-Score	0.99	0.60	0.75
	Accuracy	**76.37**	75.88	**76.37**
Drug Consumption	Precision	0.67	0.64	0.65
	Recall	0.69	0.68	0.68
	F1-Score	0.67	0.76	0.76
	Accuracy	96.69	**97.52**	**97.52**
Dermatology	Precision	0.97	0.98	0.98
	Recall	0.97	0.98	0.98
	F1-Score	0.97	0.98	0.98
	Accuracy	91.38	90.52	**91.39**
Ionosphere	Precision	0.91	0.89	0.92
	Recall	0.91	0.91	0.91
	F1-Score	0.91	0.91	0.92
	Accuracy	93.81	93.36	**95.13**
Soybean	Precision	0.94	0.93	0.95
	Recall	0.94	0.93	0.95
	F1-Score	0.94	0.93	0.95
	Accuracy	95.48	95.41	**95.58**
Census Income	Precision	0.95	0.91	0.95
	Recall	0.95	0.95	0.95
	F1-Score	0.95	0.95	0.95
	Accuracy	**95.67**	95.37	95.19
Covertype	Precision	0.96	0.95	0.95
	Recall	0.96	0.95	0.95
	F1-Score	0.96	0.95	0.95
	Accuracy	60.04	59.79	**60.53**
Diabetes	Precision	0.58	0.58	0.59
	Recall	0.56	0.56	0.61
	F1-Score	0.60	0.60	0.56
	Accuracy	**93.68**	90.38	91.50
Spambase	Precision	0.94	0.91	0.92
	Recall	0.94	0.90	0.91
	F1-Score	0.94	0.90	0.92

Table 4.5: Performance of Bagging classifier with Feature Selection Methods

Datasets	Evaluation indices	Random grouping Methods	Correlation based grouping	Attribute weighting based grouping
Adult	Accuracy	84.42	83.81	**84.58**
	Precision	0.84	0.83	0.84
	Recall	0.84	0.83	0.84
	F1-Score	0.84	0.84	0.85
Bach Choral Harmony	Accuracy	**99.30**	60.27	73.21
	Precision	0.99	0.52	0.73
	Recall	0.99	0.55	0.72
	F1-Score	0.99	0.60	0.73
Drug Consumption	Accuracy	75.24	74.12	**74.92**
	Precision	0.67	0.66	0.65
	Recall	0.70	0.69	0.68
	F1-Score	0.75	0.69	0.75
Dermatology	Accuracy	95.87	93.39	**97.52**
	Precision	0.96	0.93	0.98
	Recall	0.96	0.93	0.98
	F1-Score	0.96	0.93	0.98
Ionosphere	Accuracy	84.48	**87.93**	87.09
	Precision	0.87	0.88	0.88
	Recall	0.85	0.88	0.87
	F1-Score	0.84	0.88	0.87
Soybean	Accuracy	92.48	92.48	**95.58**
	Precision	0.93	0.93	0.96
	Recall	0.93	0.93	0.96
	F1-Score	0.92	0.93	0.96
Census Income	Accuracy	**95.20**	95.01	95.19
	Precision	0.94	0.94	0.95
	Recall	0.95	0.94	0.95
	F1-Score	0.95	0.94	0.95
Covertype	Accuracy	**95.85**	95.25	**95.85**
	Precision	0.95	0.95	0.96
	Recall	0.94	0.95	0.96
	F1-Score	0.94	0.95	0.96
Diabetes	Accuracy	55.83	55.24	**55.99**
	Precision	0.53	0.47	0.54
	Recall	0.55	0.47	0.55
	F1-Score	0.56	0.53	0.55
Spambase	Accuracy	**93.28**	90.25	90.90
	Precision	0.93	0.90	0.90
	Recall	0.93	0.90	0.90
	F1-Score	0.93	0.90	0.90

Table 4.6: Performance of Boosting(AdaBoosting) classifier with Feature Selection Methods

Datasets	Evaluation indices	Random grouping Methods	Correlation based grouping	Attribute weighting based grouping
Adult	Accuracy	**82.39**	81.98	80.86
	Precision	0.82	0.81	0.84
	Recall	0.82	0.81	0.84
	F1-Score	0.82	0.81	0.85
Bach Choral Harmony	Accuracy	**99.04**	59.68	68.45
	Precision	0.99	0.51	0.70
	Recall	0.99	0.54	0.69
	F1-Score	0.99	0.60	0.68
Drug Consumption	Accuracy	66.72	**72.02**	68.65
	Precision	0.68	0.63	0.67
	Recall	0.67	0.67	0.67
	F1-Score	0.68	0.72	0.69
Dermatology	Accuracy	textbf94.21	92.56	**94.21**
	Precision	0.94	0.84	0.93
	Recall	0.94	0.93	0.94
	F1-Score	0.94	0.93	0.94
Ionosphere	Accuracy	81.90	**86.21**	81.90
	Precision	0.84	0.86	0.83
	Recall	0.82	0.86	0.82
	F1-Score	0.82	0.86	0.82
Soybean	Accuracy	92.04	92.92	**95.58**
	Precision	0.92	0.93	0.96
	Recall	0.92	0.93	0.96
	F1-Score	0.92	0.93	0.96
Census Income	Accuracy	94.23	**95.04**	93.05
	Precision	0.94	0.94	0.93
	Recall	0.94	0.94	0.93
	F1-Score	0.94	0.95	0.93
Covertype	Accuracy	93.67	92.58	**93.63**
	Precision	0.94	0.93	0.94
	Recall	0.94	0.93	0.94
	F1-Score	0.94	0.93	0.94
Diabetes	Accuracy	48.64	53.80	**58.82**
	Precision	0.49	0.47	0.49
	Recall	0.49	0.38	0.49
	F1-Score	0.49	0.54	0.49
Spambase	Accuracy	**94.40**	86.43	87.55
	Precision	0.94	0.87	0.86
	Recall	0.94	0.86	0.86
	F1-Score	0.94	0.86	0.86

Table 4.7: Evaluation of Classifiers with Attribute Weighting Grouping method

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Adult	Feature group 1	Accuracy	83.06	**80.86**	**84.58**
		Precision	0.85	0.80	0.84
		Recall	0.85	0.80	0.84
		F1-Score	0.85	0.80	0.85
	Feature group 2	Accuracy	84.58	80.40	83.41
		Precision	0.83	0.80	0.83
		Recall	0.84	0.80	0.83
		F1-Score	0.84	0.80	0.83
	Feature group 3	Accuracy	**85.57**	80.59	82.75
		Precision	0.82	0.80	0.82
		Recall	0.85	0.80	0.82
		F1-Score	0.83	0.80	0.83
Bach Choral Harmony	Feature group 1	Accuracy	**74.65**	**68.45**	**73.21**
		Precision	0.74	0.70	0.73
		Recall	0.73	0.69	0.72
		F1-Score	0.75	0.68	0.73
	Feature group 2	Accuracy	65.29	60.16	54.28
		Precision	0.63	0.61	0.64
		Recall	0.65	0.60	0.65
		F1-Score	0.65	0.60	0.65
	Feature group 2	Accuracy	53.69	48.88	54.28
		Precision	0.52	0.61	0.53
		Recall	0.52	0.60	0.53
		F1-Score	0.54	0.48	0.54
Drug Consumption	Feature group 1	Accuracy	**76.37**	66.88	**74.92**
		Precision	0.65	0.70	0.65
		Recall	0.68	0.66	0.68
		F1-Score	0.76	0.66	0.75
	Feature group 2	Accuracy	75.88	**68.65**	74.44
		Precision	0.65	0.67	0.68
		Recall	0.68	0.67	0.69
		F1-Score	0.76	0.69	0.74
	Feature group 2	Accuracy	76.21	66.08	74.28
		Precision	0.73	0.66	0.65
		Recall	0.70	0.66	0.68
		F1-Score	0.77	0.65	0.74
Dermatology	Feature group 1	Accuracy	**97.52**	93.39	**97.52**
		Precision	0.98	0.93	0.98
		Recall	0.98	0.94	0.98
		F1-Score	0.98	0.94	0.98
	Feature group 2	Accuracy	96.70	**94.21**	96.69
		Precision	0.97	0.93	0.97
		Recall	0.97	0.94	0.97
		F1-Score	0.97	0.94	0.97
	Feature group 3	Accuracy	96.70	91.74	95.04
		Precision	0.97	0.93	0.96
		Recall	0.97	0.92	0.95
		F1-Score	0.97	0.92	0.95

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Ionosphere	Feature group 1	Accuracy	**91.38**	**81.90**	**87.07**
		Precision	0.92	0.83	0.88
		Recall	0.91	0.82	0.87
		F1-Score	0.91	0.82	0.87
	Feature group 2	Accuracy	**91.38**	78.45	85.34
		Precision	0.92	0.81	0.87
		Recall	0.91	0.79	0.87
		F1-Score	0.92	0.78	0.85
	Feature group 2	Accuracy	90.52	80.17	85.34
		Precision	0.91	0.82	0.87
		Recall	0.91	0.83	0.86
		F1-Score	0.91	0.80	0.85
Soybean	Feature group 1	Accuracy	**95.13**	**95.58**	**95.58**
		Precision	0.95	0.96	0.96
		Recall	0.95	0.96	0.96
		F1-Score	0.95	0.96	0.96
	Feature group 2	Accuracy	92.92	92.48	93.36
		Precision	0.93	0.93	0.94
		Recall	0.93	0.93	0.94
		F1-Score	0.93	0.92	0.94
	Feature group 3	Accuracy	93.81	91.15	93.36
		Precision	0.93	0.92	0.94
		Recall	0.93	0.91	0.93
		F1-Score	0.93	0.91	0.93
Census Income	Feature group 1	Accuracy	95.56	92.93	**95.19**
		Precision	0.95	0.93	0.95
		Recall	0.95	0.93	0.95
		F1-Score	0.96	0.93	0.95
	Feature group 2	Accuracy	**95.58**	**93.05**	95.17
		Precision	0.95	0.93	0.94
		Recall	0.95	0.93	0.95
		F1-Score	0.96	0.93	0.95
	Feature group 2	Accuracy	95.57	93.00	95.17
		Precision	0.95	0.93	0.94
		Recall	0.95	0.93	0.95
		F1-Score	0.96	0.93	0.95
Covertype	Feature group 1	Accuracy	**95.19**	**93.63**	**95.74**
		Precision	0.95	0.94	0.96
		Recall	0.95	0.94	0.96
		F1-Score	0.95	0.94	0.96
	Feature group 2	Accuracy	95.12	93.41	95.68
		Precision	0.95	0.93	0.96
		Recall	0.95	0.93	0.96
		F1-Score	0.95	0.93	0.96
	Feature group 3	Accuracy	95.14	93.23	95.73
		Precision	0.95	0.93	0.96
		Recall	0.95	0.93	0.96
		F1-Score	0.95	0.93	0.96

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Diabetes	Feature group 1	Accuracy	60.42	**58.82**	55.77
		Precision	0.58	0.49	0.54
		Recall	0.60	0.49	0.56
		F1-Score	0.56	0.49	0.54
	Feature group 2	Accuracy	**60.53**	48.78	55.69
		Precision	0.59	0.49	0.53
		Recall	0.61	0.48	0.56
		F1-Score	0.56	0.49	0.54
	Feature group 3	Accuracy	60.39	48.83	**55.99**
		Precision	0.58	0.49	0.54
		Recall	0.60	0.49	0.55
		F1-Score	0.56	0.49	0.55
Spambase	Feature group 1	Accuracy	**91.50**	8**7.55**	**90.90**
		Precision	0.93	0.88	0.90
		Recall	0.93	0.87	0.90
		F1-Score	0.93	0.88	0.90
	Feature group 2	Accuracy	**91.50**	**87.55**	89.98
		Precision	0.92	0.86	0.90
		Recall	0.91	0.86	0.90
		F1-Score	0.92	0.86	0.90
	Feature group 2	Accuracy	90.90	86.10	89.33
		Precision	0.90	0.86	0.89
		Recall	0.90	0.86	0.89
		F1-Score	0.90	0.86	0.89

Table 4.8: Evaluation of Classifiers with Correlation-based Grouping method

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Adult	Feature group 1	Accuracy	78.93	74.73	**83.81**
		Precision	0.75	0.73	0.83
		Recall	0.75	0.74	0.83
		F1-Score	0.76	0.75	0.84
	Feature group 2	Accuracy	**82.20**	**81.98**	79.24
		Precision	0.81	0.81	0.76
		Recall	0.81	0.81	0.78
		F1-Score	0.82	0.82	0.80
	Feature group 3	Accuracy	75.96	78.97	-
		Precision	0.77	0.77	-
		Recall	0.78	0.78	-
		F1-Score	0.79	0.79	-

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Bach Choral Harmony	Feature group 1	Accuracy	58.29	55.13	57.17
		Precision	0.56	0.56	0.57
		Recall	0.56	0.55	0.56
		F1-Score	0.58	0.55	0.57
	Feature group 2	Accuracy	**60.48**	**59.68**	**60.27**
		Precision	0.54	0.51	0.52
		Recall	0.54	0.54	0.55
		F1-Score	0.60	0.60	0.60
	Feature group 2	Accuracy	-	-	-
		Precision	-	-	-
		Recall	-	-	-
		F1-Score	-	-	-
Drug Consumption	Feature group 1	Accuracy	**75.88**	69.13	**74.12**
		Precision	0.64	0.68	0.66
		Recall	0.68	0.68	0.69
		F1-Score	0.76	0.69	0.69
	Feature group 2	Accuracy	73.63	**72.02**	71.38
		Precision	0.66	0.63	0.66
		Recall	0.68	0.67	0.69
		F1-Score	0.74	0.72	0.71
	Feature group 3	Accuracy	-	-	-
		Precision	-	-	-
		Recall	-	-	-
		F1-Score	-	-	-
Dermatology	Feature group 1	Accuracy	**97.52**	**92.56**	**93.39**
		Precision	0.98	0.94	0.95
		Recall	0.98	0.93	0.93
		F1-Score	0.98	0.93	0.93
	Feature group 2	Accuracy	81.82	75.21	78.51
		Precision	0.83	0.76	0.80
		Recall	0.82	0.75	0.79
		F1-Score	0.82	0.75	0.79
	Feature group 3	Accuracy	-	-	-
		Precision	-	-	-
		Recall	-	-	-
		F1-Score	-	-	-
Ionosphere	Feature group 1	Accuracy	89.66	**86.21**	85.34
		Precision	0.90	0.86	0.87
		Recall	0.90	0.86	0.86
		F1-Score	0.90	0.86	0.85
	Feature group 2	Accuracy	**90.52**	82.76	87.07
		Precision	0.91	0.83	0.87
		Recall	0.91	0.83	0.87
		F1-Score	0.91	0.83	0.87
	Feature group 2	Accuracy	86.21	84.48	**87.93**
		Precision	0.86	0.84	0.88
		Recall	0.86	0.84	0.88
		F1-Score	0.86	0.84	0.88

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Soybean	Feature group 1	Accuracy	**93.36**	**92.92**	**92.48**
		Precision	0.94	0.93	0.93
		Recall	0.93	0.93	0.93
		F1-Score	0.93	0.93	0.92
	Feature group 2	Accuracy	80.53	75.22	79.65
		Precision	0.81	0.76	0.79
		Recall	0.80	0.75	0.79
		F1-Score	0.80	0.75	0.80
	Feature group 2	Accuracy	-	-	-
		Precision	-	-	-
		Recall	-	-	-
		F1-Score	-	-	-
Census Income	Feature group 1	Accuracy	**95.41**	**95.04**	**95.01**
		Precision	0.95	0.94	0.94
		Recall	0.95	0.94	0.94
		F1-Score	0.95	0.95	0.95
	Feature group 2	Accuracy	93.93	93.81	93.86
		Precision	0.92	0.91	0.92
		Recall	0.92	0.92	0.92
		F1-Score	0.94	0.94	0.94
	Feature group 2	Accuracy	93.92	93.89	93.91
		Precision	0.88	0.88	0.88
		Recall	0.91	0.91	0.91
		F1-Score	0.94	0.94	0.94
Covertype	Feature group 1	Accuracy	**95.37**	**92.58**	**95.25**
		Precision	0.95	0.93	0.95
		Recall	0.95	0.93	0.95
		F1-Score	0.95	0.93	0.95
	Feature group 2	Accuracy	53.26	49.28	56.93
		Precision	0.54	0.49	0.57
		Recall	0.41	0.46	0.57
		F1-Score	0.53	0.49	0.52
	Feature group 3	Accuracy	53.26	53.26	53.26
		Precision	0.54	0.54	0.41
		Recall	0.41	0.40	0.41
		F1-Score	0.53	0.53	0.53
Diabetes	Feature group 1	Accuracy	**59.79**	48.99	**55.24**
		Precision	0.58	0.50	0.53
		Recall	0.56	0.49	0.54
		F1-Score	0.60	0.49	0.55
	Feature group 2	Accuracy	53.91	**53.27**	**55.24**
		Precision	0.48	0.47	0.47
		Recall	0.47	0.47	0.47
		F1-Score	0.54	0.53	0.53
	Feature group 3	Accuracy	53.87	**53.8**	53.86
		Precision	0.47	0.47	0.47
		Recall	0.38	0.38	0.38
		F1-Score	0.54	0.54	0.54

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
	Feature group 1	Accuracy	87.09	83.20	86.03
		Precision	0.87	0.84	0.86
		Recall	0.87	0.83	0.86
		F1-Score	0.87	0.83	0.86
Spambase	Feature group 2	Accuracy	84.91	81.29	84.52
		Precision	0.85	0.81	0.85
		Recall	0.85	0.85	0.84
		F1-Score	0.85	0.81	0.85
	Feature group 3	Accuracy	**90.38**	**86.43**	**90.25**
		Precision	0.91	0.87	0.90
		Recall	0.90	0.86	0.90
		F1-Score	0.90	0.86	0.90

Table 4.9: Evaluation of Classifiers with Random Grouping method

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
	Feature group 1	Accuracy	81.38	79.15	80.85
		Precision	0.80	0.78	0.80
		Recall	0.81	0.79	0.80
		F1-Score	0.81	0.79	0.80
Adult	Feature group 2	Accuracy	84.68	**82.39**	83.42
		Precision	0.85	0.82	0.83
		Recall	0.84	0.82	0.83
		F1-Score	0.85	0.82	0.83
	Feature group 3	Accuracy	**85.04**	81.22	**84.42**
		Precision	0.84	0.81	0.84
		Recall	0.85	0.81	0.84
		F1-Score	0.85	0.81	0.84
	Feature group 1	Accuracy	93.58	98.98	**99.30**
		Precision	0.94	0.99	0.99
		Recall	0.93	0.99	0.99
		F1-Score	0.96	0.99	0.99
Bach Choral Harmony	Feature group 2	Accuracy	93.53	**99.04**	99.14
		Precision	0.93	0.99	0.99
		Recall	0.94	0.99	0.99
		F1-Score	0.93	0.99	0.99
	Feature group 2	Accuracy	**99.09**	98.82	99.19
		Precision	0.99	0.99	0.99
		Recall	0.99	0.99	0.99
		F1-Score	0.99	0.99	0.99

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Drug Consumption	Feature group 1	Accuracy	**76.37**	66.40	**75.24**
		Precision	0.67	0.65	0.67
		Recall	0.69	0.66	0.70
		F1-Score	0.76	0.66	0.75
	Feature group 2	Accuracy	76.05	**66.72**	75.24
		Precision	0.65	0.68	0.67
		Recall	0.68	0.67	0.70
		F1-Score	0.76	0.68	0.75
	Feature group 2	Accuracy	**76.37**	66.65	74.60
		Precision	0.64	0.65	0.66
		Recall	0.68	0.68	0.68
		F1-Score	0.76	0.69	0.75
Dermatology	Feature group 1	Accuracy	**96.69**	90.91	91.74
		Precision	0.97	0.91	0.93
		Recall	0.97	0.91	0.92
		F1-Score	0.97	0.91	0.92
	Feature group 2	Accuracy	**96.69**	**94.21**	**95.87**
		Precision	0.97	0.95	0.96
		Recall	0.97	0.94	0.96
		F1-Score	0.97	0.94	0.96
	Feature group 2	Accuracy	97.52	90.91	95.04
		Precision	0.98	0.91	0.95
		Recall	0.98	0.91	0.95
		F1-Score	0.98	0.91	0.95
Ionosphere	Feature group 1	Accuracy	90.52	81.03	85.34
		Precision	0.91	0.79	0.87
		Recall	0.91	0.81	0.86
		F1-Score	0.91	0.81	0.85
	Feature group 2	Accuracy	**91.38**	79.31	81.03
		Precision	0.91	0.79	0.82
		Recall	0.91	0.79	0.81
		F1-Score	0.91	0.79	0.81
	Feature group 3	Accuracy	89.66	**81.90**	**84.48**
		Precision	0.90	0.84	0.87
		Recall	0.90	0.82	0.85
		F1-Score	0.90	0.82	0.84
Soybean	Feature group 1	Accuracy	**93.81**	90.71	91.15
		Precision	0.94	0.89	0.92
		Recall	0.94	0.88	0.91
		F1-Score	0.94	0.88	0.91
	Feature group 2	Accuracy	92.04	90.71	89.38
		Precision	0.93	0.93	0.90
		Recall	0.92	0.91	0.89
		F1-Score	0.92	0.91	0.89
	Feature group 3	Accuracy	93.36	**92.04**	**92.48**
		Precision	0.94	0.92	0.93
		Recall	0.93	0.92	0.93
		F1-Score	0.93	0.92	0.92

Dataset	Feature groups	Evaluation indices	Random Forest	AdaBoost	Bagging
Census Income	Feature group 1	Accuracy	**95.48**	92.83	**95.20**
		Precision	0.95	0.93	0.95
		Recall	0.95	0.93	0.95
		F1-Score	0.95	0.93	0.95
	Feature group 2	Accuracy	94.91	**94.23**	94.76
		Precision	0.94	0.94	0.94
		Recall	0.94	0.94	0.94
		F1-Score	0.95	0.94	0.95
	Feature group 2	Accuracy	95.41	92.91	**95.20**
		Precision	0.95	0.93	0.94
		Recall	0.95	0.93	0.95
		F1-Score	0.95	0.93	0.95
Covertype	Feature group 1	Accuracy	95.22	93.31	95.72
		Precision	0.95	0.93	0.96
		Recall	0.95	0.93	0.96
		F1-Score	0.95	0.93	0.96
	Feature group 2	Accuracy	**95.67**	**93.67**	**95.85**
		Precision	0.96	0.94	0.96
		Recall	0.96	0.94	0.96
		F1-Score	0.96	0.94	0.96
	Feature group 2	Accuracy	92.55	90.49	92.61
		Precision	0.93	0.90	0.93
		Recall	0.92	0.90	0.93
		F1-Score	0.93	0.90	0.93
Diabetes	Feature group 1	Accuracy	**60.04**	**48.69**	**55.83**
		Precision	0.58	0.49	0.53
		Recall	0.56	0.49	0.55
		F1-Score	0.60	0.49	0.56
	Feature group 2	Accuracy	**60.04**	48.59	55.44
		Precision	0.58	0.49	0.54
		Recall	0.56	0.49	0.55
		F1-Score	0.60	0.49	0.55
	Feature group 3	Accuracy	57.75	46.30	52.75
		Precision	0.55	0.47	0.50
		Recall	0.53	0.47	0.51
		F1-Score	0.58	0.46	0.53
Spambase	Feature group 1	Accuracy	93.61	94.14	91.63
		Precision	0.94	0.94	0.92
		Recall	0.94	0.94	0.92
		F1-Score	0.94	0.94	0.92
	Feature group 2	Accuracy	93.61	**94.40**	**93.28**
		Precision	0.94	0.94	0.93
		Recall	0.94	0.94	0.93
		F1-Score	0.94	0.94	0.93
	Feature group 3	Accuracy	**93.68**	93.81	92.42
		Precision	0.94	0.94	0.90
		Recall	0.94	0.94	0.92
		F1-Score	0.94	0.94	0.92

Chapter 5

Conclusion and Future Work

This chapter represents the discussions of the thesis work, focuses on the limitations and highlights the future works.

5.1 Discussion

For the research work, reducing the unrelated and less important features which do not have any contribution to the performances was the main objective. The that depends on data analyse can not afford the time or may not have proper storage to compute the immense level of data that in this real world have been producing. At the same time, in many cases when feature subsets are applied for the machine learning algorithms, the computation time takes too long and sometimes it can not be properly used in machines. Even sometimes the correlation between the features are too high to find out the real hidden pattern from data. So to handle this problem the focus is on working with feature importance. Feature importance ranking and filter system is used for datasets and grouped for analysis. For 'Attribute Weighting' grouping the weights of each and every feature from data have been calculated by applying feature importance technique. Then lowest weighted features are eliminated by 10% of the total data for the groups. Different ensemble learning algorithms like Random Forest, Bagging and AdaBoost have been used to measure the accuracy results after applying this proposed methods. The results of two other group based methods such as correlation based grouping and random grouping of features are also have been compared. The main objective was to find out that if the proposed groups of feature are applied with machine learning algorithms then how similar results or better accuracy results are produced. So, the actual accuracy result of the data is also measuerd to compare the performances. For the experiments and better analysis, literature of various types of feature selection methods is studied.

The main challenge of this task was to determine the number of features that are going to eliminate. After applying some iteration with groups of features, the experiment reached to the point that 10% of the total features can be eliminated to represent a stable performance. If the numbers of feature are lessen more than 10% then the accuracy result degraded. Again, when number of feature features remain less then 65%-70% of the total feature set and then features are eliminated to form a group, the result for that feature group was inclined in a drastic way. This characteristics depends on the number of features of selected datasets. So for attribute weighting there were maximum three groups formed for the experiment and then average were calculated from the results. For correlation attribute weighting the groups were formed according to the number of groups that had closely related features. For some datasets the features were so close that single group had to be formed based on the correlation. However, for random grouping of features, keeping the number aligned with attribute weighting 10% of the features were eliminated randomly and then averages of the results were calculated. After applying all the three group based method with the ensemble learning algorithms it can be seen that the proposed Attribute weighting grouping method has overall similar performance after eliminating the lowest important features. In some cases, for few datasets this method outperformed the actual accuracy of the datasets. From the experiment results it can be said that Adult and Ionosphere datasets have outperformed for attribute weighting technique with AdaBoost algorithm. And Diabetes and Drug Consumption datasets have outperformed for attribute weighting technique with Random forest algorithm.

5.2 Conclusion

According to the experimental analysis, the conclusion can be made that, after attribute weighting reducing the lowest important features and gathering them into groups show similar or slightly better results. In this research, attribute weighting based grouping have been introduced for feature selection along with two other feature selection methods named Random Grouping and Correlation based Grouping. Different types of datasets have been experimented for the research. The experiment shows that less important features can be removed to get better accuracy results for large datasets without loss of much properties of data. The result of the present study shows that the Attribute Weighting group method has the overall better performances and provides a promising performances for the independent data sets and algorithms.

5.3 Future Work

There are certainly many possibilities for research in future. In future work, we would like to make these methods more powerful with the fusion of the hybrid feature selection methods and have larger data sets to analyze scalability. The future work will develop a method that can automatically divide features into groups in the weighted clustering process. Finally, we will implement and improve the method on further large streams of real-time data.

Bibliography

[1] "Global datasphere to hit 175 zettabytes by 2025, idc," https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper, November 27, 2018. vii, 9

[2] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody, "Critical analysis of big data challenges and analytical methods," *Journal of Business Research*, vol. 70, pp. 263–286, 2017. 1

[3] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big data technologies: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018. 1

[4] J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9–15, 2017. 1, 2

[5] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big iot data analytics: architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017. 1

[6] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017. 2

[7] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, 2017. 2

[8] K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: constraint, relevance, and redundancy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1131–1143, 2013. 2

[9] M. Reif and F. Shafait, "Efficient feature size reduction via predictive forward selection," *Pattern Recognition*, vol. 47, no. 4, pp. 1664–1673, 2014. 2

[10] V. Kumar and S. Minz, "Feature selection: a literature review," *SmartCR*, vol. 4, no. 3, pp. 211–229, 2014. 4

[11] Y. L. andTao Li and H. Liu, "Recent advances in feature selection and its applications," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 551–577, 2017. 4, 5, 7, 10

[12] M. S. Pervez and D. M. Farid, "Literature review of feature selection for mining tasks," *International Journal of Computer Applications*, vol. 116, no. 21, 2015. 5

[13] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007. 5, 8

[14] V. Bolón-Canedo, D. Rego-Fernández, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, and N. Sánchez-Maroño, "On the scalability of feature selection methods on high-dimensional data," *Knowledge and Information Systems*, vol. 56, pp. 395–442, 2018. 5, 6, 7, 8, 11, 12, 15

[15] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018. 5

[16] B. Al-Salemi, M. Ayob, and S. A. M. Noah, "Feature ranking for enhancing boosting-based multi-label text categorization," *Expert Systems with Applications*, vol. 113, pp. 531–543, 2018. 6

[17] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowledge-Based Systems*, vol. 140, pp. 103–119, 2018. 6

[18] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018. 6

[19] D. Jain and V. Singh, "An efficient hybrid feature selection model for dimensionality reduction," *Procedia Computer Science*, vol. 132, pp. 333–341, 2018. 7, 15

[20] C. A. Kumar, M. Sooraj, and S. Ramakrishnan, “A comparative performance evaluation of supervised feature selection algorithms on microarray datasets,” *Procedia computer science*, vol. 115, pp. 209–217, 2017. 7

[21] B. Zhang, P. Cao, Y. Zhang, C. Zhang, Z. Li, Z. Qu, X. Wang, T. Iei, H. Cai, and B. Hu, “Feature selection algorithm for high dimensional biomedical data classification based on redundant removal,” in *Proceedings of the 32nd International BCS Human Computer Interaction Conference*. BCS Learning & Development Ltd., 2018, p. 231. 7

[22] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014. 8, 13, 15

[23] G. Manikandan, E. Susi, and S. Abirami, “Feature selection on high dimensional data using wrapper based subset selection,” in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*. IEEE, 2017, pp. 320–325. 8

[24] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, “The role of big data analytics in internet of things,” *Computer Networks*, vol. 129, pp. 459–471, 2017. 9

[25] E. Savitz, “Gartner: 10 critical tech trends for the next five years,” *October2012¡ http://www. forbes. com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five years*, 2012. 9

[26] S. Jahan, S. Shatabda, and D. M. Farid, “Active learning for mining big data,” in *2018 21st International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2018, pp. 1–6. 10

[27] A. S. Abdullah, C. Ramya, V. Priyadharsini, C. Reshma, and S. Selvakumar, “A survey on evolutionary techniques for feature selection,” in *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*. IEEE, 2017, pp. 58–62. 10

[28] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, “Big data preprocessing: methods and prospects,” *Big Data Analytics*, vol. 1, no. 1, p. 9, 2016. 10

[29] R. Gupta, J. Pujara, C. A. Knoblock, S. M. Sharanappa, B. Pulavarti, G. Hoberg, and G. Phillips, "Feature selection methods for understanding business competitor relationships," in *Proceedings of the Fourth International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets.* ACM, 2018, p. 2. 10

[30] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003. 11

[31] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification," *Applied Soft Computing*, vol. 62, pp. 203–215, 2018. 11

[32] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class svm," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017. 12

[33] A. J. Tallón-Ballesteros, J. Riquelme, and R. Ruiz, "Merging subsets of attributes to improve a hybrid consistency-based filter: a case of study in product unit neural networks," *Connection Science*, vol. 28, no. 3, pp. 242–257, 2016. 12

[34] S. Gnanambal, M. Thangaraj, V. Meenatchi, and V. Gayathri, "Classification algorithms with attribute selection: an evaluation study using weka," *International Journal of Advanced Networking and Applications*, vol. 9, no. 6, pp. 3640–3644, 2018. 12

[35] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 57–78, 2018. 13

[36] B. Azhagusundari and A. S. Thanamani, "Feature selection based on information gain," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 2, pp. 18–21, 2013. 13

[37] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms," *Expert Systems with Applications*, vol. 80, pp. 323–339, 2017. 13

[38] A. G. Karegowda, A. Manjunath, and M. Jayaram, "Comparative study of attribute selection using gain ratio and correlation based feature selection," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 271–277, 2010. 13

[39] T. K. Landauer, P. W. Foltz, and D. Laham, “An introduction to latent semantic analysis,” *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998. 14

[40] B. Forthmann, O. Oyebade, A. Ojo, F. Günther, and H. Holling, “Application of latent semantic analysis to divergent thinking is biased by elaboration,” *The Journal of Creative Behavior*, vol. 0, no. 0, pp. 1–17, 2018. 14

[41] F. Alam and S. Pachauri, “Detection using weka,” *Advances in Computational Sciences and Technology*, vol. 10, no. 6, pp. 1731–1743, 2017. 14

[42] R. K. Lama, J. Gwak, J.-S. Park, and S.-W. Lee, “Diagnosis of alzheimer’s disease based on structural mri images using a regularized extreme learning machine and pca features,” *Journal of healthcare engineering*, vol. 2017, no. 5485080, pp. 1–11, 2017. 14

[43] A. Moayedikia, K.-L. Ong, Y. L. Boo, W. G. Yeoh, and R. Jensen, “Feature selection for high dimensional imbalanced class data using harmony search,” *Engineering Applications of Artificial Intelligence*, vol. 57, pp. 38–49, 2017. 14

[44] I. Kononenko, “Estimating attributes: analysis and extensions of relief,” in *European conference on machine learning.* Springer, 1994, pp. 171–182. 15

[45] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml 21